Nonlinear System Identification: A State-Space Approach

Vincent Verdult

Nonlinear System Identification: A State-Space Approach

Thesis Committee:

Prof. dr. ir. J. H. A de Smit (chairman)	University of Twente
Prof. dr. ir. M. H. G. Verhaegen (supervisor)	University of Twente
Prof. dr. ir. J. B. Jonker	University of Twente
Prof. dr. L. Ljung	Linköping University, Sweden
Prof. dr. A. A. Stoorvogel	Delft University of Technology
Prof. dr. ir. P. P. J. van den Bosch	Eindhoven University of Technology
Prof. dr. ir. P. M. J. van den Hof	Delft University of Technology
Prof. dr. A. J. van der Schaft	University of Twente

Twente University **Press**

NONLINEAR SYSTEM IDENTIFICATION:
A STATE-SPACE APPROACH

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F. A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 1 maart 2002 te 13.15 uur

door

Vincent Verdult

geboren op 25 mei 1974

te Bergen op Zoom

Dit proefschrift is goedgekeurd door de promotor prof. dr. ir. M. H. G. Verhaegen

The hidden law of a probable outcome,
The numbers lead a dance.
<div align="right">—Sting, Shape of My Heart (1993)</div>

Contents

Acknowledgments

Many people have contributed directly or indirectly to this thesis and the scientific research associated with it. First, I would like to thank my supervisor Michel Verhaegen for his excellent guidance over the years and for teaching me how to perform research. His continuous stream of new ideas keeps amazing me.

During the first two years of my time as a Ph. D. student I worked at the Control Laboratory of the Faculty of Information Technology and Systems, Delft University of Technology. The people of the Control Laboratory provided me with a pleasant working atmosphere. Special thanks go to Bart De Schutter, Bert Haverkamp, Govert Monsees, Marcel Oosterom, Hans Roubos, Stefano Stramigioli, and David Westwick. The final two years of my Ph. D. research took place at the Systems and Control Engineering group, Faculty of Applied Physics, University of Twente. In this small group, a lot of interaction took place. Bas Benschop, Niek Bergboer, Rufus Fraanje, Herman Hemmes, Ichero Jikuya, Stoyan Kanev, and Gerard Nijsse, it was a pleasure to work with you.

Part III of this thesis is the result of a three months' visit at the Automatic Control group of the Department of Electrical Engineering, Linköping University, Sweden. I would like to thank Professor Lennart Ljung for making this visit possi-

ble. Thanks to all the people at the Automatic Control group for making it a very productive and pleasant stay.

Professor Jan Maciejowski (Cambridge University) and Dr. Huixin Chen (University of Sunderland) were always eager to discuss bilinear subspace identification. Over the years we had interesting discussions at several conferences. Thanks for that.

I thank Dr. Marco Lovera (Polytechnico di Milano, Italy) for sharing his ideas on bilinear subspace identification and his cooperation on the LPV identification of helicopter rotor dynamics.

I would also like to thank the members of the thesis committee for showing their interest and the people at Twente University Press for their professional support in publishing the thesis.

I am pleased to have the support of two enthusiastic 'paranimfen': Evelyn Hagenaars–Willems and Alexandra Padt. Your assistance is highly appreciated.

My father deserves my deepest gratitude for his continuous support over the years; it helped me to shape my life as it is. To my friends and family I would like to say that you make life worthwhile.

Of course, Madelon I thank you for being there whenever necessary. Despite all the maths, you managed to read the entire manuscript. Very impressive. You helped improving it a lot. I guess it means that I have to read yours in the future as well. As we say in Bergen op Zoom: *Dagge bedankt zijt, da witte*.

Chapter 1

Introduction

Many engineering applications need a compact and accurate description of the dynamic behavior of the system under consideration. This is especially true of automatic control applications. Dynamic models describing the system of interest can be constructed using the first principles of physics, chemistry, biology, and so forth. However, models constructed in this way are difficult to derive, because they require detailed specialist knowledge, which may be lacking. The resulting models are often very complex. Developing such models can be very labor-intensive, and hence expensive. For poorly understood systems, the derivation of a model from the first principles is even impossible. Because the first-principle models are often complex, their simulation takes considerable time on a computer; therefore, they are not suitable for fast on-line applications. Moreover, these models are not always very accurate, because of two reasons. First, it is difficult to decide which effects are relevant and must be included in the model, and which effects can be neglected. Second, certain quantities needed to build the model are unknown, and have to be estimated by performing dedicated experiments. The resulting estimates often differ from the real quantities, and hence some model mismatch can occur.

An alternative way of building models is by system identification. In system identification, the aim is to estimate dynamic models directly from observed input and output data. First principles are not directly used to model the system, but expert knowledge about the system still plays an important role. Such knowledge is of great value for setting up identification experiments to generate the required measurements, for deciding upon the type of models to be used, and for determining the quality and validity of the estimated models. System identification often yields compact accurate models that are suitable for fast on-line applications and for model-based predictive control, which has found widespread use in several branches of the chemical process industry. System identification still requires considerable human intervention and expert knowledge to arrive at models that are satisfactory for the application in mind. Nevertheless, compared with the development of models from first principles, it is not so labor-intensive. At present, several steps of the identification procedure can be automated.

It is important to realize that building models from first principles only and system identification based only on measurements are two extreme cases. Quite often a combination of the two is encountered. In such a combination, system identification is used to estimate the unknown parts or parameters of a model based on first principles. Therefore, a whole range of approaches exists that shows a gradual transition from first-principles modeling to system identification.

System identification proceeds as follows: First, a certain type of model is selected that is considered to be suitable for the application at hand. Second, a special input signal is designed such that the model captures the behavior of the system to be modeled. Then an identification experiment is carried out in which input and output signals are measured. An identification method is selected to estimate the parameters that describe the model from the collected input and output measurements. Finally, the validity of the obtained model is evaluated.

An important step in system identification is the determination of the type of model to be used. This decision is based on knowledge of the system under consideration, and on the properties of the model. Certain types of models can be used to approximate the input-output behavior of a smooth nonlinear dynamical system up to arbitrary accuracy. Such models have the so-called universal approximation capability. An example of a universal approximator is the neural network (Haykin 1999). The drawback of these models is that they are complex, and difficult to estimate and analyze. Therefore, other model structures have received considerable attention over the years. The linear time-invariant model is the most popular one. It has been used successfully in many engineering applications, and a considerable body of theory exists for system identification and automatic control of linear systems. The authoritative guide for linear system identification is the book by Ljung (1999). Attractive methods for linear system identification are the subspace methods developed by Larimore (1983), Verhaegen and Dewilde (1992), and Van Overschee and De Moor (1996). These are numerically robust methods that can easily deal with systems having multiple inputs and outputs and are noniterative, unlike many other identification methods.

Although linear models are attractive for several reasons, they also have their limitations. Most real-life systems show nonlinear dynamic behavior; a linear model can only describe such a system for a small range of input and output values. In the ever demanding world of today, where systems are pushed to the limits of their performance and accurate descriptions over large input and output ranges are needed, linear models are often not satisfactory anymore. Therefore, a considerable interest in identification methods for nonlinear systems has risen.

This thesis deals with the identification of certain types of discrete-time nonlinear systems. Nonlinear discrete-time systems can be represented by a state-space description, or alternatively by an input-output description. Most work in the area of nonlinear system identification has concentrated on input-output models. An overview of the field was given by Sjöberg et al. (1995). Their paper provides a nice introduction to nonlinear system identification, and is recommended to the uninitiated reader.

A state-space description of a discrete-time nonlinear system has the form:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k), \\ y_k &= g(x_k, u_k), \end{aligned}$$

where $x_k \in \mathbb{R}^n$ is the state of the system, $u_k \in \mathbb{R}^m$ the input, $y_k \in \mathbb{R}^\ell$ the output, and f and g are nonlinear vector functions. The dimension n of the state vector is called the *order* of the system. The order indicates the memory that is present in the system, and has an influence on the time behavior of the system.

An input-output description of a nonlinear system has the following form:

$$y_k = h(y_{k-1}, y_{k-2}, \ldots, y_{k-M}, u_k, u_{k-1}, \ldots, u_{k-M}),$$

where $u_k \in \mathbb{R}^m$ is the input, and $y_k \in \mathbb{R}^\ell$ the output. The integer M is an indication of the amount of memory present in the system, and is therefore called the memory lag of the system. It is closely related to the order of the system.

The two mathematical model representations, state-space and input-output, can be related to each other (Leontaritis and Billings 1985a,b; Rivals and Personnaz 1996). If the functions f, g, and h are continuous and differentiable, then every state-space system can be converted into an input-output system, but the reverse is not necessarily true (Stark et al. 1997; Stark 1999). Similar relations exist for continuous-time state-space and input-output systems (Nijmeijer and van der Schaft 1996).

State-space systems are more attractive for dealing with multivariable inputs and outputs. As argued by Rivals and Personnaz (1996), state-space systems are likely to require fewer parameters, especially for multivariable systems. One of the reasons is that the number of time-lagged inputs and outputs M used in the input-output description satisfies $n \leq M \leq 2n + 1$ with n the order of the state-space realization of this system, provided it exists (Stark et al. 1997; Stark 1999). In addition, many nonlinear control methods require a state-space model of the system. Because of these reasons, state-space models are often preferred to input-output models.

The particular choice of the functions f and g for the state-space model, or of the function h for the input-output model, determines the structure of the nonlinear model. Of course many possibilities exist. In general, two approaches can be distinguished. The first approach is to choose a simple structure for the model, such that it has certain nice properties, is easy to analyze, and leads to computationally attractive identification methods. The main drawback is that simple structures often represent a limited class of nonlinear systems. Therefore, the second approach aims at choosing the model structure such that the model can approximate a large class of nonlinear systems. The disadvantage is that this often leads to complicated identification algorithms. Examples of the first approach are the linear model, bilinear model (Mohler 1973; Bruni, Dipillo, and Koch 1974), Hammerstein model (Leontaritis and Billings 1985b), and Wiener model (Leontaritis and Billings 1985b). Examples of the second approach are sigmoidal neural networks (Narendra and Parthasarathy 1990; Billings, Jamaluddin, and Chen 1992; Haykin 1999), radial basis function networks, local linear model structures (Murray-Smith and Johansen 1997b), Takagi-Sugeno fuzzy models (Takagi and Sugeno 1985), and hinging hyperplanes models (Breiman 1993; Pucar and Sjöberg 1998).

The choice of model structure depends to a large extend on the application at hand. Each model structure has its own advantages and disadvantages, and no universal recommendation can be given. Several methods have been developed to aid in the determination of the right model structure for the problem at hand;

an overview has been given by Haber and Unbehauen (1990). To decide whether the right model structure has been chosen, the use of model validation methods is of paramount importance. Validation methods for nonlinear models have been described by Billings and Voon (1986) and Leontaritis and Billings (1987).

Because of the diversity of model structures, the literature on nonlinear system identification is vast; for the reader who wants to delve into it, some pointers to start from are: Billings (1980), Chen and Billings (1989), Narendra and Parthasarathy (1990), Haber and Unbehauen (1990), Billings, Jamaluddin, and Chen (1992), Sjöberg et al. (1995), Juditsky et al. (1995), Giannakis and Serpedin (2001).

In this thesis the identification of three different state-space model structures is discussed. The first model structure is the linear parameter-varying system. This system can be viewed as a linear system in which the system matrices are functions of a known time-varying parameter vector. This description also includes some particular nonlinear systems. If, for example, the system matrices vary as a function of the input signal, it is clear that the system is nonlinear. Only linear parameter-varying systems in which the system matrices are an affine function of the time-varying parameter vector are considered. The second model structure is the bilinear system. The bilinear system is in fact a special case of the linear parameter-varying system with affine parameter dependence, where the time-varying parameter equals the input signal. The third model structure under consideration is a weighted combination of local linear state-space models. Such a system can be used to approximate a very large class of nonlinear systems. If the weights are viewed as time-varying parameters, a linear parameter-varying system is obtained. The main difference is that for the linear parameter-varying system, the time-varying parameters are assumed to be known a priori, while for local linear state-space models the weights have to be determined from input and output data.

The thesis is divided into three parts. Each part addresses the identification of one of the above-mentioned model structures. Every part contains an introductory chapter that provides a motivation for the choice of the particular model structure. In each part the discussion focuses on the estimation of the model parameters. Identification methods are described that determine a state-space model from input and output data. The structure and the order of the system are assumed to be known a priori. Order detection, structure determination, input design and model validation are not addressed. Two different types of identification methods are discussed: nonlinear optimization-based methods and subspace methods. Subspace methods are only discussed for the linear parameter-varying system and the bilinear system. Subspace methods were originally developed for the identification of linear systems, and are based on numerical methods from linear algebra. For that reason subspace methods can easily deal with systems having multiple inputs and outputs, and are noniterative and numerically robust. Because of these properties, it is worthwhile to investigate whether subspace identification methods can be derived for the identification of certain nonlinear model structures. Several authors have worked on such extensions (an overview is given in Chapters 3 and 9) and this thesis provides a further step in that direction.

An outline of the thesis is provided in Section 1.1, along with the dependencies

among the several chapters. Section 1.2 contains a brief summary of the main contributions of the thesis. Section 1.3 describes which parts of the thesis have been published elsewhere in a different form.

1.1   Outline of the Thesis

This thesis consists of three parts:

Part I (Chapters 2–7) deals with linear parameter-varying (LPV) state-space systems. It provides a general framework for subspace identification of LPV systems (Chapter 3), describes a practical dimension reduction method for subspace identification (Chapter 4) and discusses nonlinear optimization-based identification (Chapter 5). The methods that are presented are illustrated with several simulation examples and applied to identify helicopter rotor dynamics (Chapter 6). A detailed outline of this part can be found in Section 2.5.

Part II (Chapters 8–13) discusses the identification of bilinear state-space systems. Both subspace identification methods (Chapters 9 and 10) and nonlinear optimization-based identification methods (Chapter 11) are described. Several examples are used to illustrate the methods; among them is the identification of a high-purity distillation column (Chapter 12). A detailed outline of this part can be found in Section 8.6.

Part III (Chapters 14–17) describes a new method for the identification of local linear state-space systems from input and output data. The method is tested on several examples including a bioreactor (Chapter 16). A detailed outline of this part can be found in Section 14.5.

Some chapters in Part II depend on material presented in Part I. Figure 1.1 summarizes the dependencies among the chapters in these two parts. It can be used as a guide for reading the thesis. Part III can be read independently from Parts I and II. The dependencies among the chapters in Part III are given in Figure 1.2.

1.2   Contributions

The main contributions of this thesis are:

- A general framework for subspace identification of LPV and bilinear systems is developed (Chapters 3 and 9). This framework allows a unified treatment of bilinear subspace identification methods that appear in the literature, and it leads to new methods. It also establishes a link between subspace identification of bilinear and LPV systems. Subspace identification methods for LPV systems have not been studied before.

- A dimension reduction procedure is proposed for the bilinear and LPV subspace methods (Chapter 4). The dimensions of the data matrices used in these subspace methods grow exponentially with the order of the system; this causes a major problem in the implementation of the methods. Although this problem has been recognized in the literature before, no practical solutions have been derived. The dimension reduction method proposed
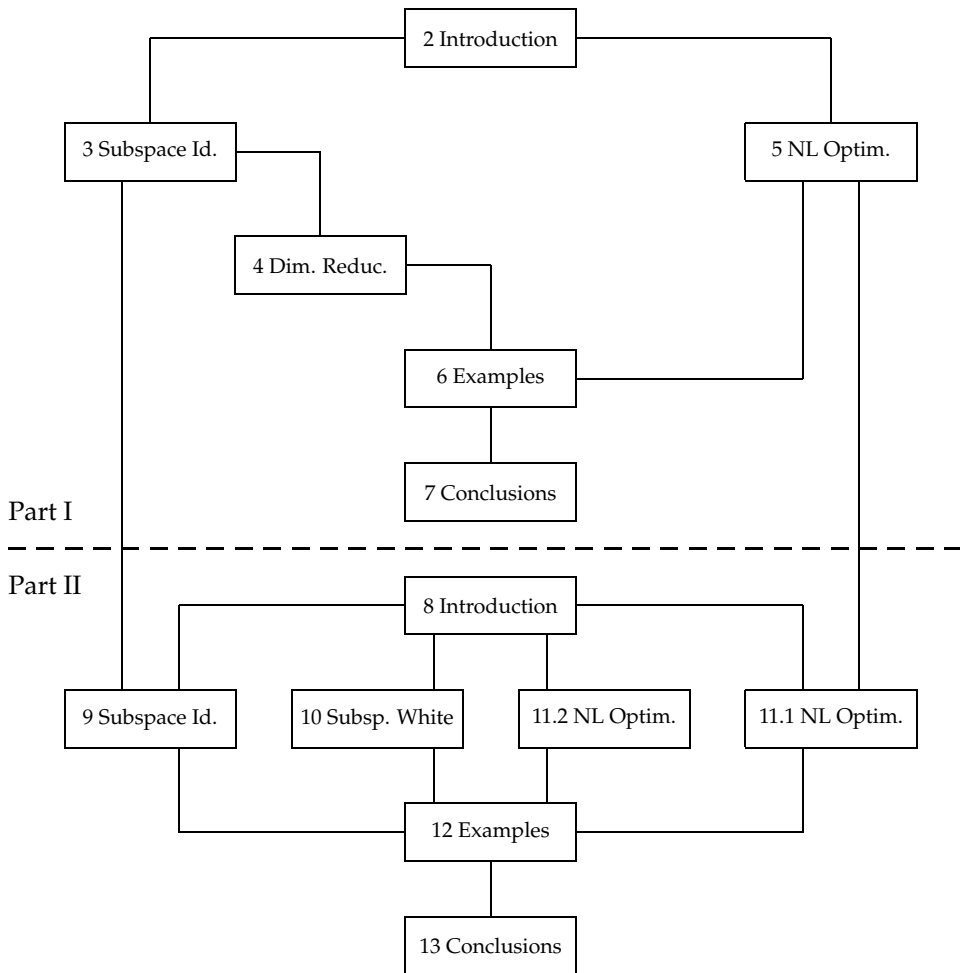
Figure 1.1: Dependencies among the chapters of Parts I and II. Each chapter depends on the chapters drawn above it and directly connected to it. Figure 1.2 describes Part III.
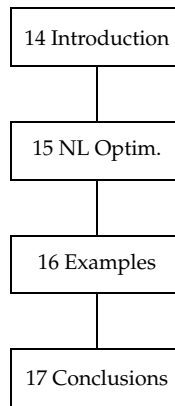
Figure 1.2: Dependencies among the chapters of Part III. Each chapter depends on the chapters drawn above it and directly connected to it.

in this thesis drastically reduces the computational complexity of the bilinear and LPV subspace identification methods, and makes them practically feasible for multivariable industrial systems.

- A new subspace identification method for bilinear systems is presented that avoids the use of large dimensional data matrices (Chapter 10). It requires that the bilinear system is driven by a white-noise input. Although this somewhat restricts the applicability of the method, it is shown that the white-noise properties can be exploited to arrive at a subspace scheme in which the dimensions of the data matrices grow linearly with the order of the system, instead of exponentially as in the conventional methods.

- Two methods are proposed for prediction-error identification of bilinear systems based on nonlinear optimization (Chapter 11). One method is derived from an LPV identification method and circumvents the unique parameterization of the bilinear system, the other method exploits the principle of separable least-squares to reduce the dimension of the parameter space. Both methods deal with general multivariable bilinear state-space systems. By contrast, several other identification methods in the literature can only deal with state-space systems having full state measurements, or with the special phase-variable state-space representation.

- A novel method is proposed that allows the determination of local linear state-space models from input and output data to approximate a nonlinear system (Chapter 15). The method does not require full state measurements like most previously suggested methods. Another advantage of the method is that it does not require a unique parameterization of the local state-space systems. It uses a full parameterization, and deals with the nonuniqueness by only updating at each iteration the parameters that actually change the input-output behavior. This approach has numerical advantages over the use of canonical parameterizations, which are known to lead to ill-conditioned problems.

1.3 Publications

Some results presented in this thesis have been published elsewhere in a different form.

The LPV subspace identification method in Section 3.5 and the dimension reduction method in Chapter 4 were also described in the papers:

Verdult, Vincent and Michel Verhaegen (2000). Identification of multivariable linear parameter-varying systems based on subspace techniques. In *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia (December).

Verdult, Vincent and Michel Verhaegen (2000). Subspace identification of multivariable linear parameter-varying systems. Accepted for publication in *Automatica*.

The multivariable LPV identification example presented in Subsection 6.1.4 appeared in the last paper. The identification of helicopter rotor dynamics using LPV methods presented in Section 6.2 was discussed in:

Verdult, Vincent, Marco Lovera and Michel Verhaegen (2001). Identification of linear parameter-varying state space models for helicopter rotor dynamics. Technical report, Faculty of Applied Physics, University of Twente, Enschede, The Netherlands.

Chapter 5 on optimization-based identification methods for LPV systems is based on:

Verdult, Vincent and Michel Verhaegen (2001). Identification of multivariable LPV state space systems by local gradient search. In *Proceedings of the European Control Conference 2001*, Porto, Portugal (September).

The subspace algorithm for bilinear systems subject to white-noise inputs in Chapter 10 was presented in the papers:

Verdult, Vincent, Michel Verhaegen, C. T. Chou and Marco Lovera (1998). Efficient and systematic identification of MIMO bilinear state space models. In *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, Florida (December), pp. 1260–1265.

Verdult, Vincent, Michel Verhaegen, C. T. Chou and Marco Lovera (1998). Efficient subspace-based identification of MIMO bilinear state space models. In *Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling: Theory and Applications*, Leuven, Belgium (July), pp. 216–221.

The combination of the bilinear subspace identification method in Section 9.3 with the dimension reduction method in Chapter 4 was discussed in:

Verdult, Vincent and Michel Verhaegen (2001). Identification of multivariable bilinear state space systems based on subspace techniques and separable least squares optimization. *International Journal of Control* **74**(18), 1824–1836.

Section 11.2 on bilinear identification by separable least-squares optimization is also based on that paper. Other papers that deal with separable least-squares are:

Verdult, Vincent and Michel Verhaegen (1999). Subspace-based identification of MIMO bilinear systems. In *Proceedings of the European Control Conference 1999*, Karlsruhe, Germany (September).

Verdult, Vincent and Michel Verhaegen (2000). Bilinear state space systems for nonlinear dynamical modelling. *Theory in Biosciences* **119**(1), 1–9.

Verdult, Vincent, Michel Verhaegen and C. T. Chou (1999). Identification of MIMO bilinear state space models using separable least squares. In *Proceedings of the 1999 American Control Conference*, San Diego, California (June), pp. 838–842.

The bilinear subspace identification method in Section 9.3 combined with the dimension reduction method in Chapter 4 and the optimization-based method in Section 11.1 was used in the following paper to obtain bilinear models for model-based predictive control of a polymerization process:

Bloemen, Hayco, Vincent Verdult, Ton van den Boom and Michel Verhaegen (2001). Bilinear versus linear MPC: Application to a polymerization process. Accepted for presentation at *IFAC World Congress 2002*, Barcelona, Spain (July).

Part III on local linear models, in particular Chapters 15 and 16, is based on:

Verdult, Vincent, Lennart Ljung and Michel Verhaegen (2001). Identification of composite local linear state-space models using a projected gradient search. Technical Report LiTH-ISY-R-2359, Department of Electrical Engineering, Linköping University, Linköping, Sweden. Accepted for publication in *International Journal of Control*.

Besides the above-mentioned publications, the following publications were written during my time as a Ph. D. student:

Bloemen, Hayco, C. T. Chou, Ton van den Boom, Vincent Verdult, Michel Verhaegen and Ton Backx (2000). Wiener model identification and predictive control for dual composition control of a distillation column. *Journal of Process Control* **11**(6), 601–620.

Bloemen, Hayco, C. T. Chou, Ton van den Boom, Vincent Verdult, Michel Verhaegen and Ton Backx (2000). Wiener MPC for high purity dual composition control of a distillation column. In *Process Control and Instrumentation 2000*, Glasgow, Scotland (July), pp. 198–203.

Bloemen, Hayco, C. T. Chou, Vincent Verdult, Ton van den Boom, Ton Backx and Michel Verhaegen (1999). Economic benefits of using nonlinear models for high purity distillation column control. Technical Report UT/TN/SCE-1999.001, Faculty of Applied Physics, University of Twente, Enschede, The Netherlands.

Chou, C. T., Hayco Bloemen, Vincent Verdult, Ton van den Boom, Ton Backx and Michel Verhaegen (1998). Identification and model predictive control of high purity distillation columns using Wiener models: A preliminary report. Technical Report TUD/ITS/ET/SCE98.008, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands.

Chou, C. T., Hayco Bloemen, Vincent Verdult, Ton van den Boom, Ton Backx and Michel Verhaegen (2000). Nonlinear identification of high purity distillation columns. In *Preprints of the IFAC Symposium on System Identification*, Santa Barbara, California (June).

Oku, Hiroshi, Gerard Nijsse, Michel Verhaegen and Vincent Verdult (2001). Change detection in the dynamics with recursive subspace identification. In *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, Florida (December).

Verdult, Vincent, Michel Verhaegen and Jacquelien Scherpen (2000). Identification of nonlinear nonautonomous state space systems from input-output measurements. In *Proceedings of the 2000 IEEE International Conference on Industrial Technology*, Goa, India (January), pp. 410–414.

Verdult, Vincent and Michel Verhaegen (2001). Identification of a weighted combination of multivariable state space systems from input and output data. In *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, Florida (December).

Verhaegen, Michel and Vincent Verdult (2001). *Filtering and System Identification: An Introduction*. Book in preparation.

Part I

Linear Parameter-Varying State-Space Systems

Chapter 2

Introduction

O ver the last three decades, considerable interest has risen in methods for de-signing controllers for nonlinear systems. One popular approach is *gain-scheduling control*, in which the nonlinear system is approximated by a number of linear models obtained in different operating points and linear control techniques are used to design a controller for each operating point. The overall controller for the nonlinear system is then obtained by combining these controllers. Gain scheduling allows the reuse of many insights obtained from linear control system design. Unfortunately, gain scheduling has several pitfalls as discussed, for example, by Shamma and Athans (1992). Often a large number of linear models is needed to achieve certain performance specifications. Moreover, to guarantee performance it is necessary to slowly vary between operating points.

To overcome these limitations of classical gain scheduling, attention has shifted towards the design of a global controller for a linear parameter-varying system (see for example Shamma and Athans 1991; Packard 1994). A linear parameter-varying (LPV) system is a linear system in which the system matrices are fixed functions of a known time-varying parameter vector. If this parameter vector is denoted by $p_k \in \mathbb{R}^s$, the LPV system can be represented as

$$
\begin{aligned}
x_{k+1} &= \mathcal{A}(p_k)x_k + \mathcal{B}(p_k)u_k, \\
y_k &= \mathcal{C}(p_k)x_k + \mathcal{D}(p_k)u_k,
\end{aligned}
$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ is the input, $y_k \in \mathbb{R}^\ell$ is the output, and $\mathcal{A}(p_k)$, $\mathcal{B}(p_k)$, $\mathcal{C}(p_k)$, and $\mathcal{D}(p_k)$ are the parameter-varying system matrices. An LPV system can be viewed as a nonlinear system that is linearized along a time-varying trajectory determined by the time-varying parameter vector p_k. Hence, the time-varying parameter vector of an LPV system corresponds to the operating point of the nonlinear system. In the LPV framework, it is assumed that this parameter is measurable for control. In many industrial applications, like flight control and process control, the operating point can indeed be determined from measurements, making the LPV approach viable.

Control design for LPV systems is an active research area. Within the LPV framework, systematic techniques for designing gain-scheduled controllers can

be developed. Such techniques allow tighter performance bounds and can deal with fast variations of the operating point. Furthermore, control design for LPV systems has a close connection with modern robust control techniques based on linear fractional transformations (Zhou, Doyle, and Glover 1996).

The important role of LPV systems in control system design motivates the development of identification techniques that can determine such systems from measured data. The identification of LPV systems is the topic of this part of the thesis. In particular, LPV systems with *affine parameter dependence* are considered. For such systems, the system matrices depend on the parameter vector p_k in the following way:

$$\mathcal{A}(p_k) = A_0 + \sum_{i=1}^{s} [p_k]_i A_i,$$

$$\mathcal{B}(p_k) = B_0 + \sum_{i=1}^{s} [p_k]_i B_i,$$

$$\mathcal{C}(p_k) = C_0 + \sum_{i=1}^{s} [p_k]_i C_i,$$

$$\mathcal{D}(p_k) = D_0 + \sum_{i=1}^{s} [p_k]_i D_i,$$

where $[p_k]_i$ denotes the ith element of the vector p_k, and $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times m}$, $C_i \in \mathbb{R}^{\ell \times n}$, and $D_i \in \mathbb{R}^{\ell \times m}$ for $i = 0, 1, 2, \ldots, s$. By defining

$$
\begin{aligned}
A &:= [A_0 \quad A_1 \quad A_2 \quad \cdots \quad A_s], \\
B &:= [B_0 \quad B_1 \quad B_2 \quad \cdots \quad B_s], \\
C &:= [C_0 \quad C_1 \quad C_2 \quad \cdots \quad C_s], \\
D &:= [D_0 \quad D_1 \quad D_2 \quad \cdots \quad D_s],
\end{aligned}
$$

the LPV system with affine parameter dependence can also be written in the more compact form

$$x_{k+1} = A \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + B \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix}, \tag{2.1}$$

$$y_k = C \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + D \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix}, \tag{2.2}$$

where $\otimes$ denotes the *Kronecker product*, which is defined as follows: let $M \in \mathbb{R}^{p \times q}$ and $N \in \mathbb{R}^{r \times t}$ be two arbitrary matrices, then the Kronecker product of these matrices is a $pr \times qt$ matrix given by

$$M \otimes N = \begin{bmatrix} m_{1,1}N & m_{1,2}N & \cdots & m_{1,q}N \\ m_{2,1}N & m_{2,2}N & \cdots & m_{2,q}N \\ \vdots & \vdots & \ddots & \vdots \\ m_{p,1}N & m_{p,2}N & \cdots & m_{p,q}N \end{bmatrix},$$

where $m_{i,j}$ denotes the (i, j)th entry of the matrix M. In the sequel, the term LPV system is often used to denote an LPV system with affine parameter dependence.

The remainder of this chapter is organized as follows: First, in Section 2.1 it is pointed out that a close connection exists between the LPV system with affine parameter dependence and the linear fractional transformation used in robust control. Next, an overview of identification methods for LPV systems is given in Section 2.2. Some of the methods discussed in this overview can also be used for identification of the linear fractional transformation descriptions. The following two sections, 2.3 and 2.4, introduce some concepts and assumptions that are needed in the forthcoming chapters. Finally, Section 2.5 describes the organization of Part I of the thesis.

2.1 Linear Fractional Transformation Descriptions

The LPV system (2.1)–(2.2) is a special case of the linear fractional transformation (LFT) description that is often used in robust control theory (Zhou, Doyle, and Glover 1996).

LFTs come in two flavors: upper and lower transformations. Only the lower LFT is needed in the sequel; it is defined as follows:

Definition 2.1 *(Zhou et al. 1996, p. 247) Let the matrix $M \in \mathbb{R}^{(p_1+p_2)\times(q_1+q_2)}$ be partitioned as*

$$M = \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array} \right].$$

The lower LFT of M with respect to the matrix $\Delta \in \mathbb{R}^{q_2 \times p_2}$ is given by

$$\mathcal{F}(M, \Delta) := M_{11} + M_{12}\Delta(I_{p_2} - M_{22}\Delta)^{-1}M_{21},$$

provided that the inverse of $(I_{p_2} - M_{22}\Delta)$ exists.

An LPV system with fractional parameter dependence can be represented by the lower LFT

$$\mathcal{F}\left(\left[\begin{array}{cc|c} A_0 & B_0 & B_f \\ C_0 & D_0 & D_f \\ \hline C_z & D_z & D_{zf} \end{array} \right], \Delta(p_k) \right), \tag{2.3}$$

with the time-varying block

$$\Delta(p_k) := \left[\begin{array}{cccc} [p_k]_1 I_{r_1} & & & 0 \\ & [p_k]_2 I_{r_2} & & \\ & & \ddots & \\ 0 & & & [p_k]_s I_{r_s} \end{array} \right],$$

and the time-invariant system

$$\begin{array}{rclclcl} x_{k+1} & = & A_0 x_k & + & B_0 u_k & + & B_f f_k, \\ y_k & = & C_0 x_k & + & D_0 u_k & + & D_f f_k, \\ z_k & = & C_z x_k & + & D_z u_k & + & D_{zf} f_k, \end{array}$$

where $z_k \in \mathbb{R}^r$, $f_k \in \mathbb{R}^r$, and $r = \sum_{i=1}^{s} r_i$.
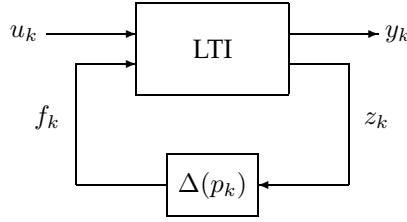
Figure 2.1: Linear fractional transformation description as a linear time-invariant (LTI) dynamic system that has a partial feedback connection with a time-varying static block ($\Delta(p_k)$).

The LFT (2.3) is a partial feedback connection of the time-invariant system with the time-varying block; the feedback law is:

$$f_k = \Delta(p_k)z_k.$$

This feedback system is illustrated in Figure 2.1.

If $D_{zf} = 0$, then there is no fractional parameter dependence in the LFT description; the parameter dependence becomes affine. Such an LFT description corresponds to an LPV system with affine parameter dependence (2.1)–(2.2). To establish this relation, the following partitions are introduced:

$$
\begin{aligned}
B_f &= \begin{bmatrix} B_{f,1} & B_{f,2} & \cdots & B_{f,s} \end{bmatrix}, \\
D_f &= \begin{bmatrix} D_{f,1} & D_{f,2} & \cdots & D_{f,s} \end{bmatrix}, \\
C_z &= \begin{bmatrix} C_{z,1} \\ C_{z,2} \\ \vdots \\ C_{z,s} \end{bmatrix}, \qquad
D_z = \begin{bmatrix} D_{z,1} \\ D_{z,2} \\ \vdots \\ D_{z,s} \end{bmatrix},
\end{aligned}
$$

with $B_{f,i} \in \mathbb{R}^{n \times r_i}$, $D_{f,i} \in \mathbb{R}^{\ell \times r_i}$, $C_{z,i} \in \mathbb{R}^{r_i \times n}$, and $D_{z,i} \in \mathbb{R}^{r_i \times m}$. Because $D_{zf} = 0$, the state equation of the LFT description can be written as

$$
\begin{aligned}
x_{k+1} &= A_0 x_k + B_0 u_k + B_f \Delta(p_k)(C_z x_k + D_z u_k) \\
&= A_0 x_k + B_0 u_k + \sum_{i=1}^{s} B_{f,i}[p_k]_i C_{z,i} x_k + \sum_{i=1}^{s} B_{f,i}[p_k]_i D_{z,i} u_k.
\end{aligned}
$$

A similar expression can be derived for the output equation:

$$
y_k = C_0 x_k + D_0 u_k + \sum_{i=1}^{s} D_{f,i}[p_k]_i C_{z,i} x_k + \sum_{i=1}^{s} D_{f,i}[p_k]_i D_{z,i} u_k.
$$

Therefore, by taking

$$
\begin{aligned}
A_i &= B_{f,i} C_{z,i}, & B_i &= B_{f,i} D_{z,i}, \\
C_i &= D_{f,i} C_{z,i}, & D_i &= D_{f,i} D_{z,i},
\end{aligned}
$$

for $i = 1, 2, \ldots, s$, the LFT description is equivalent to the system (2.1)–(2.2).

