Geometry of Linear Systems and Identification

Chun Tung Chou
Trinity College
Cambridge

March 1994

A dissertation submitted for the
degree of Doctor of Philosophy

In Memory of My Parents

“All men are like grass,
and all their glory is like the flowers of the field.
The grass withers and the flowers fall,
but the word of our God stands for ever.”
(Isaiah 40:6,8)

Preface

I would like to thank my supervisor Dr. Jan Maciejowski for his constant guidance and encouragement during the last three and a half years. It is also my pleasure to thank all my colleagues in the control group: Alan, Anita, Brian, Feng, Frank, Gavin, Geir, Haig, John, Markus, Phil, Rick, Rob, Sanj, Sharon, Simon, Ton and YC; for making life at the department so enjoyable and for putting up with my idiosyncrasies.

During the Easter term of 1992, I visited the Department of Econometrics at the Free University in Amsterdam, where I completed most of the work in the first part of this dissertation. I would like to thank Dr. Ben Hanzon for his hospitality and for giving me the inspiration for this piece of work. Sincere gratitude also goes to Ralf Peeters for being such a faithful companion for lunch and for introducing me to the life in the Netherlands.

I would like to thank SAST Ltd. and Cambridge Control Ltd. for providing me with the data for the two case studies in the second part of this dissertation; to Peter van Overschee and Bart De Moor of the Katholieke Universiteit Leuven, Belgium for providing me with a free copy of their favourite "N4SID" algorithm.

Life as a PhD student in this cosmopolitan University is challenging, inspiring and also enriching. It helps me to realize that people from different parts of this world do share the same longing for love and acceptance despite all the differences that we see on the surface. I wish to thank all my friends near and far, the Christian community in Cambridge and my family for walking with me through this adventurous life in Cambridge.

Lastly, I would like to acknowledge the financial support from Trinity College, Cambridge in the form of an External Research Studentship; and from the Committee of Vice-Chancellors and Principals of the Universities of the United Kingdom in the form of an ORS Award. I have also received financial help to attend conferences and workshops from Trinity College, the European Research Network in System Identification (ERNSI) and the Cambridge University Engineering Department.

As required by University Statute, I hereby declare that this dissertation is not substantially the same as any that I have submitted for a degree at any other university, is the result of my own work, and includes nothing which is the outcome of work done in collaboration.

Chun Tung Chou
Trinity College
Cambridge
March 15, 1994

Contents

Chapter 1

Introduction

1.1 Overview

System identification is concerned with the problem of building models of dynamical systems based on input/output measurements. The concept of a system plays an important role in many subjects, such as control theory, communication theory, econometrics and operations research. These subjects tell us how to design regulators and filters, or how to make predictions or decisions, based on a model of the system involved. For these purposes, a model can simply be a mental picture which gives us some intuition on the underlying mechanism of the system, or it can be a very complete mathematical description which accurately describes every aspect of the system behaviour.

Most real life processes are very complex and cannot be modelled perfectly. Also, the data collected are usually corrupted by noise, and this makes finding an accurate model of the system impossible. However, for most applications, an approximate description of the system is very often found to be sufficient. In the case of feedback controller design, good closed-loop performance can usually be obtained by using a 'sufficiently' accurate model.

Before a model can be found, we first need to specify the type of model to be used. The discussion here will be confined to finite-dimensional linear time-invariant (FDLTI) models only. This type of model is chosen because it is well studied and it provides a good approximation to many real life processes. A particular approach to calculate a FDLTI model from input/output data is known as parameter estimation [40, 64]. With this approach, we first need to specify a set of models, known as the model set, from which a final model is to be chosen. We then select the model which minimizes a model accuracy criterion, which quantitatively describes the mismatch between the data and the model. This selection is usually done by some kind of gradient search optimization technique, and in order to do this a parametrization for the model set is needed. A parametrization can be seen as a way of assigning every model in the model set to a point in Euclidean space. It can be shown that the problem of parametrizing a model set is closely related to the geometry of the model set itself.

The model sets which are commonly used in system identification are drawn either from the set of FDLTI systems or its subsets which consist of systems with certain specific properties such as asymptotic stability or positive realness. We shall refer to a set of FDLTI systems with certain specific properties as a class of linear systems. Individual geometric properties of a number of classes of linear systems are well studied, as can be seen by the amount of literature published in this area. A quick examination of these results shows that these classes of linear systems possess some geometric similarities. A particular type of geometric equivalence is known as diffeomorphism. If two classes of linear systems are diffeomorphic to each other, then they possess identical topological and differential geometric properties. We shall prove in the first part of this thesis that a number of classes of linear systems are diffeomorphic to each other. The results here are an extension to the work done by Delchamps [16] and Hanzon [24, 26] on diffeomorphisms between classes of linear systems.

In the second part of this thesis, we shall look at the issues related to using a particular type of parametrization, namely the balanced parametrization, which is derived from balanced state space realizations [46], for system identification. The use of balanced realizations for system identification was first suggested by Maciejowski [41] because of its model reduction properties, and it was also conjectured that a parameter estimator using these realizations would have a certain optimal property. Subsequently, Ober derived explicit parametrizations for balanced realizations of continuous time stable FDLTI systems [53] and also of several other classes of continuous time FDLTI systems [54]. The results presented here thus form part of a continuing effort in applying balanced parametrizations in system identification, and the emphasis here will be on the identification of multivariable stable systems. The reader may refer to [42] for an overview of this research programme and to [44, 43] for a discussion on the use of balanced parametrizations of stable and minimum-phase systems for time-series analysis.

1.2 Outline

In this section, we shall give a general outline of the rest of the dissertation. This dissertation consists of four parts, the first two of which contain the main contributions. Individual chapter descriptions can be found in the introductions of these two parts.

1.2.1 Part I: Diffeomorphisms between classes of linear systems

In this part of the thesis we shall prove, in detail, that a number of classes of linear systems are diffeomorphic to each other. We shall also prove the diffeomorphisms between the state bundles associated with these classes of linear systems and between their associated principal fibre bundles. This part is concluded with some applications of these geometric results to system identification.

1.2.2 Part II: System identification using balanced parametrizations

In this part of the thesis, we shall present an algorithm for the identification of state space models using balanced parametrizations of stable systems. The algorithm consists of two steps. First, an initial estimate of the state space parameters is obtained from the input/output data using either model reduction based identification methods or subspace methods. We then find an optimal model using parameter optimization with balanced canonical forms as the parametrization. Two case studies on applying the proposed algorithm to two sets of industrial data will be presented. The first set of data comes from a very detailed simulation model of a fractional distillation column. The second set comes from an industrial drying process.

1.2.3 Part III: Conclusions

This part contains a summary of the main contributions of this dissertation together with some recommendations for future work.

1.2.4 Part IV: Appendices

This part consists of two appendices. We shall prove some of the results from Part II in the first appendix. The second appendix presents an algorithm for scaling the input/output data.

Part I

Diffeomorphisms between classes of linear systems

Chapter 2

Introduction

2.1 Overview

Geometry and linear system theory are two very closely related subjects. Many objects in linear system theory are geometric in nature and many of the concepts can be given geometric interpretations. Also, geometry has always been an important tool for solving problems in linear system theory. As an example, the concepts of controllability and observability of linear state space systems can be interpreted using the theory of invariant subspaces. Many other examples can be given and they all illustrate the rich connections between geometry and linear system theory.

Geometry has also found a natural way into the field of system identification. The goal of system identification is to choose from the model set, which is the set of all admissible models, a model which accurately describes the process. In order to measure the accuracy of a model, we need the concept of the 'distance' between a model and a process. The idea of 'distance' can be given the geometric interpretation of a metric in the space of systems. A particular way to obtain a model in system identification is by parameter estimation, which, in simple terms, can be looked on as an optimization problem. Hence, questions such as the existence of global and local minima, convergence and numerical conditioning of the optimization algorithm arise naturally. It can be shown that the answers to these questions are closely related to the geometry of the model set from which the model is selected.

Although real life processes are very complex, experience shows that they can very often be described accurately by means of FDLTI systems of a sufficiently high order. Thus, one of the most commonly used model sets in system identification is the set of all FDLTI systems of a fixed order. This model set is very often used when we do not have any *a priori* information about the process. However, in the case of the identification of industrial processes which are known to be stable, a better choice for the model set is the set of all stable FDLTI systems. A few other classes of linear systems are also commonly used in system identification — the class of positive real systems is used in the identification of stochastic processes, and the class of stable and minimum-phase systems is used for time-series analysis. The geometric structures of many of these classes of linear system are well studied. A quick examination of the geometric structures of these classes of systems shows that they share a few identical features. Hence, a natural question to ask at this point is whether these classes of systems are geometrically equivalent to each other.

We shall be investigating a particular type of geometric equivalence in this part of the thesis, namely diffeomorphism. If two classes of linear systems are diffeomorphic to each other, then they possess identical topological as well as differential geometric properties. The first result on diffeomorphism between classes of linear systems was given in [16], where Delchamps proved that the set of all stable FDLTI systems of fixed dimension and order is diffeomorphic to the set of all bounded real systems of the same dimension and order. In [26], Hanzon proved a similar result between systems of a fixed order and stable systems of the same order. In this part of the dissertation, we shall prove that a number of other classes of continuous time and discrete time FDLTI systems are also diffeomorphic to each other. The results reported here were at the same time obtained by Ober [50], but his derivations made use of results from robust control theory and are different from ours.

2.2 Outline

An outline of this part of the thesis is as follows.

Chapter 3: Preliminaries

In this chapter, we shall first define the various classes of continuous time and discrete time linear systems which are going to appear in this part of the thesis. We shall also give a summary of the notation which will be used here. In order to study the geometry of these classes of linear systems, we first need to endow them with the appropriate topological and differential geometric structures. We shall discuss how this can be done and also state a few geometric properties of a number of classes of systems. We shall finish this chapter by stating the already proven results on diffeomorphisms between classes of linear systems.

Chapter 4: Summary of the results

In this chapter, we shall summarise the results which we have obtained on diffeomorphisms between classes of linear systems. The main contribution of this part of thesis is that we prove that many classes of linear systems are diffeomorphic to each other. The results are summarised in five main theorems. We shall outline the proof of each of the theorems. The detailed proofs of some selected diffeomorphisms will be given in chapter 5.

Chapter 5: Proofs of various diffeomorphisms

This chapter contains detailed proofs of some selected diffeomorphisms between classes of linear systems. The basic philosophy behind all these proofs is the same. In the first step of the proof, we construct an invertible map between the two classes of linear systems which we would like to show to be diffeomorphic to each other. We then complete the proof by showing that both this map and its inverse are real analytic.

Chapter 6: Bundle diffeomorphisms

In this chapter, we shall look at two bundles associated with classes of linear systems, namely the state bundles and the principal fibre bundles. By using the results in chapters 4 and 5, we shall prove that the state bundles associated with these classes of systems are real analytic diffeomorphic to each other. Similar results on the diffeomorphisms between principal fibre bundles will also be proved.

Chapter 7: Applications

In this chapter, we shall discuss how our results on diffeomorphisms between classes of linear systems can be used to deduce geometric properties of other classes of systems, to construct overlapping parametrizations and to construct system identification algorithms based on Riemannian geometry.

Chapter 3

Preliminaries

The main aims of this chapter are to define the various classes of linear systems which are going to appear in this thesis, and to present the already known results on the diffeomorphisms between classes of linear systems. We begin in section 3.1, by giving the definitions of a number of classes of linear systems. The notation which we shall use to denote these classes of systems will also be given in this section. Section 3.2 contains a summary of selected geometric properties of a number of classes of linear systems. We shall finish this chapter by summarising the already known results on the diffeomorphisms between classes of linear systems.

3.1 Classes of linear systems

The objects of interest in this part of the thesis are the spaces of continuous time and discrete time FDLTI input-output dynamical systems of fixed order. The relations between the inputs and outputs of these dynamical systems can be represented in many different ways. Some examples of these representations are input-output maps, transfer functions and state space representations with states as auxiliary variables. In order to study the geometry of these spaces of dynamical systems, we need to endow the corresponding sets of input-output maps with the appropriate topological and differential structures. An alternative option is to study the geometry of the corresponding spaces of transfer functions or the corresponding spaces of input-output equivalence classes of minimal state space realizations. In this section, we shall define a number of classes of continuous time and discrete time systems. We shall also define the corresponding classes of minimal state space realizations.

3.1.1 Classes of continuous time systems

In this section, we shall define the classes of continuous time systems which are going to appear in this part of the thesis.

Let σ denote a m-input p-output continuous time FDLTI dynamical system of McMillan degree n, then there exists a corresponding minimal state space realization which is given by

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) + Du(t) \end{array} \right. \tag{3.1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ for all $t \in \mathbb{R}$, and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. The realization is minimal if (A, B) is controllable and (A, C) is observable. The transfer function representation of the input-output map is given by

$$G(s) \;\; = \;\; C(sI - A)^{-1}B + D \tag{3.2}$$

The classes of continuous time systems which are of interest to us are given in the following definition.

Definition 3.1 *In the following definitions, we shall use $G_\sigma(s)$ to denote the transfer function that describes the input-output behaviour of the linear dynamical system σ.*

1. *Fixed order*
   - $\Sigma M_n^{p,m}$= *the set of all m-input, p-output continuous time linear time-invariant (LTI) dynamical systems of McMillan degree n and whose state space representations are given by equation 3.1*
2. *Asymptotically stable*
   - $\Sigma S_n^{p,m}$= $\{\sigma \in \Sigma M_n^{p,m} : G_\sigma(s)$ *is continuous time asymptotically stable i.e. all the poles of $G_\sigma(s)$ are in the open left-half-plane*$\}$
3. *Strictly bounded real*
   - $\Sigma B_n^{p,m}$= $\{\sigma \in \Sigma S_n^{p,m} : G_\sigma(-\infty)^T G_\sigma(\infty) < I, I - G_\sigma(-j\omega)^T G_\sigma(j\omega) > 0 \; \forall\omega \in \mathbb{R}\}$
4. *Strictly positive real*
   - ΣP_n^{m}= $\{\sigma \in \Sigma S_n^{m,m} : G_\sigma(\infty) + G_\sigma(-\infty)^T > 0, G_\sigma(j\omega) + G_\sigma(-j\omega)^T > 0 \; \forall\omega \in \mathbb{R}\}$
5. *Invertible*
   - ΣMI_n^{m}= $\{\sigma \in \Sigma M_n^{m,m} : \sigma^{-1} \in \Sigma M_n^{m,m}\}$
6. *Stable and strictly minimum-phase*
   - ΣSM_n^{m}= $\{\sigma \in \Sigma MI_n^{m} \; : G_\sigma(s)^{-1}$ *is asymptotically stable*$\}$

Classes of continuous time state space realizations

For each continuous time FDLTI input-output system σ, its input-output behaviour can be represented by a state space quadruple (A, B, C, D) as given by equation 3.1. For the set $\Sigma M_n^{p,m}$, we shall collect all the minimal state space realizations which correspond to the systems in $\Sigma M_n^{p,m}$ in the set $SM_n^{p,m}$. The formal definition of $SM_n^{p,m}$ is

Definition 3.2 $SM_n^{p,m}$= $\{$ *(A,B,C,D)* $\in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n} \times \mathbb{R}^{p\times m}$*:* (A, B, C, D) *is a minimal state space realization of order* n $\}$

We shall define the sets of minimal state space realizations which correspond to the other five families of linear systems in a similar way. The sets of minimal state space realizations which correspond to the classes of systems $\Sigma S_n^{p,m}$, $\Sigma B_n^{p,m}$, ΣP_n^m, ΣMI_n^m and ΣSM_n^m will be denoted respectively by $SS_n^{p,m}$, $SB_n^{p,m}$, SP_n^m, SMI_n^m and SSM_n^m.

The main reason for defining these sets of minimal state space realizations is that we can study their quotient spaces with respect to the following input-output equivalence.

Definition 3.3 *Two state space realizations (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) in $SM_n^{p,m}$ are said to be input-output equivalent if there exists an invertible $n \times n$ matrix T such that $(A_1, B_1, C_1, D_1) = (TA_2T^{-1}, TB_2, C_2T^{-1}, D_2)$.*

We shall denote the quotient space of $SM_n^{p,m}$ which is formed by this equivalence relation by $SM_n^{p,m}/\sim$. Note that there is a one-to-one correspondence between the systems in $\Sigma M_n^{p,m}$ and the equivalence classes in $SM_n^{p,m}/\sim$. Later on, we shall define topological and differential structures on the set $SM_n^{p,m}/\sim$ in order to study the geometry of $\Sigma M_n^{p,m}$. The same principle will also be used to study the geometry of the other five classes of linear systems.

3.1.2 Classes of discrete time systems

In this section, we shall define the discrete time counterparts of the classes of systems given in definition 3.1

Definition 3.4 *In the following definitions, we shall use the notation $G_\sigma(z)$ (where z denotes the forward shift operator) to denote the discrete time transfer function corresponding to the linear dynamical system σ.*

1. *Fixed order*

   - $D\Sigma M_n^{p,m}$= *the set of all m-input, p-output discrete time LTI dynamical systems of McMillan degree n and whose state space representations are of the form*

$$\begin{cases} x[t+1] &= Ax[t] + Bu[t] \\ y[t] &= Cx[t] + Du[t] \end{cases}$$

2. *Asymptotically stable*

   - $D\Sigma S_n^{p,m}$= $\{\sigma \in D\Sigma M_n^{p,m} : G_\sigma(z)$ *is discrete time asymptotically stable i.e. all the poles of $G_\sigma(z)$ are within the open unit disk*$\}$

3. *Strictly bounded real*

   - $D\Sigma B_n^{p,m}$= $\{\sigma \in D\Sigma S_n^{p,m} : I - G_\sigma(e^{-j\theta})^T G_\sigma(e^{j\theta}) > 0 \ \forall\theta \in [0, 2\pi]\}$

4. *Strictly positive real*

   - $D\Sigma P_n^m$= $\{\sigma \in D\Sigma S_n^{m,m} : G_\sigma(e^{j\theta}) + G_\sigma(e^{-j\theta})^T > 0, \ \forall\theta \in [0, 2\pi]\}$

5. *Invertible*

   - $D\Sigma MI_n^m = \{\sigma \in D\Sigma M_n^{m,m} : \sigma^{-1} \in D\Sigma M_n^{m,m}\}$

6. *Stable and strictly minimum-phase*

   - $D\Sigma SM_n^m = \{\sigma \in D\Sigma MI_n^m : G_\sigma(z)^{-1} \textit{ is asymptotically stable}\}$

We shall also define the spaces of minimal state space realizations corresponding to the above six classes of discrete time systems. We shall use the notation $DSM_n^{p,m}$, $DSS_n^{p,m}$, $DSB_n^{p,m}$, DSP_n^m, $DSMI_n^m$ and $DSSM_n^m$ to denote the spaces of minimal state space realizations corresponding to the spaces of discrete time systems $D\Sigma M_n^{p,m}$, $D\Sigma S_n^{p,m}$, $D\Sigma B_n^{p,m}$, $D\Sigma P_n^m$, $D\Sigma MI_n^m$ and $D\Sigma SM_n^m$ respectively.

Scalar discrete time systems with fixed number of finite zeros

We are also interested in the following 3 subsets of $D\Sigma M_n^{1,1}$, the set of all discrete time single-input-single-output (SISO) LTI systems of McMillan degree n. In the following definitions, $g_\sigma(z)$ is used to denote the discrete time transfer function which corresponds to the system σ.

Definition 3.5 *For $0 \leq r \leq n$, define*

- $DRat(n; r) = \{\sigma \in D\Sigma M_n^{1,1} : \textit{relative degree of } g_\sigma(z) = r\}$
- $DRat_S(n; r) = \{\sigma \in DRat(n; r) : g_\sigma(z) \textit{ is discrete time asymptotically stable}\}$
- $DRat_{SM}(n; r) = \{\sigma \in DRat_S(n; r)$: *all the finite zeros of* $g_\sigma(z)$ *are within the open unit disk* $\}$

3.1.3 Summary of notation

For ease of reference, we have summarised in table 3.1 the notation defined in sections 3.1.1 and 3.1.2.

3.2 Geometric properties of spaces of linear systems

In this section, we shall state some geometric properties of the families of linear systems defined in the last section. These geometric results came out through research on problems related to system identification, on, for example: parametrizing the set of fixed order systems, numerical conditioning of the estimator and many others. There is a lot of literature in this area, the reader may refer to the following references: [8, 9, 10, 12, 14, 15, 19, 25, 26, 30, 32, 33, 51, 52]. However, these geometric properties are also interesting results in their own right. This section is further divided into 3 sub-sections. We shall discuss topological and differential geometric properties in the first two of them. In the last sub-section, we shall summarise the already known results on diffeomorphisms between classes of linear systems.

Classes of systems	Continuous time		Discrete time	
	Systems	State space	Systems	State space
Fixed order	$\Sigma M_n^{p,m}$	$SM_n^{p,m}$	$D\Sigma M_n^{p,m}$	$DSM_n^{p,m}$
Asymptotically stable	$\Sigma S_n^{p,m}$	$SS_n^{p,m}$	$D\Sigma S_n^{p,m}$	$DSS_n^{p,m}$
Strictly bounded real	$\Sigma B_n^{p,m}$	$SB_n^{p,m}$	$D\Sigma B_n^{p,m}$	$DSB_n^{p,m}$
Strictly positive real	ΣP_n^{m}	SP_n^{m}	$D\Sigma P_n^{m}$	DSP_n^{m}
Invertible	ΣMI_n^{m}	SMI_n^{m}	$D\Sigma MI_n^{m}$	$DSMI_n^{m}$
Stable and minimum-phase	ΣSM_n^{m}	SSM_n^{m}	$D\Sigma SM_n^{m}$	$DSSM_n^{m}$

Table 3.1: Summary of notation

3.2.1 Topological properties

In order to study the topological properties of various classes of linear systems, we first need to endow them with the appropriate topological structure. The set $SM_n^{p,m}/\sim$ will be topologized as follows: embed $SM_n^{p,m}$ in $\mathbb{R}^{n^2+np+nm+pm}$ by

$$(A, B, C, D) \rightarrow (a^1, a^2, ..., a^n, b^1, ..., b^n, c^1, ..., c^p, d^1, ..., d^p)$$

where a^i, b^i, c^i, d^i are the i^{th} row of A, B, C, D respectively. We then endow $SM_n^{p,m}$, which is a subset of $\mathbb{R}^{n^2+np+nm+pm}$, with the subspace topology [5] and $SM_n^{p,m}/\sim$ with the quotient topology (also known as the identification topology, see [5, 61]). The other classes of minimal state space realizations will also be topologized in the same way.

A topological property which is of importance to system identification is the number of pathwise connected components. (See section 7.1.2 for the implications of these results on system identification.)

Lemma 3.1 *(Brockett [8])* $\Sigma M_n^{1,1}$ *has (n+1) pathwise connected components.*

Lemma 3.2 *(Glover [19]) For* $max(p, m) > 1$, $\Sigma M_n^{p,m}$ *is pathwise connected.*

Lemma 3.3 *(Hanzon [25], Ober [51]) The set* $\Sigma S_n^{p,m}$ *has (n+1) pathwise connected components when* $p = m = 1$, *otherwise it is pathwise connected.*

Note that the sets $\Sigma M_n^{p,m}$ and $\Sigma S_n^{p,m}$ have identical number of pathwise connected components. We shall now define the concept of homeomorphism or topological equivalence [5].

Definition 3.6 *Let* X *and* Y *be two topological spaces, then* X *and* Y *are homeomorphic to each other if there exists a bijective map* $f : X \rightarrow Y$ *such that both* f *and* f^{-1} *are continuous.*

The topological equivalence of $\Sigma M_n^{p,m}$ and $\Sigma S_n^{p,m}$ are stated in the lemma below.

Lemma 3.4 *(Helmke [33]) The sets* $\Sigma M_n^{p,m}$ *and* $\Sigma S_n^{p,m}$ *are homeomorphic to each other.*

If two topological spaces are homeomorphic to each other, then they possess identical topological properties such as compactness, number of pathwise connected components and many others. Hence, the sets $\Sigma M_n^{p,m}$ and $\Sigma S_n^{p,m}$ have exactly the same topological properties besides having the same number of pathwise connected components. In simple terms, topological properties can be looked at as the global geometric properties of a set. We shall now turn our attention to the local geometric properties of families of linear systems.

3.2.2 Differential geometric properties

The most basic result on the local geometric properties of $\Sigma M_n^{p,m}$ is that it has the structure of a smooth manifold.

Lemma 3.5 *(Clark [12], Hazewinkel [30]) The set $\Sigma M_n^{p,m}$ is a real analytic manifold of dimension $n(m+p)+mp$.*

There are a few different ways to prove this result. A particular method, which is given in [30], makes use of the idea of nice selections. These are ways of selecting basis row vectors of a Hankel matrix. An atlas of charts on $SM_n^{p,m}/\sim$ is first constructed by using nice selections with each chart being associated with a particular nice selection. The set $SM_n^{p,m}/\sim$ is then shown to be a real analytic manifold by proving that the coordinate transformation between any two charts is a real analytic map. These charts also provide the necessary differential structures needed to investigate the differential geometric properties of $\Sigma M_n^{p,m}$. For alternative proofs of this lemma, the reader can refer to [12] or [9].

Since $\Sigma S_n^{p,m}$, $\Sigma B_n^{p,m}$, ΣP_n^m, ΣMI_n^m and ΣSM_n^m are all open subsets of $\Sigma M_n^{p,m}$, so they are real analytic sub-manifolds of $\Sigma M_n^{p,m}$.

Lemma 3.6 *The sets $\Sigma S_n^{p,m}$ and $\Sigma B_n^{p,m}$ are real analytic manifolds of dimension $n(m+p)+mp$.*

Lemma 3.7 *The sets ΣP_n^m, ΣMI_n^m and ΣSM_n^m are real analytic manifolds of dimension $2nm+m^2$.*

Since we can identify the set $SM_n^{p,m}/\sim$ with $DSM_n^{p,m}/\sim$, we have

Lemma 3.8 *The sets $D\Sigma M_n^{p,m}$, $D\Sigma S_n^{p,m}$ and $D\Sigma B_n^{p,m}$ are real analytic manifolds of dimension $n(m+p)+mp$.*

Lemma 3.9 *The sets $D\Sigma P_n^m$, $D\Sigma MI_n^m$ and $D\Sigma SM_n^m$ are real analytic manifolds of dimension $2nm+m^2$.*

Since $DRat(n;r)$, $DRat_S(n;r)$ and $DRat_{SM}(n;r)$ are open subsets of $D\Sigma M_n^{1,1}$, we have the following lemma.

Lemma 3.10 *The sets $DRat(n;r)$, $DRat_S(n;r)$ and $DRat_{SM}(n;r)$ where $n \geq r \geq 0$ are real analytic manifolds of dimension $2n-r+1$.*

3.2.3 Diffeomorphisms

This part of the thesis is concerned with diffeomorphisms between manifolds of linear systems. We shall first give the definition of diffeomorphism [7].

Definition 3.7 *Let X and Y be two real analytic manifolds, then X and Y are real analytic diffeomorphic to each other if there exists a bijective map $f : X \to Y$ such that both f and f^{-1} are real analytic.*

Since a real analytic map is also a continuous map, it follows that if two manifolds are diffeomorphic to each other, then they are also homeomorphic to each other. Also, if two manifolds are diffeomorphic to each other, they possess identical local geometric properties on top of having identical topological properties. A few diffeomorphisms between classes of linear systems were recorded in the earlier literature and they are stated in the following three theorems. Note that the notation $SP(X)$ is used to denote the set of all linear systems that belong to X and whose associated transfer functions are strictly proper.

Theorem 3.1 *(Delchamps [16]) The sets $SP(\Sigma S_n^{p,m})$ and $SP(\Sigma B_n^{p,m})$ are diffeomorphic to each other.*

Theorem 3.2 *(Hanzon [24]) The sets $SP(\Sigma S_n^{p,m})$ and $SP(D\Sigma S_n^{p,m})$ are diffeomorphic to each other.*

Theorem 3.3 *(Hanzon [26]) The sets $SP(D\Sigma M_n^{p,m})$ and $SP(D\Sigma S_n^{p,m})$ are real analytic diffeomorphic to each other.*

Since diffeomorphism is an equivalence relation, we can immediately conclude from the above three theorems that the sets $SP(\Sigma S_n^{p,m})$, $SP(\Sigma B_n^{p,m})$, $SP(D\Sigma S_n^{p,m})$ and $SP(D\Sigma M_n^{p,m})$ are all diffeomorphic to each other. The above three theorems are the starting point of our investigation into the diffeomorphisms between classes of linear systems. In the later chapters of this thesis, we shall extend the above results to include systems with proper transfer functions. We shall also prove a number of other diffeomorphisms between classes of linear systems.

Chapter 4

Summary of the results

In this chapter, we shall state the main theorems of this part of the thesis on the diffeomorphisms between classes of linear systems. These results will be summarised in the five theorems below and we shall also give an outline of the proof of each theorem. The detailed proofs of some selected diffeomorphisms will be given in chapter 5.

Note that the results stated in theorems 3.1, 3.2 and 3.3 were proved for sets of linear systems with strictly proper transfer functions only. Most of these results can easily be generalised to include systems with proper transfer functions as well, except in the case of theorem 3.1 which requires a bit of lengthy algebra; the extension of this theorem to include systems with proper transfer functions is given in section 5.1. The main contribution here is that a number of other classes of linear systems are shown to be diffeomorphic to each other. We shall now state the main theorems of this part of the thesis.

Diffeomorphisms between rectangular systems

Theorem 4.1 *The following six classes of linear systems are real analytic diffeomorphic to each other:* $\Sigma M_n^{p,m}$, $\Sigma S_n^{p,m}$, $\Sigma B_n^{p,m}$, $D\Sigma M_n^{p,m}$, $D\Sigma S_n^{p,m}$ *and* $D\Sigma B_n^{p,m}$.

To prove the above theorem, we proceed according to the following diagram.

$$\begin{tikzcd}
\Sigma M_n^{p,m} \arrow[d, leftrightarrow, "(5.4)"] & \Sigma S_n^{p,m} \arrow[r, leftrightarrow, "(5.1)"] \arrow[d, leftrightarrow, "(5.4)"] & \Sigma B_n^{p,m} \arrow[d, leftrightarrow, "(5.4)"] \\
D\Sigma M_n^{p,m} \arrow[r, leftrightarrow] & D\Sigma S_n^{p,m} & D\Sigma B_n^{p,m}
\end{tikzcd}$$

We are able to prove that the manifolds which are connected by a double-headed arrow in the above diagram are diffeomorphic to each other. Since diffeomorphism is an equivalence relation, it is clear that the 6 classes of linear systems are all diffeomorphic to each other. The fact that $D\Sigma M_n^{p,m}$ and $D\Sigma S_n^{p,m}$ are diffeomorphic to each other is a simple extension to theorem 3.3. The proofs of the other diffeomorphisms can be found in chapter 5, the section in which the proof can be found is given by the number next to the arrow.

Diffeomorphisms between square systems (I)

Theorem 4.2 *The following 8 classes of linear systems are real analytic diffeomorphic to each other:* $\Sigma M_n^{m,m}$, $\Sigma S_n^{m,m}$, $\Sigma B_n^{m,m}$, ΣP_n^m, $D\Sigma M_n^{m,m}$, $D\Sigma S_n^{m,m}$, $D\Sigma B_n^{m,m}$ *and* $D\Sigma P_n^m$.

The proof shall proceed according to the following diagram.

$$\begin{tikzcd}
\Sigma M_n^{m,m} \arrow[d, leftrightarrow] & \Sigma S_n^{m,m} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \Sigma B_n^{m,m} \arrow[r, leftrightarrow, "(5.2)"] \arrow[d, leftrightarrow] & \Sigma P_n^m \arrow[d, leftrightarrow, "(5.4)"] \\
D\Sigma M_n^{m,m} \arrow[r, leftrightarrow] & D\Sigma S_n^{m,m} & D\Sigma B_n^{m,m} & D\Sigma P_n^m
\end{tikzcd}$$

First note that all the six classes of linear systems which are present in the previous theorem also appear here, except that we have restricted them to square systems only. Since the diffeomorphisms mentioned in the previous theorem hold for any value of p and m, they are also valid when $p = m$. The positive real system is the only new class of systems which appears here. The proofs of two of these diffeomorphisms can be found in chapter 5 and the number adjacent to the arrow indicates the section in which the proof can be found.

Diffeomorphisms between square systems (II)

Theorem 4.3 *The following 4 classes of linear systems are real analytic diffeomorphic to each other:* ΣMI_n^m, ΣSM_n^m, $D\Sigma MI_n^m$ *and* $D\Sigma SM_n^m$.

$$\begin{tikzcd}
\Sigma MI_n^m \arrow[d, leftrightarrow, "(5.4)"] & \Sigma SM_n^m \arrow[d, leftrightarrow, "(5.4)"] \\
D\Sigma MI_n^m \arrow[r, leftrightarrow, "(5.3)"] & D\Sigma SM_n^m
\end{tikzcd}$$

The proof proceeds according to the above diagram and again the number next to the arrow indicates the section in which the proof can be found.

Diffeomorphisms between scalar systems

Note that both theorems 4.2 and 4.3 involve diffeomorphisms between square systems. In the case of SISO systems, we can actually 'link' these two groups of diffeomorphisms together. However, we have to take 'half' of ΣMI_n^1, ΣSM_n^1, $D\Sigma MI_n^1$ and $D\Sigma SM_n^1$. A straightforward reason for why we have to take 'half' of these sets is that they each have $2(n+1)$ pathwise connected components while those manifolds in theorem 4.2 each have only $(n+1)$

pathwise connected components. (See section 7.1.2 for results on the number of pathwise connected components on these classes of linear systems.) We introduce the following classes of continuous time and discrete time SISO systems:

Definition 4.1 *In the following definitions, we shall use $g_\sigma(s)$ to denote the transfer function associated with the continuous time system σ.*

- $\Sigma MI^1_{n+} = \{\sigma \in \Sigma MI^1_n : \; g_\sigma(\infty) > 0\}$
- $\Sigma MI^1_{n-} = \{\sigma \in \Sigma MI^1_n : \; g_\sigma(\infty) < 0\}$
- $\Sigma SM^1_{n+} = \{\sigma \in \Sigma SM^1_n : \; g_\sigma(\infty) > 0\}$
- $\Sigma SM^1_{n-} = \{\sigma \in \Sigma SM^1_n : \; g_\sigma(\infty) < 0\}$

Definition 4.2 *In the following definitions, $f_\sigma(z)$ is used to denote the transfer function associated with the discrete time system σ.*

- $D\Sigma MI^1_{n+} = \{\sigma \in D\Sigma MI^1_n : \; f_\sigma(-1) > 0\}$
- $D\Sigma MI^1_{n-} = \{\sigma \in D\Sigma MI^1_n : \; f_\sigma(-1) < 0\}$
- $D\Sigma SM^1_{n+} = \{\sigma \in D\Sigma SM^1_n : \; f_\sigma(-1) > 0\}$
- $D\Sigma SM^1_{n-} = \{\sigma \in D\Sigma SM^1_n : \; f_\sigma(-1) < 0\}$

We shall now state the theorem on the diffeomorphisms between classes of scalar systems.

Theorem 4.4 *The following sets of SISO systems are real analytic diffeomorphic to each other: $\Sigma M^{1,1}_n$, $\Sigma S^{1,1}_n$, $\Sigma B^{1,1}_n$, ΣP^1_n, ΣMI^1_{n+}, ΣMI^1_{n-}, ΣSM^1_{n+}, ΣSM^1_{n-}, $D\Sigma M^{1,1}_n$, $D\Sigma S^{1,1}_n$, $D\Sigma B^{1,1}_n$, $D\Sigma P^1_n$, $D\Sigma MI^1_{n+}$, $D\Sigma MI^1_{n-}$, $D\Sigma SM^1_{n+}$ and $D\Sigma SM^1_{n-}$.*

The proof of this theorem will be given in section 5.5.

Diffeomorphisms between classes of discrete time scalar systems

Finally, we have the following theorem on the diffeomorphisms between classes of discrete time SISO systems.

Theorem 4.5 *The following 3 manifolds of discrete time SISO systems are real analytic diffeomorphic to each other: $DRat(n;r)$, $DRat_S(n;r)$ and $DRat_{SM}(n;r)$.*

The proof of this theorem will be given in section 5.6.

Chapter 5

Proofs of various diffeomorphisms

This chapter contains the proofs of some selected diffeomorphisms between classes of linear systems. The basic idea behind all these proofs is the same. In the first step of the proof, we construct a bijective map between the two classes of linear systems which we wish to show to be real analytic diffeomorphic to each other. In order to complete the proof, we then have to show that both this map and its inverse are real analytic mappings.

We shall be constructing mappings between classes of minimal state space realizations in most of the proofs in this chapter. We shall require the following definition.

Definition 5.1 *A map $\psi : \mathcal{S} \to \tilde{\mathcal{S}}$, where $\mathcal{S}$ and $\tilde{\mathcal{S}}$ are two classes of minimal state space realizations, is said to preserve input-output equivalence if it satisfies the following property: Let s_1 and s_2 be two state space realizations in $\mathcal{S}$, and $\sim$ be the input-output equivalence given in definition 3.3, we have $s_1 \sim s_2$ if and only if $\psi(s_1) \sim \psi(s_2)$.*

The following is a list of proofs which appear in this chapter.

Section 5.1:	Proof of diffeomorphism between $\Sigma S_n^{p,m}$ and $\Sigma B_n^{p,m}$
Section 5.2:	Proof of diffeomorphism between $\Sigma B_n^{m,m}$ and ΣP_n^m
Section 5.3:	Proof of diffeomorphism between $D\Sigma MI_n^m$ and $D\Sigma SM_n^m$
Section 5.4:	Proof of diffeomorphisms between classes of continuous time and discrete time systems
Section 5.5:	Proof of theorem 4.4
Section 5.6:	Proof of theorem 4.5

5.1 Diffeomorphism between $\Sigma S_n^{p,m}$ and $\Sigma B_n^{p,m}$

In this section, we shall prove that the two classes of linear systems, asymptotically stable and bounded real, are real analytic diffeomorphic to each other. A restricted version of this result (see theorem 3.1) was proved by Delchamps for systems with strictly proper transfer functions only. The result will be extended to include systems with proper transfer functions in this section. We shall first state the main theorem of this section.

Theorem 5.1 *The two classes of linear systems, $\Sigma S_n^{p,m}$ and $\Sigma B_n^{p,m}$, are real analytic diffeomorphic.*

The way in which this new result is proved is similar to that of the restricted version except that more lengthy algebra is needed. Basically, we need to prove that there exists a bijective map between these two classes of systems, and both this map and its inverse are real analytic. A number of auxiliary results are needed in order to prove the main theorem. The proof can be divided into three sections, as follows:

1. We shall prove in lemmas 5.2 to 5.7 that a number of mappings are real analytic. The main results are given in lemmas 5.6 and 5.7 which state that the stabilizing solution to the generalized control algebraic Riccati equation (GCARE) and the stabilizing solution to the generalized bounded real Riccati equation (GBRRE) are real analytic.

2. In lemma 5.8, we shall prove that the closed-loop system obtained from applying a certain optimal state feedback to an asymptotically stable system is bounded real. We shall then use this feedback law to define a map from the stable state space systems to the bounded real state space systems (Definition 5.2).

3. This map is then proved to be invertible (Lemma 5.9) and both the map and its inverse are proved to be real analytic (Lemma 5.10). Hence the theorem.

Before we present the auxiliary lemmas, we shall first state a standard result which relates the state space description of a bounded real system to the GBRRE.

Lemma 5.1 *([4, 80]) $(A, B, C, D) \in SS_n^{p,m}$ with $I - D^T D > 0$ is said to be strictly bounded real if and only if there exists a symmetric positive definite solution to the GBRRE*

$$A^T P + PA + (PB + C^T D)(I - D^T D)^{-1}(PB + C^T D)^T + C^T C = 0 \tag{5.1}$$

such that $A + B(I - D^T D)^{-1}(B^T P + D^T C)$ is asymptotically stable i.e. all the eigenvalues of the matrix are in the open left-half-plane.

The following two lemmas state that the stabilizing solution to the algebraic Riccati equation (ARE) and the stabilizing solution to the bounded real Riccati equation (BRRE) are real analytic.

Lemma 5.2 *(Delchamps [16]) Let $SS_{n,D=0}^{p,m}$ be the set $\{(A, B, C) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n}$: (A, B, C) is a minimal state space realization of a continuous time asymptotically stable system$\}$. Define the map*

$$\begin{array}{rccc} P_* : & SS_{n,D=0}^{p,m} & \to & \mathbb{R}^{n\times n} \\ & (A, B, C) & \mapsto & P_*(A, B, C) \end{array}$$

where $P_*(A, B, C)$ is the symmetric positive definite solution to the ARE

$$A^T P + PA - PBB^T P + C^T C \;=\; 0 \tag{5.2}$$

such that $A - BB^T P$ is asymptotically stable.

The map P_* is real analytic on the open set $SS_{n,D=0}^{p,m}$.

Proof: This theorem can be proved by using the implicit function theorem [65]. See [16] for a proof. □

Lemma 5.3 *Let $SB^{p,m}_{n,D=0}$ be the set $\{(A,B,C) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n} : (A,B,C)$ is a minimal state space realization of a continuous time strictly bounded real system$\}$. Define the map*

$$\begin{array}{rccc} P_* : & SB^{p,m}_{n,D=0} & \to & \mathbb{R}^{n\times n} \\ & (A,B,C) & \mapsto & P_*(A,B,C) \end{array}$$

where $P_*(A,B,C)$ is the symmetric positive definite solution to the BRRE

$$A^T P + PA + PBB^T P + C^T C \;=\; 0 \tag{5.3}$$

such that $A + BB^T P$ is asymptotically stable.

The map P_* is real analytic on the open set $SB^{p,m}_{n,D=0}$

Proof: This lemma can be proved in a way similar to proving lemma 5.2. The proof is omitted. □

Lemma 5.4 *Define the maps*

$$\begin{array}{rccc} g_1 : & \mathbb{R}^{p\times m} & \to & \mathbb{R}^{m\times m} \\ & D & \mapsto & (I + D^T D)^{1/2} \\ g_2 : & \mathbb{R}^{p\times m} & \to & \mathbb{R}^{p\times p} \\ & D & \mapsto & (I + DD^T)^{1/2} \end{array}$$

where $(M)^{1/2}$ denotes the positive definite square root of the symmetric matrix M.

The maps g_1 and g_2 are real analytic on the set $\mathbb{R}^{p\times m}$.

Proof: Consider the function

$$f_1(P,D) \;=\; P^2 - I - D^T D$$

which is an analytic map in P and D. The map g_1 satisfies the equation $f_1(g_1(D), D) = 0$ and since we have restricted $g_1(D)$ to be positive definite, $g_1(D)$ is the unique solution to the equation.

The differential of f_1 is given by

$$df_1 \;=\; dP.P + P.dP + (dD)^T.D + D^T.dD$$

At any point $(P,D) \in \mathbb{R}^{m\times m} \times \mathbb{R}^{p\times m}$, df_1 maps the tangent vector $(dP, dD) = (X, 0)$ to $XP + PX$. Since P is positive definite, this map is bijective. Hence by the implicit function theorem, the map g_1 is real analytic. By using similar arguments, we can show that the map g_2 is real analytic. □

Lemma 5.5 *Let U be the set $\{D \in \mathbb{R}^{p\times m} : I - D^TD > 0\}$, which is equal to $\{D \in \mathbb{R}^{p\times m} : I - DD^T > 0\}$. Define the maps*

$$\begin{array}{rccc} h_1: & U & \to & \mathbb{R}^{m\times m} \\ & D & \mapsto & (I - D^TD)^{1/2} \\ h_2: & U & \to & \mathbb{R}^{p\times p} \\ & D & \mapsto & (I - DD^T)^{1/2} \end{array}$$

The maps h_1 and h_2 are real analytic on the open set U.

Proof: The lemma can be proved by using the implicit function theorem in a way similar to proving lemma 5.4. □

We shall now generalize lemmas 5.2 and 5.3 to prove that the stabilizing solution to the GCARE and the stabilizing solution to the GBRRE are both real analytic.

Lemma 5.6 *Define the map*

$$\begin{array}{rccc} P_*: & SS_n^{p,m} & \to & \mathbb{R}^{n\times n} \\ & (A,B,C,D) & \mapsto & P_*(A,B,C,D) \end{array}$$

where P_* is the symmetric positive definite solution to the GCARE

$$A^TP + PA - (PB + C^TD)(I + D^TD)^{-1}(PB + C^TD)^T + C^TC = 0 \tag{5.4}$$

such that $A - B(I + D^TD)^{-1}(D^TC + B^TP)$ is asymptotically stable.

P_* is an analytic function of (A,B,C,D) on the open set $SS_n^{p,m}$.

Proof: Let $S = I + D^TD$ and $R = I + DD^T$.

Define

$$\begin{aligned} \bar{A} &= A - BS^{-1}D^TC \\ \bar{B} &= BS^{-1/2} \\ \bar{C} &= R^{-1/2}C \end{aligned}$$

In terms of $\bar{A}, \bar{B}, \bar{C}$, the GCARE can be written as

$$\bar{A}^TP + P\bar{A} - P\bar{B}\bar{B}^TP + \bar{C}^T\bar{C} = 0 \tag{5.5}$$

It can be shown that since (A,B,C,D) is a minimal state space realization, $(\bar{A}, \bar{B}, \bar{C})$ is also a minimal state space realization. By lemma 5.2, the stabilizing solution P_* to equation 5.5 is a real analytic function of $(\bar{A}, \bar{B}, \bar{C})$. The matrix P_* is in fact also the stabilizing solution to the GCARE as

$$\bar{A} - \bar{B}\bar{B}^TP_* = A - B(I + D^TD)^{-1}(D^TC + B^TP_*). \tag{5.6}$$

By lemma 5.4, the matrices S and R are both real analytic functions of D, hence $(\bar{A}, \bar{B}, \bar{C})$ is a real analytic function of (A,B,C,D). Hence the lemma. □

Lemma 5.7 *Define the map*

$$\begin{array}{rccc} P_*: & SB_n^{p,m} & \to & \mathbb{R}^{n\times n} \\ & (A,B,C,D) & \mapsto & P_*(A,B,C,D) \end{array}$$

where P_* is the symmetric positive definite solution to the GBRRE

$$A^TP + PA + (PB + C^TD)(I - D^TD)^{-1}(PB + C^TD)^T + C^TC = 0 \tag{5.7}$$

such that $A + B(I - D^TD)^{-1}(D^TC + B^TP)$ is asymptotically stable.
P_* is an analytic function of (A,B,C,D) on the open set $SB_n^{p,m}$.

Proof: This lemma can be proved in a way similar to proving lemma 5.6 and the proof is omitted. □

Now we shall show how, by applying an optimal state feedback [80], we can obtain a bounded real system from a stable one. Let $(A,B,C,D) \in SS_n^{p,m}$, there exists a symmetric positive definite solution P to the GCARE

$$A^TP + PA - (PB + C^TD)(I + D^TD)^{-1}(PB + C^TD)^T + C^TC = 0$$

such that $A - B(I + D^TD)^{-1}(B^TP + D^TC)$ is asymptotically stable. For the rest of this section, we shall use S and R to denote the matrices $I + D^TD$ and $I + DD^T$ respectively. We apply the state feedback law

$$u(t) = -S^{-1}(B^TP + D^TC)x(t) + S^{-1/2}v(t) \tag{5.8}$$

to the stable state space system

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) + Du(t) \end{array} \right. \tag{5.9}$$

The state feedback law $u(t) = -S^{-1}(B^TP + D^TC)x(t)$ corresponds to the solution to the steady state Linear Quadratic regulator problem [80] and the following cost function:

$$J = \int_0^\infty (u(t)^Tu(t) + y(t)^Ty(t))dt \tag{5.10}$$

The resulting closed-loop system is given by

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & A_*x(t) + B_*v(t) \\ y(t) & = & C_*x(t) + D_*v(t) \end{array} \right. \tag{5.11}$$

where

$$\left\{ \begin{array}{rclcl} B_* & = & BS^{-1/2} & & \\ D_* & = & DS^{-1/2} & & \\ A_* & = & A - BS^{-1}(B^TP + D^TC) & = & A - B_*(B_*^TP + D_*^TC) \\ C_* & = & C - DS^{-1}(B^TP + D^TC) & = & C - D_*(B_*^TP + D_*^TC) \end{array} \right. \tag{5.12}$$

We shall show in the following lemma that the quadruple (A_*, B_*, C_*, D_*) is a strictly bounded real state space system.

Lemma 5.8 *1. $I - D_*^T D_* = (I + D^T D)^{-1} > 0$*

2. The stabilizing solution P_* to the GCARE also satisfies the equation

$$A_*^T P + PA_* + C_*^T C_* + (PB_* + C_*^T D_*)(I - D_*^T D_*)^{-1}(PB_* + C_*^T D_*)^T \;=\; 0$$

which is the GBRRE associated with (A_*, B_*, C_*, D_*) and the matrix $A_* + B_*(I - D_*^T D_*)^{-1}(P_* B_* + C_*^T D_*)^T$ is asymptotically stable.

3. The state space system (A_*, B_*, C_*, D_*) is a minimal realization.

4. The quadruple (A_*, B_*, C_*, D_*) belongs to $SB_n^{p,m}$.

Proof:

1. By direct substitution.

2. Let

$$F_* = B_*^T P + D_*^T C, \tag{5.13}$$

then the last two equations in (5.12) can be written as

$$\begin{aligned} A_* &= A - B_* F_* & (5.14) \\ C_* &= C - D_* F_* & (5.15) \end{aligned}$$

By eliminating C from equations (5.13) and (5.15), we have

$$\begin{aligned} F_* &= B_*^T P + D_*^T (C_* + D_* F_*) \\ \Rightarrow \quad B_*^T P + D_*^T C_* &= (I - D_*^T D_*) F_* & (5.16) \end{aligned}$$

Further note that the GCARE can be written as

$$A^T P + PA + C^T C - F_*^T F_* \;=\; 0 \tag{5.17}$$

We shall first show that the solutions to the GCARE in (A, B, C, D) are also the solutions to the GBRRE in (A_*, B_*, C_*, D_*). Let P be any solution to the GBRRE, then by substituting (5.14),(5.15) and (5.16) into (5.13), we have

$$\begin{aligned} & A_*^T P + PA_* + C_*^T C_* + (PB_* + C_*^T D_*)(I - D_*^T D_*)^{-1}(PB_* + C_*^T D_*)^T \\ = \; & (A - B_* F_*)^T P + P(A - B_* F_*) + (C - D_* F_*)^T (C - D_* F_*) + \\ & F_*^T (I - D_*^T D_*)(I - D_*^T D_*)^{-1}(I - D_*^T D_*) F_* \\ = \; & A^T P + PA + C^T C - F_*^T (B_*^T P + D_*^T C) - (PB_* + C^T D_*) F_* + F_*^T F_* \\ = \; & A^T P + PA + C^T C - F_*^T F_* \end{aligned}$$

The GCARE in (A, B, C, D) and GBRRE in (A_*, B_*, C_*, D_*) in fact share the same stabilizing solution as

$$\begin{aligned} & A_* + B_*(I - D_*^T D_*)^{-1}(P_* B_* + C_*^T D_*)^T \\ = \; & A_* + B_* F_* \\ = \; & A \end{aligned}$$

which is asymptotically stable.

3. The controllability of (A_*, B_*) can be shown easily by the PBH test [36].

   Assume (A_*, C_*) is not observable, then there exists an eigenvalue λ (with $\mathrm{Re}(\lambda) < 0$) and the corresponding eigenvector $v(\neq 0)$ of A_* such that $A_*v = \lambda v$ and $C_*v = 0$.

   It was shown in the second section of this proof that the GCARE associated with the system (A, B, C, D) can be written as

$$A_*^T P + PA_* + C_*^T C_* + F_*^T (B_*^T P + D_*^T C_*) \;=\; 0 \tag{5.18}$$

   where F_* was defined in equation (5.13).

   Post-multiplying equation (5.18) by v and put $C_*v = 0$, we have

$$\begin{aligned} A_*^T Pv + PA_*v + F_*^T B_*^T Pv &= 0 \\ \Rightarrow \qquad (A_*^T + F_*^T B_*^T + \lambda I)Pv &= 0 \\ \Rightarrow \qquad (A^T + \lambda I)Pv &= 0 \\ \Rightarrow \qquad Pv &= 0 \quad \text{as A is stable} \end{aligned}$$

   This contradicts the fact that P is positive definite, hence (A_*, C_*) is observable.

4. Since $I - D_*^T D_* > 0$ and there exists a stabilizing solution to the GBRRE associated with (A_*, B_*, C_*, D_*), the quadruple (A_*, B_*, C_*, D_*) belongs to $SB_n^{p,m}$. □

Now we shall make use of equation (5.12) to define a map between $SS_n^{p,m}$ and $SB_n^{p,m}$.

Definition 5.2 *Define the map*

$$\begin{aligned} \psi : \quad SS_n^{p,m} \quad &\to \quad SB_n^{p,m} \\ (A, B, C, D) \quad &\mapsto \quad (A_*, B_*, C_*, D_*) \text{ as defined by equation (5.12)} \end{aligned}$$

Lemma 5.9 *The map ψ is invertible and its inverse is given by*

$$\begin{aligned} \psi^{-1} : \quad SB_n^{p,m} \quad &\to \quad SS_n^{p,m} \\ (A_*, B_*, C_*, D_*) \quad &\mapsto \quad (A, B, C, D) \end{aligned}$$

where

$$\left\{ \begin{aligned} B &= B_*(I - D_*^T D_*)^{-1/2} \\ D &= D_*(I - D_*^T D_*)^{-1/2} \\ A &= A_* + B(B^T P + D^T C_*) \\ C &= C_* + D(B^T P + D^T C_*) \end{aligned} \right. \tag{5.19}$$

where P is the stabilizing solution to the GBRRE in (A_*, B_*, C_*, D_*).

It can be shown that the maps ψ and ψ^{-1} preserve input-output equivalence.

Lemma 5.10 *The maps ψ and ψ^{-1} are real analytic.*

Proof: By lemmas 5.6 and 5.4, both the stabilizing solution of the GCARE and $(I + D^T D)^{1/2}$ are real analytic functions of (A, B, C, D), thus the map ψ is a real analytic function of (A, B, C, D). By using lemmas 5.5 and 5.7, it can be shown that the map ψ^{-1} is real analytic as well. $\square$

Since the map ψ is bijective (lemma 5.9) and both ψ and its inverse are real analytic (lemma 5.10), the theorem stated in the beginning of this section follows immediately.

Remark: Note that the map ψ takes state space systems which are Riccati balanced [55] and whose LQG singular values [35] are less than unity to state space systems which are bounded real balanced [56].

5.2 Diffeomorphism between $\Sigma B_n^{m,m}$ and ΣP_n^m

In this section, we shall prove that the set of square bounded real systems is diffeomorphic to the set of positive real systems. The transfer functions associated with these two classes of systems are related by a bijective map known as the Cayley transform [81, 4]. In this section, we shall use the notation $TB_n^{p,m}$ and TP_n^m to denote the sets of rational transfer functions which correspond to $\Sigma B_n^{p,m}$ and ΣP_n^m respectively.

The Cayley transform and its inverse are given by

$$\begin{array}{rlll} \psi: & TB_n^{m,m} & \rightarrow & TP_n^m \\ & B(s) & \mapsto & P(s) = (I - B(s))^{-1}(I + B(s)) \end{array}$$

$$\begin{array}{rlll} \psi^{-1}: & TP_n^m & \rightarrow & TB_n^{m,m} \\ & P(s) & \mapsto & B(s) = (P(s) - I)(P(s) + I)^{-1} \end{array}$$

In terms of the state space matrices, the maps ψ and ψ^{-1} are given by

$$\begin{array}{rlll} \psi_s: & SB_n^{p,m} & \rightarrow & SP_n^m \\ & (A, B, C, D) & \mapsto & (A + B(I - D)^{-1}C, \sqrt{2}B(I - D)^{-1}, \sqrt{2}(I - D)^{-1}C, \\ & & & (I - D)^{-1}(I + D)) \end{array}$$

$$\begin{array}{rlll} \psi_s^{-1}: & SP_n^m & \rightarrow & SB_n^{p,m} \\ & (A, B, C, D) & \mapsto & (A - B(I + D)^{-1}C, \sqrt{2}B(I + D)^{-1}, \sqrt{2}(I + D)^{-1}C, \\ & & & (D - I)(I + D)^{-1}) \end{array}$$

It can be shown that the maps ψ_s and ψ_s^{-1} preserve input-output equivalence.

Lemma 5.11 *The map ψ_s is real analytic.*

Proof: Since the map ψ_s involves only matrix multiplications and matrix inversions, we only need to show that the matrix $I - D$ is invertible.

Since $(A,B,C,D) \in SB_n^{m,m}$, we have $I - D^T D > 0$. Let λ be any eigenvalue of D and v be the corresponding eigenvector.

$$\begin{array}{lrcl} \text{Since} & I - D^T D & > & 0 \\ \Rightarrow & v^*(I - D^T D)v & > & 0 \\ \Rightarrow & v^*v - (Dv)^*(Dv) & > & 0 \\ \Rightarrow & v^*v - \bar{\lambda}\lambda v^*v & > & 0 \\ \Rightarrow & 1 & > & |\lambda|^2 \end{array}$$

Since all the eigenvalues of D have modulus less than one, the matrix $(I - D)$ is invertible. $\square$

Lemma 5.12 *The map ψ_s^{-1} is real analytic.*

Proof: The proof is similar to the one above. As the map ψ_s^{-1} involves only matrix multiplications and inversions, we only need to show that the matrix $I + D$ is invertible.

Since $(A,B,C,D) \in SP_n^m$, we have $D + D^T > 0$. Let λ be any eigenvalue of D, then it can be shown that $Re(\lambda) > 0$. Hence $I + D$ is invertible. $\square$

Theorem 5.2 *$\Sigma B_n^{p,m}$ and ΣP_n^m are real analytic diffeomorphic.*

Proof: Since the map ψ_s is bijective and both ψ_s and ψ_s^{-1} are real analytic, we can conclude that the manifolds $\Sigma B_n^{p,m}$ and ΣP_n^m are real analytic diffeomorphic to each other. $\square$

5.3 Diffeomorphism between $D\Sigma MI_n^m$ and $D\Sigma SM_n^m$

In this section, we shall prove that the set of invertible systems, $D\Sigma MI_n^m$, and the set of stable and minimum-phase systems, $D\Sigma SM_n^m$, are diffeomorphic to each other, as stated in the following theorem.

Theorem 5.3 *The sets $D\Sigma MI_n^m$ and $D\Sigma SM_n^m$ are real analytic diffeomorphic to each other.*

Before we give the proof to the above theorem, we shall first define some notation and give an outline of the proof.

Let (A,B,C,D) be a discrete time state space system in $DSMI_n^m$, the poles and zeros of the system are given respectively by $\{\lambda_i(A)\}$ and $\{\lambda_i(A - BD^{-1}C)\}$, where $\{\lambda_i(M)\}$ denote the set of all the eigenvalues of the matrix M. For any equivalence class $\mathcal{C} = [(A,B,C,D)] \in DSMI_n^m/\sim$, we define a one parameter family of state space systems

$$\mathcal{C}_\mu \;=\; [(\sqrt{\mu}A, \sqrt{\mu}B, C, D)] \;\text{ with } \mu > 0.$$

The poles and zeros of $\mathcal{C}_\mu$ are respectively given by $\{\lambda_i(\sqrt{\mu}A)\}$ and $\{\lambda_i(\sqrt{\mu}(A - BD^{-1}C))\}$. Note that if $\mathcal{C} \in DSMI_n^m/\sim$, then $\mathcal{C}_\mu \in DSSM_n^m/\sim$ if μ is chosen to be less than the following upper bound:

$$\mu \;<\; b(\mathcal{C}) \;=\; \max_i \left(\frac{1}{|\lambda_i(A)|^2}, \frac{1}{|\lambda_i(A - BD^{-1}C)|^2} \right) \tag{5.20}$$

We shall use $H_k(\mathcal{C})$ to denote the k^{th} $(k \geq 0)$ Markov parameter of $\mathcal{C} = [(A, B, C, D)]$. The Markov parameters are defined as follows:

$$H_k = \begin{cases} D & \text{for } k = 0, \\ CA^{k-1}B & \text{for } k \geq 1 \end{cases} \tag{5.21}$$

For $\mathcal{C} \in DSMI_n^m/\sim$, we shall use $\mathcal{C}^{-1}$ to denote the inverse system of $\mathcal{C}$. Note that for $k \geq 1$,

$$H_k(\mathcal{C}_\mu) = \mu^{\frac{k}{2}} H_k(\mathcal{C}) \tag{5.22}$$

$$H_k(\mathcal{C}_\mu^{-1}) = \mu^{\frac{k}{2}} H_k(\mathcal{C}^{-1}) \tag{5.23}$$

The way in which theorem 5.3 will be proved is similar to the one used by Hanzon [26] to prove the diffeomorphism between $D\Sigma M_n^{p,m}$ and $D\Sigma S_n^{p,m}$. Basically, we need to prove that there exists a bijective map between $DSMI_n^m/\sim$ and $DSSM_n^m/\sim$, and both the map and its inverse are real analytic. The following is an outline of the proof:

1. In lemma 5.13, we define two real analytic bijective maps, $f_\mathcal{C}$ and $g_\mathcal{C}$, which share the same image space.

2. The most important definition in the proof is given in definition 5.3 where we define the maps ψ and ν. We then proceed to show that these two maps are inverses of each other (Lemma 5.18) and are real analytic (Lemma 5.19). In order to prove these two lemmas, a number of auxiliary results are needed and we shall outline them here.

   (a) The map ψ takes the system $\mathcal{C} \in DSMI_n^m/\sim$ to the system $\mathcal{C}_{\phi(\mathcal{C})} \in DSSM_n^m/\sim$, and these two systems are related by the equation $g_\mathcal{C}(\phi) = f_\mathcal{C}(1)$. Lemma 5.14 is needed to make sure that this map is well defined and the analyticity of the map follows directly from the result given in lemma 5.15.

   (b) There exists a parallel set of results for the map ν which takes the system $\mathcal{C} \in DSSM_n^m/\sim$ to the system $\mathcal{C}_{\gamma(\mathcal{C})} \in DSMI_n^m/\sim$. These two systems are related by the equation $f_\mathcal{C}(\gamma) = g_\mathcal{C}(1)$. This map is well defined because of lemma 5.16 and is real analytic due to lemma 5.17.

We shall first define the maps $f_\mathcal{C}$ and $g_\mathcal{C}$.

Lemma 5.13 *1. For $n \geq 1$, and $\mathcal{C} = [(A, B, C, D)] \in DSMI_n^m/\sim$, define the polynomial map*

$$\begin{aligned} f_\mathcal{C} : \ (0, \infty) &\rightarrow (0, \infty) \\ \mu &\mapsto f_\mathcal{C}(\mu) \end{aligned}$$

where

$$\begin{aligned} f_\mathcal{C}(\mu) &= \textstyle\sum_{k=1}^{2n} \|H_k(\mathcal{C}_\mu)\|_F^2 + \sum_{k=1}^{2n} \|H_k(\mathcal{C}_\mu^{-1})\|_F^2 \\ &= \textstyle\sum_{k=1}^{2n} \mu^k \left(\|H_k(\mathcal{C})\|_F^2 + \|H_k(\mathcal{C}^{-1})\|_F^2\right) \end{aligned} \tag{5.24}$$

$f_\mathcal{C}$ is monotonically increasing, has positive derivative everywhere and is surjective. Hence $f_\mathcal{C}$ is bijective.

2. *For $n \geq 1$, let $\mathcal{C} = [(A,B,C,D)] \in DSMI_n^m/\sim$ and $b(\mathcal{C})$ be defined by equation 5.20, then $\mathcal{C}_\mu \in DSSM_n^m/\sim \ \forall\, \mu \in (0, b(\mathcal{C}))$.*

   Define the function

$$\begin{array}{rrcl} g_c: & (0, b(\mathcal{C})) & \rightarrow & (0,\infty) \\ & \mu & \mapsto & g_c(\mu) \end{array}$$

   where

$$\begin{aligned} g_c(\mu) &= \textstyle\sum_{k=1}^{\infty} \|H_k(\mathcal{C}_\mu)\|_F^2 \ + \sum_{k=1}^{\infty} \|H_k(\mathcal{C}_\mu^{-1})\|_F^2 \\ &= \textstyle\sum_{k=1}^{\infty} \mu^k \left(\|H_k(\mathcal{C})\|_F^2 \ + \|H_k(\mathcal{C}^{-1})\|_F^2\ \right) \end{aligned} \tag{5.25}$$

   The map g_c is rational in μ. It is monotonically increasing and thus injective. It is also a surjective map and hence it is bijective.

Proof: We shall first show that g_c is a rational map. Note that the two infinite sums in the definition of g_c are in fact given by

$$\textstyle\sum_{k=1}^{\infty} \|H_k(\mathcal{C}_\mu)\|_F^2 \quad = \quad \mathrm{tr}(CLC^T)$$

where L is the solution to the discrete time Lyapunov equation $L - \mu ALA^T = \mu BB^T$, and

$$\textstyle\sum_{k=1}^{\infty} \|H_k(\mathcal{C}_\mu^{-1})\|_F^2 \quad = \quad \mathrm{tr}(D^{-1}CMC^TD^{-T})$$

where M is the solution to the discrete time Lyapunov equation $M - \mu(A - BD^{-1}C)M(A - BD^{-1}C)^T = \mu BD^{-1}D^{-T}B^T$.

As the discrete time Lyapunov equation is a linear equation in its unknowns, the solutions L and M are thus rational functions of the known parameters in the equations. Hence g_c is a rational map in μ.

We shall now show that g_c is a surjective map. First note that for any minimial state space realization (A,B,C,D), the realization $(\sqrt{\mu}A, \sqrt{\mu}B, C, D)$ is also minimal. Hence the map g_c is continuous and it can be shown easily that it is also surjective. □

Lemma 5.14 *For $n \geq 1$ and $\mathcal{C} \in DSMI_n^m/\sim$, there exists a unique $\phi \in (0, b(\mathcal{C}))$ such that*

$$g_c(\phi) \quad = \quad f_c(1) \tag{5.26}$$

Proof: Since $f_c(1) \in (0,\infty)$ and g_c is a bijective map with image space $(0,\infty)$, there exists a unique $\phi \in (0, b(\mathcal{C}))$ such that (5.26) holds. □

Lemma 5.15 *Define the map*

$$\begin{array}{rrcl} \phi: & DSMI_n^m/\sim & \rightarrow & (0,\infty) \\ & \mathcal{C} & \mapsto & \phi(\mathcal{C}) \text{ such that (5.26) holds} \end{array}$$

The map $\phi(\mathcal{C})$ is real analytic.

Proof: Define the rational function in μ, A, B, C and D,

$$
\begin{aligned}
& r(\mu, A, B, C, D) \\
= \ & g_{\mathcal{C}}(\mu) - f_{\mathcal{C}}(1) \\
= \ & \textstyle\sum_{k=1}^{\infty} \left[\|H_k(\mathcal{C}_\mu)\|_F^2 + \|H_k(\mathcal{C}_\mu^{-1})\|_F^2 \right] - \sum_{k=1}^{2n} \left[\|H_k(\mathcal{C})\|_F^2 + \|H_k(\mathcal{C}^{-1})\|_F^2 \right]
\end{aligned}
$$

then the map $\phi(\mathcal{C})$ is the unique solution to the equation $r(\phi(\mathcal{C}), A, B, C, D) = 0$. The derivative of r with respect to μ is

$$
\begin{aligned}
\frac{\partial r}{\partial \mu} &= \frac{\partial}{\partial \mu} \left\{ \textstyle\sum_{k=1}^{\infty} \left(\|H_k(\mathcal{C}_\mu)\|_F^2 + \|H_k(\mathcal{C}_\mu^{-1})\|_F^2 \right) - \sum_{k=1}^{2n} \left(\|H_k(\mathcal{C})\|_F^2 + \|H_k(\mathcal{C}^{-1})\|_F^2 \right) \right\} \\
&= \frac{\partial}{\partial \mu} \left\{ \textstyle\sum_{k=1}^{\infty} \left[\mu^k \|H_k(\mathcal{C})\|_F^2 + \mu^k \|H_k(\mathcal{C}^{-1})\|_F^2 \right] \right\} \\
&= \textstyle\sum_{k=1}^{\infty} k\mu^{k-1} \left[\|H_k(\mathcal{C})\|_F^2 + \|H_k(\mathcal{C}^{-1})\|_F^2 \right] > 0 \quad \forall \mu \in (0, b(\mathcal{C}))
\end{aligned}
$$

Hence it follows from the implicit function theorem that $\phi(\mathcal{C})$ is a real analytic function. □

Lemma 5.16 *For $n \geq 1$ and $\mathcal{C} \in DSSM_n^m/\sim$, there exists a unique $\gamma \in (0, \infty)$ such that*

$$
f_{\mathcal{C}}(\gamma) = g_{\mathcal{C}}(1) \tag{5.27}
$$

Proof: Note that $g_{\mathcal{C}}(1) = \sum_{k=1}^{\infty} \|H_k(\mathcal{C})\|_F^2 + \sum_{k=1}^{\infty} \|H_k(\mathcal{C}^{-1})\|_F^2 \in (0, \infty)$ as $\mathcal{C}$ is a stable and minimum phase system. By lemma 5.13, $f_{\mathcal{C}}$ is a bijective map with image space $(0, \infty)$, hence there exists a unique $\gamma \in (0, \infty)$ such that (5.27) holds. □

Lemma 5.17 *Define the map*

$$
\begin{aligned}
\gamma : \ DSSM_n^m/\sim \ &\to \ (0, \infty) \\
\mathcal{C} \ &\mapsto \ \gamma(\mathcal{C}) \text{ such that (5.27) holds}
\end{aligned}
$$

The map $\gamma(\mathcal{C})$ is real analytic.

Proof: Define the polynomial function in μ, A, B, C and D,

$$
\begin{aligned}
& \tilde{r}(\mu, A, B, C, D) \\
= \ & f_{\mathcal{C}}(\mu) - g_{\mathcal{C}}(1) \\
= \ & \textstyle\sum_{k=1}^{2n} \left(\|H_k(\mathcal{C}_\mu)\|_F^2 + \|H_k(\mathcal{C}_\mu^{-1})\|_F^2 \right) - \sum_{k=1}^{\infty} \left(\|H_k(\mathcal{C})\|_F^2 + \|H_k(\mathcal{C}^{-1})\|_F^2 \right)
\end{aligned}
$$

then the map $\gamma(\mathcal{C})$ is the unique solution to the equation $\tilde{r}(\gamma(\mathcal{C}), A, B, C, D) = 0$. The derivative of $\tilde{r}$ with respect to μ is

$$
\begin{aligned}
\frac{\partial \tilde{r}}{\partial \mu} &= \frac{\partial}{\partial \mu} \left\{ \textstyle\sum_{k=1}^{2n} \left(\|H_k(\mathcal{C}_\mu)\|_F^2 + \|H_k(\mathcal{C}_\mu^{-1})\|_F^2 \right) - \sum_{k=1}^{\infty} \left(\|H_k(\mathcal{C})\|_F^2 + \|H_k(\mathcal{C}^{-1})\|_F^2 \right) \right\} \\
&= \frac{\partial}{\partial \mu} \left\{ \textstyle\sum_{k=1}^{2n} \left(\mu^k \|H_k(\mathcal{C})\|_F^2 + \mu^k \|H_k(\mathcal{C}^{-1})\|_F^2 \right) \right\} \\
&= \textstyle\sum_{k=1}^{2n} k\mu^{k-1} \left(\|H_k(\mathcal{C})\|_F^2 + \|H_k(\mathcal{C}^{-1})\|_F^2 \right) > 0 \quad \forall \mu \in (0, \infty)
\end{aligned}
$$

Hence it follows from the implicit function theorem that $\gamma(\mathcal{C})$ is a real analytic function. □

Definition 5.3 *By using the map $\phi(\mathcal{C})$ we define the map*

$$\begin{array}{rrcl}\psi: & DSMI_n^m/\sim & \rightarrow & DSSM_n^m/\sim \\ & \mathcal{C} & \mapsto & \mathcal{C}_{\phi(\mathcal{C})}\end{array}$$

Similarly, by using the map $\gamma(\mathcal{C})$, we define

$$\begin{array}{rrcl}\nu: & DSSM_n^m/\sim & \rightarrow & DSMI_n^m/\sim \\ & \mathcal{C} & \mapsto & \mathcal{C}_{\gamma(\mathcal{C})}\end{array}$$

Lemma 5.18 *The maps ψ and ν are inverses of each other.*

Proof: Note that by the definitions of ϕ, γ, ψ and ν, we have the following two properties.

1. $\forall \mathcal{C} \in DSMI_n^m/\sim$,

$$\begin{array}{rl} & \sum_{k=1}^{\infty} \ [\|H_k(\psi(\mathcal{C}))\|_F^2 + \|H_k((\psi(\mathcal{C}))^{-1})\|_F^2] \\ = & \sum_{k=1}^{2n} \ [\|H_k(\mathcal{C})\|_F^2 \ + \|H_k(\mathcal{C}^{-1})\|_F^2 \ ]\end{array}$$

2. $\forall \mathcal{C} \in DSSM_n^m/\sim$,

$$\begin{array}{rl} & \sum_{k=1}^{2n} \ [\|H_k(\nu(\mathcal{C}))\|_F^2 + \|H_k((\nu(\mathcal{C}))^{-1})\|_F^2] \\ = & \sum_{k=1}^{\infty} \ [\|H_k(\mathcal{C})\|_F^2 \ + \|H_k(\mathcal{C}^{-1})\|_F^2 \ ]\end{array}$$

We shall first prove that $\psi \circ \nu(\mathcal{C}) = \mathcal{C}$ for any $\mathcal{C}$ in $DSSM_n^m/\sim$.
$\forall \mathcal{C} \in DSSM_n^m/\sim$,

$$\begin{array}{rll} & \sum_{k=1}^{\infty} \ [\|H_k(\psi \circ \nu(\mathcal{C}))\|_F^2 + \|H_k((\psi \circ \nu(\mathcal{C}))^{-1})\|_F^2] & \\ = & \sum_{k=1}^{2n} \ [\|H_k(\nu(\mathcal{C}))\|_F^2 + \|H_k((\nu(\mathcal{C}))^{-1})\|_F^2] & \text{(by property 1)} \\ = & \sum_{k=1}^{\infty} \ [\|H_k(\mathcal{C})\|_F^2 \ + \|H_k(\mathcal{C}^{-1})\|_F^2 \ ] & \text{(by property 2)}\end{array}$$

Since $\psi \circ \nu(\mathcal{C}) = \mathcal{C}_\alpha$ for some $\alpha \in (0, b(\mathcal{C}))$, the above equation can be written as

$$\begin{array}{rrcl} & g_c(\alpha) & = & g_c(1) \\ \Rightarrow & \alpha & = & 1 \quad (g_c \text{ is injective})\end{array}$$

hence $\psi \circ \nu(\mathcal{C}) = \mathcal{C}$ for any $\mathcal{C} \in DSSM_n^m/\sim$. Similarly we can prove that $\nu \circ \psi(\mathcal{C}) = \mathcal{C}$ for all $\mathcal{C} \in DSMI_n^m/\sim$. Thus ψ and ν are inverses of each other. □

Lemma 5.19 *The maps ψ and ν are real analytic.*

Proof: Note that the map ψ takes the equivalence class $\mathcal{C} = [(A, B, C, D)] \in DSMI_n^m/\sim$ to the equivalence class $[(\sqrt{\phi(\mathcal{C})}A, \sqrt{\phi(\mathcal{C})}B, C, D)] \in DSSM_n^m/\sim$. By lemma 5.15, $\phi(\mathcal{C})$ is a real analytic function of (A, B, C, D), it follows that ψ is real analytic. The map ν is real analytic for similar reasons. □

By the results given in lemmas 5.18 and 5.19, we can conclude that the maps given in definition 5.3 are in fact diffeomorphisms. This proves the theorem stated in the beginning of this section.

5.4 Diffeomorphisms between classes of continuous time and discrete time systems

In this section, we shall prove the diffeomorphisms between a number of classes of continuous time systems and their corresponding discrete time counterparts. The proofs are rather straightforward in all the cases and they are included here for completeness. We shall first state the main theorem of this section.

Theorem 5.4 *The two classes of linear systems in each of the following six pairs of manifolds of linear systems are real analytic diffeomorphic to each other.*

1. $\Sigma M_n^{p,m}$ *and* $D\Sigma M_n^{p,m}$,
2. $\Sigma S_n^{p,m}$ *and* $D\Sigma S_n^{p,m}$,
3. $\Sigma B_n^{p,m}$ *and* $D\Sigma B_n^{p,m}$,
4. ΣP_n^{m} *and* $D\Sigma P_n^{m}$,
5. ΣMI_n^{m} *and* $D\Sigma MI_n^{m}$, *and*
6. ΣSM_n^{m} *and* $D\Sigma SM_n^{m}$.

Proof: We shall prove the above theorem in two parts, as follows:

1. **Part I**: We prove the 1^{st} and 5^{th} pairs of diffeomorphisms.
2. **Part II**: We prove the remaining four pairs of diffeomorphisms.

(Part I) Define the map

$$\begin{array}{cccc} \phi : & SM_n^{p,m} & \to & DSM_n^{p,m} \\ & (A_c, B_c, C_c, D_c) & \mapsto & (A_d, B_d, C_d, D_d) = (A_c, B_c, C_c, D_c) \end{array}$$

and it can easily be shown that ϕ is bijective. If $p = m$ and we restrict the domain of ϕ to SMI_n^m, then the restriction induces a bijection between SMI_n^m and $DSMI_n^m$. Since ϕ preserves input-output equivalence and is obviously real analytic, this completes the first part of the proof.

(Part II) We shall now prove the remaining four diffeomorphisms. Let $G_c(s)$ be the transfer function associated with the continuous time system $\sigma_c \in \Sigma S_n^{p,m}$ and $G_d(z)$ be the transfer function associated with the discrete time system $\sigma_d \in D\Sigma S_n^{p,m}$, then we define the following bilinear transformation:

$$G_d(z) = G_c(\frac{z-1}{z+1}) \text{ and whose inverse is} \tag{5.28}$$

$$G_c(s) = G_d(\frac{1+s}{1-s}) \tag{5.29}$$

In terms of state space matrices, the bilinear transformation can be stated as follows [20].

$$\begin{aligned}\psi\ :\qquad\qquad SS_n^{p,m} &\rightarrow DSS_n^{p,m}\\ (A_c,B_c,C_c,D_c) &\mapsto (A_d,B_d,C_d,D_d)\end{aligned}$$

where

$$A_d = (I-A_c)^{-1}(I+A_c) \tag{5.30}$$
$$B_d = \sqrt{2}(I-A_c)^{-1}B_c \tag{5.31}$$
$$C_d = \sqrt{2}C_c(I-A_c)^{-1} \tag{5.32}$$
$$D_d = D_c + C_c(I-A_c)^{-1}B_c \tag{5.33}$$

It can be shown that the inverse of ψ is given by

$$\begin{aligned}\psi^{-1}\ :\qquad\qquad DSS_n^{p,m} &\rightarrow SS_n^{p,m}\\ (A_d,B_d,C_d,D_d) &\mapsto (A_c,B_c,C_c,D_c)\end{aligned}$$

where

$$A_c = (I+A_d)^{-1}(A_d-I) \tag{5.34}$$
$$B_c = \sqrt{2}(I+A_d)^{-1}B_d \tag{5.35}$$
$$C_c = \sqrt{2}C_d(I+A_d)^{-1} \tag{5.36}$$
$$D_c = D_d - C_d(I+A_d)^{-1}B_d \tag{5.37}$$

Note that if we restrict the domain of the map ψ to $SB_n^{p,m}$, then the restriction induces a bijection between $SB_n^{p,m}$ and $DSB_n^{p,m}$. If $p=m$, then ψ induces a bijection between SP_n^m and DSP_n^m as well as between SSM_n^m and $DSSM_n^m$. Also note that the transformations ψ and ψ^{-1} preserve input-output equivalence.

Now, it remains to prove that the maps ψ and ψ^{-1} are real analytic. Since the maps ψ and ψ^{-1} consists of matrix multiplications and matrix inversions, we only need to make sure that the matrices $(I-A_c)$ and $(I+A_d)$ are invertible. The non-singularities of these two matrices follow from the fact that A_c and A_d are respectively continuous time and discrete time stable. This completes the second part of the proof. □

5.5 Diffeomorphisms between classes of scalar systems

In this section, we shall prove the diffeomorphisms between a number of classes of scalar systems. The definitions of these classes of systems were given in definitions 3.1, 3.4, 4.1 and 4.2. The main theorem of this section has already been stated in chapter 4 but we shall repeat it here for ease of reference.

Theorem 4.4 The following sets of SISO systems are real analytic diffeomorphic to each other: $\Sigma M_n^{1,1}$, $\Sigma S_n^{1,1}$, $\Sigma B_n^{1,1}$, ΣP_n^1, ΣMI_{n+}^1, ΣMI_{n-}^1, ΣSM_{n+}^1, ΣSM_{n-}^1, $D\Sigma M_n^{1,1}$, $D\Sigma S_n^{1,1}$, $D\Sigma B_n^{1,1}$, $D\Sigma P_n^1$, $D\Sigma MI_{n+}^1$, $D\Sigma MI_{n-}^1$, $D\Sigma SM_{n+}^1$ and $D\Sigma SM_{n-}^1$.

The above theorem can be shown by proving the following three theorems and by the fact that diffeomorphism is an equivalence relation.

Theorem 5.5 *The following sets of SISO systems are real analytic diffeomorphic to each other:* $\Sigma M_n^{1,1}$, $\Sigma S_n^{1,1}$, $\Sigma B_n^{1,1}$, ΣP_n^1, $D\Sigma M_n^{1,1}$, $D\Sigma S_n^{1,1}$, $D\Sigma B_n^{1,1}$ *and* $D\Sigma P_n^1$.

Theorem 5.6 *The following sets of SISO systems are real analytic diffeomorphic to each other:* ΣMI_{n+}^1, ΣMI_{n-}^1, ΣSM_{n+}^1, ΣSM_{n-}^1, $D\Sigma MI_{n+}^1$, $D\Sigma MI_{n-}^1$, $D\Sigma SM_{n+}^1$ *and* $D\Sigma SM_{n-}^1$.

Theorem 5.7 ΣP_n^1 *and* ΣSM_{n+}^1 *are real analytic diffeomorphic to each other.*

Note that theorem 5.5 is simply a special case of theorem 4.2 with $m = 1$. In order to prove theorem 5.6, we need to use theorem 4.3 and the fact that bilinear transform takes the point $s = j\infty$ on the imaginary axis to the point $z = -1$ on the unit circle. Finally, theorem 5.7 provides us with the link between the two groups of diffeomorphisms stated in theorems 5.5 and 5.6, and we shall prove this theorem in detail in this section.

Let us first recall the definitions of ΣP_n^1 and ΣSM_{n+}^1. In the following definitions, $g_\sigma(s)$ denotes the transfer function associated with the continuous time system σ.

1. Scalar strictly positive real systems
   - $\Sigma P_n^1 = \{\sigma \in \Sigma S_n^{1,1} : g_\sigma(\infty) + g_\sigma(-\infty) > 0, g_\sigma(j\omega) + g_\sigma(-j\omega) > 0 \; \forall \omega \in \mathbb{R}\}$
2. Scalar stable and strictly minimum-phase systems with positive feedthrough term
   - $\Sigma SM_{n+}^1 = \{\sigma \in \Sigma SM_n^1 : g_\sigma(\infty) > 0\}$

We shall use SP_n^1 and SSM_{n+}^1 to denote the corresponding sets of minimal state space realizations.

In order to prove theorem 5.7, we basically need to prove that there exists a bijective map between these two manifolds and both this map and its inverse are real analytic. We shall first give an outline of the proof.

1. In lemma 5.20, we shall state a standard result which relates the state space description of a positive real system to the positive real Riccati equation (PRRE). We then prove in lemma 5.21 that the stabilizing solution to the PRRE is a real analytic function.
2. Lemma 5.22 is on spectral factorization and it will be made use of in definition 5.4 to define the map ψ from SP_n^1 to SSM_{n+}^1. This map will then be shown to be invertible in lemma 5.23.
3. Finally, we shall show in lemma 5.24 that both ψ and its inverse are real analytic maps. Hence the theorem.

Lemma 5.20 *([4]) The state space system* $(A, B, C, D) \in SS_n^{m,m}$ *with* $D + D^T > 0$ *is strictly positive real if and only if there exists a symmetric positive definite solution to the PRRE*

$$A^T P + PA - (C - B^T P)^T (D + D^T)^{-1} (C - B^T P) \;=\; 0 \qquad (5.38)$$

such that $A - B(D + D^T)^{-1}(C - B^T P)$ *is asymptotically stable.*

Lemma 5.21 *Define the map*

$$
\begin{array}{rccc}
P_*: & SP_n^m & \rightarrow & \mathbb{R}^{n\times n} \\
 & (A,B,C,D) & \mapsto & P_*(A,B,C,D)
\end{array}
$$

where P_* is the symmetric positive definite solution to the PRRE such that $A - B(D + D^T)^{-1}(C - B^T P)$ is asymptotically stable.

P_* is an analytic function of (A,B,C,D) on the open set SP_n^m.

Proof: This lemma can be proved by using the implicit function theorem in a way similar to proving lemma 5.2. The proof is omitted. □

Lemma 5.22 *(Spectral factorization [4]) Let $(A,B,C,D) \in SP_n^m$ be a positive real state space system and $P(s) = D + C(sI - A)^{-1}B$ be the corresponding transfer function, then*

$$
P(j\omega) + P(-j\omega)^T \;=\; M(-j\omega)^T M(j\omega) \quad \forall \omega \in \mathbb{R} \cup \{\infty\} \tag{5.39}
$$

for some stable and minimum-phase transfer function $M(s)$. Furthermore, a minimal realization of $M(s)$ is given by

$$
(A, B, C, D_m^{-T}(C - B^T P), D_m) \tag{5.40}
$$

where P is the stabilizing solution to the PRRE associated with (A,B,C,D) and $D_m \in \mathbb{R}^{m\times m}$ is such that $D_m^T D_m = D + D^T$.

Definition 5.4 *Define the map*

$$
\begin{array}{rccc}
\psi: & SP_n^1 & \rightarrow & SSM_{n+}^1 \\
 & (A,B,C,d) & \mapsto & (A_m, B_m, C_m, d_m)
\end{array}
$$

where

$$
\left\{
\begin{array}{rcl}
A_m & = & A \\
B_m & = & B \\
C_m & = & \frac{1}{\sqrt{2d}}(C - B^T P) \\
d_m & = & \sqrt{2d}
\end{array}
\right. \tag{5.41}
$$

where P is the symmetric positive definite solution to the PRRE

$$
A^T P + PA + (C - B^T P)^T \frac{1}{2d}(C - B^T P) \;=\; 0 \tag{5.42}
$$

such that $A - \frac{1}{2d}B(C - B^T P)$ is asymptotically stable.

Lemma 5.23 *The map ψ defined in definition 5.4 is a bijective map and its inverse is given by*

$$
\begin{array}{rccc}
\nu: & SSM_{n+}^1 & \rightarrow & SP_n^1 \\
 & (A_m, B_m, C_m, d_m) & \mapsto & (A,B,C,d)
\end{array}
$$

where

$$\left\{\begin{array}{rcl} A &=& A_m \\ B &=& B_m \\ C &=& d_m C_m + B_m^T P_m \\ d &=& \frac{1}{2} d_m^2 \end{array}\right. \tag{5.43}$$

where P_m *is the solution to*

$$A_m^T P_m + P_m A_m + C_m^T C_m \;=\; 0 \tag{5.44}$$

Proof: By replacing A and $\frac{1}{\sqrt{2d}}(C - B^T P)$ by A_m and C_m respectively in equation 5.42, it turns out immediately that P is also the solution to the Lyapunov equation $A_m^T P + P A_m + C_m^T C_m = 0$. Hence ν is the inverse of ψ. □

Note that both ψ and ν preserve input-output equivalence.

Lemma 5.24 *The maps* ψ *and* ν *are real analytic.*

Proof: This lemma can be proved by observing the following two facts:

1. P is a real analytic function of (A, B, C, d), by lemma 5.21, and
2. P_m is a real analytic function of (A_m, C_m) as A_m is stable and the matrix Lyapunov equation is a linear equation in its unknowns. □

Since there exists a bijective map between the sets SP_n^1 and SSM_{n+}^1, and both this map and its inverse are real analytic, hence theorem 5.7.

Remark:

The reader may wonder whether theorem 5.7 can be extended to the multivariable case, the answer is no but the reason behind it is not very straightforward. Let us first define the following sets of matrices.

Definition 5.5 *1.* $S_P = \{M \in \mathbb{R}^{m\times m} : M + M^T > 0\}$

2. $S_{MI} = \{M \in \mathbb{R}^{m\times m} : M^T M > 0$ *and* $det(M) > 0\}$

3. $S(m) = \{M \in \mathbb{R}^{m\times m} : M + M^T = 0\}$ *i.e. skew-symmetric matrices*

4. $SO(m) = \{M \in \mathbb{R}^{m\times m} : MM^T = M^T M = I$ *and* $det(M) = 1\}$ *i.e. the special orthogonal group*

The reason why the spectral factorization theorem cannot be used to give a multivariable extension of theorem 5.7 is that the sets S_P and S_{MI} cannot be diffeomorphic to each other. We shall prove this by the method of contradiction.

Assume there exists a diffeomorphism ϕ between S_P and S_{MI} which is given by

$$\begin{aligned} \phi: \quad S_P &\rightarrow S_{MI} \\ P &\mapsto M \\ \text{such that } P + P^T &= M^T M \end{aligned} \tag{5.45}$$

We shall show that if S_P and S_{MI} are diffeomorphic to each other, then it implies that $S(m)$ and $SO(m)$ are diffeomorphic to each other.

Let P_0 be an arbitrary element in S_P and $M_0 = \phi(P_0)$, then

$$P_0 + P_0^T = M_0^T M_0 \tag{5.46}$$

Let Δ be an arbitrary element in $S(m)$, then $P_0 + \Delta \in S_P$ and

$$\begin{aligned} (P_0 + \Delta) + (P_0 + \Delta)^T &= \phi(P_0 + \Delta)^T \phi(P_0 + \Delta) \\ \Rightarrow \qquad P_0 + P_0^T &= \phi(P_0 + \Delta)^T \phi(P_0 + \Delta) \\ \Rightarrow \qquad M_0^T M_0 &= \phi(P_0 + \Delta)^T \phi(P_0 + \Delta) \\ \Rightarrow \quad \phi(P_0 + \Delta) M_0^{-1} \in SO(m) & \end{aligned} \tag{5.47}$$

Define the map

$$\begin{aligned} F: \quad S(m) &\rightarrow SO(m) \\ \Delta &\mapsto \phi(P_0 + \Delta) M_0^{-1} \end{aligned}$$

then it can be shown that the inverse of F is given by

$$\begin{aligned} G: \quad SO(m) &\rightarrow S(m) \\ U &\mapsto \phi^{-1}(U M_0) - P_0 \end{aligned}$$

Since ϕ is a diffeomorphism, it implies that the two maps F and G are also diffeomorphisms. In other words, the sets $S(m)$ and $SO(m)$ are diffeomorphic to each other. However, these two sets cannot be diffeomorphic to each other as $SO(m)$ is a compact set but $S(m)$ is not. Hence the sets S_P and S_{MI} cannot be diffeomorphic to each other.

5.6 Diffeomorphisms between classes of discrete time SISO systems

In this section, we shall prove that 3 classes of SISO discrete time systems are diffeomorphic to each other. The main theorem of this section has already been stated in chapter 4 but we shall state it below for ease of reference.

Theorem 4.5 The following 3 manifolds of SISO discrete time systems are real analytic diffeomorphic to each other: $DRat(n;r)$, $DRat_S(n;r)$ and $DRat_{SM}(n;r)$.

The above theorem can be shown by proving the following two theorems and by the fact that diffeomorphism is an equivalence relation.

Theorem 5.8 *$DRat(n;r)$ and $DRat_S(n;r)$ are real analytic diffeomorphic to each other.*

Theorem 5.9 *$DRat(n;r)$ and $DRat_{SM}(n;r)$ are real analytic diffeomorphic to each other.*

Note that theorem 5.9 is an extension to the following lemma.

Lemma 5.25 *(Helmke [33]) The topological spaces $DRat(n;r)$ and $DRat_{SM}(n;r)$ are homeomorphic to each other.*

In order to make the language in this section less clumsy, we shall ignore the difference between a dynamical system and its corresponding transfer function. As an example, we shall write '$g(z) \in DRat(n;r)$' instead of '$\sigma \in DRat(n;r)$ and $g(z)$ is the corresponding transfer function'.

Before we give the proofs of the two theorems, we shall first define some notation for this section.

For any discrete time system $g(z) \in DRat(n;r)$, we introduce a one-parameter family of systems

$$g_\mu(z) \quad = \quad g(\frac{z}{\sqrt{\mu}}) \ \text{ with } \ \mu > 0$$

If z_o is a pole (zero) of g(z) , then $\sqrt{\mu} z_o$ is a pole (zero) of $g_\mu(z)$. So if μ is chosen small enough then $g_\mu(z)$ will be a stable and minimum-phase discrete time system. Let p_i $(i = 1, ..., n)$ and z_i $(i = 1, .., n-r)$ be respectively the poles and zeros of $g(z)$, then the upper bound of μ such that $g_\mu(z)$ is stable and minimum-phase is given by

$$b(g) \quad = \quad \min\left(\frac{1}{\max\limits_i |p_i|}, \frac{1}{\max\limits_i |z_i|}\right) \tag{5.48}$$

Let $p(z)$ and $q(z)$ be respectively the numerator and denominator of $g(z)$ and we shall assume throughout this section that the leading coefficient of $q(z)$ is 1, so that the decomposition of $g(z)$ into $p(z)$ and $q(z)$ is unique. In other words,

$$\begin{aligned} g(z) &= \tfrac{p(z)}{q(z)} \ \text{ with} && (5.49) \\ p(z) &= p_{n-r} z^{n-r} + p_{n-r-1} z^{n-r-1} + ... + p_0 \ \text{ where } \ p_{n-r} \neq 0 && (5.50) \\ q(z) &= z^n + q_{n-1} z^{n-1} + ... + q_0 && (5.51) \end{aligned}$$

In a similar way, we define

$$\begin{aligned} p_\mu(z) &= p(\frac{z}{\sqrt{\mu}}) && (5.52) \\ q_\mu(z) &= q(\frac{z}{\sqrt{\mu}}) \ \text{ where } \ \mu > 0 && (5.53) \end{aligned}$$

and note that

$$g_\mu(z) \quad = \quad \frac{p_\mu(z)}{q_\mu(z)} \tag{5.54}$$

We shall use $H_k(g)$ to denote the k^{th} Markov parameter of the discrete time system $g(z)$ and in particular we have the following relationships

$$H_k(\frac{1}{p_\mu(z)}) = \mu^{\frac{k}{2}} H_k(\frac{1}{p(z)}) \tag{5.55}$$

$$H_k(\frac{1}{q_\mu(z)}) = \mu^{\frac{k}{2}} H_k(\frac{1}{q(z)}) \tag{5.56}$$

Proof of theorem 5.8
Define the map

$$\begin{array}{rccc} \eta: & DRat_S(n;r) & \rightarrow & DRat(n;r) \\ & g(z) & \mapsto & g_\mu(z) = g(\frac{z}{\sqrt{\mu}}) \end{array}$$

such that

$$\|g_\mu(z)\|_{2n}^2 = \|g(z)\|_2^2 \text{ where} \tag{5.57}$$

$$\|g(z)\|_{2n} = (\sum_{k=1}^{2n} \|H_k(g)\|_F^2)^{1/2} \text{ for } g(z) \in DRat(n;r) \text{ and} \tag{5.58}$$

$$\|g(z)\|_2^2 = (\sum_{k=1}^{\infty} \|H_k(g)\|_F^2)^{1/2} \text{ for } g(z) \in DRat_S(n;r) \tag{5.59}$$

Note that η is formulated in a way exactly identical to the map G defined by equation 4.18 in [26]. It can be proved, using arguments identical to those used in [26], that η is bijective. The real analyticities of the map η and its inverse follow directly from the fact that both G and its inverse are real analytic. This completes the proof of the theorem. □

The technique which we shall use to prove theorem 5.9 is the same as the one used in [26] and also in section 5.3. As the proof is rather long and a number of auxiliary lemmas are required, we shall first give an outline of the proof.

1. In lemmas 5.26 and 5.27, we define two real analytic bijective maps, α_g and β_g, which share the same image space.

2. The diffeomorphisms of this theorem, ψ and ν, are defined in lemma 5.32. We show in the same lemma that these two maps are inverses of each other. The analyticities of these two maps are then proved in lemma 5.33. A number of auxiliary results are needed in order to prove these two lemmas and we shall outline them here.

   (a) The map ψ takes the discrete time system $g(z) \in DRat(n;r)$ to the system $g_{\phi(g)}(z) \in DRat_{SM}(n;r)$, and these two systems are related by the equation $\beta_g(\phi) = \alpha_g(1)$. Lemma 5.28 is needed to make sure that this map is well defined and the analyticity of the map follows directly from the result given in lemma 5.29.

   (b) A parallel set of results exists for the map ν which takes the system $g(z) \in DRat_{SM}(n;r)$ to the system $g_{\gamma(g)}(z) \in DRat(n;r)$. These two systems are related by the equation $\alpha_g(\gamma) = \beta_g(1)$. This map is well defined because of lemma 5.30 and is real analytic due to lemma 5.31.

We shall start with the auxiliary lemmas.

Lemma 5.26 *Let* $g(z) = \frac{p(z)}{q(z)} \in DRat(n;r)$*, define the polynomial map*

$$\begin{array}{rccc} \alpha_g : & (0,\infty) & \rightarrow & (0,\infty) \\ & \mu & \mapsto & \alpha_g(\mu) \end{array}$$

where

$$\begin{aligned} \alpha_g(\mu) &= \sum_{k=1}^{2n} (H_k(\frac{1}{q_\mu(z)}))^2 + \sum_{k=1}^{2(n-r)} (H_k(\frac{1}{p_\mu(z)}))^2 \qquad (5.60) \\ &= \sum_{k=1}^{2n} \mu^k (H_k(\frac{1}{q(z)}))^2 + \sum_{k=1}^{2(n-r)} \mu^k (H_k(\frac{1}{p(z)}))^2 \end{aligned}$$

α_g *is monotonically increasing and surjective, hence it is a bijection.*

Proof: The proof is quite straightforward and is omitted. □

Lemma 5.27 *Let* $g(z) = \frac{p(z)}{q(z)} \in DRat(n;r)$*, we have* $g_\mu(z) \in DRat_{SM}(n;r)$ $\forall \mu \in (0, b(g))$ *where* $b(g)$ *is defined in equation 5.48. Define the map*

$$\begin{array}{rccc} \beta_g : & (0,b(g)) & \rightarrow & (0,\infty) \\ & \mu & \mapsto & \beta_g(\mu) \end{array}$$

where

$$\begin{aligned} \beta_g(\mu) &= \|\frac{1}{p_\mu(z)}\|_2^2 + \|\frac{1}{q_\mu(z)}\|_2^2 \qquad (5.61) \\ &= \sum_{k=1}^{\infty} (H_k(\frac{1}{p_\mu(z)}))^2 + \sum_{k=1}^{\infty} (H_k(\frac{1}{q_\mu(z)}))^2 \qquad (5.62) \\ &= \sum_{k=1}^{\infty} \mu^k (H_k(\frac{1}{p(z)}))^2 + \sum_{k=1}^{\infty} \mu^k (H_k(\frac{1}{q(z)}))^2 \end{aligned}$$

β_g *is a rational map in* μ*. It is monotonically increasing and surjective, hence it is a bijective map.*

Proof: We shall show that $\beta_g(\mu)$ is a rational function in μ by showing that both $\|\frac{1}{p_\mu(z)}\|_2^2$ and $\|\frac{1}{q_\mu(z)}\|_2^2$ are rational functions in μ. Note that the discrete time transfer function $\frac{1}{q_\mu(z)}$ can be realized by the following state space system

$$\begin{cases} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] \end{cases}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & 0 & 1 \\ -q_0 & -q_1 & -q_2 & \dots & -q_{n-1} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}$$

It is a standard result from linear system theory that

$$\|\frac{1}{q_\mu(z)}\|_2^2 \;=\; CPC^T$$

where P is the solution to the discrete time Lyapunov equation $P - (\sqrt{\mu}A)P(\sqrt{\mu}A)^T = (\sqrt{\mu}B)(\sqrt{\mu}B)^T \Leftrightarrow P - \mu APA^T = \mu BB^T$. Since the solution to the discrete time Lyapunov equation is a rational function of the unknown parameters in the equation, $\|\frac{1}{q_\mu(z)}\|_2^2$ is thus a rational function of μ. In a similar way, we can show that $\|\frac{1}{p_\mu(z)}\|_2^2$ is also a rational function of μ. The remaining properties of β_g can be proved rather easily and the proof is omitted. □

Lemma 5.28 *For every $g(z) \in DRat(n;r)$, there exists an unique $\phi \in (0, b(g))$ such that*

$$\beta_g(\phi) \;=\; \alpha_g(1) \tag{5.63}$$

Proof: This follows from the fact that β_g is a bijective map. □

Lemma 5.29 *Define the map*

$$\begin{array}{rccc} \phi: & DRat(n;r) & \to & (0,\infty) \\ & g(z) & \mapsto & \phi(g) \text{ such that (5.63) holds} \end{array}$$

The map ϕ is real analytic.

Proof: Define the function

$$\begin{aligned} & r(\mu, \underline{p}, \underline{q}) \\ = \;& \beta_g(\mu) - \alpha_g(1) \\ = \;& \sum_{k=1}^{\infty} (H_k(\frac{1}{p_\mu(z)}))^2 + \sum_{k=1}^{\infty} (H_k(\frac{1}{q_\mu(z)}))^2 - \sum_{k=1}^{2(n-r)} (H_k(\frac{1}{p(z)}))^2 - \sum_{k=1}^{2n} (H_k(\frac{1}{q(z)}))^2 \end{aligned}$$

where $\underline{p} = [p_0, p_1, ..., p_{n-r}]^T$ and $\underline{q} = [q_0, q_1, ..., q_{n-1}]^T$. The map $\phi(g)$ is the unique solution to the equation $f(\phi(g), \underline{p}, \underline{q}) = 0$ and

$$\begin{aligned} \frac{\partial f}{\partial \mu} \;&=\; \frac{\partial}{\partial \mu}\left\{ \sum_{k=1}^{\infty} (H_k(\frac{1}{p_\mu(z)}))^2 + \sum_{k=1}^{\infty} (H_k(\frac{1}{q_\mu(z)}))^2 \right\} \\ &=\; \sum_{k=1}^{\infty} k\mu^{k-1}(H_k(\frac{1}{p(z)}))^2 + \sum_{k=1}^{\infty} k\mu^{k-1}(H_k(\frac{1}{q(z)}))^2 \;>\; 0 \quad \forall\, \mu \in (0, b(g)) \end{aligned}$$

Hence by the implicit function theorem, $\phi(g)$ is a real analytic function of $\underline{p}$ and $\underline{q}$. □

Lemma 5.30 *For every $g(z) \in DRat_{SM}(n;r)$, there exists an unique $\gamma \in (0,\infty)$, such that*

$$\alpha_g(\gamma) \;=\; \beta_g(1) \tag{5.64}$$

Proof: This follows from the fact that α_g is a bijective map. □

Lemma 5.31 *Define the map*

$$\begin{array}{rcll} \gamma: & DRat_{SM}(n;r) & \to & (0,\infty) \\ & g(z) & \mapsto & \gamma(g) \textit{ such that } (5.64) \textit{ holds} \end{array}$$

The map $\gamma(g)$ is real analytic.

Proof: Define the map

$$\begin{aligned} & \tilde{f}(\mu, \underline{p}, \underline{q}) \\ = \; & \alpha_g(\mu) - \beta_g(1) \\ = \; & \sum_{k=1}^{2(n-r)} (H_k(\frac{1}{p_\mu(z)}))^2 + \sum_{k=1}^{2n} (H_k(\frac{1}{q_\mu(z)}))^2 - \sum_{k=1}^{\infty} (H_k(\frac{1}{p(z)}))^2 - \sum_{k=1}^{\infty} (H_k(\frac{1}{q(z)}))^2 \end{aligned}$$

where $\underline{p}$ and $\underline{q}$ are the two vectors defined in the proof of lemma 5.29.

The map $\gamma(g)$ is implicitly defined by the unique solution to the equation $\tilde{f}(\gamma(g), \underline{p}, \underline{q}) = 0$. The derivative of $\tilde{f}$ with respect to μ is given by

$$\begin{aligned} \frac{\partial \tilde{f}}{\partial \mu} &= \frac{\partial}{\partial \mu} \left\{ \sum_{k=1}^{2(n-r)} (H_k(\frac{1}{p_\mu(z)}))^2 + \sum_{k=1}^{2n} (H_k(\frac{1}{q_\mu(z)}))^2 \right\} \\ &= \sum_{k=1}^{2(n-r)} k\mu^{k-1}(H_k(\frac{1}{p(z)}))^2 + \sum_{k=1}^{2n} k\mu^{k-1}(H_k(\frac{1}{q(z)}))^2 > 0 \quad \forall\, \mu \in (0,\infty) \end{aligned}$$

Thus by the implicit function theorem, the map $\gamma(g)$ is a real analytic function of $\underline{p}$ and $\underline{q}$. □

Lemma 5.32 *Define the maps*

1.

$$\begin{array}{rcll} \psi: & DRat(n;r) & \to & DRat_{SM}(n;r) \\ & g(z) & \mapsto & g_{\phi(g)}(z) \end{array}$$

where $\phi(g)$ is the function defined in lemma (5.29) and we shall write $g_\phi(z)$ instead of $g_{\phi(g)}(z)$ for brevity.

2.

$$\begin{array}{rcll} \nu: & DRat_{SM}(n;r) & \to & DRat(n;r) \\ & g(z) & \mapsto & g_{\gamma(g)}(z) \end{array}$$

where $\gamma(g)$ is the function defined in lemma (5.31) and we shall write $g_\gamma(z)$ instead of $g_{\gamma(g)}(z)$ for brevity.

ψ is the inverse of ν.

Proof: Note that by the definitions of $\phi(g)$ and $\gamma(g)$, we have the following 2 properties:

1. $\forall\, g(z) = \dfrac{p(z)}{q(z)} \in DRat(n;r)$,

$$\sum_{k=1}^{\infty} (H_k(\frac{1}{p_\phi(z)}))^2 + \sum_{k=1}^{\infty} (H_k(\frac{1}{q_\phi(z)}))^2 = \sum_{k=1}^{2(n-r)} (H_k(\frac{1}{p(z)}))^2 + \sum_{k=1}^{2n} (H_k(\frac{1}{q(z)}))^2$$

2. $\forall\, g(z) = \dfrac{p(z)}{q(z)} \in DRat_{SM}(n;r)$,

$$\sum_{k=1}^{2(n-r)} (H_k(\frac{1}{p_\gamma(z)}))^2 + \sum_{k=1}^{2n} (H_k(\frac{1}{q_\gamma(z)}))^2 = \sum_{k=1}^{\infty} (H_k(\frac{1}{p(z)}))^2 + \sum_{k=1}^{\infty} (H_k(\frac{1}{q(z)}))^2$$

We shall first show that $\nu \circ \psi(g(z)) = g(z)$ for all $g(z) \in DRat(n;r)$.
For any $g(z) = \dfrac{p(z)}{q(z)} \in DRat(n;r)$, let

$$\psi(g(z)) = g_\phi(z) = \frac{p_\phi(z)}{q_\phi(z)}$$

From property (1) above, we have

$$\sum_{k=1}^{\infty} (H_k(\frac{1}{p_\phi(z)}))^2 + \sum_{k=1}^{\infty} (H_k(\frac{1}{q_\phi(z)}))^2 = \sum_{k=1}^{2(n-r)} (H_k(\frac{1}{p(z)}))^2 + \sum_{k=1}^{2n} (H_k(\frac{1}{q(z)}))^2 \quad (5.65)$$

Also let

$$\nu \circ \psi(g(z)) = \nu(g_\phi(z)) = g_{\phi\gamma}(z) = \frac{p_{\phi\gamma}(z)}{q_{\phi\gamma}(z)}$$

We shall write $g_\phi(z) \in DRat_{SM}(n;r)$ as $\frac{\phi^{\frac{n}{2}} p_\phi(z)}{\phi^{\frac{n}{2}} q_\phi(z)}$ so that the leading coefficient of the polynomial at the denominator has unit leading coefficient;by property (2), we have

$$\begin{aligned}
& \sum_{k=1}^{2(n-r)} (H_k(\frac{1}{\phi^{\frac{n}{2}} p_{\phi\gamma}(z)}))^2 + \sum_{k=1}^{2n} (H_k(\frac{1}{\phi^{\frac{n}{2}} q_{\phi\gamma}(z)}))^2 \\
& = \sum_{k=1}^{\infty} (H_k(\frac{1}{\phi^{\frac{n}{2}} p_\phi(z)}))^2 + \sum_{k=1}^{\infty} (H_k(\frac{1}{\phi^{\frac{n}{2}} q_\phi(z)}))^2 \\
\Leftrightarrow \quad & \sum_{k=1}^{2(n-r)} (H_k(\frac{1}{p_{\phi\gamma}(z)}))^2 + \sum_{k=1}^{2n} (H_k(\frac{1}{q_{\phi\gamma}(z)}))^2 \\
& = \sum_{k=1}^{\infty} (H_k(\frac{1}{p_\phi(z)}))^2 + \sum_{k=1}^{\infty} (H_k(\frac{1}{q_\phi(z)}))^2 \qquad (5.66)
\end{aligned}$$

By comparing equations (5.65) and (5.66), we have

$$\begin{aligned}
\sum_{k=1}^{2(n-r)} (H_k(\tfrac{1}{p_{\phi\gamma}(z)}))^2 + \sum_{k=1}^{2n} (H_k(\tfrac{1}{q_{\phi\gamma}(z)}))^2 &= \sum_{k=1}^{2(n-r)} (H_k(\tfrac{1}{p(z)}))^2 + \sum_{k=1}^{2n} (H_k(\tfrac{1}{q(z)}))^2 \\
\Leftrightarrow \qquad \alpha_g(\phi\gamma) &= \alpha_g(1) \\
\Leftrightarrow \qquad \phi\gamma &= 1 \quad \text{as } \alpha_g \text{ is injective}
\end{aligned}$$

Hence $\nu \circ \psi(g(z)) = g(z)$ for any system $g(z) \in DRat(n;r)$. It can be proved in a similar way that $\psi \circ \nu(g(z)) = g(z)$ for all $g(z) \in DRat_{SM}(n;r)$. Hence ψ and ν are inverses of each other. $\square$

Lemma 5.33 *The maps ψ and ν are real analytic.*

Proof: The map ψ takes $g(z) \in DRat(n;r)$ to $g_\phi(z) = g(z/\sqrt{\phi(g)}) \in DRat_{SM}(n;r)$ and we prove in lemma 5.29 that $\phi(g)$ is a real analytic function of g, hence it follows that ψ is also a real analytic map. The map ν is real analytic for similar reasons. $\square$

Now we are ready to prove theorem 5.9.

Proof of theorem 5.9: As the map ψ defines a bijection between the sets $DRat(n;r)$ and $DRat_{SM}(n;r)$ (by lemma 5.32), and both ψ and its inverse ν are real analytic (by lemma 5.33), hence $DRat(n;r)$ and $DRat_{SM}(n;r)$ are real analytic diffeomorphic to each other. $\square$

Chapter 6

Bundle diffeomorphisms

We have proved in the previous chapters that a number of classes of linear systems are diffeomorphic to each other. Each of these classes of linear systems, which has the structure of a differential manifold, is associated with two different bundles, namely the state bundle and the principal fibre bundle. These bundles were first introduced by Hazewinkel [30] in the investigation of the (non)-existence of continuous canonical forms on the class of fixed order systems. They were also used by Delchamps in [14, 15] to investigate the global convergence properties of system identification algorithms. By defining a metric on the state bundle of the class of fixed order systems, Hanzon [23] proposed a prediction error system identification algorithm which is coordinate free, or in other words, independent of the parametrization used.

This chapter consists of two sections. Section 6.1 is on the diffeomorphisms between state bundles and section 6.2 is on the diffeomorphisms between principal fibre bundles.

6.1 State bundle diffeomorphisms

In this section, we shall prove the bundle diffeomorphisms between the state bundles associated with those classes of linear systems which were defined in section 3.1. Before proving the main theorems in this section, we shall first look at what systems-with-states and state bundles are.