We see that converting an LFT description into the LPV system (2.1)–(2.2) is just a matter of partitioning the system matrices. The other way around, converting the LPV system into the LFT description, involves somewhat more work, because the block sizes r_i ($i = 1, 2, \ldots, s$) have to be determined. As discussed by Zhou, Doyle, and Glover (1996, Chapter 10), the block size r_i equals the the rank of the matrix

$$\begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix},$$

and can be determined by performing a singular value decomposition.

From the relation between the LFT description and the LPV system with affine parameter dependence, it follows that the LFT description is more general. The LFT description can deal with fractional parameter dependence, while the LPV system with affine parameter dependence cannot. However, by redefining and adding parameters to the vector p_k used in the LPV system, the LPV system can be used to approximate an LFT description with $D_{zf} \neq 0$. The state equation for $D_{zf} \neq 0$ becomes

$$x_{k+1} = A_0 x_k + B_0 u_k + B_f \Delta(p_k) \Big(I_r - D_{zf} \Delta(p_k) \Big)^{-1} (C_z x_k + D_z u_k).$$

To establish an approximation by an LPV system, the matrix

$$\Delta(p_k) \Big(I_r - D_{zf} \Delta(p_k) \Big)^{-1}$$

needs to be approximated by a matrix that depends linearly on the time-varying parameter used in the LPV system. Let this parameter be denoted by $\bar{p}_k$. Provided that for all p_k the eigenvalues of $D_{zf} \Delta(p_k)$ have magnitudes less than one,

$$\Big(I_r - D_{zf} \Delta(p_k) \Big)^{-1} = I_r + D_{zf} \Delta(p_k) + \Big(D_{zf} \Delta(p_k) \Big)^2 + \Big(D_{zf} \Delta(p_k) \Big)^3 + \cdots$$

Then taking

$$\bar{p}_k = \big[(p_k)^T \quad (p_k^2)^T \quad (p_k^3)^T \quad \ldots \quad (p_k^a)^T \big]^T$$

yields such an approximation. The number of polynomial terms a included in $\bar{p}_k$ determines the degree of approximation. Note that instead of using this polynomial expansion, more sophisticated approximations can also be used.

2.2   Overview of Identification Methods

The literature on identification of linear time-varying systems is vast. Most of the work is devoted to recursive identification methods (Ljung and Söderström 1983). An overview of some recent work in this area was given by Kleiman (2000). However, the literature on identification of LPV systems —systems where the time variation is explicitly known through some measurable parameter— is still rather limited.

Nemani, Ravikanth, and Bamieh (1995) were the first to address the identification of LPV systems with linear fractional parameter dependence. They considered an LFT description (2.3) with one time-varying parameter, one input and full state measurement and showed that in the noise-free case the identification problem can be reduced to a recursive linear least-squares problem. They showed that when the state measurements are corrupted by additive white noise, an instrumental variable approach can be used such that under some mild conditions the parameter estimates are consistent.

An identification method that deals with multiple time-varying parameters was described by Lovera, Verhaegen, and Chou (1998). Their method is not applicable to the general LFT description, but deals with LPV systems with affine parameter dependence in the state equation, as in (2.1), and full state measurements, that is, $y_k = x_k + v_k$. They showed that in the noise-free case the system matrices can be estimated from a linear least-squares problem. When the state measurements are noisy, an instrumental variable approach can be used. Their method is described in more detail in Subsection 3.6.1.

To overcome the restriction of full state measurements, Lovera (1997, Chapter 6) described an identification method that exploits the fact that the LFT description is an LTI system with a time-varying feedback. If the signals in the closed loop can be measured, then a subspace identification method for LTI systems in closed loop can be used to estimate the system matrices. However, these signals are not always available for measurement, which limits the application of the method.

Lee and Poolla (1996, 1999) proposed an identification method to identify LFT descriptions with multiple inputs, outputs, and time-varying parameters without the need for full state measurement. Their method is based on minimizing the output error or prediction error with a Gauss-Newton type of gradient search method. They showed that gradient calculations can be performed efficiently by simulating a set of LPV systems. Their main contribution is the use of a local parameterization of the LFT description that is determined numerically by the algorithm. A detailed description of this method was given by Lee (1997) in his Ph. D. thesis. In Chapter 5 of this thesis, the method is summarized for the special case of LPV systems with affine parameter dependence and it is explicitly shown how to compute the prediction error in the case of process noise. Because of the nonconvex nature of this nonlinear optimization-based procedure, a good initial estimate of the model is of paramount importance. The LPV subspace identification methods that are discussed in Chapter 3 are especially suitable for providing such an initial estimate.

Other approaches to LPV system identification are the so-called robust identification methods related to set-membership identification (Milanese and Vicino 1991) where not one model is identified, but rather a whole set of models that is consistent with the measured data and a priori made assumptions. Mazzaro, Movsichoff, and Sánchez Peña (1999) considered set-membership identification for the same LFT descriptions as Nemani, Ravikanth, and Bamieh (1995), with one time-varying parameter and with bounded noise on the states. Robust identification for LFT descriptions with multiple time-varying parameters was discussed by Sznaier, Mazzaro, and Inanc (1999). For impulse input signals, they determined a set of input-output models that is consistent with the LFT description,

and consistent with bounds on the noise, on the time-varying parameters and on a nonparametric uncertain part of the system. In addition, they determined a nominal model in this set. They showed that the determination of the consistency set and the nominal model can be formulated as a feasibility problem in terms of linear matrix inequalities.

In addition to these state-space-based methods, some authors have considered LPV systems in an input-output form:

$$A(q, p_k)y_k = B(q, p_k)u_k,$$

with

$$
\begin{aligned}
A(q, p_k) &:= 1 + a_1(p_k)q^{-1} + a_2(p_k)q^{-2} + \cdots + a_{n_a}(p_k)q^{-n_a}, & (2.4) \\
B(q, p_k) &:= b_0(p_k) + b_1(p_k)q^{-1} + b_2(p_k)q^{-2} + \cdots + b_{n_b}(p_k)q^{-n_b}, & (2.5)
\end{aligned}
$$

where q is the shift operator: $q^{-n}u_k = u_{k-n}$. Bamieh and Giarré (1999) considered the special case where the parameters $a_i(p_k)$, $b_i(p_k)$ are polynomials in the time-varying parameter p_k. They showed that the coefficients of these polynomials can be determined by solving a linear least-squares problem and they derived least-mean-square (LMS) and recursive least-squares (RLS) identification algorithms. In addition, they discussed a persistency of excitation condition for the input u_k and time-varying parameter p_k that guarantees that the recursive identification algorithms converge. In following papers (Bamieh and Giarré 2000, 2001) these results were extended by replacing the polynomial dependence by a linear combination of known nonlinear functions of p_k.

Previdi and Lovera (1999, 2001) also considered LPV systems in input-output from, but they assumed that the coefficients of A and B in (2.4)–(2.5) are the outputs of a single hidden layer neural network that depends on a vector constructed from time-delayed versions of the input u_k, the output y_k, and the time-varying parameter p_k. These models are closely related to the local linear modeling approach of Murray-Smith and Johansen (1997b); see also Part III of this thesis.

2.3   System Analysis

The goal of this section is to discuss some properties of the LPV system (2.1)–(2.2) that are needed in the forthcoming chapters. Throughout this part of the thesis it is assumed that the parameter vector p_k is fixed and does not depend on the input or the state of the system.

2.3.1   Similarity Transformations

Given a certain input signal u_k, there are many LPV systems of the form (2.1)–(2.2) that yield exactly the same output signal. To put it differently: from an input-output point of view, the LPV system representation is not unique. Reason being that the state x_k can be chosen in different ways without changing the input-output behavior of the system. Given a certain state x_k for the system (2.1)–(2.2), another state that yields the same input-output behavior is $\overline{x}_k = T^{-1}x_k$,

with $T \in \mathbb{R}^{n \times n}$ a square nonsingular matrix. The matrix T is called a *similarity transformation*. It is easy to see that the system

$$
\overline{x}_{k+1} = A_T \left[\begin{array}{c} \overline{x}_k \\ p_k \otimes \overline{x}_k \end{array} \right] + B_T \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right],
$$

$$
y_k = C_T \left[\begin{array}{c} \overline{x}_k \\ p_k \otimes \overline{x}_k \end{array} \right] + D_T \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right],
$$

with

$$
\begin{aligned}
A_T &:= \begin{bmatrix} T^{-1}A_0 T & T^{-1}A_1 T & \cdots & T^{-1}A_s T \end{bmatrix}, \\
B_T &:= T^{-1}B, \\
C_T &:= \begin{bmatrix} C_0 T & C_1 T & \cdots & C_s T \end{bmatrix}, \\
D_T &:= D,
\end{aligned}
$$

has the same input-output behavior as the system (2.1)–(2.2). By varying the similarity transformation T over the set of all nonsingular n-by-n matrices, a whole family of LPV systems is created that has the same input-output behavior.

Without any additional assumptions on the state or the system matrices A, B, C, and D, the system (2.1)–(2.2) cannot be uniquely identified from a set of input and output measurements. Therefore, in state-space identification, the aim is often to estimate the system matrices A, B, C and D up to a similarity transformation. In other words, one does not care about the similarity transform; any state-space realization of the system is acceptable. This is especially true for subspace identification methods (see Chapter 3).

2.3.2 Stability

Roughly speaking, the state and the output of a *stable* system remain bounded as time proceeds. Stability results for general time-varying linear systems are well-known and are described, for example, in the book by Rugh (1996, Chapters 22, 23, 24 and 27). The results presented below are borrowed from this book, and adapted to suit the special structure of the LPV system (2.1)–(2.2).

First, the concept of *internal stability* is introduced. Internal stability deals with the boundedness and asymptotic behavior of the state equation

$$
x_{k+1} = A \left[\begin{array}{c} x_k \\ p_k \otimes x_k \end{array} \right]. \tag{2.6}
$$

Definition 2.2 *(Rugh 1996, p. 425) The LPV state equation (2.6) is called uniformly exponentially stable if for every finite constant $\beta > 0$ there exist a finite constant $\gamma > 0$ and a constant λ, $0 \le \lambda < 1$ such that for every $j \le k$, every x_j, and every p_k satisfying*

$$
\sup_k \|p_k\|_2 \le \beta,
$$

the corresponding solution satisfies

$$
\|x_k\|_2 \le \gamma \lambda^{k-j} \|x_j\|_2.
$$

Theorem 22.7 from the book by Rugh (1996, p. 426) can be used to derive the following corollary:

Corollary 2.1 *The linear state equation (2.6) is uniformly exponentially stable if given a finite constant $\beta > 0$, there exist a finite constant $\gamma > 0$ and a constant λ, $0 \leq \lambda < 1$ such that*

$$\left\| \prod_{t=j}^{k-1} \left(A_0 + \sum_{i=1}^{s} [p_t]_i A_i \right) \right\|_2 \leq \gamma \lambda^{k-j},$$

for all k, j such that $k > j$, and for all p_k such that

$$\sup_k \|p_k\|_2 \leq \beta.$$

In this corollary, the product of square matrices M_t, for $t = 0, 1, \ldots, k - 1$, is defined as

$$\prod_{t=0}^{k-1} M_t = M_{k-1} M_{k-2} \cdots M_1 M_0.$$

Besides exponential stability, another important stability concept is *bounded-input, bounded-output stability*.

Definition 2.3 *(Rugh 1996, p. 509) The LPV system (2.1)–(2.2) is called uniformly bounded-input, bounded-output stable if for any bounded p_k there exists a finite constant η such that for any j and any input signal u_k the corresponding response y_k with $x_j = 0$ satisfies*

$$\sup_{k \geq j} \|y_k\|_2 \leq \eta \sup_{k \geq j} \|u_k\|_2.$$

From Lemma 27.2 in the book by Rugh (1996, p. 511), it is easy to derive the following corollary:

Corollary 2.2 *If the LPV system (2.1)–(2.2) with $u_k = 0$ is uniformly exponentially stable, then the LPV system is uniformly bounded-input, bounded-output stable.*

2.3.3   Observability

Observability deals with the ability to uniquely determine the state sequence from output measurements of the system.

Definition 2.4 *(Rugh 1996, p. 467) The LPV system (2.1)–(2.2) is called observable on $[j, k]$ if every initial state x_j is uniquely determined by the corresponding zero-input response y_i for $i = j, j + 1, \ldots, k - 1$.*

The *observability matrix* of the system (2.1)–(2.2) is defined as

$$
\mathcal{O}(j,k) := \begin{bmatrix} C_0 + \sum_{i=1}^{s} [p_j]_i C_i \\ \left(C_0 + \sum_{i=1}^{s} [p_{j+1}]_i C_i \right) \left(A_0 + \sum_{i=1}^{s} [p_j]_i A_i \right) \\ \vdots \\ \left(C_0 + \sum_{i=1}^{s} [p_{k-1}]_i C_i \right) \prod_{t=j}^{k-2} \left(A_0 + \sum_{i=1}^{s} [p_t]_i A_i \right) \end{bmatrix}.
\tag{2.7}
$$

A slight modification of Theorem 25.9 from the book by Rugh (1996) can be used to determine the observability of the system from the observability matrix. This theorem reads as follows:

Theorem 2.1 *(Rugh 1996, p. 467) The LPV system (2.1)–(2.2) is observable on $[j,k]$ for a fixed sequence p_i, $i \in [j,k]$ if and only if*

$$
\text{rank}\Big(\mathcal{O}(j,k) \Big) = n,
$$

where $\mathcal{O}(j,k)$ is the observability matrix given by equation (2.7)

2.4 Assumptions

To develop identification procedures for LPV systems, several assumptions on the LPV system and the corresponding signals are needed. These assumptions are summarized below.

Throughout this part of the thesis, it is assumed that the parameter p_k of the LPV system is fixed and that the system is bounded-input, bounded-output stable in the sense of Definition 2.3, and observable in the sense of Definition 2.4 on the entire interval $[1, N]$ that corresponds to the measurements that are available. In the part dealing with subspace identification (Chapter 3) a somewhat stronger observability assumption is made: the linear system constructed from A_0 and C_0 must be observable. In other words, the matrix

$$
\begin{bmatrix} C_0 \\ C_0 A_0 \\ \vdots \\ C_0 A_0^{n-1} \end{bmatrix}
$$

must have full column rank. This assumption is not needed in the optimization-based identification methods in Chapter 5.

The input u_k to the LPV system is assumed to be a *deterministic sequence*, such that for all $j \in \mathbb{Z}$

$$
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} u_k u_{k+j}^T = R_u(j),
$$

with $R_u(j)$ a bounded matrix that is positive definite for $j = 0$. A similar assumption is made for the deterministic sequence p_k.

The LPV system is assumed to be subject to process noise w_k and measurement noise v_k in the following way:

$$x_{k+1} = A \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + B \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix} + w_k,$$

$$y_k = C \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + D \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix} + v_k.$$

The disturbances w_k and v_k are assumed to be sequences of independent random variables with zero-mean and bounded second- and fourth-order moments.

It is also assumed that p_k is not a linear combination of u_k. The case where $p_k = u_k$ is similar to the bilinear system, which is treated in Part II of this thesis. The main difference between the LPV system with $p_k = u_k$ and the bilinear system is that the state equation of the LPV system depends on $u_k \otimes u_k$, while the state equation of the bilinear system is not influenced by $u_k \otimes u_k$. Since p_k is assumed to be deterministic, it cannot be equal to the output y_k, which contains a stochastic component due to w_k and v_k. The case where p_k depends on the output y_k is treated in Part III of this thesis.

The assumptions stated above result in some useful asymptotic properties. The strong law of large numbers (Grimmett and Stirzaker 1983, p. 193) implies that for all $j \in \mathbb{Z}$

$$\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} u_k v_{k+j}^T = 0, \quad \text{w. p. 1,} \tag{2.8}$$

where 'w. p. 1' is used to mean 'with probability 1'. A similar expression holds for p_k: for all $j \in \mathbb{Z}$ it holds that

$$\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} p_k v_{k+j}^T = 0, \quad \text{w. p. 1.}$$

Of course the relations stated above also hold if v_k is replaced by w_k. Because of the assumption of uniformly bounded-input, bounded-output stability of the LPV system, and the assumptions on the input and noise sequences stated above, Theorem 2.3 from the book by Ljung (1999, p. 43) implies that for all $j \in \mathbb{Z}$

$$\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} y_k y_{k+j}^T = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \mathbf{E}[y_k y_{k+j}^T], \quad \text{w. p. 1,}$$

where $\mathbf{E}[\cdot]$ denotes statistical expected value; it also implies that these limits exist. Using again the strong law of large numbers yields

$$\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} y_k v_{k+j}^T = 0, \quad \text{w. p. 1,}$$

for all $j > 0$; again a similar expression holds for w_k.

2.5 Outline of Part I

This part of the thesis consists of six chapters (including this one) that deal with identification of LPV systems with affine parameter dependence. First, in Chapter 3 subspace identification methods are discussed. Subspace methods are well suited for the identification of state-space systems. They have proven their viability in the area of linear time-invariant system identification. However, for LPV systems, subspace methods suffer from the curse of dimensionality: the number of rows in the data matrices can be enormous. Therefore, in Chapter 4 a method is discussed that reduces the dimension of the data matrices considerably, at the expense of introducing an approximation error. Because of this approximation error the performance of the estimated LPV model is not optimal. Therefore, it is desirable to have a method that iteratively refines the LPV model. Such a method, based on nonlinear optimization, is described in Chapter 5. This method is a special case of the work of Lee and Poolla (1999) on identification of LFT descriptions. To illustrate the subspace and nonlinear optimization-based identification methods, several examples are presented in Chapter 6. Conclusions and recommendations for further research can be found in Chapter 7.

Chapter 3

Subspace Identification

Subspace identification is by now a well-accepted method for the identification of multivariable linear time-invariant (LTI) systems. In many cases these methods provide a good alternative to the classical nonlinear optimization-based prediction-error methods (Ljung 1999). Subspace methods do not require a particular parameterization of the system; this makes them numerically attractive and especially suitable for multivariable systems. Subspace methods can also be used to generate an initial starting point for the iterative prediction-error methods. This combination of subspace and prediction-error methods is a powerful tool for determining an LTI system from input and output measurements. Unfortunately, in many applications LTI systems do not provide an accurate description of the underlying real system. Therefore, identification methods for other descriptions, like time-varying and nonlinear systems, are needed. In recent years subspace identification methods have been developed for certain nonlinear systems: Wiener systems (Westwick and Verhaegen 1996; Chou and Verhaegen 1999), Hammerstein systems (Verhaegen and Westwick 1996) and bilinear systems (see Chapter 9 for references and an elaborate discussion). In this chapter, subspace identification methods are developed for LPV systems with affine parameter dependence.

Subspace identification methods for LTI systems can basically be classified into two different groups. The first group consists of methods that aim at recovering the column space of the extended observability matrix and use the shift-invariant structure of this matrix to estimate the matrices A and C; this group consists of the so-called MOESP methods (Verhaegen and Dewilde 1992; Verhaegen 1994; Chou and Verhaegen 1997). The methods in the second group aim at approximating the state sequence of the system and use this approximate state in a second step to estimate the system matrices; the methods that constitute this group are the N4SID methods (Moonen et al. 1989; Van Overschee and De Moor 1994, 1996), the CVA methods (Larimore 1983; Peternell, Scherrer, and Deistler 1996) and the orthogonal decomposition methods (Katayama and Picci 1999). In the appendix on page 227 a summary of linear subspace identification is given along with a discussion on its extension to nonlinear systems.

For LPV systems, the observability matrix does not have a shift-invariant structure like in the LTI case. Therefore, subspace identification methods for LPV systems aim at estimating the state sequence. Unfortunately, it is only possible to determine an approximation of the state sequence, because the influence of the initial state is unknown. This problem also occurs for LTI subspace methods based on estimating the state sequence, as pointed out by Van Overschee and De Moor (1994) and Picci and Katayama (1996). The approximation error that results from neglecting the initial state in LTI subspace methods, can be made arbitrarily small by taking a sufficiently large block size for the data matrices involved. A similar observation can be made for the LPV case. However, to keep the computational requirements of the LPV subspace methods within reasonable limits, it is often necessary to use a small block size: the number of rows in the data matrices used in the LPV subspace methods grows exponentially with this block size. Different LPV subspace methods use different ways of approximating the state, leading to different approximation errors.

The subspace methods discussed in this chapter are designed for an LPV system with a time-invariant output equation:

$$x_{k+1} = A \left[\begin{array}{c} x_k \\ p_k \otimes x_k \end{array} \right] + B \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right] + w_k, \qquad (3.1)$$

$$y_k = Cx_k + Du_k + v_k. \qquad (3.2)$$

Such a system corresponds to the system (2.1)–(2.2), with $C_i = 0$, $D_i = 0$ for $i = 1, 2, \ldots, s$; for simplification and with a slight abuse of notation, C and D are used in this chapter instead of C_0 and D_0. The identification problem that is addressed is: given a finite number of measurements of the input u_k, the output y_k and the time-varying parameters p_k, determine the system matrices A, B, C and D of (3.1)–(3.2) up to a similarity transformation.

The LPV system (3.1)–(3.2) is closely related to the bilinear system discussed in Part II of this thesis. The bilinear system can be regarded as a special type of LPV system in which the time-varying parameter p_k equals the input u_k. In fact, the theory for LPV subspace identification presented in this chapter is an extension of the theory for bilinear subspace identification developed by Favoreel (1999) and Chen and Maciejowski (2000b). This chapter provides a general framework for the derivation of subspace identification techniques for LPV systems. Since the bilinear system can be regarded as a special case of the LPV system, this framework can also be used to derive subspace identification methods for bilinear systems. A detailed description of bilinear subspace identification methods can be found in Chapter 9 of this thesis. Chapter 9 also describes how the bilinear subspace methods of Favoreel (1999) and Chen and Maciejowski (2000b) fit into this general framework.

This chapter is organized as follows: In Section 3.1 the notation is introduced, and the data equations for LPV systems are derived. Section 3.2 contains the basic ideas underlying subspace identification for LPV systems. Based on these ideas a number of subspace identification methods are derived in Section 3.3. Section 3.4 discusses the implementation of the LPV subspace identification methods. The chapter is concluded with a description of subspace methods for some special cases of the system (3.1)–(3.2): Section 3.5 discusses LPV systems with an

innovation-type of noise model and Section 3.6 discusses full state measurements and the case of a time-invariant B matrix.

3.1 Data Equations

The *data equations* describe the relation between structured matrices built from the available measurements of the input u_k, output y_k, and time-varying parameters p_k. These structured matrices are called the *data matrices*. In this section, the data matrices are defined, followed by a derivation of the data equations.

The definitions of the data matrices make use of the *Khatri-Rao* product (Khatri and Rao 1968). This product, denoted by the symbol $\odot$, is a column-wise Kronecker product for two matrices with an equal number of columns. The formal definition is as follows: let $M \in \mathbb{R}^{p \times q}$ and $N \in \mathbb{R}^{r \times q}$ be two arbitrary matrices, then the Khatri-Rao product of these matrices equals:

$$M \odot N = \begin{bmatrix} m_1 \otimes n_1 & m_2 \otimes n_2 & \dots & m_q \otimes n_q \end{bmatrix},$$

where m_i and n_i $(i = 1, 2, \dots, q)$ denote the columns of the matrices M and N, respectively.

In the derivations of the data equations, certain properties of the Kronecker product and the Khatri-Rao product are used. An important property of the Kronecker product is given below. Brewer (1978) describes some more properties of the Kronecker product.

Lemma 3.1 *(Horn and Johnson 1991, p. 244) Given the matrices $M \in \mathbb{R}^{p \times q}$, $N \in \mathbb{R}^{r \times s}$, $P \in \mathbb{R}^{q \times m}$, and $Q \in \mathbb{R}^{s \times t}$, then*

$$(MP \otimes NQ) = (M \otimes N)(P \otimes Q).$$

The proof of this lemma is an exercise in partitioned multiplication. From this lemma it is easy to deduce a similar property for the Khatri-Rao product.

Corollary 3.1 *Given the matrices $M \in \mathbb{R}^{p \times q}$, $N \in \mathbb{R}^{r \times q}$, $P \in \mathbb{R}^{q \times t}$, and $Q \in \mathbb{R}^{q \times t}$, then*

$$(MP \odot NQ) = (M \otimes N)(P \odot Q).$$

What follows is a list of definitions of data matrices used throughout this chapter.

$$
\begin{align}
P_k &:= [p_k, p_{k+1}, \dots, p_{k+N-1}] \in \mathbb{R}^{s \times N}, \tag{3.3} \\
X_k &:= [x_k, x_{k+1}, \dots, x_{k+N-1}] \in \mathbb{R}^{n \times N}, \tag{3.4} \\
X_{j|j} &:= \begin{bmatrix} X_j \\ P_j \odot X_j \end{bmatrix} \in \mathbb{R}^{(s+1)n \times N}, \tag{3.5} \\
X_{k+j|j} &:= \begin{bmatrix} X_{k+j-1|j} \\ P_{k+j} \odot X_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((s+1)^{k+1})n \times N}, \tag{3.6}
\end{align}
$$

$$U_k := [u_k, u_{k+1}, \ldots, u_{k+N-1}] \in \mathbb{R}^{m \times N}, \tag{3.7}$$

$$U_{j|j} := \begin{bmatrix} U_j \\ P_j \odot U_j \end{bmatrix} \in \mathbb{R}^{(s+1)m \times N}, \tag{3.8}$$

$$U_{k+j|j} := \begin{bmatrix} U_{k+j} \\ P_{k+j} \odot U_{k+j} \\ U_{k+j-1|j} \\ P_{k+j} \odot U_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((s+1)^{k+1}-1)m(s+1)/s \times N}, \tag{3.9}$$

$$W_k := [w_k, w_{k+1}, \ldots, w_{k+N-1}] \in \mathbb{R}^{n \times N}, \tag{3.10}$$

$$W_{j|j} := W_j \in \mathbb{R}^{n \times N}, \tag{3.11}$$

$$W_{k+j|j} := \begin{bmatrix} W_{k+j} \\ W_{k+j-1|j} \\ P_{k+j} \odot W_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((s+1)^{k+1}-1)n/s \times N}. \tag{3.12}$$

Note that the matrices $X_{k+j|j}$, $W_{k+j|j}$ and $U_{k+j|j}$ have different structures. The integer j in the subscript $k + j|j$ indicates that the data matrix contains measurements from time instant j and further; the integer k is called the *block size* and determines the number of block rows in the data matrix.

The following two examples give the reader an idea of the contents of these matrices:

$$U_{1+j|j} = \begin{bmatrix} U_{1+j} \\ P_{1+j} \odot U_{1+j} \\ U_j \\ P_j \odot U_j \\ P_{1+j} \odot \begin{bmatrix} U_j \\ P_j \odot U_j \end{bmatrix} \end{bmatrix},$$

$$U_{2+j|j} = \begin{bmatrix} U_{2+j} \\ P_{2+j} \odot U_{2+j} \\ U_{1+j|1} \\ P_{2+j} \odot U_{1+j|j} \end{bmatrix} = \begin{bmatrix} U_{2+j} \\ P_{2+j} \odot U_{2+j} \\ U_{1+j} \\ P_{1+j} \odot U_{1+j} \\ U_j \\ P_j \odot U_j \\ P_{1+j} \odot \begin{bmatrix} U_j \\ P_j \odot U_j \end{bmatrix} \\ P_{2+j} \odot \begin{bmatrix} U_{1+j} \\ P_{1+j} \odot U_{1+j} \\ U_j \\ P_j \odot U_j \\ P_{1+j} \odot \begin{bmatrix} U_j \\ P_j \odot U_j \end{bmatrix} \end{bmatrix} \end{bmatrix}.$$

These examples clearly illustrate that the number of rows in the data matrices grows exponentially with the block size k.

The data matrix $X_{k+j|j}$ has a simple relation with X_j.

Lemma 3.2 *Given the matrices (3.3)–(3.6). The relation between X_j and $X_{k+j|j}$ is*

$$X_{k+j|j} = \begin{bmatrix} X_j \\ P_{k+j|j} \odot X_j \end{bmatrix}, \tag{3.13}$$

where

$$
\begin{aligned}
P_{j|j} &:= P_j \in \mathbb{R}^{s \times N}, \\
P_{k+j|j} &:= \begin{bmatrix} P_{k+j-1|j} \\ P_{k+j}(1,:) \\ P_{k+j}(1,:) \odot P_{k+j-1|j} \\ \vdots \\ P_{k+j}(s,:) \\ P_{k+j}(s,:) \odot P_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((s+1)^{k+1}-1) \times N},
\end{aligned}
$$

and $P_k(i,:)$ denotes the ith row of the matrix P_k.

Proof: The proof follows from a simple induction argument. By definition

$$X_{j|j} = \begin{bmatrix} X_j \\ P_{j|j} \odot X_j \end{bmatrix}.$$

Suppose that equation (3.13) holds for $k = i$:

$$X_{i+j|j} = \begin{bmatrix} X_j \\ P_{i+j|j} \odot X_j \end{bmatrix}.$$

With this assumption it can be shown that (3.13) also holds for $k = i+1$:

$$
\begin{aligned}
X_{i+1+j|j} &= \begin{bmatrix} X_{i+j|j} \\ P_{i+1+j} \odot X_{i+j|j} \end{bmatrix} \\
&= \begin{bmatrix} \begin{bmatrix} X_j \\ P_{i+j|j} \odot X_j \end{bmatrix} \\ P_{i+1+j} \odot \begin{bmatrix} X_j \\ P_{i+j|j} \odot X_j \end{bmatrix} \end{bmatrix} \\
&= \begin{bmatrix} X_j \\ P_{i+j|j} \odot X_j \\ P_{i+1+j}(1,:) \odot X_j \\ P_{i+1+j}(1,:) \odot P_{i+j|j} \odot X_j \\ \vdots \\ P_{i+1+j}(s,:) \odot X_j \\ P_{i+1+j}(s,:) \odot P_{i+j|j} \odot X_j \end{bmatrix} \\
&= \begin{bmatrix} X_j \\ P_{i+1+j|j} \odot X_j \end{bmatrix}.
\end{aligned}
$$

Hence, equation (3.13) holds for every $k \geq 0$.   □

The previously defined data matrices are related to each other through the LPV system (3.1)–(3.2). These relations constitute the data equations. They are presented in the following two lemmas:

Lemma 3.3 *Given the matrices (3.3)–(3.12). For the LPV system (3.1)–(3.2) it holds that*

$$X_{k+j} = \Delta_k^x X_{k+j-1|j} + \Delta_k^u U_{k+j-1|j} + \Delta_k^w W_{k+j-1|j}, \qquad (3.14)$$

where

$$
\begin{aligned}
\Delta_1^x &:= [A_0, A_1, \ldots, A_s], \\
\Delta_k^x &:= [A_0 \Delta_{k-1}^x, A_1 \Delta_{k-1}^x, \ldots, A_s \Delta_{k-1}^x], \\
\Delta_1^u &:= [B_0, B_1, \ldots, B_s], \\
\Delta_k^u &:= [\Delta_1^u, A_0 \Delta_{k-1}^u, A_1 \Delta_{k-1}^u, \ldots, A_s \Delta_{k-1}^u], \\
\Delta_1^w &:= I_n, \\
\Delta_k^w &:= [\Delta_1^w, A_0 \Delta_{k-1}^w, A_1 \Delta_{k-1}^w, \ldots, A_s \Delta_{k-1}^w].
\end{aligned}
$$

Proof: The proof is based on an induction argument. From equation (3.1) it follows that equation (3.14) holds for $k = 1$,

$$
\begin{aligned}
X_{j+1} &= \Delta_1^x \begin{bmatrix} X_j \\ P_j \odot X_j \end{bmatrix} + \Delta_1^u \begin{bmatrix} U_j \\ P_j \odot U_j \end{bmatrix} + \Delta_1^w W_j \\
&= \Delta_1^x X_{j|j} + \Delta_1^u U_{j|j} + \Delta_1^w W_{j|j}.
\end{aligned}
$$

Now assume that equation (3.14) holds for $k = i$:

$$X_{i+j} = \Delta_i^x X_{i+j-1|j} + \Delta_i^u U_{i+j-1|j} + \Delta_i^w W_{i+j-1|j}.$$

With this assumption equation (3.14) also holds for $k = i + 1$:

$$
\begin{aligned}
X_{i+1+j} &= \Delta_1^x \begin{bmatrix} X_{i+j} \\ P_{i+j} \odot X_{i+j} \end{bmatrix} + \Delta_1^u \begin{bmatrix} U_{i+j} \\ P_{i+j} \odot U_{i+j} \end{bmatrix} + \Delta_1^w W_{i+j} \\
&= A_0(\Delta_i^x X_{i+j-1|j} + \Delta_i^u U_{i+j-1|j} + \Delta_i^w W_{i+j-1|j}) \\
&\quad + [A_1, \ldots, A_s] P_{i+j} \odot (\Delta_i^x X_{i+j-1|j} + \Delta_i^u U_{i+j-1|j} + \Delta_i^w W_{i+j-1|j}) \\
&\quad + \Delta_1^u \begin{bmatrix} U_{i+j} \\ P_{i+j} \odot U_{i+j} \end{bmatrix} + \Delta_1^w W_{i+j} \\
&= [A_0 \Delta_i^x, \ldots, A_s \Delta_i^x] \begin{bmatrix} X_{i+j-1|j} \\ P_{i+j} \odot X_{i+j-1|j} \end{bmatrix} \\
&\quad + [\Delta_1^u, A_0 \Delta_i^u, \ldots, A_s \Delta_i^u] \begin{bmatrix} U_{i+j} \\ P_{i+j} \odot U_{i+j} \\ U_{i+j-1|j} \\ P_{i+j} \odot U_{i+j-1|j} \end{bmatrix} \\
&\quad + [\Delta_1^w, A_0 \Delta_i^w, \ldots, A_s \Delta_i^w] \begin{bmatrix} W_{i+j} \\ W_{i+j-1|j} \\ P_{i+j} \odot W_{i+j-1|j} \end{bmatrix} \\
&= \Delta_{i+1}^x X_{i+j|j} + \Delta_{i+1}^u U_{i+j|j} + \Delta_{i+1}^w W_{i+j|j}.
\end{aligned}
$$

Hence, equation (3.14) holds for every $k \geq 0$.   □

Before stating the second lemma, some additional matrices are defined:

$$Y_k \quad := \quad [y_k, y_{k+1}, \ldots, y_{k+N-1}] \in \mathbb{R}^{\ell \times N}, \tag{3.15}$$

$$\mathcal{Y}_{j|j} \quad := \quad Y_j \in \mathbb{R}^{\ell \times N}, \tag{3.16}$$

$$\mathcal{Y}_{k+j|j} \quad := \quad \begin{bmatrix} Y_{k+j} \\ \mathcal{Y}_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{(k+1)\ell \times N}, \tag{3.17}$$

$$V_k \quad := \quad [v_k, v_{k+1}, \ldots, v_{k+N-1}] \in \mathbb{R}^{\ell \times N}, \tag{3.18}$$

$$\mathcal{V}_{j|j} \quad := \quad V_j \in \mathbb{R}^{\ell \times N}, \tag{3.19}$$

$$\mathcal{V}_{k+j|j} \quad := \quad \begin{bmatrix} V_{k+j} \\ \mathcal{V}_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{(k+1)\ell \times N}. \tag{3.20}$$

Note that the matrix $\mathcal{Y}_{k+j|j}$ is constructed from the same rows as the data matrix corresponding to y_k in linear subspace identification.