The way which systems-with-states and state bundles are defined is similar for all classes of systems and we shall take $\Sigma M_n^{p,m}$ as an example. Let $(A, B, C, D; x)$ denote a state space representation $(A, B, C, D) \in SM_n^{p,m}$ together with a state vector $x \in \mathbb{R}^n$, then $(A, B, C, D; x)$ and $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}; \tilde{x})$ are input-output equivalent if and only if there exists $T \in Gl_n(\mathbb{R})$ such that $(TAT^{-1}, TB, CT^{-1}, D; Tx) = (\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}; \tilde{x})$. We shall denote the equivalence classes with respect to this input-output equivalence by $[(A, B, C, D; x)]$; these equivalence classes are known as systems-with-states. The collection of all the systems-with-states $[(A, B, C, D; x)]$ such that $(A, B, C, D) \in SM_n^{p,m}$ is the state bundle associated with $\Sigma M_n^{p,m}$. This state bundle will be denoted by $\Sigma M_{S;n}^{p,m}$. The following lemma states the property of the state bundle $\Sigma M_{S;n}^{p,m}$.

Lemma 6.1 *(Hazewinkel [30]) $\Sigma M_{S;n}^{p,m}$ has the structure of a real analytic n-vector bundle,* *with projection*

$$\begin{array}{rrcl} \pi: & \Sigma M_{S;n}^{p,m} & \to & SM_n^{p,m}/\sim \\ & [(A,B,C,D),x] & \mapsto & [(A,B,C,D)] \end{array}$$

The state bundles associated with the other classes of systems are defined in a similar manner and they also have the structure of n-vector bundles.

As an example of bundle diffeomorphism between state bundles, we shall define the bundle diffeomorphism between the state bundle $\Sigma S_{S;n}^{p,m}$ associated with stable systems and the state bundle $\Sigma B_{S;n}^{p,m}$ associated with bounded real systems. We shall make use of the diffeomorphism ψ between stable state space systems and bounded real state space systems as defined in definition 5.2. The map ψ has the following special property.

Lemma 6.2 *Let (A_1, B_1, C_1, D_1),(A_2, B_2, C_2, D_2) be two state space quadruples in $SS_n^{p,m}$, $(A_{1*}, B_{1*}, C_{1*}, D_{1*})$, $(A_{2*}, B_{2*}, C_{2*}, D_{2*})$ be two state space quadruples in $SB_n^{p,m}$ and ψ be the function defined in definition 5.2. If $(A_{1*}, B_{1*}, C_{1*}, D_{1*}) = \psi((A_1, B_1, C_1, D_1))$ and $(A_{2*}, B_{2*}, C_{2*}, D_{2*}) = \psi((A_2, B_2, C_2, D_2))$, then $(A_1, B_1, C_1, D_1; x_1) \sim (A_2, B_2, C_2, D_2; x_2)$ if and only if $(A_{1*}, B_{1*}, C_{1*}, D_{1*}; x_1) \sim (A_{2*}, B_{2*}, C_{2*}, D_{2*}; x_2)$.*

Hence the map ψ_S in the following definition is well defined.

Definition 6.1 *Define the map*

$$\begin{array}{rlcl} \psi_S: & \Sigma S_{S;n}^{p,m} & \to & \Sigma B_{S;n}^{p,m} \\ & [(A,B,C,D;x)] & \mapsto & [(A_*,B_*,C_*,D_*);x)] \\ & & & \text{where } (A_*,B_*,C_*,D_*) = \psi((A,B,C,D)) \end{array}$$

Since the map ψ is a real analytic diffeomorphism, the map ψ_S will be a real analytic bundle diffeomorphism; in other words the state bundles $\Sigma S_{S;n}^{p,m}$ and $\Sigma B_{S;n}^{p,m}$ are real analytic diffeomorphic to each other. It can be shown that the various diffeomorphisms given in chapter 4 also have the property stated in lemma 6.2, thus we can make use of them to define bundle diffeomorphisms, in a way analogous to definition 6.1, between the state bundles associated with various classes of linear systems. We shall now summarise the various bundle diffeomorphisms in the following theorems.

Theorem 6.1 *The following state bundles are real analytic diffeomorphic to each other: $\Sigma M_{S;n}^{p,m}$, $\Sigma S_{S;n}^{p,m}$, $\Sigma B_{S;n}^{p,m}$, $D\Sigma M_{S;n}^{p,m}$, $D\Sigma S_{S;n}^{p,m}$ and $D\Sigma B_{S;n}^{p,m}$.*

Theorem 6.2 *The following state bundles are real analytic diffeomorphic to each other: $\Sigma M_{S;n}^{m,m}$, $\Sigma S_{S;n}^{m,m}$, $\Sigma B_{S;n}^{m,m}$, $\Sigma P_{S;n}^{m}$, $D\Sigma M_{S;n}^{m,m}$, $D\Sigma S_{S;n}^{m,m}$, $D\Sigma B_{S;n}^{m,m}$ and $D\Sigma P_{S;n}^{m}$.*

Theorem 6.3 *The following state bundles are real analytic diffeomorphic to each other: $\Sigma MI_{S;n}^{m}$, $\Sigma SM_{S;n}^{m}$, $D\Sigma MI_{S;n}^{m}$ and $D\Sigma SM_{S;n}^{m}$.*

Theorem 6.4 *The state bundles associated with the manifolds $DRat(n;r)$, $DRat_S(n;r)$ and $DRat_{SM}(n;r)$ are real analytic diffeomorphic to each other.*

6.2 Principal fibre bundle diffeomorphisms

In this section, we shall prove the bundle diffeomorphisms between the various principal fibre bundles associated with the classes of linear systems defined in section 3.1. We shall start with the following lemma on the $Gl_n(\mathbb{R})$ principal fibre bundle associated with $\Sigma M_n^{p,m}$.

Lemma 6.3 *(Hazewinkel [30]) Define the following projection map*

$$\begin{array}{rccc} \beta: & SM_n^{p,m} & \to & \Sigma M_n^{p,m} \\ & (A,B,C,D) & \mapsto & \sigma \end{array}$$

where $\sigma \in \Sigma M_n^{p,m}$ is the system with transfer function $C(sI-A)^{-1}B+D$. Also define the right action of $Gl_n(\mathbb{R})$ on $SM_n^{p,m}$ as

$$\begin{array}{ccc} SM_n^{p,m} \times Gl_n(\mathbb{R}) & \to & SM_n^{p,m} \\ ((A,B,C,D),T) & \mapsto & (A,B,C,D)\circ T \end{array}$$

where $(A,B,C,D)\circ T = (T^{-1}AT, T^{-1}B, CT, D)$. The triple $(SM_n^{p,m}, \beta, \circ)$ is a real analytic principal fibre bundle [66] over $\Sigma M_n^{p,m}$ with group $Gl_n(\mathbb{R})$.

It can be shown that the real analytic $Gl_n(\mathbb{R})$ principal fibre bundles associated with the other classes of linear systems are given in a similar manner.

Since the bundle diffeomorphisms between the various principal fibre bundles can be defined in a similar way, we shall take the bundle diffeomorphism between the the principal fibre bundles associated with $\Sigma S_n^{p,m}$ and $\Sigma B_n^{p,m}$ as an example. Recall the map $\psi: SS_n^{p,m} \to SB_n^{p,m}$ defined in definition 5.2, since it has the property stated in lemma 6.2, it defines the bundle diffeomorphism between the principal fibre bundles associated with $\Sigma S_n^{p,m}$ and $\Sigma B_n^{p,m}$. Note that the fibre bundle diffeomorphisms between the other principal fibre bundles are given in an analogous way. We shall summarise the results in the following theorems.

Theorem 6.5 *The principal fibre bundles associated with the following classes of linear systems are real analytic diffeomorphic to each other: $\Sigma M_n^{p,m}$, $\Sigma S_n^{p,m}$, $\Sigma B_n^{p,m}$, $D\Sigma M_n^{p,m}$, $D\Sigma S_n^{p,m}$ and $D\Sigma B_n^{p,m}$*

Theorem 6.6 *The principal fibre bundles associated with the following classes of linear systems are real analytic diffeomorphic to each other: $\Sigma M_n^{m,m}$, $\Sigma S_n^{m,m}$, $\Sigma B_n^{m,m}$, ΣP_n^{m}, $D\Sigma M_n^{m,m}$, $D\Sigma S_n^{m,m}$, $D\Sigma B_n^{m,m}$ and $D\Sigma P_n^{m}$.*

Theorem 6.7 *The principal fibre bundles associated with the following classes of linear systems are real analytic diffeomorphic to each other: ΣMI_n^m, ΣSM_n^m, $D\Sigma MI_n^m$ and $D\Sigma SM_n^m$.*

Theorem 6.8 *The principal fibre bundles associated with $DRat(n;r)$, $DRat_S(n;r)$ and $DRat_{SM}(n;r)$ are real analytic diffeomorphic to each other.*

Chapter 7

Applications

We have mentioned earlier that the behaviour of a system identification algorithm on a space of systems is related to the geometry of that particular space of systems. In the earlier chapters of this thesis, we have proved that a number of classes of linear systems are diffeomorphic to each other. In section 7.1, we shall examine the implications of these results on geometry of spaces of systems. In sections 7.2 and 7.3, we shall respectively discuss how these diffeomorphisms can be used to construct overlapping parametrizations and Riemannian gradient algorithms on various classes of systems.

7.1 Geometric properties of spaces of linear systems

In section 3.2, it is stated that the classes of systems $\Sigma M_n^{p,m}$, $\Sigma S_n^{p,m}$, $\Sigma B_n^{p,m}$, ΣP_n^m, ΣMI_n^m, ΣSM_n^m, $D\Sigma M_n^{p,m}$, $D\Sigma S_n^{p,m}$, $D\Sigma B_n^{p,m}$, $D\Sigma P_n^m$, $D\Sigma MI_n^m$ and $D\Sigma SM_n^m$ have the structure of real analytic manifolds. When two manifolds are diffeomorphic to each other, it implies that they have identical geometric properties. Hence, if the geometric properties of one of these manifolds are known, we can immediately conclude that the other manifold also possesses the same geometric properties. In this section, we shall give three examples on how we can use this fact to deduce the geometric properties of a number of classes of linear systems. These three examples will be on compactness, on the number of pathwise connected components and on the (non)-existence of continuous canonical forms.

7.1.1 Compactness

We shall look at the compactness of various classes of linear systems in this section. The following lemma addresses compactness of the set $\Sigma M_n^{p,m}$.

Lemma 7.1 *(Hazewinkel [30]) The set $\Sigma M_n^{p,m}$ is not compact.*

This lemma has important implication on the convergence property of system identification algorithms as it now becomes clear that it is possible for a system identification algorithm on $\Sigma M_n^{p,m}$ to converge to points outside $\Sigma M_n^{p,m}$. We shall not look into this degeneration phenomenon any further here. The reader can refer to the following references: [29, 31]. Since the manifold $\Sigma M_n^{p,m}$ is non-compact, it implies that the classes of linear systems which are diffeomorphic to $\Sigma M_n^{p,m}$ are also non-compact. The result is stated in the following lemma.

Lemma 7.2 *The following manifolds are all non-compact manifolds:* $\Sigma S_n^{p,m}$, $\Sigma B_n^{p,m}$, ΣP_n^m, ΣMI_n^m, ΣSM_n^m, $D\Sigma M_n^{p,m}$, $D\Sigma S_n^{p,m}$, $D\Sigma B_n^{p,m}$, $D\Sigma P_n^m$, $D\Sigma MI_n^m$ *and* $D\Sigma SM_n^m$.

Note that the comment on the degeneration phenomenon also applies to the above classes of linear systems.

7.1.2 Number of pathwise connected components

Results on the number of pathwise connected components in the set of fixed order systems were first obtained by Brockett [8] for the SISO case and by Glover [19] for the multi-input multi-output (MIMO) case (see lemmas 3.1 and 3.2). We shall discuss the implications of these results on the behaviour of system identification algorithm and we shall start with the SISO case. The manifold $\Sigma M_n^{1,1}$ consists of $(n+1)$ pathwise connected components and the boundaries of these components are made up of systems of lower orders. If the system identification procedure is operating near the boundaries of these components, the numerical conditioning of the estimator is poor due to its proximity to possible pole-zero cancellations. However, in the MIMO case, the manifold $\Sigma M_n^{p,m}$ is pathwise connected. This is due to the fact that the systems of order less than n form a space of dimension less than or equal to $n(m+p)+mp-2$ while the dimension of $\Sigma M_n^{p,m}$ is $n(m+p)+mp$. Although numerical ill-conditioning of the estimator also occurs while it is operating near the proximity of the boundary of the manifold, the probability for this to occur is less.

We shall now use our results on the diffeomorphisms between classes of systems to obtain the number of pathwise connected components on the other manifolds of systems. By using the results on the number of pathwise connected components on the manifold of fixed order systems, and the fact that fixed order, stable, bounded real and positive real systems are all diffeomorphic to each other (theorems 4.1 and 4.2), we have the following lemma:

Lemma 7.3 *The manifolds* $\Sigma M_n^{p,m}$, $\Sigma S_n^{p,m}$, $\Sigma B_n^{p,m}$, $D\Sigma M_n^{p,m}$, $D\Sigma S_n^{p,m}$ *and* $D\Sigma B_n^{p,m}$ *are pathwise connected for* $\max(p,m)>1$. *When* $p=m=1$, *they each have* $(n+1)$ *pathwise connected components. The manifolds* ΣP_n^m *and* $D\Sigma P_n^m$ *each have* $(n+1)$ *pathwise connected components when* $m=1$ *but are pathwise connected otherwise.*

By using the geometric results on fixed order systems, theorem 4.3 and the fact that the set $\{D \in \mathbb{R}^{m\times m} : D \text{ is invertible}\}$ has two pathwise connected components, we obtain the following geometric results:

Lemma 7.4 *The manifolds* ΣMI_n^m, ΣSM_n^m, $D\Sigma MI_n^m$ *and* $D\Sigma SM_n^m$ *each have* 2 *pathwise connected components for* $m>1$ *and* $2(n+1)$ *pathwise connected components for* $m=1$.

Note that the results on the number of pathwise connected components on the class of stable systems were obtained independently by Hanzon and Ober (see lemma 3.3). The results on other classes of systems, except that on invertible systems, were obtained by Ober [52] using balanced parametrizations of these classes of systems.

7.1.3 (Non)-existence of continuous canonical forms

Note that the last two geometric properties which we have discussed are both topological properties. So the geometric properties given in the last two sections could have been deduced from the fact that these classes of linear systems are homeomorphic to each other. However, in order to prove the results in this section on the (non)-existence of canonical forms, we shall require these spaces of systems to be diffeomorphic with each other.

It is a well known fact that, for many types of system representations — for example, state space models and autoregressive moving-average (ARMA) models — there exist an infinite number of input-output equivalent representations for each system in $\Sigma M_n^{p,m}$. However, when we perform system identification on the model set $\Sigma M_n^{p,m}$, it is desirable that each system in $\Sigma M_n^{p,m}$ has a unique representation. A solution to this problem is to use canonical forms. Also, it is desirable that these canonical forms are continuous so that the objective function is a continuous function of the parameters.

The question on whether continuous canonical forms exist on the set $\Sigma M_n^{p,m}$ is answered in the following lemma.

Lemma 7.5 *(Hazewinkel-Kalman [32], Byrnes-Hurt [10]) There exists a continuous canonical form on the set $\Sigma M_n^{p,m}$ if and only if $min(m,p) = 1$.*

Basically this result states that continuous canonical forms exist only on the spaces of single-input or single-output systems. We saw in section 6.2 that the triple $(SM_n^{p,m}, \beta, \circ)$ is a principal fibre bundle over $\Sigma M_n^{p,m}$. The existence of continuous canonical forms on a class of systems is related to whether the principal fibre bundle associated with that particular class of systems has a global continuous section or not. The section of the principal fibre bundle associated with $\Sigma M_n^{p,m}$ is simply the map $s : \Sigma M_n^{p,m} \to SM_n^{p,m}$. A global continuous section exists on a principal fibre bundle if and only if the bundle is trivial [66]. The triviality of principal fibre bundles is preserved by bundle diffeomorphisms, so by using the known results on the (non)-existence of continuous canonical forms on $\Sigma M_n^{p,m}$ and the principal fibre bundle diffeomorphisms stated in theorems 6.5 and 6.6, we have the following two lemmas.

Lemma 7.6 *Continuous canonical forms exist on these classes of linear systems: $\Sigma M_n^{p,m}$, $\Sigma S_n^{p,m}$, $\Sigma B_n^{p,m}$, $D\Sigma M_n^{p,m}$, $D\Sigma S_n^{p,m}$ and $D\Sigma B_n^{p,m}$, if and only if $min(m,p) = 1$.*

Lemma 7.7 *Continuous canonical forms exist on these classes of linear systems: ΣP_n^m, ΣMI_n^m, ΣSM_n^m, $D\Sigma P_n^m$, $D\Sigma MI_n^m$ and $D\Sigma SM_n^m$, if and only if $m = 1$.*

Note that the result on the (non)-existence of continuous canonical forms on $\Sigma S_n^{p,m}$ and $D\Sigma S_n^{p,m}$ was stated in [26].

7.2 Construction of overlapping parametrizations

In this section, we shall discuss how our results on diffeomorphisms can be used to construct overlapping parametrizations on spaces of linear systems. As an example, we shall demonstrate how an overlapping parametrization can be constructed on $\Sigma B_n^{p,m}$ and how the overlapping parametrization so constructed can be used in system identification.

We have discussed in section 7.1.3 that the problem of parameter identifiability can be overcome by the use of canonical forms. However, there are two major problems with the use of canonical forms. Firstly, continuous canonical forms do not exist on the spaces of multi-input or multi-output systems, as we have seen in the last section. Secondly, the parameter estimator based on these canonical forms becomes numerically ill-conditioned when the optimization algorithm is operating at the proximity of the 'boundary' of the manifold. In order to overcome these two problems, it was suggested in [21] that overlapping parametrizations should be used instead. An overlapping parametrization for, say $\Sigma M_n^{p,m}$, may be defined as follows:

Let U_i for $i = 1, ..., l$ be subsets of $\mathbb{R}^N$ and ρ_i be injective maps from U_i to $\Sigma M_n^{p,m}$. We shall call (U_i, ρ_i) an overlapping parametrization on $\Sigma M_n^{p,m}$ if the image spaces of the maps ρ_i satisfy the following 2 properties.

1. For every $i \in \{1, \ldots, l\}$, there exist at least one $j \in \{1, \ldots, l\}$ such that $\rho_i(U_i) \cap \rho_j(U_j) \neq \phi$, and
2. $\bigcup_i \rho_i(U_i) = \Sigma M_n^{p,m}$.

Note that the basic definition of overlapping parametrizations only requires the injectivity of the maps ρ_i but it is possible to impose restrictions on these maps so that they have some desirable properties like continuity or differentiability.

Overlapping parametrizations for the manifolds $\Sigma M_n^{p,m}$ [68] and $\Sigma S_n^{1,1}$ [28] have been derived. Both of these overlapping parametrizations were derived by modifying the canonical forms on these two manifolds. The one on $\Sigma M_n^{p,m}$ was obtained from the observable canonical form while the one on $\Sigma S_n^{1,1}$ came from balanced canonical form for stable systems. For a discussion on using these overlapping parametrizations in system identification, the reader can refer to the following references: [68, 25, 58, 57, 59].

For the rest of this section, we shall give an example of how our results on diffeomorphisms between classes of systems can be used to derive overlapping parametrizations for other classes of systems. We shall assume that we already have obtained an overlapping parametrization (U_i, ρ_i) for the state space systems in $\Sigma S_n^{p,m}$ and we shall derive an overlapping parametrization for $\Sigma B_n^{p,m}$ using the diffeomorphism between $\Sigma S_n^{p,m}$ and $\Sigma B_n^{p,m}$.

Let (U_i, ρ_i) be an overlapping parametrization for $\Sigma S_n^{p,m}$ where the maps ρ_i for $i = 1, ..., l$ are defined as follows

$$\begin{array}{rccc} \rho_i: & U_i & \rightarrow & SS_n^{p,m} \\ & \theta & \mapsto & (A(\theta), B(\theta), C(\theta), D(\theta)) \end{array}$$

We shall now derive an overlapping parametrization on $\Sigma B_n^{p,m}$ given the diffeomorphism between $\Sigma S_n^{p,m}$ and $\Sigma B_n^{p,m}$. The following diffeomorphism is taken from definition 5.2.

$$\begin{array}{rccc} \psi: & SS_n^{p,m} & \rightarrow & SB_n^{p,m} \\ & (A, B, C, D) & \mapsto & (A_*, B_*, C_*, D_*) \end{array}$$

where

$$B_* = BS^{-1/2} \tag{7.1}$$

$$\begin{aligned}
D_* &= DS^{-1/2} && (7.2)\\
A_* &= A - BS^{-1}(B^TP + D^TC) = A - B_*(B_*^TP + D_*^TC) && (7.3)\\
C_* &= C - DS^{-1}(B^TP + D^TC) = C - D_*(B_*^TP + D_*^TC) && (7.4)\\
S &= I + D^TD && (7.5)
\end{aligned}$$

and P is the stabilizing solution of the GCARE

$$A^TP + PA - (PB + C^TD)(I + D^TD)^{-1}(PB + C^TD)^T + C^TC = 0 \tag{7.6}$$

An overlapping parametrization for $\Sigma B_n^{p,m}$ is given by

$$\begin{aligned}
\psi \circ \rho_i : \;\; U_i &\rightarrow \quad SB_n^{p,m}\\
\theta &\mapsto (A_*(\theta), B_*(\theta), C_*(\theta), D_*(\theta))
\end{aligned}$$

Note that since ψ is real analytic, the map $\psi \circ \rho_i$ will be C^p (C^ω) if ρ_i is C^p (C^ω). Note also that since no closed form solution exists for the GCARE, the above overlapping parametrization can only be used implicitly. We shall discuss this point further.

Assume that we shall use this overlapping parametrization for bounded real systems together with a gradient algorithm of the kind

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \mu g(\hat{\theta}_t) \tag{7.7}$$

where $g(\theta)$ is the gradient of the objective function $V(\theta)$ and μ is the step size. There are two questions which we need to address here, they are

1. Given the state space system (A_*, B_*, C_*, D_*), how to obtain the parameter vector θ?

2. How to find the gradient of the objective function?

The answer to the first question is rather straightforward as the map ψ is invertible, we shall first find $(A, B, C, D) = \psi^{-1}(A_*, B_*, C_*, D_*)$ and the parameter vector θ can then be found from (A, B, C, D).

In order to find the gradient of the objective function (see also the discussion in section 12.4), we note that the function $V(\theta)$ is basically a function of the state space matrices $(A_*(\theta), B_*(\theta), C_*(\theta), D_*(\theta))$ and the data. If the function $V(\theta)$ is an analytic function of these state space matrices, then its gradient $g(\theta)$ can be expressed explicitly in terms of the derivatives

$$\frac{\partial A_*(\theta)}{\partial \theta_i}, \frac{\partial B_*(\theta)}{\partial \theta_i}, \frac{\partial C_*(\theta)}{\partial \theta_i} \text{ and } \frac{\partial D_*(\theta)}{\partial \theta_i}$$

where θ_i is any element of the parameter vector θ. These derivatives can in turn be expressed in terms of the following derivatives

$$\frac{\partial A(\theta)}{\partial \theta_i}, \frac{\partial B(\theta)}{\partial \theta_i}, \frac{\partial C(\theta)}{\partial \theta_i} \text{ and } \frac{\partial D(\theta)}{\partial \theta_i}$$

which are explicit functions of θ. We shall now give the formulae for the derivatives of $(A_*(\theta), B_*(\theta), C_*(\theta), D_*(\theta))$. From this point onwards, in order to simplify our notation, we shall drop the parameter θ behind the state space matrices. From equations 7.1 to 7.4, we have

$$\frac{\partial B_*}{\partial \theta_i} = \frac{\partial B}{\partial \theta_i} S^{-1/2} + B\frac{\partial (S^{-1/2})}{\partial \theta_i} \tag{7.8}$$

$$\frac{\partial D_*}{\partial \theta_i} = \frac{\partial D}{\partial \theta_i} S^{-1/2} + D\frac{\partial (S^{-1/2})}{\partial \theta_i} \tag{7.9}$$

$$\begin{aligned}\frac{\partial A_*}{\partial \theta_i} &= \frac{\partial A}{\partial \theta_i} - \frac{\partial B_*}{\partial \theta_i}(B_*^T P + D_*^T C) - \\ &\quad B_*[(\frac{\partial B_*}{\partial \theta_i})^T P + B_*^T \frac{\partial P}{\partial \theta_i} + (\frac{\partial D_*}{\partial \theta_i})^T C + D_*^T \frac{\partial C}{\partial \theta_i}]\end{aligned} \tag{7.10}$$

$$\begin{aligned}\frac{\partial C_*}{\partial \theta_i} &= \frac{\partial C}{\partial \theta_i} - \frac{\partial D_*}{\partial \theta_i}(B_*^T P + D_*^T C) - \\ &\quad D_*[(\frac{\partial B_*}{\partial \theta_i})^T P + B_*^T \frac{\partial P}{\partial \theta_i} + (\frac{\partial D_*}{\partial \theta_i})^T C + D_*^T \frac{\partial C}{\partial \theta_i}]\end{aligned} \tag{7.11}$$

The derivative of $S^{-1/2}$ can be obtained by noting that

$$\begin{aligned} S^{-1/2} \times S^{-1/2} &= (I + D^T D)^{-1} \\ \Rightarrow \quad \frac{\partial (S^{-1/2})}{\partial \theta_i} S^{-1/2} + S^{-1/2}\frac{\partial (S^{-1/2})}{\partial \theta_i} &= -S^{-1}[(\frac{\partial D}{\partial \theta_i})^T D + D^T \frac{\partial D}{\partial \theta_i}] S^{-1} \end{aligned} \tag{7.12}$$

Hence the derivative of $S^{-1/2}$ can be obtained by solving the above matrix Lyapunov equation.

In order to obtain the derivative of P, we need to use the GCARE. By taking the derivative of equation 7.6 with respect to θ_i and after simplification, it can be shown that $\frac{\partial P}{\partial \theta_i}$ is the solution to the following matrix Lyapunov equation.

$$\begin{aligned} &\frac{\partial P}{\partial \theta_i} A_* + A_*^T \frac{\partial P}{\partial \theta_i} \\ = \; &-P\frac{\partial A}{\partial \theta_i} - (\frac{\partial A}{\partial \theta_i})^T P - (\frac{\partial C}{\partial \theta_i})^T C_* - C_*^T \frac{\partial C}{\partial \theta_i} + (P\frac{\partial B}{\partial \theta_i} + C_*^T \frac{\partial D}{\partial \theta_i}) S^{-1} (PB + C^T D)^T + \\ &(PB + C^T D) S^{-1} (P\frac{\partial B}{\partial \theta_i} + C_*^T \frac{\partial D}{\partial \theta_i})^T \end{aligned} \tag{7.13}$$

We see in the above example that we need to solve two Lyapunov equations in order to obtain the derivatives of the state space matrices. We would like to remark that if we construct overlapping parametrizations on other classes of systems using the diffeomorphisms given in chapter 5, again these overlapping parametrizations can only be used implicitly. In order to calculate the gradient of the objective function, we need to solve some linear matrix equations (Lyapunov equation in the above example) or some linear scalar equations.

7.3 System identification based on Riemannian Geometry

We stated in section 3.2 that the classes of systems $\Sigma M_n^{p,m}$, $\Sigma S_n^{p,m}$, $\Sigma B_n^{p,m}$, ΣP_n^m, ΣMI_n^m, ΣSM_n^m, $D\Sigma M_n^{p,m}$, $D\Sigma S_n^{p,m}$, $D\Sigma B_n^{p,m}$, $D\Sigma P_n^m$, $D\Sigma MI_n^m$ and $D\Sigma SM_n^m$ have the structure of real analytic manifolds. It is a standard result in differential geometry that any smooth differentiable manifold can be endowed with a Riemannian metric [7]. Hence it is possible to define Riemannian metrics on all of these classes of systems. The Riemannian metrics so defined will give us a measure of the distance between two systems on these manifolds. A recursive system identification algorithm on the manifold $\Sigma S_n^{p,m}$ based on Riemannian geometry was proposed in [25]. The basic idea is to replace the following gradient algorithm

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \mu g(\hat{\theta}_t) \tag{7.14}$$

where $\hat{\theta}_t$ is the estimate at time t, $g(\theta)$ is the gradient of the objective function $V(\theta)$, and μ is the step size, by the following Riemannian gradient algorithm

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \mu R(\hat{\theta}_t)^{-1} g(\hat{\theta}_t) \tag{7.15}$$

where $R(\theta)$ is the Riemannian metric tensor.

This algorithm has the advantage that when it is used for on-line system identification, the path taken by the optimization algorithm to reach the minimum is almost independent of the chosen parametrization. In this section, we shall discuss how Riemannian gradient algorithms can be constructed on other classes of linear systems given our results on diffeomorphisms. We shall begin our discussion by giving a review of the Riemannian gradient algorithm.

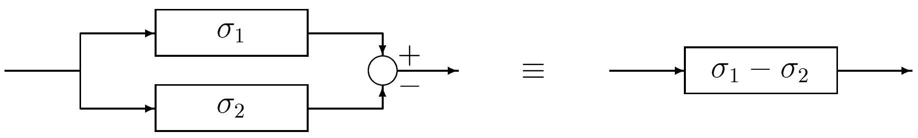
The following discussion is based on the algorithm proposed in [25]. Let θ denote the parameter vector of a parametrization for $\Sigma S_n^{p,m}$ and $\sigma(\theta)$ be its corresponding system in $\Sigma S_n^{p,m}$. Consider the following curve in the space $\Sigma S_n^{p,m}$

$$\begin{aligned} [0,1] &\to \Sigma S_n^{p,m} \\ x &\mapsto \sigma(\theta(x)) \end{aligned}$$

The derivative of σ is defined as follows

$$\dot{\sigma}(\theta(x_0)) = \lim_{x \to x_0} \frac{\sigma(\theta(x)) - \sigma(\theta(x_0))}{x - x_0} \tag{7.16}$$

where the dot above σ denotes derivative with respect to x. Note that the subtraction between two systems σ_1 and σ_2 are defined as in the diagram below.



If the state space realization of $\sigma(\theta(x))$ is denoted by $(A(x), B(x), C(x), D(x))$ (we shall write $A(x)$ instead of $A(\theta(x))$ for brevity), it can be shown that

$$\sigma(\theta(x)) - \sigma(\theta(x_0))$$
$$= \left(\begin{bmatrix} A(x) & 0 \\ 0 & A(x_0) \end{bmatrix}, \begin{bmatrix} B(x) \\ B(x_0) \end{bmatrix}, \begin{bmatrix} C(x) & -C(x_0) \end{bmatrix}, D(x) - D(x_0) \right) \tag{7.17}$$

By performing the state space coordinate transformation

$$x_{\text{new}} = \begin{bmatrix} \frac{1}{x-x_0}I & 0 \\ \frac{1}{x-x_0}I & I \end{bmatrix} x_{\text{old}} \tag{7.18}$$

to the system given in equation 7.17 and after taking the limit $x \to x_0$, we have

$$\dot{\sigma}(\theta(x_0)) = \left(\begin{bmatrix} A(x_0) & 0 \\ \dot{A}(x_0) & A(x_0) \end{bmatrix}, \begin{bmatrix} B(x_0) \\ \dot{B}(x_0) \end{bmatrix}, \begin{bmatrix} \dot{C}(x_0) & C(x_0) \end{bmatrix}, \dot{D}(x_0) \right) \tag{7.19}$$

By defining an inner product on the tangent space of $\Sigma S_n^{p,m}$ based on the $\mathcal{H}_2$ norm, it can be shown that the entity $\|\dot{\sigma}(\theta)\|_2^2$ is a quadratic function of $\frac{d\theta}{dx}$ and there exists a positive definite matrix $R(\theta)$ such that

$$\|\dot{\sigma}(\theta)\|_2^2 = (\frac{d\theta}{dx})^T R(\theta) \frac{d\theta}{dx} \tag{7.20}$$

The matrix R is known as the Riemannian metric tensor. Given a particular class of systems, there are many different ways of defining a metric on it, see [59, 58] for a more detailed discussion on this topic.

Given the Riemannian metric tensor $R_S(\theta)$ on $\Sigma S_n^{p,m}$ and the diffeomorphism ψ between $\Sigma S_n^{p,m}$ and $\Sigma B_n^{p,m}$ as given in definition 5.2, how can we construct a Riemannian gradient algorithm on $\Sigma B_n^{p,m}$? We have discussed in section 7.2 that an explicit parametrization on $\Sigma B_n^{p,m}$ cannot be obtained, so we have to be content with using the parametrization on $\Sigma S_n^{p,m}$ for doing system identification on $\Sigma B_n^{p,m}$. There are a few different ways to define a metric on the space $\Sigma B_n^{p,m}$ and each one of these definitions results in a different algorithm. A way to define a metric is to equate the 'distance' between corresponding systems on the manifolds $\Sigma S_n^{p,m}$ and $\Sigma B_n^{p,m}$ i.e. by assuming that

$$< \dot{\psi}(\sigma), \dot{\psi}(\sigma) >_{\Sigma B_n^{p,m}} = < \dot{\sigma}, \dot{\sigma} >_{\Sigma S_n^{p,m}} \tag{7.21}$$

where $< , >_X$ is the inner product defined on the tangent space of the manifold X. By the definition of $R_S(\theta)$, we have

$$< \dot{\sigma}, \dot{\sigma} >_{\Sigma S_n^{p,m}} = (\frac{d\theta}{dx})^T R_S(\theta) \frac{d\theta}{dx} \tag{7.22}$$

Hence, by comparing equations 7.21 and 7.22, the Riemannian metric on $\Sigma B_n^{p,m}$ is in fact $R_S(\theta)$. The Riemannian gradient algorithm on $\Sigma B_n^{p,m}$ is given by

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \mu R_S(\hat{\theta}_t)^{-1} g(\hat{\theta}_t) \tag{7.23}$$

where $R(\theta)$ is the Riemannian metric tensor on $\Sigma S_n^{p,m}$ and the gradient $g(\hat{\theta}_t)$ can be calculated as suggested in section 7.2.

Since bounded real systems are also stable systems, an alternative way to define a metric on $\Sigma B_n^{p,m}$ is to embed $\Sigma B_n^{p,m}$ in $\Sigma S_n^{p,m}$, and we shall define the Riemannian metric on $\Sigma B_n^{p,m}$ by

$$< \dot{\psi}(\sigma), \dot{\psi}(\sigma) >_{\Sigma B_n^{p,m}} = \|\dot{\psi}(\sigma)\|_2^2 \tag{7.24}$$

Unfortunately, there does not exist a simple formula relating $R_B(\theta)$ and $R_S(\theta)$, so we have to resort to calculating $R_B(\theta)$ using

$$\|\dot{\psi}(\sigma)\|_2^2 \;=\; (\frac{d\theta}{dx})^T R_B(\theta)\frac{d\theta}{dx} \tag{7.25}$$

where

$$\dot{\psi}(\sigma) \;=\; \left(\begin{bmatrix} A_*(x_0) & 0 \\ \dot{A}_*(x_0) & A_*(x_0) \end{bmatrix}, \begin{bmatrix} B_*(x_0) \\ \dot{B}_*(x_0) \end{bmatrix}, \begin{bmatrix} \dot{C}_*(x_0) & C_*(x_0) \end{bmatrix}, \dot{D}_*(x_0) \right) \tag{7.26}$$

where (A_*, B_*, C_*, D_*) are given by equations 7.1 to 7.4, and $(\dot{A}_*, \dot{B}_*, \dot{C}_*, \dot{D}_*)$ are given by equations 7.8 to 7.11 but with θ_i replaced by x.

The Reimannian gradient algorithm constructed on $\Sigma S_n^{p,m}$ was shown be to convergent in [25], hence the Riemannian gradient algorithms constructed on the other classes of systems using our results on diffeomorphism are guaranteed to converge.

As a final remark, the Riemannian gradient algorithm given above is almost independent of the parametrizations used. A coordinate free state space identification procedure on $\Sigma S_n^{p,m}$ was suggested in [23] by constructing a metric on the state bundle. Similar coordinate free algorithms can be constructed on other manifolds by using the state bundle diffeomorphisms stated in section 6.1.

Part II

System identification using balanced parametrizations

Chapter 8

Introduction

8.1 Overview

The goal of system identification is to find a model from experimental data. In control engineering, the models obtained are used for synthesis of controllers, for prediction and for simulation. Many different types of models are used in control-oriented applications; they can be parametric or non-parametric, linear or non-linear, time-invariant or time-varying, deterministic or stochastic, or of many other different natures. Although most of the processes which we try to model in control-oriented applications are very complex, our experience shows that many of them can actually be modelled rather accurately by using finite-dimensional linear time-invariant (FDLTI) systems of a sufficiently high order.

There are a number of different ways to represent FDLTI systems. Examples of these include transfer functions, autoregressive moving-average (ARMA) models and state space models. Although it is possible to transform between these different types of system representations, state space models are particularly popular, for two reasons: first, most of the modern controller design techniques such as linear quadratic Gaussian (LQG) and $\mathcal{H}_\infty$ control are based on the assumption that a state space model of the plant is available; second, state space models are superior to input-output models, such as transfer function models and ARMA models, in the sense that they are numerically better conditioned when used for calculating the input-output behaviour of the system.

In this part of the thesis, we shall propose an algorithm for the identification of stable FDLTI state space systems from input/output data. This algorithm consists of two steps. In the first step, an initial state space estimate is calculated from the input/output data. This state space model is then supplied to a parameter optimization algorithm to find an optimal model. We shall use either model reduction based system identification methods or subspace methods to find a state space model in the first step. These two classes of algorithms are chosen because they are able to give sufficiently good initial estimates. The novelty of our identification scheme lies in the use of balanced parametrizations for stable systems in parameter estimation. There are a few reasons for this particular choice. First of all, these parametrizations are state space canonical forms, so there is no problem with parametric identifiability. Also, the balanced canonical forms guarantee stability and minimality of the state space models when some simple equality and inequality constraints on the parameters are satisfied. Note that this particular property is not found in other types of state space canonical forms such as the controllable and observable canonical forms. For these canonical forms, complicated non-linear inequality constraints are generally required to guarantee the stability and minimality of the state space models. Balanced state space realizations are also known to have good numerical properties when they are used for simulation of input-output behaviour. It was shown in [47] that out of all the equivalent state space realizations of a FDLTI system, the balanced form gives the least round-off error under finite wordlength implementation. Finally, for the purpose of parameter estimation, it was conjectured in [41] that the parameter estimator using balanced canonical forms has optimal conditioning in the sense that the distance from singularity of both the controllability and observability Gramians is greater than that given by other canonical forms.

8.2 Outline

The system identification algorithm outlined above will be the subject of discussion in this part of the thesis. The organization of this part of the thesis is as follows.

Chapter 9 Problem description

In this chapter we shall give a brief overview of the problem of system identification. In particular, we shall look at three classes of methods for obtaining state space models from input/output data. These three classes of methods are parameter estimation, system identification via model reduction and subspace methods. We shall then give an outline of the system identification algorithm which was mentioned in section 8.1.

Chapter 10 Balanced realizations and parametrizations

This chapter contains the background materials on both balanced realizations and balanced parametrizations for stable FDLTI systems. We shall present results on both continuous time and discrete time balanced realizations, and their model reduction properties. The balanced parametrizations derived by Ober will then be presented. A particular component of these balanced parametrizations is the orthogonal matrix group. We shall discuss how this matrix group can be parametrized by using rotation matrices. The result of parametrizing the orthogonal matrix group will be used to derive a local coordinate system for the set of all unit vectors — which is also a component of balanced parametrizations.

Chapter 11 Finding an initial estimate

In this chapter, we shall discuss three different methods to find an initial state space model from input/output data. These three methods are Kung's algorithm, a model reduction method due to Al-Saggaf, and N4SID (which stands for Numerical Algorithm for Subspace State Space System Identification). The first two methods are related to balanced truncation of stable systems. The last method is a subspace identification algorithm. We shall also give a comparison of these three methods.

Chapter 12 Parameter estimation

In this chapter, we shall present an algorithm for parameter optimization using balanced parametrizations for stable systems. In order to improve the efficiency of the algorithm, we shall supply to the algorithm an explicit formula for the gradient of the objective function. We shall discuss how the gradient of the objective function can be calculated, and state explicit formulae for the derivatives of multi-input multi-output (MIMO) balanced state space matrices. A few examples of gradient calculations will also be given.

Chapter 13 Case studies

In this chapter we shall present two case studies on applying our proposed system identification algorithm to two sets of industrial data. The first set of data comes from a very detailed simulation model of a fractional distillation column. The second set of the data comes from an industrial process which dries the raw material by passing hot gas over it.

Chapter 9

Problem description

The aim of this chapter is to provide a brief overview of the subject of system identification of state space models. We shall begin the overview in section 9.1 by discussing the general system identification procedure. One of the steps of this procedure is called model estimation. This is the step in which a model is calculated based on input/output measurements. There are three main approaches to the problem of calculating a state space model based on input/output data. These three classes of methods are known as parameter estimation, system identification via model reduction and subspace methods. They will be the subjects of discussion in sections 9.2 (parameter estimation), 9.3 (system identification via model reduction) and 9.4 (subspace methods). We shall finish this chapter by proposing an algorithm for model estimation which combines all these three classes of methods together. This algorithm will be the subject of discussion in the later chapters of this part of the thesis.

9.1 System identification — general implementation

The problem of system identification can be stated as follows: based on the *a priori* knowledge and the *input/output measurements* of a dynamical process, find a *model* which *adequately* describes the dynamical behaviour of the process. Hence, system identification is different from the art of 'physical' modelling which relies entirely on the physical understanding, i.e. *a priori* knowledge, that we have on the process. In this part of the thesis, we shall be looking at what is commonly known as the 'black box' modelling technique and we shall assume *a priori* that

1. the division between the inputs and the outputs is known,
2. only the outputs are corrupted by noise, and
3. the process is stable.

Since the product of system identification is a model, so at this stage we need to decide on the type of model which we are going to use. This decision is very often dependent on how the model is going to be used, for example, whether it is for prediction, for simulation or for synthesis of controllers. For a survey on the use of different types of models, the reader may refer to [40, 64, 22]. In this part of the thesis, we have chosen to use FDLTI state space models for the reasons which have already been stated in section 8.1.

After the *a priori* assumptions have been made, the system identification procedure is now entirely relying on the input/output measurements we have on the process. We shall divide our system identification procedure into the following four steps.

1. Experiment design.
2. Data collection.
3. Model estimation.
4. Model validation.

The first step of the system identification procedure is to design an experiment during which input/output measurements will be collected. The major decision to make at this step is on the type of input to be used to excite the process. For some processes, especially those which are already in operation, we may not have a choice on the type of excitation to be used and we have to settle for the data provided by the plant. In the case in which we are allowed to conduct an experiment on the process, there are many different choices of excitation signals and each one of these is able to provide us with accurate information on certain aspects of the process. For example, if we are going to use a frequency response model and we would like to know the bandwidth of the system accurately, then sinusoidal excitation is a good choice. As another example, if we would like to obtain accurate information on the steady state gains of the process, then step response test is appropriate. We also need to make sure that the input test signals are persistently exciting so that sufficient information can be extracted from the experimental data later on. So, in order to decide on the type of excitation to be used, we first need to ask ourselves the following question: What kind of information on the process would we like to extract from the input/output measurements? We shall not discuss the topic on experiment design further; the reader may refer to [40, 64] and the references therein for a more complete treatment of this subject.

The step of model estimation deals with finding a model of the process based on the input/output measurements collected during the system identification experiment. Usually, a model is chosen from a set of admissible models, which we shall refer to as the model set, based upon some selection criteria. For the identification of state space models, there are three main approaches — parameter estimation, system identification via model reduction and subspace methods. We shall defer the discussion of these three classes of methods to the next few sections of this chapter.

At the step of model validation, we decide whether the model which we have obtained at the model estimation step is acceptable or not. A common method to validate a model is to divide the input/output data into two parts. The first part of the data will be used to calculate a model while the second part will be used for model validation. The model is accepted or rejected based on how well it is able to predict the second part of the process output data i.e. the validation data. Other methods of model validation include using statistical tests to check the whiteness of the prediction error or the correlation between the inputs and the prediction error. We shall not be looking into the topic of model validation in this thesis; the reader may refer to [40, 64] for a more complete treatment of this subject.

Note that these four steps — experiment design, data collection, model estimation and model validation — are not entirely independent of each other and it must also be emphasized that these steps form part of an iterative scheme. For example, if the model is found to be unacceptable at the model validation step, we may decide to re-design the experiment, or to re-run the experiment under some different conditions, or to select the model according to a different selection criterion. The above procedure may have to repeat several times before an acceptable model is finally obtained.

In this part of the dissertation, we shall focus our discussion on the the step of model estimation only. In particular, we shall develop a method for the identification of state space models from input/output measurements.

9.2 Parameter estimation

Parameter estimation is one of the most widely used methods in system identification. It includes methods such as linear least squares, instrumental variable (IV) and prediction error [40, 64]. Basically, the problem of parameter estimation can be formulated as follows: given the model set $\mathcal{M}$, and an objective function $V(M)$ which quantifies the mismatch between the model M and the data, find the model M_{opt} which is the solution to the optimization problem

$$\arg \min_{M \in \mathcal{M}} V(M)$$

The main characteristic of the parameter estimation methods is that they give an optimal model, so we are assured of the quality of the model. In order to solve the above optimization problem numerically, a parametrization for the model set $\mathcal{M}$ will be required. Thus, the three basic components of a parameter estimation scheme are a parametrization for the model set, an objective function and an optimization algorithm. We shall first discuss a few general issues related to these three basic components, and then we shall present a general procedure for parameter estimation at the end of this section.

In simple terms, a parametrization of a model set can be looked at as a way of assigning to each model in the model set a point in the Euclidean space $\mathbb{R}^N$. The main difficulty in parametrizing state space models for FDLTI systems is that of parameter identifiability, as every FDLTI system can be represented by an infinite number of equivalent state space models. Two solutions have been suggested to overcome this problem: one is to use canonical forms for state space systems and the other one is to use overlapping parametrizations.

In the case of canonical forms, every system in the model set is given a unique representation in the parameter space. Usually, two different types of parameters are needed to specify a state space model. They are the integer-valued structural parameters and the real-valued parameters. The structural parameters determine the block structures of the state space matrices, the number of real-valued parameters needed and also the way in which the real-valued parameters enter the state space matrices. Hence, the real-valued parameters can only be determined after the structural parameters are known. However, the structural parameters, which include the model order, are notoriously hard to determine from the input/output data. So, in the absence of a way of determining the structural parameters, the optimal model can only be obtained by scanning through all the possible model structures. In practice, this is not a feasible solution as the number of model structures increases rapidly with the order of the model. Canonical forms also suffer from problems such as the non-existence of continuous canonical forms in the multi-variable case (see section 7.1.3), and the parameter estimator is close to singularity when the optimization is operating at the proximity of the 'boundary' of the model set. The concept of overlapping parametrizations was introduced to overcome these problems with using canonical forms in system identification. For a brief discussion on the topic of overlapping parametrizations, see section 7.2 in the first part of this thesis.

The purpose of the objective function is to judge the quality of the models in the model set. There are many different ways to define the accuracy of a model and each definition results in a different problem formulation. A way to define the accuracy of a model is to compare how well the model predicts the measured process output when both the model and the process are subjected to the same input excitation. Let θ be the parameter vector, $y(k)$ $(k = 1, ..., N)$ be the output of the process due to the input signal $u(k)$ $(k = 1, ..., N)$, and $\hat{y}(k, \theta)$ $(k = 1, ..., N)$ be the output of the model due to the same input, then a common way to define the objective function is the weighted least square sum of the output error.

$$V(\theta) = \sum_{k=1}^{N} \varepsilon(k, \theta)^T \Lambda \varepsilon(k, \theta) \tag{9.1}$$

where

$$\varepsilon(k, \theta) = y(k) - \hat{y}(k, \theta) \tag{9.2}$$

and Λ is a positive definite matrix. Alternatively, if the model output is calculated based on a prediction based on the past input/output measurements, then the above formulation can be given an interpretation in terms of prediction error [40]. In general, the choice of the objective function should reflect the 'goodness' and 'badness' of the models. So, sometimes it is necessary to introduce a variety of input/output or frequency weighting functions in order that the best model will be selected.

After the parametrization for the model set and the objective function have been specified, we are ready to select a model from the model set. In the discussion to follow, we shall assume that a canonical form is used as the parametrization, and that a time domain l_2-norm output error criterion is used as the objective function. A possible algorithm for model estimation using parameter estimation is as follows.

1. Choose the model order and the structural parameters of the canonical form.

2. By numerical optimization, find a set of real-valued parameters which minimizes V.

Note that the above steps are only part of an iterative scheme. Even after a model which may seem acceptable has been obtained at step 2, the user may choose to go back to step 1 to choose a different value for the model order or a different set of structural parameters, in order to check whether a better model can be obtained.

The above algorithm has basically turned the model estimation problem into a non-linear least squares optimization problem. Many different techniques are available to solve this optimization problem, however none of these techniques are guaranteed to converge and even if the algorithms do converge, there is the question of the rate of convergence. Another problem is that the algorithm may converge to some local minimum which is far from the global minimum, so there is the question of how to choose a good initial estimate for the real-valued parameters. Despite all these drawbacks, the parameter estimation method is still the most commonly used method in model estimation as models of acceptable quality can be obtained in most of the cases. Also, most of the parameter estimation algorithms can be adapted for recursive system identification.

9.3 System identification via model reduction

In this section, we shall discuss how the technique of model reduction can be used in system identification. We shall begin our discussion with some background material on model reduction.

The goal of model reduction for linear systems is to find a lower order approximation to a higher order system. It is desirable that the lower order model approximates the higher order model in some optimal way, perhaps according to some kind of system norm. Alternatively, we may require that the lower order model is a good approximation to the higher order model over some frequency bands. Many algorithms have been proposed in the past, however they are either computationally simple (for example, modal approximation or Padé approximation) but with no guaranteed performance or computationally too demanding. However, there are now at least two very important and computationally simple algorithms for model reduction of stable state space systems. They are balanced truncation [46] and the optimal Hankel norm approximation [20].

The basic idea behind balanced truncation is that every stable state space system can be transformed, via state space transformation, into an internally balanced form. An internally balanced state space system has equal and diagonal controllability and observability Gramians. The diagonal elements of the balanced Gramians are the Hankel singular values of the dynamical systems. The magnitude of a singular value is a measure of how controllable as well as how observable the corresponding state is. Hence, the number of significant singular values can be used as a rough guide to the possible model order of the reduced order model. For balanced truncation, the reduced order model is simply obtained by discarding those states which correspond to the insignificant singular values. It is known that the reduced order model so obtained remains stable and minimal under some easily checked conditions [60] and there exists an explicit formula for the upper bound of the L_∞ error [17, 20, 34]. As it may be desirable that the reduced order model approximates the original system over some frequency bands, Enns [17] introduced the idea of frequency weighted balanced truncation but we shall not go into details here. The unweighted version of balanced truncation will be discussed further in chapters 10 and 11.

Another important model reduction technique is the optimal Hankel norm approximation. As the solution to this approximation problem can be expressed in terms of the balanced state space realization of the higher order system, the computation cost required is rather low. Again, the Hankel singular values of the higher order system provide a rough guide to the possible model order of the lower order model. A frequency weighted version of this method is also available [39]. For both the frequency weighted and the unweighted case, the L_∞ error bound is known [20, 3]. If we are interested in obtaining a good L_2 approximation to the higher order model, it was mentioned in [78] that balanced truncation generally gives a slightly better result than the optimal Hankel norm approximation. This is due to the fact that the optimal Hankel norm approximation is closely related to $\mathcal{L}_\infty$ model reduction.

It has long been established that there is a close link between the subjects of system identification and model reduction, since a set of input/output data of length N can be interpreted as a (not necessarily) minimal linear model of order Nl (l = min(number of inputs, number of outputs)). In [77, 78], Wahlberg proposed an algorithm to do system identification via model reduction. He first estimates a high order ARX estimate from the data and then uses frequency weighted model reduction techniques to find a lower order model using the inverse of the sample variance as the weighting function. In general, the steps to obtain a model are as follows.

1. Form or estimate a high order linear model from the data.

2. Choose a particular model order, possibly based on some characteristics of the high order model, and a frequency weighting if frequency weighted model reduction is to be used.

3. Model reduce the high order model to the chosen model order.

Again, the above steps are part of an iterative scheme. After a model has been obtained, the user may decide to return to step 2 to choose a different model order or a different frequency weighting function.

The main advantage of using this method for system identification is that the algorithm is simple and not computationally demanding. However, the optimality of the model obtained cannot be guaranteed although it was demonstrated in [78] that close to optimality result can be obtained if the frequency weights are chosen carefully.

9.4 Subspace methods

In this section, we shall discuss a new class of methods for system identification which is known as the subspace methods. The term subspace model identification was first introduced by Verhaegen and Deprettere in [72]. It refers to a class of algorithms whose main characteristic is the approximation of subspaces generated by the rows or columns of some block matrices of the input/output data. There are now many different versions of subspace algorithms. These include an early version of subspace algorithm given in the paper by Moonen *et al* [45], Canonical Variate Analysis by Larimore (CVA) [38], Multivariable Output-Error State Space Model Identification (MOESP) by Verhaegen and Dewilde [73, 74], and Numerical algorithm for Subspace State Space System Identification (N4SID) by van Overschee and De Moor [69, 71]. These algorithms differ in the type of state space models which they identified, for example: state space model with output error in the case of MOESP and state space model with both measurement and process noise in the case of N4SID. Although these algorithms are derived using different arguments, it was proved in [70] that the subspace algorithms CVA, MOESP and N4SID are actually related to each other and that they differ only in the choice of weighting functions in a minimization problem. Note that on-line versions of subspace algorithms are also available, see [45, 11]. For a more general survey on subspace algorithms for system identification and signal processing, the reader may refer to the paper [67].

In order to illustrate the features of these subspace methods, we shall now give the outline of a particular subspace algorithm here. The following outline is taken from [45]. Let the input and output data sequences of the process be $u_0, u_1, ..., u_{2i+j-2}$ and $y_0, y_1, ..., y_{2i+j-2}$ respectively. The input sequence is assumed to be deterministic and the aim of the algorithm is to find the following deterministic state space system

$$\begin{cases} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{cases}$$

which approximates the data. The steps to obtain a model are

1. Form the matrices

$$U_{h1} = \begin{bmatrix} u_0 & u_1 & \dots & u_{j-1} \\ u_1 & u_2 & \dots & u_j \\ \vdots & \vdots & \ddots & \vdots \\ u_{i-1} & u_i & \dots & u_{i+j-2} \end{bmatrix}$$

$$U_{h2} = \begin{bmatrix} u_i & u_{i+1} & \dots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & \dots & u_{i+j} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2i-1} & u_{2i} & \dots & u_{2i+j-2} \end{bmatrix}$$

where the number of block columns j is chosen to be much larger than the number of block rows i. The reasons for this choice are that it improves the efficiency of the computation and reduces the algorithm's sensitivity to noise. In a similar manner, we define the matrices Y_{h1} and Y_{h2}. Further define

$$H_1 = \begin{bmatrix} Y_{h1} \\ U_{h1} \end{bmatrix} \qquad H_2 = \begin{bmatrix} Y_{h2} \\ U_{h2} \end{bmatrix}$$

2. Under the assumptions that the input signal is persistently exciting (this is equivalent to checking whether the rank of U_{h1} is equal to mi where m is the number of inputs) and some other technical conditions (see [45] for details), it can be shown that if the data is free of noise then the state space is given by the intersection of the row spaces of H_1 and H_2. Hence the order of the system is given by the dimension of this intersection and at the same time the state sequence

$$X = \begin{bmatrix} x_i & x_{i+1} & \dots & x_{i+j-1} \end{bmatrix} \tag{9.3}$$

can be calculated from the basis vectors of this intersection. In practice, since the data are corrupted by noise, these two row spaces do not intersect with each other. So, we have to resort to finding an approximate intersection of these two spaces and the dimension of this intersection is the order of this system. An approximate state sequence can then be obtained by using an appropriate projection.