Lemma 3.4 *Given the matrices (3.3)–(3.12), and (3.15)–(3.20). For the LPV system (3.1)–(3.2) it holds that*

$$\mathcal{Y}_{k+j|j} = H_k^x X_{k+j-1|j} + H_k^u U_{k+j-1|j} + H_k^w W_{k+j-1|j} + G_k^u U_{k+j} + \mathcal{V}_{k+j|j}, \quad (3.21)$$

where

$$H_1^x \quad := \quad \begin{bmatrix} CA_0 & CA_1 & \cdots & CA_s \\ C & 0 & \cdots & 0 \end{bmatrix},$$

$$H_k^x \quad := \quad \begin{bmatrix} CA_0\Delta_{k-1}^x & CA_1\Delta_{k-1}^x & \cdots & CA_s\Delta_{k-1}^x \\ H_{k-1}^x & 0 & \cdots & 0 \end{bmatrix},$$

$$H_1^u \quad := \quad \begin{bmatrix} CB_0 & CB_1 & \cdots & CB_s \\ D & 0 & \cdots & 0 \end{bmatrix},$$

$$H_k^u \quad := \quad \begin{bmatrix} CB_0 & CB_1 & \cdots & CB_s & CA_0\Delta_{k-1}^u & CA_1\Delta_{k-1}^u & \cdots & CA_s\Delta_{k-1}^u \\ G_{k-1}^u & 0 & \cdots & 0 & H_{k-1}^u & 0 & \cdots & 0 \end{bmatrix},$$

$$H_1^w \quad := \quad \begin{bmatrix} C \\ 0 \end{bmatrix},$$

$$H_k^w \quad := \quad \begin{bmatrix} C & CA_0\Delta_{k-1}^w & CA_1\Delta_{k-1}^w & \cdots & CA_s\Delta_{k-1}^w \\ 0 & H_{k-1}^w & 0 & \cdots & 0 \end{bmatrix},$$

$$G_k^u \quad := \quad \begin{bmatrix} D \\ 0 \end{bmatrix},$$

and where Δ_k^x, Δ_k^u, and Δ_k^w are as in Lemma 3.3.

Proof: From equations (3.1) and (3.2) it follows that

$$\begin{aligned} Y_j &= CX_j + DU_j + V_j, \\ Y_{j+1} &= CX_{j+1} + DU_{j+1} + V_{j+1} \\ &= C(AX_{j|j} + BU_{j|j} + W_{j|j}) + DU_{j+1} + V_{j+1}. \end{aligned}$$

Therefore,

$$
\begin{aligned}
\mathcal{Y}_{j+1|j} &= \left[\begin{array}{c} Y_{j+1} \\ Y_j \end{array} \right] \\
&= \left[\begin{array}{cccc} CA_0 & CA_1 & \cdots & CA_s \\ C & 0 & \cdots & 0 \end{array} \right] \left[\begin{array}{c} X_j \\ P_j \odot X_j \end{array} \right] \\
&\quad + \left[\begin{array}{cccc} CB_0 & CB_1 & \cdots & CB_s \\ D & 0 & \cdots & 0 \end{array} \right] \left[\begin{array}{c} U_j \\ P_j \odot U_j \end{array} \right] \\
&\quad + \left[\begin{array}{c} C \\ 0 \end{array} \right] W_j + \left[\begin{array}{c} D \\ 0 \end{array} \right] U_{j+1} + \left[\begin{array}{c} V_{j+1} \\ V_j \end{array} \right] \\
&= H_1^x X_{j|j} + H_1^u U_{j|j} + H_1^w W_{j|j} + G_1^u U_{j+1} + \mathcal{V}_{j+1|j},
\end{aligned}
$$

and thus equation (3.21) holds for $k = 1$. To prove that it holds for every k, again an induction argument is used. Assume that equation (3.21) holds for $k = i$:

$$
\mathcal{Y}_{i+j|j} = H_i^x X_{i+j-1|j} + H_i^u U_{i+j-1|j} + H_i^w W_{i+j-1|j} + G_i^u U_{i+j} + \mathcal{V}_{i+j|j}.
$$

With this assumption it also holds for $k = i + 1$:

$$
\begin{aligned}
\mathcal{Y}_{i+1+j|j} &= \left[\begin{array}{c} Y_{i+j+1} \\ \mathcal{Y}_{i+j|j} \end{array} \right] \\
&= \left[\begin{array}{c} C(\Delta_{i+1}^x X_{i+j|j} + \Delta_{i+1}^u U_{i+j|j} + \Delta_{i+1}^w W_{i+j|j}) + DU_{i+1+j} + V_{i+1+j} \\ H_i^x X_{i+j-1|j} + H_i^u U_{i+j-1|j} + H_i^w W_{i+j-1|j} + G_i^u U_{i+j} + \mathcal{V}_{i+j|j} \end{array} \right] \\
&= \left[\begin{array}{cccc} CA_0 \Delta_i^x & CA_1 \Delta_i^x & \cdots & CA_s \Delta_i^x \\ H_i^x & 0 & \cdots & 0 \end{array} \right] \left[\begin{array}{c} X_{i+j-1|j} \\ P_{i+j} \odot X_{i+j-1|j} \end{array} \right] \\
&\quad + \left[\begin{array}{cccccccc} CB_0 & CB_1 & \cdots & CB_s & CA_0 \Delta_i^u & CA_1 \Delta_i^u & \cdots & CA_s \Delta_i^u \\ G_i^u & 0 & \cdots & 0 & H_i^u & 0 & \cdots & 0 \end{array} \right] \\
&\qquad \times \left[\begin{array}{c} U_{i+j} \\ P_{i+j} \odot U_{i+j} \\ U_{i+j-1|j} \\ P_{i+j} \odot U_{i+j-1|j} \end{array} \right] \\
&\quad + \left[\begin{array}{ccccc} C & CA_0 \Delta_i^w & CA_1 \Delta_i^w & \cdots & CA_s \Delta_i^w \\ 0 & H_i^w & 0 & \cdots & 0 \end{array} \right] \left[\begin{array}{c} W_{i+j} \\ W_{i+j-1|j} \\ P_{i+j} \odot W_{i+j-1|j} \end{array} \right] \\
&\quad + \left[\begin{array}{c} D \\ 0 \end{array} \right] U_{i+1+j} + \left[\begin{array}{c} V_{i+j+1} \\ \mathcal{V}_{i+j|j} \end{array} \right] \\
&= H_{i+1}^x X_{i+j|j} + H_{i+1}^u U_{i+j|j} + H_{i+1}^w W_{i+j|j} + G_{i+1}^u U_{i+1+j} + \mathcal{V}_{i+1+j|j}.
\end{aligned}
$$

This completes the proof. $\qquad\square$

3.2 Basic Ideas Behind the Methods

This section contains the basic ideas for deriving subspace identification methods for LPV systems. These methods consist of two steps. First, the state sequence is approximated by manipulating the data equations derived in Section 3.1. Second, the estimated state sequence is used to set up a linear least-squares problem

from which the system matrices of the LPV system follow. There are different ways to approximate the state; this gives rise to a number of different identification methods. These methods are discussed in Section 3.3. In this section, a general framework is created that allows a unified treatment of these different subspace identification methods.

3.2.1 Generalized Data Equation

Subspace identification is not based on only one data equation, but on a whole set of data equations starting at different time instants. Since these data equations all have the same structure, it is convenient to work with a generalized expression, the so-called *generalized data equation*.

To derive the generalized data equation, the result of Lemma 3.2 is combined with Lemma 3.4. For this purpose the matrix

$$\Gamma_k := H_k^x(\,:\,,1\!:n) = \begin{bmatrix} CA_0^k \\ \vdots \\ CA_0 \\ C \end{bmatrix} \tag{3.22}$$

is introduced. Note that for $k \geq n$ this matrix equals the extended observability matrix for a linear system, constructed from the matrices A_0 and C, with its block rows turned upside down. The matrix Γ_k is used to partition the matrix H_k^x defined in Lemma 3.4 as follows:

$$H_k^x = \begin{bmatrix} \Gamma_k & \widetilde{H}_k^x \end{bmatrix}, \tag{3.23}$$

where $\widetilde{H}_k^x$ denotes the remaining columns of H_k^x. Combining Lemma 3.2 with Lemma 3.4 transforms equation (3.21) on page 31 into

$$\begin{aligned} \mathcal{Y}_{k+j|j} &= \Gamma_k X_j + \widetilde{H}_k^x(P_{k+j-1|j} \odot X_j) + H_k^u U_{k+j-1|j} + G_k^u U_{k+j} \\ &\quad + H_k^w W_{k+j-1|j} + \mathcal{V}_{k+j|j}. \end{aligned} \tag{3.24}$$

This equation has the following basic structure:

$$Y = \Gamma X + H(P \odot X) + \Phi_U U + V, \tag{3.25}$$

where the time indices have been neglected. Equation (3.25) is called the *generalized data equation*; it consists of four parts:

1. A linear combination of the rows of the state sequence X. This linear combination is determined by Γ.

2. A linear combination of the rows of the matrix $P \odot X$. This linear combination is determined by H.

3. A linear combination of the rows of the input data matrix U. This linear combination is determined by Φ_U.

4. Noise terms, represented by V.

The generalized data equation plays a central role in the remainder of this chapter and also in Chapter 9.

An important step in LPV subspace identification is the approximation of the state sequence from an estimate of the product ΓX. Provided that Γ and X are of rank n, a singular value decomposition (SVD) of their product can be used to recover the state sequence X up to a similarity transformation: let the SVD of ΓX be

$$\Gamma X = U_n \Sigma_n V_n^T,$$

with $\Sigma_n \in \mathbb{R}^{n \times n}$ containing the nonzero singular values, then

$$\begin{aligned} \Gamma &= U_n \Sigma_n^{1/2} T, \\ X &= T^{-1} \Sigma_n^{1/2} V_n^T, \end{aligned}$$

with $T \in \mathbb{R}^{n \times n}$ an arbitrary nonsingular matrix that represents the similarity transformation. The requirement that Γ has full rank is fulfilled if the block size satisfies $k \geq n-1$, and the linear system constructed from A_0 and C is observable. The requirement that X has full rank can be thought of as a persistency of excitation condition, like in linear subspace methods (Verhaegen and Dewilde 1992; Jansson and Wahlberg 1998).

Since the state sequence can be recovered from the product ΓX, the goal is to approximate this product. The first step is to determine a matrix Z_X that can be constructed from the available measurements of u_k, p_k and y_k, such that it has approximately the same row space as the matrix X. An appropriate linear combination of the rows of Z_X then yields $\Phi_X Z_X \approx \Gamma X$. The matrix Z_X can be determined in different ways, by manipulation of the data equations derived in Section 3.1. The different possibilities are discussed in more detail in Section 3.3. In general, the following relation holds:

$$\Gamma X = \Phi_X Z_X + V_X + \Xi_X, \tag{3.26}$$

where V_X denotes noise terms that appear due to the manipulation of the data equations, and where Ξ_X is an error due to neglecting certain terms in the data equations. Equation (3.26) will be clarified with an example. Based on equation (3.14) on page 30, a possible approximation would be $\Gamma_k X_{k+j} \approx \Gamma_k \Delta_k^u U_{k+j-1|j}$, that is, the influence of the noise $W_{k+j-1|j}$ and of the past history of the state sequence $X_{k+j-1|j}$ are neglected. With this choice, the matrices appearing on the right-hand side of equation (3.26) can be identified as follows:

$$\begin{aligned} \Phi_X &\leftarrow \Gamma_k \Delta_k^u, \tag{3.27} \\ Z_X &\leftarrow U_{k+j-1|j}, \tag{3.28} \\ V_X &\leftarrow \Gamma_k \Delta_k^w W_{k+j-1|j}, \tag{3.29} \\ \Xi_X &\leftarrow \Gamma_k \Delta_k^x X_{k+j-1|j}. \tag{3.30} \end{aligned}$$

Since the state sequence also appears in the term $P \odot X$ in the generalized data equation (3.25), the product $H(P \odot X)$ must also be approximated. Let $Z_{P \odot X}$ be a matrix constructed from the available measurements of u_k, p_k, and y_k, such

that it has approximately the same row space as the matrix $P \odot X$, then similar to equation (3.26):

$$H(P \odot X) = \Phi_{P \odot X} Z_{P \odot X} + V_{P \odot X} + \Xi_{P \odot X}, \tag{3.31}$$

where $V_{P \odot X}$ represents noise terms and $\Xi_{P \odot X}$ is an error due to neglecting certain terms.

Using equations (3.26) and (3.31), the generalized data equation (3.25) can be reformulated as

$$Y = \Phi_X Z_X + \Phi_{P \odot X} Z_{P \odot X} + \Phi_U U + \Phi_V Z_V + \Xi_X + \Xi_{P \odot X}, \tag{3.32}$$

with $\Phi_V Z_V = V + V_X + V_{P \odot X}$. This equation is used in the following subsection to explain how an approximation of the state sequence can be computed. The terms Ξ_X and $\Xi_{P \odot X}$ can be thought of as bias terms.

3.2.2 Approximating the State Sequence

From the previous discussion it follows that $\Gamma X \approx \Phi_X Z_X$, thus it seems natural to estimate the state sequence from an SVD of the product $\Phi_X Z_X$. The matrix Z_X in this product is known, because it is constructed from the available measurements. The matrix Φ_X on the other hand is unknown. To explain how this matrix can be determined, the matrices Φ and Ω are introduced:

$$\Phi \quad := \quad \begin{bmatrix} \Phi_X & \Phi_{P \odot X} & \Phi_U \end{bmatrix} \tag{3.33}$$

$$\Omega \quad := \quad \begin{bmatrix} Z_X \\ Z_{P \odot X} \\ U \end{bmatrix}. \tag{3.34}$$

Based on equation (3.32) on page 35, an estimate of Φ can be obtained by solving the linear least-squares problem

$$\min_{\Phi} \| Y - \Phi \Omega \|_F^2, \tag{3.35}$$

where $\| \cdot \|_F$ denotes the Frobenius norm. Recall that the Frobenius norm of a matrix $A \in \mathbb{R}^{m \times n}$ equals

$$\| A \|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,j}^2 \right)^{1/2} = \left(\mathrm{trace}(A A^T) \right)^{1/2} = \| \mathrm{vec}(A) \|_2,$$

where $a_{i,j}$ denotes the (i, j)th element of the matrix A.

The least-squares estimate of Φ equals

$$\widehat{\Phi} = Y \Omega^T \left(\Omega \Omega^T \right)^{-1}. \tag{3.36}$$

From this it is immediately clear that the matrix Ω must have full rank. This full rank condition can be considered as a kind of persistency of excitation requirement. It means that certain restrictions must be imposed on the sequences u_k

and p_k, and the system (2.1)–(2.2). The difference between Φ and its least-squares estimate $\widehat{\Phi}$ equals

$$\widehat{\Phi} - \Phi = (\Phi_V Z_V + \Xi_X + \Xi_{P \odot X})\Omega^T \left(\Omega\Omega^T\right)^{-1}.$$

So the estimation error depends on the noise term Z_V and the approximation errors Ξ_X and $\Xi_{P \odot X}$. If Ω is uncorrelated with Z_V in the sense that

$$\lim_{N \to \infty} \frac{1}{N} Z_V \Omega^T = 0, \tag{3.37}$$

where N is the number of columns in Z_V and Ω, then the estimation error $\widehat{\Phi} - \Phi$ becomes independent of the noise term Z_V for $N \to \infty$.

An estimate of Φ_X, denoted by $\widehat{\Phi}_X$, can now be constructed from the first set of columns of $\widehat{\Phi}$ that correspond to Z_X. Since the goal is to use $\widehat{\Phi}_X Z_X$ as an approximation for ΓX, it is important to analyze the error $\Gamma X - \widehat{\Phi}_X Z_X$. This error can be split into three parts

$$\begin{aligned} \Gamma X - \widehat{\Phi}_X Z_X &=& (\Gamma X - \Phi_X Z_X) + (\Phi_X Z_X - \widehat{\Phi}_X Z_X) \\ &=& \Xi_X + V_X + (\Phi_X - \widehat{\Phi}_X)Z_X. \end{aligned} \tag{3.38}$$

The first two terms come from the approximation $\Gamma X \approx \Phi_X Z_X$, the third term is due to the estimation of Φ_X. The following theorem gives some insight in the size of the third term in terms of the Frobenius norm:

Theorem 3.1 *Given equation (3.32) and the matrices Φ and Ω by equations (3.33) and (3.34), respectively. Let $\widehat{\Phi}$ be given by equation (3.36), and let the following conditions be satisfied:*

1. $\lim_{N \to \infty} \dfrac{1}{N} Z_V \Omega^T = 0,$

2. $\lim_{N \to \infty} \dfrac{1}{N} \Omega\Omega^T = M,$ *with $\|M\|_F < \infty$ and such that M has full row rank,*

3. $\limsup_{N \to \infty} \dfrac{1}{N} \|\Xi_X + \Xi_{P \odot X}\|_F^2 \leq \xi < \infty,$

4. $\sup_i \|\Omega(:,i)\|_2^2 = \gamma < \infty$ *with $\Omega(:,i)$ the ith column of the matrix Ω.*

Then

$$\limsup_{N \to \infty} \frac{1}{N} \left\|(\widehat{\Phi} - \Phi)\Omega\right\|_F^2 \leq r\xi, \tag{3.39}$$

with r the total number of rows of the matrix Ω.

Proof: From (3.32) and (3.36)

$$(\widehat{\Phi} - \Phi)\Omega = \frac{1}{N}(\Phi_V Z_V + \Xi_X + \Xi_{P \odot X})\Omega^T \left(\frac{1}{N}\Omega\Omega^T\right)^{-1}\Omega,$$

where it is assumed that N is sufficiently large such that the matrix $\Omega\Omega^T/N$ is invertible. Using some properties of the Frobenius norm, this becomes

$$\|(\widehat{\Phi} - \Phi)\Omega\|_F \quad \leq \quad \left\|\frac{1}{N}\Phi_V Z_V \Omega^T\right\|_F \left\|\left(\frac{1}{N}\Omega\Omega^T\right)^{-1}\right\|_F \|\Omega\|_F$$

$$+\|\Xi_X + \Xi_{P\odot X}\|_F \left\|\frac{1}{N}\Omega^T \left(\frac{1}{N}\Omega\Omega^T\right)^{-1}\Omega\right\|_F.$$

The next step is to evaluate the norms in this expression:

$$\|\Omega\|_F^2 \quad = \quad \sum_{i=1}^{N}\|\Omega(:,i)\|_2^2 \leq N\gamma,$$

$$\left\|\frac{1}{N}\Omega^T \left(\frac{1}{N}\Omega\Omega^T\right)^{-1}\Omega\right\|_F^2 \quad = \quad \text{trace}\left(\frac{1}{N}\Omega^T \left(\frac{1}{N}\Omega\Omega^T\right)^{-1}\Omega\right)$$

$$= \quad \text{trace}\left(\frac{1}{N}\Omega\Omega^T \left(\frac{1}{N}\Omega\Omega^T\right)^{-1}\right)$$

$$= \quad r.$$

Therefore,

$$\frac{1}{\sqrt{N}}\|(\widehat{\Phi} - \Phi)\Omega\|_F \quad \leq \quad \sqrt{\gamma}\left\|\frac{1}{N}\Phi_V Z_V \Omega^T\right\|_F \left\|\left(\frac{1}{N}\Omega\Omega^T\right)^{-1}\right\|_F$$

$$+\frac{\sqrt{r}}{\sqrt{N}}\|\Xi_X + \Xi_{P\odot X}\|_F.$$

Taking the limit for $N \to \infty$ and observing that

$$\lim_{N\to\infty}\left\|\frac{1}{N}\Phi_V Z_V \Omega^T\right\|_F \quad = \quad 0,$$

$$\limsup_{N\to\infty}\left\|\left(\frac{1}{N}\Omega\Omega^T\right)^{-1}\right\|_F \quad < \quad \infty,$$

yields the desired result.   □

This theorem shows that with the assumption (3.37), the error $(\widehat{\Phi} - \Phi)\Omega$ does not depend on the noise Z_V; it only depends on the approximation errors Ξ_X and $\Xi_{P\odot X}$.

If equation (3.37) is not satisfied, it is still possible to remove the noise term Z_V from the error $(\widehat{\Phi} - \Phi)\Omega$, by using an appropriate instrumental variable matrix (Söderström and Stoica 1983). An instrumental variable matrix for the least-squares problem (3.35) is a matrix, denoted by Z_{IV}, such that the product ΩZ_{IV}^T has full row rank and such that

$$\lim_{N\to\infty}\frac{1}{N}Z_V Z_{IV}^T = 0.$$

The instrumental variable matrix is used to determine an estimate of Φ by solving the linear least-squares problem

$$\min_{\Phi} \left\| (Y - \Phi\Omega) Z_{IV}^T \right\|_F^2 ,$$

which has the solution

$$\widehat{\Phi} = Y Z_{IV}^T Z_{IV} \Omega^T \left(\Omega Z_{IV}^T Z_{IV} \Omega^T \right)^{-1}. \tag{3.40}$$

Along the same lines as the proof of Theorem 3.1, it can be verified that the estimation error $\widehat{\Phi} - \Phi$ becomes independent of the noise term Z_V for $N \to \infty$.

The idea of using a set of instrumental variables Z_{IV} in LPV subspace identification is based on a similar idea by Chen and Maciejowski (2000a) for bilinear subspace identification; although this idea is somewhat hidden in the description of their algorithm. Their method for bilinear systems is briefly discussed in Subsection 9.2.3.

From the previous discussion it follows that an estimate $\widehat{\Phi}_X$ can be obtained by either using equation (3.36) or (3.40). The next step is to approximate the state sequence by computing the SVD of $\widehat{\Phi}_X Z_X$. Let this SVD be given by

$$\widehat{\Phi}_X Z_X = \left[\begin{array}{cc} U_1 & U_2 \end{array} \right] \left[\begin{array}{cc} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{array} \right] \left[\begin{array}{c} V_1^T \\ V_2^T \end{array} \right], \tag{3.41}$$

where $\Sigma_1 \in \mathbb{R}^{n \times n}$ is a diagonal matrix, then an approximation of the state sequence is given by

$$\widehat{X} = \Sigma_1^{1/2} V_1^T .$$

The nonzero singular values Σ_2 result from the noise and the approximation errors Ξ_X and $\Xi_{P \odot X}$. If the noise and the approximation errors are not too large, the singular values contained in Σ_1 will be much larger than the ones in Σ_2. In this case, the order n of the system can be determined from this SVD, by locating a gap in the singular value plot.

The estimate of the state $\widehat{X}$ is only an approximation of the real state sequence, because of the relation (3.38) on page 36. From this equation it follows that the error on the state can be divided into three parts:

- the error due to the difference between Φ_X and $\widehat{\Phi}_X$, caused by Ξ_X and $\Xi_{P \odot X}$;

- the error due to neglecting certain terms in the approximation of ΓX (the term Ξ_X);

- the error due to the noise (the term V_X).

The approximation errors Ξ_X and $\Xi_{P \odot X}$ depend of course on the particular choice of the matrices Z_X and $Z_{P \odot X}$. These matrices can be constructed from the data matrices $U_{k+j|j}$ and $P_{k+j|j}$ defined in Section 3.1 in different ways. Since the number of rows in these data matrices grows exponentially with the block size, the number of rows in Z_X and $Z_{P \odot X}$ can become very large. A very large number

of rows limits the practical applicability of the identification method. Therefore, Z_X and $Z_{P \odot X}$ should be chosen carefully. The particular choice of Z_X and $Z_{P \odot X}$ is always a trade-off between the size of the approximation errors Ξ_X and $\Xi_{P \odot X}$, and the dimensions of the matrices Z_X and $Z_{P \odot X}$. In Section 3.3 some different choices for Z_X and $Z_{P \odot X}$ are discussed.

For reasonable signal-to-noise ratios, the error due to V_X can often be neglected, compared to the other two error sources that depend on Ξ_X and $\Xi_{P \odot X}$. Therefore, no attempt is made to remove the influence of V_X. Note that for the identification methods discussed in Section 3.5 that deal with a special type of noise model, the term V_X is always equal to zero, and the error in the state is completely determined by Ξ_X and $\Xi_{P \odot X}$.

3.2.3 Estimating the System Matrices

Once an estimate of the state sequence has been determined, the system matrices can be estimated by solving a set of linear least-squares problems. The system equations (3.1)–(3.2) motivate the following two least-squares problems:

$$\min_{A,B} \quad \|\widehat{X}_{k+1} - A\widehat{X}_{k|k} - BU_{k|k}\|_2^2,$$

$$\min_{C,D} \quad \|Y_k - C\widehat{X}_k - DU_k\|_2^2,$$

where $\widehat{X}_k$ denotes an estimate of the state sequence, and

$$\widehat{X}_{k|k} := \begin{bmatrix} \widehat{X}_k \\ P_k \odot \widehat{X}_k \end{bmatrix}.$$

The least-squares solutions are

$$[\widehat{A}, \widehat{B}] \quad = \quad \widehat{X}_{k+1} \begin{bmatrix} \widehat{X}_{k|k}^T & U_{k|k}^T \end{bmatrix} \left(\begin{bmatrix} \widehat{X}_{k|k} \\ U_{k|k} \end{bmatrix} \begin{bmatrix} \widehat{X}_{k|k}^T & U_{k|k}^T \end{bmatrix} \right)^{-1}, \quad (3.42)$$

$$[\widehat{C}, \widehat{D}] \quad = \quad Y_k \begin{bmatrix} \widehat{X}_k^T & U_k^T \end{bmatrix} \left(\begin{bmatrix} \widehat{X}_k \\ U_k \end{bmatrix} \begin{bmatrix} \widehat{X}_k^T & U_k^T \end{bmatrix} \right)^{-1}. \quad (3.43)$$

The system matrices are estimated up to a similarity transformation compared to the real system matrices, because the estimate of the state sequence is obtained by using an SVD as explained in the previous subsection. Obviously, the approximation used in estimating the state sequence will influence the estimates of the system matrices.

3.3   Identification Methods

As mentioned in the previous section, it is possible to derive different identification methods by changing the approximation used for the state sequence. In other words, by making different choices for the matrices Z_X and $Z_{P \odot X}$ in the generalized data equation (3.32) on page 35. This section describes four different identification methods that are based on different approximations of the state sequence.

3.3.1   A Two-Block Method

First, a two-block identification method is described. It is called a two-block method, because two sets of data matrices are used; one set consists of matrices starting at $j = 0$, the other set consists of matrices starting at $j = k$, where k is the block size used for the data matrices in both sets. The first set of matrices is used to approximate the state sequence and the second set is used to identify the system. For $j = k$ it follows from (3.24) on page 33 that the data matrices are related by

$$\mathcal{Y}_{2k|k} \; = \; \Gamma_k X_k + \widetilde{H}_k^x(P_{2k-1|k} \odot X_k) + H_k^u U_{2k-1|k} + G_k^u U_{2k} + H_k^w W_{2k-1|k} + \mathcal{V}_{2k|k}. \tag{3.44}$$

The state sequence X_k in this equation is approximated based on equation (3.14) on page 30 for $j = 0$:

$$X_k = \Delta_k^x X_{k-1|0} + \Delta_k^u U_{k-1|0} + \Delta_k^w W_{k-1|0}.$$

The terms $\Delta_k^w W_{k-1|0}$ and $\Delta_k^x X_{k-1|0}$ in this equation are neglected, and the approximation becomes

$$X_k \approx \Delta_k^u U_{k-1|0}. \tag{3.45}$$

The error due to neglecting $\Delta_k^w W_{k-1|0}$ depends of course on the signal-to-noise ratio. The error due to neglecting the term $\Delta_k^x X_{k-1|0}$ is bounded if the LPV system is uniformly exponentially stable, as shown in the following lemma:

Lemma 3.5 *Consider the LPV system (3.1)–(3.2). If the state equation (3.1) for $u_k = 0$ is uniformly exponentially stable, and for any j*

$$\sup_{k \geq j} \|u_k\|_2 < \infty, \qquad \sup_{k \geq j} \|w_k\|_2 < \infty,$$

then given a finite constant β, there exist a finite constant $\gamma > 0$ and a constant λ, $0 \leq \lambda < 1$, such that

$$\frac{1}{\sqrt{N}} \|\Delta_k^x X_{k+j-1|j}\|_F \leq \gamma \lambda^k,$$

for all k, j, such that $k \geq j$, and such that

$$\sup_k \|p_k\|_2 \leq \beta,$$

where Δ_k^x is as in Lemma 3.3 and $\| \cdot \|_F$ denotes the Frobenius norm.

Proof: Lemma 3.2 on page 28 yields

$$\Delta_k^x X_{k+j-1|j} = \Delta_k^x \left[\begin{array}{c} X_j \\ P_{k+j-1|j} \odot X_j \end{array} \right].$$

By the definition of the Frobenius norm,

$$\left\| \Delta_k^x X_{k+j-1|j} \right\|_F^2 = \sum_{q=j}^{N+j-1} \left\| \Delta_k^x \left[\begin{array}{c} x_q \\ p_{k+q-1|q} \odot x_q \end{array} \right] \right\|_2^2,$$

where $p_{k+q-1|q}$ is the $(q-j+1)$th column of $P_{k+j-1|j}$. First, it is shown that

$$\prod_{t=0}^{k-1}\left(A_0 + \sum_{i=1}^{s}[p_{t+q}]_i A_i\right) x_q = \Delta_k^x \left[\begin{array}{c} x_q \\ p_{k+q-1|q} \odot x_q \end{array}\right]. \qquad (3.46)$$

For $k=1$ equation (3.46) holds:

$$\Delta_1^x \left[\begin{array}{c} x_q \\ p_{q|q} \odot x_q \end{array}\right] = [A_0, A_1, \ldots, A_s] \left[\begin{array}{c} x_q \\ [p_q]_1 x_q \\ \vdots \\ [p_q]_s x_q \end{array}\right]$$

$$= \left(A_0 + \sum_{i=1}^{s}[p_q]_i A_i\right) x_q.$$

Assume that equation (3.46) holds for a certain k, then it can be proven that it holds for every k, by showing that it holds for $k+1$:

$$\prod_{t=0}^{k}\left(A_0 + \sum_{i=1}^{s}[p_{t+q}]_i A_i\right) x_q = \left(A_0 + \sum_{i=1}^{s}[p_{k+q}]_i A_i\right)\Delta_k^x\left[\begin{array}{c} x_q \\ p_{k+q-1|q} \odot x_q \end{array}\right]$$

$$= [A_0\Delta_k^x, A_1\Delta_k^x, \ldots, A_s\Delta_k^x] \left[\begin{array}{c} x_q \\ p_{k+q-1|q} \odot x_q \\ [p_{k+q}]_1 x_q \\ [p_{k+q}]_1 p_{k+q-1|q} \odot x_q \\ \vdots \\ [p_{k+q}]_s x_q \\ [p_{k+q}]_s p_{k+q-1|q} \odot x_q \end{array}\right]$$

$$= \Delta_{k+1}^x \left[\begin{array}{c} x_q \\ p_{k+q|q} \odot x_q \end{array}\right].$$

Since the state equation (3.1) is uniformly exponentially stable, due to Corollary 2.1 on page 21 there exist a finite constant $\kappa > 0$ and a constant λ, $0 \le \lambda < 1$, such that for every $k > 0$

$$\left\|\prod_{t=0}^{k-1}\left(A_0 + \sum_{i=1}^{s}[p_{t+q}]_i A_i\right)\right\|_2 \le \kappa\lambda^k.$$

Thus, the left-hand side of equation (3.46) satisfies

$$\left\|\prod_{t=0}^{k-1}\left(A_0 + \sum_{i=1}^{s}[p_{t+q}]_i A_i\right) x_q\right\|_2 \le \|x_q\|_2 \cdot \kappa\lambda^k.$$

Now putting everything together, we get

$$\left\|\Delta_k^x X_{k+j-1|j}\right\|_F \le \sqrt{N}\gamma\lambda^k,$$

with

$$\gamma = \kappa \sup_{q \geq j} ||x_q||_2,$$

The constant γ is finite, because by Corollary 2.2 on page 21 the system (3.1)–(3.2) is uniformly bounded-input, bounded-output stable, and because the signals p_k, u_k, and w_k are bounded. □

This lemma shows that if the LPV system is uniformly asymptotically stable, the perturbation due to neglecting $\Delta_k^x X_{k+j-1|j}$ decreases with increasing block size k. However, it does not give the rate of decrease: it only shows that there exists a λ, but does not provide its value. The rate of decrease can be quantified by using

$$\sigma = \sup_{q \geq j} \left\| A_0 + \sum_{i=1}^{s} [p_q]_i A_i \right\|_2, \tag{3.47}$$

provided that $\sigma < 1$. It is easy to deduce from the proof of Lemma 3.5 that

$$\frac{1}{\sqrt{N}} \| \Delta_k^x X_{k+j-1|j} \| \leq \beta \sigma^k,$$

for some finite constant $\beta > 0$. The idea of using (3.47) to analyze the error with respect to the block size k has been put forward by Chen and Maciejowski (2000a) in the context of subspace identification for bilinear systems (see also Chen and Maciejowski 2000b). It is however a conservative approach, because there exist uniformly asymptotically stable LPV systems with $\sigma > 1$.

The two-block method is obtained by using the approximation (3.45) for the terms X_k and $(P_{2k-1|k} \odot X_k)$ in equation (3.44). The next theorem shows the relation of the two-block method with the generalized data equation (3.32) and with the procedure outlined in Subsection 3.2.2 to approximate the state sequence. Recall that the key step in this procedure is the estimation of the matrix Φ given by equation (3.33) on page 35.

Theorem 3.2 *Given the bounded-input, bounded-output stable system (3.1)–(3.2) with bounded sequences p_k, u_k, w_k, and v_k, the data equations of Lemmas 3.2, 3.3, and 3.4, and the generalized data equation (3.32):*

$$Y = \Phi_X Z_X + \Phi_{P \odot X} Z_{P \odot X} + \Phi_U U + \Phi_V Z_V + \Xi_X + \Xi_{P \odot X},$$

with the following substitutions:

$$
\begin{aligned}
Y &\leftarrow \mathcal{Y}_{2k|k}, \\
Z_X &\leftarrow U_{k-1|0}, \\
Z_{P \odot X} &\leftarrow P_{2k-1|k} \odot U_{k-1|0}, \\
U &\leftarrow \begin{bmatrix} U_{2k-1|k} \\ U_{2k} \end{bmatrix}, \\
Z_V &\leftarrow \begin{bmatrix} W_{2k-1|k} \\ \mathcal{V}_{2k|k} \\ W_{k-1|0} \\ P_{2k-1|k} \odot W_{k-1|0} \end{bmatrix}.
\end{aligned}
$$

Then

$$\lim_{N \to \infty} \frac{1}{N} Z_V \Omega^T = 0 \quad with \quad \Omega := \begin{bmatrix} Z_X \\ Z_{P \odot X} \\ U \end{bmatrix} \tag{3.48}$$

holds with probability one, and the approximation errors are

$$\begin{aligned} \Xi_X &\leftarrow \Gamma_k \Delta_k^x X_{k-1|0}, \\ \Xi_{P \odot X} &\leftarrow (\widetilde{H}_k^x \otimes \Delta_k^x)(P_{2k-1|k} \odot X_{k-1|0}), \end{aligned}$$

with Γ_k given by (3.22) and $\widetilde{H}_k^x$ given by (3.23).
   If the matrix Ω has full row rank, the matrix

$$\Phi := \begin{bmatrix} \Phi_X & \Phi_{P \odot X} & \Phi_U \end{bmatrix}$$

can be estimated as

$$\widehat{\Phi} = Y \Omega^T \left(\Omega \Omega^T \right)^{-1},$$

and $(\widehat{\Phi} - \Phi)\Omega$ does not depend on the noise term Z_V with probability one for $N \to \infty$.

Proof: The proof of (3.48) follows from the fact that p_k and u_k are independent from the noise sequences v_k and w_k, in the sense of equation (2.8) on page 23. The expressions for the approximation errors follow by comparing equation (3.44) on page 40 with the generalized data equation (3.32). The approximations errors are bounded, because the system is bounded-input, bounded-output stable. The remaining part of the proof follows by applying Theorem 3.1 on page 36.   □

3.3.2   A Three-Block Method

To reduce the error in approximating the state sequence, in addition to the input data, the output data can be used. This idea stems from Chen and Maciejowski (2000b), who used it in bilinear subspace identification. It is based on decomposing X_{j+1} into two parts, as follows:

$$X_{k+j} = MCX_{k+j} + (I - MC)X_{k+j},$$

for some arbitrary matrix M. The first part can be described by the output equation $Y_{k+j} = CX_{k+j} + DU_{k+j} + V_{k+j}$, the second part by equation (3.14):

$$\begin{aligned} X_{k+j} &= M(Y_{k+j} - DU_{k+j} - V_{k+j}) \\ &\quad + (I - MC)\Big(\Delta_k^u U_{k+j-1|j} + \Delta_k^x X_{k+j-1|j} + \Delta_k^w W_{k+j-1|j}\Big). \end{aligned} \tag{3.49}$$

By neglecting $\Delta_k^x X_{k+j-1|j}$, $\Delta_k^w W_{k+j-1|j}$, and also V_{k+j}, the state can be approximated as

$$X_{k+j} \approx M(Y_{k+j} - DU_{k+j}) + (I - MC)\Delta_k^u U_{k+j-1|j}. \tag{3.50}$$

Since part of the state is modeled by Y_{k+j} and U_{k+j}, more information is used to approximate the state, compared with equation (3.45). Therefore, it is expected that the approximation error becomes smaller. The matrix M can be arbitrary. It

does not need to be specified, because it will end up being part of Φ, which is to be estimated.

In the special case that C has full column rank, taking $M = (C^T C)^{-1} C^T$ in equation (3.49) shows that X_{k+j} can be determined from Y_{k+j} and U_{k+j}. Of course, the noise term V_{k+j} should be taken care of in an appropriate way. However, if C has full column rank, it is not interesting to use the approach from the previous section, because much simpler algorithms exist, as described in Subsection 3.6.1.

If $M = 0$, the approximation (3.50) in equation (3.44) on page 40 yields a two-block method that equals the one presented in the previous subsection; for $M \neq 0$ the method is different and is explained below. The term $(P_{2k-1|k} \odot X_k)$ in (3.44) is approximated as

$$
\begin{aligned}
(P_{2k-1|k} \odot X_k) \quad \approx \quad & (I \otimes M)(P_{2k-1|k} \odot Y_k) - (I \otimes MD)(P_{2k-1|k} \odot U_k) \\
& + \Big(I \otimes (I - MC) \Delta_k^u \Big)(P_{2k-1|k} \odot U_{k-1|0}),
\end{aligned} \tag{3.51}
$$

where Corollary 3.1 on page 27 has been used. This approximation corresponds to making the following identifications for the terms in the generalized data equation (3.32):

$$
\begin{aligned}
Y \quad &\leftarrow \quad \mathcal{Y}_{2k|k}, \\
Z_X \quad &\leftarrow \quad \begin{bmatrix} Y_k \\ U_k \\ U_{k-1|0} \end{bmatrix}, \\
Z_{P \odot X} \quad &\leftarrow \quad \begin{bmatrix} P_{2k-1|k} \odot Y_k \\ P_{2k-1|k} \odot U_k \\ P_{2k-1|k} \odot U_{k-1|0} \end{bmatrix}, \\
U \quad &\leftarrow \quad \begin{bmatrix} U_{2k-1|k} \\ U_{2k} \end{bmatrix}, \\
Z_V \quad &\leftarrow \quad \begin{bmatrix} W_{2k-1|k} \\ \mathcal{V}_{2k|k} \\ W_{k-1|0} \\ V_k \\ P_{2k-1|k} \odot W_{k-1|0} \\ P_{2k-1|k} \odot V_k \end{bmatrix}.
\end{aligned}
$$

However, because Y_k is part of Z_X,

$$
\lim_{N \to \infty} \frac{1}{N} Z_V Z_X^T \neq 0.
$$

This means that if the matrix Φ is estimated as in equation (3.36) on page 35, the estimation error depends on the noise term Z_V. To avoid this, an instrumental variable matrix is needed, and the matrix Φ must be estimated as in equation (3.40) on page 38.

Since a part of Z_X can be used as an instrumental variable matrix, Z_X is divided into two parts:

$$
\begin{aligned}
Z_{X,R} &\leftarrow Y_k, \\
Z_{X,IV} &\leftarrow \begin{bmatrix} U_k \\ U_{k-1|0} \end{bmatrix},
\end{aligned}
$$

where $Z_{X,IV}$ is the part that can be used as an instrumental variable matrix, because with this substitution

$$
\lim_{N \to \infty} \frac{1}{N} Z_V Z_{X,IV}^T = 0,
$$

with probability one.

Recall from the discussion in Subsection 3.2.2 that the instrumental variable matrix Z_{IV} must be such that the matrix

$$
\Omega Z_{IV}^T = \begin{bmatrix} Z_X \\ Z_{P \odot X} \\ U \end{bmatrix} Z_{IV}^T
$$

has full row rank. To construct Z_{IV}, the matrices $Z_{X,IV}$, $Z_{P \odot X}$, and U can be used, but this is not enough. To fulfill the rank condition, Z_{IV} has to have at least the same number of rows as the matrix Ω. Therefore, an additional instrumental variable matrix is needed; let this matrix be denoted by $Z_{A,IV}$. Hence,

$$
Z_{IV} = \begin{bmatrix} Z_{A,IV} \\ Z_{X,IV} \\ Z_{P \odot X} \\ U \end{bmatrix}.
$$

The only data left that can be used as additional instrumental variables $Z_{A,IV}$ are

$$
Z_{A,IV} = \begin{bmatrix} P_{2k-1|k} \odot U_{2k-1|k} \\ P_{2k-1|k} \odot U_{2k} \end{bmatrix}. \tag{3.52}
$$

This matrix has a huge number of rows, because it contains Khatri-Rao products that involve matrices of which the number of rows grows exponentially with the block size k. A better choice, resulting in a significantly smaller number of rows, is to divide the data into three blocks, and use the additional block as instrumental variable. This gives rise to a three-block method.

As the name suggests, the three-block method is based on three sets of data matrices. Compared with the two-block method, all data matrices are shifted by k samples, and an additional set of data matrices is added. The set that consists of matrices starting at $j = 2k$ is used to identify the system, the set that consists of matrices starting at $j = k$ is used to approximate the state sequence X_{2k}, and the third set, starting at $j = 0$, is used as additional instrumental variables:

$$
Z_{A,IV} \leftarrow \begin{bmatrix} Y_k \\ U_k \\ U_{k-1|0} \end{bmatrix}. \tag{3.53}
$$

Note the much smaller dimension compared to the choice of (3.52): the number of rows in (3.52) equals $d_k(m + d_k m(s + 1)/s)$, with $d_k = (s + 1)^{k+1} - 1$, the number of rows in (3.53) equals $\ell + m + d_k m(s+1)/s$. The choice of instrumental variables (3.53) leads to the following result:

Theorem 3.3 *Given the bounded-input, bounded-output stable system (3.1)–(3.2) with bounded sequences p_k, u_k, w_k, and v_k, the data equations of Lemmas 3.2, 3.3, and 3.4, and the generalized data equation (3.32):*

$$Y = \Phi_X Z_X + \Phi_{P \odot X} Z_{P \odot X} + \Phi_U U + \Phi_V Z_V + \Xi_X + \Xi_{P \odot X},$$

with

$$Z_X := \begin{bmatrix} Z_{X,R} \\ Z_{X,IV} \end{bmatrix},$$

and with the following substitutions:

$$
\begin{aligned}
Y &\leftarrow \mathcal{Y}_{3k|2k}, \\
Z_{A,IV} &\leftarrow \begin{bmatrix} Y_k \\ U_k \\ U_{k-1|0} \end{bmatrix}, \\
Z_{X,R} &\leftarrow Y_{2k}, \\
Z_{X,IV} &\leftarrow \begin{bmatrix} U_{2k} \\ U_{2k-1|k} \end{bmatrix}, \\
Z_{P \odot X} &\leftarrow \begin{bmatrix} P_{3k-1|2k} \odot Y_{2k} \\ P_{3k-1|2k} \odot U_{2k} \\ P_{3k-1|2k} \odot U_{2k-1|k} \end{bmatrix}, \\
U &\leftarrow \begin{bmatrix} U_{3k-1|2k} \\ U_{3k} \end{bmatrix}, \\
Z_V &\leftarrow \begin{bmatrix} W_{3k-1|2k} \\ \mathcal{V}_{3k|2k} \\ W_{2k-1|k} \\ V_{2k} \\ P_{3k-1|2k} \odot W_{2k-1|k} \\ P_{3k-1|2k} \odot V_{2k} \end{bmatrix}.
\end{aligned}
$$

Then

$$\lim_{N \to \infty} \frac{1}{N} Z_V Z_{IV}^T = 0 \quad \text{with} \quad Z_{IV} := \begin{bmatrix} Z_{A,IV} \\ Z_{X,IV} \\ Z_{P \odot X} \\ U \end{bmatrix}$$

holds with probability one, and the approximation errors are

$$
\begin{aligned}
\Xi_X &\leftarrow \Gamma_k (I - MC) \Delta_k^x X_{2k-1|k}, \\
\Xi_{P \odot X} &\leftarrow \left(\widetilde{H}_k^x \otimes (I - MC) \Delta_k^x \right) (P_{3k-1|2k} \odot X_{2k-1|k}),
\end{aligned}
$$

with Γ_k given by (3.22) and $\widetilde{H}_k^x$ given by (3.23).

If the matrix ΩZ_{IV}^T has full row rank, the matrix

$$\Phi := \begin{bmatrix} \Phi_X & \Phi_{P \odot X} & \Phi_U \end{bmatrix}$$

can be estimated as

$$\widehat{\Phi} = Y Z_{IV}^T Z_{IV} \Omega^T \left(\Omega Z_{IV}^T Z_{IV} \Omega^T \right)^{-1},$$

and $(\widehat{\Phi} - \Phi)\Omega$ does not depend on the noise term Z_V with probability one for $N \to \infty$.

The proof is similar to the proof of Theorem 3.2.

3.3.3 A Two-Stage Three-Block Method

Another way to use three sets of data matrices, is to approximate the state sequence in two stages. This method is therefore called the two-stage three-block method. The idea of a two-stage approximation has been used by Chen and Maciejowski (2000b) in bilinear subspace identification methods for combined deterministic and stochastic systems. Their three-block and four-block identification algorithms are based on a two-stage approximation, although that might not be immediately clear from their exposition. Their four-block algorithms are further discussed in Subsection 9.2.3.

Again, three sets of data matrices are constructed; one starting at $j = 0$, another at $j = k$, and a third at $j = 2k$. The data matrices that start at $j = 2k$ are used to identify the system matrices. Because of equation (3.24) on page 33, these matrices are related by

$$
\begin{aligned}
\mathcal{Y}_{3k|2k} &= \Gamma_k X_{2k} + \widetilde{H}_k^x (P_{3k-1|2k} \odot X_{2k}) + H_k^u U_{3k-1|2k} + G_k^u U_{3k} \\
&\quad + H_k^w W_{3k-1|2k} + \mathcal{V}_{3k|2k}.
\end{aligned}
\tag{3.54}
$$

The data matrices that start at $j = k$ are used to approximate X_{2k} in two stages, and the matrices that start at $j = 0$ form an additional set of instrumental variables.

The first approximation stage is based on equation (3.14) on page 30 for $j = k$:

$$X_{2k} = \Delta_k^x X_{2k-1|k} + \Delta_k^u U_{2k-1|k} + \Delta_k^w W_{2k-1|k}.$$

Combining this with Lemma 3.2 on page 28 gives

$$X_{2k} = \Omega_k^x X_k + \widetilde{\Delta}_k^x (P_{2k-1|k} \odot X_k) + \Delta_k^u U_{2k-1|k} + \Delta_k^w W_{2k-1|k},$$

where $\Delta_k^x = \begin{bmatrix} \Omega_k^x & \widetilde{\Delta}_k^x \end{bmatrix}$. This shows that the rows of X_{2k} are linear combinations of the rows of the matrices

$$X_k, \quad (P_{2k-1|k} \odot X_k), \quad U_{2k-1|k}, \quad \text{and} \quad W_{2k-1|k}. \tag{3.55}$$

The second stage consists of approximating X_k. For this the expression (3.44) on page 40 is used. It is easy to see that

$$X_k = \Gamma_k^\dagger \left(\mathcal{Y}_{2k|k} - \widetilde{H}_k^x (P_{2k-1|k} \odot X_k) - H_k^u U_{2k-1|k} - H_k^w W_{2k-1|k} - G_k^u U_{2k} - \mathcal{V}_{2k|k} \right).$$

Hence, the rows of X_k are linear combinations of the rows of the matrices

$$\mathcal{Y}_{2k|k}, \quad (P_{2k-1|k} \odot X_k), \quad U_{2k-1|k}, \quad U_{2k}, \quad W_{2k-1|k}, \quad \text{and} \quad \mathcal{V}_{2k|k}.$$

Comparing this with the matrices (3.55) we see that there is an overlap between the terms; this is an advantage in keeping the dimension of the data matrices in equation (3.32) on page 35 low. The result is that the rows of X_{2k} are linear combinations of the rows of the matrices

$$\mathcal{Y}_{2k|k}, \quad (P_{2k-1|k} \odot X_k), \quad U_{2k-1|k}, \quad U_{2k}, \quad W_{2k-1|k}, \quad \text{and} \quad \mathcal{V}_{2k|k}.$$

To complete the second stage, the state X_k in the term $(P_{2k-1|k} \odot X_k)$ has to be approximated. To keep the dimensions low, the approximation is based on equation (3.51) on page 44.

The final step is to approximate $(P_{3k-1|2k} \odot X_{2k})$ in equation (3.54). The state X_{2k} in this term could again be approximated using the two-stage method. However, the number of rows in $Z_{P \odot X}$ would be enormous, because of the Khatri-Rao product with $P_{3k-1|2k}$. To avoid this, the approximation of $(P_{3k-1|2k} \odot X_{2k})$ is based on an expression similar to (3.51) on page 44.

The two-stage three-block method that results from the analysis given above, is summarized in the following theorem:

Theorem 3.4 *Given the bounded-input, bounded-output stable system (3.1)–(3.2) with bounded sequences p_k, u_k, w_k, and v_k, the data equations of Lemmas 3.2, 3.3, and 3.4, and the generalized data equation (3.32):*

$$Y = \Phi_X Z_X + \Phi_{P \odot X} Z_{P \odot X} + \Phi_U U + \Phi_V Z_V + \Xi_X + \Xi_{P \odot X},$$

with

$$Z_X := \begin{bmatrix} Z_{X,R} \\ Z_{X,IV} \end{bmatrix},$$

and with the following substitutions:

$$
\begin{aligned}
Y &\leftarrow \mathcal{Y}_{3k|2k}, \\
Z_{A,IV} &\leftarrow \begin{bmatrix} Y_k \\ U_k \\ U_{k-1|0} \end{bmatrix}, \\
Z_{X,R} &\leftarrow \mathcal{Y}_{2k|k}, \\
Z_{X,IV} &\leftarrow \begin{bmatrix} P_{2k-1|k} \odot Y_k \\ P_{2k-1|k} \odot U_k \\ P_{2k-1|k} \odot U_{k-1|0} \\ U_{2k-1|k} \\ U_{2k} \end{bmatrix},
\end{aligned}
$$

$$
Z_{P \odot X} \quad \leftarrow \quad
\begin{bmatrix}
P_{3k-1|2k} \odot Y_{2k} \\
P_{3k-1|2k} \odot U_{2k} \\
P_{3k-1|2k} \odot U_{2k-1|k}
\end{bmatrix},
$$

$$
U \quad \leftarrow \quad
\begin{bmatrix}
U_{3k-1|2k} \\
U_{3k}
\end{bmatrix},
$$

$$
Z_V \quad \leftarrow \quad
\begin{bmatrix}
W_{3k-1|2k} \\
\mathcal{V}_{3k|2k} \\
W_{2k-1|k} \\
\mathcal{V}_{2k|k} \\
P_{3k-1|2k} \odot W_{2k-1|k} \\
P_{3k-1|2k} \odot V_{2k} \\
P_{2k-1|k} \odot W_{k-1|0} \\
P_{2k-1|k} \odot V_k
\end{bmatrix}.
$$

Then

$$
\lim_{N \to \infty} \frac{1}{N} Z_V Z_{IV}^T = 0 \quad \text{with} \quad Z_{IV} :=
\begin{bmatrix}
Z_{A,IV} \\
Z_{X,IV} \\
Z_{P \odot X} \\
U
\end{bmatrix}
$$

holds with probability one, and the approximation errors are

$$
\Xi_X \quad \leftarrow \quad \Gamma_k \Big((\widetilde{\Delta}_k^x - \Omega_k^x \Gamma_k^\dagger \widetilde{H}_k^x) \otimes (I - MC)\Delta_k^x \Big)(P_{2k-1|k} \odot X_{k-1|0}),
$$

$$
\Xi_{P \odot X} \quad \leftarrow \quad \Big(\widetilde{H}_k^x \otimes (I - MC)\Delta_k^x \Big)(P_{3k-1|2k} \odot X_{2k-1|k}),
$$

with Γ_k given by (3.22) and $\widetilde{H}_k^x$ given by (3.23).

If the matrix ΩZ_{IV}^T has full row rank, the matrix

$$
\Phi := \begin{bmatrix} \Phi_X & \Phi_{P \odot X} & \Phi_U \end{bmatrix}
$$

can be estimated as

$$
\widehat{\Phi} = Y Z_{IV}^T Z_{IV} \Omega^T \left(\Omega Z_{IV}^T Z_{IV} \Omega^T \right)^{-1},
$$

and $(\widehat{\Phi} - \Phi)\Omega$ does not depend on the noise term Z_V with probability one for $N \to \infty$.

The proof is similar to the proof of Theorem 3.2.

3.3.4 A Two-Stage Three-Block Method with Reduced Dimensions

The dimensions of the data matrices used in the method of the previous subsection can be reduced by taking $M = 0$ instead of $M \neq 0$ in the approximation of the

terms $(P_{2k-1|k} \odot X_k)$ and $(P_{3k-1|2k} \odot X_{2k})$. In this case the data matrices become:

$$
\begin{aligned}
Y &\leftarrow \mathcal{Y}_{3k|2k}, \\
Z_{A,IV} &\leftarrow \begin{bmatrix} Y_k \\ U_k \\ U_{k-1|0} \end{bmatrix}, \\
Z_{X,R} &\leftarrow \mathcal{Y}_{2k|k}, \\
Z_{X,IV} &\leftarrow \begin{bmatrix} P_{2k-1|k} \odot U_{k-1|0} \\ U_{2k-1|k} \\ U_{2k} \end{bmatrix}, \\
Z_{P\odot X} &\leftarrow P_{3k-1|2k} \odot U_{2k-1|k}, \\
U &\leftarrow \begin{bmatrix} U_{3k-1|2k} \\ U_{3k} \end{bmatrix}, \\
Z_V &\leftarrow \begin{bmatrix} W_{3k-1|2k} \\ \mathcal{V}_{3k|2k} \\ W_{2k-1|k} \\ \mathcal{V}_{2k|k} \\ P_{3k-1|2k} \odot W_{2k-1|k} \\ P_{2k-1|k} \odot W_{k-1|0} \end{bmatrix}, \\
\Xi_X &\leftarrow \Gamma_k \Big((\widetilde{\Delta}_k^x - \Omega_k^x \Gamma_k^\dagger \widetilde{H}_k^x) \otimes \Delta_k^x \Big)(P_{2k-1|k} \odot X_{k-1|0}), \\
\Xi_{P\odot X} &\leftarrow \Big(\widetilde{H}_k^x \otimes \Delta_k^x \Big)(P_{3k-1|2k} \odot X_{2k-1|k}),
\end{aligned}
$$

and a similar result as presented in Theorem 3.4 can be derived.

3.3.5 Discussion

The four identification methods described above all differ in the way that they approximate the state sequence. Consequently, they all have different expressions for the approximation errors Ξ_X and $\Xi_{P\odot X}$. It would of course be of interest to compare the approximation errors of the different methods for finite block size k. Intuition suggests that including more terms to approximate the state sequence would lead to a smaller approximation error. Thus, the two-stage methods would have the smallest approximation errors. However, the complex expressions for the matrices Ξ_X and $\Xi_{P\odot X}$ make it difficult to prove this explicitly.

It is expected that the approximation errors, Ξ_X and $\Xi_{P\odot X}$, decrease with increasing block size k. This statement is supported by two facts. First, the description of the identification methods shows that the matrices Ξ_X and $\Xi_{P\odot X}$ result from neglecting the term $\Delta_k^x X_{k+j-1|j}$ for some j. Second, Lemma 3.5 shows that the Frobenius norm of $\Delta_k^x X_{k+j-1|j}$ decreases with increasing block size k. To make a precise statement, the propagation of the error due to neglecting the term $\Delta_k^x X_{k+j-1|j}$ should be computed. However, it is difficult to come up with norm bounds, because the block size varies, and thus the number of rows in the matrices involved varies. Furthermore, it is not sufficient to have a norm bound on Ξ_X and $\Xi_{P\odot X}$, because the bound in Theorem 3.1 also depends on the number of rows in the data matrix Ω.

Because of the difficulties in making precise statements about the approximation errors, the hypothesis that the approximation error decreases with increasing block size will be verified by performing a set of Monte-Carlo simulations. These simulations are presented in Subsection 6.1.1. This subsection also compares the approximation errors of the different identification methods.

The approximation error is not the only concern in choosing the block size. Assuming that the approximation error decreases with increasing block size k, a large block size is desirable. However, a large block size increases the number of rows in the data matrices. Since as discussed earlier, the number of rows in these matrices grows exponentially with the block size, huge dimensional matrices can result. For larger block sizes, the computer memory required to store these matrices is enormous. Therefore, the block size should be kept small. Hence, choosing the block size is a trade-off between accuracy and practical usefulness.

The choice of the identification method is also a trade-off between accuracy and practical usefulness. The identification methods differ in the way the matrices Z_X and $Z_{P \odot X}$ are chosen. Consequently, they differ in accuracy and in computer memory requirements.

Because of the huge dimensional matrices involved, the actual implementation of the methods is of paramount importance. The next section shows that the RQ factorization can be used to efficiently implement the methods.

In the description of the identification methods, it was assumed that the block sizes used to construct the different data matrices were the same. It is also possible to use different sizes. For example, the two block method in Subsection 3.3.1 can also be formulated by making the following identifications:

$$
\begin{aligned}
Y &\leftarrow \mathcal{Y}_{k+j|j}, \\
Z_X &\leftarrow U_{j-1|0}, \\
Z_{P \odot X} &\leftarrow P_{k+j-1|j} \odot U_{j-1|0}, \\
U &\leftarrow \begin{bmatrix} U_{k+j-1|j} \\ U_{k+j} \end{bmatrix}.
\end{aligned}
$$

So, the first set of data matrices starting at 0 has a block size j, and the second set of data matrices starting at j has a block size k. From Subsection 3.2.1 we know that the matrix Γ_k must have full rank, therefore $k \geq n - 1$. From equation (3.14) on page 30 it follows that if X_j is approximated by $U_{j-1|0}$, we should have at least $j \geq n$. Combining these two requirements shows that we can choose $k = j - 1$ with $j \geq n$. This results in a slight dimension reduction compared to the case where the block sizes are equal (that is, $k = j$).

3.4 Efficient Implementation Using the RQ Factorization

The RQ factorization provides an efficient way to compute an estimate of the matrix Φ that is needed to determine the state sequence (see Subsection 3.2.2). It can be used to implement equation (3.36) on page 35 as well as equation (3.40) on page 38.

First, the estimate given by equation (3.36) is discussed. It is immediately clear that equation (3.36) is the solution to a least-squares problem by solving the normal equations. It is well-known that a numerically more reliable and computationally less demanding solution can be computed using an RQ factorization (Golub and Van Loan 1996). Given the RQ factorization

$$
\begin{bmatrix} Z_X \\ \begin{bmatrix} Z_{P\odot X} \\ U \\ Y \end{bmatrix} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix},
\tag{3.56}
$$

equation (3.36) becomes

$$
\widehat{\Phi} = \begin{bmatrix} R_{31} & R_{32} \end{bmatrix} \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix}^{-1},
\tag{3.57}
$$

where it is assumed that N is sufficiently large, such that Ω has full row rank. Equation (3.57) shows that the least-squares estimate of Φ_X is given by

$$
\widehat{\Phi}_X = R_{31} R_{11}^{-1} - R_{32} R_{22}^{-1} R_{21} R_{11}^{-1}.
\tag{3.58}
$$

The identification method based on this RQ factorization is called *Identification Procedure I*, and is summarized on page 53. The two-block method in Subsection 3.3.1 can be implemented using Identification Procedure I.

Now consider the estimate involving instrumental variables, given by equation (3.40) on page 38. Only a particular choice of instrumental variables will be treated here. This choice corresponds to the one made in the three-block methods discussed in the previous section. It is assumed that a part of Z_X can be used as an instrumental variable. The matrix Z_X is split into two parts;

$$
Z_X = \begin{bmatrix} Z_{X,R} \\ Z_{X,IV} \end{bmatrix},
$$

where only the part $Z_{X,IV}$ can be used as an instrumental variable. It is also assumed that $Z_{P\odot X}$ and U can be used as instrumental variables. Note that it is also possible to split $Z_{P\odot X}$ into two parts, but that is not considered here; the presented scheme can be easily be extended to deal with this case. The instrumental variable matrix is taken equal to

$$
Z_{IV} = \begin{bmatrix} Z_{A,IV} \\ Z_{X,IV} \\ Z_{P\odot X} \\ U \end{bmatrix}.
$$

With these choices the RQ factorization

$$
\begin{bmatrix} Z_{A,IV} \\ Z_{X,R} \\ Z_{X,IV} \\ \begin{bmatrix} Z_{P\odot X} \\ U \\ Y \end{bmatrix} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} & 0 \\ R_{51} & R_{52} & R_{53} & R_{54} & R_{55} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{bmatrix}
\tag{3.59}
$$

Identification Procedure I

Assumptions:

1. $\displaystyle\lim_{N\to\infty}\frac{1}{N}Z_V\begin{bmatrix}Z_X^T & Z_{P\odot X}^T & U^T\end{bmatrix}=0$

2. The matrix $\begin{bmatrix}Z_X\\Z_{P\odot X}\\U\end{bmatrix}$ has full row rank.

Procedure:

1. Compute RQ factorization:

$$\begin{bmatrix}Z_X\\ \begin{bmatrix}Z_{P\odot X}\\U\\Y\end{bmatrix}\end{bmatrix}=\begin{bmatrix}R_{11}&0&0\\R_{21}&R_{22}&0\\R_{31}&R_{32}&R_{33}\end{bmatrix}\begin{bmatrix}Q_1\\Q_2\\Q_3\end{bmatrix}$$

2. Estimate Φ_X: $\widehat{\Phi}_X=R_{31}R_{11}^{-1}-R_{32}R_{22}^{-1}R_{21}R_{11}^{-1}$

3. Compute singular value decomposition:

$$\widehat{\Phi}_X Z_X=\begin{bmatrix}U_1&U_2\end{bmatrix}\begin{bmatrix}\Sigma_1&0\\0&\Sigma_2\end{bmatrix}\begin{bmatrix}V_1^T\\V_2^T\end{bmatrix}$$

4. Approximate state sequence: $\widehat{X}_k=\Sigma_1^{1/2}V_1$

5. Estimate system matrices:

$$[\widehat{A},\widehat{B}]=\widehat{X}_{k+1}\begin{bmatrix}\widehat{X}_{k|k}^T&U_{k|k}^T\end{bmatrix}\left(\begin{bmatrix}\widehat{X}_{k|k}\\U_{k|k}\end{bmatrix}\begin{bmatrix}\widehat{X}_{k|k}^T&U_{k|k}^T\end{bmatrix}\right)^{-1}$$

$$[\widehat{C},\widehat{D}]=Y_k\begin{bmatrix}\widehat{X}_k^T&U_k^T\end{bmatrix}\left(\begin{bmatrix}\widehat{X}_k\\U_k\end{bmatrix}\begin{bmatrix}\widehat{X}_k^T&U_k^T\end{bmatrix}\right)^{-1}$$

can be used to turn equation (3.40) into

$$\widehat{\Phi}=L_5 L_{1,3,4}^T L_{1,3,4}L_{2,3,4}^T\left(L_{2,3,4}L_{1,3,4}^T L_{1,3,4}L_{2,3,4}^T\right)^{-1},$$

where it is assumed that N is sufficiently large, such that ΩZ_{IV}^T has full row rank, and where

$$L_{1,3,4}=\begin{bmatrix}R_{11}&0&0&0\\R_{31}&R_{32}&R_{33}&0\\R_{41}&R_{42}&R_{43}&R_{44}\end{bmatrix},$$

$$L_{2,3,4} = \begin{bmatrix} R_{21} & R_{22} & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix},$$

$$L_5 = \begin{bmatrix} R_{51} & R_{52} & R_{53} & R_{54} \end{bmatrix},$$

and use has been made of the relations

$$YZ_{IV}^T = L_5 L_{1,3,4}^T,$$
$$\Omega Z_{IV}^T = L_{2,3,4} L_{1,3,4}^T.$$

Note that computing $\widehat{\Phi}$ in this way still involves a considerable amount of computations.

It is possible to derive a computationally more attractive scheme, by taking a slightly different instrumental variable matrix:

$$Z_{IV} = \begin{bmatrix} Q_1 \\ Q_3 \\ Q_4 \end{bmatrix}. \tag{3.60}$$

With this choice of Z_{IV}, we get

$$YZ_{IV}^T = \begin{bmatrix} R_{51} & R_{53} & R_{54} \end{bmatrix},$$

$$\Omega Z_{IV}^T = \begin{bmatrix} R_{21} & 0 & 0 \\ R_{31} & R_{33} & 0 \\ R_{41} & R_{43} & R_{44} \end{bmatrix},$$

and thus

$$\widehat{\Phi} = \begin{bmatrix} R_{51} & R_{53} & R_{54} \end{bmatrix} \begin{bmatrix} R_{21} & 0 & 0 \\ R_{31} & R_{33} & 0 \\ R_{41} & R_{43} & R_{44} \end{bmatrix}^{\dagger}, \tag{3.61}$$

which requires less computations. It follows that the matrix ΩZ_{IV}^T, with Z_{IV} given by (3.60) must have full row rank. This can again be regarded as a persistency of excitation condition. In general, it cannot be concluded that

$$\Omega \begin{bmatrix} Q_1^T & Q_3^T & Q_4^T \end{bmatrix}$$

has full row rank if

$$\Omega \begin{bmatrix} Z_{A,IV}^T & Z_{X,IV}^T & Z_{P\odot X}^T & U^T \end{bmatrix}$$

has full row rank. However, since it is not clear yet what conditions have to be imposed on the system and the signals u_k and p_k to fulfill the latter rank condition, it might as well be replaced by the first rank condition.

The identification method based on the RQ factorization (3.59) with the choice of instrumental variables given by (3.60) is called *Identification Procedure II*, and is summarized on page 55. This procedure can be used to implement the three-block method in Subsection 3.3.2, the two-stage three-block method in Subsection 3.3.3, and the two-stage three-block method with reduced dimensions in Subsection 3.3.4.

Identification Procedure II

Assumptions:

1. $\displaystyle \lim_{N\to\infty} \frac{1}{N} Z_V \begin{bmatrix} Z_{A,IV}^T & Z_{X,IV}^T & Z_{P\odot X}^T & U^T \end{bmatrix} = 0$

2. The matrices $\begin{bmatrix} Z_{A,IV} \\ Z_{X,IV} \\ Z_{P\odot X} \\ U \end{bmatrix}$ and $\begin{bmatrix} R_{21} & 0 & 0 \\ R_{31} & R_{33} & 0 \\ R_{41} & R_{43} & R_{44} \end{bmatrix}$ obtained from the RQ
   factorization given below, have full row rank.

Procedure:

1. Compute RQ factorization:

$$\begin{bmatrix} Z_{A,IV} \\ Z_{X,R} \\ Z_{X,IV} \\ \begin{bmatrix} Z_{P\odot X} \\ U \\ Y \end{bmatrix} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} & 0 \\ R_{51} & R_{52} & R_{53} & R_{54} & R_{55} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{bmatrix}$$

2. Estimate Φ_X:

$$\widehat{\Phi} = \begin{bmatrix} R_{51} & R_{53} & R_{54} \end{bmatrix} \begin{bmatrix} R_{21} & 0 & 0 \\ R_{31} & R_{33} & 0 \\ R_{41} & R_{43} & R_{44} \end{bmatrix}^{\dagger},$$

$$\widehat{\Phi}_X = \widehat{\Phi}(1:q,:),$$

   where q is the number of rows in the matrix Z_X.

3. Compute singular value decomposition:

$$\widehat{\Phi}_X Z_X = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

4. Approximate state sequence: $\widehat{X}_k = \Sigma_1^{1/2} V_1$

5. Estimate system matrices:

$$[\widehat{A}, \widehat{B}] = \widehat{X}_{k+1} \begin{bmatrix} \widehat{X}_{k|k}^T & U_{k|k}^T \end{bmatrix} \left(\begin{bmatrix} \widehat{X}_{k|k} \\ U_{k|k} \end{bmatrix} \begin{bmatrix} \widehat{X}_{k|k}^T & U_{k|k}^T \end{bmatrix} \right)^{-1}$$

$$[\widehat{C}, \widehat{D}] = Y_k \begin{bmatrix} \widehat{X}_k^T & U_k^T \end{bmatrix} \left(\begin{bmatrix} \widehat{X}_k \\ U_k \end{bmatrix} \begin{bmatrix} \widehat{X}_k^T & U_k^T \end{bmatrix} \right)^{-1}$$

The identification methods discussed in the previous section use different data matrices in the RQ factorizations (3.56) and (3.59). Consequently, they differ in accuracy, and in the total number of rows in the RQ factorization. For each method, the number of rows in the RQ factorization is given below. In these expressions d_k is defined as $d_k := (s+1)^{k+1} - 1$.