3. The state space matrices are given by the linear least squares solution to the following equation

$$\begin{bmatrix} x_{i+1} & x_{i+2} & \dots & x_{i+j-1} \\ y_i & y_{i+1} & \dots & y_{i+j-2} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_i & x_{i+1} & \dots & x_{i+j-2} \\ u_i & u_{i+1} & \dots & u_{i+j-2} \end{bmatrix} \tag{9.4}$$

The basic structures of all the other subspace algorithms are similar to that stated above. The main difference between the subspace algorithms and the other methods for model estimation is that an approximate state sequence is estimated from input/output data before the state space matrices are calculated. The system matrices are then calculated from a set of linear equations such as the one used above.

The subspace methods have attracted a lot of attention recently due to the fact that they address a few problems faced by the parameter estimation methods. We shall now compare the subspace methods with the other two methods of model estimation.

1. Parameter estimation is an iterative non-linear optimization scheme. There are problems with convergence and the existence of local minima. However, both model reduction and subspace methods are non-iterative scheme. They are also generally computationally less demanding compared with the parameter estimation methods.

2. A parametrization is required for the parameter estimation method. There is the problem of model structure selection. Also, the model order has to be chosen beforehand. For both model reduction and subspace methods, a parametrization is not required. In the case of model reduction, the model order can be determined by the number of significant singular values. In the case of subspace methods, the model order is determined by the dimension of some subspace which approximates the state space.

3. The model reduction based methods generally require that the Markov parameters are available. The Markov parameters can be estimated if the test signals used are impulses, steps or pseudo random binary signals (PRBSs). However, in the case of PRBS, the signal usually has to be of a sufficiently long duration in order that good estimates of the Markov parameters can be obtained from correlation analysis. Also, in the multi-input case, the PRBS applied to each input has to be uncorrelated with the others. However, the subspace methods can be used with arbitrary excitation signals provided that these signals are persistently exciting. They can also be used with data generated from impulse or step tests but modification to the data is required beforehand. (See the discussion in section 11.4 on how the impulse or step data can be modified to use with subspace algorithms.)

4. Both model reduction and subspace methods employ numerically reliable tools such as QR factorization and singular value decomposition.

5. For the parameter estimation methods, the model obtained can be guaranteed to be stable if the optimization does converge to a minimum. In general, we cannot guarantee that an optimal model can be obtained but very often a close to optimal model can be obtained. Also, the choice of the objective function allows us to select the models

based on different criteria. For the model reduction based methods, both stability and minimality of the model can be guaranteed but normally not its optimality. However, a frequency weighted version of model reduction can be used to obtain a model which is closed to optimal. For subspace methods, neither stability or minimality of the model can be guaranteed.

6. The distinction between model reduction and subspace methods should not be overemphasized, as some model reduction algorithms can be interpreted as subspace algorithms in the sense that they approximate the higher order state space by one of a lower dimension. In fact, from the comparison above, we find that these two methods do in fact share quite a lot of common features.

9.5 An algorithm for model estimation — implementation

In this section, we shall suggest a scheme for model estimation which combines the three classes of model estimation algorithms discussed above. From the discussion above, it is known that the parameter estimation methods give an optimal model but suffer from the drawback that a non-linear optimization problem has to be solved. There are also problems like the existence of local minima and slow convergence rate. However, the number of iterations required to reach a minimum (possibly global or local) depends very much on how close the initial estimate is to the minimum. This is especially true for numerical optimization algorithms based on the technique of Gauss-Newton which has a good local convergence rate. Thus a good initial estimate will improve the rate of convergence. It will also be able to avoid some undesirable local minima. On the other hand, both model reduction and subspace methods provide simple and efficient ways to obtain a model but with no guarantee on model optimality. However, sufficiently good estimates can usually be obtained from these methods. Based on the discussion above, we propose the following two-step algorithm for model estimation.

1. Use either the model reduction based technique or one of the subspace methods to find a model from input/output data. The model order is decided at this stage based on the information supplied by the algorithm used.

2. Find an optimal model using the parameter estimation method with the model obtained in step 1 as the initial estimate and balanced canonical form for stable systems as the parametrization.

This scheme is motivated by the fact that both model reduction based methods and subspace methods are able to give models which are close to optimal, thus they provide good initial estimate for the parameter estimation algorithm at step 2. As a result, a better convergence rate for the optimization algorithm will be expected and the possibility that the algorithm converges to some undesirable local minimum is reduced. Another desirable aspect of this algorithm is that we do not have to specify the order of the model beforehand. The model order is estimated by the either model reduction based methods or subspace methods. Also, the initial state space estimate obtained at step 1 is (possibly) a helpful guideline to select a model structure of the balanced canonical form. (See section 12.2 for a discussion on using this initial state space estimate for model structure selection.)

This model estimation scheme will be elaborated in the later chapters of this thesis. We shall discuss the issues related to the step of finding an initial estimate in chapter 11 and those related to parameter estimation using balanced canonical forms in chapter 12.

Chapter 10

Balanced realizations and parametrizations

The concept of balanced state space realizations was first introduced by Mullis and Roberts in [47]. Basically a balanced state space realization has equal and diagonal controllability and observability Gramians. It was shown in [47] that fixed point digital filters based on discrete time balanced realizations have minimum roundoff noise at the outputs of the filter. Based on system theoretic arguments, Moore conjectured in [46] that balanced realizations were useful for model reduction of stable systems. Subsequently, balanced realizations were proved to be useful tools in model reduction [17, 20]. In [41], it was suggested that balanced realizations provided 'good' canonical forms for system identification. An explicit parametrization for balanced realizations of stable systems was given in [53] and this makes parameter estimation using balanced realizations more convenient.

In this chapter, we shall give a review of both the balanced realizations and the balanced parametrizations. In section 10.1, we shall define balanced realizations for both continuous time and discrete time stable systems, and discuss their model reduction properties. In section 10.2, we shall present an explicit parametrization for balanced realizations which was stated in [54]. Also, a few alternative parametrizations will be suggested.

10.1 Balanced realizations and truncation

We shall summarise in this section the properties of both continuous time and discrete time balanced state space realizations for stable systems. The balanced realizations given here are also known as Lyapunov balanced realizations due to the fact that they are obtained by balancing the solutions to two dual Lyapunov equations. There are many other different types of balancing, for example, for analytic linear time-varying systems [76, 75], for bounded real systems [56], for stable and minimum-phase systems [43] and for many other types of linear and non-linear systems [35, 54, 62]. The common feature among all these different types of balancing is that they are all based on balancing the solutions to two dual equations.

10.1.1 Discrete time case

Consider the standard linear time-invariant (LTI) discrete time state space system, $S_d = (A_d, B_d, C_d, D_d)$, which is defined by

$$\left\{ \begin{array}{rcl} x[k+1] &=& A_d x[k] + B_d u[k] \\ y[k] &=& C_d x[k] + D_d u[k] \end{array} \right. \tag{10.1}$$

where $x[k] \in \mathbb{R}^n$, $u[k] \in \mathbb{R}^m$ and $y[k] \in \mathbb{R}^p$ for all $k \in \mathbb{Z}$, and $A_d \in \mathbb{R}^{n\times n}$, $B_d \in \mathbb{R}^{n\times m}$, $C_d \in \mathbb{R}^{p\times n}$ and $D_d \in \mathbb{R}^{p\times m}$.

Throughout this section, we shall assume that S_d is asymptotically stable. From standard linear system theory, the controllability Gramian P_d and the observability Gramian Q_d of S_d are respectively given by the solutions to the following two dual discrete time Lyapunov equations.

$$P_d - A_d P_d A_d^T = B_d B_d^T \tag{10.2}$$

$$Q_d - A_d^T Q_d A_d = C_d^T C_d \tag{10.3}$$

Discrete time balanced realizations are defined as follows.

Definition 10.1 *(Moore [46]) The minimal state space model (A_d, B_d, C_d, D_d) is balanced if $P_d = Q_d = \Sigma_d = diag(\sigma_1, ..., \sigma_n)$.*

The quantities $\sigma_1, ..., \sigma_n$ are known as the Hankel singular values of the dynamical system and are input-output invariants of the dynamical system.

In the case where S_d is a minimal realization, the existence of balanced realizations is stated in the following lemma.

Lemma 10.1 *(Moore [46]) For every minimal state space realization (A_d, B_d, C_d, D_d), there exists $T \in Gl_n(\mathbb{R})$ such that $(TA_dT^{-1}, TB_d, C_dT^{-1}, D_d)$ is balanced.*

A more general form of balanced realization exists when S_d is not a minimal realization.

Lemma 10.2 *(Glover [20]) For every state space realization (A_d, B_d, C_d, D_d), there exists $T \in Gl_n(\mathbb{R})$ such that $(TA_dT^{-1}, TB_d, C_dT^{-1}, D_d)$ has controllability and observability Gramians given by*

$$P_d = diag(\Sigma_1, \Sigma_2, 0, 0) \tag{10.4}$$

$$Q_d = diag(\Sigma_1, 0, \Sigma_3, 0) \tag{10.5}$$

where $\Sigma_1, \Sigma_2, \Sigma_3 > 0$ and are diagonal.

The main motivation behind the introduction of balanced state space realizations is that of model reduction. We shall now summarise the model reduction properties of discrete time balanced realizations. Assume that we start with the minimal state space realization $S_d = (A_d, B_d, C_d, D_d)$ and we shall denote the balanced realization of S_d by $S_b = (A_b, B_b, C_b, D_b)$. Let the balanced Gramian of S_b be $\Sigma_b = diag(\sigma_1, ..., \sigma_n)$ such that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n > 0$ and also let x_b be the balanced state vector. In balanced state space coordinates, the singular value corresponding to the i^{th} state of x_b is σ_i.

The reason why balanced realizations are useful for model reduction can be explained as follows: in terms of balanced state coordinates, the states which correspond to the larger singular values are the more controllable and observable states, and vice versa. In other words, the states which correspond to the smaller singular values do not contribute much to the input-output response of the system. Hence, a lower order approximation can be obtained by discarding the states which correspond to the smaller singular values. We shall partition the balanced state vector x_b, the balanced state space matrices A_b, B_b, C_b and the balanced Gramian Σ_b compatibly as follows:

$$\begin{aligned}
x_b &= \begin{bmatrix} x_{b1} \\ x_{b2} \end{bmatrix} && (10.6)\\
A_b &= \begin{bmatrix} A_{b11} & A_{b12} \\ A_{b21} & A_{b22} \end{bmatrix} && (10.7)\\
B_b &= \begin{bmatrix} B_{b1} \\ B_{b2} \end{bmatrix} && (10.8)\\
C_b &= \begin{bmatrix} C_{b1} & C_{b2} \end{bmatrix} && (10.9)\\
\Sigma_b &= \begin{bmatrix} \Sigma_{b1} & 0 \\ 0 & \Sigma_{b2} \end{bmatrix} && (10.10)
\end{aligned}$$

where $\Sigma_{b1} = diag(\sigma_1, ..., \sigma_k)$ and $\Sigma_{b2} = diag(\sigma_{k+1}, ..., \sigma_n)$. The following lemmas state the properties of the reduced order model $(A_{b11}, B_{b1}, C_{b1}, D_b)$.

Lemma 10.3 *(Pernebo and Silverman [60]) If $\sigma_k > \sigma_{k+1}$, then the reduced order model is asymptotically stable and minimal.*

Lemma 10.4 *(Hinrichsen and Pritchard [34]) Define*

$$\begin{aligned}
G_b(z) &= C_b(zI - A_b)^{-1}B_b + D_b && (10.11)\\
G_{br}(z) &= C_{b1}(zI - A_{b11})^{-1}B_{b1} + D_b && (10.12)
\end{aligned}$$

then $\|G_b(z) - G_{br}(z)\|_\infty$ is bounded from above by twice the sum of the discarded singular values with multiple singular values only counted once.

10.1.2 Continuous time case

Consider the standard LTI continuous time state space system $S_c = (A_c, B_c, C_c, D_c)$ defined by

$$\left\{ \begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t) \\ y(t) &= C_c x(t) + D_c u(t) \end{aligned} \right. \qquad (10.13)$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ for all $t \in \mathbb{R}$, and $A_c \in \mathbb{R}^{n \times n}$, $B_c \in \mathbb{R}^{n \times m}$, $C_c \in \mathbb{R}^{p \times n}$ and $D_c \in \mathbb{R}^{p \times m}$.

We shall assume throughout this section that S_c is asymptotically stable. From linear system theory, the controllability Gramian P_c and the observability Gramian Q_c of S_c are respectively given by the solutions to the following two dual Lyapunov equations.

$$A_c P_c + P_c A_c^T = -B_c B_c^T \tag{10.14}$$
$$A_c^T Q_c + Q_c A_c = -C_c^T C_c \tag{10.15}$$

We shall now define balanced state space realizations for stable systems.

Definition 10.2 *(Moore [46]) The minimal state space realization (A_c, B_c, C_c, D_c) is said to be balanced if $P_c = Q_c = \Sigma_c = diag(\sigma_1, ..., \sigma_n)$.*

The quantities $\sigma_1, ..., \sigma_n$ are known as the Hankel singular values and are input-output invariants of the dynamical system. The following lemma states the existence of balanced realizations.

Lemma 10.5 *(Moore [46]) For every minimal (A_c, B_c, C_c, D_c), there exists $T \in Gl_n(\mathbb{R})$ such that $(TA_cT^{-1}, TB_c, C_cT^{-1}, D_c)$ is balanced.*

The following lemma relates the continuous time balanced realizations to their discrete time counterparts.

Lemma 10.6 *(Glover [20]) Let (A_c, B_c, C_c, D_c) be continuous time balanced with Gramian Σ_c, then the discrete time state space model (A_d, B_d, C_d, D_d) as defined by equations 5.30 to 5.33 is discrete time balanced with Gramian Σ_c.*

We shall now summarise the model reduction properties of continuous time balanced realizations.

Let (A_b, B_b, C_b, D_b) be a balanced state space realization with Gramian Σ_b. We shall partition the balanced state space matrices A_b, B_b, C_b and the balanced Gramian Σ_b compatibly as follows:

$$A_b = \begin{bmatrix} A_{b11} & A_{b12} \\ A_{b21} & A_{b22} \end{bmatrix} \tag{10.16}$$

$$B_b = \begin{bmatrix} B_{b1} \\ B_{b2} \end{bmatrix} \tag{10.17}$$

$$C_b = \begin{bmatrix} C_{b1} & C_{b2} \end{bmatrix} \tag{10.18}$$

$$\Sigma_b = \begin{bmatrix} \Sigma_{b1} & 0 \\ 0 & \Sigma_{b2} \end{bmatrix} \tag{10.19}$$

where $\Sigma_{b1} = diag(\sigma_1, ..., \sigma_k)$ and $\Sigma_{b2} = diag(\sigma_{k+1}, ..., \sigma_n)$. The following lemmas state the properties of the reduced order model $(A_{b11}, B_{b1}, C_{b1}, D_b)$.

Lemma 10.7 *(Pernebo and Silverman [60]) If Σ_{b1} and Σ_{b2} have no diagonal entries in common, then the reduced order model is asymptotically stable and minimal. Also the reduced order model is balanced with Gramian Σ_{b1}.*

Lemma 10.8 *(Enns [17], Glover [20]) Define*

$$\begin{aligned} G_b(s) &= C_b(sI - A_b)^{-1}B_b + D_b & (10.20) \\ G_{br}(s) &= C_{b1}(sI - A_{b11})^{-1}B_{b1} + D_b & (10.21) \end{aligned}$$

then $\|G_d(s) - G_{dr}(s)\|_\infty$ is bounded from above by twice the sum of the discarded singular values with multiple singular values only counted once.

10.2 Continuous time balanced canonical forms

It was shown in [41] that balanced realizations were canonical forms for stable state space systems of a fixed McMillan degree. Canonical forms play an important role in system identification as they provide a unique state space representation for each system in the model set. In this section, we shall present a parametrization of the balanced realizations for continuous time stable systems. This parametrization was derived by Ober [53, 54]. In fact, this particular parametrization is a special case of the linear time-varying balanced parametrization introduced in [76, 75]. Note also that the parametrizations for various other forms of balancing are also available, see [54, 43]. In particular, we would like to mention the balanced parametrizations for continuous time stable and minimum-phase systems given in [43] as they are useful for parametrizing state space predictors. (See section 13.2 for an example on the use of stable and minimum-phase balanced realizations for system identification.)

This section is further divided into 5 sub-sections. In section 10.2.1, we shall define a few matrices which are the basic components of the balanced parametrization to be given in section 10.2.2. One of the components of this balanced parametrization is the set of orthogonal matrices. We shall discuss how we can parametrize these orthogonal matrices without introducing non-linear equality constraints in section 10.2.3. A particular structure of this balanced parametrization is known as the generic structure (to be defined in section 10.2.2) and we shall discuss the parametrization of this structure in section 10.2.4. Finally, by putting together the results in sections 10.2.3 and 10.2.4, we shall derive a local coordinate system for the generic structure in section 10.2.5.

10.2.1 Notation

We shall first define what positive upper triangular matrices and balanced matrices are.

Definition 10.3 *A matrix $B = [b_{i,j}]_{k\times l}$ is said to be positive upper triangular if there exist indices $1 \le t_1 \le \ldots \le t_j \le \ldots \le t_k \le l$ such that*

$$\begin{cases} b_{i,t_i} > 0 & \text{for } 1 \le i \le k \\ b_{i,j} = 0 & \text{for } 1 \le j \le t_i \text{ and } 1 \le i \le k \\ b_{i,j} \in \mathbb{R} & \text{otherwise} \end{cases}$$

i.e.

$$B = \begin{bmatrix} 0 & \dots & 0 & b_{1,t_1} & b_{1,t_1+1} & \dots & & & & & & & & \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & b_{2,t_2} & b_{2,t_2+1} & \dots & & & & \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & & & & & & & \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & b_{k,t_k} & b_{k,t_k+1} & \dots \end{bmatrix}$$

Definition 10.4 *A matrix $A \in \mathbb{R}^{n\times n}$ is said to be in r-balanced form, $1 \le r \le n$, if for*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \text{ where } A_{11} \in \mathbb{R}^{r\times r}$$

we have

1. *A_{11} is skew symmetric.*
2. *A_{12} and A_{22} are given by the set of indices*

$$\begin{array}{ccccccccccccc} 1 & = & h_1 & < & \dots & < & h_i & < & h_{i+1} & < & \dots & < & h_q & \le & n-r \\ 1 & \le & g_q & < & \dots & < & g_{i+1} & < & g_i & < & \dots & < & g_1 & \le & r \end{array}$$

in the following ways

(a) *for $A_{12} = [a_{s,t}]_{r\times n-r}$*

$$\begin{cases} a_{g_i,h_i} > 0 & \text{for } 1 \le i \le q \\ a_{g_i,t} = 0 & \text{for } t > h_i \text{ where } 1 \le i \le q \\ a_{s,t} = 0 & \text{for } t \ge h_i \text{ and } s > g_i, \text{ where } 1 \le i \le q \end{cases}$$

i.e.

$$A_{12} = \begin{bmatrix} \vdots & \vdots & & \vdots & \vdots & \vdots & \\ . & . & \dots & a_{g_2-1,h_2-1} & a_{g_2-1,h_2} & a_{g_2-1,h_2+1} & \dots \\ . & . & \dots & a_{g_2,h_2-1} & a_{g_2,h_2} & 0 & \dots \\ . & . & \dots & a_{g_2+1,h_2-1} & 0 & 0 & \dots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \\ a_{g_1-1,h_1} & a_{g_1-1,h_1+1} & \dots & a_{g_1-1,h_2-1} & 0 & 0 & \dots \\ a_{g_1,h_1} & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots \end{bmatrix}$$

(b) A_{22} *is given by*

$$A_{22} = \begin{bmatrix} 0 & \alpha_2 & & & & \\ -\alpha_2 & 0 & \alpha_3 & & & \\ & -\alpha_3 & 0 & . & & 0 \\ & & \ddots & \ddots & \ddots & \\ & 0 & & . & 0 & \alpha_{n-r} \\ & & & & -\alpha_{n-r} & 0 \end{bmatrix}$$

where for $2 \leq i \leq (n-r)$

$$\begin{cases} \alpha_i = 0 & \textit{if } i = h_s \textit{ for some } 1 \leq s \leq q \\ \alpha_i > 0 & \textit{otherwise} \end{cases}$$

3. $A_{21} = -A_{12}^T$

We shall use $[A]_l$ and $[A]_d$ to denote the lower triangular part and the diagonal part of the matrix $A \in \mathbb{R}^{n \times n}$ respectively. If $A = [a_{i,j}]_{n \times n}$, then

1. $[A]_l = [\tilde{a}_{i,j}]_{n \times n}$ is given by

$$\tilde{a}_{i,j} = \begin{cases} 0 & \text{for } j \geq i \\ a_{i,j} & \text{for } j < i \end{cases}$$

2. $[A]_d = [\bar{a}_{i,j}]_{n \times n}$ is given by

$$\bar{a}_{i,j} = \begin{cases} 0 & \text{for } j \neq i \\ a_{i,j} & \text{for } j = i \end{cases}$$

10.2.2 Balanced parametrization

In this section, we shall present a parametrization for the set of all Lyapunov balanced state space systems of a given McMillan degree. This particular parametrization is taken from [54]. Recently, an alternative parametrization for balanced stable state space systems is given in [27], the main feature of this new parametrization being that the structure of the sub-matrices in the 'B' matrix is related to the Kronecker indices of the system.

Lemma 10.9 *(Ober [54]) The following two statements are equivalent.*

1. *$G(s)$ is a $p \times m$ stable rational transfer function of McMillan degree n.*

2. *$G(s)$ has a state space realization $(A, B, C, D) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} \times \mathbb{R}^{p \times n} \times \mathbb{R}^{p \times m}$ given by the parameters:*

$$\begin{array}{llllll}
k & & & & & k \in \mathbb{N},\ 1 \le k \le n \\
\sigma_1 & > \ldots > & \sigma_j & > \ldots > & \sigma_k & \sigma_j \in \mathbb{R},\ \sigma_j > 0 \\
n_1 & \ldots & n_j & \ldots & n_k & n_j \in \mathbb{N} \textit{ such that } \sum_{j=1}^{k} n_j = n \\
r_1 & \ldots & r_j & \ldots & r_k & r_j \in \mathbb{N},\ 1 \le r_j \le\ min(n_j, m, p) \\
U_1 & \ldots & U_j & \ldots & U_k & U_j \in \mathbb{R}^{p \times r_j},\ U_j^T U_j = I_{r_j} \\
\tilde{B}_1 & \ldots & \tilde{B}_j & \ldots & \tilde{B}_k & \tilde{B}_j \in \mathbb{R}^{r_j \times m} \textit{ positive upper triangular} \\
\tilde{A}_1 & \ldots & \tilde{A}_j & \ldots & \tilde{A}_k & \tilde{A}_j \in \mathbb{R}^{n_j \times n_j} \textit{ in } r_j\textit{-balanced form} \\
D & & & & & D \in \mathbb{R}^{p \times m}
\end{array}$$

The state space matrices A, B and C are partitioned according to $n_1, ..., n_k$ in the following way:

$$A = (A_{ij})_{1 \le i,j \le k} \qquad \textit{where } A_{ij} \in \mathbb{R}^{n_i \times n_j}$$

$$B = \begin{pmatrix} B_1 \\ \vdots \\ B_j \\ \vdots \\ B_k \end{pmatrix} \qquad \textit{where } B_j \in \mathbb{R}^{n_j \times m}$$

$$C = \begin{pmatrix} C_1 & \ldots & C_j & \cdots & C_k \end{pmatrix} \quad \textit{where } C_j \in \mathbb{R}^{p \times n_j}$$

The sub-matrices inside A, B, C and D are given by:

$$(a) \quad B_j = \begin{bmatrix} \tilde{B}_j \\ 0 \end{bmatrix} \qquad 1 \le j \le k$$

$$(b) \quad C_j = \begin{bmatrix} U_j \Delta_j & 0 \end{bmatrix} \ \textit{ where } \ \Delta_j = (\tilde{B}_j \tilde{B}_j^T)^{1/2} \qquad 1 \le j \le k$$

$$(c) \quad A_{jj} = \tilde{A}_j - \frac{1}{\sigma_j}\left[diag(\Delta_j^2, 0)\right]_l - \frac{1}{2\sigma_j}\left[diag(\Delta_j^2, 0)\right]_d \quad 1 \le j \le k$$

$$(d) \quad A_{ij} = \frac{1}{\sigma_i^2 - \sigma_j^2}\ diag((\sigma_j \tilde{B}_i \tilde{B}_j^T - \sigma_i \Delta_i U_i^T U_j \Delta_j), 0) \qquad 1 \le i, j \le k, i \ne j$$

$$(e) \quad D \in \mathbb{R}^{p \times m}$$

In (b) above, $(M)^{1/2}$ denotes the positive definite square root of the matrix M.

Moreover, (A, B, C, D) is balanced with Gramian $\Sigma = diag(\sigma_1 I_{n_1}, ..., \sigma_j I_{n_j}, ..., \sigma_k I_{n_k})$ where I_j is the $j \times j$ identity matrix.

The above lemma tells us that each $p \times m$ stable rational transfer function has a unique state space representation given by a set of parameters. There are two types of parameters within the parametrization — structural parameters and real-valued parameters. The structural parameters are the integer-valued parameters k, n_j $(1 \le j \le k)$, and r_j $(1 \le j \le k)$. Each set of structural parameters specifies a certain 'structure' of a system, whereas each set of real-valued parameters specifies a particular system given such a structure. Note that the number of real-valued parameters required to specify a system varies from structure to structure. We shall use the name generic structure to refer to the structure which occurs when all the σ_j's are distinct, or equivalently, when all n_j's are ones. This structure is 'generic' in the sense that if the parameters σ_1 to σ_n are chosen randomly from the interval $(0, \infty)$ with a finite probability distribution, then this structure occurs with probability 1. For the generic structure, the number of real-valued parameters needed to specify a system is more than that of any other structure. The number of real-valued parameters required in this case is $n(m+p)+mp$. It is interesting to compare this with the fact that the set of all $p \times m$ stable LTI of systems of McMillan degree n has the structure of a $n(m+p)+mp$ dimensional manifold (see lemma 3.6). We shall discuss the generic structure further in section 10.2.4.

One particular reason for obtaining a parametrization for a class of systems is so that the parametrization can be used in parameter estimation. The matrix parameters $\tilde{B}_j$ and $\tilde{A}_j$ can be directly parametrized as given in definitions 10.3 and 10.4. The same is true for the matrix parameter D, which can be directly parametrized by $D = [d_{i,j}]_{p \times m}$ where $d_{i,j} \in \mathbb{R}$. However, if we directly parametrize the matrices U_j, we will also need to introduce the non-linear equality constraints $U_j^T U_j = I_{r_j}$ on the parameters. In the next section, we shall give an alternative way to parametrize U_j without introducing any non-linear equality constraints on the parameters. However, this alternative parametrization of orthogonal matrices suffers from two major drawbacks: first, the parametrization is not unique; two, it has integer-valued structural parameters. The result on parametrizing the orthogonal matrices will be used later to develop a local coordinate system for unit vectors in section 10.2.5.

10.2.3 Parametrizations of orthogonal matrices

In this section, we shall give an alternative parametrization for the matrix parameter U_j without introducing any non-linear equality constraints on the parameters. The parametrization given here is adapted from standard work on parametrizing the orthogonal matrix group $\{M \in \mathbb{R}^{p\times p} \ : \ M^T M = I_p\}$, see for example [48, 49], but it has been modified to suit our purpose. The basic idea is that an orthogonal matrix in $\mathbb{R}^{p\times p}$ can be expressed in terms of a sequence of rotation actions in $\mathbb{R}^p$. In the following definition, $R_{st}(\phi_{st})$ is the matrix which performs a rotation of angle ϕ_{st} in the s-t plane in $\mathbb{R}^p$.

Definition 10.5 *For $p > 1$, let $R_{st}(\phi_{st}) = [r_{i,j}] \ \in \mathbb{R}^{p\times p}$ with $1 \le s < t \le p$ be defined as follows:*

$$r_{i,j} = \begin{cases} 1 & \text{for } i = j \text{ and } i \neq s \text{ and } i \neq t \\ cos(\phi_{st}) & \text{for } i = j = s \text{ or } i = j = t \\ sin(\phi_{st}) & \text{for } i = t \text{ and } j = s \\ -sin(\phi_{st}) & \text{for } i = s \text{ and } j = t \\ 0 & \text{otherwise} \end{cases}$$

In other words, $R_{st}(\phi_{st})$ is the matrix

$$
\begin{array}{cc}
\left[\begin{array}{ccccccccccc}
1 & & & & & & & & & & 0\\
0 & \ddots & & & & & & & & & \vdots\\
 & & 1 & & & & & & & & \\
 & & & cos(\phi_{st}) & & & & -sin(\phi_{st}) & & & \\
 & & & & 1 & & & & & & \\
 & & & & & \ddots & & & & & \\
 & & & & & & 1 & & & & \\
 & & & sin(\phi_{st}) & & & & cos(\phi_{st}) & & & \\
 & & & & & & & & 1 & & \\
 & & & & & & & & & \ddots & 0\\
0 & & & & & & & & & & 1
\end{array}\right]
&
\begin{array}{l}
\\ \\ \\ \leftarrow s^{th}\ row \\ \\ \\ \\ \leftarrow t^{th}\ row \\ \\ \\ \\
\end{array}
\\
\begin{array}{ccccccccccc}
 & & & \uparrow & & & & \uparrow & & & \\
 & & & s^{th}\ column & & & & t^{th}\ column & & &
\end{array}
&
\end{array}
$$

The matrices $V \in \mathbb{R}^{p\times p}$ such that $V^TV = I_p$ can be parametrized using $p(p-1)/2$ real-valued parameters and p integer-valued parameters, as follows.

Lemma 10.10 *For $p > 1$, let V be a real $p \times p$ matrix such that $V^TV = I_p$. V can be parametrized by*

1. *$p(p-1)/2$ real parameters $\phi_{s,t}$ where $s, t \in \mathbb{N}$, $1 \le s < t \le p$, $\phi_{s,t} \in (-\pi/2, \pi/2]$*
2. *p integer parameters $k_1, k_2, \ldots, k_p$ where $k_i = \pm 1$*

Let $K_p \in \mathbb{R}^{p\times p}$ be the matrix $diag(k_1, k_2, \ldots, k_p)$, then the parametrization can be written as follows:

For $p = 2$,

$$V = R_{12}(\phi_{12})K_2$$

For $p = 3$,

$$V = R_{12}(\phi_{12})R_{13}(\phi_{13})R_{23}(\phi_{23})K_3$$

For $p = 4$,

$$V = R_{12}(\phi_{12})R_{13}(\phi_{13})R_{14}(\phi_{14})R_{23}(\phi_{23})R_{24}(\phi_{24})R_{34}(\phi_{34})K_4$$

In general,

$$\begin{aligned} V = {} & R_{12}(\phi_{12})\ldots R_{1,p}(\phi_{1,p})R_{23}(\phi_{23})\ldots R_{2p}(\phi_{2p})R_{34}(\phi_{34})\ldots \\ & R_{p-2,p-1}(\phi_{p-2,p-1})R_{p-2,p}(\phi_{p-2,p})R_{p-1,p}(\phi_{p-1,p})K_p \end{aligned} \tag{10.22}$$

Note that the above parametrization is only unique for $p = 2$. For $p > 2$, the parametrization is not unique when $\phi_{s,t} = \pi/2$ and we shall therefore restrict the range of $\phi_{s,t}$ to $(-\pi/2, \pi/2)$. Although this removes certain orthogonal matrices from the range space, this parametrization can still be used for optimization as the points removed are at the boundary of the domain. So, for this parametrization, we have removed the non-linear equality constraints at the expense of introducing p integer-valued structural parameters.

A natural question to ask at this point is how the parameters ϕ_{st}'s and k_i's can be calculated given an orthogonal matrix V which can be represented by the above parametrization. In order to answer this question, we need the following theorem on the property of the product given in equation 10.22.

Theorem 10.1 *Let $s, t \in \mathbb{N}$, $1 \le s < t \le p$, $(s,t) \neq (1,2)$ and*

$$V_{s,t} = R_{st}(\phi_{st}) \dots R_{p-1,p}(\phi_{p-1,p}) K_p \tag{10.23}$$

where the order in which the matrices are multiplied together is the same as that in equation 10.22.

For any (s,t), define

$$(i,j) = \begin{cases} (t-1, s) & \text{if } t \neq s+1 \\ (p, s-1) & \text{if } t = s+1 \end{cases}$$

then the (i,j) element of $V_{s,t}$ is zero.

Proof: Note that $R_{j,i}(\phi_{j,i})$ is the rotation matrix just before $R_{s,t}(\phi_{s,t})$ in the sequence of matrix product in equation 10.22. Recall that the matrix $R_{x,y}(\phi_{x,y})$ is a rotation matrix on the (x,y)-plane, thus the matrix product

$$V_{s,t} = R_{st}(\phi_{st}) \dots R_{p-1,p}(\phi_{p-1,p}) K_p \tag{10.24}$$

performs rotations sequentially on the $(p-1,p)$-plane, $(p-2,p)$-plane, $(p-2,p-1)$-plane, $(p-3,p)$-plane, ..., (s,t)-plane. Thus no rotation has been performed on the $(1,2)$-plane, $(1,3)$-plane, ..., (j,i)-plane yet. Hence the (i,j) element of $V_{s,t}$ is zero. In fact, the $(2,1)$, $(3,1)$, ..., (i,j) elements of $V_{s,t}$ are all zeros. $\square$

By using the property given in the above theorem, the parameters ϕ_{st}'s and k_i's can be calculated from V by recursively multiplying V from the left by $R_{s,t}^T$ and setting the (t,s) element of the product to zero. The steps to obtain the parameters are:

1. Set $s = 1$, $t = 2$ and $M = V$.
2. By the property stated in theorem 10.1, by equating the (t,s) element of $R_{st}(\phi_{st})^T M$ to zero, we have

$$-sin(\phi_{st}) m_{ss} + cos(\phi_{st}) m_{st} = 0 \tag{10.25}$$

   Set ϕ_{st} equal to $\tan^{-1}(\frac{m_{st}}{m_{ss}})$ and M equal to $R_{st}(\phi_{st})^T M$.

3. If $s = p-1$ and $t = p$, goto step 4. Otherwise, update the values of s and t according to the following rules.

   (a) If $t < p$, then increase t by 1.

   (b) If $t = p$ and $s < p-1$, then increase s by 1 and set $t = s+1$.

   Go back the step 2.

4. Read the values of k_i off the diagonal of M.

By using the parametrization for the orthogonal matrices given above, we shall now show how we can parametrize the set $\{U \in \mathbb{R}^{p\times r} \ : \ U^T U = I_r\}$ for all positive integers $r \leq p$.

Theorem 10.2 *Let U be a real $p \times r$ matrix such that $U^T U = I_r$. U can be parametrized by*

1. *$r(2p-r-1)/2$ real parameters $\phi_{s,t}$ where $s, t \in \mathbb{N}$, $s = r, r+1, \ldots, p-1$, $t = s+1, \ldots, p$, and $\phi_{s,t} \in (-\pi/2, \pi/2]$*

2. *r integer parameters $k_1, k_2, \ldots, k_r$ where $k_i = \pm 1$*

   Let

$$K_r \;=\; \begin{bmatrix} diag(k_1, k_2, \ldots, k_r) \\ 0_{p-r,r} \end{bmatrix} \tag{10.26}$$

where $0_{p-r,r}$ is a $(p-r) \times r$ zero matrix, then the parametrization of U is given by

$$\begin{aligned} U \;=\;& R_{1,2}(\phi_{1,2}) \ldots R_{1,p}(\phi_{1,p}) R_{2,3}(\phi_{2,3}) \ldots R_{2,p}(\phi_{2,p}) \ldots \\ & R_{r,r+1}(\phi_{r,r+1}) \ldots R_{r,p}(\phi_{r,p}) K_r \end{aligned} \tag{10.27}$$

Proof: Define

$$M \;=\; \begin{bmatrix} I_r \\ 0_{p-r,r} \end{bmatrix}$$

and note that K_r has the following property

$$R_{s,t}(\phi_{s,t})\; K_r \;=\; K_r \quad \text{for all } s, t \text{ such that } t > s > r \tag{10.28}$$

Let V be the matrix defined in equation 10.22. We shall take U to be the first r columns of V, or

$$\begin{aligned} U \;&=\; V\,M \\ &=\; R_{12}(\phi_{12}) R_{13}(\phi_{13}) \ldots R_{p-2,p}(\phi_{p-2,p}) R_{p-1,p}(\phi_{p-1,p}) K_p\, M \qquad (10.29) \\ &\;\; \text{(by equation 10.22)} \\ &=\; R_{12}(\phi_{12}) R_{13}(\phi_{13}) \ldots R_{p-2,p}(\phi_{p-2,p}) R_{p-1,p}(\phi_{p-1,p}) K_r \\ &=\; R_{1,2}(\phi_{1,2}) \ldots R_{1,p}(\phi_{1,p}) R_{2,3}(\phi_{2,3}) \ldots R_{2,p}(\phi_{2,p}) \ldots \\ &\;\; R_{r,r+1}(\phi_{r,r+1}) \ldots R_{r,p}(\phi_{r,p}) K_r \qquad (10.30) \\ &\;\; \text{(by equation 10.28)} \end{aligned}$$

Note that only $r(2p-r-1)/2$ real-valued parameters appear in U. Hence the lemma. $\Box$

Again the parameters $\phi_{s,t}$'s and k_i's can be calculated using a procedure which is similar to the one used above to calculate the parameters of orthogonal matrices.

A special case of the above theorem is when r is 1, then we obtain a parametrization for the set of p-dimensional unit vectors. We shall defer the discussion of this special case to the next section where we shall discuss the generic case of balanced parametrizations.

10.2.4 Generic structure

As mentioned in the previous section, the generic structure of the balanced canonical form consists of all systems with distinct Hankel singular values. It was also mentioned that this structure was called generic because if we choose the Hankel singular values randomly from $(0, \infty)$ with a finite probability distribution, then the probability of getting such a system is 1. The number of real-valued parameters needed to specify a system in the generic structure is always $n(m+p)+mp$, whereas in the non-generic case, for multi-variable systems, the number of real-valued parameters needed is always less than or equal to $n(m+p)+mp-2$. Hence, in the multi-variable case, the space of all generic systems forms a connected region (see [19] or section 7.1.2).

Another interesting fact concerning the generic structure is that its parametrization is much simpler than all the other structures. The key point is, when all singular values are distinct, the structural parameter k takes the value of n and what follows is:

1. $n_j = 1$ for $j = 1, ..., n$
2. $r_j = 1$ for $j = 1, ..., n$ since r_j is constrained to be between 1 and $min(n_j, m, p)$
3. $\tilde{A}_j$ is a scalar and in 1-balanced form, so $\tilde{A}_j = 0$

Similar to the general case, the matrix parameters U_i can be parametrized in a few different ways. In the generic case, these matrix parameters become unit vectors. If these unit vectors are directly parametrized, again a number of non-linear equality constraints will need to be introduced. Although the unit vectors can also be parametrized by other methods, it is a known fact that the set of all unit vectors cannot be parametrized by using only one coordinate system, so it becomes necessary to introduce some extra structural parameters. In this section, we shall give a parametrization which covers almost all the generic systems without introducing either non-linear equality constraints or any extra structural parameters. This parametrization will form the basis on which we develop a local parametrization for the generic structure in the next section. We shall start with parametrizing 'half' of the set of all p-dimensional unit vectors.

Lemma 10.11 *For $p > 1$, let $U \in \{(u_1, u_2, \ldots, u_p) \in \mathbb{R}^{p\times 1} : u_1^2 + u_2^2 + \ldots + u_p^2 = 1, u_1 > 0\}$ then U can be uniquely parametrized by the parameters $\phi_1, \phi_2, \ldots, \phi_{p-1}$ where $\phi_i \in (-\pi/2, \pi/2)$*

U is given by

$$\left\{\begin{array}{lcl} u_1 & = & cos(\phi_{p-1})cos(\phi_{p-2})cos(\phi_{p-3})\ldots cos(\phi_2)cos(\phi_1) \\ u_2 & = & cos(\phi_{p-1})cos(\phi_{p-2})cos(\phi_{p-3})\ldots cos(\phi_2)sin(\phi_1) \\ u_3 & = & cos(\phi_{p-1})cos(\phi_{p-2})cos(\phi_{p-3})\ldots sin(\phi_2) \\ \vdots & & \vdots \\ u_{p-1} & = & cos(\phi_{p-1})sin(\phi_{p-2}) \\ u_p & = & sin(\phi_{p-1}) \end{array}\right. \tag{10.31}$$

Also, the parameters $\phi_1, \phi_2, \ldots, \phi_{p-1}$ *can be obtained from* $(u_1, u_2, \ldots, u_p)$ *using the following equations*

$$\left\{\begin{array}{lcl} tan(\phi_1) & = & \frac{u_2}{u_1} \\ tan(\phi_2) & = & \frac{u_3}{u_1}cos(\phi_1) \\ tan(\phi_3) & = & \frac{u_4}{u_1}cos(\phi_1)cos(\phi_2) \\ \vdots & & \vdots \\ tan(\phi_{p-1}) & = & \frac{u_p}{u_1}cos(\phi_1)cos(\phi_2)\ldots cos(\phi_{p-2}) \end{array}\right. \tag{10.32}$$

Proof: This lemma is a special case of theorem 10.2 for $r = 1$. By theorem 10.2, if u is $p \times 1$ unit vector, then it can be parametrized by

1. $(p-1)$ real-valued parameters ϕ_{12}, ϕ_{13}, ..., ϕ_{1p}
2. an integer parameter $k_1 = \pm 1$

and the parametrization is given by

$$u \;=\; R_{12}(\phi_{12})R_{13}(\phi_{13})\ldots R_{1p}(\phi_{1p})K_1 \tag{10.33}$$

where $K_1 = [k_1, 0, \ldots, 0]^T$. If we rename ϕ_{1i} to ϕ_{i-1} and let $k_1 = 1$, then it can be proved by mathematical induction that the vector u has the form given above. Since we have restricted the range of ϕ_i to $(-\pi/2, \pi/2)$, u_1 is strictly positive. Finally, it can easily be seen from equations 10.31 and 10.32 that this parametrization is unique. □

By changing the range of ϕ_1 from $(-\pi/2, \pi/2)$ to $(-\pi/2, \pi/2) \cup (\pi/2, 3\pi/2)$, then the equations given in the above lemma parametrize all the $p \times 1$ unit vectors such that $u_1 \neq 0$. Since we can identify the set of all $p \times 1$ unit vectors such that $u_1 = 0$ with the set of all $(p-1) \times 1$ unit vectors, we can use the above parametrization recursively to parametrize all the p-dimensional unit vectors uniquely. The only drawback with this parametrization for the unit vectors is that it requires several different structures to cover the whole set. Based on the parametrization for the unit vectors as given in lemma 10.11, we shall now give a parametrization which covers almost all the generic systems.

Lemma 10.12 *Define the map*

$$\begin{array}{rccc} f: & \{(-\pi/2, \pi/2) \cup (\pi/2, 3\pi/2)\} \times (-\pi/2, \pi/2)^{p-2} & \to & U \\ & (\phi_1, \phi_2, ..., \phi_{p-1}) & \mapsto & (u_1, u_2, \ldots, u_p) \end{array} \tag{10.34}$$

where $(u_1, u_2, \ldots, u_p)$ is given by equation 10.31.

Almost all of the $p \times m$ stable rational transfer functions of McMillan degree n with distinct Hankel singular values can be parametrized in the following way:

$$\begin{array}{lllllll}
\sigma_1 & > \ldots > & \sigma_j & > \ldots > & \sigma_n & \sigma_j \in \mathbb{R},\ \sigma_j > 0 \\
\phi_1 & \ldots & \phi_j & \ldots & \phi_n & \phi_j = [\phi_{j,i}] \in \mathbb{R}^{(p-1)\times 1},\ \phi_{j,i} \in (-\pi/2, \pi/2) \\
B_1 & \ldots & B_j & \ldots & B_n & B_j \in \mathbb{R}^{1\times m} \text{ such that the first element in the} \\
 & & & & & \text{row is positive} \\
D & & & & & D \in \mathbb{R}^{p\times m}
\end{array}$$

The state space matrices A,B,C and D are given by:

1. $$B = \begin{pmatrix} B_1 \\ \vdots \\ B_j \\ \vdots \\ B_n \end{pmatrix}$$

2. $C = \begin{pmatrix} C_1 & \ldots & C_j & \cdots & C_n \end{pmatrix}$ *where* $C_j = f(\phi_j)\sqrt{B_j B_j^T}$

3. $A = [a_{ij}]$ *for* $i, j = 1, ..., n$ *where*

$$a_{ij} = \begin{cases} -\frac{1}{2\sigma_j} B_j B_j^T & \text{for } i = j \\ \frac{1}{\sigma_i^2 - \sigma_j^2}\left(\sigma_j B_i B_j^T - \sigma_i C_i^T C_j\right) & \text{for } i \neq j \end{cases}$$

4. $D = D$

Moreover, (A, B, C, D) is balanced with Gramian $\Sigma = diag(\sigma_1, ..., \sigma_j, ..., \sigma_n)$

The above lemma leaves out the systems of the following form: let (A, B, C, D) be a generic system in balanced canonical form and $C = [c_{ij}]_{p\times n}$, then (A, B, C, D) is not covered by the above parametrization if $c_{1j} = 0$ for any $j = 1, \ldots n$.

10.2.5 Local coordinate system for unit vectors

In the previous sections, we found that the matrix parameter U_i can be parametrized in two ways — direct parametrization or parametrization using rotation matrices. However, we need to introduce non-linear equality constraints in the former case and structural parameters in the latter. Also, it is a known fact that there does not exist a single coordinate system which covers the set of all orthogonal matrices. In the last section, we have introduced a coordinate system which covers almost all the $p \times 1$ unit vectors. However, with this particular parametrizations, some points are permanently excluded. In order to overcome this drawback, we shall present a local coordinate system for the set of all $p \times 1$ unit vectors in this section.

Theorem 10.3 *Let $u = (u_1^0, u_2^0, \ldots, u_p^0)$ be a unit vector such that $u_1^0 > 0$ and $\phi_1^0, \phi_2^0, \ldots, \phi_{p-1}^0$ be the angles corresponding to the unit vector $(u_1^0, u_2^0, \ldots, u_p^0)$ as given by equation 10.32. Also let $R_{s,t}(\phi_{s,t})$ be the rotation matrix as defined before, define the unit vector*

$$\begin{aligned} \tilde{u} &= R_{12}(\phi_1^0)R_{13}(\phi_2^0)\ldots R_{1p}(\phi_{p-1}^0) \times \\ & \begin{bmatrix} cos(\phi_{p-1})cos(\phi_{p-2})cos(\phi_{p-3})\ldots cos(\phi_2)cos(\phi_1) \\ cos(\phi_{p-1})cos(\phi_{p-2})cos(\phi_{p-3})\ldots cos(\phi_2)sin(\phi_1) \\ cos(\phi_{p-1})cos(\phi_{p-2})cos(\phi_{p-3})\ldots sin(\phi_2) \\ \vdots \\ cos(\phi_{p-1})sin(\phi_{p-2}) \\ sin(\phi_{p-1}) \end{bmatrix} \end{aligned} \tag{10.35}$$

where $\phi_i \in (-\pi/2, \pi/2)$.

The angle between the vectors u and $\tilde{u}$ is always less than $\pi/2$.

Proof: Note that by theorem 10.2, the unit vector u is in fact given by

$$u = R_{12}(\phi_1^0)R_{13}(\phi_2^0)\ldots R_{1p}(\phi_{p-1}^0)e_1 \tag{10.36}$$

where $e_1 = (1, 0, \ldots, 0)$. If α denotes the angle between the vectors u and $\tilde{u}$, then

$$\begin{aligned} cos(\alpha) &= u^T\,\tilde{u} \\ &= cos(\phi_{p-1})cos(\phi_{p-2})cos(\phi_{p-3})\ldots cos(\phi_2)cos(\phi_1) \end{aligned} \tag{10.37}$$

as $R_{ij}(\phi_{ij})^T R_{ij}(\phi_{ij}) = I$. Since the angles $\phi_1, \phi_2, \ldots, \phi_{p-1}$ lie between $-\pi/2$ and $\pi/2$, hence $cos(\alpha)$ is strictly positive i.e. $0 \leq \alpha < \pi/2$. □

For the case where $u_1 < 0$, a similar expression can be derived but ϕ_1^0 will be between $(\pi/2, 3\pi/2)$ rather than $(-\pi/2, \pi/2)$. For the case when $u_1 = 0$, the situation is not much more complicated. Since u is a unit vector, we can always find a non-zero element u_j in u. We then use a linear transformation to swap the first and the j^{th} elements in u and then we can proceed according to the above theorem.

There are two advantages in using this particular local coordinate system. Firstly, no points in the range space are permanently excluded. Secondly, when this parametrization is used for optimization, the distortion which this local coordinate system introduces around the current estimate is always the same; whereas the coordinate system given in lemma 10.11 introduces the least amount of distortion only at the point $(1, 0, \ldots, 0)$. The second observation can be explained by noting the following two facts:

1. Theorem 10.3 can be given an alternative interpretation as follows: since we can identify all the $p \times 1$ unit vectors with the points on the 'sphere' S^{p-1} (S^1 being the unit circle in $\mathbb{R}^2$), the above theorem parametrizes the 'hemisphere' whose pole is the point $u = (u_1^0, u_2^0, \ldots, u_p^0)$.

2. Lemma 10.11 parametrizes the 'hemisphere' in $\mathbb{R}^p$ whose pole is $(1, 0, \ldots, 0)$.

For the case where $r > 1$, an expression which is similar to equation 10.35 can be derived. However, this expression still involves the integer parameters $k_1, k_2, ..., k_r$ (see theorem 10.2). These integer parameters are necessary because they allow the relative orientation between vectors (in the case of $\mathbb{R}^3$, there are two relative orientations — right-hand and left-hand) to change.

10.3 Summary

In section 10.1, we have defined balanced realizations in the sense of Moore and summarised their model reduction properties. In section 10.2.2, we have presented the balanced parametrization for stable state space systems derived by Ober. A particular component of this parametrization is the group of orthogonal matrices and we have discussed the parametrization of this group using rotation matrices in section 10.2.3. Although this parametrization of the orthogonal group does not introduce any non-linear equality constraints, it suffers from the two drawbacks: one, integer-valued structural parameters are required; two, the parametrization is not unique. However, this theory of parametrization of the orthogonal matrices led us to derive a local coordinate system for the generic case. This local coordinate system was presented in section 10.2.5.

Chapter 11

Finding an initial estimate

In this chapter, we shall study the problem of finding an initial state space estimate for parameter estimation. This forms the first step of our algorithm for model estimation which was given in section 9.5. It was also suggested in section 9.5 that both model reduction based algorithms and subspace methods were able to give sufficiently good initial estimates for parameter estimation. We shall look at three specific algorithms for calculating state space models from input/output data.

The three methods are Kung's algorithm, a model reduction algorithm due to Al-Saggaf and N4SID (which stands for Numerical Algorithm for Subspace State Space System Identification). Kung's algorithm has its origin from realization theory but it is both a model reduction algorithm as well as a subspace method. The algorithm due to Al-Saggaf's was proposed for model reduction of discrete time systems, but since it is related to continuous time balanced truncation, it can also be looked at as a subspace algorithm. These three methods will be discussed in sections 11.1 (Kung's), 11.2 (Al-saggaf's) and 11.3 (N4SID). Finally, we shall compare these three methods in section 11.4.

11.1 Kung's algorithm

Kung's algorithm [37] was suggested as a solution to the following approximation problem: given the Markov parameter sequence $\{J_0, J_1, J_2, \ldots J_N\}$, find a state space realization which approximates the sequence. Many algorithms have been suggested before Kung's and the main problem with these algorithms is that they only work well when the Markov parameter sequence is generated by a finite dimensional linear system. Thus, these algorithms cannot be used if the Markov parameters have come from a non-linear dynamical system. However, even in the case when the Markov parameters have come from a finite dimensional linear system, a slight numerical perturbation of some of the Markov parameter entries can destroy this property. Kung's algorithm provides a numerically robust solution to this problem based on its use of singular value decomposition which is itself a robust numerical tool.

This section is further divided into 4 sub-sections. In section 11.1.1, we shall give an outline of the algorithm. The approximate realization given by Kung's algorithm is closely related to discrete time balanced truncation and we shall discuss this in section 11.1.2. In section 11.1.3, we shall discuss a practical aspect on using Kung's algorithm, namely the pre-scaling of the data. Finally, we shall give a simulation example in section 11.1.4.

11.1.1 Outline of the algorithm

In this section, we shall discuss how an n^{th} order state space realization can be obtained from the Markov parameter sequence $\{J_0, J_1, ..., J_N\}$ using Kung's algorithm. Throughout this section, we shall use m and p to denote the numbers of inputs and outputs respectively. The steps to obtain an n^{th} order state space model are as follows:

Form the block Hankel matrix

$$H_N = \begin{bmatrix} J_1 & J_2 & \dots & J_N \\ J_2 & J_3 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ J_N & 0 & \dots & 0 \end{bmatrix} \tag{11.1}$$

whose lower triangular blocks are padded with zero matrices. We shall find the singular value decomposition

$$H_N = U\Sigma V \tag{11.2}$$

where U is $pN \times lN$, V is $lN \times mN$, $l = min(m,p)$ and

$$\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_{lN}) \tag{11.3}$$

with $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{lN} \geq 0$.

Now partition the matrices U, Σ and V in a compatible way as

$$U = \begin{bmatrix} U_n & \bar{U}_n \end{bmatrix} \quad \Sigma = \begin{bmatrix} \Sigma_n & 0 \\ 0 & \bar{\Sigma}_n \end{bmatrix} \quad V = \begin{bmatrix} V_n \\ \bar{V}_n \end{bmatrix} \tag{11.4}$$

where Σ_n is n by n and $\sigma_n > 0$. We further partition U_n and V_n as

$$U_n = \begin{bmatrix} U_n^1 \\ U_n^2 \\ \vdots \\ U_n^N \end{bmatrix} \qquad V_n = \begin{bmatrix} V_n^1 & V_n^2 & \cdots & V_n^N \end{bmatrix} \tag{11.5}$$

where each U_n^i is $p \times n$ and each V_n^i is $n \times m$. Define

$$U_n^{\uparrow} = \begin{bmatrix} U_n^2 \\ \vdots \\ U_n^N \\ 0_{p,n} \end{bmatrix} \qquad \overleftarrow{V}_n = \begin{bmatrix} V_n^2 & \cdots & V_n^N & 0_{n,m} \end{bmatrix} \tag{11.6}$$

According to Kung, an n^{th} order approximate realization is given by

$$A_n = \Sigma_n^{-\frac{1}{2}} U_n^T U_n^{\uparrow} \Sigma_n^{\frac{1}{2}} \tag{11.7}$$

$$B_n = \Sigma_n^{\frac{1}{2}} V_n^1 \tag{11.8}$$

$$C_n = U_n^1 \Sigma_n^{\frac{1}{2}} \tag{11.9}$$

$$D_n = J_0 \tag{11.10}$$

or

$$
\begin{aligned}
\tilde{A}_n &= \Sigma_n^{\frac{1}{2}} \overleftarrow{V}_n \, V_n^T \Sigma_n^{-\frac{1}{2}} && (11.11)\\
\tilde{B}_n &= \Sigma_n^{\frac{1}{2}} V_n^1 && (11.12)\\
\tilde{C}_n &= U_n^1 \Sigma_n^{\frac{1}{2}} && (11.13)\\
\tilde{D}_n &= J_0 && (11.14)
\end{aligned}
$$

Remarks

There is an alternative version of Kung's algorithm which starts with the following Hankel matrix

$$
\tilde{H} = \begin{bmatrix} J_1 & J_2 & \dots & J_k \\ J_2 & J_3 & \dots & J_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ J_k & J_{k+1} & \dots & J_{2k-1} \end{bmatrix} \qquad (11.15)
$$

where $N = 2k - 1$ (assuming that N is odd). Since a smaller Hankel matrix is used for this version, the amount of computer memory storage needed will be much smaller. However, we have chosen to use the first version of Kung's algorithm because the state space models obtained by this method can be proved to be both stable and minimal.

There are also a few other state space realization algorithms which are based on singular value decomposition of matrices formed by the Markov parameters [82, 13], however the stability and minimality of the realizations obtained from these algorithms cannot be guaranteed.

11.1.2 Properties of Kung's estimate

In [37], Kung showed that the 'A' matrix of the state space model is asymptotically stable and its spectral norm is less than or equal to unity. In this section, we shall state the properties of the state space model obtained via Kung's algorithm. The main result of this section is that Kung's algorithm is equivalent to discrete time balanced truncation. Throughout this section, we shall assume that the Hankel matrix H_N given in equation 11.1 has rank r and $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0$.