- Two-block method (Subsection 3.3.1, Identification Procedure I):

$$(2d_k + d_k^2)\frac{m(s+1)}{s} + m + (k+1)\ell.$$

- Three-block method (Subsection 3.3.2, Identification Procedure II):

$$(3d_k + d_k^2)\frac{m(s+1)}{s} + d_k(m+\ell) + 3m + 2\ell + (k+1)\ell.$$

- Two-stage three-block method (Subsection 3.3.3, Identification Procedure II):

$$(3d_k + 2d_k^2)\frac{m(s+1)}{s} + 2d_k(m+\ell) + 3m + 2\ell + (k+1)\ell.$$

- Two-stage three-block method with reduced dimensions (Subsection 3.3.4, Identification Procedure II):

$$(3d_k + 2d_k^2)\frac{m(s+1)}{s} + 3m + 2\ell + (k+1)\ell.$$

Tables 3.1 and 3.2 illustrate these expressions. Clearly, the number of rows in the RQ factorization grows very rapidly with the block size k. For certain block sizes, the dimensions of the data matrices become so large that alternative methods are desirable; Chapter 4 describes such a method.

3.5   Identification for Innovation-Type Noise Models

This section deals with LPV subspace identification for a special type of noise model. It is assumed that the process noise w_k in the LPV system (3.1)–(3.2) depends on the measurement noise v_k in the following way:

$$w_k = K \begin{bmatrix} v_k \\ p_k \otimes v_k \end{bmatrix},$$

with the matrix K given by

$$K := \begin{bmatrix} K_0 & K_1 & \cdots & K_s \end{bmatrix},$$

where $K_i \in \mathbb{R}^{n \times \ell}$ for $i = 0, 1, 2, \ldots, s$. Hence, the system description becomes

$$
\begin{aligned}
x_{k+1} &= A \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + B \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix} + K \begin{bmatrix} v_k \\ p_k \otimes v_k \end{bmatrix}, & (3.62) \\
y_k &= C x_k + D u_k + v_k. & (3.63)
\end{aligned}
$$

Table 3.1: Total number of rows in the RQ factorization for the identification methods discussed in Subsections 3.3.1, 3.3.2, 3.3.3, and 3.3.4, as a function of the block size k in the case that $s = m = \ell = 1$.

Identification method	Procedure	$k = 1$	$k = 2$	$k = 3$	$k = 4$
Two-block	I	33	130	515	2 052
Three-block	II	49	162	579	2 180
Two-stage three-block	II	74	276	1 062	4 168
Two-stage three-block red.	II	62	248	1 002	4 044

Table 3.2: Total number of rows in the RQ factorization for the identification methods discussed in Subsections 3.3.1, 3.3.2, 3.3.3, and 3.3.4, as a function of the block size k in the case that $s = m = \ell = 2$.

Identification method	Procedure	$k = 1$	$k = 2$	$k = 3$	$k = 4$
Two-block	I	246	2 192	19 690	177 156
Three-block	II	310	2 382	20 258	178 858
Two-stage three-block	II	536	4 518	39 784	355 526
Two-stage three-block red.	II	472	4 310	39 144	353 590

The assumption on the process noise may seem strange at first sight, but if the state equation is written as a weighted combination of linear models,

$$x_{k+1} = A_0 x_k + B_0 u_k + K_0 v_k + \sum_{i=1}^{s} [p_k]_i \left(A_i x_k + B_i u_k + K_i v_k \right),$$

it becomes clear that we have an innovation-type of model for each linear model described by A_i, B_i, C, and D.

An interesting feature of this model structure is that it is easy to formulate a one-step-ahead predictor:

$$x_{k+1} = \overline{A} \left[\begin{array}{c} x_k \\ p_k \otimes x_k \end{array} \right] + \overline{B} \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right] + K \left[\begin{array}{c} y_k \\ p_k \otimes y_k \end{array} \right], \qquad (3.64)$$

$$y_k = C x_k + D u_k + v_k, \qquad (3.65)$$

with

$$\overline{A} := [\, \overline{A_0} \quad \overline{A_1} \quad \overline{A_2} \quad \cdots \quad \overline{A_s} \,],$$
$$\overline{B} := [\, \overline{B_0} \quad \overline{B_1} \quad \overline{B_2} \quad \cdots \quad \overline{B_s} \,],$$

and

$$\overline{A_i} := A_i - K_i C,$$
$$\overline{B_i} := B_i - K_i D,$$

for $i = 0, 1, 2, \ldots, s$.

The remainder of this section is organized as follows: First, the data equations in Section 3.1 are slightly modified to take the additional assumption on the process noise into account. The resulting data equations have again the general structure discussed in Section 3.2, and thus several identification methods can be developed by applying the ideas of that section. To illustrate this, a two-block identification method will be discussed in detail.

3.5.1   Data Equations

To derive the data equations for the system (3.62)–(3.63), some additional data matrices are defined:

$$Y_{j|j} \quad := \quad \begin{bmatrix} Y_j \\ P_j \odot Y_j \end{bmatrix} \in \mathbb{R}^{(s+1)\ell \times N}, \tag{3.66}$$

$$Y_{k+j|j} \quad := \quad \begin{bmatrix} Y_{k+j} \\ P_{k+j} \odot Y_{k+j} \\ Y_{k+j-1|j} \\ P_{k+j} \odot Y_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((s+1)^{k+1}-1)\ell(s+1)/s \times N}, \tag{3.67}$$

$$V_{j|j} \quad := \quad \begin{bmatrix} V_j \\ P_j \odot V_j \end{bmatrix} \in \mathbb{R}^{(s+1)\ell \times N}, \tag{3.68}$$

$$V_{k+j|j} \quad := \quad \begin{bmatrix} V_{k+j} \\ P_{k+j} \odot V_{k+j} \\ V_{k+j-1|j} \\ P_{k+j} \odot V_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((s+1)^{k+1}-1)\ell(s+1)/s \times N}. \tag{3.69}$$

From Lemma 3.3 on page 30 and Lemma 3.4 on page 31 it is straightforward to derive the data equations for the system (3.62)–(3.63). These equations are presented in the following two corollaries:

Corollary 3.2 *Given the matrices (3.3)–(3.9), (3.18), (3.68), and (3.69). For the LPV system (3.62)–(3.63) it holds that*

$$X_{k+j} = \Delta_k^x X_{k+j-1|j} + \Delta_k^u U_{k+j-1|j} + \Delta_k^v V_{k+j-1|j}, \tag{3.70}$$

where

$$\begin{aligned} \Delta_1^v \quad &:= \quad [K_0, K_1, \dots K_s], \\ \Delta_k^v \quad &:= \quad [\Delta_1^v, A_0 \Delta_{k-1}^v, A_1 \Delta_{k-1}^v, \dots, A_s \Delta_{k-1}^v], \end{aligned}$$

and Δ_k^x and Δ_k^u are as in Lemma 3.3.

Corollary 3.3 *Given the matrices (3.3)–(3.9), (3.15)–(3.18), (3.68), and (3.69). For the LPV system (3.62)–(3.63) it holds that*

$$\mathcal{Y}_{k+j|j} = H_k^x X_{k+j-1|j} + H_k^u U_{k+j-1|j} + H_k^v V_{k+j-1|j} + G_k^u U_{k+j} + G_k^v V_{k+j}, \tag{3.71}$$

where

$$
\begin{aligned}
H_1^v &:= \begin{bmatrix} CK_0 & CK_1 & \cdots & CK_s \\ I_\ell & 0 & \cdots & 0 \end{bmatrix}, \\
H_k^v &:= \begin{bmatrix} CK_0 & CK_1 & \cdots & CK_s & CA_0\Delta_{k-1}^v & CA_1\Delta_{k-1}^v & \cdots & CA_s\Delta_{k-1}^e \\ G_{k-1}^v & 0 & \cdots & 0 & H_{k-1}^v & 0 & \cdots & 0 \end{bmatrix}, \\
G_k^v &:= \begin{bmatrix} I_\ell \\ 0 \end{bmatrix},
\end{aligned}
$$

and H_k^x, H_k^u, and G_k^u are as in Lemma 3.4 and Δ_k^v as in Corollary 3.2.

Using the one-step-ahead predictor (3.64)–(3.65), an additional data equation can be derived.

Lemma 3.6 *Given the matrices (3.3)–(3.9), (3.15), (3.66), and (3.67). For the LPV system (3.64)–(3.65) it holds that*

$$
X_{k+j} = \overline{\Delta}_k^x X_{k+j-1|j} + \overline{\Delta}_k^u U_{k+j-1|j} + \overline{\Delta}_k^y Y_{k+j-1|j}, \tag{3.72}
$$

where

$$
\begin{aligned}
\overline{\Delta}_1^x &:= [\overline{A}_0, \overline{A}_1, \ldots, \overline{A}_s], \\
\overline{\Delta}_k^x &:= [\overline{A}_0\overline{\Delta}_{k-1}^x, \overline{A}_1\overline{\Delta}_{k-1}^x, \ldots, \overline{A}_s\overline{\Delta}_{k-1}^x], \\
\overline{\Delta}_1^u &:= [\overline{B}_0, \overline{B}_1, \ldots, \overline{B}_s], \\
\overline{\Delta}_k^u &:= [\overline{\Delta}_1^u, \overline{A}_0\overline{\Delta}_{k-1}^u, \overline{A}_1\overline{\Delta}_{k-1}^u, \ldots, \overline{A}_s\overline{\Delta}_{k-1}^u], \\
\overline{\Delta}_1^y &:= [K_0, K_1, \ldots K_s], \\
\overline{\Delta}_k^y &:= [\overline{\Delta}_1^y, \overline{A}_0\overline{\Delta}_{k-1}^y, \overline{A}_1\overline{\Delta}_{k-1}^y, \ldots, \overline{A}_s\overline{\Delta}_{k-1}^y].
\end{aligned}
$$

Proof: The proof follows from an induction argument, similar to the one used in the proof of Lemma 3.3 on page 30. □

A combination of Lemma 3.2 with Corollary 3.3 yields the following data equation:

$$
\begin{aligned}
\mathcal{Y}_{k+j|j} = {}& \Gamma_k X_j + \widetilde{H}_k^x(P_{k+j-1|j} \odot X_j) + H_k^u U_{k+j-1|j} + G_k^u U_{k+j} \\
& + H_k^v V_{k+j-1|j} + G_k^v V_{k+j}.
\end{aligned} \tag{3.73}
$$

This equation is similar to equation (3.24) on page 33 and has the same basic structure as the generalized data equation (3.25) on page 33. Therefore, it is possible to apply the ideas discussed in Section 3.2 to derive identification algorithms, like the two-block method, three-block method, and two-stage three-block method. Only the two-block method will be derived here.

3.5.2 A Two-Block Method

The two-block identification method for the system (3.62)–(3.63) derived below is very similar to the two-block method described in Subsection 3.3.1. It has previously been described by Verdult and Verhaegen (2000c) and can be viewed as

the LPV extension of the bilinear subspace identification method described by Favoreel (1999).

With the result of Lemma 3.6, the state sequence can be approximated as

$$X_{k+j} \approx \overline{\Delta}_k^u U_{k+j-1|j} + \overline{\Delta}_k^y Y_{k+j-1|j}.$$

So the term $\overline{\Delta}_k^x X_{k+j-1|j}$ is neglected. Similar to Lemma 3.5 on page 40 it is possible do derive an upper bound for the Frobenius norm of this term.

Corollary 3.4 *Consider the LPV system (3.62)–(3.63) with a fixed sequence p_k. If the state equation*

$$x_{k+1} = [(A_0 - K_0 C), (A_1 - K_1 C), \dots, (A_s - K_s C)] \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix}$$

is uniformly exponentially stable, and for any j

$$\sup_{k \geq j} \|u_k\|_2 < \infty, \qquad \sup_{k \geq j} \|v_k\|_2 < \infty,$$

then given a finite constant β, there exist a finite constant $\gamma > 0$, and a constant λ, $0 \leq \lambda < 1$, such that for every j and every $k \geq j$

$$\frac{1}{\sqrt{N}} \|\overline{\Delta}_k^x X_{k+j-1|j}\|_F \leq \gamma \lambda^k,$$

for all k, j, such that $k \geq j$, and such that

$$\sup_k \|p_k\|_2 \leq \beta,$$

where $\overline{\Delta}_k^x$ is as in Lemma 3.6 and $\| \cdot \|_F$ denotes the Frobenius norm.

The two-block method makes use of two sets of data matrices: the first set starts at $j = k$, and is used to identify the system; the second set starts at $j = 0$, and is used to approximate the state sequence. For the first data set ($j = k$) the data equation is

$$\mathcal{Y}_{2k|k} = \Gamma_k X_k + \widetilde{H}_k^x (P_{2k-1|k} \odot X_k) + H_k^u U_{2k-1|k} + H_k^v V_{2k-1|k} + G_k^u U_{2k} + G_k^v V_{2k}, \tag{3.74}$$

in which the state sequence X_k is approximated by

$$X_k \approx \overline{\Delta}_k^u U_{k-1|0} + \overline{\Delta}_k^y Y_{k-1|0}.$$

The next theorem summarizes the main result on which the two-block method is based.

Theorem 3.5 *Given the bounded-input, bounded-output stable system (3.62)–(3.63) with bounded sequences p_k, u_k, and v_k; the data equations of Lemmas 3.2 and 3.6, and of Corollary 3.3; and the generalized data equation (3.32):*

$$Y = \Phi_X Z_X + \Phi_{P \odot X} Z_{P \odot X} + \Phi_U U + \Phi_V Z_V + \Xi_X + \Xi_{P \odot X},$$

with the following substitutions:

$$
\begin{aligned}
Y &\leftarrow \mathcal{Y}_{2k|k}, \\
Z_X &\leftarrow \begin{bmatrix} U_{k-1|0} \\ Y_{k-1|0} \end{bmatrix}, \\
Z_{P\odot X} &\leftarrow P_{2k-1|k} \odot \begin{bmatrix} U_{k-1|0} \\ Y_{k-1|0} \end{bmatrix}, \\
U &\leftarrow \begin{bmatrix} U_{2k-1|k} \\ U_{2k} \end{bmatrix}, \\
Z_V &\leftarrow \begin{bmatrix} V_{2k-1|k} \\ V_{2k} \end{bmatrix}.
\end{aligned}
$$

Then

$$
\lim_{N\to\infty} \frac{1}{N} Z_V \Omega^T = 0 \quad with \quad \Omega := \begin{bmatrix} Z_X \\ Z_{P\odot X} \\ U \end{bmatrix}
$$

holds with probability one, and the approximation errors are

$$
\begin{aligned}
\Xi_X &\leftarrow \Gamma_k \overline{\Delta}_k^x X_{k-1|0}, \\
\Xi_{P\odot X} &\leftarrow (\widetilde{H}_k^x \otimes \overline{\Delta}_k^x)(P_{2k-1|k} \odot X_{k-1|0}),
\end{aligned}
$$

with Γ_k given by (3.22) and $\widetilde{H}_k^x$ given by (3.23).
   If the matrix Ω has full row rank, the matrix

$$
\Phi := \begin{bmatrix} \Phi_X & \Phi_{P\odot X} & \Phi_U \end{bmatrix}
$$

can be estimated as

$$
\widehat{\Phi} = Y\Omega^T \left(\Omega\Omega^T\right)^{-1},
$$

and $(\widehat{\Phi} - \Phi)\Omega$ does not depend on the noise term Z_V with probability one for $N \to \infty$.

From this theorem it follows that the two-block method can be implemented with Identification Procedure I, described on page 53.

The system matrices can be determined from the state estimate $\widehat{X}_k$ as described in Subsection 3.2.3 and in Identification Procedure I. The matrix K needed to construct the one-step-ahead predictor (3.64)–(3.65), can be estimated as follows:

$$
\begin{aligned}
[\widehat{C}, \widehat{D}] &= Y_k \begin{bmatrix} \widehat{X}_k^T & U_k^T \end{bmatrix} \left(\begin{bmatrix} \widehat{X}_k \\ U_k \end{bmatrix} \begin{bmatrix} \widehat{X}_k^T & U_k^T \end{bmatrix} \right)^{-1}, \\
\widehat{V}_k &= Y_k - \widehat{C}\widehat{X}_k - \widehat{D}U_k, \\
[\widehat{A}, \widehat{B}, \widehat{K}] &= \widehat{X}_{k+1} \begin{bmatrix} \widehat{X}_{k|k}^T & U_{k|k}^T & \widehat{V}_{k|k}^T \end{bmatrix} \left(\begin{bmatrix} \widehat{X}_{k|k} \\ U_{k|k} \\ \widehat{V}_{k|k} \end{bmatrix} \begin{bmatrix} \widehat{X}_{k|k}^T & U_{k|k}^T & \widehat{V}_{k|k}^T \end{bmatrix} \right)^{-1},
\end{aligned}
$$

with

$$
\widehat{V}_{k|k} := \begin{bmatrix} \widehat{V}_k \\ P_k \odot \widehat{V}_k \end{bmatrix}.
$$

Table 3.3: Total number of rows in the RQ factorization for the two-block identification method described in Subsection 3.5.2 as a function of the block size k.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$s = m = \ell = 1$	45	186	755	3 044
$s = m = \ell = 2$	318	2 894	26 170	235 962

3.5.3 Discussion

The system (3.62)–(3.63) is a special case of (3.1)–(3.2). Therefore, the identification methods described in Section 3.3 can be used. However, by taking the special noise model into account, specialized methods can be derived such as the two-block method discussed above. These methods originate from approximating the state based on the expression

$$
\begin{aligned}
X_{k+j} \;=\; & M(Y_{k+j} - DU_{k+j} - V_{k+j}) \\
& + (I - MC)\Big(\overline{\Delta}_k^u U_{k+j-1|j} + \overline{\Delta}_k^y Y_{k+j-1|j} + \overline{\Delta}_k^x X_{k+j-1|j}\Big).
\end{aligned} \tag{3.75}
$$

The two-block method described above corresponds to the case $M = 0$. For $M \neq 0$, a three-block method similar to the one described in Subsection 3.3.2 can be derived. Furthermore, a two-stage state approximation could be applied, along the same lines as in Subsection 3.3.3. These derivations are rather straightforward, so they are not presented here. Compared to the two-block method described above, a drawback of these more complex methods is that the dimensions of the matrices used in the RQ factorization become much larger.

The specialized methods that take into account the special noise model have two main advantages compared to the general methods described in Section 3.3.

First, they allow to estimate the matrix K needed to construct the one-step-ahead predictor. The methods in Section 3.3 cannot be used for this, because the state sequence in (3.62)–(3.63) is different from the one in (3.1)–(3.2).

Second, for these specialized methods, the approximation of the state based on (3.26) on page 34 does not contain the term V_X. The approximation error therefore only depends on the term $\overline{\Delta}_k^x X_{k+j-1|j}$. Of course, it would be interesting to compare the approximation error of this two-block method with the approximation errors of the methods discussed in Section 3.3, but again for the same reasons as discussed in Subsection 3.3.5, it is difficult to make any precise statements.

A drawback of the specialized methods is the increased number of rows used in the RQ factorizations (3.56) and (3.59). For the two-block method in Subsection 3.5.2, the total number of rows in the RQ factorization equals

$$
(2d_k + d_k^2)\frac{m(s+1)}{s} + d_k^2 \frac{\ell}{s} + m + (k+1)\ell.
$$

Table 3.3 illustrates this expression, and can be used for comparison with the other identification methods in Tables 3.1 and 3.2 on page 57.

For the special case that $K_i = 0$, $i = 1, 2, \ldots, s$ in equation (3.62), the specialized methods like the two-block method in Subsection 3.5.2 can be simplified. In this case, $Y_{k+j|j}$ and $V_{k+j|j}$ can be replaced by

$$
\begin{aligned}
Y_{j|j} &:= Y_j \in \mathbb{R}^{\ell \times N}, \\
Y_{k+j|j} &:= \begin{bmatrix} Y_{k+j} \\ Y_{k+j-1|j} \\ P_{k+j} \odot Y_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((s+1)^{k+1}-1)\ell/s \times N}, \\
V_{j|j} &:= V_j \in \mathbb{R}^{\ell \times N}, \\
V_{k+j|j} &:= \begin{bmatrix} V_{k+j} \\ V_{k+j-1|j} \\ P_{k+j} \odot V_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((s+1)^{k+1}-1)\ell/s \times N}.
\end{aligned}
$$

A two-block method for this case has previously been described by Verdult and Verhaegen (2000b).

A system that is driven only by noise ($u_k = 0$) is another special case that fits into the framework described above. Such a system is given by

$$
x_{k+1} = A \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + K \begin{bmatrix} v_k \\ p_k \otimes v_k \end{bmatrix}, \tag{3.76}
$$

$$
y_k = C x_k + v_k. \tag{3.77}
$$

A one-step-ahead predictor for this system can be determined by a slight modification of the two-block identification method in Subsection 3.5.2; the terms in equation (3.32) on page 35 are then given by

$$
\begin{aligned}
Y &\leftarrow \mathcal{Y}_{2k|k}, \\
Z_X &\leftarrow Y_{k-1|0}, \\
Z_{P \odot X} &\leftarrow P_{2k-1|k} \odot Y_{k-1|0}, \\
Z_V &\leftarrow \begin{bmatrix} V_{2k-1|k} \\ V_{2k} \end{bmatrix},
\end{aligned}
$$

and U is identically zero, and has to be omitted in the RQ factorization.

3.6   Special Cases

The previous section described LPV subspace identification for a special type of noise model. In this section, some other special cases of the LPV system are considered: full state measurement and time-invariant B matrix.

Bilinear systems discussed in Part II of this thesis are also a special case of the LPV system; for a bilinear system the time-varying parameter p_k equals the input u_k. Bilinear systems are not discussed below, but separately in Part II of this thesis, because they form an important class of nonlinear systems, which has been widely studied in the literature (an overview of the literature is given in Subsection 8.3).

3.6.1 Full State Measurement

Lovera, Verhaegen, and Chou (1998) have developed an identification method for LPV systems where the outputs are noisy measurements of the state, that is,

$$
\begin{aligned}
x_{k+1} &= A \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + B \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix} + w_k, \\
y_k &= x_k + v_k.
\end{aligned}
$$

Note that if the noise v_k and the noise w_k are identically zero ($y_k = x_k$) the system matrices of (3.1)–(3.2) can be estimated consistently from a least-squares problem (see also Subsection 3.2.3):

$$
\begin{aligned}
[\widehat{A}, \widehat{B}] &= Y_{k+1} \begin{bmatrix} Y_{k|k}^T & U_{k|k}^T \end{bmatrix} \left(\begin{bmatrix} Y_{k|k} \\ U_{k|k} \end{bmatrix} \begin{bmatrix} Y_{k|k}^T & U_{k|k}^T \end{bmatrix} \right)^{-1}, \\
[\widehat{C}, \widehat{D}] &= Y_k \begin{bmatrix} Y_k^T & U_k^T \end{bmatrix} \left(\begin{bmatrix} Y_k \\ U_k \end{bmatrix} \begin{bmatrix} Y_k^T & U_k^T \end{bmatrix} \right)^{-1}.
\end{aligned}
$$

If the noise v_k is a zero-mean white noise and the noise w_k is identically zero, consistent estimates of the system matrices can be obtained by using appropriate instrumental variables. Lovera, Verhaegen, and Chou (1998) proposed to use the matrices

$$
\begin{bmatrix} Y_{k-1|k-1} \\ U_{k|k} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} Y_{k-1} \\ U_k \end{bmatrix}
$$

as instrumental variable matrices for the state and output equation, respectively. This leads to the following estimates:

$$
\begin{aligned}
[\widehat{A}, \widehat{B}] &= Y_{k+1} \begin{bmatrix} Y_{k-1|k-1}^T & U_{k|k}^T \end{bmatrix} \left(\begin{bmatrix} Y_{k|k} \\ U_{k|k} \end{bmatrix} \begin{bmatrix} Y_{k-1|k-1}^T & U_{k|k}^T \end{bmatrix} \right)^{-1}, \\
[\widehat{C}, \widehat{D}] &= Y_k \begin{bmatrix} Y_{k-1}^T & U_k^T \end{bmatrix} \left(\begin{bmatrix} Y_k \\ U_k \end{bmatrix} \begin{bmatrix} Y_{k-1}^T & U_k^T \end{bmatrix} \right)^{-1}.
\end{aligned}
$$

Since it is sufficient to estimate the system matrices up to a similarity transformation, the above scheme can also be used if the C matrix is a square and invertible, but unknown matrix. The C matrix can then be regarded as a similarity transformation of the state. Furthermore, if the number of outputs is larger than the number of states ($\ell > n$), and if C has full column rank, an SVD of Y_k can be used to determine the state sequence up to a similarity transformation. Let the SVD of Y_k be given by

$$
Y_k = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},
$$

with $\Sigma_1 \in \mathbb{R}^n$ containing the dominant singular values, then the state sequence can be estimated as $\hat{X}_k = \Sigma_1^{1/2} V_1^T$, and the system matrices can be estimated as

$$
\begin{aligned}
[\widehat{A}, \widehat{B}] &= \widehat{X}_{k+1} \begin{bmatrix} \widehat{X}_{k-1|k-1}^T & U_{k|k}^T \end{bmatrix} \left(\begin{bmatrix} \widehat{X}_{k|k} \\ U_{k|k} \end{bmatrix} \begin{bmatrix} \widehat{X}_{k-1|k-1}^T & U_{k|k}^T \end{bmatrix} \right)^{-1}, \\
[\widehat{C}, \widehat{D}] &= Y_k \begin{bmatrix} \widehat{X}_{k-1}^T & U_k^T \end{bmatrix} \left(\begin{bmatrix} \widehat{X}_k \\ U_k \end{bmatrix} \begin{bmatrix} \widehat{X}_{k-1}^T & U_k^T \end{bmatrix} \right)^{-1}.
\end{aligned}
$$

3.6.2 Time-Invariant B Matrix

If the 'B matrix' of the system (3.1)–(3.2) is time-invariant, that is, $B_i = 0$ for $i = 1, 2, \ldots, s$, then the identification methods discussed in Sections 3.3 and 3.5 can be simplified. It is easy to verify that the data matrices $U_{k+j|j}$ can be replaced by

$$
\begin{aligned}
U_{j|j} &:= U_j \in \mathbb{R}^{m \times N}, \\
U_{k+j|j} &:= \begin{bmatrix} U_{k+j} \\ U_{k+j-1|j} \\ P_{k+j} \odot U_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((s+1)^{k+1}-1)m/s \times N}.
\end{aligned}
$$

This obviously results in a dimension reduction.

Chapter 4

Dimension Reduction in Subspace Identification

A major problem with subspace methods for LPV systems is the enormous dimension of the data matrices involved. The number of rows that needs to be included in the data matrices grows exponentially with the order of the system. Even for relatively low-order systems with only a few inputs and outputs, the amount of memory required to store these data matrices exceeds the limits of what is currently available on the average desktop computer. This limits the applicability of the methods to high-order systems with several inputs and outputs.

Equally important, to implement the methods with an RQ factorization, the number of columns of the data matrices involved needs to be larger than their total number of rows. Since the number of columns equals the number of data samples, an enormous amount of data is required to fulfill this condition.

The same explosion of dimensions occurs for bilinear subspace identification methods, which are discussed in Part II. Favoreel (1999) was the first to recognize this problem. He pointed out some possible directions to reduce this curse of dimensionality, but he did not try out or evaluate any of them. Hence, their viability still needs to be proven.

This chapter presents a novel approach to reduce the dimensions of the data matrices. The discussion focuses on the two-block methods. These methods are implemented with an RQ factorization as in *Identification Procedure I* on page 53. The idea is to select a subset of rows from the data matrices Z_X, $Z_{P \odot X}$, and U appearing in this RQ factorization, and to discard the remaining rows. Only the selected rows are used to compute an RQ factorization. This later RQ factorization, which has less rows than the original one involving the complete data matrices, will be called the *reduced RQ factorization*. The rows that are selected for the reduced RQ factorization are the ones that contribute most to the matrix Y.

The selection of the rows is basically a subset-selection problem (Lawson and Hanson 1995, p. 194–196). Roughly speaking, subset selection can be performed by either using a singular value decomposition or an RQ (or QR) factorization (Golub and Van Loan 1996, p. 590–595). A method based on the RQ factorization is

preferred, because it allows to compute the reduced RQ factorization step by step during the selection process. This means that at the end of the selection process, the reduced RQ factorization has already been computed, which obviously results in considerable computational savings.

Reducing the number of rows in this way will of course be at the expense of introducing an error, so that the identified LPV model is only an approximate model. Simulations presented in Chapter 6 show that quite often the number of rows can be reduced considerably, without deteriorating the quality of the model too much. However, if the performance of the obtained LPV model is not satisfactory, the model can be improved using a nonlinear optimization-based identification procedure, like the one presented in Chapter 5.

The selection procedure that is described below can also be applied to the three-block methods that are based on *Identification Procedure II* on page 55. In addition to reducing the number of rows in the matrices Z_X, $Z_{P \odot X}$, and U, it might also be desirable to reduce the number of rows in the matrix $Z_{A,IV}$ appearing in the RQ factorization of *Identification Procedure II*.

This chapter is organized as follows: In Section 4.1 the difference between the full RQ factorization and the reduced RQ factorization is studied. Section 4.2 presents a procedure to select a subset of the most dominant rows from the data matrices that does not require the complete formation of these matrices, but processes them row by row. This procedure also minimizes the error introduced by the reduced RQ factorization for a fixed number of selected rows. Section 4.3 discusses some issues related to the implementation of the selection procedure.

4.1   Error Analysis of the Reduced RQ Factorization

To reduce the dimension in LPV subspace identification, the RQ factorization is only carried out for a subset of the rows of the matrices Z_X, $Z_{P \odot X}$, and U. Below, this reduced RQ factorization is compared with the full factorization. For this purpose, let Ω^s contain the rows of Z_X, $Z_{P \odot X}$, and U that are selected for the reduced RQ factorization, and let the matrix Ω^d contain the remaining rows that are discarded. Now, with appropriate definitions of Φ^s and Φ^d, the generalized data equation (3.32) on page 35 can be rewritten as

$$Y = \begin{bmatrix} \Phi^s & \Phi^d \end{bmatrix} \begin{bmatrix} \Omega^s \\ \Omega^d \end{bmatrix} + \Phi_V Z_V + \Xi_X + \Xi_{P \odot X}.$$

For simplicity, it is assumed that the matrix

$$\Omega = \begin{bmatrix} \Omega^s \\ \Omega^d \end{bmatrix}$$

has full rank for every N that is larger than the total numbers of rows in Ω. Note that to use the subspace identification methods outlined in the previous chapter for finite N, this assumption is needed anyway.

Let the RQ factorization

$$
\begin{bmatrix} \Omega^s \\ \Omega^d \\ Y \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} \tag{4.1}
$$

be equivalent to the RQ factorization used in *Identification Procedure I* on page 53. From the description of this identification procedure it follows that

$$
\Upsilon := \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^{-1} \begin{bmatrix} \Omega^s \\ \Omega^d \end{bmatrix} \tag{4.2}
$$

is the quantity that needs to be computed. Discarding the rows contained in Ω^d is equivalent to computing a reduced RQ factorization given by

$$
\begin{bmatrix} \Omega^s \\ Y \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ \overline{L}_{31} & \overline{L}_{32} \end{bmatrix} \begin{bmatrix} S_1 \\ \overline{S}_3 \end{bmatrix}, \tag{4.3}
$$

which can be used to approximate Υ by

$$
\widehat{\Upsilon} := L_{31} L_{11}^{-1} \Omega^s. \tag{4.4}
$$

The error due to this approximation can easily be computed and is given in the following lemma:

Lemma 4.1 *Given the full RQ factorization (4.1) and the reduced RQ factorization (4.3). The matrices Υ and $\widehat{\Upsilon}$ given by (4.2) and (4.4), respectively, satisfy*

$$
\|\Upsilon - \widehat{\Upsilon}\|_F = \|L_{32}\|_F,
$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

   Proof: Combining the RQ factorization (4.1) with equations (4.2) and (4.4) it is easy to see that

$$
\begin{aligned}
\Upsilon &= L_{31}S_1 + L_{32}S_2, \\
\widehat{\Upsilon} &= L_{31}S_1.
\end{aligned}
$$

Hence,

$$
\|\Upsilon - \widehat{\Upsilon}\|_F = \|L_{32}S_2\|_F = \|L_{32}\|_F.
$$

This completes the proof.   □

4.2   Selection of Dominant Rows

The selection of the rows from the matrices Z_X, $Z_{P \odot X}$, and U is carried out by computing an RQ factorization row by row. The selection of a row is based on its contribution to the matrix Y in a least-squares sense.

Let the matrix Z_X^r contain a subset of the rows of the matrix Z_X, let $Z_{P \odot X}^r$ contain a subset of $Z_{P \odot X}$, and U^r a subset of U, such that

$$\Omega^s := \left[\begin{array}{c} Z_X^r \\ Z_{P \odot X}^r \\ U^r \end{array} \right].$$

Let n_z^r denote the number of rows of the matrix Z_X^r. Provided that at least n rows are selected from Z_X ($n_z^r \geq n$), an approximation of the state can be computed, from the singular value decomposition of the matrix

$$L_{31} L_{11}^{-1} Z_X^r,$$

similar to Steps 2, 3, and 4 in *Identification Procedure I* on page 53 (see also Subsection 3.2.2). This state estimate can be used to compute the system matrices in a similar way as described in Subsection 3.2.3.

Since the matrix $L_{31} L_{11}^{-1} Z_X^r$ needs to be computed, the matrix Z_X^r needs to be available. Therefore, the selection of the rows from Z_X is separated from the selection of the rows from $Z_{P \odot X}$ and U. Below, it is explained how to select the rows from Z_X. A similar procedure can be used to select rows from $Z_{P \odot X}$ and U. Note that there is no need to separate the selection of rows from $Z_{P \odot X}$ and the selection of rows from U.

The first row that is selected from Z_X is the one that has the largest contribution to the matrix Y. To determine this row, the set of least-squares problems

$$\min_{\psi_i} ||Y - \psi_i Z_X(i, :)||_F^2, \quad i = 1, 2, \ldots, n_z \tag{4.5}$$

is solved and the corresponding residuals are computed. The row with the smallest residual has the largest contribution to the matrix Y and should hence be selected. To select the next row of Z_X, the effect of the previously selected row needs to be taken into account. This means that the residuals have to be recomputed after which the row that yields the smallest residual can be selected. More specifically, the procedure given below (Verhaegen 1987) can be used to select the n_z^r most dominant rows from Z_X.