We shall begin with the following theorem which states that the two alternative state space realizations (A_n, B_n, C_n, D_n) (given in equations 11.7 to 11.10) and $(\tilde{A}_n, \tilde{B}_n, \tilde{C}_n, \tilde{D}_n)$ (given in equations 11.11 to 11.14) are in fact identical.

Theorem 11.1 *For all $n \leq r$, $A_n = \tilde{A}_n$.*

The proof of this theorem is given in appendix A.

We shall now state the main result of this section.

Theorem 11.2 *(Kung's algorithm is equivalent to discrete time balanced truncation) Let (A, B, C, D) be a discrete time state space system with finite impulse response and whose* *finite Markov parameter sequence is given by* $\{J_0, J_1, ..., J_N\}$*. Also let* n *be an integer such that* $1 \leq n \leq r$*, then the* n^{th} *order state space system obtained from discrete time balanced truncation of* (A, B, C, D) *is identical to the* n^{th} *order state space system obtained from applying Kung's algorithm to the sequence* $\{J_0, J_1, ..., J_N\}$*.*

The proof of this theorem can be found in appendix A. We shall give an outline of the proof here. The finite Markov parameter sequence $\{J_0, J_1, ..., J_N\}$ can be generated by the following discrete time state space system

$$\begin{cases} x[k+1] = Ax[k] + Bu[k] \\ y[k] = Cx[k] + Du[k] \end{cases} \tag{11.16}$$

where

$$A = \begin{bmatrix} 0 & I_p & & \\ & & \ddots & \\ & & & I_p \\ & & & 0 \end{bmatrix} \tag{11.17}$$

$$B = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_N \end{bmatrix} \tag{11.18}$$

$$C = \begin{bmatrix} I_p & 0 & \dots & 0 \end{bmatrix} \tag{11.19}$$

$$D = J_0 \tag{11.20}$$

Note that this state space realization is asymptotically stable but is not necessarily minimal. If we find the balanced realization of (A, B, C, D) and truncate it to the n^{th} order, then it can be shown that the state space matrices of this n^{th} order model are in fact given by equations 11.7 to 11.10.

A similar result was proved in [63] but based on the assumptions that an infinite block Hankel matrix is available and that the Markov parameters are generated by a minimal state space realization of a certain McMillan degree. An alternative but more abstract interpretation of this result is as follows: the infinite block Hankel matrix corresponding to the state space system (A, B, C, D) is given by

$$H_\infty = \begin{bmatrix} H_N & 0 & \dots \\ 0 & 0 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{11.21}$$

The singular value decomposition of H_∞ is

$$H_\infty = \underbrace{\begin{bmatrix} U & 0 \\ 0 & Q_1 \end{bmatrix}}_{U_\infty} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & Q_2 \end{bmatrix} \tag{11.22}$$

where Q_1 and Q_2 are some arbitrary infinite orthogonal matrices. It is known that if we use the columns from U_∞ which correspond to non-zero singular values as basis vectors for the state space, then we obtain the balanced state space realization of (A, B, C, D). In the case of Kung's algorithm, we select the first n columns of U_∞ and this corresponds to truncation to the n^{th} order.

We shall now turn to the properties of Kung's state space models. We shall start with the full order model i.e. the r^{th} order model.

Theorem 11.3 *(Properties of the r^{th} order model) The r^{th} order state space realization (A_r, B_r, C_r, D_r) is an exact realization of the finite impulse response with Markov parameters $\{J_0, J_1, ..., J_N\}$. This realization is asymptotically stable, minimal and balanced with Gramian given by $diag(\sigma_1, \sigma_2, ..., \sigma_r)$. Moreover, the spectral norm of A_r ($\|A_r\|$) is less than or equal to one. If the singular values $\sigma_1, \sigma_2, ..., \sigma_r$ are all distinct then $\|A_r\|$ is strictly less than 1.*

The proof of this theorem can be found in appendix A.

The fact that $\|A_r\| < 1$ means that the matrix A_r has the following desirable property: let $\lambda_i(A_r)$ be any eigenvalue of A_r, then

$$|\lambda_i(A_r)| \leq \|A_r\|$$

If the matrix A_r is perturbed to $A_r + E$, then we have

$$|\lambda_i(A_r + E)| \leq \|A_r + E\| \leq \|A_r\| + \|E\|$$

Hence if $\|A_r\| < 1$, then the matrix A_r remains asymptotically stable under reasonably small perturbations. This property is only found in discrete time balanced canonical forms. For canonical forms such as the controllable canonical forms, small perturbations in some of the parameters may cause the system to become unstable [79].

We shall now state the properties of Kung's n^{th} $(n < r)$ order model.

Theorem 11.4 *(Properties of the n^{th} $(n < r)$ order estimate) The n^{th} order state space model (A_n, B_n, C_n, D_n) is always asymptotically stable and if $\sigma_n > \sigma_{n+1}$ then the realization is both controllable and observable. The spectral norm of A_n ($\|A_n\|$) is always less than or equal to unity. In the case when the singular values $\sigma_1, \ldots, \sigma_n$ are all distinct, $\|A_n\|$ is strictly less than unity. Let $G_r(z) = C_r(zI - A_r)^{-1}B_r + D_r$ and $G_n(z) = C_n(zI - A_n)^{-1}B_n + D_n$, then $\|G_r(z) - G_n(z)\|_\infty$ is bounded from above by twice the sum of the last $(r - n)$ singular values with multiple singular values counted only once.*

Since the n^{th} order Kung's model is identical to that obtained from truncating the last (r-n) states of the r^{th} order model which is stable, minimal and balanced. Hence the results on stability, minimality and error bounds follow from the results on discrete time balanced truncation, see lemmas 10.3 and 10.4.

Since the error between the full order model, which is an exact realization, and the reduced order model is bounded from above by the sum of the discarded singular values, it seems that the number of significant singular values is a rough indication of what the appropriate model order is.

11.1.3 Scaling of signals

In industrial practice, the measured input and output signals usually do not have the same order of magnitude and in fact they may differ by several orders of magnitudes. As an example, the tube glass production process described in [6] has two important outputs, the tube diameter and the wall thickness, and they may differ by a factor of 50 in magnitude. Our experience of using Kung's algorithm shows that this difference in order of magnitude does affect the quality of the model produced. In this section, we shall suggest a method to scale the input/output data so that a better model can be obtained. Before that, we first define a few terms which we are going to use in this section.

For the discussion to follow, we shall refer to the connection between the i^{th} output and the j^{th} input to as the (i,j) input-output channel. So, for a system with m inputs and p outputs, there are altogether mp of these input-output channels. Let $(J_0, J_1, ..., J_N)$ be the given sequence of Markov parameters and J_k^{ij} be the (i,j) element of J_k. Define the sum

$$e_{ij} = \sum_{k=1}^{N} (J_k^{ij})^2 \tag{11.23}$$

for all $i = 1, \ldots, p$, $j = 1, \ldots, m$. Note that in the above definition, J_0 is ignored. The quantity e_{ij} will be referred to as the energy content of the (i,j) input-output channel. If we judge the quality of fit at each input-output channel by comparing how well the model output matches that of the process output, then our experience of using Kung's algorithm is that those input-output channels whose energy contents are higher are usually better fitted. A plausible explanation to this observation is that these input-output channels which contain a higher amount of energy (as measured in the l_2 sense) receive comparatively more 'attention' from the Kung's algorithm which is equivalent to discrete time balanced truncation. (Note that this argument is similar to the one used by Moore in [46] to derive balanced realizations.)

In order to produce a model which better fits the original data, we suggest that one should first scale the data such that the amount of energy in each of the input-output channels is of a similar order of magnitude. If we pre-multiply each Markov parameter by the diagonal matrix $D_o = diag(d_o^1, d_o^2, ..., d_o^p)$ ('o' for output) and post-multiply each one of them by $D_i = diag(d_i^1, d_i^2, ..., d_i^m)$ ('i' for input), then the energy content of the scaled (i,j) input-output channel will be given by

$$\tilde{e}_{ij} = \sum_{k=1}^{N} (d_o^i J_k^{ij} d_i^j)^2 \tag{11.24}$$

We shall choose the diagonal matrices D_o and D_i such that all the $\tilde{e}_{ij}$ are of a similar order of magnitude. In appendix B we shall present a procedure to find these two matrices given a Markov parameter sequence. Assume that both D_o and D_i have been found and that we supply to Kung's algorithm the scaled Markov parameter sequence $(\tilde{J}_0, \tilde{J}_1, ..., \tilde{J}_N)$, where

$$\tilde{J}_k = D_o J_k D_i \quad \forall\, k = 0, 1, ..., N. \tag{11.25}$$

If Kung's algorithm gives the following state space realization

$$\left\{ \begin{array}{rcl} x[k+1] & = & \tilde{A}x[k] + \tilde{B}u[k] \\ y[k] & = & \tilde{C}x[k] + \tilde{D}u[k] \end{array} \right. \tag{11.26}$$

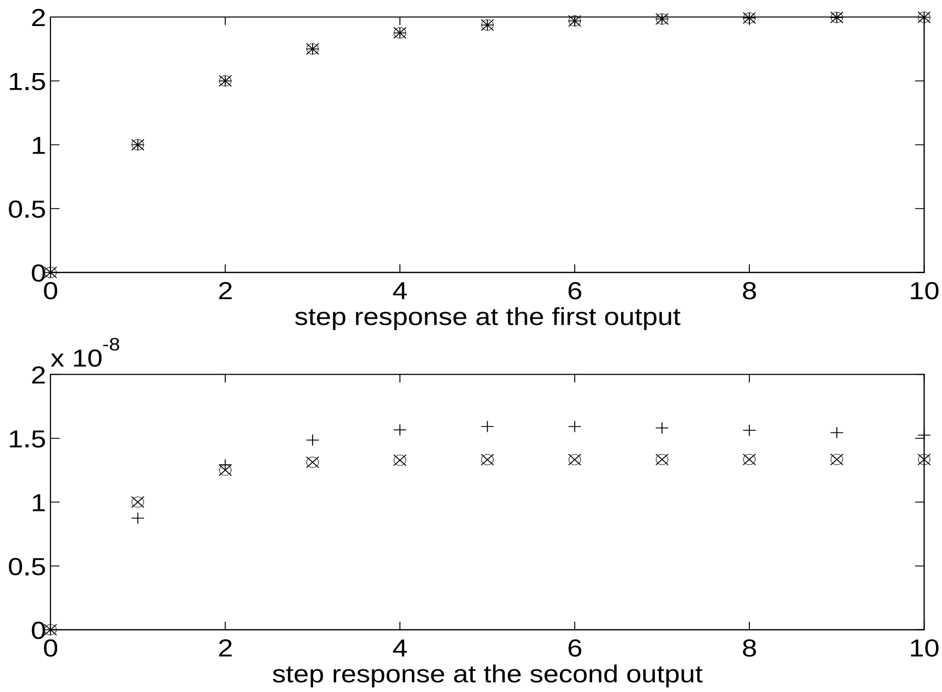


Figure 11.1: Step responses of the simulation model ('o'), 2nd order Kung's model without scaling ('+') and 2nd order Kung's model with scaling ('x')

then the correct unscaled model is given by

$$\left\{ \begin{array}{rcl} x[k+1] & = & Ax[k] + Bu[k] \\ y[k] & = & Cx[k] + Du[k] \end{array} \right. \tag{11.27}$$

where

$$\begin{aligned} A &= \tilde{A} && (11.28) \\ B &= \tilde{B}D_i^{-1} && (11.29) \\ C &= D_o^{-1}\tilde{C} && (11.30) \\ D &= D_o^{-1}\tilde{D}D_i^{-1} && (11.31) \end{aligned}$$

11.1.4 Simulation example

We shall give a simple simulation example in this section to show how scaling improves the fit of the input-output channels. The data are given by the first 11 Markov parameters of the following 2-output 1-input system whose state space matrices are

$$A = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.25 \end{bmatrix} B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} C = \begin{bmatrix} 1 & 0 \\ 0 & e \end{bmatrix} D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{11.32}$$

where $e = 10^{-8}$. The energy contents in the two input-output channels are

$$\begin{bmatrix} 1.15 \\ 1.03 \times 10^{-8} \end{bmatrix} \tag{11.33}$$

and the following output-scaling is applied to the Markov parameters

$$D_o = diag(1, 1.12 \times 10^8) \tag{11.34}$$

so that the resulting energy content in each of the channels is unity. In figure 11.1, we compare the unit step responses, up to 11 data-points, of the following three models.

1. The simulation model ('o').
2. The 2nd order model from Kung's algorithm without scaling of data ('+'). The state space matrices of this model after transforming into modal form are

$$A = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.92 \end{bmatrix} B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} C = \begin{bmatrix} 1 & 0 \\ 0 & -4.36 \times 10^{-10} \end{bmatrix} D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{11.35}$$

3. The 2nd order model from Kung's algorithm with the above scaling factor ('x'). The state space matrices of this model after transforming into modal form are exactly the same as those given in equation 11.32.

It can be seen from the step response plots and the state space matrices of the models that scaling does improve the quality of the estimates. The above example shows that a possible reason why scaling improves the accuracy of the model is that the round-off error is reduced by the suggested type of scaling. However we do not yet have a concrete proof of this proposition.

11.2 Model Reduction Method due to Al-Saggaf

In this section, we shall discuss how the model reduction algorithm suggested by Al-Saggaf [1, 2] can be used to obtain an initial state space model given a set of Markov parameter data. The algorithm of Al-Saggaf is also based on balanced truncation. The idea is to first transform the discrete time balanced model to a continuous time balanced model, truncate the model in continuous time and then transform the continuous time reduced order model back to discrete time. The final result is a reduced order model which is balanced as opposed to Kung's algorithm which does not give a balanced model. Another important feature of this method is that by using a certain bilinear transformation, we are able to match the steady state gain of the original model with that of the reduced order model.

This section is further divided into three sub-sections. Section 11.2.1 gives an outline of the algorithm. We shall discuss the properties of the model in section 11.2.2. Finally a simulation example will be given in section 11.2.3

11.2.1 Outline of the algorithm

Assume that we are given the Markov parameter sequence $\{J_0, J_1, ..., J_N\}$. We shall show in this section how a state space model which approximates this sequence can be obtained. The steps to obtain a model are

1. Use Kung's algorithm to obtain the full order discrete time minimal and balanced state space model (A_d, B_d, C_d, D_d) which exactly realizes the finite impulse response with Markov parameters $\{J_0, J_1, ..., J_N\}$. (This is the model (A_r, B_r, C_r, D_r) in section 11.1.2.)

2. Transform the discrete time balanced model (A_d, B_d, C_d, D_d) whose Gramian is Σ to the continuous time balanced model (A_c, B_c, C_c, D_c) by using the following bilinear transformation which preserves balancedness.

$$z = \alpha\frac{\beta - s}{\beta + s} \tag{11.36}$$

   where $\alpha = e^{j\theta}$ for $\theta \in [0, 2\pi)$, and β is any positive real number which is not an eigenvalue of A_c. In terms of state space matrices, we have

$$\begin{aligned} A_c &= \beta(\alpha I + A_d)^{-1}(A_d - \alpha I) && (11.37)\\ B_c &= \sqrt{2\alpha\beta}(\alpha I + A_d)^{-1}B_d && (11.38)\\ C_c &= \sqrt{2\alpha\beta}C_d(\alpha I + A_d)^{-1} && (11.39)\\ D_c &= D_d - C_d(\alpha I + A_d)^{-1}B_d && (11.40) \end{aligned}$$

   Note that the Gramian of the continuous time balanced model is also Σ.

3. Truncate the model (A_c, B_c, C_c, D_c) to the n^{th} order. The reduced order model $(A_{cn}$, B_{cn}, $C_{cn}, D_{cn})$ is continuous time balanced with Gramian Σ_1 where Σ_1 is the diagonal matrix containing the first n singular values of Σ.

4. Transform the continuous time model $(A_{cn}, B_{cn}, C_{cn}, D_{cn})$ to the discrete time model $(A_{dn}, B_{dn}, C_{dn}, D_{dn})$ by the inverse transformation of equation 11.36, which is

$$s = \beta\frac{\alpha - z}{\alpha + z} \tag{11.41}$$

   In terms of state space matrices, this transformation is given by

$$\begin{aligned} A_{dn} &= \alpha(\beta I - A_{cn})^{-1}(\beta I + A_{cn}) && (11.42)\\ B_{dn} &= \sqrt{2\alpha\beta}(\beta I - A_{cn})^{-1}B_{cn} && (11.43)\\ C_{dn} &= \sqrt{2\alpha\beta}C_{cn}(\beta I - A_{cn})^{-1} && (11.44)\\ D_{dn} &= D_{cn} + C_{cn}(\beta I - A_{cn})^{-1}B_{cn} && (11.45) \end{aligned}$$

   Note that this transformation preserves balancedness and the Gramian of the reduced order model $(A_{dn}, B_{dn}, C_{dn}, D_{dn})$ is Σ_1.

The state space matrices $(A_{dn}, B_{dn}, C_{dn}, D_{dn})$ can be expressed in terms of sub-matrices inside (A_d, B_d, C_d, D_d). If we partition compatibly the full order discrete time balanced state space matrices and its Gramian as follows

$$A_d = \begin{bmatrix} A_{d11} & A_{d12} \\ A_{d21} & A_{d22} \end{bmatrix} \quad B_d = \begin{bmatrix} B_{d1} \\ B_{d2} \end{bmatrix} \quad C_d = \begin{bmatrix} C_{d1} & C_{d2} \end{bmatrix} \quad \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \tag{11.46}$$

where Σ_1 is an n by n matrix, then it can be shown that the reduced order model from the above scheme is given by

$$\begin{aligned} A_{dn} &= A_{d11} - A_{d12}(\alpha I + A_{d22})^{-1}A_{d21} && (11.47) \\ B_{dn} &= B_{d1} - A_{d12}(\alpha I + A_{d22})^{-1}B_{d2} && (11.48) \\ C_{dn} &= C_{d1} - C_{d2}(\alpha I + A_{d22})^{-1}A_{d21} && (11.49) \\ D_{dn} &= D_d - C_{d2}(\alpha I + A_{d22})^{-1}B_{d2} && (11.50) \end{aligned}$$

Note that the matrix $(\alpha I + A_{d22})$ is always invertible due to the fact that A_{d22} is discrete time asymptotically stable, see [60].

11.2.2 Properties of the models

In this section, we shall summarise the properties of the reduced order model $(A_{dn}, B_{dn}, C_{dn}, D_{dn})$. Since we are only interested in real state space models, we shall only consider the case where α takes the value of either $+1$ or -1.

1. If Σ_1 and Σ_2 have no diagonal entries in common, then the reduced order model is asymptotically stable, minimal and balanced with gramian Σ_1.

2. The spectral norm of A_{dn} ($\|A_{dn}\|$) is less than or equal to 1. If all the diagonal entries in Σ_1 are distinct, then $\|A_{dn}\| < 1$. In other words, A_{dn} is robustly stable.

3. Define

$$\begin{aligned} G_d(z) &= D_d + C_d(zI - A_d)^{-1}C_d && (11.51) \\ G_{dn}(z) &= D_{dn} + C_{dn}(zI - A_{dn})^{-1}C_{dn} && (11.52) \end{aligned}$$

   then

   (a) $(G_d(z) - G_{dn}(z))|_{z=-\alpha} = 0$

   (b) $\|G_d(z) - G_{dn}(z)\|_\infty \leq 2\mathrm{tr}(\Sigma_2)$

Note that property 1 comes from results on continuous time balanced truncation (see lemma 10.7) and property 2 is due to the fact that the reduced order model is discrete time balanced.

The distinctive feature of this model reduction scheme is that of property 3a. This property can be proved by noticing the fact that the point $z = -\alpha$ is mapped to the point $s = \infty$ under the transformation and the fact that continuous time balanced truncation preserves the value of the transfer function at $s = \infty$. There are two choices for the value of α. For $\alpha = 1$, this corresponds to the bilinear transformation given in section 5.4. Our experience shows that the reduced order model obtained from this choice of α is similar to that obtained from Kung's algorithm. However, for the choice of $\alpha = -1$, the steady state gain of the reduced order model ($G_{dn}(1)$) is equal to that of the full order model ($G_d(1)$). Hence this method of finding an initial model is best when the steady state gain of the reduced order model is important.

Finally, property 3b tells us that we should again choose the model order based on the number of significant singular values. This property and all the others were proved in [1, 2].

In section 11.1.3, we discussed that when the outputs of the plants are of very different order of magnitude, we should scale the data such that the energy in each input-output channel is of a similar order of magnitude before we supply the data to Kung's algorithm. Since Al-Saggaf's algorithm is also based on balanced truncation, we shall also advise that scaling of data be done, if necessary.

11.2.3 Simulation example

In this section, we shall give a simulation example whose aim is to compare the following 3 algorithms

1. Kung's algorithm,
2. Al-Saggaf's algorithm with $\alpha = 1$, and
3. Al-Saggaf's algorithm with $\alpha = -1$.

The simulation model is a random discrete time stable system generated by MATLAB. It has 2 inputs, 2 outputs and 30 states. A unit step is applied to the two inputs in turns and the corresponding outputs are measured. The data are obtained from corrupting the measured outputs with white noise and are plotted in figure 11.2. We then supply the data to these three algorithms and we obtain a 4^{th} order state space model from each one of the them. The simulated outputs of these three models are plotted in figures 11.3, 11.4 and 11.5. The results from these plots suggest that the model obtained from Kung's algorithm is similar to that obtained from Al-Saggaf's algorithm with $\alpha = 1$. On the other hand, Al-Saggaf's algorithm with $\alpha = -1$ gives an excellent fit to the data.

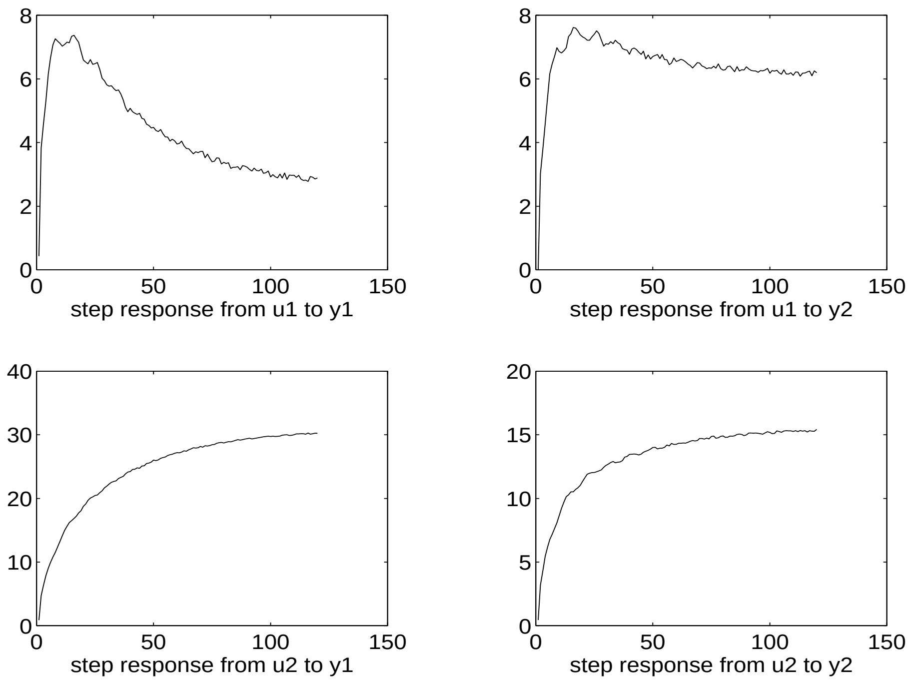


Figure 11.2: Step responses of the simulation model

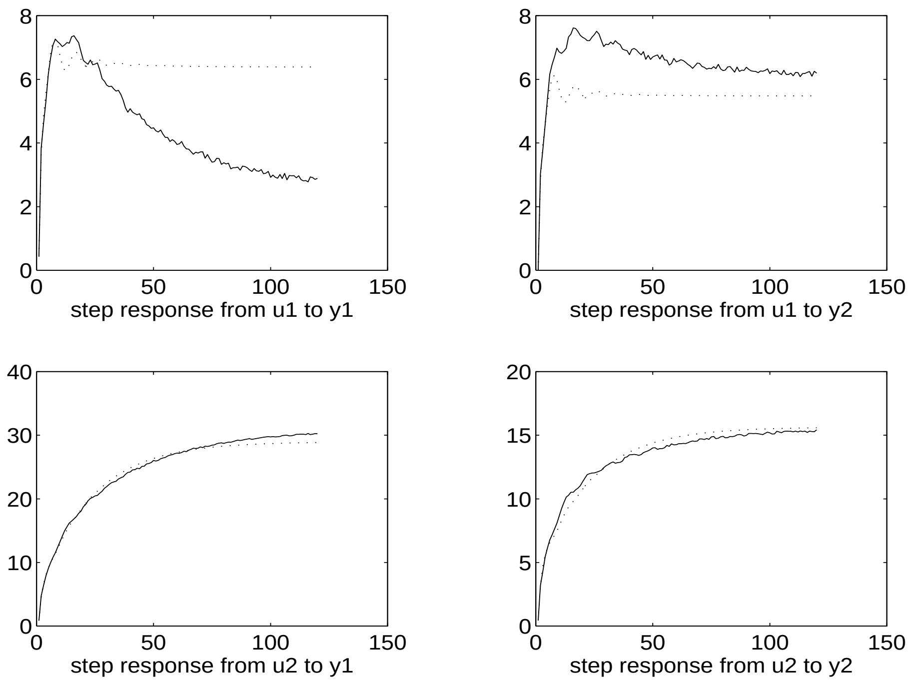


Figure 11.3: Step responses of the simulation model (solid) and Kung's 4^{th} order model (dotted)

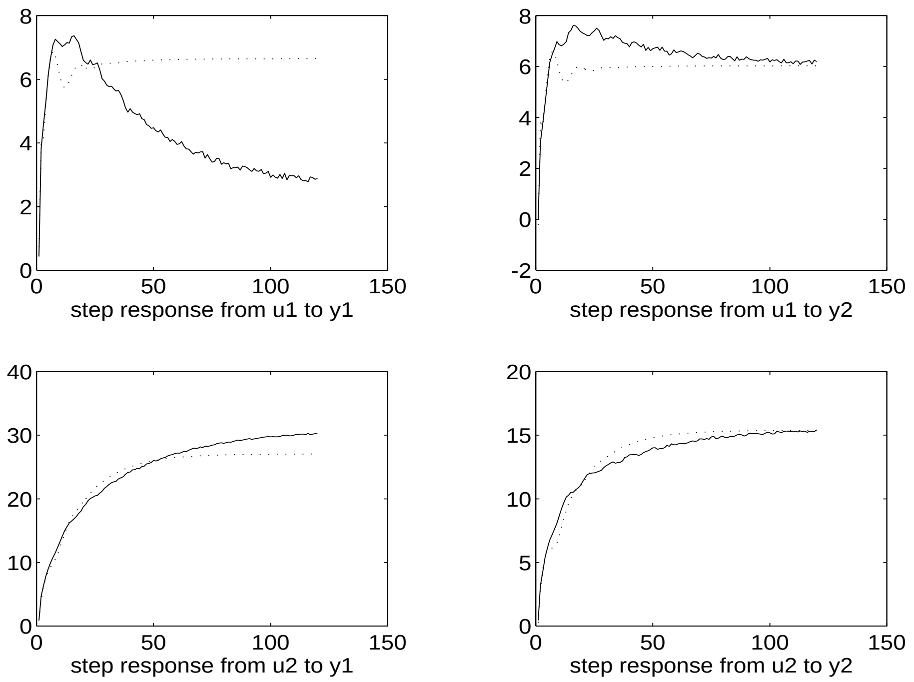


Figure 11.4: Step responses of the simulation model (solid) and Al-Saggaf 4^{th} order model with $\alpha = 1$ (dotted)

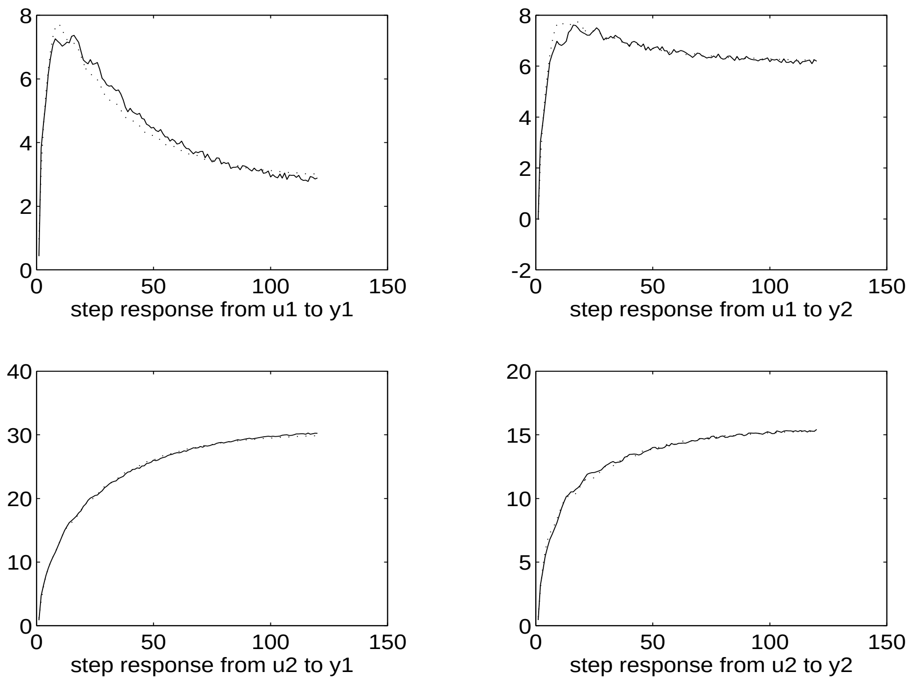


Figure 11.5: Step responses of the simulation model (solid) and Al-Saggaf 4^{th} order model with $\alpha = -1$ (dotted)

11.3 N4SID:Numerical Algorithm for Subspace State Space System Identification

In this section, we shall briefly describe the subspace algorithm N4SID derived by van Overschee and De Moor [71]. We shall only give the essential features of the algorithm here as the derivation for the whole algorithm is rather involved. The aim of the N4SID algorithm is to solve the combined deterministic-stochastic identification problem, which can be stated as follows:

Given an input signal sequence $u_1, u_2, \ldots, u_N$ and an output signal sequence $y_1, y_2, \ldots, y_N$ of a dynamical process, find the following combined deterministic-stochastic state space system which approximates the data.

$$\left\{ \begin{array}{rcl} x_{k+1} & = & Ax_k + Bu_k + w_k \\ y_k & = & Cx_k + Du_k + v_k \end{array} \right. \tag{11.53}$$

with

$$E\left[\begin{pmatrix} w_k \\ v_k \end{pmatrix} \begin{pmatrix} w_k & v_k \end{pmatrix} \right] \; = \; \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{kl} \; \geq \; 0 \tag{11.54}$$

where E denotes the expected value operator, δ_{kl} the Kronecker index, and both v_k and w_k are Gaussian distributed, zero mean, white noise vector sequences. In other words, the N4SID algorithm finds the state space matrices A, B, C and D and also the covariance matrices Q, R and S. This algorithm assumes that the stochastic part of the output is uncorrelated with both the deterministic input and the deterministic part of the state. It further assumes that the input is persistently exciting. See [71] for details of the assumptions made.

The outline of this algorithm is very similar to the subspace algorithm described in section 9.4. The main idea is to divide the input/output data into two parts. The first part of the data will be referred to as the past while the second part as the future. Similar to the algorithm described before, we form block Hankel matrices using the past as well as the future data. By projecting the space spanned by an appropriate matrix of the future output onto the space generated by the past data, we are able to obtain a matrix $\hat{\Gamma}_i$ whose column space coincides with the column space of the extended observability matrix of the system:

$$\Gamma_i \;=\; \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^i \end{bmatrix} \tag{11.55}$$

where i is an integer greater than the order of the system. Hence the order of the system can be determined by the dimension of the column span of $\hat{\Gamma}_i$ or rather an approximate dimension of the state space is given by the number of significant singular values of $\hat{\Gamma}_i$. Again by finding the appropriate projections, we are able to estimate the past and future state sequences $\hat{X}_i$ and $\hat{X}_{i+1}$ respectively. Then we shall set up a linear equation similar to equation 9.4, as follows

$$\begin{bmatrix} \hat{X}_{i+1} \\ Y_i \end{bmatrix} = \begin{bmatrix} A & \bar{B} \\ C & \bar{D} \end{bmatrix} \begin{bmatrix} \hat{X}_i \\ U_i \end{bmatrix} + \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \tag{11.56}$$

where U_i and Y_i are some appropriate Hankel matrices formed by the input and output data respectively. The deterministic state space matrices A and C are given by the least squares solution to the above linear equation. The matrices B and D can be found from $(A, \bar{B}, C, \bar{D})$ via a set of linear equations. Finally, the covariance matrices are calculated from

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = \frac{1}{j} \begin{bmatrix} \rho_1\rho_1^T & \rho_1\rho_2^T \\ \rho_2\rho_1^T & \rho_2\rho_2^T \end{bmatrix} \tag{11.57}$$

where j is the number of block columns in the Hankel matrices of input and output data.

It can be shown that, if $j \to \infty$ then the algorithm described above gives an unbiased estimate of the state space and covariance matrices. Another N4SID algorithm was described in [71] and it gives a biased estimate but the amount of computation required is less than the one described above. A major difference between these two N4SID algorithms and the other subspace algorithms is that they determine the system matrices and covariance matrices at the same time. For the other subspace algorithms, the deterministic system matrices are determined first and then the covariance matrices are obtained from a stochastic realization of the residuals.

The N4SID algorithms also share a lot of common features with the other subspace algorithms. The two N4SID algorithms are both implemented in a numerical robust and efficient way using reliable numerical tools such as the QR factorization and singular value decomposition. For these two algorithms, the model order is determined by the number of significant singular values of a matrix which approximates the state space. Unfortunately, the stability and minimality of the models cannot be guaranteed.

11.4 Summary

In this chapter, we have looked at three different algorithms for computing state space models from input/output data. Although both Kung's and Al-Saggaf's algorithms are related to the balanced model reduction technique, both of them can be interpreted as subspace methods in the sense that they both approximate the higher order state space by a lower order one. A common feature among these three algorithms is that the model order is determined from the number of significant singular values of a certain matrix which approximates the extended observability matrix.

The main differences between these three methods are on the type of model which they produce and on the type of data which we supply to these algorithms. Both Kung's and Al-Saggaf's algorithms give an output-error type state state space model while N4SID gives a state space model with both process and measurement noise. In fact it is also possible to obtain a state space model with both process and measurement noise using Kung's or Al-Saggaf's algorithms but only in two steps. In the first step, we find a state space model with output error. We then find an approximate stochastic realization from the residuals using Kung's algorithm. By combining these two models, we then obtain a combined deterministic-stochastic model . A drawback of this method is that the combined deterministic-stochastic system is usually of a much higher order than is required. The fact that a combined deterministic-stochastic model can be obtained from N4SID in one go is clearly a merit of this method.

For both Kung's and Al-Saggaf's algorithms, we assume that the Markov parameters are available. Markov parameters can be determined from the input/output data if the input excitation used is a step or a PRBS. If a step test signal is used and the model is required to approximate the steady gain of the process well, it was shown in section 11.2 that Al-Saggaf's algorithm is a good choice. For the case of subspace methods, they can be used with an arbitrary input signal provided that it is persistently exciting. In the case where step data are available, we can first convert these data into Markov parameters to form a finite impulse response simulation model. We then apply an input signal (which is persistently exciting) to this simulation model and we record the corresponding output. The input/output data from this simulation model can then be supplied to a subspace algorithm. If the real process is linear time-invariant, the number of Markov parameters exceeds the process response time and the signal-to-noise ratio is high, then the output of this simulation model will be similar to that obtained from applying the same input to the real process. Since the subspace algorithms assume that the underlying process is linear time-invariant (although the real process is normally not), in principle they are still able to give an accurate model using the simulation model data if the other two assumptions are met. Since N4SID gives a combined deterministic-stochastic model, so it may seem justifiable to modify the data if this type of model is required but then the choice of step input is a poor one as it does not give accurate information at high frequencies. So, we shall conclude this chapter by reiterating what we have mentioned in section 9.1 — the decisions on selecting the type of model, the type of input excitation and the type of system identification algorithm are inter-dependent on each other.

Chapter 12

Parameter estimation

In this chapter, we shall discuss how the continuous time balanced canonical forms given in section 10.2 can be used in parameter estimation. This forms the second part of the system identification algorithm which was proposed in section 9.5. We shall first present a general algorithm for parameter estimation using canonical forms in section 12.1. A fundamental difficulty in using canonical forms for parameter estimation is the selection of suitable model structures. This will be the topic of discussion in section 12.2. In section 12.3 we shall present a parameter estimation algorithm based on the generic structure of balanced parametrizations. Since an explicit parametrization is available, the efficiency of this parameter estimation algorithm can be improved by supplying it with an explicit formula for the gradient of the objective function. We shall discuss the computation of gradients in section 12.4. Finally, we shall present a simulation example in section 12.5.

12.1 A general algorithm for parameter estimation

The parameter estimation method for system identification was discussed in section 9.2. In this section, we shall present a general algorithm for parameter estimation of linear state space models using canonical forms. The problem of parameter estimation can be stated as follows. Given

1. The model set $\mathcal{M}$ which is the set of all admissible state space models.
2. A canonical form (U_i, ρ_i) $(i \in \{1, \dots, l\})$ of the model set $\mathcal{M}$ where

$$\begin{array}{rccc} \rho_i: & U_i \subset \mathbb{R}^{N_i} & \to & \mathcal{M} \\ & \theta & \mapsto & S(\theta) = (A(\theta), B(\theta), C(\theta), D(\theta)) \end{array}$$

   Note that $\{1, \dots, l\}$ is an index set and each index corresponds to a particular set of structural parameters. Hence l is the number of different structures in this state space canonical form. Since (U_i, ρ_i) is a canonical form, the image spaces of the maps $\rho_i(U_i)$ are disjoint subsets of $\mathcal{M}$.

3. An objective function $V(\theta)$.

The aim of parameter estimation is to find the best model from $\mathcal{M}$ which is the solution to the following optimization problem.

$$\theta_{opt} = \arg \min_{i \in \{1,\ldots,l\}} \min_{\theta \in U_i} V(\theta) \tag{12.1}$$

Since the image spaces $\rho_i(U_i)$ are disjoint subsets of $\mathcal{M}$, a possible way to solve this optimization problem is to find the solutions of the following l optimization problems

$$\theta_{opt,i} = \arg \min_{\theta \in U_i} V(\theta) \tag{12.2}$$

and then the true solution is

$$\theta_{opt} = \arg \min_{i \in \{1,\ldots,l\}} V(\theta_{opt,i}) \tag{12.3}$$

Although the above procedure gives the true minimum (if it exists and if the global minimum of each of the l optimization problems can be found), it involves searching through all the possible structures. Since the number of structures of many state space canonical forms increases rapidly as the model order increases, the above algorithm can be computationally rather burdensome. However, since the aim of system identification is to obtain a model which *adequately* describes a process, we suggest the following algorithm which only partially solves the parameter estimation problem.

1. Choose an index i which corresponds to a particular set of structural parameters.
2. Define the parameter vector θ and choose an initial estimate θ_0.
3. Use an optimization algorithm to find the solution to the minimization problem

$$\theta_{opt,i} = \arg \min_{\theta \in U_i} V(\theta)$$

4. If $\theta_{opt,i}$ is a satisfactory model, stop. Otherwise, if the optimization problem fails to find a minimum or if the model $\theta_{opt,i}$ is not satisfactory, then choose a different index i and return to step 2.

So instead of scanning through all the possible structures of the canonical form to find a minimum, the above algorithm suggests that we take the first satisfactory solution from all the structures which we have scanned through. Hence, the efficiency of the algorithm depends very much on the choice of the first index i as well as how we choose a different structure if it is needed. The selection of model structure will be the topic of discussion in the next section. Finally, note that the above algorithm can also be used to solve the parameter estimation problem if (U_i, ρ_i) is an overlapping parametrization rather than a canonical form; however we shall not discuss this further.

12.2 Model structure selection

In this section, we shall discuss two issues related to the topic of model structure selection which forms one of the basic components in the algorithm outlined above. The first issue is how to select an initial model structure, and the second one is how to choose a different model structure when it is required. The problem of model structure selection is itself a difficult topic. In fact the difficulty in selecting a model structure from input/output data is very often cited as an objection to the use of canonical forms in system identification. An added difficulty to the model structure selection problem is the vast number of structures there are to choose from. This is especially a problem for the balanced canonical form given in section 10.2.2 as the number of model structures in this canonical form far exceeds those of other known canonical forms. As an example, $2 \times 3^{n-1}$ different model structures are required to parametrize the set of all stable single-input single-output (SISO) LTI systems of McMillan degree n but only one structure is required to parametrize the same set of systems if either the controllable or observable canonical form is used. We must say that we do not yet have a satisfactory answer to the model structure selection problem and much work remains to be done. We shall first discuss the issue of selecting an initial model structure.

At the moment, we do not yet have a solution to the problem of selecting an initial model structure of the balanced canonical form based on input/output data. In the absence of a complete solution, we shall propose two partial solutions here. The first solution is to use the generic structure of balanced canonical form as the initial model structure. The generic structure of balanced canonical form was defined in section 10.2.2 and it refers to the structure which corresponds to distinct Hankel singular values. We also saw in section 10.2.4 that the parametrization for the generic structure is much simpler than the others and that the number of real-valued parameters required is always $n(m+p)+mp$. However, in the case of balanced canonical form for scalar systems of order n, even after the generic structure has been chosen, there are still n other structural parameters which are yet to be determined! In the case of the generic structure of the multivariable balanced canonical form, the situation depends on how we parametrize the unit vectors U_i (see section 10.2.4 for the definition of U_i). If we directly parametrize U_i, then no more structural decisions need to be made. However, when we supply the parametrization to an optimization algorithm, this will introduce a number of non-linear equality constraints. The situation is not much better if we parametrize U_i according to lemma 10.11, then we need to make decisions on the signs and there are n of them. Despite all these drawbacks, the generic structure is a good choice as most of the systems can be represented by this structure.

Another partial solution to the problem is to use the algorithms given in chapter 11 to find an initial state space model from input/output data and then choose a model structure based on this initial estimate. In practice, since we only have a finite set of data and the data are corrupted by noise, the Hankel singular values of these initial models are usually distinct i.e. these initial estimates are usually of the generic structure. However, these initial estimates may indicate that some Hankel singular values are close to each other and that the data could have been generated by a non-generic system. Our experience on simulation exercises shows that if the input/output data have in fact come from a non-generic system and if the signal-to-noise ratio is high, then these algorithms do indicate that some singular values are close to each other. As a final remark on the issue of choosing an initial model structure, in the case of time-series identification using balanced canonical form of SISO stable and minimum-phase systems, there are n (= model order) sign decisions to be made and McGinnie [43] has shown that in principle it is possible to obtain consistent estimates of these sign parameters from the data.

On the issue of how to choose a new model structure, we again do not yet have a solution to this problem. However, our experience on using the generic structure as an initial structure shows that very often a satisfactory model can be found within this structure. In section 12.5, we shall present a simulation example with data generated by a linear system of order 10 with 3 sets of identical Hankel singular values and the optimization results show that the original system can be approximated by generic systems satisfactorily. An appropriate question to ask at this point is whether the optimized model shows that some Hankel singular values are similar to each other. Unfortunately, the answer to this question is not affirmative. For the simulation example in section 12.5, some singular values of the optimized model are very similar to each other. However, our experience also shows that some non-generic systems can be approximated by generic ones with very different singular values.

Due to the fact that we do not have a solution to the model structure selection problem, we have implemented an algorithm based on the generic structure only. This algorithm will be presented in the next section.

12.3 Algorithm for parameter estimation using balanced canonical forms

In this section, we shall present a parameter estimation algorithm for MIMO systems based on the generic structure of balanced canonical forms. The parametrization given in lemma 10.12 will be used, but the unit vectors will be parametrized using the local coordinate system given in section 10.2.5 rather than that stated in the lemma. For this parametrization, we shall further assume that B_j and D are parametrized as follows.

1. B_j is parametrized as $\begin{bmatrix} b_{j1} & \dots & b_{jm} \end{bmatrix}$
2. D is given by $\begin{bmatrix} d_{11} & \dots & d_{1m} \\ \vdots & \ddots & \vdots \\ d_{p1} & \dots & d_{pm} \end{bmatrix}$

We shall define the parameter vector θ as

$$\theta = \begin{bmatrix} s & b & \phi & d \end{bmatrix} \tag{12.4}$$

where

$$\begin{aligned} s &= \begin{bmatrix} s_1 & s_2 & \dots & s_{n-1} & s_n \end{bmatrix} && (12.5)\\ &= \begin{bmatrix} \sigma_1 - \sigma_2 & \sigma_2 - \sigma_3 & \dots & \sigma_{n-1} - \sigma_n & \sigma_n \end{bmatrix} && (12.6)\\ b &= \begin{bmatrix} b_{11} & \dots & b_{1m} & b_{21} & \dots & b_{2m} & \dots & b_{n1} & \dots & b_{nm} \end{bmatrix} && (12.7)\\ \phi &= \begin{bmatrix} \phi_{1,1} & \dots & \phi_{1,p-1} & \phi_{2,1} & \dots & \phi_{2,p-1} & \dots & \phi_{n,1} & \dots & \phi_{n,p-1} \end{bmatrix} && (12.8)\\ d &= \begin{bmatrix} d_{11} & \dots & d_{p1} & d_{12} & \dots & d_{p2} & \dots & d_{1m} & \dots & d_{pm} \end{bmatrix} && (12.9) \end{aligned}$$

In order that the above parameter vector represents a stable and minimal state space system with distinct Hankel singular values, the following linear constraints (or bounds) on the parameters are required.

1. $s_i > 0$ for $i = 1, ..., n$
2. $b_{i1} > 0$ for $i = 1, ..., n$
3. $\phi_{i,j} \in (-\pi/2, \pi/2)$ for $i = 1, ..., n$ and $j = 1, ..., p-1$

The parameter estimation problem can now be stated as the following constrained optimization problem

$$\min_{\theta} V(\theta) \text{ subject to the constraints on } \theta \text{ given above} \tag{12.10}$$

which can be solved by various types of gradient type optimization algorithm. In order to improve the efficiency of the optimization procedure, we shall supply to the optimization algorithm explicit formulae for gradients. The computation of gradients will be the topic of discussion in the next section.

12.4 Gradient calculations

One of the basic components of our parameter estimation algorithm is an optimization algorithm. Since the optimization problem which we have posed is a non-linear one, a faster rate of convergence can be obtained if we have a good initial estimate. Another way to improve the efficiency of the optimization algorithm is to exploit the fact that explicit formulae for the gradient of the objective function can be obtained beforehand, provided that the objective function is sufficiently smooth. Another advantage of using formulae for the gradient is that they improve the accuracy of the calculations.

In this section, we shall discuss how formulae for the gradient of the objective function can be calculated. We shall first give an overview of the problem in section 12.4.1. Basically, the problem of finding the gradients is one of calculating the derivatives of the state space matrices. The formulae for the derivatives of the state space matrices parametrized by balanced realizations will be stated in section 12.4.2. In section 12.4.3 we shall give a few examples of gradient calculations using different objective functions. Finally, we shall summarise our discussion on computing the gradient in section 12.4.4.

12.4.1 Overview

In this section, we shall give an overview of how the gradient of the objective function can be calculated. Although the parameter estimation algorithm which we have proposed is an off-line one, the following discussion on gradient computations is also applicable to on-line system identification algorithms using balanced canonical forms.

In order to calculate the gradient of the objective function V, which is a function of the discrete time state space matrices $(A_d(\theta), B_d(\theta), C_d(\theta), D_d(\theta))$ and the data, with respect to the parameter vector $\theta = [\theta_i]$, we basically need to find the derivatives $\frac{\partial(A_d(\theta))}{\partial\theta_i}$, $\frac{\partial(B_d(\theta))}{\partial\theta_i}$,

$\frac{\partial(C_d(\theta))}{\partial\theta_i}$ and $\frac{\partial(D_d(\theta))}{\partial\theta_i}$ for all θ_i in θ. Since balanced parametrizations are given for continuous time stable systems only, we first need to relate the discrete time state space matrices to their corresponding continuous time state space matrices $(A_c(\theta), B_c(\theta), C_c(\theta), D_c(\theta))$ by the bilinear transform, which preserves balancedness [20]. The formulae for the bilinear transformation have already been stated in section 5.4, however we shall repeat them here for ease of reference.

$$
\begin{aligned}
A_d(\theta) &= (I + A_c(\theta))(I - A_c(\theta))^{-1} && (12.11)\\
B_d(\theta) &= \sqrt{2}(I - A_c(\theta))^{-1}B_c(\theta) && (12.12)\\
C_d(\theta) &= \sqrt{2}C_c(\theta)(I - A_c(\theta))^{-1} && (12.13)\\
D_d(\theta) &= D_c(\theta) + C_c(\theta)(I - A_c(\theta))^{-1}B_c(\theta) && (12.14)
\end{aligned}
$$

By using the above relationships, we can relate the derivatives of the discrete time state space matrices to the derivatives of the continuous time state space matrices, as follows.

$$
\begin{aligned}
\frac{\partial(A_d(\theta))}{\partial\theta_i} &= (I + A_d(\theta))\frac{\partial(A_c(\theta))}{\partial\theta_i}(I - A_c(\theta))^{-1} && (12.15)\\
\frac{\partial(B_d(\theta))}{\partial\theta_i} &= (I - A_c(\theta))^{-1}\frac{\partial(A_c(\theta))}{\partial\theta_i}B_d(\theta) + \sqrt{2}(I - A_c(\theta))^{-1}\frac{\partial(B_c(\theta))}{\partial\theta_i} && (12.16)\\
\frac{\partial(C_d(\theta))}{\partial\theta_i} &= \sqrt{2}\frac{\partial(C_c(\theta))}{\partial\theta_i}(I - A_c(\theta))^{-1} + C_d(\theta)\frac{\partial(A_c(\theta))}{\partial\theta_i}(I - A_c(\theta))^{-1} && (12.17)\\
\frac{\partial(A_d(\theta))}{\partial\theta_i} &= \frac{\partial(D_c(\theta))}{\partial\theta_i} + \frac{1}{\sqrt{2}}\frac{\partial(C_c(\theta))}{\partial\theta_i}B_d(\theta) + \frac{1}{2}C_d(\theta)\frac{\partial(A_c(\theta))}{\partial\theta_i}B_d(\theta) + \\
&\quad \frac{1}{\sqrt{2}}C_d(\theta)\frac{\partial(B_c(\theta))}{\partial\theta_i} && (12.18)
\end{aligned}
$$

The formulae for the derivatives of SISO continuous time balanced state space matrices were given in [42]. The major difference between the SISO case and MIMO case is that in the MIMO case, closed form formulae for the derivatives are not always available; we may need to solve a few sets of linear equations and a few matrix Lyapunov equations in order to obtain the derivatives. The number and type of equations which we need to solve also depend on how we parametrize the matrix parameters (see section 10.2.2). In general, in the MIMO case, closed form formulae are available when all the Hankel singular values are distinct i.e. in the generic case.

Here we shall give two examples to illustrate the differences between finding derivatives in the generic case and in the non-generic case. In both examples, a continuous time system with 2 inputs, 2 outputs and 2 states will be considered. The system in example 1 has 2 identical Hankel singular values while the system in example 2 has 2 distinct Hankel singular values. Also, in both examples, the matrix parameter U will be parametrized in two different ways to show the effect of the choice of parametrization on gradient calculations.

Example 1

In this example, we shall show how we can calculate the derivatives of the balanced state space matrices of a system with repeated Hankel singular values. The matrix parameter U will be parametrized in two different ways.

Case 1

In this case, the system is parametrized by the parameter vector $\theta = [\sigma, b_{11}, b_{12}, b_{22}, u_{11}, a]$ where $\sigma, b_{11}, b_{22} > 0$, the state space matrices are given by

$$\begin{aligned}
B &= \begin{bmatrix} b_{11} & b_{12} \\ 0 & b_{22} \end{bmatrix} \\
C &= U\Delta \text{ where } U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \text{ with } U^T U = I_2, \text{ and } \Delta = (BB^T)^{1/2} \\
A &= \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix} - \frac{1}{2\sigma}[BB^T]_d - \frac{1}{\sigma}[BB^T]_l
\end{aligned}$$

where the notation $[\ ]_d$ and $[\ ]_l$ was defined in section 10.2.1.

The way in which the parameters enter the A and B matrices is rather straightforward, so finding the derivatives of these two matrices with respect to the parameters is a simple exercise in differentiation. However this is not true for the C matrix, as $C = U\Delta$ where U is an orthogonal matrix and Δ is the positive definite square root of BB^T. To obtain the derivative $\frac{\partial U}{\partial u_{11}}$, we note that

$$\begin{aligned}
U^T U &= I_2 \\
\Rightarrow \left(\frac{\partial U}{\partial u_{11}}\right)^T U + U^T \frac{\partial U}{\partial u_{11}} &= 0
\end{aligned} \tag{12.19}$$

The last equation is in fact a set of three linear equations in $\frac{\partial u_{12}}{\partial u_{11}}$, $\frac{\partial u_{21}}{\partial u_{11}}$, and $\frac{\partial u_{22}}{\partial u_{11}}$. It can be rewritten as

$$\begin{bmatrix} 0 & u_{21} & 0 \\ u_{11} & u_{22} & u_{21} \\ u_{12} & 0 & u_{22} \end{bmatrix} \begin{bmatrix} \frac{\partial u_{12}}{\partial u_{11}} \\ \frac{\partial u_{21}}{\partial u_{11}} \\ \frac{\partial u_{22}}{\partial u_{11}} \end{bmatrix} = \begin{bmatrix} -u_{11} \\ -u_{12} \\ 0 \end{bmatrix} \tag{12.20}$$

Hence $\frac{\partial U}{\partial u_{11}}$ can be calculated by solving the above linear matrix equation. Note that there is a systematic procedure to transform equation 12.19 to the form of equation 12.20.

To find the derivative $\frac{\partial \Delta}{\partial \theta_i}$ where θ_i is either b_{11}, b_{12} or b_{22}, we first write

$$\begin{aligned}
\Delta^2 &= BB^T \\
\Rightarrow \Delta\frac{\partial \Delta}{\partial \theta_i} + \frac{\partial \Delta}{\partial \theta_i}\Delta &= \frac{\partial (BB^T)}{\partial \theta_i},
\end{aligned}$$

which is an easily-solved matrix Lyapunov equation for $\frac{\partial \Delta}{\partial \theta_i}$.

Case 2

In this case, we shall parametrize the system by a different parameter vector, which is $\theta = [\sigma, b_{11}, b_{12}, b_{22}, \phi, a]$ where $\sigma, b_{11}, b_{22} > 0$ and $\phi \in (-\pi, \pi]$. The way in which the parameters enter the A and B matrices is identical to that in the first case, so we still have to solve 3 matrix Lyapunov equations in order to find the derivatives of Δ. However, in this case, U is given either by

$$U_1(\phi) = \begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix}$$

or by

$$U_2(\phi) = \begin{bmatrix} cos(\phi) & sin(\phi) \\ sin(\phi) & -cos(\phi) \end{bmatrix}$$

Their derivatives with respect to ϕ are given respectively by

$$\frac{\partial U_1(\phi)}{\partial \phi} = \begin{bmatrix} -sin(\phi) & -cos(\phi) \\ cos(\phi) & -sin(\phi) \end{bmatrix} = U_1(\phi + \frac{\pi}{2})$$

and

$$\frac{\partial U_2(\phi)}{\partial \phi} = \begin{bmatrix} -sin(\phi) & cos(\phi) \\ cos(\phi) & sin(\phi) \end{bmatrix} = U_2(\phi + \frac{\pi}{2})$$

As a remark, if U is given by equation 10.22, i.e.

$$\begin{aligned} & U(\phi_{12}, \phi_{13}, \ldots, \phi_{ij}, \ldots, \phi_{p-1,p}) \\ = \; & R_{12}(\phi_{12}) R_{1,3}(\phi_{1,3}) \ldots R_{i,j}(\phi_{i,j}) \ldots R_{p-1,p}(\phi_{p-1,p}) K_p \end{aligned} \tag{12.21}$$

then its derivative with respect to ϕ_{ij} is

$$\frac{\partial U}{\partial \phi_{ij}} = R_{12}(\phi_{12}) R_{1,3}(\phi_{1,3}) \ldots R_{i,j}(\phi_{i,j} + \frac{\pi}{2}) \ldots R_{p-1,p}(\phi_{p-1,p}) K_p \tag{12.22}$$

□

Example 2

In this example, we shall show how the derivatives of the balanced state matrices of a system with distinct Hankel singular values can be calculated. Again, the matrix parameter U will be parametrized in two different ways.

Case 1

In this case, the system will be parametrized by $\theta = [\sigma_1, \sigma_2, b_{11}, b_{12}, b_{21}, b_{22}, u_{11}, u_{12}]$ where $\sigma_1 > \sigma_2 > 0$. The state space matrices are given by

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \text{ where } B_i = \begin{bmatrix} b_{i1} & b_{i2} \end{bmatrix} \text{ for } i = 1, 2$$

$$C = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \text{ where for } i = 1, 2,\ C_i = U_i \Delta_i, U_i = \begin{bmatrix} u_{1i} \\ u_{2i} \end{bmatrix}$$

$$\text{such that } U_i^T U_i = 1, \quad \text{and} \quad \Delta_i = \sqrt{B_i B_i^T}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{where } a_{ij} = \begin{cases} -\frac{1}{2\sigma_i} B_i B_i^T & \text{for } i = j \\ \frac{1}{\sigma_i^2 - \sigma_j^2}(\sigma_j B_i B_j^T - \sigma_i C_i^T C_j) & \text{for } i \neq j \end{cases}$$

Note that, in this case, since both U_1 and U_2 are column vectors, it can easily be shown that

$$\frac{\partial U_i}{\partial u_{1i}} = \begin{bmatrix} 1 \\ -\frac{u_{1i}}{u_{2i}} \end{bmatrix} \quad \text{for } i = 1, 2$$

Although the example given here is for a 2×1 vector, explicit formulae for the derivatives of U_i can still be found for vectors with more than two elements.

Also note that we do not have to solve matrix Lyapunov equations to find the derivatives of Δ_1 and Δ_2 as they are both scalars.

Case 2

In this case, we shall parametrize the U vectors in a different way. The parameter vector is given by $\theta = [\sigma_1, \sigma_2, b_{11}, b_{12}, b_{21}, b_{22}, \phi_1, \phi_2]$ where $\sigma_1 > \sigma_2 > 0$ and $\phi_1, \phi_2 \in (-\pi, \pi]$. The vectors U_1 and U_2 are

$$U_1 = \begin{bmatrix} cos(\phi_1) \\ sin(\phi_1) \end{bmatrix} \qquad U_2 = \begin{bmatrix} cos(\phi_2) \\ sin(\phi_2) \end{bmatrix}$$

and their derivatives with respect to ϕ_1 and ϕ_2 can easily be calculated. $\square$

We see from the above examples that in the non-generic case, explicit formulae for derivatives of state space matrices are not available but they are always available in the generic case. In the non-generic case, we need to solve a number of matrix Lyapunov equations to find the derivatives of Δ_i. However, in the case of the parameter U_i, its derivatives can be found explicitly if it is parametrized according to theorem 10.2.

12.4.2 Derivatives of state space matrices

We have seen in the previous section that explicit formulae for the derivatives of balanced state space matrices are available in the generic case. We shall summarise these formulae in this section. We shall use the parametrization used in the parameter estimation algorithm given in section 12.3.

The formulae for the derivatives of MIMO balanced state space matrices are as follows.

1. Derivatives of B
   Since B depends on the parameters in the b vector only. We have

   (a) $\frac{\partial B}{\partial x} = 0$ if x is any parameter from the s, ϕ or d vectors.

(b) Let $\frac{\partial B}{\partial b_{ij}} = [\tilde{b}_{kl}]$, then

$$\tilde{b}_{kl} = \begin{cases} 1 & \text{if } k = i \text{ and } l = j \\ 0 & \text{otherwise} \end{cases} \tag{12.23}$$

2. Derivatives of C
The matrix C depends on the parameters in the b and ϕ vectors only.

(a) $\frac{\partial C}{\partial x} = 0$ if x is any parameter from the s or d vectors.

(b) Let $\frac{\partial C}{\partial b_{ij}} = [\tilde{c}_{kl}]$, then

$$\tilde{c}_{kl} = \begin{cases} c_{ij}\frac{b_{ij}}{\Delta_i^2} & \text{if } k = i \\ 0 & \text{otherwise} \end{cases} \tag{12.24}$$

where $\Delta_i = \sqrt{B_i B_i^T}$. In other words,

$$\frac{\partial C}{\partial b_{ij}} = \begin{array}{c} \left[\begin{array}{ccc} 0_{p\times(i-1)} & C_i\frac{b_{ij}}{\Delta_i^2} & 0_{p\times(n-i)} \end{array} \right] \\ \uparrow \\ i^{th} \text{ column} \end{array} \tag{12.25}$$

(c) Before we give an expression for the derivative $\frac{\partial C}{\partial \phi_{i,j}}$, we first define the following function:

$$\begin{array}{rrcl} g_q : & \mathbb{R}\times\mathbb{R}^q & \rightarrow & \mathbb{R}^{q+1} \\ & (K,\beta) & \mapsto & w \end{array}$$

such that

$$\begin{cases} w_1 &= Kcos(\beta_1)cos(\beta_2)cos(\beta_3)\dots cos(\beta_{q-1})cos(\beta_q) \\ w_2 &= Kcos(\beta_1)cos(\beta_2)cos(\beta_3)\dots cos(\beta_{q-1})sin(\beta_q) \\ w_3 &= Kcos(\beta_1)cos(\beta_2)cos(\beta_3)\dots sin(\beta_{q-1}) \\ \vdots & \quad \vdots \\ w_q &= Kcos(\beta_1)sin(\beta_2) \\ w_{q+1} &= Ksin(\beta_1) \end{cases} \tag{12.26}$$

Also let

$$K_j = \begin{cases} 1 & \text{if } j = 1 \\ cos(\phi_{i,p-1})cos(\phi_{i,p-2})\dots cos(\phi_{i,j}) & \text{if } j > 1 \end{cases}$$

$$\gamma_j = \left[\begin{array}{cccc} \phi_{i,j-1}+\frac{\pi}{2} & \phi_{i,j-2} & \dots & \phi_{i,1} \end{array} \right]$$

The derivative $\frac{\partial C}{\partial \phi_{i,j}}$ is

$$\frac{\partial C}{\partial \phi_{i,j}} = \alpha \begin{array}{c} \left[\begin{array}{ccc} 0_{p\times(i-1)} & \begin{array}{c} g_{j-1}(K_j,\gamma_j) \\ 0_{(p-j)\times 1} \end{array} & 0_{p\times(n-i)} \end{array} \right] \\ \uparrow \\ i^{th} \text{ column} \end{array} \tag{12.27}$$

where

$$\alpha = R_{12}(\phi_1^0)R_{13}(\phi_2^0)\ldots R_{1p}(\phi_{p-1}^0)\sqrt{B_iB_i^T} \tag{12.28}$$

For the definitions of $\phi_1^0, \phi_2^0, \ldots, \phi_{p-1}^0$, see theorem 10.3.

3. Derivatives of A
   The matrix A depends on the parameters in the s, b and ϕ vectors only.

   (a) The derivative $\frac{\partial A}{\partial s_k}$ is given by the sum

$$\frac{\partial A}{\partial s_k} = \sum_{j=1}^{k} \frac{\partial A}{\partial \sigma_k} \tag{12.29}$$

Let $\frac{\partial A}{\partial \sigma_i} = [\tilde{a}_{kl}]$, then

$$\tilde{a}_{kl} = \begin{cases} -\frac{a_{kk}}{\sigma_k} & \text{if } k = l = i \\ \frac{-2\sigma_k\sigma_l}{(\sigma_k^2-\sigma_l^2)^2}B_kB_l^T + \frac{(\sigma_k^2+\sigma_l^2)}{(\sigma_k^2-\sigma_l^2)^2}C_k^TC_l & \text{for } k = i \text{ and } l \neq i \\ \frac{(\sigma_k^2+\sigma_l^2)}{(\sigma_k^2-\sigma_l^2)^2}B_kB_l^T - \frac{2\sigma_k\sigma_l}{(\sigma_k^2-\sigma_l^2)^2}C_k^TC_l & \text{for } l = i \text{ and } k \neq i \\ 0 & \text{otherwise} \end{cases} \tag{12.30}$$

   (b) Let $\frac{\partial A}{\partial b_{ij}} = [\bar{a}_{kl}]$, then

$$\bar{a}_{kl} = \begin{cases} -\frac{b_{kl}}{\sigma_k} & \text{for } k = l = i \\ \frac{1}{\sigma_k^2-\sigma_l^2}(\sigma_l b_{lj} - \frac{1}{\Delta_k^2}\sigma_k b_{kj}C_k^TC_l) & \text{for } k = i \text{ and } l \neq i \\ \frac{1}{\sigma_k^2-\sigma_l^2}(\sigma_l b_{kj} - \frac{1}{\Delta_l^2}\sigma_k b_{lj}C_k^TC_l) & \text{for } l = i \text{ and } k \neq i \\ 0 & \text{otherwise} \end{cases} \tag{12.31}$$

   (c) Let $\frac{\partial A}{\partial \phi_{i,j}} = [\hat{a}_{kl}]$, then

$$\hat{a}_{kl} = \begin{cases} -\frac{\sigma_k}{\sigma_k^2-\sigma_l^2}(\frac{\partial C_k}{\partial \phi_{k,j}})^TC_l & \text{for } k = i \text{ and } l \neq i \\ -\frac{\sigma_k}{\sigma_k^2-\sigma_l^2}C_k^T(\frac{\partial C_l}{\partial \phi_{l,j}}) & \text{for } l = i \text{ and } k \neq i \\ 0 & \text{otherwise} \end{cases} \tag{12.32}$$

Note that $\frac{\partial C_k}{\partial \phi_{i,j}}$ is the k^{th} column of $\frac{\partial C}{\partial \phi_{i,j}}$.

(d) $\frac{\partial A}{\partial d_{ij}} = 0$

4. Derivatives of D
   Since D depends on the parameters in the d vector only. We have

   (a) $\frac{\partial D}{\partial x} = 0$ if x is any parameter from the s, ϕ or b vector.

   (b) Let $\frac{\partial D}{\partial d_{ij}} = [\tilde{d}_{kl}]$, then

$$\tilde{d}_{kl} = \begin{cases} 1 & \text{if } k = i \text{ and } l = j \\ 0 & \text{otherwise} \end{cases} \tag{12.33}$$

12.4.3 Examples

In order to put together what we have discussed in the previous sections, we shall now give a few examples to show how the gradients of some objective functions can be calculated. In the following examples, we shall use θ to denote the parameter vector, then an estimate of the discrete time state space model is

$$(A_d(\theta), B_d(\theta), C_d(\theta), D_d(\theta))$$

and the Markov parameters of the model are

$$J_k(\theta) = \begin{cases} D_d(\theta) & k = 0 \\ C_d(\theta)A_d(\theta)^{k-1}B_d(\theta) & k \geq 1 \end{cases}$$

Example 1: Error function based on impulse response data

In this example, we shall assume that we have available the impulse response at each input-output channel of a p-output m-input process. It will also be assumed that the length of data is $(N+1)$. It is possible to assemble the impulse response data into the following Markov parameter sequence:

$$H_0, H_1, H_2, \ldots, H_N$$

We define the following objective function

$$V_1 = \sum_{k=0}^{N} w_k^2 \|W_o(J_k(\theta) - H_k)W_i\|_F^2 \tag{12.34}$$

where W_o and W_i are respectively the weights on the outputs and the inputs, and w_k is a weight which varies with time. In the rest of this example, we shall show how we can obtain the derivative of V_1 with respect to θ_i where θ_i is any element in the parameter vector θ.

Equation 12.34 can be re-written as

$$V_1 = \left\| \begin{bmatrix} w_0 W_o(J_0(\theta) - H_0)W_i \\ w_1 W_o(J_1(\theta) - H_1)W_i \\ w_2 W_o(J_2(\theta) - H_2)W_i \\ \vdots \\ w_N W_o(J_N(\theta) - H_N)W_i \end{bmatrix} \right\|_F^2$$

$$
\begin{aligned}
&= \left\| \underbrace{\begin{bmatrix} w_0 W_o & & & \\ & w_1 W_o & & \\ & & \ddots & \\ & & & w_N W_o \end{bmatrix}}_{U_o} \left(\underbrace{\begin{bmatrix} J_0(\theta) \\ J_1(\theta) \\ \vdots \\ J_N(\theta) \end{bmatrix}}_{J(\theta)} - \underbrace{\begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_N \end{bmatrix}}_{H} \right) \underbrace{\begin{bmatrix} W_i & & & \\ & W_i & & \\ & & \ddots & \\ & & & W_i \end{bmatrix}}_{U_i} \right\|_F^2 \\
&= \|U_o(J(\theta) - H)U_i\|_F^2 \qquad (12.35)
\end{aligned}
$$

By the fact that

$$
\frac{\partial}{\partial x}(\|X\|_F^2) = \frac{\partial}{\partial x}(tr(XX^T)) = 2tr(\frac{\partial X}{\partial x}X^T) \qquad (12.36)
$$

the derivative of V_1 is given by

$$
\begin{aligned}
\frac{\partial V_1}{\partial \theta_i} &= 2tr(U_o \frac{\partial J(\theta)}{\partial \theta_i} U_i U_i^T (J(\theta) - H)^T U_o^T) \qquad (12.37) \\
&= 2tr(U_o^T U_o \frac{\partial J(\theta)}{\partial \theta_i} U_i U_i^T (J(\theta) - H)^T) \qquad (12.38)
\end{aligned}
$$

Note that the second expression for $\frac{\partial V_1}{\partial \theta_i}$ is computationally more efficient than the first. The problem of finding the derivative of V_1 becomes one of finding the derivatives of the Markov parameters. The derivatives of the first few Markov parameters are

$$
\begin{aligned}
\frac{\partial J_0}{\partial \theta_i} &= \frac{\partial D_d}{\partial \theta_i} \qquad (12.39) \\
\frac{\partial J_1}{\partial \theta_i} &= \frac{\partial (C_d B_d)}{\partial \theta_i} = \frac{\partial C_d}{\partial \theta_i} B_d + C_d \frac{\partial B_d}{\partial \theta_i} \qquad (12.40) \\
\frac{\partial J_2}{\partial \theta_i} &= \frac{\partial (C_d A_d B_d)}{\partial \theta_i} = \frac{\partial C_d}{\partial \theta_i} A_d B_d + C_d \frac{\partial A_d}{\partial \theta_i} B_d + C_d A_d \frac{\partial B_d}{\partial \theta_i} \qquad (12.41) \\
\frac{\partial J_3}{\partial \theta_i} &= \frac{\partial (C_d A_d^2 B_d)}{\partial \theta_i} = \frac{\partial C_d}{\partial \theta_i} A_d^2 B_d + C_d \frac{\partial A_d}{\partial \theta_i} A_d B_d + C_d A_d \frac{\partial A_d}{\partial \theta_i} B_d + C_d A_d^2 \frac{\partial B_d}{\partial \theta_i} \qquad (12.42)
\end{aligned}
$$

By observing the pattern in the above formulae, the general equations for the derivatives of the Markov parameters are given as below. Let $J_{1:N}$ denote the lower N blocks of J, then

$$
\frac{\partial J}{\partial \theta_i} = \begin{bmatrix} \frac{\partial J_0}{\partial \theta_i} \\ \frac{\partial J_{1:N}}{\partial \theta_i} \end{bmatrix} \qquad (12.43)
$$

and

$$
\frac{\partial J_{1:N}}{\partial \theta_i} = D_1^i + D_2^i + D_3^i \qquad (12.44)
$$

where

$$
D_1^i = \begin{bmatrix} \frac{\partial C_d}{\partial \theta_i} B_d \\ \frac{\partial C_d}{\partial \theta_i} A_d B_d \\ \vdots \\ \frac{\partial C_d}{\partial \theta_i} A_d^{N-1} B_d \end{bmatrix} = (I_N \otimes \frac{\partial C_d}{\partial \theta_i}) \begin{bmatrix} B_d \\ A_d B_d \\ \vdots \\ A_d^{N-1} B_d \end{bmatrix} \qquad (12.45)
$$

$$
\begin{aligned}
D_2^i &= \begin{bmatrix} C_d \frac{\partial A_d}{\partial \theta_i} B_d \\ C_d A_d \frac{\partial A_d}{\partial \theta_i} B_d + C_d \frac{\partial A_d}{\partial \theta_i} A_d B_d \\ \vdots \\ C_d A_d^{N-2} \frac{\partial A_d}{\partial \theta_i} B_d + C_d A_d^{N-3} \frac{\partial A_d}{\partial \theta_i} A_d B_d + \ldots + C_d \frac{\partial A_d}{\partial \theta_i} A_d^{N-2} B_d \end{bmatrix} \\
&= \begin{bmatrix} 0_{p\times m} \\ \begin{bmatrix} C_d & 0 & \ldots & 0 \\ C_d A_d & C_d & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_d A_d^{N-2} & C_d A_d^{N-3} & & C_d \end{bmatrix} (I_{N-1} \otimes \frac{\partial A_d}{\partial \theta_i}) \begin{bmatrix} B_d \\ A_d B_d \\ \vdots \\ A_d^{N-2} B_d \end{bmatrix} \end{bmatrix} \qquad (12.46)
\end{aligned}
$$

$$
D_3^i = \begin{bmatrix} C_d \frac{\partial B_d}{\partial \theta_i} \\ C_d A_d \frac{\partial B_d}{\partial \theta_i} \\ \vdots \\ C_d A_d^{N-1} \frac{\partial B_d}{\partial \theta_i} \end{bmatrix} = \begin{bmatrix} C_d \\ C_d A_d \\ \vdots \\ C_d A_d^{N-1} \end{bmatrix} \frac{\partial B_d}{\partial \theta_i} \qquad (12.47)
$$

Example 2: Error function based on step response data

This example is similar to the previous one, except that we assume the step response at each input-output channel is available and a slightly different error function will be used. The length of data is assumed to be $N+1$. We shall first assemble the step response data into the following $N+1$ matrices:

$$S_0, S_1, S_2, \ldots, S_N$$

where the (i,j) element of S_k is the response at the i^{th} output at time k due to a unit step at the j^{th} input.

Further we define

$$M_k(\theta) = \sum_{i=0}^{k} J_i(\theta) \quad \text{for } k \geq 0 \qquad (12.48)$$

We define the objective function as

$$V_2(\theta) = \sum_{k=0}^{N} w_k^2 \| W_o (M_k(\theta) - S_k) W_i \|_F^2 \qquad (12.49)$$

which can be re-written as

$$V_2(\theta) = \| U_o (M(\theta) - S) U_i \|_F^2 \qquad (12.50)$$

where U_o and U_i are identical to those in equation 12.35, and $M(\theta)$ and S are defined in the same way as $J(\theta)$ and H.

Thus the derivative of V_2 is given by

$$\frac{\partial V_2}{\partial \theta_i} = 2tr(U_o \frac{\partial M(\theta)}{\partial \theta_i} U_i U_i^T (M(\theta) - S)^T U_o^T) \quad (12.51)$$

$$= 2tr(U_o^T U_o \frac{\partial M(\theta)}{\partial \theta_i} U_i U_i^T (M(\theta) - S)^T) \quad (12.52)$$

Note that the second expression for $\frac{\partial V_2}{\partial \theta_i}$ is computationally more efficient than the first. By the definition of $M(\theta)$, it can easily be shown that

$$\frac{\partial M(\theta)}{\partial \theta_i} = \begin{bmatrix} I_p & 0 & \dots & & 0 \\ I_p & I_p & 0 & \dots & 0 \\ I_p & I_p & I_p & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_p & I_p & I_p & \dots & I_p \end{bmatrix} \frac{\partial J(\theta)}{\partial \theta_i} \quad (12.53)$$

and $\frac{\partial J(\theta)}{\partial \theta_i}$ can be calculated in the same way as that given in the last example.

Example 3: Error function based on an arbitrary input sequence

In this example, we shall assume that we are given an input sequence

$$u_0, u_1, u_2, \dots, u_N$$

of length $(N+1)$ of a process whose output measurements are given by

$$y_0, y_1, y_2, \dots, y_N$$

The output of the model due to the above input sequence will be denoted by

$$\hat{y}_0(\theta), \hat{y}_1(\theta), \hat{y}_2(\theta), \dots, \hat{y}_N(\theta)$$

where

$$\hat{y}_t(\theta) = \sum_{k=0}^{t} J_{t-k}(\theta) u_k \quad (12.54)$$

We define the objective function as

$$V_3(\theta) = \sum_{k=0}^{N} \|\hat{y}_k(\theta) - y_k\|_2^2 \quad (12.55)$$

which can be rewritten as

$$V_3(\theta) = \left\| \underbrace{\begin{bmatrix} \hat{y}_0(\theta) \\ \hat{y}_1(\theta) \\ \hat{y}_2(\theta) \\ \vdots \\ \hat{y}_N(\theta) \end{bmatrix}}_{\hat{Y}(\theta)} - \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{Y} \right\|_2^2 \quad (12.56)$$

$$= \|\hat{Y}(\theta) - Y\|_2^2 \quad (12.57)$$

By the fact that

$$\frac{\partial}{\partial x}(\|X\|_2^2) = \frac{\partial}{\partial x}(X^T X) = 2(\frac{\partial X}{\partial x})^T X \tag{12.58}$$

the derivative of $V_3(\theta)$ is given by

$$\frac{\partial V_3(\theta)}{\partial \theta_i} = 2(\frac{\partial \hat{Y}(\theta)}{\partial \theta_i})^T(\hat{Y}(\theta) - Y) \tag{12.59}$$

By equation 12.54, the elements in the vector $\frac{\partial \hat{Y}(\theta)}{\partial \theta_i}$ are given by

$$\frac{\partial \hat{y}_t(\theta)}{\partial \theta_i} = \sum_{k=0}^{t} \frac{\partial J_{t-k}(\theta)}{\partial \theta_i} u_k \tag{12.60}$$

These elements can also be generated by passing the input sequence $u_0, u_1, \ldots, u_N$ through a finite impulse response system whose Markov parameters are

$$\frac{\partial J_0(\theta)}{\partial \theta_i}, \frac{\partial J_1(\theta)}{\partial \theta_i}, \frac{\partial J_2(\theta)}{\partial \theta_i}, \ldots, \frac{\partial J_N(\theta)}{\partial \theta_i}$$

The derivatives of the Markov parameters can be calculated in the same way as that given in example 1.

12.4.4 Summary on gradient calculations

From the discussion given in the previous few sections, we can summarise the steps to calculate the gradient of the objective function $V(\theta)$ as follows:

1. Express the derivative of $V(\theta)$ in terms of the derivatives of the Markov parameters. Generally, this can only be done on a case by case basis.

2. The derivatives of the Markov parameters are then expressed in terms of $\frac{\partial A_d}{\partial \theta_i}$, $\frac{\partial B_d}{\partial \theta_i}$, $\frac{\partial C_d}{\partial \theta_i}$ and $\frac{\partial D_d}{\partial \theta_i}$ using equations 12.43 to 12.47.

3. The derivatives of the discrete time state space matrices are then expressed in terms of $\frac{\partial A_c}{\partial \theta_i}$, $\frac{\partial B_c}{\partial \theta_i}$, $\frac{\partial C_c}{\partial \theta_i}$ and $\frac{\partial D_c}{\partial \theta_i}$ using equations 12.15 to 12.18. Finally explicit formulae for the derivatives of the continuous time state space matrices are stated in section 12.4.2.

12.5 Simulation example

In this section, we shall present a simulation example on using the parameter estimation algorithm which we have proposed in section 12.3. Our simulation model has 2 inputs, 2 outputs and 10 states. The Hankel singular values of the simulation model are

```
10, 10, 10, 10, 5, 5, 3, 3, 3, 3
```

Objective function $V(\theta)$	Kung's 6^{th} order	Kung's 8^{th} order
Before optimization	17.10	13.72
After optimization	8.32	3.85

Table 12.1: The model accuracy of various models before and after optimization

Since the simulation model is a non-generic system, we would like to check how well this system can be approximated by a generic one.

The outputs are obtained from applying a unit step at each of the inputs and then noise is added to the outputs to obtain the data. We then find our initial state space models using Kung's algorithm and 2 different models are obtained — a 6^{th} order model and an 8^{th} order model. These two initial models are then supplied to our parameter estimation algorithm using an objective function identical to the one used in example 2 in section 12.4.3. The weights w_t, W_i and W_o are all chosen to be unity.

We have plotted the output responses of the two optimized models in figures 12.1 and 12.2. It can be seen from the figures that this non-generic simulation model can be approximated rather well by generic systems. We have also summarised the values of the objective function before and after optimization in table 12.1.

We are also interested in the Hankel singular values of the optimized model. The Hankel singular values of the 6^{th} order optimized model are

```
9.4020, 9.3850, 9.3776, 9.3307, 4.0219, 2.9302
```

and those of the 8^{th} order optimized model are

```
10.5624, 10.3974, 10.0942, 10.0471, 4.8173, 4.3419, 3.0047, 2.8693
```

It is interesting to find that these results do indicate that some Hankel singular values of the simulation model are close to each other.

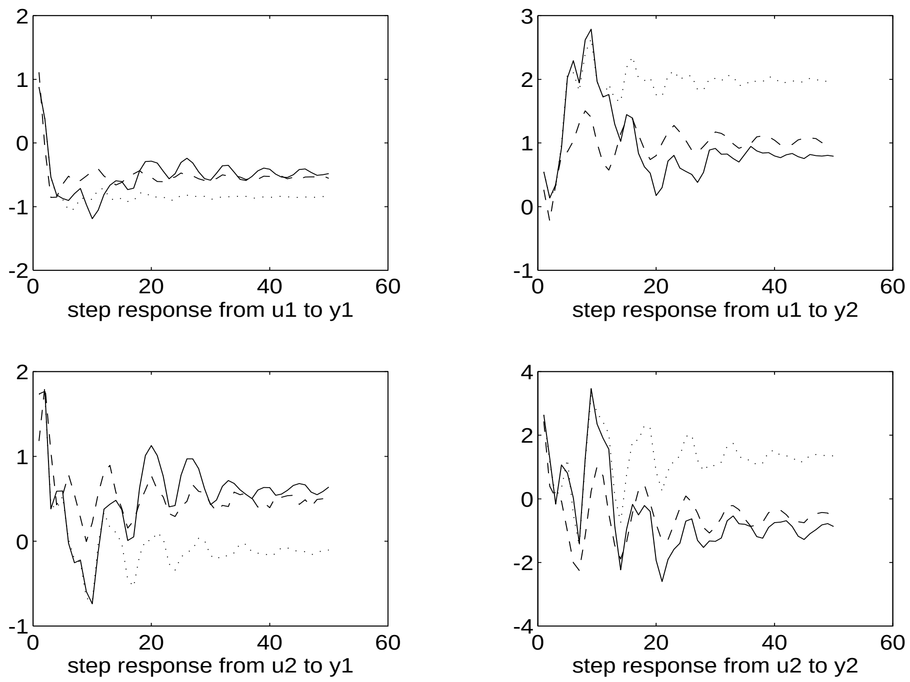


Figure 12.1: Simulation model (solid). 6^{th} order Kung's model (dotted). Optimised model (dashed).

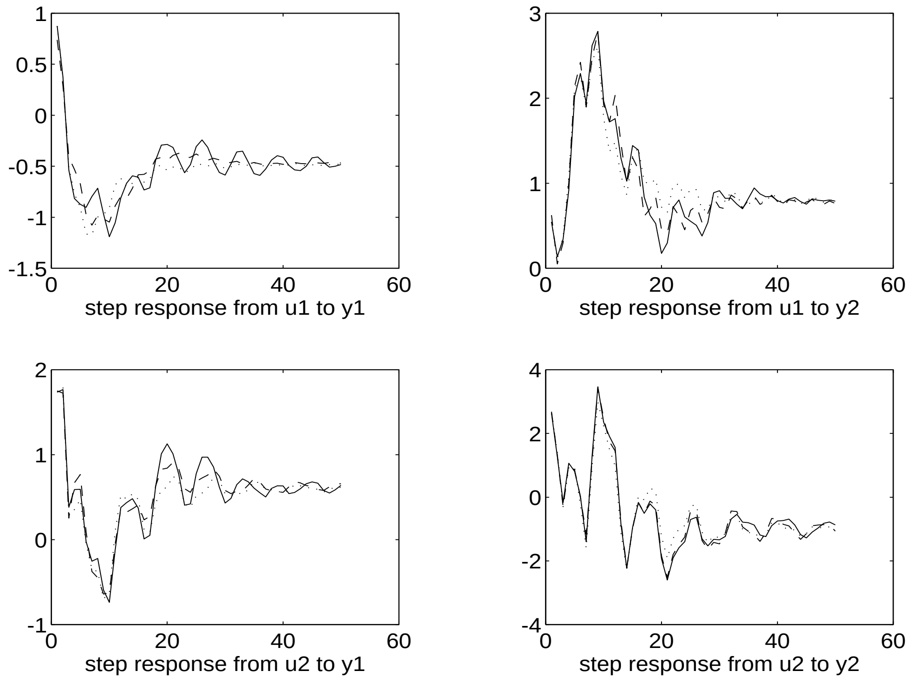


Figure 12.2: Simulation model (solid). 8^{th} order Kung's model (dotted). Optimised model (dashed).

Chapter 13

Case studies

In this chapter, we shall present two case studies on applying our proposed system identification algorithm to two different sets of industrial data. The first set of industrial data comes from a very detailed simulation model of a fractional distillation column. The second set comes from an industrial drying process.

13.1 Simulation model of a fractional distillation column

In this section, we shall present a case study of using the proposed system identification algorithm on data obtained from a very detailed simulation model of a fractional distillation column. The simulation model is made up of more than a thousand non-linear partial differential equations and can be run at or faster than real time. Although the simulation model provides a very accurate description of the real plant, it is not a suitable model for controller design due to its high complexity. So, there is a need for a simple linear model which can be used for the synthesis of controllers. We shall demonstrate how a simple linear state space nominal model can be obtained from this complicated simulation model using our proposed algorithm. We shall first start with a description of the 'plant' which is the simulation model.

13.1.1 Description of the plant and data

The simulation model has of 3 inputs and 2 outputs. The inputs are

- $u1$: input cooling temperature
- $u2$: reboiling temperature
- $u3$: pressure

and the outputs are

- $y1$: top product flow rate
- $y2$: C4 concentration

A total of six sets of data are available. The first two sets of data are obtained by applying a unit step at the input $u1$ while keeping the other two inputs at zero. The responses at the two outputs $y1$ and $y2$ are then sampled at a rate of once every second for a period of 2500 seconds, hence there are 2501 data points for each step response. The remaining four sets of data are obtained in a similar way by applying a unit step at the second and third input in turns. A step input is chosen because it provides accurate information on the steady state gain of the system. These step responses are plotted in figure 13.2 on page 127.

13.1.2 Obtaining a model

In this section, we shall describe how a nominal state space model can be obtained from the step response data of the simulation model. The steps to obtain a model are

1. Pretreatment of data.
2. Finding an initial state space estimate.
3. Parameter optimization.

In the step of finding an initial state space estimate, both Kung's algorithm and the model reduction algorithm due to Al-Saggaf will be used. We shall also make a comparison between these two methods.

Pretreatment of data

Since the data obtained from the simulation model is rather noise free, no pre-filtering of the data is required. A quick examination of the step responses shows that the response time of the plant is much greater than the sampling time, so a higher sampling rate can actually be used for the data. We shall first sample the output responses at a rate of once every ten samples to obtain a shorter set of data.

Since both Kung's and Al-Saggaf's algorithms require that the Markov parameters of the plant to be identified be made available, we shall first convert the step response data into impulse response data by performing successive differences on the data. The impulse response data can then be assembled into a Markov parameter sequence.

In chapter 11, it was conjectured that a better initial estimate can be obtained from Kung's or Al-Saggaf's algorithm if the energy content in each of the input-output channels is of nearly the same order of magnitude. Due to the units which the two outputs are measured in, the first output seems to contain a much greater amount of energy than the second one. We shall first perform scaling on the data before finding a state space estimate.

Let e_{ij} be the energy content in the input-output channel from the j^{th} input to the i^{th} output. For the original data, the energy contents in the input-output channels are

$$E = \begin{bmatrix} 127 & 50 & 463 \\ 6 \times 10^{-4} & 1 \times 10^{-4} & 3.1 \times 10^{-3} \end{bmatrix} \tag{13.1}$$

As anticipated, the energy contents at the second output channels are much lower than those at the first. We used the scaling algorithm as given in appendix B and the following

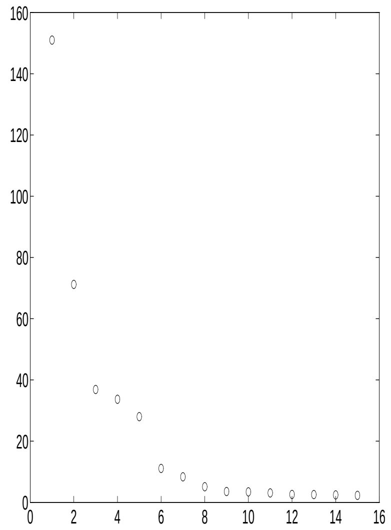


Figure 13.1: The largest 15 singular values

input-output scaling factors were obtained.

$$D_i = \text{diag}(18.6, 59.5, 1) \tag{13.2}$$
$$D_o = \text{diag}(3.6 \times 10^{-3}, 2.8 \times 10^{3}) \tag{13.3}$$

The energy contents of the input-output channels after scaling are given by

$$\tilde{E} = D_o \times E \times D_i = \begin{bmatrix} 8.5 & 10.7 & 30.1 \\ 30.1 & 23.9 & 8.5 \end{bmatrix} \tag{13.4}$$

Note that before scaling, the ratio of the highest energy contents to the lowest is 463×10^4 while that after scaling is reduced to 3.5. Now we are ready to use both Kung's and Al-Saggaf's algorithms to find initial state space estimates form the data. In the discussion to follow, the models are assumed to have been calculated from the scaled data unless otherwise indicated.

Finding an initial estimate using Kung's algorithm

The major decision which we need to make after applying Kung's algorithm to the data is to find a suitable model order for the initial state space estimate. As mentioned in chapter 11, the number of significant Hankel singular values gives a rough indication as to what the order of the model can be. However, we still need to compare the response of the model with the data to check whether the model order selected is a reasonable one.

In figure 13.1, we plotted the largest 15 Hankel singular values obtained from Kung's algorithm. The numerical values of the first 12 singular values are as follows:

`151, 71, 37, 34, 28, 11, 8, 5, 3.5, 3.4, 3.1, 2.6`

Since the sixth Hankel singular value is roughly one tenth of the largest one, it seems that it is appropriate to start with a sixth order model. In figure 13.3 (page 128), we plotted the step responses of the 6^{th} order Kung's estimate against those from the data. It can be seen from the diagram that Kung's estimate gives a rather good fit to the original data. The fit between the model and the data is improved by increasing the order of the state space estimate, as can be seen in figure 13.4 (page 129) which shows the step responses of the 8^{th} order Kung's estimate against those from the data.

In order to show the effect of scaling on the fit of the model, we calculated the 6^{th} order Kung's estimate using the original data. The step responses of the model and those of the data are plotted in figure 13.5 (page 130). It can be seen that the fit for two of the input-output channels is much worst than that obtained from the model with scaling. Also, the fit for the other channels are generally not as good as that obtained from the model with scaling.

Finding an initial estimate using Al-Saggaf's model reduction algorithm

Similar to Kung's algorithm, the major decision which we need to make when using Al-Saggaf's algorithm is to choose an appropriate model order for the initial state space estimate. Again, the Hankel singular values can be used as a rough guide to what an appropriate model order may be. Since Kung's algorithm and Al-Saggaf's algorithm share the same set of singular values, so from the analysis above, we shall first start with a model of order 6. However, for Al-Saggaf's algorithm, there is an extra parameter α which can take the value of either $+1$ or -1. Since we would like to obtain a good fit to the step response data, it seems appropriate to choose the value of -1 for α as this choice will give a good approximation to the steady state gain.

In figure 13.6 (page 131), the step responses from the 6^{th} order Al-Saggaf's model (with $\alpha = -1$) are plotted against the data. When compared this with figure 13.3 (page 128), it can be seen that this model gives excellent steady state response compared with Kung's 6^{th} order model. An improved fit is obtained if we use an 8^{th} order model instead, see figure 13.7 on page 132.

In order to show the effect of scaling on the fit of the model, we calculated the 6^{th} order Al-Saggaf's estimate using the original data. The step responses of the model and those of the data are plotted in figure 13.8 (page 133). Since the parameter $\alpha = -1$ is used, the steady state responses of the model are similar to those from the data. However, for the 3 input-output channels which have a lower energy content, the model has introduced some high frequency signals at the initial part of these step responses.

Finally, we would like to compare the model given by Al-Saggaf's algorithm with the parameter α chosen to be 1 with the model of the same order obtained from Kung's algorithm. For the purpose of comparison, we shall choose the model order to be 6 and we plotted the step responses of these two models in figure 13.9 (page 134). Our experience so far seems to suggest that these two methods give roughly the same model, as can be seen in this figure of comparison.

Parameter optimization

Since one of the goals of system identification is to obtain a model which describes the data in a sufficiently accurate manner, a natural question to ask at this point is whether the models obtained from Kung's and Al-Saggaf's algorithms are accurate enough. We have chosen to use step inputs for our identification experiment in order that we can get good estimates for the steady state gains, so it is appropriate to define the objective function in the following way. Let $\{y_{ij}(t)\}_{t=0,\ldots,N}$ be the response at the i^{th} output of the simulation model due to a unit step at the j^{th} input. We shall assemble these data into the matrices $Y(t)$ for $t = 0, ..., N$ such that the (i, j) element of $Y(t)$ is $y_{ij}(t)$. We shall use θ to denote the parameter vector and $\{\hat{y}_{ij}(\theta, t)\}_{t=0,\ldots,N}$ to denote the response at the i^{th} output of the model due to a unit step at the j^{th} input. Again, we shall assemble these data into the matrices $\hat{Y}(\theta, t)$ for $t = 0, ..., N$. We shall define the objective function as

$$V(\theta) \quad = \quad \sum_{t=0}^{N} \|D_o Y(t) D_i - \hat{Y}(\theta, t)\|_F^2 \tag{13.5}$$

where D_i and D_o are respectively the input and output scaling matrices which we have used earlier to scale the data. Scaling factors are used here so that the optimization algorithm will put nearly the same emphasis on each input-output channel. After the optimization has been completed, the correct model can be recovered by using equations 11.28 to 11.31. Note that the above objective function is mathematically equivalent to the following:

$$\tilde{V}(\theta) \quad = \quad \sum_{t=0}^{N} \|D_o(Y(t) - \hat{Y}(\theta, t))D_i\|_F^2 \tag{13.6}$$

However, the criterion defined by equation 13.5 is numerically better conditioned as the parameters in the θ vector are of nearly the same order of magnitude [18]. In the case of $\tilde{V}(\theta)$, the parameters in θ which are associated with the second output generally have a much smaller order of magnitude, hence this problem formulation is numerically less well conditioned.

We have applied the parameter estimation algorithm given in section 12.3 to the following four initial state space estimates.

1. 6^{th} order model from Kung's algorithm
2. 8^{th} order model from Kung's algorithm
3. 6^{th} order model from Al-Saggaf's model reduction algorithm with $\alpha = -1$
4. 8^{th} order model from Al-Saggaf's model reduction algorithm with $\alpha = -1$

In order to compare the quality of the optimized models, we have plotted, for each of the models above, the step responses of the simulation model, initial state space estimate and the optimized model on the same figure. These figures can be found on pages 135 to 138. We have also summarised the values of the objective functions before and after optimization in table 13.1.

	Kung's		Al-Saggaf's	
Objective function $V(\theta)$	6^{th} order	8^{th} order	6^{th} order	8^{th} order
Before optimization	116.57	32.68	42.73	16.55
After optimization	27.54	9.94	23.84	13.68

Table 13.1: The model accuracy of various models before and after optimization

The results of these experiments confirm that the optimization problem which we are solving is a non-convex one as the optimized models obtained from different initial estimates are different from each other. An interesting point to note is that although Al-Saggaf's 8^{th} order model is more accurate than Kung's 8^{th} order model, the optimized model from Kung's is more accurate than that from Al-Saggaf's. This again confirms the fact that the optimization problem is non-convex and is highly non-linear.

Note:

Section 13.2 begins on page 139.

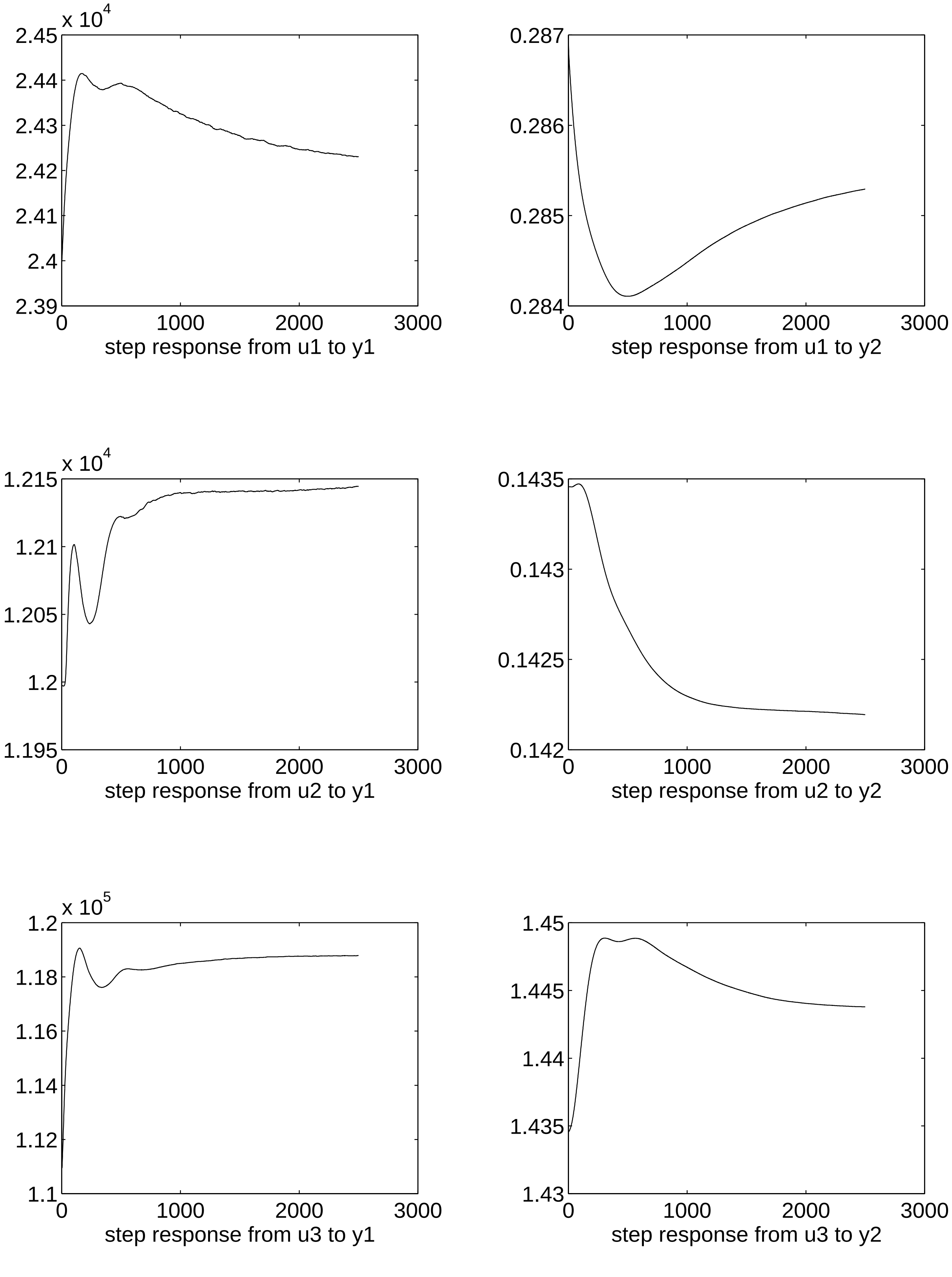


Figure 13.2: Unit step response at each input-output channel of the simulation model. Unit time interval = 1s

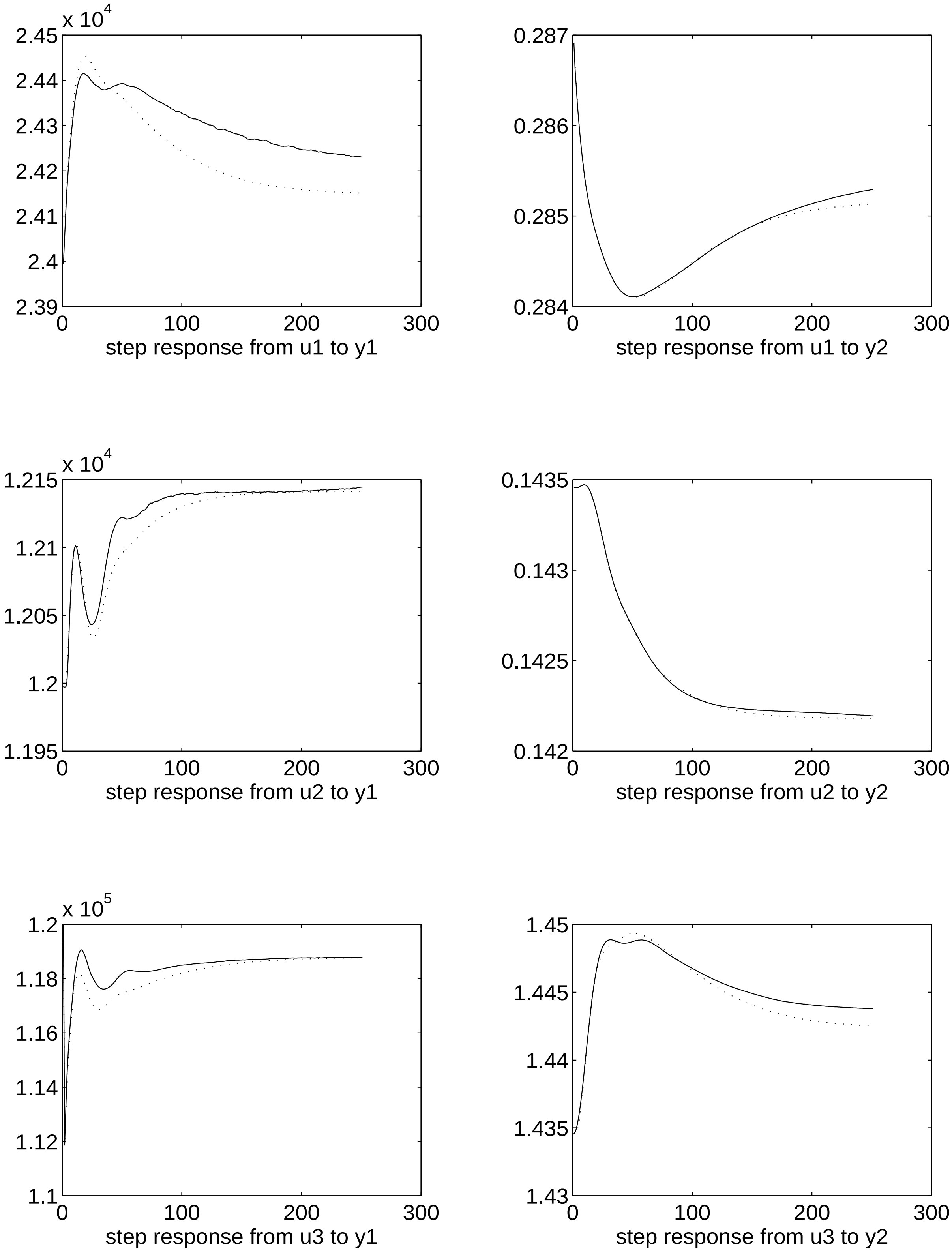


Figure 13.3: Simulation model (solid). 6^{th} order Kung's estimate obtained from scaled data (dotted). Unit time interval = 10s.

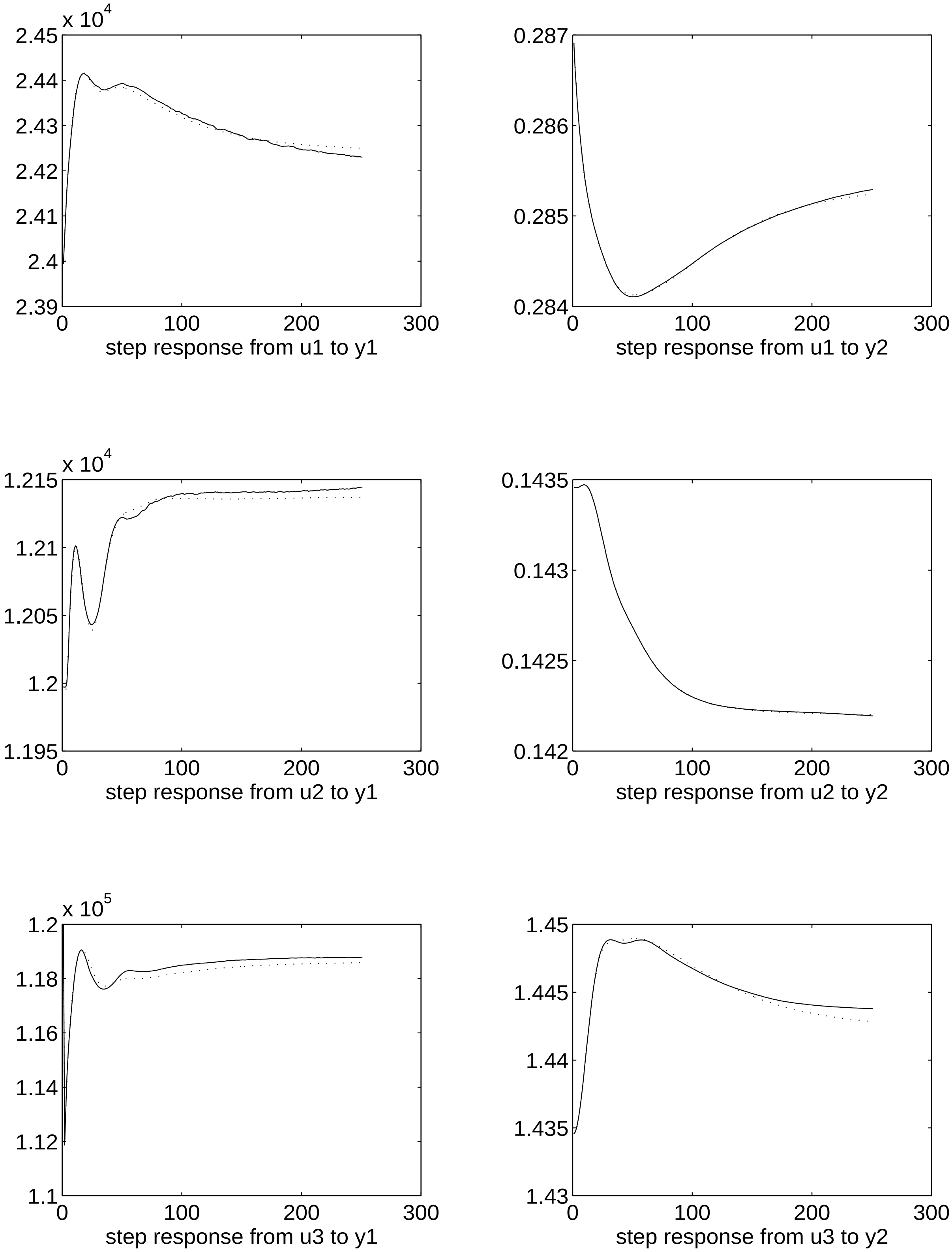


Figure 13.4: Simulation model (solid). 8^{th} order Kung's estimate obtained from scaled data (dotted). Unit time interval = 10s.

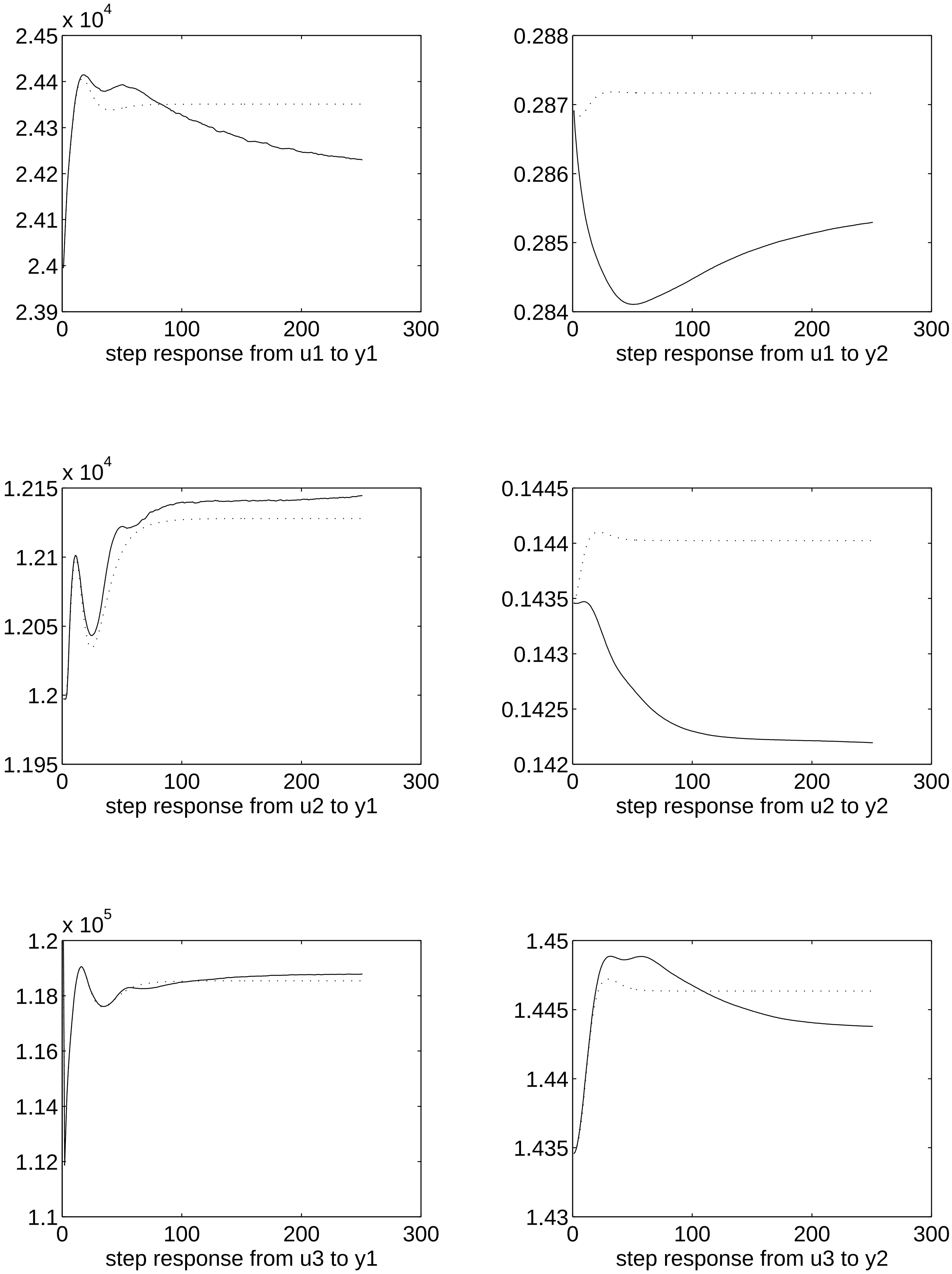


Figure 13.5: Simulation model (solid). 6^{th} order Kung's estimate obtained from the original data i.e. without scaling (dotted). Unit time interval = 10s.