Take $Z^1 := Z_X$ and $Y^1 := Y$. For i ranging from 1 to n_z^r, perform the following steps:

1. Select the row of Z^i that has the smallest residual when solving

$$\min_{\Psi_q^i} \left| \left| Y^i - \Psi_q^i Z^i(q, :) \right| \right|_F^2, \quad q = i, i + 1, \ldots, n_z.$$

2. Interchange the ith row of Z^i with the row selected in the previous step, by the permutation matrix Π^i.

3. Compute an orthogonal projection H^i such that

$$\Pi^i \left[\begin{array}{cc} T^i & 0 \\ \overline{T}^i & Z^i \end{array} \right] H^i = \left[\begin{array}{cc} T^{i+1} & 0 \\ \overline{T}^{i+1} & Z^{i+1} \end{array} \right],$$

where $T^i \in \mathbb{R}^{(i-1) \times (i-1)}$, and $\overline{T}^i \in \mathbb{R}^{(n_z - i + 1) \times (i - 1)}$.
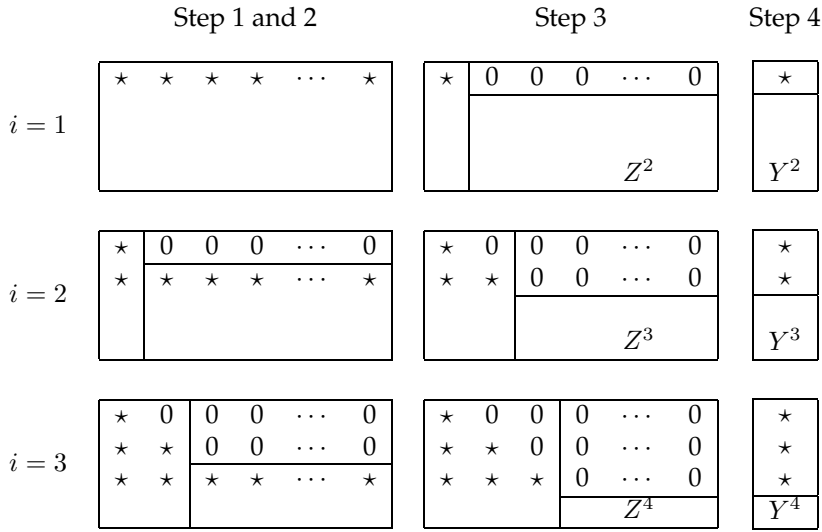
Figure 4.1: Graphical representation of the row selection procedure described in Section 4.2. The row selected in Step 1 is brought to the top of the matrix Z^i by Step 2. Step 3 then computes a rotation such that all the entries of this row become zero, except for the first i entries. The 'new' matrix Z^{i+1} is one column and one row smaller than Z^i. In Step 4 the rotation is also applied to Y^i, and this yields Y^{i+1}.

4. Apply H^i to Y^i:

$$\left[\ \overline{Y}^i\quad Y^i\ \right] H^i = \left[\ \overline{Y}^{i+1}\quad Y^{i+1}\ \right],$$

where $\overline{Y}^i \in \mathbb{R}^{(i-1)\times(i-1)}$.

Figure 4.1 gives a graphical representation of this selection algorithm. It shows that in every step the matrix Z^i is reduced by one column and one row, and that the matrix T^i becomes a lower triangular matrix.

The number of rows that are selected to form Z_X^r determines the approximation error. If the amount of computer memory available is limited, it makes sense to fix this number of rows beforehand. Alternatively, the norm of the residual can be used to decide when to stop the selection process.

Let the matrix $\overline{Z}_X$ be the matrix that is obtained by applying all the computed permutation matrices Π^i, $i = 1, 2, \ldots, n_z^r$ to the matrix Z_X, that is, $\overline{Z}_X := \Pi^{n_z^r} \cdot \Pi^{n_z^r-1}\cdots\Pi^1 \cdot Z_X$. Since the top n_z^r rows of $\overline{Z}_X$ are now the ones that have the largest contribution to Y_j, these are the rows to construct the matrix Z_X^r from: $Z_X^r = \overline{Z}_X(1: n_z^r, :)$. An important motivation for using the RQ factorization in selecting the rows, is that such a selection procedure yields already the first part of the reduced RQ factorization (4.3):

$$Z_X^r = \left[\ L_{11}(1: n_z^r, 1: n_z^r)\quad 0\ \right] S_1(1: n_z^r, :).$$

The next step is to reduce the number of rows in the matrix $Z_{P\odot X}$. For this a similar procedure as for the selection of the rows of Z_X is used. By continuing the application of orthogonal rotations, the remaining part of the matrix L_{11} is computed. Applying all the rotations used in the selection procedures for Z_X and $Z_{P\odot X}$ to the matrix Y yields the matrix L_{31} of (4.3). Since the matrix $\overline{L}_{32}$ is not needed in the subsequent computations, it is not necessary to complete the reduced RQ factorization.

The selection procedure described above minimizes the Frobenius norm of the approximation error

$$\|\Upsilon - \widehat{\Upsilon}\|_F,$$

for a fixed number of rows n_z^r. To see this, note that

$$\min_{\Phi^s} ||Y - \Phi^s \Omega^s||_F^2 = \min_{\Phi^s} ||[L_{31}, \overline{L}_{32}] - \Phi^s[L_{11}, 0]||_F^2 = ||\overline{L}_{32}||_F^2.$$

Comparing equation (4.1) with (4.3) yields

$$\overline{L}_{32}\overline{S}_3 = L_{32}S_2 + L_{33}S_3.$$

Since $S_2 S_3^T = 0$, it follows that

$$||\overline{L}_{32}||_F^2 = ||L_{32}||_F^2 + ||L_{33}||_F^2.$$

Equation (4.1) shows that the selection of the rows, that is, which rows go into Ω^s, and which go into Ω^d, does not influence the matrix L_{33}. Therefore, minimizing $||\overline{L}_{32}||_F^2$ is equivalent to minimizing $||L_{32}||_F^2$. With the result of Lemma 4.1, the conclusion is that the described selection procedure minimizes the Frobenius norm of the approximation error.

4.3 Implementation Issues

Because of the huge dimensions of the data matrices, an efficient implementation of the selection algorithm described above is of paramount importance. Storing the complete matrices Z_X and $Z_{P\odot X}$ into the memory is often not feasible. Therefore, a procedure has been developed that constructs only one row at a time. After evaluating the associated value for the residual, the row is replaced by the next one. In this way, only the residuals need to be stored. If a row is encountered that has the lowest residual thus far, this row is stored into the memory. The Householder rotations associated with the selected rows are also stored. For an efficient implementation, the Householder matrices H^j should not be explicitly formed. It is much more efficient to use and store the associated Householder vectors (Golub and Van Loan 1996, p. 211).

An algorithm for selecting n_z^r rows from $Z_X \in \mathbb{R}^{n_z \times N}$ is given below. It constructs the matrix $R \in \mathbb{R}^{n_z^r \times n_z^r}$, which is a part of the reduced RQ factorization (4.3): $R = L_{11}(1\colon n_z^r, 1\colon n_z^r)$.

Define the index set $\mathcal{I}^1 := \{1, 2, \ldots, n_z\}$.

for each $j \in \{1, 2, \ldots, n_z^r\}$

1. Select the row from $\mathcal{I}^j$ that has the smallest residual:
   for each $k \in \mathcal{I}^j$

   (a) Construct the kth row of Z_X: $Z_k^1 = Z_X(k, 1\colon N)$.

   (b) Apply all previously computed Householder rotations $Z_k^{i+1} = Z_k^i H^i$ for $i \in \{1, 2, \ldots, j-1\}$; this yields Z_k^j.

   (c) Compute the residuals

   $$r_k^j = \min_{\phi_k^j} ||Y^j(:\,, j\colon N) - \Psi_k^j Z_k^j(:\,, j\colon N)||_F^2.$$

   Note that this can be done efficiently by first computing a Householder rotation Q_k^j such that $Z_k^j(:\,, j\colon N)Q_k^j = [P_k^j,\ 0_{1\times(N-j)}]$ with $P_k^j \in \mathbb{R}$ and then evaluating

   $$r_k^j = ||Y^j(:\,, j\colon N)Q_k^j(:\,, 2\colon N - j + 1)||_F^2.$$

   (d) If $r_k^j \leq r_i^j$ for all $i \in \mathcal{I}^j$ then row k has the smallest residual thus far and we select this row, that is, we make the jth row of Z_X^r equal to Z_k^j and we take $\mathcal{I}^{j+1} = \mathcal{I}^j \setminus \{k\}$ as the new index set.

   end

2. Compute a Householder rotation H^j such that $Z_X^r(j, :\,)H^j = [R^j,\ 0_{1\times(N-j)}]$ with $R^j \in \mathbb{R}^{1\times j}$.

3. Make the jth row of the R matrix equal to $[R^j,\ 0_{1\times(n_z^r-j)}]$.

4. Apply the rotation H^j to Y^j: $Y^{j+1} = Y^j H^j$.

end

Observe that a further improvement can be obtained by updating the residuals at each step using information from the previous step. After computing the residuals in the first step, the residuals for the following steps can be obtained with the update rule

$$r_k^j = r^{j-1} - ||Y^j(:\,, j\colon N)Q_k^j(:\,, 1)||_2^2,$$

where r^{j-1} is the residual of the row selected in the previous step.

With this implementation, it is possible to handle systems of moderate dimensions, because the data matrices do not need to be stored. However, the computation time increases drastically. This is mainly due to the need to construct one row at a time. Depending on its place in the data matrix, the construction of a row involves several times N multiplications. It is possible to reduce this computational burden a bit by stepping through the rows in a special order. For example, if we are processing the rows of $U_{j-1|0}$, we could after processing a row in the first part of the matrix (that is, $U_{j-2|0}$), store the row, and use it to first compute the rows

in the third part of the matrix (that is, $P_{j-1} \odot U_{j-2|0}$), which can be obtained by element-wise multiplication with a row from P_{j-1}. This requires only N multiplications, in contrast with several times N when the row needs to be constructed from scratch.

The computation time can also be reduced by shrinking the set of rows to be processed at each step. Some rows that have the largest residuals are discarded; the set of rows is shrunk at every step with a constant factor, such that in the final step the number of rows left to choose from, equals the total number of rows that have been selected. Although this scheme does not give any guarantee on not losing important rows, it turned out to work quite well for the examples presented in Chapter 6.

Finally, the initial set of rows can be reduced by discarding the ones that are formed by more than h multiplications. This in fact boils down to discarding the terms with order higher than h. It can be argued that because of the limited number of samples available, higher-order terms cannot be estimated accurately anyway.

Chapter 5

Identification Based on Nonlinear Optimization

S ystem identification is usually carried out by minimizing the difference be-
tween the output of a preselected model and the measured output of the sys-
tem to be identified. The model mismatch is quantified by a cost function that
decreases when the output predicted by the model approaches the measured out-
put. A commonly used cost function is the sum of squared errors between the
predicted output and the measured output for a fixed, large number of samples.
When the model is completely described by a set of parameters, system identi-
fication boils down to finding the values of the parameters that minimize this
cost function. This approach is commonly known as the *prediction-error* approach
(Ljung 1999). Note that subspace identification methods —both the LPV meth-
ods discussed in the previous two chapters and the more commonly known LTI
versions— are a notable exception to this approach.

This chapter describes a prediction-error type of identification method for LPV
state-space systems. A drawback of most prediction-error methods, including the
one described here, is that the model output depends nonlinearly on the parame-
ters that describe the model. As a result, the minimization of the prediction-error
cost function becomes a nonlinear, nonconvex optimization problem that can have
several local minima. It is common practice to use a gradient search algorithm to
solve this optimization problem. In this case the initial starting point determines
in which local minimum the algorithm will end up. Therefore, it is of paramount
importance to have a good initial estimate of the LPV system available when using
prediction-error methods.

Subspace identification methods for LPV systems do not suffer from the prob-
lem of multiple local minima. However, as discussed in the previous two chap-
ters, these subspace identification methods only yield approximate models. In
Chapter 3 it became clear that it is necessary to approximate the state sequence,
and in this approximation an error is introduced. Furthermore, it turned out that
dimensions of the data matrices can be too large to be handled, and therefore in

Chapter 4 a method was discussed that only uses the most dominant rows in the data matrices, at the expense of introducing an additional approximation error.

Approximate models estimated by subspace identification can be improved with prediction-error methods. The subspace method provides a good initial estimate of the model to start up the prediction-error method. This makes it less likely that the prediction-error method ends up in a local minimum that corresponds to a bad model performance. This combination of two methods is a powerful tool to tackle LPV identification problems. Such a combination has also been suggested for the identification of LTI systems by Ljung (1999, p. 211) and by Haverkamp (2001, Chapter 4).

The prediction-error methods described in this chapter deal with LPV systems of the form (2.1)–(2.2). They can also be used for systems with a time-invariant output equation, as in (3.1)–(3.2). With respect to disturbances acting on the system, two cases will be considered.

Section 5.1 focuses on LPV systems with white measurement noise at the output; the corresponding system description is

$$x_{k+1} = A \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + B \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix}, \tag{5.1}$$

$$y_k = C \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + D \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix} + v_k, \tag{5.2}$$

where is v_k a zero-mean white-noise sequence that is independent of the input u_k. Given a finite number of measurements of the input u_k, the output y_k, and the parameter p_k, and initial estimates of the matrices A, B, C, and D, the goal is to improve these estimates with a prediction-error method. Since in this case the prediction-error equals the difference between the measured output and the simulated output of the model, this problem is called the *output-error problem*.

Section 5.2 deals with measurement noise and process noise of the following form:

$$x_{k+1} = A \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + B \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix} + K \begin{bmatrix} v_k \\ p_k \otimes v_k \end{bmatrix}, \tag{5.3}$$

$$y_k = C \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} + D \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix} + v_k, \tag{5.4}$$

where is v_k a zero-mean white-noise sequence that is independent of the input u_k. The goal is to improve the initial estimates of the matrices A, B, C, D, and K with respect to a prediction-error criterion, using only a finite number of measurements of the input u_k, the output y_k, and the parameter p_k. Note that this formulation makes it possible to construct a one-step-ahead predictor for the output y_k. To distinguish this problem from the previous one, it will be referred to as the *prediction-error problem*.

These two identification problems are solved by using a local parameterization of the LPV state-space system, and by performing a gradient search in the resulting local parameter space. This method is similar to the one presented by Lee and Poolla (1999), who discussed the output-error identification problem for LFT descriptions. As mentioned in Section 2.1, the LPV model (2.1)–(2.2) is a special case of the LFT description. This chapter presents an alternative way to derive

the local gradient search method, without using the fact that the system (2.1)–(2.2) is a special case of the LFT description. It discusses both the output-error and prediction-error problem (based on Verdult and Verhaegen 2001c), while Lee and Poolla only described the output-error problem.

5.1 Projected Gradient Search for Output-Error Optimization

First, the output-error identification problem is discussed. A model of the system (5.1)–(5.2) is represented by a set of model parameters θ (which are not to be confused with the time-varying parameter p_k). The matrices A, B, C, and D are fully parameterized, which means that the model parameters are given by $\theta = \text{vec}(\Theta)$ with Θ given by

$$\Theta := \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

Given the input u_k, the parameter p_k, and the output y_k of the real system (5.1)–(5.2), the goal is to determine the matrices A, B, C, and D of the model

$$\widehat{x}_{k+1}(\theta) = A \begin{bmatrix} \widehat{x}_k(\theta) \\ p_k \otimes \widehat{x}_k(\theta) \end{bmatrix} + B \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix}, \tag{5.5}$$

$$\widehat{y}_k(\theta) = C \begin{bmatrix} \widehat{x}_k(\theta) \\ p_k \otimes \widehat{x}_k(\theta) \end{bmatrix} + D \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix}, \tag{5.6}$$

such that its output $\widehat{y}_k(\theta)$ approximates the output y_k of the real system sufficiently accurately. To achieve this goal, the output error is minimized with respect to the parameters θ. The output-error cost function is given by

$$J_N(\theta) := \sum_{k=1}^{N} ||y_k - \widehat{y}_k(\theta)||_2^2 = E_N(\theta)^T E_N(\theta), \tag{5.7}$$

where

$$E_N(\theta)^T = \left[\left(y_1 - \hat{y}_1(\theta)\right)^T \quad \left(y_2 - \hat{y}_2(\theta)\right)^T \quad \cdots \quad \left(y_N - \hat{y}_N(\theta)\right)^T \right],$$

and N is the total number of measurements available.

Minimization of (5.7) is a nonlinear, nonconvex optimization problem because of the nonlinear dependence of $\widehat{y}_k(\theta)$ on θ. Different algorithms exist to numerically search for a solution to this optimization problem. One popular choice is a gradient search method known as the Levenberg-Marquardt algorithm (Moré 1978). This is an iterative procedure that updates the system parameters θ with the following rule:

$$\theta^{(i+1)} = \theta^{(i)} - \mu^{(i)} \left(\lambda^{(i)} I + \Psi_N^T(\theta^{(i)}) \Psi_N(\theta^{(i)})\right)^{-1} \Psi_N^T(\theta^{(i)}) E_N(\theta^{(i)}), \tag{5.8}$$

where $\mu^{(i)}$ is the step size, $\lambda^{(i)}$ is the Levenberg-Marquardt regularization parameter and

$$\Psi_N(\theta) := \frac{\partial E_N(\theta)}{\partial \theta^T}$$

is the gradient of $E_N(\theta)$. The step size $\mu^{(i)}$ can be determined by performing a line search, for example, with mixed quadratic and polynomial interpolation as in the *Matlab Optimization Toolbox* (Coleman, Branch, and Grace 1999).

It is important to realize that the minimization of $J_N(\theta)$ does not have a unique solution, because there exist different systems Θ that have the same input-output behavior. If the state of the system (5.5)–(5.6) is transformed into $\overline{x}_k = T^{-1}\widehat{x}_k$ with $T \in \mathbb{R}^{n \times n}$ nonsingular, the system description becomes

$$\overline{\Theta} = \left[\begin{array}{c|c} \overline{A} & \overline{B} \\ \hline \overline{C} & \overline{D} \end{array} \right] = \left[\begin{array}{cc} T^{-1} & 0 \\ 0 & I_\ell \end{array} \right] \left[\begin{array}{cc} A & B \\ C & D \end{array} \right] \left[\begin{array}{cc} T_{s+1} & 0 \\ 0 & I_{m(s+1)} \end{array} \right],$$

where T_s denotes a block diagonal matrix with s copies of the matrix T along the diagonal. Obviously, this system has the same input-output behavior as the system Θ (see also Subsection 2.3.1). This means that $J_N(\theta) = J_N(\overline{\theta})$ with $\overline{\theta} = \mathrm{vec}(\overline{\Theta})$. The nonuniqueness of the optimal θ leads to problems with the updating rule (5.8). For certain directions in the parameter space, the cost function does not change.

The nonuniqueness of the optimal θ is a direct consequence of the full parameterization. By choosing an appropriate parameterization, the optimal θ can be made unique. For LTI systems, a number of such parameterizations exists. However, they are not always numerically reliable (McKelvey 1995, Chapters 2 and 3) and can become quite involved for systems with multiple inputs and outputs. Currently, for LPV systems such parameterizations are not known.

The nonuniqueness problem can be tackled in other ways. Lee and Poolla (1999) have proposed a solution in which at each iteration the directions that do not change the value of the cost function are identified and are projected out of the search direction used in the update rule. This solution can be interpreted as letting the algorithm decide the parameterization that is used at each iteration. The active parameters are determined from the data; this leads to numerical advantages. A similar method has been derived by McKelvey and Helmersson (1997) for identification of LTI state-space systems. The solution of Lee and Poolla can also be applied to the model structure (5.1)–(5.2), as explained below.

5.1.1 Local Parameterization

The nonuniqueness of the state-space system (5.5)–(5.6) can be characterized by the *similarity map* $S_\Theta : T, \det(T) \neq 0 \rightarrow \mathbb{R}^{(n+\ell) \times (n+m)(s+1)}$, which is defined as (Lee and Poolla 1999):

$$S_\Theta(T) := \left[\begin{array}{cc} T^{-1} & 0 \\ 0 & I_\ell \end{array} \right] \left[\begin{array}{cc} A & B \\ C & D \end{array} \right] \left[\begin{array}{cc} T_{s+1} & 0 \\ 0 & I_{m(s+1)} \end{array} \right]. \tag{5.9}$$

Taking a particular fixed model Θ, all models with the same input-output behavior that can be created by performing a linear state transformation are given by

$$\mathcal{I}_\Theta := \left\{ \overline{\Theta} \mid \overline{\Theta} = S_\Theta(T), \ \det(T) \neq 0 \right\}.$$
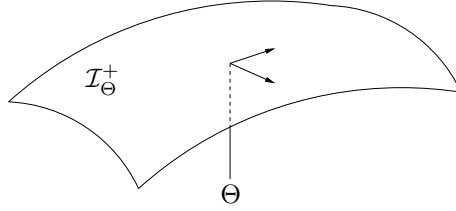
Figure 5.1: Vectors that span the tangent space of the manifold $\mathcal{I}_\Theta^+$ at the point Θ.

This set is called the *indistinguishable set at* Θ, because it contains all the models that cannot be distinguished from Θ by looking at their input-output behavior.

According to Lee and Poolla (1999), if the similarity map S_Θ is locally one-to-one at I_n, that is, if $S_\Theta(T) = \Theta$ implies $T = I_n$, the connected component of $\mathcal{I}_\Theta$,

$$\mathcal{I}_\Theta^+ := \left\{ \overline{\Theta} \,|\, \overline{\Theta} = S_\Theta(T),\, \det(T) > 0 \right\},$$

is a differentiable manifold. The tangent space of this manifold at the point Θ contains the directions along which a change in the parameters θ does not influence the value of the cost function $J_N(\theta)$ (see Figure 5.1). To improve the optimization algorithm, the update rule (5.8) should be changed such that the parameters θ are not changed along these directions. For this reason, the tangent space of the manifold $\mathcal{I}_\Theta^+$ needs to be determined.

In the sequel it is assumed that the similarity map is locally one-to-one. Lee and Poolla (1999) derived some, possibly conservative, conditions on the system matrices that guarantee this. In the following lemma their result is reformulated for the LPV model structure (5.5)–(5.6):

Lemma 5.1 *Let the observability matrix for* A_i *and* C_i *be given by*

$$\mathcal{O}_{n,i} := \begin{bmatrix} C_i \\ C_i A_i \\ C_i A_i^2 \\ \vdots \\ C_i A_i^{n-1} \end{bmatrix}.$$

If the matrix $\mathcal{O}_{n,i}$ *has full rank for some* $i \in \{1, 2, \ldots, s\}$*, then* $S_\Theta(T) = \Theta$ *implies* $T = I_n$*.*

Proof: If $\mathcal{O}_{n,i}$ has full rank, there exists a right pseudo-inverse $\mathcal{O}_{n,i}^+$ such that $\mathcal{O}_{n,i}^+ \mathcal{O}_{n,i} = I_n$. With the assumption $S_\Theta(T) = \Theta$ it follows that $\mathcal{O}_{n,i} T = \mathcal{O}_{n,i}$ and thus $T = \mathcal{O}_{n,i}^+ \mathcal{O}_{n,i} = I_n$. □

Note that a similar result can be proven by replacing the observability condition for A_i and C_i with a controllability condition for A_i and B_i.

To determine the tangent space of the manifold $\mathcal{I}_\Theta^+$, the similarity map $S_\Theta(T)$

is linearized:

$$
\begin{aligned}
S_\Theta(T) &\approx S_\Theta(I_n + T) - S_\Theta(I_n) \\
&= \begin{bmatrix} (T + I_n)^{-1} & 0 \\ 0 & I_\ell \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} T_{s+1} + I_{n(s+1)} & 0 \\ 0 & I_{m(s+1)} \end{bmatrix} \\
&\quad - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \\
&\approx \begin{bmatrix} A \\ C \end{bmatrix} \begin{bmatrix} T_s & 0_{n \times m(s+1)} \end{bmatrix} - \begin{bmatrix} T \\ 0_{\ell \times n} \end{bmatrix} \begin{bmatrix} A & B \end{bmatrix}, \quad (5.10)
\end{aligned}
$$

where the last expression is obtained by approximating $(I_n + T)^{-1}$ by $I_n - T$, and neglecting all second-order terms. By defining

$$
\Pi_i := \begin{bmatrix} 0_{n \times (i-1)n} & I_n & 0_{n \times (s+1-i)n} \end{bmatrix},
$$

it is possible to write (5.10) as

$$
S_\Theta(T) \approx \sum_{i=1}^{s+1} \begin{bmatrix} A \\ C \end{bmatrix} \Pi_i^T T \begin{bmatrix} \Pi_i & 0_{n \times m(s+1)} \end{bmatrix} - \begin{bmatrix} I_n \\ 0_{\ell \times n} \end{bmatrix} T \begin{bmatrix} A & B \end{bmatrix}.
$$

Using the property $\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B)$, this becomes

$$
\text{vec}\Big(S_\Theta(T)\Big) \approx M_\Theta \text{vec}(T), \quad (5.11)
$$

where

$$
M_\Theta := \sum_{i=1}^{s+1} \begin{bmatrix} \Pi_i^T \\ 0_{m(s+1) \times n} \end{bmatrix} \otimes \begin{bmatrix} A\Pi_i^T \\ C\Pi_i^T \end{bmatrix} - \begin{bmatrix} A^T \\ B^T \end{bmatrix} \otimes \begin{bmatrix} I_n \\ 0_{\ell \times n} \end{bmatrix}.
$$

The next lemma shows that in general the matrix M_Θ has full rank.

Lemma 5.2 *The similarity map S_Θ, given by (5.9), is locally one-to-one, that is, $S_\Theta(I_n + T) = \Theta$ implies $T = 0$, if and only if the matrix M_Θ has full rank.*

Proof: The equality $S_\Theta(I_n + T) = \Theta$ is equivalent to

$$
\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} T_{s+1} + I_{n(s+1)} & 0 \\ 0 & I_{m(s+1)} \end{bmatrix} - \begin{bmatrix} T + I_n & 0 \\ 0 & I_\ell \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = 0,
$$

which can be rewritten as

$$
\begin{bmatrix} A \\ C \end{bmatrix} \begin{bmatrix} T_{s+1} & 0_{n \times m(s+1)} \end{bmatrix} - \begin{bmatrix} T \\ 0_{\ell \times n} \end{bmatrix} \begin{bmatrix} A & B \end{bmatrix} = 0.
$$

With the definition of M_Θ, this is in turn equivalent to

$$
M_\Theta \text{vec}(T) = 0. \quad (5.12)
$$

It follows that S_Θ is locally one-to-one if and only if the matrix M_Θ has full rank, since in this case the only solution to (5.12) is $T = 0$.   □

Equation (5.11) shows that the tangent space of the manifold $\mathcal{I}_\Theta^+$ at the point Θ equals the column space of the matrix M_Θ. Since the matrix M_Θ has more rows than columns, it has a nonempty left null space. The left null space of the matrix M_Θ, which is the orthogonal complement of the tangent space of the manifold $\mathcal{I}_\Theta^+$ at the point Θ, contains the directions in which the parameters should be changed to obtain a change in the cost function $J_N(\theta)$. The left null space of the matrix M_Θ can be determined using a singular value decomposition. Given the singular value decomposition

$$M_\Theta = \begin{bmatrix} U_1(\theta) & U_2(\theta) \end{bmatrix} \begin{bmatrix} \Sigma(\theta) \\ 0 \end{bmatrix} V^T(\theta), \tag{5.13}$$

every parameter vector θ can be decomposed into two parts: $\theta = U_1(\theta)U_1^T(\theta)\theta + U_2(\theta)U_2^T(\theta)\theta$, where the first part corresponds to the directions that do not influence the cost function, and the second part to the directions that change the value of the cost function. Based upon this observation, the update rule for the parameters (5.8) is changed such that the update is restricted to the directions that change the cost function:

$$\begin{aligned} \theta^{(i+1)} &= \theta^{(i)} - \mu^{(i)} U_2(\theta^{(i)}) \\ &\times \left(\lambda^{(i)} I + U_2^T(\theta^{(i)}) \Psi_N^T(\theta^{(i)}) \Psi_N(\theta^{(i)}) U_2(\theta^{(i)}) \right)^{-1} \\ &\times U_2^T(\theta^{(i)}) \Psi_N^T(\theta^{(i)}) E_N(\theta^{(i)}). \end{aligned} \tag{5.14}$$

Note that the matrix U_2 depends on the current parameters $\theta^{(i)}$, therefore the singular value decomposition (5.13) must be computed at each iteration. This is however not so demanding compared with the computation of the gradient $\Psi_N(\theta)$.

Until now the discussion has focused on minimizing (5.7) over all entries of the matrix Θ. However, it might happen that some prior information is available for some of the entries of Θ, such that these entries do not have to be optimized. For example, say we know that the output equation of the system to be identified does not equal (2.2), but is simply

$$y_k = C_0 x_k + D_0 u_k + v_k, \tag{5.15}$$

and thus the entries of Θ that correspond to $C_1, C_2, \ldots, C_s$ and $D_1, D_2, \ldots, D_s$ equal zero. Then, these entries can be deleted from the parameter vector θ; the parameter vector becomes $\theta = \Xi \operatorname{vec}(\Theta)$, where the matrix Ξ selects only the rows from $\operatorname{vec}(\Theta)$ that do not correspond to $C_1, C_2, \ldots, C_s$ or $D_1, D_2, \ldots, D_s$. To use the local parameterization in this case, not the the matrix M_Θ must be used to compute the singular value decomposition (5.13), but the matrix ΞM_Θ. With this modification the update rule (5.14) can be used again.

5.1.2 Computing the Iterative Update of the Parameters

The previous subsection described how to compute $U_2(\theta)$ for the update rule (5.14). Two other quantities that need to be computed are $E_N(\theta)$ and $\Psi_N(\theta)$. The

vector $E_N(\theta)$ follows by simulating the system (5.5)–(5.6) that corresponds to θ, since this yields $\widehat{y}_k(\theta)$, for $k = 1, 2, \ldots, N$. Note that this simulation also yields the state $\widehat{x}_k(\theta)$. This state can be used to efficiently compute $\Psi_N(\theta)$ by simulating an LPV system with the same structure as (5.5)–(5.6) for every parameter in θ. To see this, observe that $\Psi_N(\theta)$ follows from computing $\partial\widehat{y}_k/\partial\theta_i$ for every parameter θ_i, and every time instant $k = 1, 2, \ldots, N$. Taking the derivative of $\widehat{y}_k$ with respect to θ_i yields

$$\frac{\partial\widehat{y}_k(\theta)}{\partial\theta_i} = C \left[\begin{array}{c} \mathcal{X}_k^i \\ p_k \otimes \mathcal{X}_k^i \end{array} \right] + \frac{\partial C}{\partial\theta_i} \left[\begin{array}{c} \widehat{x}_k(\theta) \\ p_k \otimes \widehat{x}_k(\theta) \end{array} \right] + \frac{\partial D}{\partial\theta_i} \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right], \qquad (5.16)$$

where

$$\mathcal{X}_k^i := \frac{\partial\widehat{x}_k(\theta)}{\partial\theta_i}.$$

The key point is that the states $\mathcal{X}_k^i$ can be computed by simulating the equation

$$\mathcal{X}_{k+1}^i = A \left[\begin{array}{c} \mathcal{X}_k^i \\ p_k \otimes \mathcal{X}_k^i \end{array} \right] + \frac{\partial A}{\partial\theta_i} \left[\begin{array}{c} \widehat{x}_k(\theta) \\ p_k \otimes \widehat{x}_k(\theta) \end{array} \right] + \frac{\partial B}{\partial\theta_i} \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right], \qquad (5.17)$$

where $\widehat{x}_k(\theta)$ plays the role of an input signal. For successful identification the gradient $\Psi_N(\theta)$ must be bounded, and hence $\mathcal{X}_{k+1}^i$ must be bounded for every i. The following lemma shows that this is the case if the system (5.5)–(5.6) is uniformly exponentially stable:

Lemma 5.3 *If the system (5.5)–(5.6) is uniformly exponentially stable, and for any j*

$$\sup_{k \geq j} \|p_k\|_2 < \infty, \quad \sup_{k \geq j} \|u_k\|_2 < \infty,$$

then for every i, $\mathcal{X}_{k+1}^i$ given by (5.17) satisfies

$$\sup_{k \geq j} \|\mathcal{X}_k^i\|_2 < \infty.$$

Proof: First, note that (5.17) is uniformly asymptotically stable if and only if (5.5) is uniformly asymptotically stable. Application of Corollary 2.2 on page 21 to the system (5.5)–(5.6) shows that

$$\sup_{k \geq j} \|x_k\|_2 < \infty.$$

Since x_k is bounded, application of Corollary 2.2 to (5.17) shows that $\mathcal{X}_k^i$ is also bounded. □

During optimization the signal p_k is bounded and fixed, therefore the stability of (5.5)–(5.6) depends on the matrix A. Since the optimization method aims at minimizing the output error, it is very unlikely that the parameters describing the matrix A are modified towards instability of the system. Therefore, the gradient will not become unbounded during optimization.

5.1.3   Obtaining an Initial Starting Point

Since the output error cost function (5.7) can have several local minima, it is of paramount importance to obtain a good initial starting point θ. For this the LPV subspace identification methods in Chapters 3 and 4 can be used. If the system to be identified has a linear time-invariant output equation, like (3.2) on page 26, this is immediately clear. The subspace methods can easily be extended to deal with a linear output equation with a time-varying 'D' matrix:

$$y_k = C_0 x_k + D \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right] + v_k. \tag{5.18}$$

However, what if the system has a time-varying output equation like (5.2)? At present, no subspace method exists for this case. A reasonable initial estimate can be obtained with the following procedure:

1. Use the LPV subspace method that is modified to deal with (5.18) to compute an estimate of the state sequence. Hence, only the influence of the matrices $C_1, C_2, \ldots, C_s$ is neglected.

2. Compute initial estimates of the system matrices as

$$\begin{aligned} [\widehat{A}, \widehat{B}] &= \widetilde{X}_{2,N+1} \Phi_{1,N}^T \left(\Phi_{1,N} \Phi_{1,N}^T \right)^{-1}, \\ [\widehat{C}, \widehat{D}] &= Y_{1,N} \Phi_{1,N}^T \left(\Phi_{1,N} \Phi_{1,N}^T \right)^{-1}, \end{aligned}$$

   where

$$\Phi_{1,N} := \left[\begin{array}{c} \widetilde{X}_{1,N} \\ P_{1,N} \odot \widetilde{X}_{1,N} \\ U_{1,N} \\ P_{1,N} \odot U_{1,N} \end{array} \right], \tag{5.19}$$

   with $\widetilde{X}_{1,N} = \left[\begin{array}{cccc} \widetilde{x}_1 & \widetilde{x}_2 & \cdots & \widetilde{x}_N \end{array} \right]$ the state estimate obtained from the LPV subspace method, and $U_{1,N}$ and $P_{1,N}$ defined similar to $\widehat{X}_{1,N}$.

   Simulation results presented in Subsection 6.1.3 confirm that this procedure generates a reasonable initial set of parameters θ for the optimization-based identification method, even if the system to be identified has an output that is generated by equation (5.2).

   Another way of generating initial estimates has been proposed by Lee and Poolla (1999). They use LTI subspace methods with an extended input given by

$$\left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right],$$

to estimate the matrices A_0, C_0, and B_i, D_i for $i = 0, 1, \ldots, s$. The remaining matrices A_i, C_i, for $i = 1, 2, \ldots, s$ are taken equal to zero, or initialized with random values. This procedure does not always yield a good initial estimate, especially when the matrices A_i, C_i, for $i = 1, 2, \ldots, s$ play an important role. This is supported by simulations performed by Lee and Poolla in which the algorithm did

not converge due to a bad initial estimate (see also Subsection 6.1.4). Therefore, in general the LPV subspace methods are to be preferred. However, the LPV subspace methods are much more computationally demanding, thus if the trick with the extended input works, calculation time can be saved.

5.2 Projected Gradient Search for Prediction-Error Optimization

Next, the prediction-error identification problem is treated. First, the one-step-ahead predictor for the system (5.3)–(5.4) is derived.

Lemma 5.4 *Given the system (5.3)–(5.4), the state equation can also be written as*

$$
x_{k+1} = \left(\begin{bmatrix} A & 0 \end{bmatrix} - M(K,C) \right) \begin{bmatrix} x_k \\ p_k \otimes x_k \\ p_k \otimes p_k \otimes x_k \end{bmatrix}
$$
$$
+ \left(\begin{bmatrix} B & 0 \end{bmatrix} - M(K,D) \right) \begin{bmatrix} u_k \\ p_k \otimes u_k \\ p_k \otimes p_k \otimes u_k \end{bmatrix} + K \begin{bmatrix} y_k \\ p_k \otimes y_k \end{bmatrix},
$$

where $M(K,C)$ is defined as

$$
M(K,C) := \Big[K_0 C_0, K_0 C_1 + K_1 C_0, K_0 C_2 + K_2 C_0, \ldots, K_0 C_s + K_s C_0,
$$
$$
K_1[C_1, C_2, \ldots, C_s], K_2[C_1, C_2, \ldots, C_s], \ldots, K_s[C_1, C_2, \ldots, C_s] \Big],
$$

and $M(K,D)$ is defined in a similar way.

Proof: This state equation is obtained by eliminating the signal v_k in equation (5.3). Equation (5.4) is used to express v_k as

$$
v_k = y_k - C \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} - D \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix}.
$$

Now it is easy to see that

$$
K \begin{bmatrix} v_k \\ p_k \otimes v_k \end{bmatrix} = K \begin{bmatrix} y_k \\ p_k \otimes y_k \end{bmatrix} - K \underbrace{\begin{bmatrix} C \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} \\ p_k \otimes C \begin{bmatrix} x_k \\ p_k \otimes x_k \end{bmatrix} \end{bmatrix}}
$$
$$
- K \underbrace{\begin{bmatrix} D \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix} \\ p_k \otimes D \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix} \end{bmatrix}}. \tag{5.20}
$$

The result follows by rewriting the under-braced terms using the definition of the Kronecker product. □

Given the input u_k, the parameter p_k, and the output y_k of the real system (5.3)–(5.4), the goal is to determine the matrices A, B, C, D, and K of the model

$$
\widehat{x}_{k+1}(\theta) = \left(\begin{bmatrix} A & 0 \end{bmatrix} - M(K, C) \right) \begin{bmatrix} \widehat{x}_k(\theta) \\ p_k \otimes \widehat{x}_k(\theta) \\ p_k \otimes p_k \otimes \widehat{x}_k(\theta) \end{bmatrix}
$$

$$
+ \left(\begin{bmatrix} B & 0 \end{bmatrix} - M(K, D) \right) \begin{bmatrix} u_k \\ p_k \otimes u_k \\ p_k \otimes p_k \otimes u_k \end{bmatrix}
$$

$$
+ K \begin{bmatrix} y_k \\ p_k \otimes y_k \end{bmatrix}, \tag{5.21}
$$

$$
\widehat{y}_k(\theta) = C \begin{bmatrix} \widehat{x}_k(\theta) \\ p_k \otimes \widehat{x}_k(\theta) \end{bmatrix} + D \begin{bmatrix} u_k \\ p_k \otimes u_k \end{bmatrix}, \tag{5.22}
$$

such that the output $\widehat{y}_k(\theta)$ of this model approximates the output y_k of the real system sufficiently accurately.

This identification problem can again be solved by using the same gradient search technique and local parameterization as described in Subsection 5.1.1. This follows from the fact that the state equation (5.21) can be written as

$$
\widehat{x}_{k+1}(\theta) = \mathcal{A} \begin{bmatrix} \widehat{x}_k(\theta) \\ \rho_k \otimes \widehat{x}_k(\theta) \end{bmatrix} + \mathcal{B} \begin{bmatrix} u_k \\ \rho_k \otimes u_k \end{bmatrix},
$$

with

$$
\rho_k = \begin{bmatrix} p_k \\ p_k \otimes p_k \end{bmatrix},
$$

and with the appropriate definitions of the matrices $\mathcal{A}$ and $\mathcal{B}$. This equation has the same structure as equation (5.5).

Solving the prediction-error problem differs from the output-error case in two respects.

First, Θ is now equal to

$$
\Theta = \begin{bmatrix} A & B_0 & K_0 & B_1 & K_1 & \cdots & B_s & K_s \\ C & D_0 & I_\ell & D_1 & I_\ell & \cdots & D_s & I_\ell \end{bmatrix},
$$

and θ is taken as $\theta = \Xi \, \mathrm{vec}(\Theta)$, with Ξ such that the block entries of Θ that, by definition are equal to the identity matrix, are not present in θ.

Second, the calculation of $\Psi_N(\theta)$ changes, because the state equation is now given by (5.21). The gradient $\Psi_N(\theta)$ can again be obtained by computing $\partial \widehat{y}_k / \partial \theta_i$, which is given by (5.16), for every parameter θ_i, and every time instant $k = 1, 2, \ldots, N$, but now the state $\mathcal{X}_k^i$ in (5.16) is obtained by simulating the state equa-

tion

$$
\begin{aligned}
\mathcal{X}^i_{k+1} \;=\;& \left(\begin{bmatrix} A & 0 \end{bmatrix} - M(K,C)\right) \begin{bmatrix} \mathcal{X}^i_k \\ p_k \otimes \mathcal{X}^i_k \\ p_k \otimes p_k \otimes \mathcal{X}^i_k \end{bmatrix} \\
& + \left(\begin{bmatrix} \dfrac{\partial A}{\partial \theta_i} & 0 \end{bmatrix} - M\left(\dfrac{\partial K}{\partial \theta_i}, C\right) - M\left(K, \dfrac{\partial C}{\partial \theta_i}\right)\right) \\
& \times \begin{bmatrix} \widehat{x}_k(\theta) \\ p_k \otimes \widehat{x}_k(\theta) \\ p_k \otimes p_k \otimes \widehat{x}_k(\theta) \end{bmatrix} \\
& + \left(\begin{bmatrix} \dfrac{\partial B}{\partial \theta_i} & 0 \end{bmatrix} - M\left(\dfrac{\partial K}{\partial \theta_i}, D\right) - M\left(K, \dfrac{\partial D}{\partial \theta_i}\right)\right) \\
& \times \begin{bmatrix} u_k \\ p_k \otimes u_k \\ p_k \otimes p_k \otimes u_k \end{bmatrix} \\
& + \dfrac{\partial K}{\partial \theta_i} \begin{bmatrix} y_k \\ p_k \otimes y_k \end{bmatrix}.
\end{aligned}
$$

This equation follows from (5.21) by straightforward differentiation. Similar to Lemma 5.3 on page 82 it can be shown that if the predictor of Lemma 5.4 on page 84 is uniformly bounded-input, bounded-output stable, and u_k and p_k are bounded, then $\mathcal{X}^i_{k+1}$ is bounded for all i.

To obtain an initial starting point, the LPV subspace identification methods discussed in Section 3.5 can be used, with or without the dimension reduction in Chapter 4. The key point is to compute an estimate of the state sequence (see also Subsection 5.1.3). The initial estimates of the system matrices are then obtained as follows:

$$
\begin{aligned}
[\widehat{C}, \widehat{D}] &= Y_{1,N} \Phi^T_{1,N} \left(\Phi_{1,N} \Phi^T_{1,N}\right)^{-1}, \\
\widehat{E}_{1,N} &= Y_{1,N} - [\widehat{C}, \widehat{D}]\Phi_{1,N}, \\
[\widehat{A}, \widehat{B}, \widehat{K}] &= \widetilde{X}_{2,N+1} \overline{\Phi}^T_{1,N} \left(\overline{\Phi}_{1,N} \overline{\Phi}^T_{1,N}\right)^{-1},
\end{aligned}
$$

where $\Phi_{1,N}$ is given by (5.19), and

$$
\overline{\Phi}_{1,N} := \begin{bmatrix} \Phi_{1,N} \\ \widehat{E}_{1,N} \\ P_{1,N} \odot \widehat{E}_{1,N} \end{bmatrix},
$$

and $\widetilde{X}_{1,N}$ and $\widetilde{X}_{2,N+1}$ contain the state estimate obtained from the LPV subspace method.

Chapter 6

Examples

This chapter presents several examples to illustrate the usefulness of the identification methods discussed in Chapters 3, 4, and 5. Several properties of the algorithms are investigated by running Monte-Carlo simulations in Section 6.1. In addition, this section contains an artificial multidimensional example, that illustrates the dimensionality problem in LPV subspace identification. In Section 6.2 the LPV identification methods are used to identify a dynamical model of helicopter rotor dynamics that can be used for active control of vibrations in helicopters.

6.1 Simulations

What follows are four different simulation experiments. Subsection 6.1.1 contains Monte-Carlo simulations to compare the LPV subspace identification methods discussed in Chapter 3. Subsection 6.1.2 is concerned with the dimension reduction method in Chapter 4. Subsection 6.1.3 deals with the projected gradient search method in Chapter 5. Finally, in Subsection 6.1.4 an example with multiple inputs and outputs is presented, and the dimension reduction as well as the projected gradient search methods are applied to this example.

6.1.1 Comparison of the LPV Subspace Methods

To compare the LPV subspace identification methods discussed in Section 3.3, a number of Monte-Carlo simulations were performed using an LPV system of the form (3.1)–(3.2) with the following system matrices:

$$
\begin{aligned}
A_0 &= \begin{bmatrix} 0 & 0.35 \\ -0.35 & 0 \end{bmatrix}, & A_1 &= \begin{bmatrix} 0.7 & 0 \\ -0.3 & 0.3 \end{bmatrix}, \\
B_0 &= \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, & B_1 &= \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}, \\
C &= \begin{bmatrix} 1 & -1 \end{bmatrix}, & D &= 0.05.
\end{aligned}
$$

Table 6.1: Total number of rows in the RQ factorizations for the identification methods described in Section 3.3 as function of the block size k, in case of a SISO system with one time-varying parameter.

Identification method	Procedure	$k = 1$	$k = 2$	$k = 3$
Two-block	I	33	130	515
Three-block	II	49	162	579
Two-stage three-block	II	74	276	1 062
Two-stage three-block red.	II	62	248	1 002

The input signal was generated by filtering a unit-variance zero-mean Gaussian distributed white-noise signal with a fourth-order Butterworth filter with a cut-off frequency of half the sample frequency. The system had one time-varying parameter p_k, which was taken to be a zero-mean white-noise sequence, uniformly distributed between -1 and 1. The disturbances w_k and v_k were zero-mean white-noise sequences with a Gaussian distribution. The covariance matrix of w_k equaled

$$\begin{bmatrix} \left(\frac{2\sigma}{3}\right)^2 & 0 \\ 0 & \left(\frac{\sigma}{3}\right)^2 \end{bmatrix},$$

where σ is the standard deviation of v_k. The value of σ was used to control the signal-to-noise ratio in the experiments.

The four identification methods in Section 3.3 were compared for different choices of the block size k, different numbers of data samples N, and different signal-to-noise ratios (SNR). Note that the minimum allowable block size is $k = 1$, because $k \geq n - 1$ should hold (see Section 3.2). The following combinations were tested:

- $N = 1500$, SNR $= 20$ dB, with $k = 1$, $k = 2$, and $k = 3$;

- SNR $= 20$ dB, $k = 3$, with $N = 500$ and $N = 1000$;

- $N = 1000$, $k = 3$, with SNR $= 10$ dB and SNR $= 15$ dB.

For each combination 100 experiments were performed, with different realizations for u_k, p_k, v_k, and w_k. Table 6.1 shows the number of rows in the RQ factorizations of the different identification methods for the block sizes listed above.

To compare the methods, the eigenvalues of the estimated A_0 and A_1 matrices were used. These eigenvalues are invariant with respect to a similarity transformation and give an indication of the dynamics of the system; these two properties make them suitable for comparing the subspace methods. The true eigenvalues of these matrices were $0 \pm 0.35i$ for A_0, and 0.3, 0.7 for A_1. In addition, as a measure of the quality of the output signals generated by the identified LPV model, the variance-accounted-for (VAF) was used; the VAF is defined as

$$\text{VAF} = \max\left\{1 - \frac{\text{var}(y_k - \widehat{y}_k)}{\text{var}(y_k)}, 0\right\} \times 100\%, \tag{6.1}$$

where $\widehat{y}_k$ denotes the estimated output signal, y_k is the real output signal, and $\mathrm{var}(\cdot)$ denotes the variance of a quasi-stationary signal. The VAF was evaluated using a fresh data set, that is, a data set that was generated using a realization of the input signal that was different from the data set used to estimate the model.

To show that the example is relevant for testing the LPV identification methods, the PI-MOESP subspace method (Verhaegen 1993) was used to estimate an LTI model. The VAF of this model on noise-free data was only 60%. Hence, the time-variation plays an important role in the dynamics of the system, and LTI subspace methods are not suitable for this example.

The identification results are presented in Figures 6.1–6.3. Note that for ease of comparison there is some overlap between the information presented in these figures; the second column of Figure 6.1 equals the first column of Figure 6.2, and the second column of Figure 6.2 equals first column of Figure 6.3.

The influence of the block size k is shown in Figure 6.1. Increasing the block size from $k = 1$ to $k = 2$ improves the performance for the two-block and three-block methods, but not for the two-stage methods. Increasing the block size from $k = 2$ to $k = 3$ deteriorates the performance for all four methods. This seems to contradict the analysis in Section 3.3.5, where it was concluded that the approximation error of the methods decreases with increasing block size. However, this is only true for infinite data lengths. If the block size is increased, the number of rows in the data matrices also increases while the number of columns corresponding to the data length stays the same. Therefore, by increasing the block size, the effect of the finite data length becomes more visible. This is especially true for the two-stage methods, which have about 1000 rows for $k = 3$, while the data length is only 1500 samples. This leads to the conclusion that for finite data lengths, a larger block size does not necessarily yield better models. Thus, the number of data points should be taken into account when choosing the block size.

Figure 6.2 shows that the performance of the methods deteriorates if the number of data points is decreased. From the discussion on the choice of the block size it is expected that the amount of deterioration depends to a large extend on the block size, and on the particular method used. The two-stage methods have much more rows in the data matrices than the other methods, so the effect of the finite data length shows up earlier. The plots do not show this effect, because 500 data samples can still be regarded large for a block size $k = 2$ (see Table 6.1).

If the SNR is decreased, the performance also decreases. This is illustrated by Figure 6.3. For a properly chosen block size, the amount of deterioration is about the same for each method.

A comparison of the four methods using Figures 6.1–6.3 shows that the two-stage three-block method and the two-stage three-block method with reduced dimensions have a comparable performance under varying experimental conditions. Since the method with reduced dimensions requires less computer memory, this method is preferred to the other one. The two-stage methods perform badly for large block size, because the effect of the finite data lengths becomes visible. The two-block method performs badly for small block sizes. The three-block method seems to be a nice compromise with respect to the choice of the block size. However, on the whole, the differences in performance among the methods are rather small.
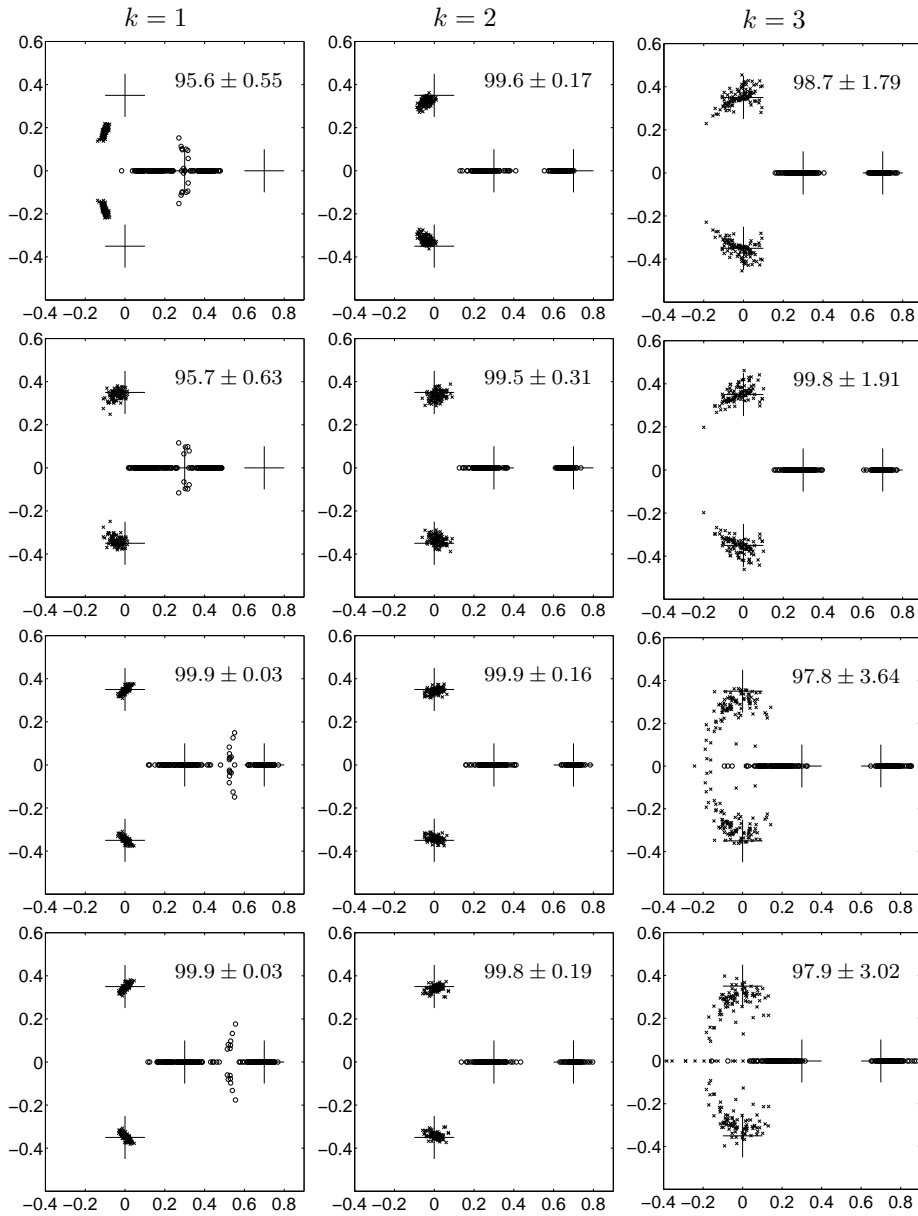
Figure 6.1: Estimated eigenvalues of the A_0 (*crosses*) and A_1 (*circles*) matrices in the complex plane, for 100 experiments with $N = 1500$, SNR = 20 dB, and block size $k = 1, 2, 3$. *From top to bottom:* two-block, three-block, two-stage three-block, and two-stage three-block method with dimension reduction. The big crosses correspond to the real values of the eigenvalues of the matrices A_0 and A_1. The numbers in the top right corner of each graph are the mean VAF and standard deviation for fresh data.
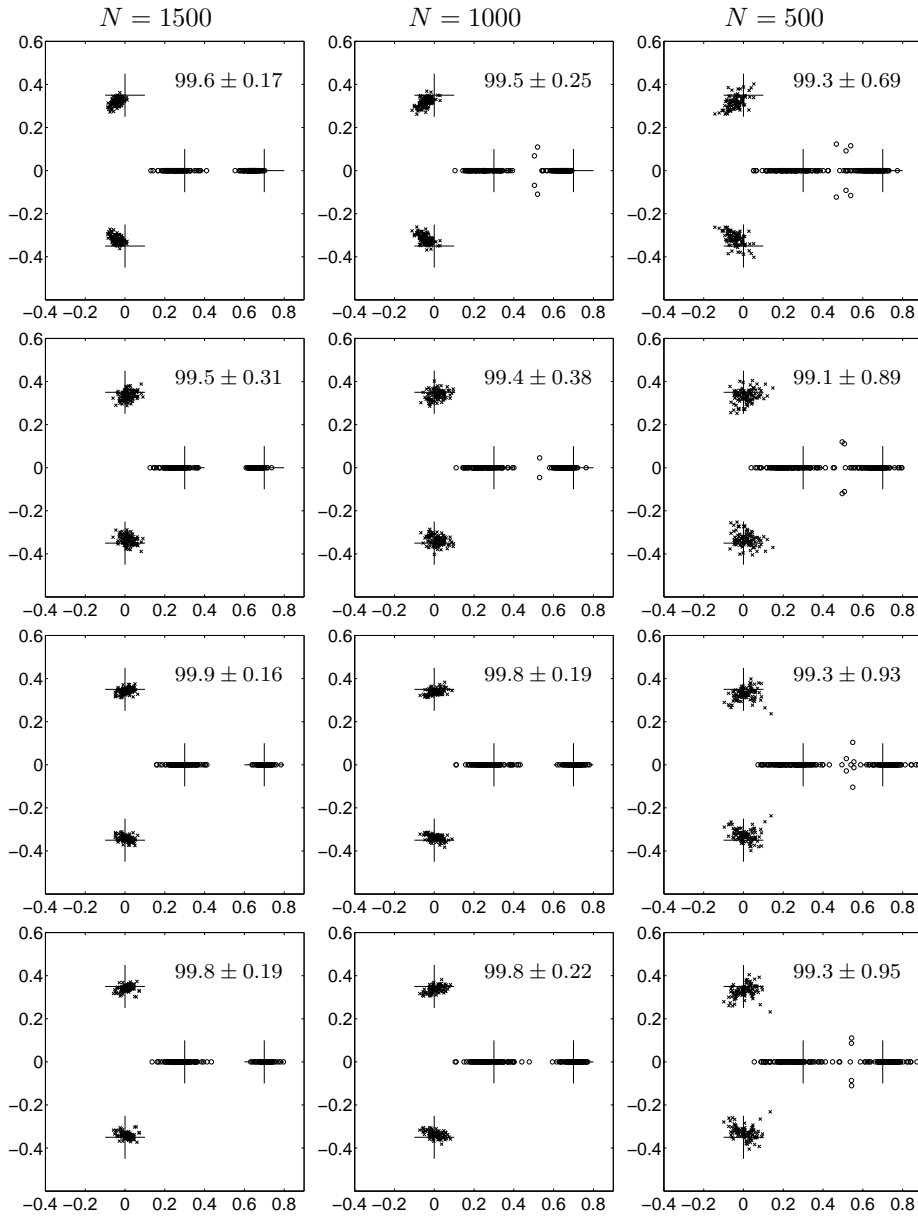
Figure 6.2: Estimated eigenvalues of the A_0 (*crosses*) and A_1 (*circles*) matrices in the complex plane, for 100 experiments with SNR = 20 dB, block size $k = 2$, and $N = 1500, 1000, 500$. *From top to bottom:* two-block, three-block, two-stage three-block, and two-stage three-block method with dimension reduction. The big crosses correspond to the real values of the eigenvalues of the matrices A_0 and A_1. The numbers in the top right corner of each graph are the mean VAF and standard deviation for fresh data.
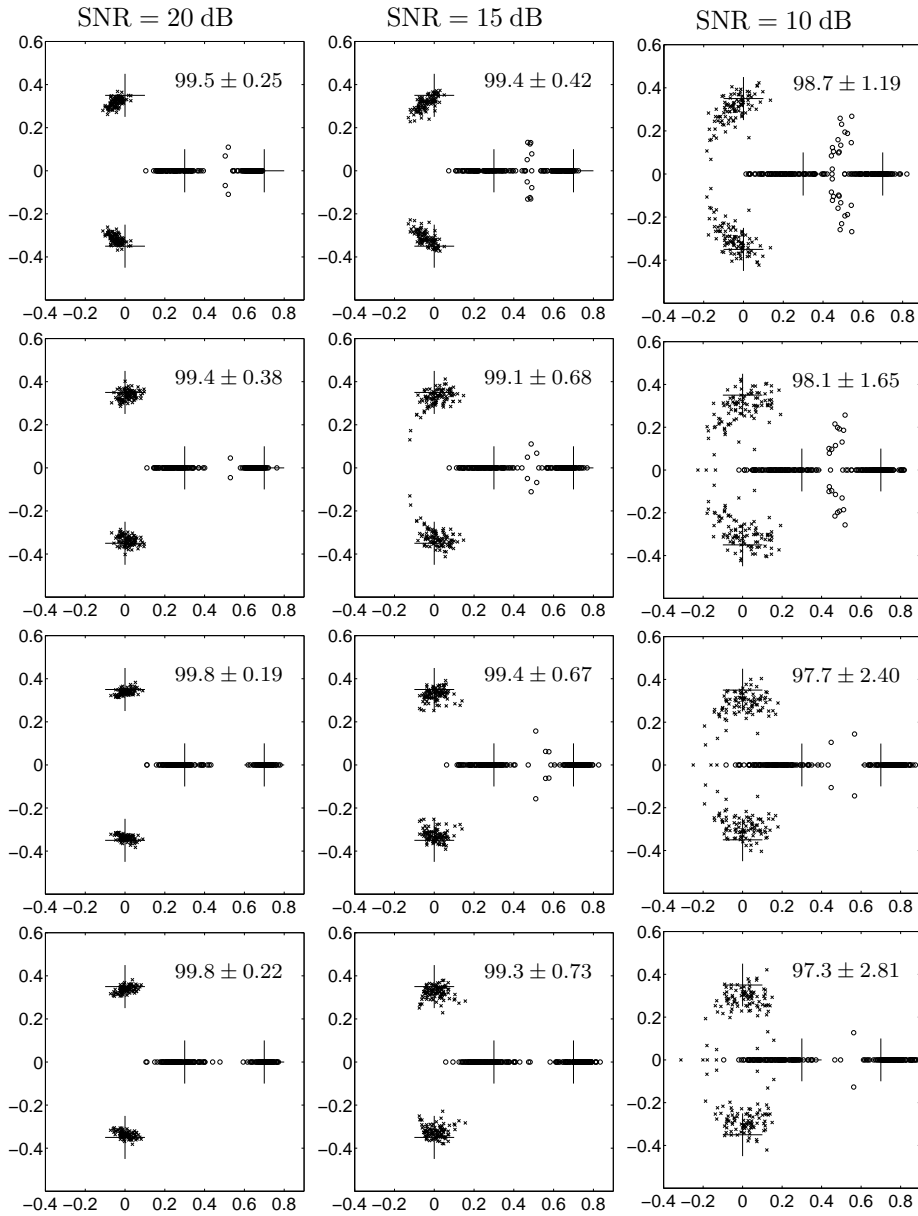
Figure 6.3: Estimated eigenvalues of the A_0 (*crosses*) and A_1 (*circles*) matrices in the complex plane, for 100 experiments with $N = 1000$, block size $k = 2$, and SNR $= 20, 15, 10$ dB. *From top to bottom:* two-block, three-block, two-stage three-block, and two-stage three-block method with dimension reduction. The big crosses correspond to the real values of the eigenvalues of the matrices A_0 and A_1. The numbers in the top right corner of each graph are the mean VAF and standard deviation for fresh data.
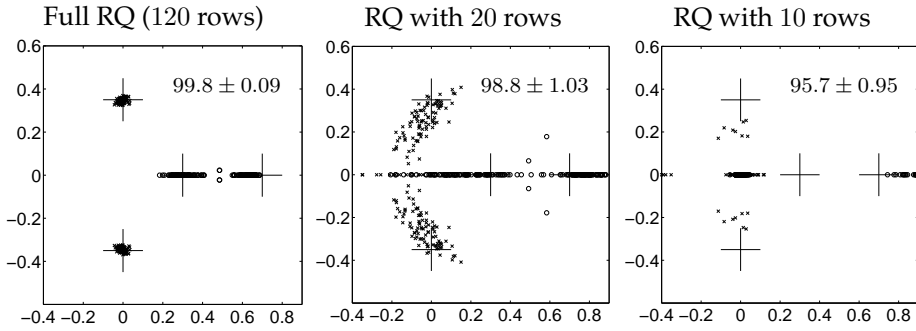
Figure 6.4: Estimated eigenvalues of the A_0 (*crosses*) and A_1 (*circles*) matrices in the complex plane for 100 experiments. The estimates are obtained with the identification method in Subsection 3.5.2, with and without dimension reduction. *From left to right:* no dimension reduction, using only the 20 most dominant rows, and using only the 10 most dominant rows. The big crosses correspond to the real values of the eigenvalues of the matrices A_0 and A_1. The numbers in the top right corner of each graph are the mean VAF and standard deviation for fresh data.

6.1.2   Subspace Identification with Dimension Reduction

The effect of the subset selection method for dimension reduction described in Chapter 4 was studied by means of a Monte-Carlo simulation. The simulations were carried out using the same system as in the previous section, and using the same input signal and time-varying parameter. The noise was generated as $w_k = K v_k$, with $K = [2/3, 1/3]^T$ and v_k a zero-mean white-noise sequence, with a standard deviation of 0.065. This corresponds to a signal-to-noise ratio of 20 dB.

Identification was performed with the two-block method in Subsection 3.5.2. The block size was $k = 2$. With this choice, the matrix Z_X contains 28 rows, and the matrices $Z_{P \odot X}$ and U together contain 92 rows. Three different models were estimated:

- without using dimension reduction; the full matrices Z_X, $Z_{P \odot X}$, and U are used;

- with dimension reduction where only the 20 most dominant rows are used (10 rows are selected from Z_X, and 10 rows are selected from $Z_{P \odot X}$ and U together);

- with dimension reduction where only the 10 most dominant rows are used (5 rows are selected from Z_X, and 5 rows are selected from $Z_{P \odot X}$ and U together).

The experiment was performed 100 times with different realizations for u_k, p_k, v_k, and w_k. The data sets contained 1500 samples. As in the previous subsection, the quality of the models was evaluated using the eigenvalues of the estimated A_0 and A_1 matrices and the VAF computed on a fresh data set. The results are shown in Figure 6.4.

It appears that a large dimension reduction from 120 rows to 20 rows can be achieved without deteriorating the model performance much. Although the estimated eigenvalues are slightly biased and have a large variance, the VAF values are still quite good. Pushing the dimension reduction further to 10 rows yields very biased eigenvalues, but the VAF is still reasonable. The reasons for this is that the rows that are selected by the dimension reduction procedure are such that they minimize the output error.

These results show that although the dimension reduction deteriorates the estimated models, the performance of these models is still quite reasonable. Hence, it is a viable approach when dimensions are too large to be handled.

6.1.3   Projected Gradient Search

This section presents some simulation results of the projected gradient search method for identification of LPV systems described in Chapter 5. Both the output-error algorithm and the prediction-error algorithm are treated.

Output-Error Example

To keep the computational load of the simulations low, a SISO system was chosen. A MIMO example is presented in the Subsection 6.1.4. The SISO system has a time-varying output equation and is of the form (5.1)–(5.2) on page 76 with the system matrices given by

$$A = \begin{bmatrix} 0 & 0.5 & 0 & 0.8 & 0.5 & 0 \\ -0.5 & 0 & -0.8 & 0.5 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & 0.4 & 0 \\ 0.5 & 0 & 0.3 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.5 & 2 & 1 & 0.5 & 2 & 1 \end{bmatrix},$$

$$D = \begin{bmatrix} 0.5 & 1 & 0.5 \end{bmatrix}.$$

The time-varying parameter vector p_k and the input signal u_k were generated as

$$p_k = \begin{bmatrix} \eta_k \\ \frac{1}{2}\sin(\frac{2\pi k}{100})\eta_k \end{bmatrix}, \qquad u_k = G(q)\mu_k + 4H(q)\nu_k,$$

where η_k is a uniformly distributed random sequence with unit variance, μ_k and ν_k are zero-mean white Gaussian noise sequences of unit variance, q denotes the shift operator, $G(q) := 0.75 + 1.05q^{-1} + 0.15q^{-2}$, and $H(q)$ is a second-order low-pass Butterworth filter with a cut-off frequency of one-fifth of the sample frequency. The system was simulated for 1000 samples, and the generated data were used to estimate the system matrices. Measurement noise v_k of unit variance was added to the simulated output signals.

The LPV subspace method in Subsection 3.5.2 with the dimension reduction in Chapter 4 was used to generate initial estimates of the system matrices. The block size was equal to 3. Only 3 rows were selected from the matrix Z_X, and only 9 rows from the matrices $Z_{P \odot X}$ and U together. Another data set of 1000 samples

(without noise) was generated to validate the obtained models. The quality of the output signals obtained by simulating the estimated LPV models, was measured using the VAF; see equation (6.1) on page 88. This identification experiment was repeated 100 times using different realizations for u_k, p_k, and v_k.

Figure 6.5 on page 96 shows the eigenvalues of the A_0, A_1, and A_2 matrices of the initial models and the optimized models, along with the real values. The true eigenvalues were: $0 \pm 0.5i$ for A_0, $0.25 \pm 0.76i$ for A_1, and $0, 0.5$ for A_2. The VAF values obtained from the data set used for validation are shown in Figure 6.6 on page 96. This figure shows both the VAF values of the initial model obtained from subspace identification and those of the optimized model. From these results it can be concluded that the subspace method yields reasonable initial estimates and that the optimization procedure always significantly improves the models.

Prediction-Error Example

To illustrate the prediction-error method described in Section 5.2, a set of 100 simulations was carried out along the same lines as for the output-error example of the previous section. In this case the system is of the form (5.3)–(5.4) on page 76, with a nonzero matrix K that equals

$$K = \left[\begin{array}{ccc} 0.1 & 0.1 & 0.2 \\ 0.5 & 0.2 & 0.1 \end{array} \right],$$

and of course, instead of the output-error optimization, a prediction-error optimization is carried out.

Figure 6.7 on page 97 shows the eigenvalues of the A_0, A_1, and A_2 matrices of the initial models and the optimized models, along with the real values. The VAF values obtained from the data set used for validation are shown in Figure 6.8 on page 97. This figure shows both the VAF values of the initial model obtained from subspace identification and those of the optimized model. From this figure it follows that the performance of the optimized models is not always good. Comparing the two eigenvalue plots in Figure 6.7, we see that in some cases the optimization algorithm fails to obtain a model with the correct eigenvalues. These cases correspond to the low values of the VAF in Figure 6.8. This bad performance is due to local minima in the cost function. Some of the initial estimates obtained from the subspace method are not close enough to the global optimum and cause the algorithm to get stuck in a local minimum. This behavior was not observed for the output-error case, because the output-error cost function is very different from the prediction-error cost function. Equation (5.21) on page 85 shows that the parameter dependence of the cost function is much more complicated in the prediction-error case. In a practical setting, it is advisable to try different initial models. These initial models can be easily generated by selecting a different number of rows in the dimension reduction procedure for LPV subspace identification discussed in Chapter 4.