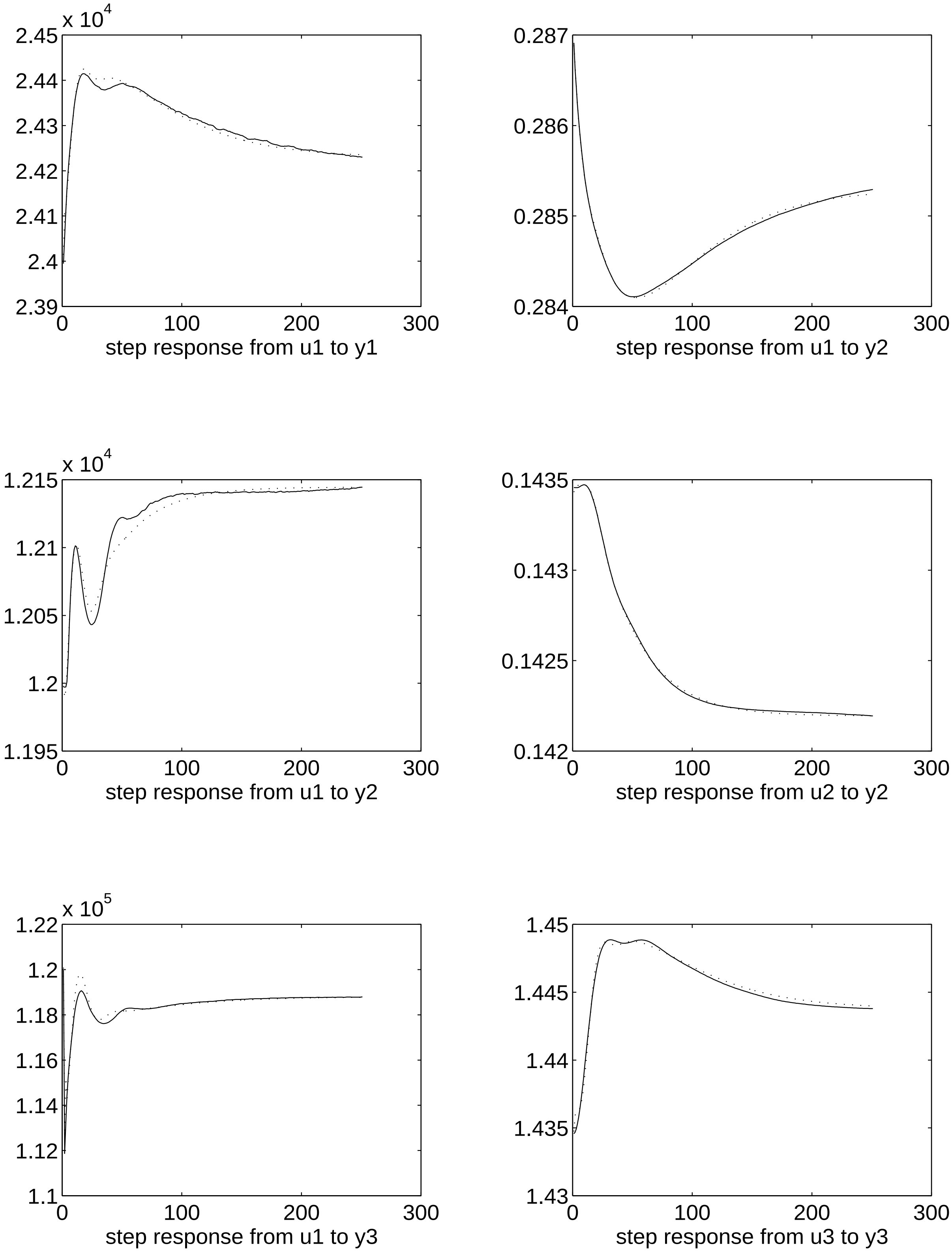


Figure 13.6: Simulation model (solid). 6^{th} order Al-Saggaf's estimate obtained from scaled data (dotted). Unit time interval = 10s.

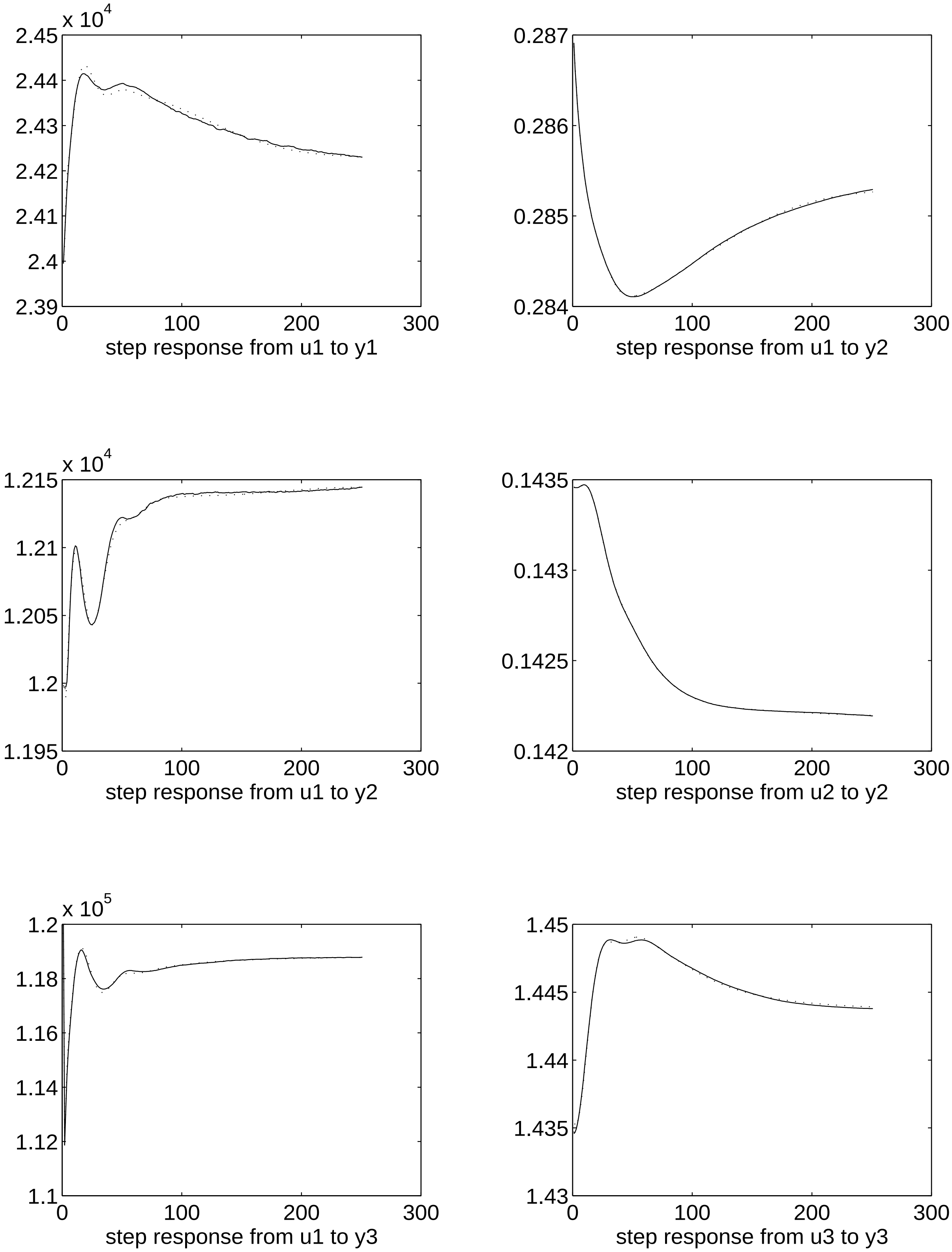


Figure 13.7: Simulation model (solid). 8^{th} order Al-Saggaf's estimate obtained from scaled data (dotted). Unit time interval = 10s.

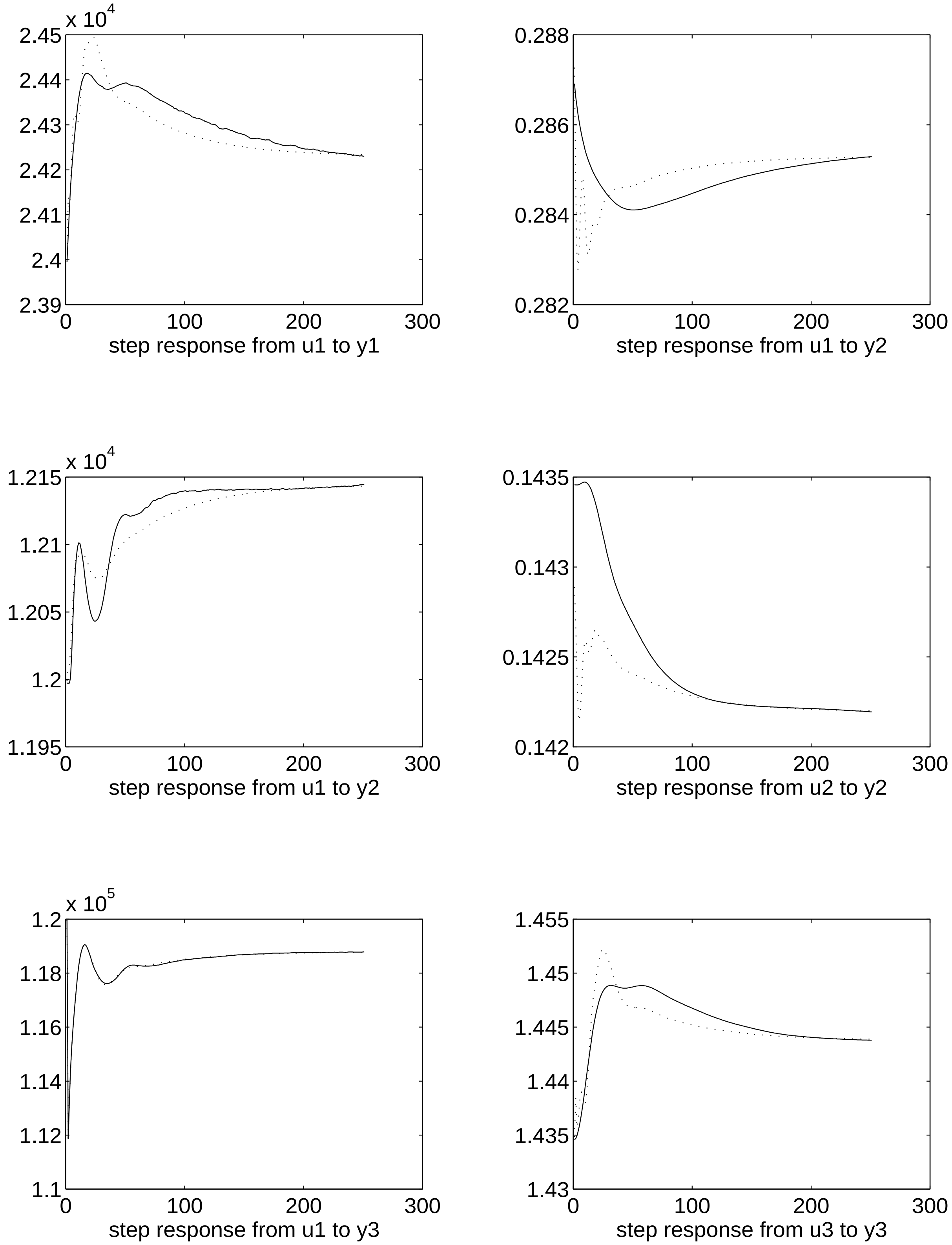


Figure 13.8: Simulation model (solid). 6^{th} order Al-Saggaf's estimate obtained from the original data i.e. without scaling (dotted). Unit time interval = 10s.

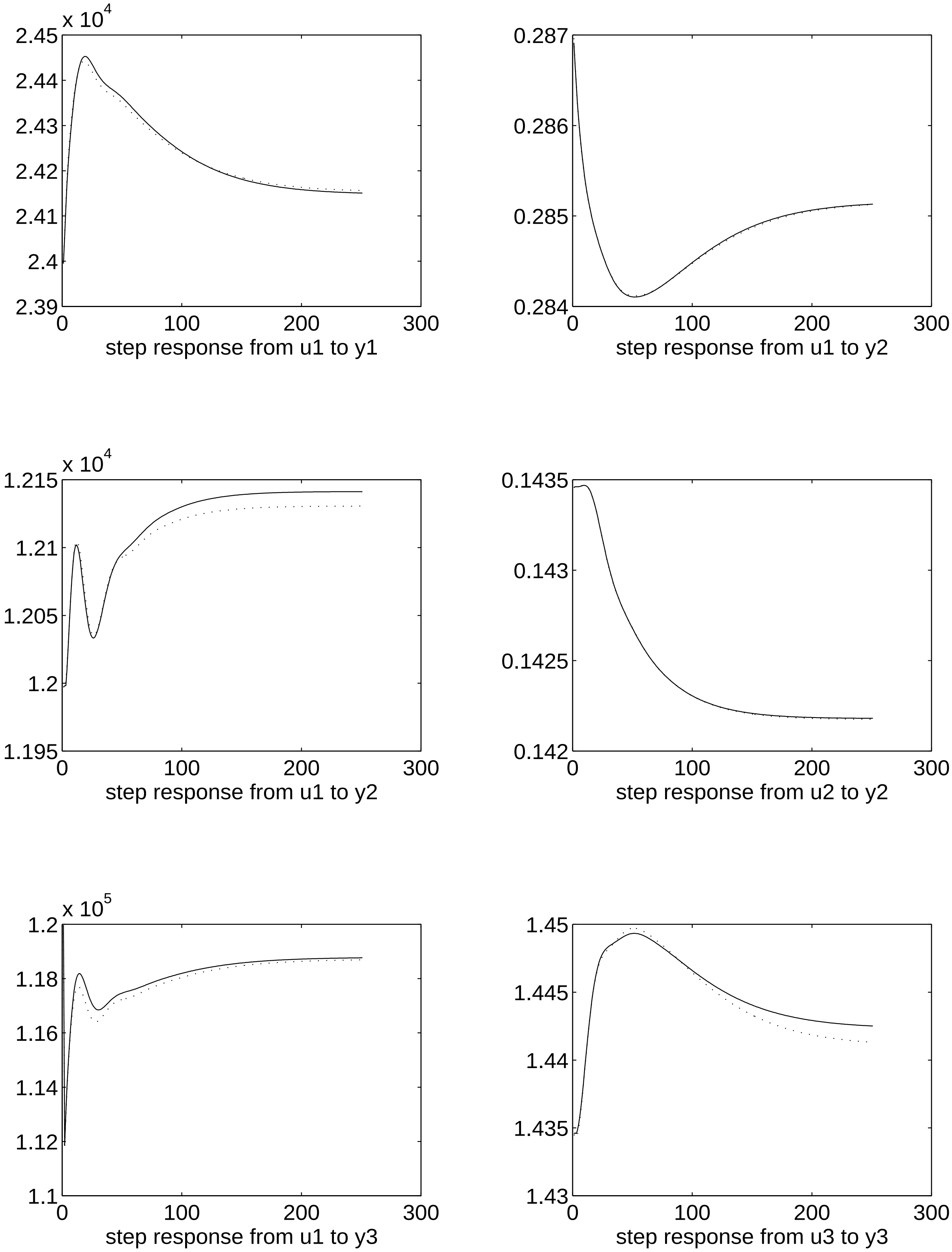


Figure 13.9: 6^{th} order Kung's model (solid). 6^{th} order Al-Saggaf's model with $\alpha = 1$ (dotted). Unit time interval = 10s.

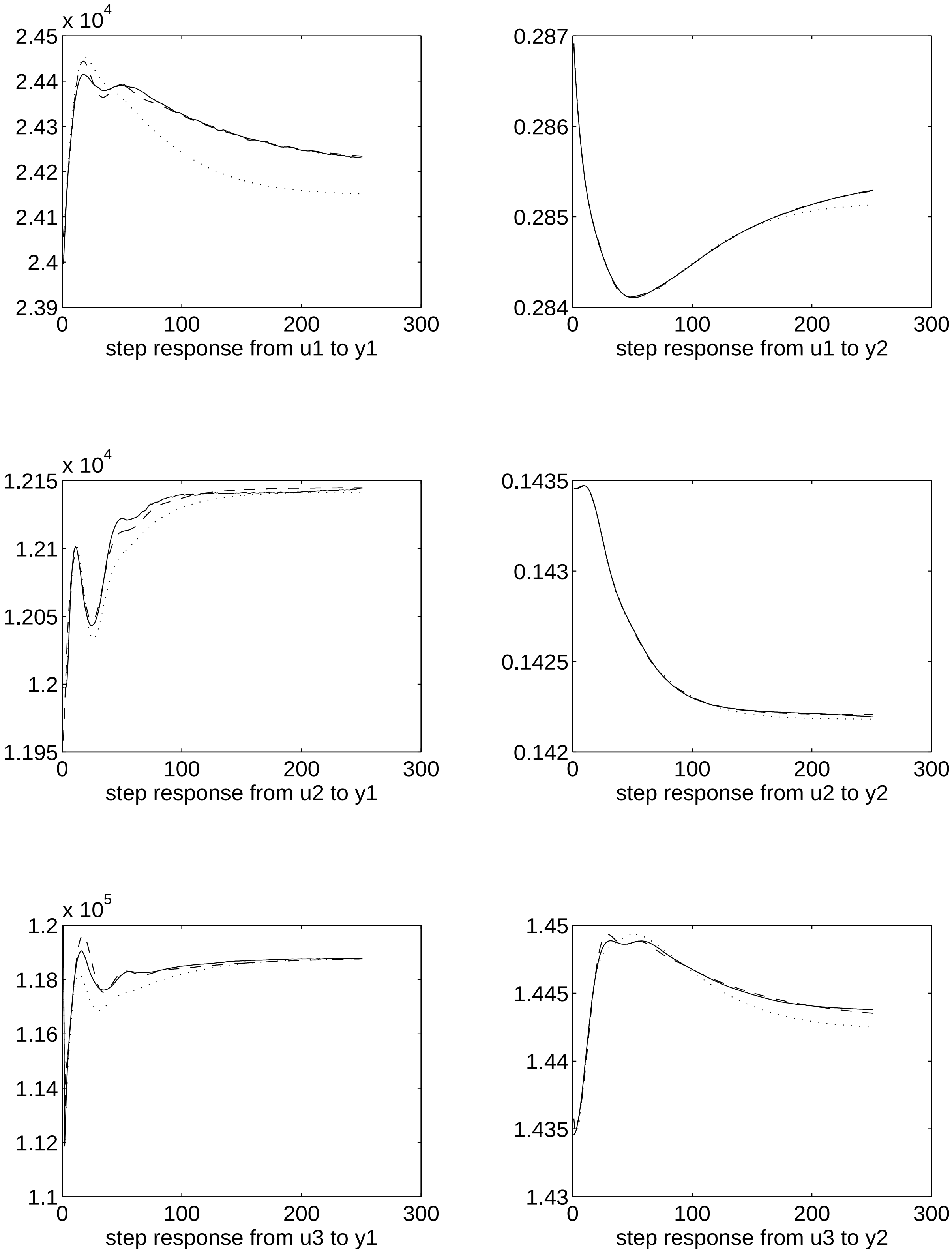


Figure 13.10: Simulation model (solid). 6^{th} order Kung's estimate obtained from scaled data (dotted). Optimised model (dashed). Unit time interval = 10s.

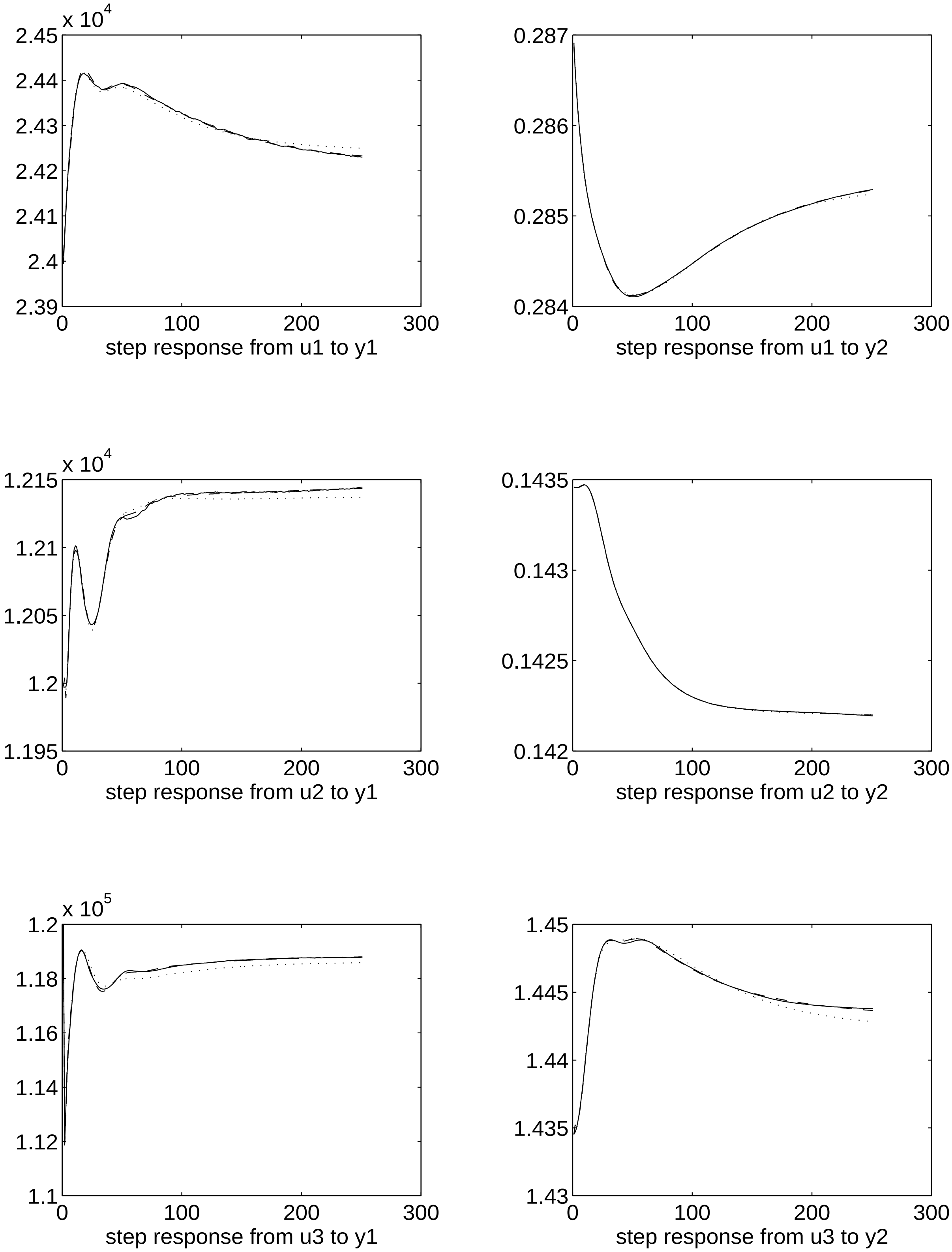


Figure 13.11: Simulation model (solid). 8^{th} order Kung's estimate obtained from scaled data (dotted). Optimised model (dashed). Unit time interval = 10s.

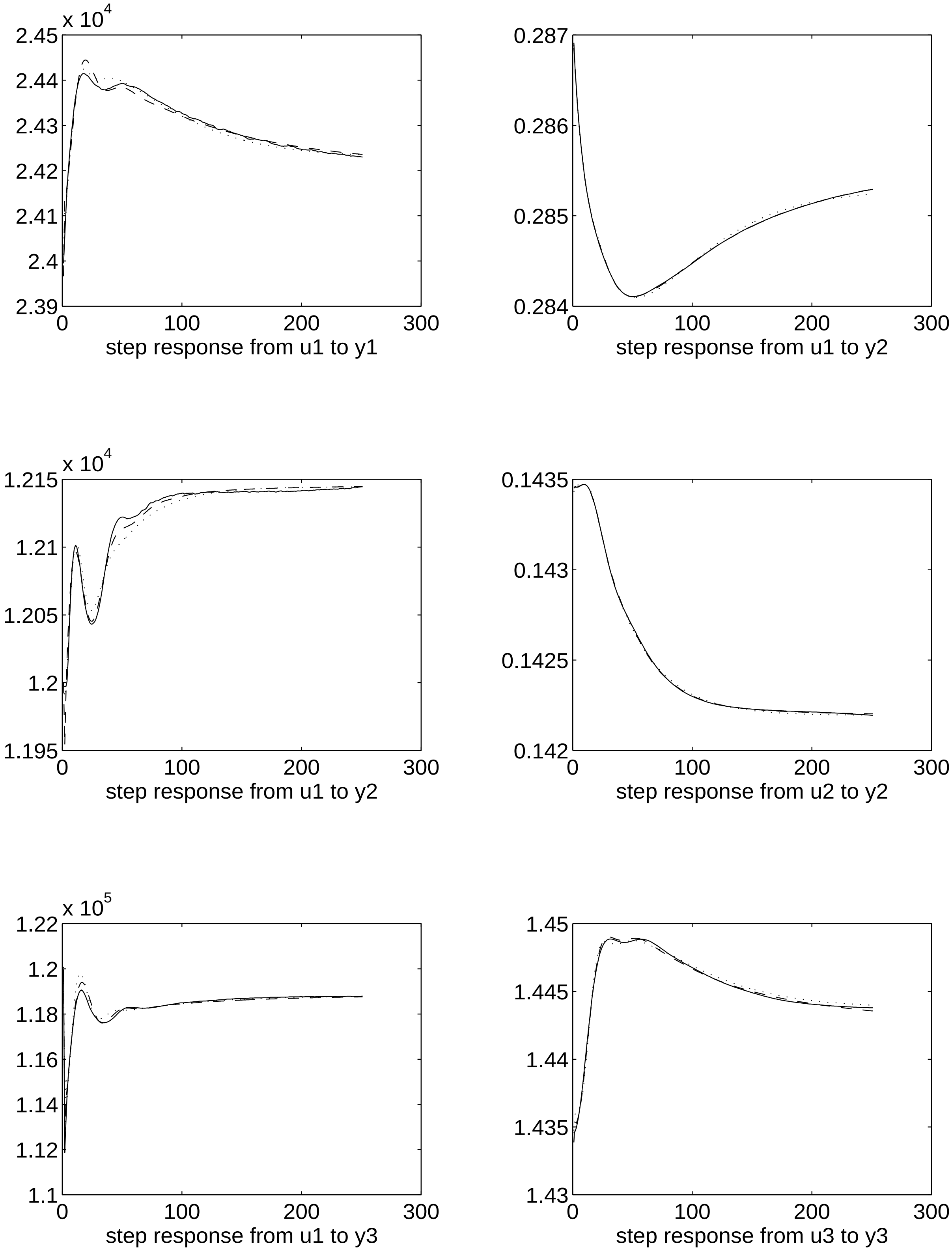


Figure 13.12: Simulation model (solid). 6^{th} order Al-Saggaf's estimate obtained from scaled data (dotted). Optimised model (dashed). Unit time interval = 10s.

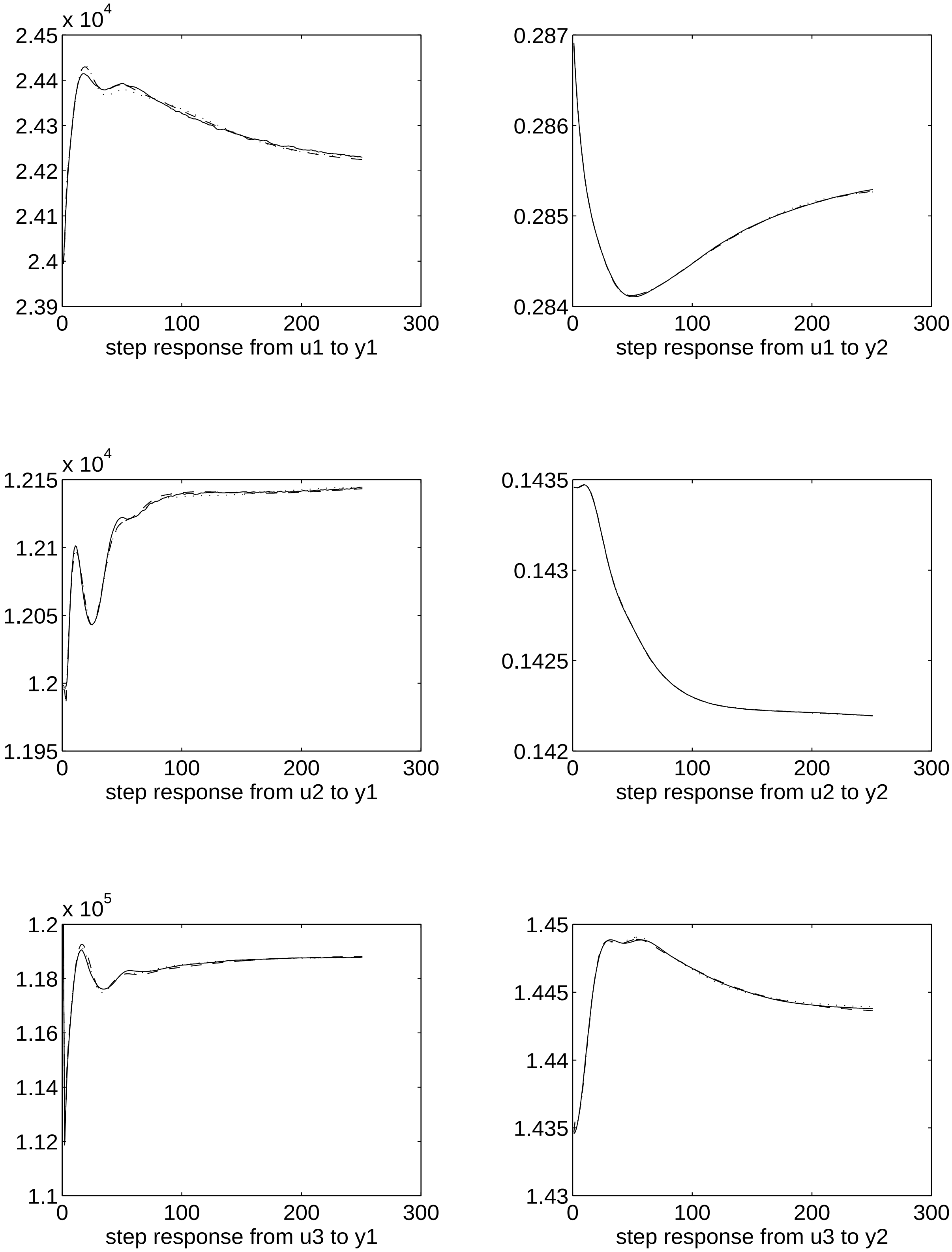


Figure 13.13: Simulation model (solid). 8^{th} order Al-Saggaf's estimate obtained from scaled data (dotted). Optimised model (dashed). Unit time interval = 10s.

13.2 An industrial process

In this section, we shall present a case study on applying a slightly modified version of our system identification algorithm to data obtained from an industrial process. This process dries the raw material supplied to the plant by passing hot gas over it. This section is further divided into 3 sub-sections. In section 13.2.1, we shall give a brief description of the plant and the data. In section 13.2.2, we shall present a slightly modified version of our system identification algorithm, the main difference being the use of minimum-phase balanced parametrizations for the predictors. Finally, we shall describe how a model can be obtained in section 13.2.3.

13.2.1 Description of the plant and data

The industrial process has 3 inputs and 3 outputs. The inputs are

- $u1$: gas valve position
- $u2$: speed of the exhaust fan
- $u3$: rate of flow of the raw material

The outputs are

- $y1$: dry bulb temperature
- $y2$: wet bulb temperature
- $y3$: change in the moisture content of the material

To obtain the data for identification, pseudo random binary signals are applied to the first two inputs while the third input is unregulated. The input and output signals are measured and sampled at a rate of once every second to obtain 11665 samples.

13.2.2 A modified algorithm

In this section, we shall present a slightly modified version of our system identification algorithm presented in section 9.5. The slightly modified algorithm is

1. An initial state space estimate of the form given by equations 11.53 and 11.54 is first obtained by using the unbiased version of N4SID algorithm described in section 11.3. The model is then converted into the following form using standard theory on Kalman filtering:

$$\left\{ \begin{array}{rcl} x_{k+1} & = & Ax_k + Bu_k + K\nu_k \\ y_k & = & Cx_k + Du_k + \nu_k \end{array} \right. \tag{13.7}$$

   where K is the Kalman filter gain and ν_k is an innovation process.

2. A prediction error type parameter estimation algorithm will be used here with the predictor given by

$$\begin{cases} \hat{x}_{k+1} &= A\hat{x}_k + Bu_k + K(y_k - \hat{y}_k) \\ \hat{y}_k &= C\hat{x}_k + Du_k \end{cases} \tag{13.8}$$

The discrete time stable and minimum-phase system (A, K, C, I) will be parametrized by using the balanced canonical forms for continuous time stable and minimum-phase systems given in [43].

For the rest of this section, we shall briefly describe the continuous time minimum-phase balanced realizations and their parametrizations here. We start with the definition of continuous time minimum-phase balanced realizations.

Definition 13.1 *(McGinnie [43]) Let $S_c = (A_c, B_c, C_c, D_c) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times n} \times \mathbb{R}^{p\times p}$ be a minimal state space realization of a continuous time stable and minimum-phase system. Also let P_c and Q_c be the solutions to the following two continuous time Lyapunov equations.*

$$A_cP_c + P_cA_c^T + B_cB_c^T = 0 \tag{13.9}$$
$$(A_c - B_cD_c^{-1}C_c)^TQ_c + Q_c(A_c - B_cD_c^{-1}C_c) + C_c^TD_c^{-T}D_c^{-1}C_c = 0 \tag{13.10}$$

Note that Q_c is in fact the observability Gramian of the inverse state space system of S_c. The state space system S_c is said to be continuous time minimum-phase balanced with minimum-phase Gramian Σ_c if $P_c = Q_c = \Sigma_c = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)$ where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$. We shall call the diagonal elements of Σ_c i.e. $\sigma_1, \ldots, \sigma_n$ the minimum phase characteristic values of the system.

The existence of minimum-phased balanced realizations was also proved in [43]. The discrete time minimum-phase balanced realizations are defined in an analogous way i.e. a discrete time stable and minimum-phase state space system is balanced if the controllability Gramian of the system and the observability Gramian of the inverse system are equal and diagonal. The continuous time and discrete time minimum-phase balanced realizations are related by the bilinear transformation (equations 5.30 to 5.33) which preserves minimum-phase balancedness (see [43] for a proof).

A parametrization for continuous time minimum-phase balanced systems was given in [43]. The structure of this parametrization is very similar to the parametrization for the continuous time balanced stable systems given in section 10.2.2. The only difference being in the way in which the real-valued parameters enter the state space matrices. Similar to the parameter estimation algorithm suggested in section 12.3, we shall only use the generic structure for parameter optimization. In an analogous way, we define a generic minimum-phase system as one whose minimum-phase characteristic values are distinct. Again, the generic structure has a simpler parametrization compared with the non-generic structures. Here, we shall give a parametrization for the generic structure of minimum-phase balanced realizations. The following lemma is an adapted version of the result given in [43].

Lemma 13.1 *Let $(A_c, B_c, C_c, D_c) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times n} \times \mathbb{R}^{p\times p}$ be a minimal state space realization of a continuous time stable and minimum-phase system. If (A_c, B_c, C_c, D_c) is a* *minimum-phase balanced with distinct minimum-phase characteristic values, then it can be parametrized by the following set of parameters:*

$$\begin{array}{llllll} \sigma_1 & > \ldots > & \sigma_j & > \ldots > & \sigma_n & \sigma_j \in \mathbb{R},\ \sigma_j > 0 \\ U_1 & \ldots & U_j & \ldots & U_n & U_j \in \mathbb{R}^{p\times 1} \text{ where } U_j^T U_j = 1 \\ B_1 & \ldots & B_j & \ldots & B_n & B_j \in \mathbb{R}^{1\times p} \text{ such that the first element in the} \\ & & & & & \text{row is positive} \\ D & & & & & D \in \mathbb{R}^{p\times p} \text{ such that } D \text{ is invertible} \end{array}$$

The state space matrices A_c, B_c, C_c and D_c are given by:

1. $$B_c = \begin{pmatrix} B_1 \\ \vdots \\ B_j \\ \vdots \\ B_n \end{pmatrix}$$

2. *Let* $\tilde{C}_c = \begin{pmatrix} \tilde{C}_1 & \ldots & \tilde{C}_j & \cdots & \tilde{C}_n \end{pmatrix}$ *where*

$$\tilde{C}_j = \sigma_j B_j^T + \sqrt{1+\sigma_i^2}\, U_j \sqrt{B_j B_j^T} \tag{13.11}$$

then

$$C_c = D\tilde{C}_c \tag{13.12}$$

3. $A_c = [a_{ij}]$ *for* $i, j = 1, ..., n$ *where*

$$a_{ij} = \begin{cases} \frac{1}{2\sigma_j} B_j B_j^T & \text{for } i = j \\ \frac{1}{\sigma_i^2 - \sigma_j^2} \left(\sigma_j (1+\sigma_i^2) B_i B_j^T - \right. & \\ \left. \sigma_i \sqrt{1+\sigma_i^2}\sqrt{1+\sigma_j^2}\sqrt{B_i B_i^T}\sqrt{B_j B_j^T} U_i^T U_j\right) & \text{for } i \neq j \end{cases} \tag{13.13}$$

4. $D_c = D$

Note that the unit vectors U_j's will be parametrized by using the local coordinate system given in section 10.2.5.

Moreover, the minimum-phase Gramian of (A, B, C, D) is $\Sigma = diag(\sigma_1, ..., \sigma_j, ..., \sigma_n)$.

The above parametrization can be used to parametrize all the discrete time predictors of the innovation form and whose minimum-phase characteristic values are distinct. In order to parametrize these predictors, we need to put in the following constraint (compare with equation 5.33 of the bilinear transformation)

$$D_c + C_c(I - A_c)^{-1} B_c = I \tag{13.14}$$

Thus the matrix D_c is no longer an independent variable but is dependent on A_c, B_c and C_c. We have the following lemma.

Lemma 13.2 *Let $\bar{S}_c = (\bar{A}_c, \bar{B}_c, \bar{C}_c, \bar{D}_c)$ be a continuous time minimum-phase balanced system such that*

1. *its minimum-phase characteristic values are distinct, and*
2. *if $\bar{S}_d = (\bar{A}_d, \bar{B}_d, \bar{C}_d, \bar{D}_d)$ is the discrete time minimum-phase balanced system obtained from applying bilinear transformation (equations 5.30 to 5.33) to $\bar{S}_c$, then $\bar{S}_d$ is in innovation form i.e. $\bar{D}_d = I$.*

All such systems can be parametrized by using the parameters σ_i's, B_i's and U_i's as in lemma 13.1, as follows

$$\bar{A}_c = A_c \tag{13.15}$$
$$\bar{B}_c = C_c \tag{13.16}$$
$$\bar{D}_c = I - \tilde{C}_c(I - A_c + B_c\tilde{C}_c)^{-1}B_c \tag{13.17}$$
$$\bar{C}_c = \bar{D}_c\tilde{C}_c \tag{13.18}$$

where A_c, B_c and $\tilde{C}_c$ are as given in lemma 13.1.

Proof: Due to the second restriction stated in the lemma, the matrix $\bar{D}_c$ is no longer a free parameter. Let $(\bar{A}_c, \bar{B}_c, \bar{C}_c, \bar{D}_c) = (A_c, B_c, X\tilde{C}_c, X)$ where X is an invertible $p \times p$ matrix. In order that $\bar{D}_d = I$,

$$X + X\tilde{C}_c(I - A_c)^{-1}B_c = I \tag{13.19}$$

By using the matrix inversion lemma, we have

$$X = I - \tilde{C}_c(I - A_c + B_c\tilde{C}_c)^{-1}B_c \tag{13.20}$$

Hence the lemma. □

13.2.3 Finding a model

In this section, we shall describe how a model of the type stated in equation 13.7 can be found from the input/output data of the industrial process. The steps to obtain a model are

1. Pretreatment of data.
2. Obtaining an initial model using the unbiased version of the N4SID algorithm.
3. Parameter optimization using minimum-phase balanced parametrization as described in section 13.2.2.

We shall also discuss the model validation of the subspace models and optimized models in this section.

Pretreatment of data

Some standard treatments are applied to the raw input/output data before they are used for identification. These steps are

1. The first 3000 data points are removed in order to ensure that the process has reached the steady state.

2. The mean is subtracted from the data.

3. Filter the input/output data through an anti-aliasing filter of an appropriate cut-off frequency and sample the filtered data at a rate of once every 10 samples i.e. once every 10 seconds.

As a result of the treatments above, the data set now contains only 867 data points. We shall use the first 600 of them to obtain a model and the rest of it for model validation. We have plotted in figure 13.15 (page 146) the input/output data after step 2.

Finding an initial estimate

The major decision which we need to make at this step is to decide on the order of the model. As mentioned in section 11.3, the model order is determined by the dimension of a matrix whose column space coincides with that of the extended observability matrix. In other words, we can determine the model order by the number of significant singular values of this matrix. We have plotted in figure 13.14 (page 145) the 30 most significant singular values of this matrix. The numerical values of the first 12 of them are as follows:

```
14.6982, 2.5392, 0.8613, 0.4999, 0.2795, 0.2459, 0.1849, 0.1274, 0.1076,
0.0905, 0.0657, 0.0485
```

Note that the third singular value is already an order of magnitude less than the most significant one. We shall use two different model orders — fourth and sixth. The simulated outputs of these two models are plotted in figures 13.16 (page 147) and 13.17 (page 148). Note that these two models are able to predict the outputs very well, especially for the first two outputs.

Parameter optimization

The prediction error method will be used for parameter optimization. We shall first define the components of this parameter optimization problem.

1. The predictor is given by the following state space model

$$\left\{ \begin{array}{rcl} \hat{x}_{k+1} & = & A\hat{x}_k + Bu_k + K(y_k - \hat{y}_k) \\ \hat{y}_k & = & C\hat{x}_k + Du_k \end{array} \right. \tag{13.21}$$

2. The model set is $\mathcal{M} = \{(A,B,C,D,K) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n} \times \mathbb{R}^{p\times m} \times \mathbb{R}^{n\times p} :$ (A,K,C,I) is a discrete time stable and minimum-phase system whose minimum-phase characteristic values are distinct $\}$

model order	subspace models	optimized models
2	16.7696	9.9611
3	11.5606	7.0891
4	10.0386	7.0494
5	9.2587	6.6609
6	9.1455	6.5825
7	8.4016	6.5261
8	9.2280	6.4894
9	9.7619	6.5688
10	9.7531	6.5421

Table 13.2: Prediction errors for models of various orders.

3. The stable and minimal-phase system (A, K, C, I) will be parametrized using lemma 13.2. The matrices B and D will be directly parametrized i.e. $B = [b_{ij}]_{n \times m}$ and $D = [d_{ij}]_{p \times m}$.

4. Let θ denote the parameter vector. Define the objective function as

$$V(\theta) \;=\; \sum_{t=0}^{N} \varepsilon(t;\theta)^T \varepsilon(t;\theta) \quad \text{where} \quad \varepsilon(t;\theta) \;=\; y(t) - \hat{y}(t) \tag{13.22}$$

where $(N+1)$ is the number of data points.

We have applied the above parameter optimization algorithm to nine different state space models of different model orders obtained from the N4SID algorithm. We have summarised the prediction errors of these models before and after optimization in table 13.2. It is interesting to note that the prediction errors of the subspace models drop from model order 2 to 7 and then rise again. On the other hand, the prediction errors of the optimized models decrease in an almost steady manner. We have plotted in figures 13.18 (page 149) and 13.19 (page 150) the simulated response of the fourth and sixth order optimized models. Note that the optimized models are able to predict the third output better than the subspace models.

Model validation

We have simulated the following four models for the purpose of model validation: 4^{th} and 6^{th} order subspace models as well as 4^{th} and 6^{th} order optimized models. The results of these simulations are plotted on figures 13.20 (page 151) and 13.21 (page 152). For the first two outputs, all the four models are able to predict the outputs fairly well. However, the optimized models are able to predict the third output better than the subspace models.

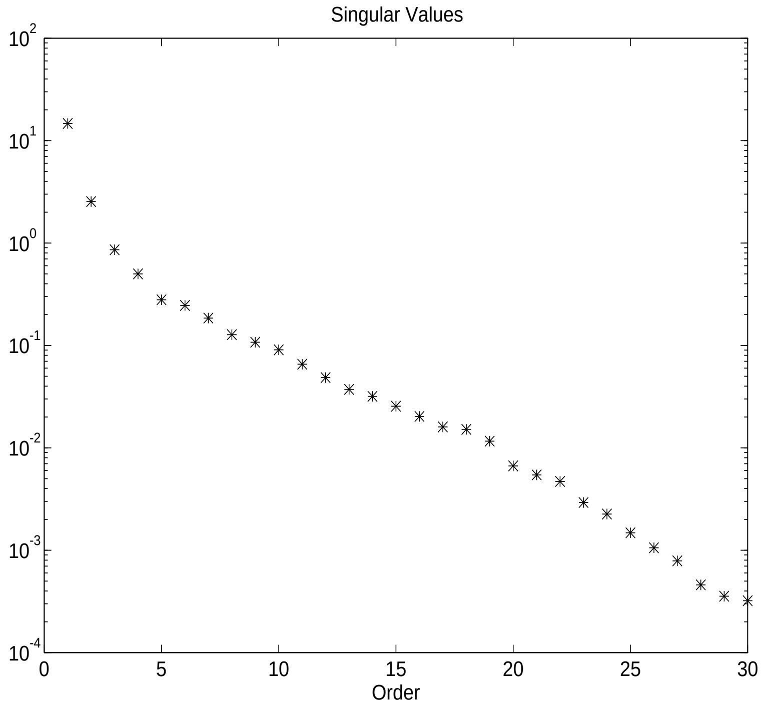


Figure 13.14: Singular values from N4SID.

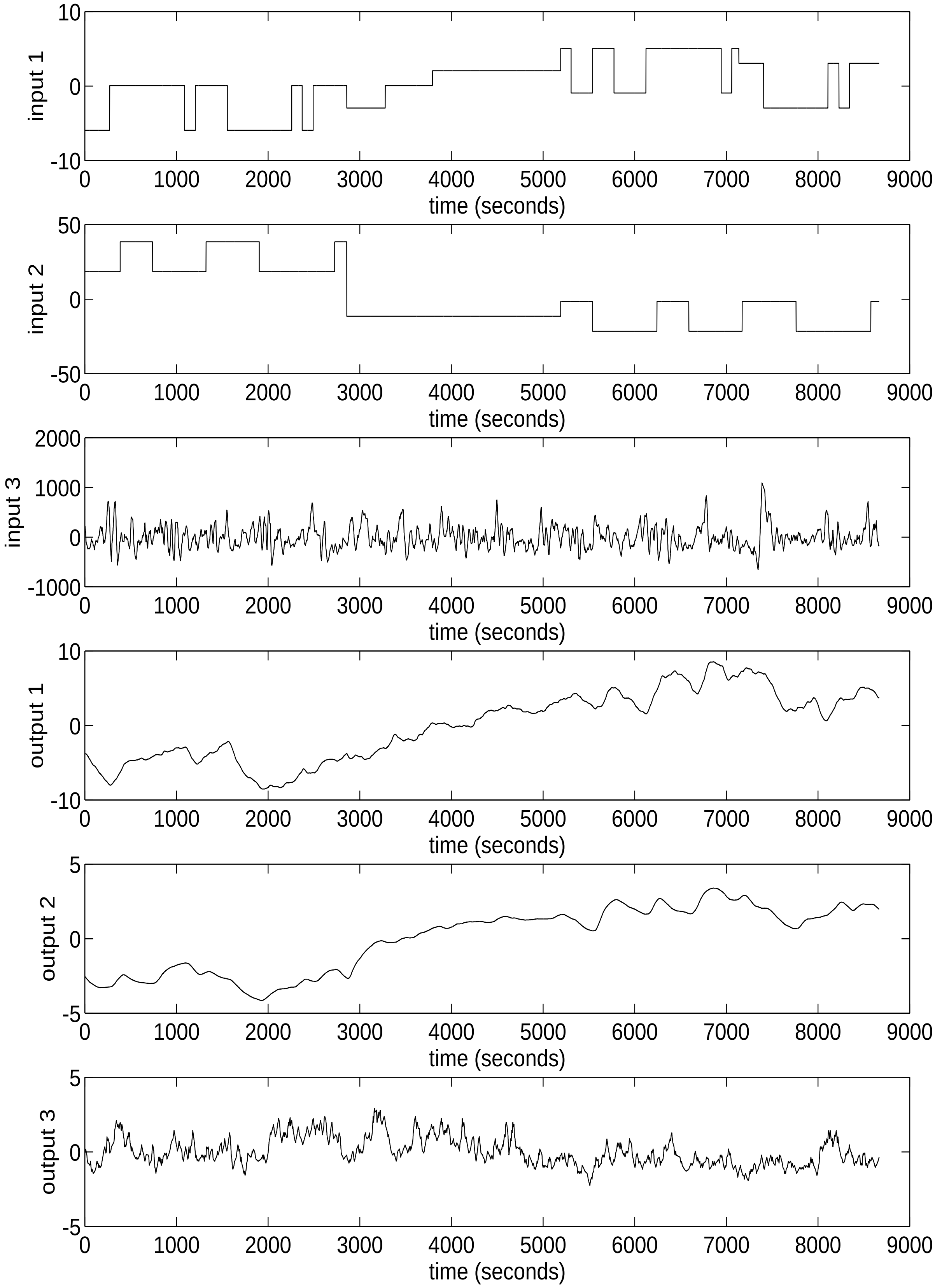


Figure 13.15: Process input/output data with the first 3000 points and mean removed.

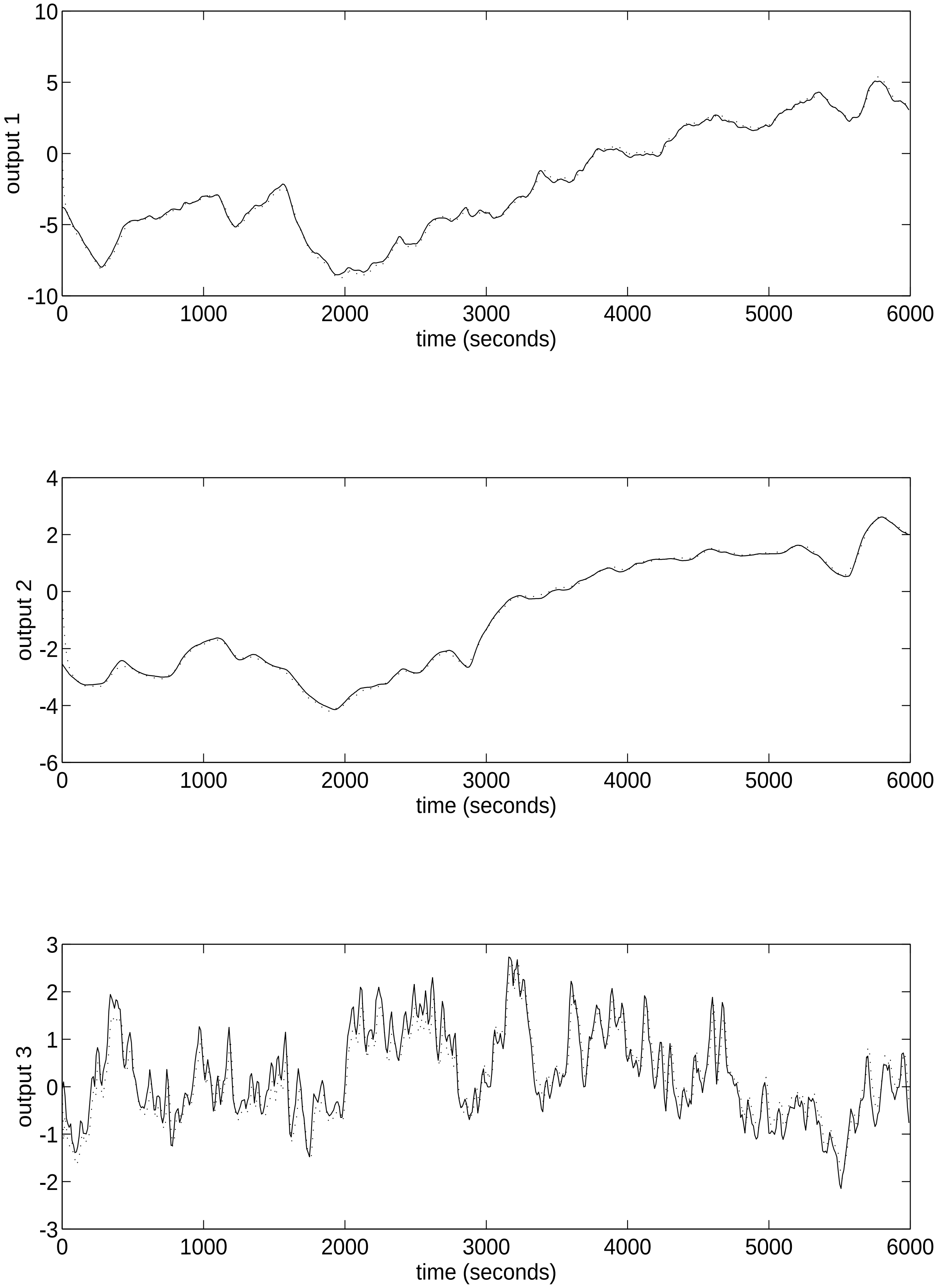


Figure 13.16: Process output (solid). Simulated output of the 4^{th} order subspace model (dotted).

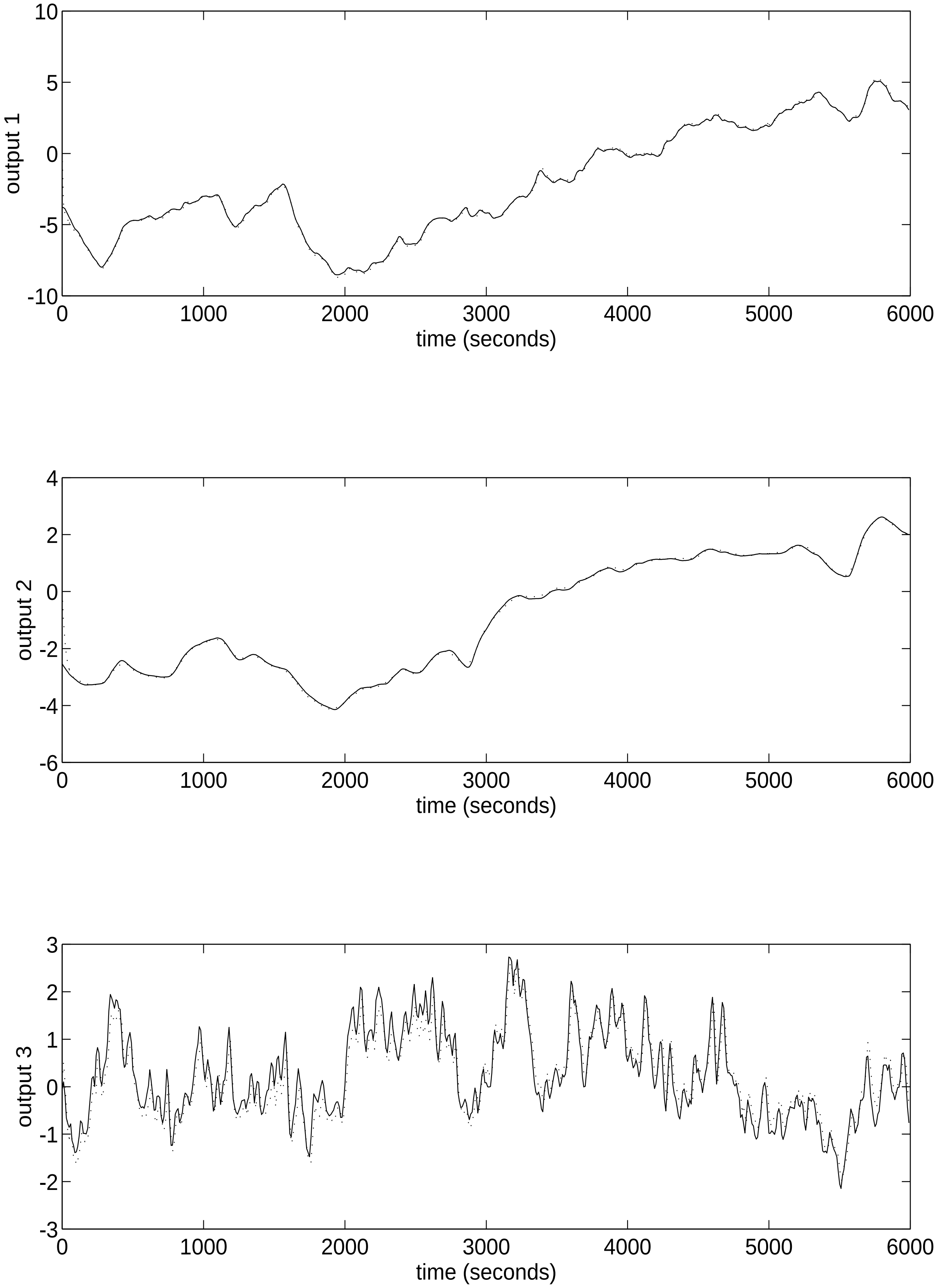


Figure 13.17: Process output (solid). Simulated output of the 6^{th} order subspace model (dotted).

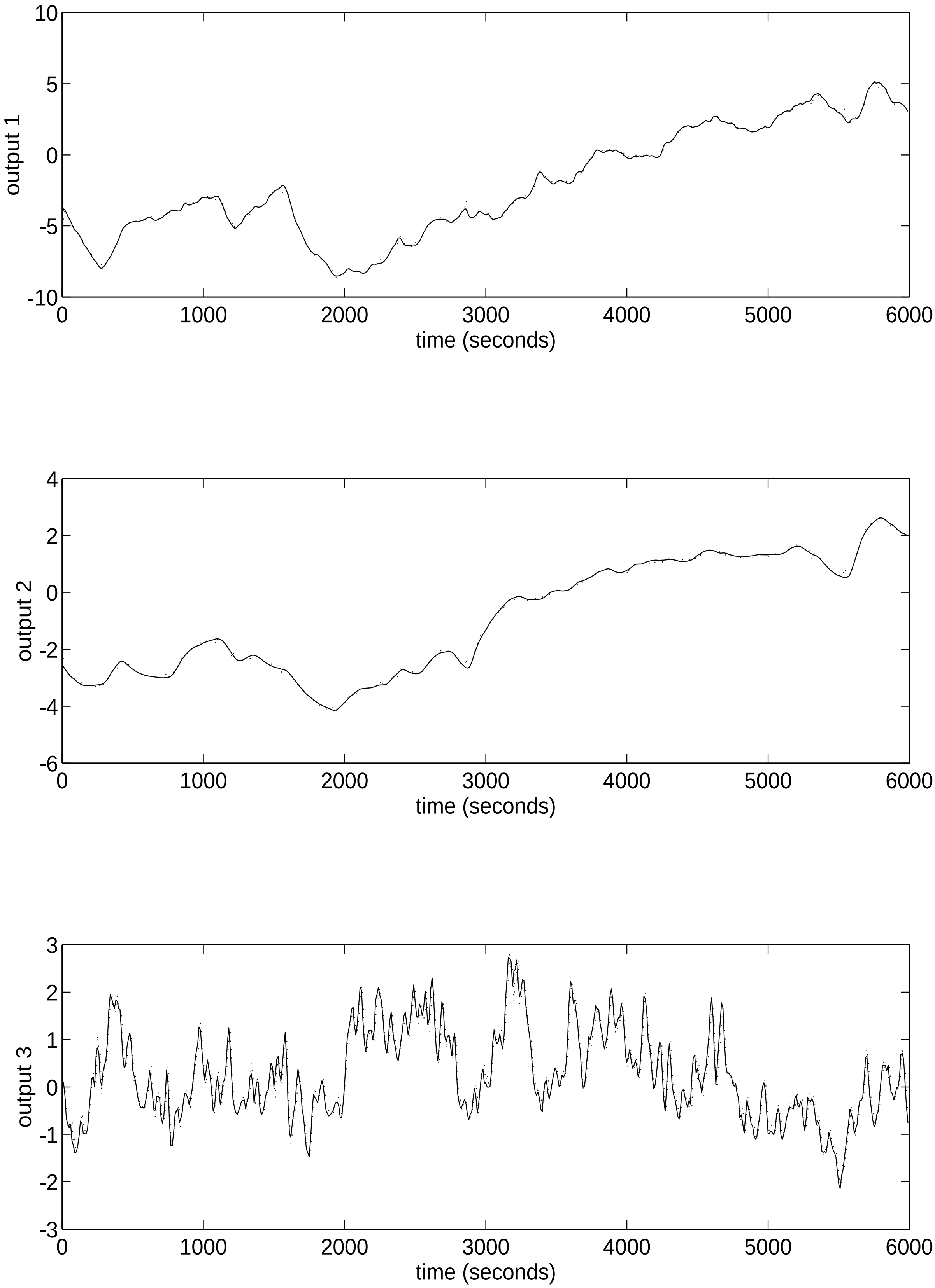


Figure 13.18: Process output (solid). Simulated output of the 4^{th} order optimized model (dotted).

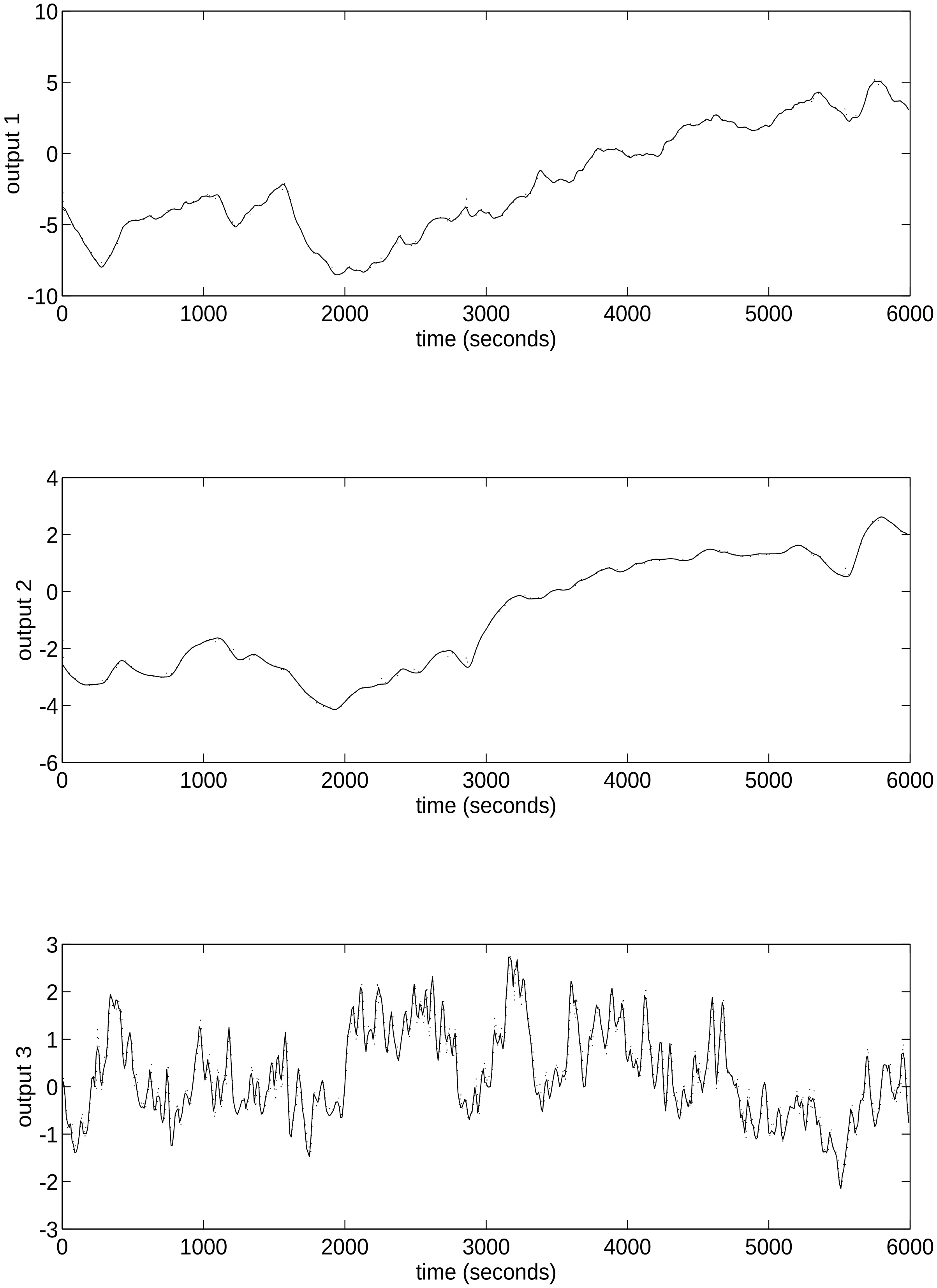


Figure 13.19: Process output (solid). Simulated output of the 6^{th} order optimized model (dotted).

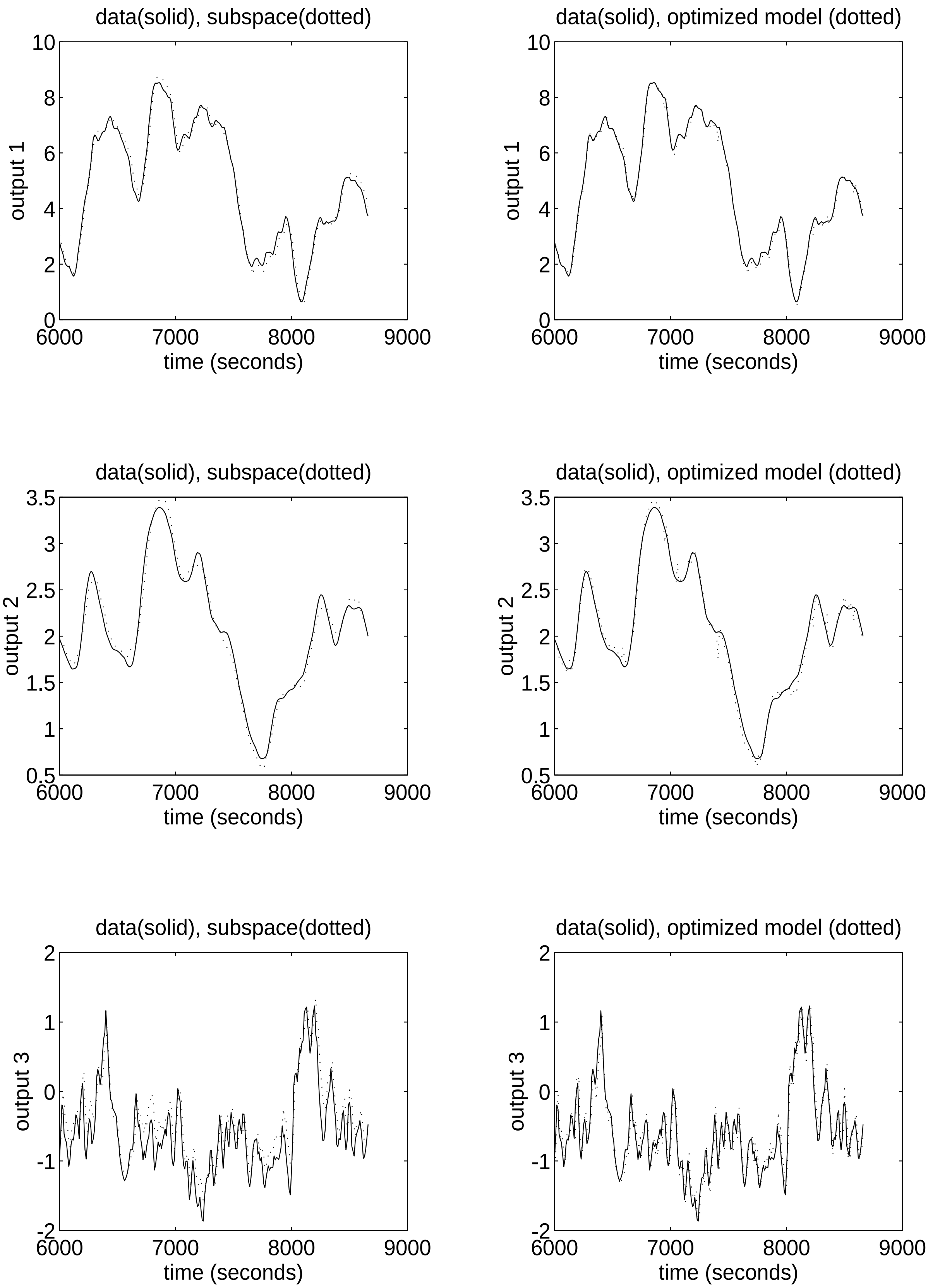


Figure 13.20: Model validation. Simulated output of the 4^{th} order subspace and optimized models.

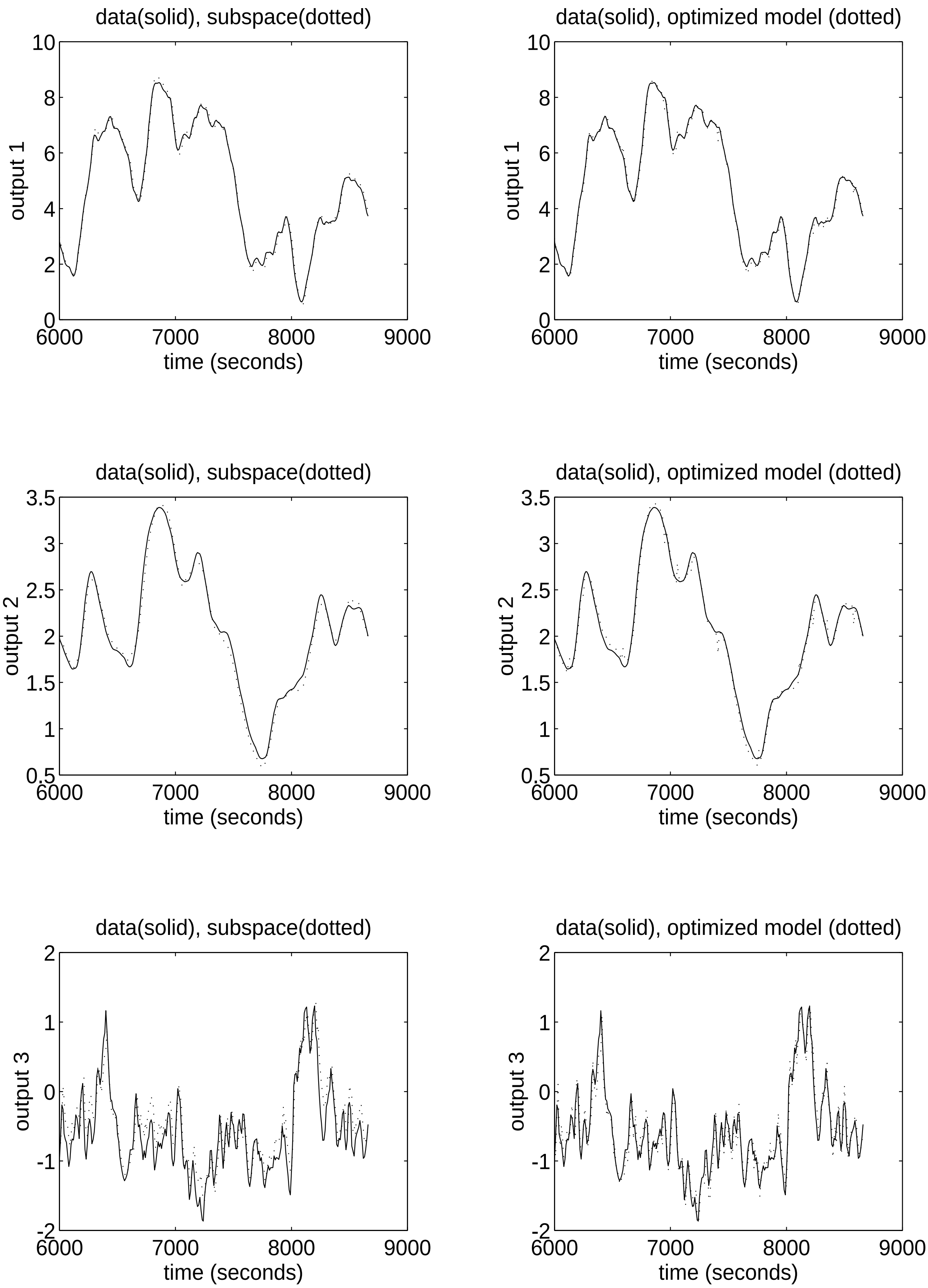


Figure 13.21: Model validation. Simulated output of the 6^{th} order subspace and optimized models.

Part III

Conclusions

Chapter 14

Conclusions

In this dissertation, we have looked at two topics related to the subject of system identification. In this chapter, we shall summarise the contributions we have made and also suggest areas of further research which are natural continuations of the work.

14.1 Contributions

The main contributions of this thesis are:

Part I

- We have proved that a number of classes of continuous time and discrete time finite dimensional linear time-invariant systems are real analytic diffeomorphic to each other. Explicit and implicit maps for these diffeomorphisms have also been derived. (Chapters 4 and 5)

- By using our results on the diffeomorphisms between classes of linear systems, we proved that the state bundles associated with these classes of systems are real analytic diffeomorphic to each other. (Chapter 6)

- The principal fibre bundles associated with a number of classes linear systems were shown to be diffeomorphic to each other. (Chapter 6) These results were used to show the (non)-existence of continuous canonical forms on these classes of linear systems. (Chapter 7).

- These results on diffeomorphisms were used to deduce the geometric properties of a number of classes of linear systems. (Chapter 7)

- We have demonstrated how these results on diffeomorphisms between classes of linear systems can be used to construct overlapping parametrizations and Riemannian gradient identification algorithms on classes of linear systems. (Chapter 7)

Part II

- An algorithm for the identification of stable linear time-invariant state space systems has been proposed. The first step of the algorithm uses either model reduction schemes or subspace methods to produce an initial model. This initial model is then supplied to a parameter optimization algorithm which uses balanced parametrizations for stable systems. This algorithm was successfully applied to two sets of industrial data.

- We have derived a local coordinate system for the generic structure of multivariable balanced canonical forms. (Chapter 10)

- An alternative proof which shows that Kung's algorithm is equivalent to discrete time balanced truncation was given. (Chapter 11)

- A version of the model reduction due to Al-Saggaf (with the parameter α set to -1) was shown to be useful in finding an initial state space model especially when a step input is used to excite the system. The other version of Al-Saggaf's algorithm (with α set to 1) was shown to give similar result to that obtained from Kung's algorithm. (Chapter 11)

- We have proposed a method of scaling the input/output data so that the model obtained from either Kung's or Al-Saggaf's algorithm fits the data more accurately. An algorithm to perform this type of scaling was proposed. This proposed algorithm can be implemented in an automatic manner. (Chapter 11 and Appendix B)

- We have shown how the gradient of the objective function can be computed. We have also derived the formulae for the derivatives of the state space matrices of the generic structure of the multivariable balanced canonical form. (Chapter 12)

14.2 Topics for further research

- The system identification algorithm which we have proposed uses only the generic structure of the canonical form. The problems of selecting an initial structure and switching to a different structure are still unresolved.

- Work should be done on improving the efficiency of the proposed system identification algorithm. Currently we are only using general purpose optimization software for solving the optimization problem, so tailor-made software which will possibly take into account the structure of the parametrizations should be considered. Also, we should look into a numerically efficient and accurate way to compute the gradient.

- Although we did not run into any numerical problems in using our proposed system identification algorithm, this does not imply that balanced parametrizations are a numerically well conditioned for parameter estimation. It will be interesting to find out how the way in which real-valued parameters enter the state space matrices affects the numerical conditioning of the parameter estimation algorithm.

Part IV

Appendices

Appendix A

Proofs of theorems in part II

Proof of theorem 11.1

In order to show that $\Sigma_n^{-1/2} U_n^T U_n^{\uparrow} \Sigma_n^{1/2}$ and $\Sigma_n^{1/2} \overleftarrow{V}_n V_n^T \Sigma_n^{-1/2}$ are equal, it is equivalent to show the following equality

$$\Sigma_n^{-1} U_n^T U_n^{\uparrow} \Sigma_n = \overleftarrow{V}_n V_n^T \tag{A.1}$$

We shall first establish the following few equalities.

1. Using the same notation as that used in section 11.1, the singular value decomposition of H_N can be written as

$$\begin{aligned} H_N &= \begin{bmatrix} U_n & \bar{U}_n \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \bar{\Sigma}_n \end{bmatrix} \begin{bmatrix} V_n \\ \bar{V}_n \end{bmatrix} \\ \Rightarrow \quad H_N \begin{bmatrix} V_n^T & \bar{V}_n^T \end{bmatrix} &= \begin{bmatrix} U_n & \bar{U}_n \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \bar{\Sigma}_n \end{bmatrix} \end{aligned}$$

   Comparing the first n columns on both sides of the above equation, we have

$$U_n \Sigma_n = H_N V_n^T \tag{A.2}$$

2. Now consider the pseudo-inverse of H_N,

$$\begin{aligned} H_N^{\dagger} &= \begin{bmatrix} V_n^T & \bar{V}_n^T \end{bmatrix} \begin{bmatrix} \Sigma_n^{-1} & 0 \\ 0 & \bar{\Sigma}_n^{\dagger} \end{bmatrix} \begin{bmatrix} U_n^T \\ \bar{U}_n^T \end{bmatrix} \\ \Rightarrow \quad \begin{bmatrix} V_n \\ \bar{V}_n \end{bmatrix} H_N^{\dagger} &= \begin{bmatrix} \Sigma_n^{-1} & 0 \\ 0 & \bar{\Sigma}_n^{\dagger} \end{bmatrix} \begin{bmatrix} U_n^T \\ \bar{U}_n^T \end{bmatrix} \end{aligned}$$

   Comparing the first n rows on both sides of the above equation, we have

$$\Sigma_n^{-1} U_n^T = V_n H_N^{\dagger} \tag{A.3}$$

3. For $n \leq r$,

$$\begin{aligned} V_n H_N^{\dagger} H_N &= \left(\begin{bmatrix} I_n & 0 \end{bmatrix} V\right) V^T \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} V \\ &= \begin{bmatrix} I_n & 0 \end{bmatrix} \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} V \\ &= \begin{bmatrix} I_n & 0 \end{bmatrix} V \text{ provided that } n \leq r \\ &= V_n \end{aligned} \tag{A.4}$$

4. Let A and $\bar{A}$ be the following matrices

$$A = \underbrace{\begin{bmatrix} 0 & I_p & & \\ & & \ddots & \\ & & & I_p \\ & & & 0 \end{bmatrix}}_{\text{size } Np \times Np} \tag{A.5}$$

$$\bar{A} = \underbrace{\begin{bmatrix} 0 & \dots & & 0 \\ I_m & & & \\ & \ddots & & \\ & & I_m & 0 \end{bmatrix}}_{\text{size } Nm \times Nm} \tag{A.6}$$

Due to the block structure of the Hankel matrix H_N, we have

$$AH_N = H_N \bar{A} \tag{A.7}$$

and due to the shifting properties of the matrices A and $\bar{A}$

$$U_n^{\uparrow} = AU_n \tag{A.8}$$

$$\overleftarrow{V}_n = V_n \bar{A} \tag{A.9}$$

Now we shall prove the equality given in equation A.1.

$$\begin{aligned} &\Sigma_n^{-1} U_n^T U_n^{\uparrow} \Sigma_n \\ &= \Sigma_n^{-1} U_n^T A U_n \Sigma_n && \text{by equation A.8} \\ &= V_n H_N^{\dagger} A H_N V_n^T && \text{by equations A.2 and A.3} \\ &= V_n H_N^{\dagger} H_N \bar{A} V_n^T && \text{by equation A.7} \\ &= V_n \bar{A} V_n^T && \text{by equation A.4} \\ &= \overleftarrow{V}_n V_n^T && \text{by equation A.9} \end{aligned}$$

Proof of theorem 11.2

The finite impulse response system whose Markov parameters are $(J_0, J_1, ..., J_N)$ can be generated by the state space realization

$$\left\{ \begin{array}{rcl} x[k+1] & = & Ax[k] + Bu[k] \\ y[k] & = & Cx[k] + Du[k] \end{array} \right. \tag{A.10}$$

where

$$A = \begin{bmatrix} 0 & I_p & & \\ & & \ddots & \\ & & & I_p \\ & & & 0 \end{bmatrix} \tag{A.11}$$

$$B = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_N \end{bmatrix} \tag{A.12}$$

$$C = \begin{bmatrix} I_p & 0 & \dots & 0 \end{bmatrix} \tag{A.13}$$

$$D = J_0 \tag{A.14}$$

Note that A is asymptotically stable and the realization is always observable but not necessarily controllable.

It can be shown that the controllability Gramian P and observability Gramian Q of (A, B, C, D) are respectively given by $H_N H_N^T$ and I_{Np} where H_N was given in equation 11.1.

Let the singular value decomposition of H_N be

$$H_N = U \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} V \tag{A.15}$$

where Σ_r is a $r \times r$ diagonal matrix containing all the positive singular values of H_N. The Gramians of (A, B, C, D) can be written as

$$P = U \begin{bmatrix} \Sigma_r^2 & 0 \\ 0 & 0 \end{bmatrix} U^T \qquad Q = I_{Np} \tag{A.16}$$

Under the state space transformation T, the state space matrices and the Gramians will be transformed as follows:

$$(A, B, C, D) \longrightarrow (TAT^{-1}, TB, CT^{-1}, D) \tag{A.17}$$

$$P \longrightarrow TPT^T \tag{A.18}$$

$$Q \longrightarrow T^{-T}QT^{-1} \tag{A.19}$$

To balance the system, we apply the transformation

$$T = \begin{bmatrix} \Sigma_r^{-\frac{1}{2}} & 0 \\ 0 & I \end{bmatrix} U^T \tag{A.20}$$

to the original state space matrices. With the same notation as that used in section 11.1, the balanced state space matrices and Gramians are given by

$$A_b = \begin{bmatrix} \Sigma_r^{-1/2}\, U_r^T U_r^{\uparrow} \Sigma_r^{1/2} & \Sigma_r^{-1/2}\, U_r^T \bar{U}_r^{\uparrow} \\ \bar{U}_r^T U_r^{\uparrow} \Sigma_r^{1/2} & \bar{U}_r^T \bar{U}_r^{\uparrow} \end{bmatrix} \tag{A.21}$$

$$B_b = \begin{bmatrix} \Sigma_r^{1/2}\, V_r^1 \\ 0 \end{bmatrix} \tag{A.22}$$

$$C_b = \begin{bmatrix} U_r^1 \Sigma_r^{1/2} & \bar{U}_r^1 \end{bmatrix} \tag{A.23}$$

$$D_b = J_0 \tag{A.24}$$

$$P_b = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \tag{A.25}$$

$$Q_b = \begin{bmatrix} \Sigma_r & 0 \\ 0 & I \end{bmatrix} \tag{A.26}$$

Note that the last $(Np - r)$ states of the above realization are uncontrollable. To obtain an n^{th} $(n \leq r)$ order model, we truncate the last $(Np - n)$ states away. This is equivalent to extracting the top n rows and leftmost n columns of A_b, the top n rows of B_b and the leftmost n columns of C_b. It can easily be shown that the reduced order state space model is in fact given by equations 11.7 to 11.10 .

The finite impulse response $\{J_0, J_1, \ldots, J_N\}$ can also be generated by the following state space realization

$$\bar{A} = \begin{bmatrix} 0 & \ldots & & 0 \\ I_m & & & \\ & \ddots & & \\ & & I_m & 0 \end{bmatrix} \tag{A.27}$$

$$\bar{B} = \begin{bmatrix} I_m \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{A.28}$$

$$\bar{C} = \begin{bmatrix} J_1 & J_2 & \ldots & J_N \end{bmatrix} \tag{A.29}$$

$$\bar{D} = J_0 \tag{A.30}$$

Note that this realization is asymptotically stable and always controllable but not necessarily observable. It can be shown that if we balance and truncate the above realization to order n, then we shall obtain the state space realization given by equations 11.11 to 11.14 .

Proof of theorem 11.3

We shall first define some notation. Let (A, B, C, D) be the state space realization as defined in equations A.11 to A.14. We shall use (A_b, B_b, C_b, D_b) to denote the balanced state space realization of (A, B, C, D). The expressions for A_b, B_b, C_b and D_b are given in equations A.21 to A.24. Using the same block structure as that given in equations A.21 to A.24, we shall write these matrices in the following form.

$$A_b = \begin{bmatrix} A_r & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{A.31}$$

$$B_b = \begin{bmatrix} B_r \\ B_2 \end{bmatrix} \tag{A.32}$$

$$C_b = \begin{bmatrix} C_r & C_2 \end{bmatrix} \tag{A.33}$$

$$D_b = D_r \tag{A.34}$$

Note that the state space realization (A_r, B_r, C_r, D_r) is in fact the r^{th} order Kung's model. The balanced Gramians of (A_b, B_b, C_b, D_b) are given by

$$P_b = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \tag{A.35}$$

$$Q_b = \begin{bmatrix} \Sigma_r & 0 \\ 0 & I \end{bmatrix} \tag{A.36}$$

It can be shown that the balanced state space realization of $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$ (defined by equations A.27 to A.30) is given by

$$\bar{A}_b = \begin{bmatrix} A_r & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \tag{A.37}$$

$$\bar{B}_b = \begin{bmatrix} B_r \\ \bar{B}_2 \end{bmatrix} \tag{A.38}$$

$$\bar{C}_b = \begin{bmatrix} C_r & \bar{C}_2 \end{bmatrix} \tag{A.39}$$

$$\bar{D}_b = D_r \tag{A.40}$$

where (A_r, B_r, C_r, D_r) is the r^{th} order Kung's model. The balanced Gramians are

$$\bar{P}_b = \begin{bmatrix} \Sigma_r & 0 \\ 0 & I \end{bmatrix} \tag{A.41}$$

$$\bar{Q}_b = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \tag{A.42}$$

We shall now prove the properties of the full order Kung's model (A_r, B_r, C_r, D_r).

Asymptotic stability

The observability Gramian of (A_b, C_b) is Q_b,

$$\begin{bmatrix} \Sigma_r & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} A_r & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^T \begin{bmatrix} \Sigma_r & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_r & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} C_r^T \\ C_2^T \end{bmatrix} \begin{bmatrix} C_r & C_2 \end{bmatrix} \quad \text{(A.43)}$$

It was proved in [60] that if a system is stable and that *either* the controllability Gramian or the observability Gramian is nonsingular and diagonal, then its balanced truncation is asymptotically stable. Since the Gramian Q_b is non-singular and diagonal, hence A_r is asymptotically stable.

Minimality and balancedness

Since (A_b, B_b) has controllability Gramian P_b,

$$\begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} A_r & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} A_r & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^T = \begin{bmatrix} B_r \\ B_2 \end{bmatrix} \begin{bmatrix} B_r \\ B_2 \end{bmatrix}^T \quad \text{(A.44)}$$

and the top-left block of the equation gives

$$\Sigma_r - A_r \Sigma_r A_r^T = B_r B_r^T \quad \text{(A.45)}$$

Since A_r is asymptotically stable, (A_r, B_r) is controllable and the controllability Gramian is Σ_r.

The observability Gramian of $(\bar{A}_b, \bar{C}_b)$ is $\bar{Q}_b$,

$$\begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} A_r & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix}^T \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} A_r & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} = \begin{bmatrix} C_r^T \\ \bar{C}_2^T \end{bmatrix} \begin{bmatrix} C_r & \bar{C}_2 \end{bmatrix} \quad \text{(A.46)}$$

and the top-left hand block of the equation gives

$$\Sigma_r - A_r^T \Sigma_r A_r = C_r^T C_r \quad \text{(A.47)}$$

Since A_r is asymptotically stable, (A_r, C_r) is observable and the observability Gramian is Σ_r. Thus (A_r, B_r, C_r, D_r) is balanced with Gramian Σ_r.

Spectral norm of A_r

The result on the spectral norm of A_r follows directly from the following lemma which is taken from [60].

Lemma A.1 *Let (A, B, C, D) be a discrete time balanced realization with Gramian Σ, then $\|A\| \leq 1$. If Σ has distinct diagonal entries, then $\|A\|$ is strictly less than unity.*

Exact realization

The observability Gramian of (A_b, C_b) is Q_b and the top-left hand block of equation A.43 gives

$$\Sigma_r - A_r^T \Sigma_r A_r - A_{21}^T A_{21} \;=\; C_r^T C_r \tag{A.48}$$

Comparing equations A.47 and A.48, we have

$$\begin{aligned} A_{21}^T A_{21} &= 0 \\ \Rightarrow \quad A_{21} &= 0 \end{aligned} \tag{A.49}$$

Since A_{21} is 0 and recall from equation A.22 that B_2 is also 0, hence the last $(Np - r)$ states of (A_b, B_b) are uncontrollable. Note that these states are observable but uncontrollable, hence the impulse response of (A_r, B_r, C_r, D_r) is identical to that of (A_b, B_b, C_b, D_b). In other words, (A_r, B_r, C_r, D_r) realizes the finite impulse response $\{J_0, J_1, \ldots, J_N\}$.

Appendix B

Algorithm for scaling

In chapter 11, we discussed the effect of scaling the data on the fit of the model obtained from Kung's algorithm as well as that from Al-Saggaf's model reduction algorithm. In section 11.1.3, we put forward the conjecture that if we scale the data in such a way that the energy content in each input-output channel is of a similar order of magnitude, then a better model will normally be obtained. This conjecture has so far been confirmed by simulation examples (for example, see section 11.1.4) and also by the case study on the simulation model of fractional distillation column given in chapter 13. However, we are still lacking of formal support for this conjecture.

In this appendix, we shall present an algorithm which finds the input- and output-scaling factors such that the energy content in each of the input-output channels is of a similar order of magnitude. The procedure below actually gives an optimal scaling in the following sense: if e_h and e_l are respectively the highest and lowest energy levels among all the input-output channels after scaling has been performed, then the ratio e_h to e_l is the smallest possible. However, since we do not yet have a concrete idea on how scaling affects the Kung's algorithm or the Al-Saggaf's, except that we know from our experience that scaling of this type does improve the fit, we do not yet know whether this form of scaling does give an optimal fit to the input-output channels in any sense. Another feature of this algorithm is that it can be implemented in an automatic manner by a computer programme.

This appendix is divided into 2 parts. In the first part, we shall illustrate our scaling procedure using a 3-output 3-input system with the following general 'energy matrix':

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \tag{B.1}$$

where e_{ij} is the energy content of the input-output channel connecting the j^{th} input to the i^{th} output. This procedure can easily be generalized to use with systems of a larger dimension. In the second part, a numerical example of this scaling procedure will be given.

Algorithm

The scaling procedure aims at finding the diagonal matrices

$$D_i = diag(d_{i1}, d_{i2}, d_{i3}) \qquad D_o = diag(d_{o1}, d_{o2}, d_{o3}) \tag{B.2}$$

such that the elements in the scaled energy matrix $E_s = D_o * E * D_i$ are of a similar order of magnitude. The procedure is divided into the following 2 steps.

1. In this step, we shall show that by using the appropriate input- and output-scaling matrices, we can transform the matrix E into one of the following two forms or forms obtained by swapping rows of these two forms.

$$\begin{bmatrix} \alpha & \beta & 1 \\ \delta & 1/\beta & 1 \\ 1/\alpha & \gamma & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \alpha & \beta & 1 \\ 1/\alpha & 1/\beta & 1 \\ \delta & \gamma & 1 \end{bmatrix} \tag{B.3}$$

   where γ is between α and $1/\alpha$, and δ is between β and $1/\beta$.

   In order to bring the matrix E to the above form, we use the scaling matrices D_{o1} and D_{i1}, as follows.

   (a) Let

$$\begin{aligned} D_{o1} &= diag(1/e_{13}, 1/e_{23}, 1/e_{33}) \quad \text{and} && \text{(B.4)} \\ E_1 &= D_{o1} * E && \text{(B.5)} \end{aligned}$$

   Note that the last column of E_1 is $[1, 1, 1]^T$.

   (b) Let

$$\begin{aligned} D_{i1} &= diag(\zeta, \nu, 1) \quad \text{and} && \text{(B.6)} \\ E_2 &= E_1 * D_{i1} = D_{o1} * E * D_{i1} && \text{(B.7)} \end{aligned}$$

   where ζ is chosen in the following way: let s and l be respectively the smallest and largest element in the first column of E_1, then ζ is the reciprocal of the geometric mean of s and l. In other words, ζ is chosen such that the geometric mean of the largest and smallest element in the first column of E_2 is 1. ν will also be chosen in the same manner. Now E_2 is in one of the forms mentioned above.

2. The way in which this step will proceed depends on the form of the matrix E_2, however the basic principle behind is rather similar. So, instead of going through all the different forms of E_2, we shall assume that it takes the following form

$$E_2 = \begin{bmatrix} \alpha & \beta & 1 \\ \delta & 1/\beta & 1 \\ 1/\alpha & \gamma & 1 \end{bmatrix} \tag{B.8}$$

   Also, we shall assume without loss of generality that $\beta \geq \alpha \geq 1$. Instead of looking at the matrix E_2, we shall look at the matrix E_2^{log} which is defined as

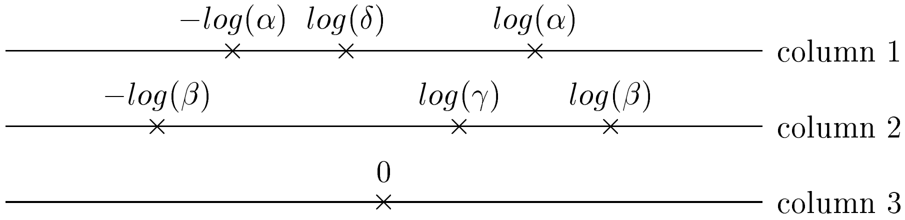
$$E_2^{log} = \begin{bmatrix} log(\alpha) & log(\beta) & 0 \\ log(\delta) & -log(\beta) & 0 \\ -log(\alpha) & log(\gamma) & 0 \end{bmatrix} \tag{B.9}$$

This turns the problem of finding the optimal ratio of the largest element in E_2 to that of the smallest in the matrix into the problem of minimizing the difference between the largest element and the smallest element in E_2^{log}. We shall first show that input-scaling, which corresponds to scaling of the columns, does not decrease the ratio between the largest and smallest elements in E_2. Note that the following 3 actions have the same effect.

(a) Multiplying the i^{th} column of E_2 by μ.

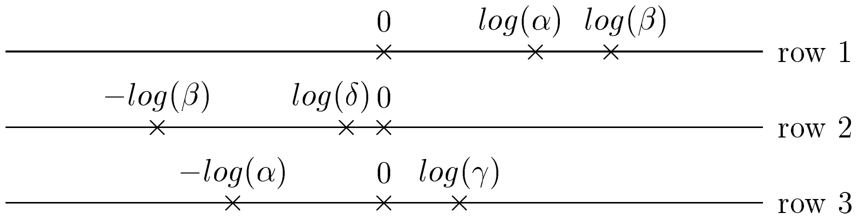
(b) Adding $log(\mu)$ to each element in the i^{th} column of E_2^{log}.

(c) Shifting the line 'column i' in the following figure by a distance of $log(\mu)$ to the right.



It can easily be seen that the distance between the largest and the smallest elements in E_2^{log} either stays the same or increases by the action of shifting the lines. In other words, input-scaling does not improve the ratio which we wish to minimize.

We shall now find the output-scaling matrix which will bring E_2 to an optimally scaled matrix. We first plot the rows of E_2^{log} in the following figure.



As output-scaling is identical to scaling the rows, the following 3 actions have the same effect.

(a) Multiplying the i^{th} row of E_2 by μ.

(b) Adding $log(\mu)$ to each element in the i^{th} row of E_2^{log}.

(c) Shifting the line 'row i' in the figure above by a distance of $log(\mu)$ to the right.

It can be shown that if we align the leftmost points of the three lines in the figure above by

(a) shifting the line 'row 2' to the right by $log(\beta)$, and

(b) shifting the line 'row 3' to the right by $log(\alpha)$, then optimal scaling is achieved.

Let D_{o2} be the output-scaling which brings E_2 to the optimally scaled matrix E_{opt}, i.e. $E_{opt} = D_{o2}E_{int}$, then the overall scaling matrices are

$$D_i = D_{i1} \quad \text{and} \quad D_o = D_{o2}D_{o1} \tag{B.10}$$

Numerical example

Given the following energy matrix

$$E = \begin{bmatrix} 0.1648 & 0.0367 & 0.0073 \\ 0.0366 & 0.0175 & 0.0012 \\ 0.2587 & 0.0775 & 0.0069 \end{bmatrix}, \tag{B.11}$$

we would like to find scaling factors

$$D_i = diag(d_{i1}, d_{i2}, d_{i3}) \qquad D_o = diag(d_{o1}, d_{o2}, d_{o3}) \tag{B.12}$$

such that the elements in the scaled energy matrix $E_s = D_o * E * D_i$ are of a similar order of magnitude.

Let D_{o1} be the matrix $diag(1/e_{13}, 1/e_{23}, 1/e_{33})$, then

$$E_1 = D_{o1} * E \tag{B.13}$$

$$= \begin{bmatrix} 22.5402 & 5.0137 & 1.0000 \\ 30.2507 & 14.4372 & 1.0000 \\ 37.4540 & 11.2132 & 1.0000 \end{bmatrix} \tag{B.14}$$

In order to bring E to one of the standard forms, we need to post-multiply E_1 by $D_{i1} = diag(\zeta, \nu, 1)$, where

$$\begin{aligned} \zeta &= (\sqrt{37.4540 \times 22.5402})^{-1} &= 0.03442 \\ \nu &= (\sqrt{14.4372 \times 5.0137})^{-1} &= 0.1175, \end{aligned}$$

to obtain

$$E_2 = E_1 * D_{i1} \tag{B.15}$$

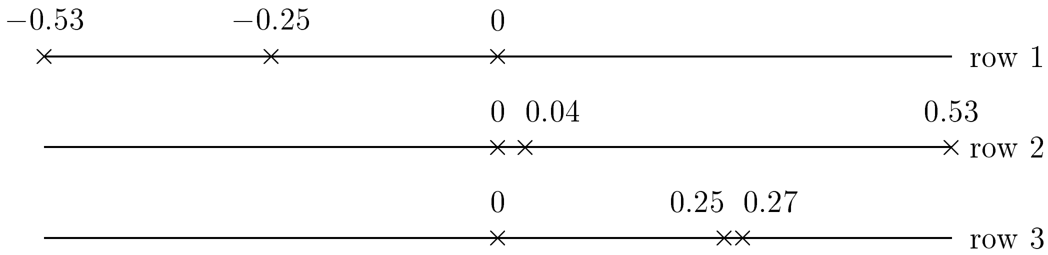
$$= \begin{bmatrix} 0.7758 & 0.5893 & 1.0000 \\ 1.0411 & 1.6969 & 1.0000 \\ 1.2891 & 1.3180 & 1.0000 \end{bmatrix} \tag{B.16}$$

E_2 is now in one of the standard form with $E_2(1,1) = E_2(3,1)^{-1}$ and $E_2(1,2) = E_2(2,2)^{-1}$.

Now we take the logarithm of the elements in E_2

$$E_2^{log} = \begin{bmatrix} -0.2539 & -0.5288 & 0 \\ 0.0403 & 0.5288 & 0 \\ 0.2539 & 0.2761 & 0 \end{bmatrix} \tag{B.17}$$

The elements in the three rows of E_2^{log} are plotted in the following diagram.



Now the problem becomes one of minimizing the distance between the leftmost and rightmost points in the above diagram by shifting the lines horizontally. A solution is provided by moving the first row to the right by 0.53 units and this is equivalent to multiplying the first row of E_2 by $0.5893^{-1} = 1.6969$. Hence $D_{o2} = diag(1.6969, 1, 1)$ and

$$E_{opt} \;=\; \begin{bmatrix} 1.3164 & 1.0000 & 1.6969 \\ 1.0411 & 1.6969 & 1.0000 \\ 1.2891 & 1.3180 & 1.0000 \end{bmatrix} \tag{B.18}$$

The overall scaling matrices are

$$D_i = \quad D_{i1} \quad = diag(0.0344, 0.1175, 1) \tag{B.19}$$
$$D_o = \quad D_{o1} * D_{o2} \quad = diag(232.0912, 826.8739, 144.7502) \tag{B.20}$$

Note that the ratio of the largest element to the smallest in E is $0.2587/0.0012 = 215.6$ while that for E_{opt} is $1.6969/1 = 1.6969$.

Bibliography

[1] U.M. Al-Saggaf. *On model reduction and control of discrete time systems.* PhD thesis, Stanford University, Stanford, California, 1986.

[2] U.M. Al-Saggaf and G.F. Franklin. Model reduction via balanced realizations: an extension and frequency weighting techniques. *IEEE Transactions on Automatic Control*, 33:687–692, 1988.

[3] B.D.O. Anderson. Weighted Hankel norm approximation: calculations of bounds. *Systems and Control Letters*, 7(4):247–255, 1986.

[4] B.D.O. Anderson and S. Vongpanitlerd. *Network analysis and synthesis.* Prentice Hall, Englewood Cliffs, N.J., 1973.

[5] M.A. Armstrong. *Basic Topology.* McGraw-Hill, London, 1979.

[6] A.C.P.M. Backx. *Identification of an industrial process: A Markov parameter approach.* PhD thesis, Technische Universiteit Eindhoven, The Netherlands, 1987.

[7] W.M. Boothby. *An introduction to differential manifolds and Riemannian geometry 2nd edition.* Academic Press, Inc, San Diego, 1986.

[8] R.W. Brockett. Some geometric questions in the theory of linear systems. *IEEE Transactions on Automatic Control*, 21:449–455, 1976.

[9] C.I. Byrnes and T.E. Duncan. On certain topological invariants arising in systems theory. In Hilton P. and Young G., editors, *New Directions in Applied Mathematics*, pages 29–71. Springer-Verlag, New York, 1981.

[10] C.I. Byrnes and N. Hurt. On the moduli of linear dynamical systems. *Advances in Mathematical Studies in Analysis*, 4:83–122, 1979.

[11] Y.M. Cho, G. Xu, and T. Kailath. Fast recursive identification of state space models via exploitation of displacement structure. *Automatica*, 30(1):45–59, 1994.

[12] J.M.C. Clark. The consistent selection of local coordinates in linear system identification. In *Proceedings Joint Automatic Control Conference*, Purdue, 1976.

[13] A.A.H. Damen, P.M.J. van den Hof, and A.K. Hajdasinski. Approximate realization based upon an alternative to the Hankel matrix: the Page matrix. *Systems and Control Letters*, 2:202–208, 1982.

[14] D.F. Delchamps. *The geometry of spaces of linear systems with an application to the identification problem.* PhD thesis, Harvard University, 1982.

[15] D.F. Delchamps. Global structure of families of multivariable linear systems with an application to identification. *Mathematical Systems Theory*, 18:329–380, 1985.

[16] D.F. Delchmaps. Analytic stabilization and the algebraic Riccati equation. In *Proceedings of the 22nd Conference on Decision and Control*, 1983.

[17] D. Enns. *Model reduction for control system design*. PhD thesis, Stanford University, Stanford, California, 1984.

[18] P.E. Gill, W. Murray, and M.H. Wright. *Practical optimization*. Academic Press, London, 1981.

[19] K. Glover. Some geometric properties of linear systems with implications in identification. In *Proceedings 6th IFAC World Congress*, Boston, Massachusetts, 1975.

[20] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^{∞}-error bounds. *International Journal of Control*, 39:1115–1193, 1984.

[21] K. Glover and J.C. Willems. Parametrization of linear dynamical systems: Canonical forms and identifiability. *IEEE Transactions on Automatic Control*, 19:640–645, 1974.

[22] A.K. Hajdasinski, P. Eykhoff, A.A.H. Damen, and A.J.W. van den Boom. The choice and use of different model sets for system identification. In *Proceedings 6th IFAC Symposium on Identification and System Parameter Estimation*, Washington D.C., 1982.

[23] B. Hanzon. On a coordinate free prediction error algorithm for system identification. In C.I. Byrnes and A. Lindquist, editors, *Modelling, Identification and Control*, pages 595–604. North-Holland, Amsterdam, 1986.

[24] B. Hanzon. Riemannian geometry on families of linear systems, the deterministic case. Technical Report 88-62, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, 1988.

[25] B. Hanzon. *Identifiability, recursive identification and spaces of linear dynamical systems*. Numbers 63 and 64 in CWI Tracts. Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 1989.

[26] B. Hanzon. On the differential manifold of fixed order stable linear system. *Systems and Control Letters*, 13:345–352, 1989.

[27] B. Hanzon. A new balanced canonical form for stable multivariable systems. Research Memorandum 1993-13, Vrije Universiteit, Amsterdam, 1993.

[28] B. Hanzon and R. Ober. Overlapping block-balanced canonical forms and parametrizations: the stable SISO case. In *Proceedings of the 31st IEEE Conference Decision and Control*, pages 2835–2838, 1992.

[29] M. Hazewinkel. Degenerating families of linear dynamical systems I. In *Proceedings of the 16th IEEE Conference on Decision and Control*, pages 258–264, 1977.

[30] M. Hazewinkel. Moduli and canonical forms for linear dynamical systems II: the topological case. *Mathematical Systems Theory*, 10:363–385, 1977.

[31] M. Hazewinkel. On families of linear systems: Degeneration phenomena. In *Lectures in Applied Mathematics*, volume 18, pages 157–189. American Mathematical Society, Providence, 1980.

[32] M. Hazewinkel and R.E. Kalman. On invariants, canonical forms, and moduli for linear constant, finite-dimensional dynamical systems. In *Lecture Notes in Econ.-Mathematical System Theory*, volume 131, pages 48–60. Springer-Verlag, New York, 1976.

[33] U. Helmke. A global parametrization of asymptotically stable linear systems. *Systems and Control Letters*, 13:383–389, 1989.

[34] D. Hinrichsen and A.J. Pritchard. An improved error estimate for reduced-order models of discrete-time systems. *IEEE Transactions on Automatic Control*, 35(3):317–320, 1990.

[35] E.A. Jonckheere and L.M. Silverman. A new set of invariants for linear systems - applications to reduced order compensator design. *IEEE Transactions on Automatic Control*, 28:953–964, 1983.

[36] T. Kailath. *Linear Systems.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1980.

[37] S.K. Kung. A new low-order approximation algorithm via singular value decomposition. In *Proceedings 12th Asilomar Conference on Circuits, Systems and Computers*, pages 705–714, 1978.

[38] W.E. Larimore. Canonical variate analysis in identification, filtering and adaptive control. In *Proceedings of the 29th Conference on Decision and Control*, pages 596–604, Hawaii, 1990.

[39] G.A. Latham and B.D.O. Anderson. Frequency-weighted optimal Hankel norm approximation of stable transfer functions. *Systems and Control Letters*, 5(4):229–236, 1985.

[40] L. Ljung. *System Identification: Theory for the User.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[41] J.M. Maciejowski. Balanced realizations in system identification. In *Proceedings IFAC Symposium of System Identification*, York, 1985.

[42] J.M. Maciejowski and R.J. Ober. Balanced parametrizations and canonical forms for system identification. In *Proceedings IFAC Symposium of System Identification*, Beijing, 1988.

[43] B.P. McGinnie. *A balanced view of system identification.* PhD thesis, Cambridge University, Cambridge, United Kingdom, 1984.

[44] B.P. McGinnie, R.J. Ober, and J.M. Maciejowski. Balanced parametrizations in time-series identification. In *Proceedings of the 29th Conference on Decision and Control*, Hawaii, 1990.

[45] M. Moonen, B. De Moor, L. Vandenberghe, and J. VandeWalle. On- and off-line identification of linear state-space models. *International Journal of Control*, 49(1):219–232, 1989.

[46] B.C. Moore. Principal component analysis in linear systems : Controllability, observability , and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.

[47] C.T. Mullis and R.A. Roberts. Synthesis of minimum roundoff noise fixed point digital filter. *IEEE Transactions on Circuits and Systems*, 23:551–562, 1976.

[48] F.D. Murnaghan. The orthogonal and symplectic groups. Communications of the Dublin Institute for Advanced Studies. Series A 13, Dublin Institute for Advanced Studies, Dublin, 1958.

[49] F.D. Murnaghan. *The Unitary and Rotation Groups.* Spartan Books, Washington, 1962.

[50] R. Ober. Private communication.

[51] R. Ober. Topology of the set of asymptotically stable minimal systems. *International Journal of Control*, 46:263–280, 1987.

[52] R. Ober. Connectivity properties of classes of linear systems. *International Journal of Control*, 50:2049–2073, 1989.

[53] R.J. Ober. Balanced realizations: Canonical form, parametrization, model reduction. *International Journal of Control*, 46:643–670, 1987.

[54] R.J. Ober. Balanced parametrization of classes of linear systems. *SIAM Journal Control and Optimization*, 29:1251–1287, 1991.

[55] R.J. Ober and D. McFarlane. Balanced canonical forms for minimal systems: a normalized coprime factor approach. *Linear Algebra and Applications*, pages 23–64, October 1989.

[56] P.C. Opdenacker and E.A. Jonckheere. A contraction mapping preserving balanced reduction scheme and its infinity norm bounds. *IEEE Transactions on Circuits and Systems*, 35:184–189, 1988.

[57] R.L.M. Peeters. Identification on a manifold of systems. Research Memorandum 1992-7, Vrije Universiteit, Amsterdam, 1992.

[58] R.L.M. Peeters. *System Identification Based on Riemannian Geometry Theory and Algorithms.* PhD thesis, Vrije Universiteit, Amsterdam, The Netherlands, 1994.

[59] R.L.M. Peeters and B. Hanzon. The Riemannian interpretation of Gauss-Newton and scoring, with application to system identification. Research Memorandum 1992-22, Vrije Universiteit, Amsterdam, 1992.

[60] L. Pernebo and L.M. Silverman. Model reduction via balanced state space reduction. *IEEE Transactions on Automatic Control*, 27(2):382–387, 1982.

[61] W.J. Pervin. *Foundations of General Topology.* Academic Press Inc., London, 1964.

[62] J.M.A. Scherpen. Balancing for non-linear systems. *Systems and Control Letters*, 21(2):143–153, 1993.

[63] L.M. Silverman and M. Bettayeb. Optimal approximation of linear systems. In *Proceedings Joint Automatic Control Conference*, San Francisco, CA, 1980. Paper FA8-A.

[64] T. Söderström and P. Stoica. *System Identification.* Prentice-Hall, Inc., Hemel Hempsted, 1989.

[65] M. Spivak. *Calculus on manifolds.* Benjamin, Reading, NY, 1965.

[66] M. Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 2. Brandeis, 1970.

[67] A-J. van der Veen, E.F. Deprettere, and A.L. Swindlehurst. Subspace-based signal analysis using singular value decomposition. *Proceedings of the IEEE*, 81(9):1277–1308, 1993.

[68] A.J.M. van Overbeek and L. Ljung. On-line structure selection for multivariable state space models. *Automatica*, 18:529–543, 1982.

[69] P. van Overschee and B. De Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, 1993.

[70] P. van Overschee and B. De Moor. A unifying theorem for three subspace system identification algorithms. Technical Report ESAT-SISTA/TR 1993-50, Department of Electrical Engineering, Katholieke Universiteit Leuven, Belgium, 1993. Draft.

[71] P. van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[72] M. Verhaegen and E. Deprettere. Subspace model identification. In E.F. Deprettere and A.J. van der Veen, editors, *Algorithms and parallel VLSI architectures*, volume II. North-Holland, Amsterdam, 1991.

[73] M. Verhaegen and P. Dewilde. Subspace model identification Part I. The output-error state space model identification class of algorithms. *Internal Journal of Control*, 56(5):1187–1210, 1992.

[74] M. Verhaegen and P. Dewilde. Subspace model identification Part II. Analysis of the elementary output-error state space model identification class of algorithms. *Internal Journal of Control*, 56(5):1211–1241, 1992.

[75] E.I. Verriest. The structure of multivariable balanced realizations. In *IEEE International Symposium on Circuits and Systems*, pages 110–113, Newport Beach, CA, 1983.

[76] E.I. Verriest and T. Kailath. On generalized balanced realizations. *IEEE Transactions on Automatic Control*, 28(8):833–844, 1983.

[77] B. Wahlberg. On model reduction in system identification. In *Proceedings American Control Conference*, Seattle, 1986.

[78] B. Wahlberg. *On the identification and approximation of linear systems.* PhD thesis, Linköping University, Sweden, 1987.

[79] J.H. Wilkinson. *The Algebraic Eigenvalue Problem.* Clarendon Press, Oxford, 1965.

[80] J.C. Willems. Least squares stationary optimal control and the algebraic Riccati equation. *IEEE Transactions on Automatic Control*, 16:621–634, 1971.

[81] N. Young. *An introduction to Hilbert space.* Cambridge University Press, Cambridge, 1988.

[82] H.P. Zeiger and A.J. McEwen. Approximate linear realizations of given dimension via Ho's algorithm. *IEEE Transactions on Automatic Control*, 19:153, 1974.