6.1.4   Identification of a Multivariable LPV System

Below, simulation results are presented in which the dimension reduction method for LPV subspace identification in Chapter 4 and the projected gradient search
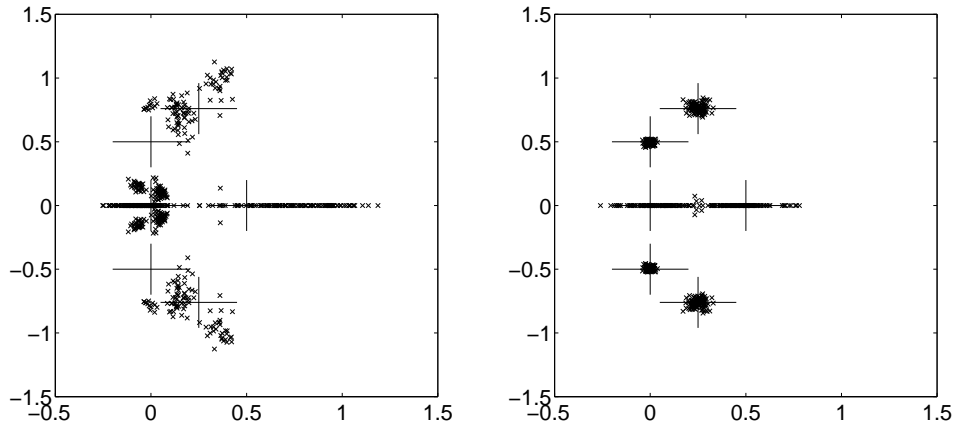
Figure 6.5: Output-error example in Subsection 6.1.3. Eigenvalues of the A_0, A_1, and A_2 matrices of the initial models (*left*) and of the optimized models (*right*) in the complex plane for 100 experiments. The big crosses correspond to the eigenvalues of the real system.
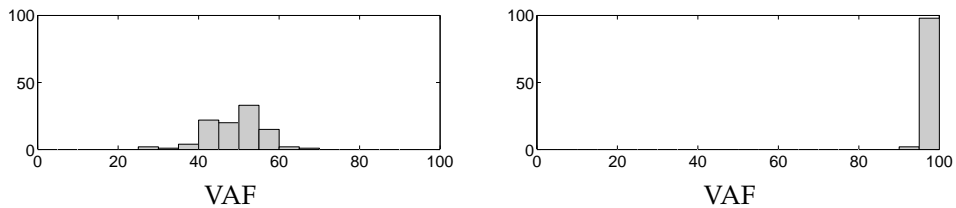
Figure 6.6: Output-error example in Subsection 6.1.3. Histogram of VAF values (%) obtained on a validation data set for the 100 initial models (*left*) and for the corresponding optimized models (*right*). The range of VAF values from 0 to 100% is divided into bins of 5%. For each bin, it is shown how many data sets out of the total 100 resulted in VAF values that fall into that bin.
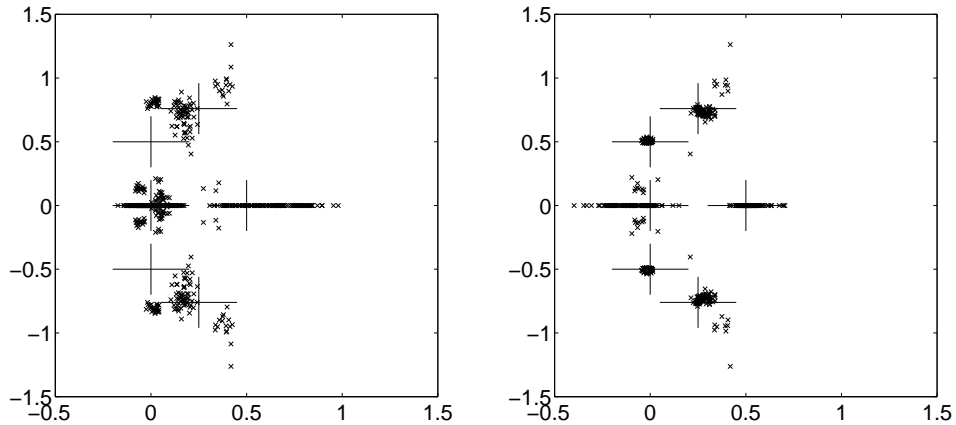
Figure 6.7: Prediction-error example in Subsection 6.1.3. Eigenvalues of the A_0, A_1, and A_2 matrices of the initial models (*left*) and of the optimized models (*right*) in the complex plane for 100 experiments. The big crosses correspond to the eigenvalues of the real system.
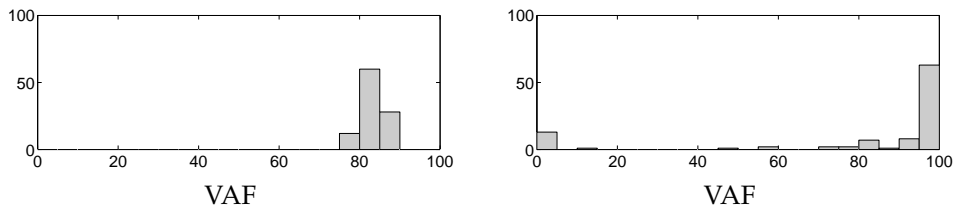
Figure 6.8: Prediction-error example in Subsection 6.1.3. Histogram of VAF values (%) obtained on a validation data set for the 100 initial models (*left*) and for the corresponding optimized models (*right*). The range of VAF values from 0 to 100% is divided into bins of 5%. For each bin, it is shown how many data sets out of the total 100 resulted in VAF values that fall into that bin.

method in Chapter 5 were used to identify a MIMO LPV system. The system under consideration is a fourth-order LPV system with three outputs, two inputs, and three time-varying parameters. It is given by equations (3.62)–(3.63) on page 56 with system matrices equal to:

$$A_0 = \begin{bmatrix} -1.3 & -0.6325 & -0.1115 & 0.0596 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} -0.51 & -0.1075 & -0.007275 & -0.0000625 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad A_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0.3 \end{bmatrix},$$

$$B_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad B_1 = B_2 = B_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.3 & 0.3 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The time-varying parameter vector p_k and the input signal u_k were generated as

$$p_k = \begin{bmatrix} \rho_k \\ \frac{1}{2}\sin(\frac{2\pi k}{100})\rho_k \\ \frac{1}{2}\cos(\frac{2\pi k}{100})\rho_k \end{bmatrix}, \qquad u_k = \begin{bmatrix} G(q)\mu_k + 4H(q)\nu_k \\ \xi_k \end{bmatrix},$$

where ρ_k is a uniformly distributed random sequence with unit-variance, μ_k, ν_k, and ξ_k are zero-mean white Gaussian noise sequences of unit-variance, q denotes the shift operator, $G(q) := 0.75 + 1.05q^{-1} + 0.15q^{-2}$, and $H(q)$ is a second-order low-pass Butterworth filter with a cut-off frequency of one-fifth of the sample frequency. The noise sequence v_k was generated as a zero-mean, unit-variance white-noise sequence with a Gaussian distribution; the matrix K equaled

$$K = \begin{bmatrix} 0.16 & 0 & 0 \\ 0 & 0.16 & 0 \\ 0 & 0 & 0.16 \\ 0.16 & 0 & 0 \end{bmatrix}.$$

The system was simulated for 1000 samples and the generated data set consisting of the time signals u_k, p_k, and y_k was used to identify a number of LPV models. This data set is called the estimation data set. Another data set of 1000 samples, called the validation set, was generated to evaluate the performance of the models.

Table 6.2: Model performance for the multivariable LPV example in Subsection 6.1.4: VAF values (%) for the one-step-ahead prediction and free-run simulation of the model obtained from subspace identification, and for the free-run simulation of the optimized model. The values are given for both the estimation and validation data sets.

	Estimation data			Validation data		
	one-step pred.	free run	optimized free run	one-step pred.	free run	optimized free run
Output 1	96.94	84.26	97.42	97.42	77.37	94.24
Output 2	98.29	85.43	98.75	98.82	76.93	94.88
Output 3	99.13	86.53	99.22	99.36	76.72	95.01

The two-block identification method in Subsection 3.5.2 with the dimension reduction in Chapter 4 was used to identify the system. The block size used for Z_X was different from the one used for $Z_{P \odot X}$ and U (see the discussion at the end of Subsection 3.3.5). The block size for Z_X was $j = 4$, the block size for the other matrices was $k = 5$. With these choices, the data matrix Z_X contains 6820 rows and the matrices $Z_{P \odot X}$ and U together contain 1 739 788 rows. Therefore, at least 1 746 608 samples are needed to estimate a model using the LPV subspace method. This in turn requires the storage of a 1 746 608 by 1 746 608 matrix. With 8 bytes for each element such a matrix takes up 22 729 gigabyte of computer memory. Consequently, this approach is not feasible.

To reduce the dimensions, all the rows that contained more than five multiplications were discarded. This reduced the number of rows to 5200 for Z_X and to 119 788 for $Z_{P \odot X}$ and U. To reduce the dimensions even further, the subset selection procedure in Chapter 4 was used to select the 52 most dominant rows from the remaining rows of Z_X and the 119 most dominant rows from the remaining rows of $Z_{P \odot X}$ and U. Selecting the rows took approximately 20 hours with Matlab running on a 400 MHz Pentium II.

The performance of the estimated models is summarized in Table 6.2. The one-step-ahead predictions on both estimation and validation data are almost perfect. The free-run simulation results are somewhat worse, but regarding the enormous reduction of the number of rows in the data matrices, still reasonable. Figure 6.9 on page 100 shows the time signals obtained for the first output, for both the one-step-ahead prediction and free-run simulation on the validation data.

Next, the LPV model obtained from the subspace identification algorithm was used as an initial estimate for the nonlinear optimization-based method in Chapter 5. After 394 iterations, the free-run VAF values on the validation data were improved by about 17% (see Table 6.2).

The generation of an initial model by the row selection procedure took considerable computation time. As discussed in Subsection 5.1.3, Lee and Poolla (1999) presented an alternative way of providing an initial estimate to the nonlinear op-
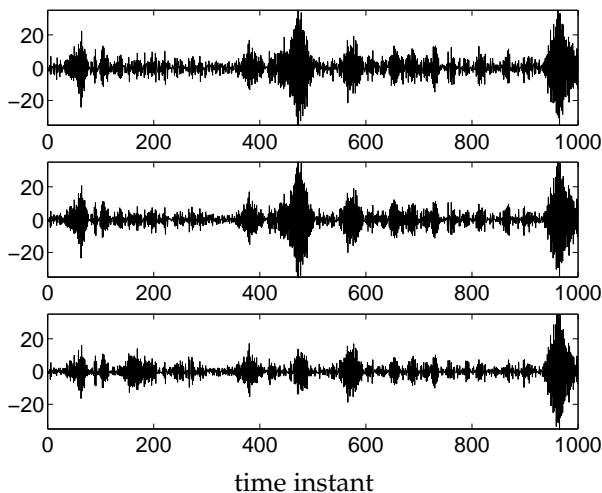
Figure 6.9: Time signals for the first output of the multivariable LPV example in Subsection 6.1.4. *From top to bottom:* the original output, the one-step-ahead prediction of the model, and the free-run simulation of the model.

timization method. With this method, an initial estimate can be computed within one minute, instead of several hours. The method consists of generating an initial model using an LTI subspace identification method with an extended input $[u_k^T, (p_k \otimes u_k)^T]^T$ to estimate the matrices A_0, B_i, D_i for $i = 0, 1, \ldots, s$, and taking $A_i = 0$ for $i = 1, 2, \ldots, s$. However, the initial estimate generated with this method was not very good. On the data set used for identification, the VAF values were only: 41.98%, 43.78%, and 42.44% for output 1, 2, and 3, respectively. Moreover, this initial model resulted in divergence of the nonlinear optimization. Thus, for this example the LPV subspace method with dimension reduction has to be used to obtain a useful initialization for the nonlinear optimization.

6.2 Identification of Helicopter Rotor Dynamics

Active control for the attenuation of vibrations in helicopters improves the comfort for crew and passengers and reduces fatigue in the rotor and other components of the helicopter. It has been a subject of research for many years and a number of approaches to the problem have been proposed in the literature; see for example the recent survey papers by Teves et al. (1995) and Friedmann and Millott (1995). The control methodologies vary considerably according to the design and implementation of the control algorithms, the choice of sensors and the location of actuators; however, they share the common goal of minimizing the vibratory loads that the main rotor applies to the body of the helicopter. It is well-known that the relation between vibratory loads and the pitch angles of the blades can be described by a time-varying linear model. In hover, when the helicopter hangs still in the air, the model becomes time-invariant; in forward flight at

constant speed, the model becomes time-periodic; but when the speed is allowed to change with time, the model becomes linear parameter-varying (LPV). In general, it is not easy to extract linear models in LPV form from high-order, complex simulators such as the ones used in rotorcraft work for vibration prediction. This section shows that the LPV identification methods discussed in the previous chapters can be used to determine from measurements of the vibratory loads and the pitch angle, an LPV model that describes the dynamics of a helicopter rotor blade. The detailed physical model used by Bittanti and Lovera (1996) for the analysis of the periodic dynamics of a rotor blade will be used as a benchmark to generate the data. The model is a simple one, based on classical modeling techniques, as described by Johnson (1980). First, this physical model is described, then the identification results are presented and discussed.

6.2.1 Helicopter Rotor Dynamics

This subsection gives a brief overview of the out-of-plane bending dynamics of a helicopter rotor blade as derived by Bittanti and Lovera (1996). For a more detailed discussion the reader is referred to their paper.

The dynamics of one blade of a helicopter rotor can be described by a time-varying linear state-space system of the form

$$\dot{x}_k = \mathcal{A}_k x_k + \mathcal{B}_k u_k, \tag{6.2}$$
$$y_k = \mathcal{C}_k x_k + \mathcal{D}_k u_k, \tag{6.3}$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}$ the input, and $y_k \in \mathbb{R}$ the output. The input signal corresponds to the pitch angle of the blade. The output equals the normalized vertical shear force at the root of the blade. The vertical shear is the difference between the aerodynamic load and the total inertia load acting on the blade. It is normalized to make it a dimensionless quantity. The dimension of the state vector n equals twice the number of out-of-plane bending modes q that are taken into account ($n = 2q$).

The time-varying system matrices are of the following form:

$$\mathcal{A}_k = \mathcal{A}_0(\mu_k) + \mathcal{A}_{1c}(\mu_k)\cos(\Omega k) + \mathcal{A}_{1s}(\mu_k)\sin(\Omega k), \tag{6.4}$$
$$\mathcal{B}_k = \mathcal{B}_0(\mu_k) + \mathcal{B}_{1s}(\mu_k)\sin(\Omega k) + \mathcal{B}_{2c}(\mu_k)\cos(2\Omega k), \tag{6.5}$$
$$\mathcal{C}_k = \mathcal{C}_0(\mu_k) + \mathcal{C}_{1c}(\mu_k)\cos(\Omega k) + \mathcal{C}_{1s}(\mu_k)\sin(\Omega k) + \mathcal{C}_{2c}\cos(2\Omega k), \tag{6.6}$$
$$\mathcal{D}_k = \mathcal{D}_0(\mu_k) + \mathcal{D}_{1s}(\mu_k)\sin(\Omega k) + \mathcal{D}_{2c}(\mu_k)\cos(2\Omega k), \tag{6.7}$$

where Ω is the frequency of rotation of the rotors, which is constant under normal flight conditions, and where μ_k is the *rotor advance ratio*, which depends on the forward velocity v_k of the helicopter as follows:

$$\mu_k = \frac{v_k}{\Omega R},$$

with R the radius of the rotor. The matrix $\mathcal{A}_0$ depends on the advance ratio μ_k in the following way:
$$\mathcal{A}_0(\mu_k) = \mathcal{A}_{00} + \mathcal{A}_{01}\mu_k + \mathcal{A}_{02}\mu_k^2.$$

Similar expressions hold for the matrices $\mathcal{B}_0$, $\mathcal{C}_0$, and $\mathcal{D}_0$. The other matrices in (6.4)–(6.7) can each be written as a linear combination of μ_k and μ_k^2; for example,

$$\mathcal{A}_{1c} = \mathcal{A}_{1c1}\mu_k + \mathcal{A}_{1c2}\mu_k^2.$$

From the expressions for the matrices in (6.4)–(6.7), it is clear that in hover ($v_k = 0$ and thus $\mu_k = 0$) the system (6.2)–(6.3) is time-invariant. In general, the system (6.2)–(6.3) is time-varying and can be written as an LPV system with affine parameter dependence,

$$\dot{x}_k = \left(A_0 + \sum_{i=1}^{s}[p_k]_i A_i \right) x_k + \left(B_0 + \sum_{i=1}^{s}[p_k]_i B_i \right) u_k, \tag{6.8}$$

$$y_k = \left(C_0 + \sum_{i=1}^{s}[p_k]_i C_i \right) x_k + \left(D_0 + \sum_{i=1}^{s}[p_k]_i D_i \right) u_k, \tag{6.9}$$

where $[p_k]_i$ denotes the ith element of the time-varying parameter vector $p_k \in \mathbb{R}^s$ given by

$$p_k = \begin{bmatrix} \mu_k & \mu_k^2 & \mu_k \cos(\Omega k) & \mu_k^2 \cos(\Omega k) & \mu_k \sin(\Omega k) \\ & & \mu_k^2 \sin(\Omega k) & \mu_k \cos(2\Omega k) & \mu_k^2 \cos(2\Omega k) \end{bmatrix}^T. \tag{6.10}$$

In the special case of constant forward velocity v_k, the advance ratio μ_k is also constant and a simplification can be made. In this case the dynamics of the rotor blade can be described by an LPV model of the form (6.8)–(6.9) with

$$p_k = \begin{bmatrix} \cos(\Omega k) & \sin(\Omega k) & \cos(2\Omega k) \end{bmatrix}^T. \tag{6.11}$$

Below, the continuous-time LPV model, given by equations (6.8)–(6.9), will be approximated by a discrete-time model that has a similar structure:

$$x_{k+1} = \left(A_0 + \sum_{i=1}^{s}[p_k]_i A_i \right) x_k + \left(B_0 + \sum_{i=1}^{s}[p_k]_i B_i \right) u_k, \tag{6.12}$$

$$y_k = \left(C_0 + \sum_{i=1}^{s}[p_k]_i C_i \right) x_k + \left(D_0 + \sum_{i=1}^{s}[p_k]_i D_i \right) u_k, \tag{6.13}$$

The goal is to identify such a discrete-time model from measurements of the input u_k, output y_k, and time-varying parameters p_k. Both the case of constant forward velocity and the case of variable forward velocity will be analyzed.

6.2.2 Simulation Results

This subsection presents some numerical examples that show that the identification methods discussed in the previous chapters yield LPV models that accurately describe the dynamic behavior of the rotor that was discussed in the previous subsection. The blade model used for this study resembles the rotor of commercial and military Agusta A109 helicopters (http://www.agusta.com). The main characteristics of this rotor are summarized in Table 6.3.

Table 6.3: Mechanical and aerodynamical characteristics of the rotor of Agusta A109 helicopters (Bittanti and Lovera 1996).

Number of blades M	4
Rotor angular frequency Ω	40 rad/s
Rotor radius R	5.5 m
Mass per unit length m	5 kg/m
Stiffness EI	$1.8 \cdot 10^3$ Nm^2
Lift-curve slope α	5.7 rad^{-1}
Lock number γ	7.84
Blade chord c	0.3 m

The experiments proceed in the following way. The physical model is used to generate two data sets of 1000 samples with a sampling time of 0.0013 seconds. The input signal for both data sets equals a unit-variance, zero-mean Gaussian white noise. The generated output signal is contaminated with a zero-mean Gaussian white noise such that the signal-to-noise ratio amounts 15 dB. One data set is used for estimating the LPV models, and is called the identification data. The other data set is used to evaluate the performance of the models, and is called the validation data. The variance-accounted-for (VAF), which is defined by (6.1) on page 88, is used as a measure of the quality of the output signals generated by the identified LPV models.

Constant Forward Velocity

First, the case of constant forward velocity was considered. The rotor advance ratio μ_k was fixed at a value of 0.3 and the number of out-of-plane bending modes taken into account was 2. The generated identification data was used to estimate a fourth-order discrete-time LPV model of the form (6.12)–(6.13), with p_k given by (6.11).

The subset-selection procedure in Chapter 4 was used in combination with the two-block subspace method in Subsection 3.5.2. The block size used for Z_X was different from the one used for $Z_{P \odot X}$ and U (see the discussion at the end of Subsection 3.3.5). The block size for Z_X was $j = 3$, the block size for the other matrices was $k = 4$. With this choice, the matrix Z_X contains 2728 rows, and the matrices $Z_{P \odot X}$ and U contain 695 984 rows. These numbers make it clear that dimension reduction is desirable. The rows of $Z_{P \odot X}$ and U that contain more than six multiplications were discarded. This reduced the number of rows to 55 760. The rationale behind this preliminary truncation is that these higher-order terms will not contribute much to the accuracy of the estimates when only 1000 samples are available. The subset-selection technique was used to select the 28 most dominant rows from Z_X and the 55 most dominant rows from the remaining rows of $Z_{P \odot X}$ and U. The model obtained by this subspace procedure, had a VAF on the validation data of 94.5%. This is a remarkably good performance taking into account the enormous dimension reduction that has been carried out on the matrices
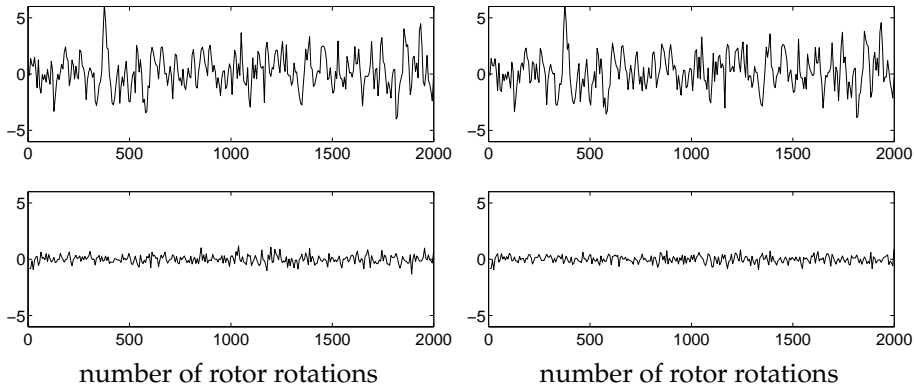
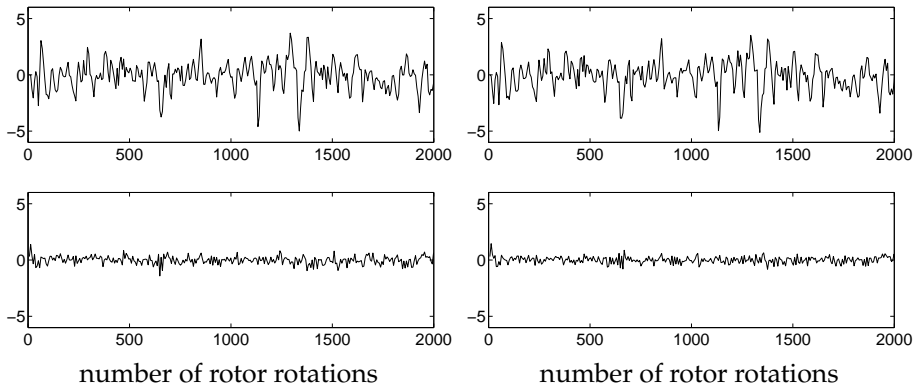**Figure 6.10:** *Top:* first part of the output of the LPV model obtained by subspace identification (*left*) and of the output of the optimized LPV model (*right*) for constant forward velocity; *bottom:* difference between the model output and the real output.
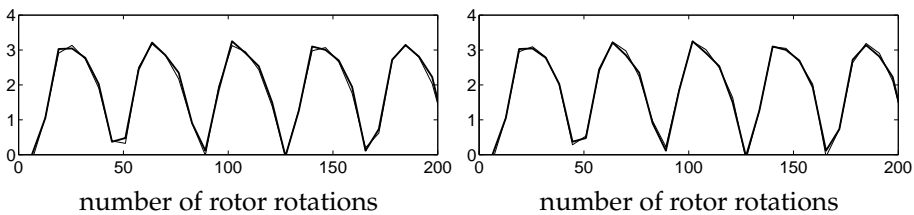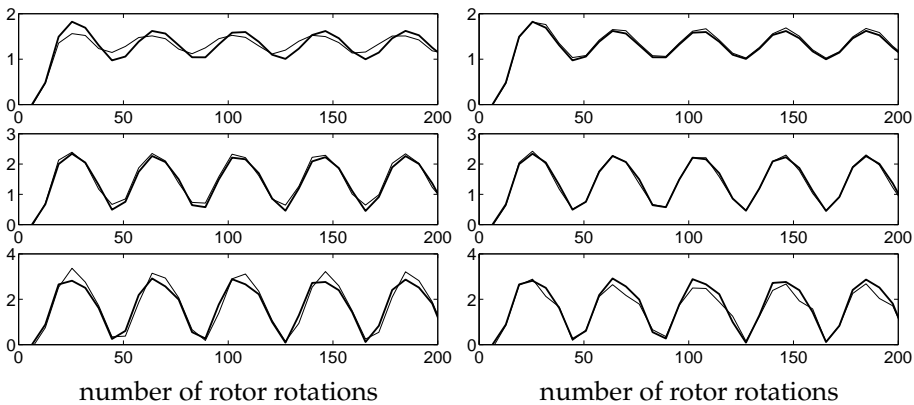

Figure 6.11: Step response of the LPV model obtained by subspace identification (*left*) and of the optimized LPV model (*right*) for constant forward velocity. The *thick lines* are the real step responses, the *thin lines* the responses from the model.

Z_X, $Z_{P \odot X}$, and U. Next, the optimization-based procedure in Chapter 5 was used to improve the model obtained by subspace identification. The optimized model had a VAF of 95.9% on the validation data. The first part of the output signal of the subspace model and the optimized model are shown in Figure 6.10, along with the errors between the real output and the model outputs. Figure 6.11 compares the step responses of the models with the real step response. It can be concluded that the model obtained by subspace identification is already a very good model, because it was only slightly improved by optimization.

Variable Forward Velocity

To simulate variable forward velocity, the rotor advance ratio μ_k was taken equal to a random sequence, uniformly distributed between 0 and 0.3. The number of out-of-plane bending modes equaled 2. The generated data were used to estimate

a fourth-order discrete-time LPV model of the form (6.12)–(6.13) with p_k given by (6.10).

The parameter vector p_k contained a lot more terms compared with the constant velocity case. This results in an enormous increase in the number of rows in the data matrices Z_X, $Z_{P \odot X}$, and U. Therefore, the block sizes were slightly reduced to $k = j - 1 = 3$. With this choice, the matrix Z_X contains $14\,760$ rows, and the matrices $Z_{P \odot X}$ and U contain $10\,746\,108$ rows. Again, a preliminary reduction on the number of rows in $Z_{P \odot X}$ and U was performed; the rows that contain more than six multiplications were discarded. This reduced the number of rows to $571\,644$. The subset-selection technique was used to select the 14 most dominant rows from Z_X and the 57 most dominant rows from the remaining rows of $Z_{P \odot X}$ and U. The model obtained in this way had a VAF of 92.1% on the validation data. Note again that the performance is surprisingly good, despite the enormous dimension reduction that was carried out on the matrices Z_X, $Z_{P \odot X}$, and U. Optimization of the LPV model slightly increased the VAF to 95.1%. Figure 6.12 on page 106 shows the first part of the output signal of the subspace model, the first part of the output signal of the optimized model, and the errors between the real output and the model outputs. Figure 6.13 on page 106 compares the step responses of the models with the real step response for three different values of μ_k. These results again confirm that subspace identification yields a good model.

In practice, the rotor has a large number of out-of-plane bending modes, some more relevant than others. To obtain a compact model for control purposes, not all modes are taken into account. An identification experiment was performed that resembles this situation. A data set was generated using an eighth-order (four modes) physical model, and this data set was used to estimate a fourth-order LPV model with the LPV identification methods. The truncation and choice of the block sizes was the same as in the previous experiment. The VAF on the validation data of the model obtained from subspace identification was 92.5%. After optimization the VAF became 95.3%. Figure 6.14 on page 107 shows the first parts of the output signals obtained from the initial subspace model and from the optimized model; it also shows the errors between the real output and the model outputs. Figure 6.15 on page 107 shows the step responses of the models for different values of μ_k. Again, the difference in performance between the model obtained by subspace identification and the optimized model was small.
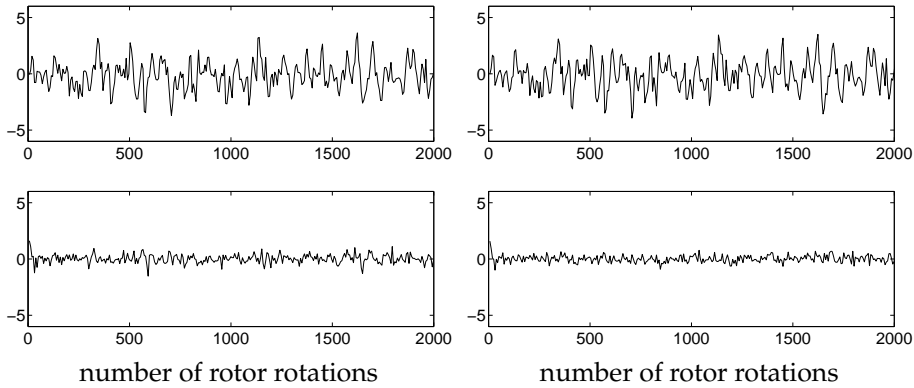
Figure 6.12: *Top:* first part of the output of the LPV model obtained by subspace identification (*left*) and of the output of the optimized LPV model (*right*) for varying forward velocity; *bottom:* difference between the model output and the real output.
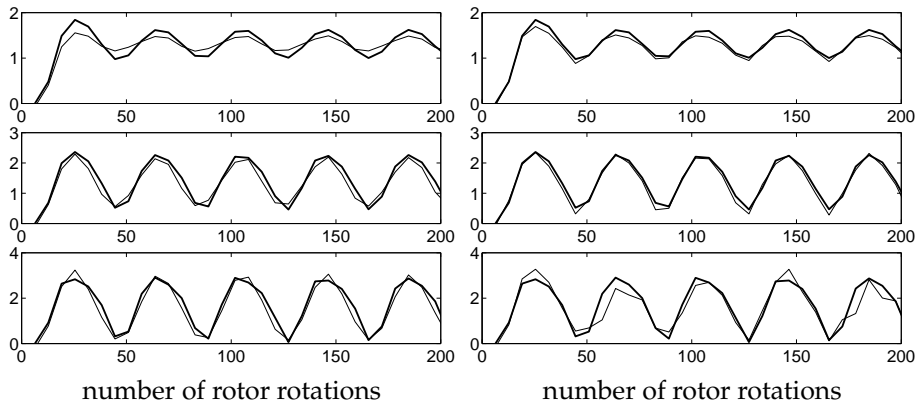
Figure 6.13: Step responses of the LPV model obtained by subspace identification (*left*) and of the optimized LPV model (*right*) for varying forward velocity; *from top to bottom:* step response for $\mu_k = 0.05$, $\mu_k = 0.15$, and $\mu_k = 0.25$. The *thick lines* are the real step responses, the *thin lines* the responses from the model.

Figure 6.14: *Top:* first part of the output of the fourth-order LPV model obtained by subspace identification (*left*) and of the output of the optimized LPV model (*right*), from data generated by an eighth-order physical model with varying forward velocity; *bottom:* difference between the model output and the real output.

Figure 6.15: Step responses of the fourth-order LPV model obtained by subspace identification (*left*) and of the optimized LPV model (*right*), from data generated by an eighth-order physical model with varying forward velocity; *from top to bottom:* step response for $\mu_k = 0.05$, $\mu_k = 0.15$, and $\mu_k = 0.25$. The *thick lines* are the real step responses, the *thin lines* the responses from the model.

Chapter 7

Conclusions and Recommendations for Part I

To complete the first part of this thesis, a summary of the results on LPV identification is provided and some suggestions for further research are made.

Subspace Identification

Subspace techniques can be used to identify multivariable LPV systems with a time-invariant output equation and an affine time-varying parameter dependence in the state equation. The key point in LPV subspace identification is the reconstruction of the state sequence. Once the state sequence has been reconstructed, the system matrices follow from a linear least-squares problem. The reconstruction of the state sequence is always an approximation, because the effect of the unknown initial state has to be neglected. The state sequence is approximated by a linear combination of the rows of certain data matrices that contain the available measurements of the input, output, time-varying parameter, and products of time-lagged versions of these signals. This linear combination is unknown, but follows by solving a linear least-squares problem. Different identification algorithms can be derived, depending on which combination of data matrices is used to approximate the state. Several methods have been discussed in this part of the thesis: the two-block, three-block, and two-stage methods. In the two-block method, the available data are split into two parts that are used to construct two sets of data matrices. Similarly, in the three-block methods three sets of data matrices are constructed. The two-stage methods approximate the state sequence in two steps that involve the same set of data matrices. The two-stage methods use larger data matrices and hence result in computationally more burdensome schemes. Extensive simulation results showed that the accuracy of the above-mentioned methods differ, but that the differences are usually small. Therefore, the computationally demanding two-stage methods are not to be preferred.

It can be argued that if the LPV system is uniformly asymptotically stable, and the time-varying parameter vector is bounded, the approximation error decreases by increasing the block size used for constructing the data matrices. This suggests that a large block size is desirable. However, the number of rows in the data matrices grows exponentially with the block size; thus a large block size results in an enormous number of rows in the data matrices. Hence, the choice of the block size is a trade-off between accuracy and computational load. Furthermore, with a large block size the effect of the finite data length becomes visible. The number of columns in the data matrices is proportional to the number of data points. If the number of columns is much larger than the number of rows in the data matrices, then the influence of the noise on the state estimate can be neglected. However, this is no longer true if the block size is such that the number of rows approaches the number of columns. Monte-Carlo simulations confirmed that too small a block size and too large a block size both lead to large approximation errors.

Three open problems in LPV subspace identification are: (1) the derivation of norm bounds on the approximation errors of the methods that were discussed; (2) the derivation of conditions on the system, time-varying parameter, and input that ensure that a certain rank condition needed to approximate the state sequence is satisfied; (3) the reduction of the bias that results from neglecting the initial state.

A solution to the first problem would allow a direct comparison of the approximation errors of the different methods. The norm bounds on the approximation errors are likely to depend on the block size. Deriving this dependency is however not trivial, because it involves products of matrices with varying dimensions. Ultimately, what one likes to show is that the approximation errors decrease with increasing block size. Simulation results suggest that such a relation between block size and approximation errors exists.

The second problem of the rank condition is closely related to the notion of persistency of excitation. Certain data matrices have to have full row rank for the subspace identification procedure to work. Since the data matrices contain products of time-lagged versions of the input, output, and time-varying parameter, this rank condition is related to the invertibility of higher-order moment matrices. The work of Nowak and Van Veen (1994, 1995) on invertibility of such matrices might be a good starting point to derive conditions on the signals and the LPV system for which the rank condition is satisfied. Other work that might be useful is on persistency of excitation for input-output LPV systems by Bamieh and Giarré (1999) and on persistency of excitation for bilinear systems by Dasgupta, Shrivastava, and Krenzer (1992) (a bilinear system can be regarded as a special case of the LPV system; see Part II).

The third problem is to derive subspace methods that have reduced approximation error or bias that results from neglecting the unknown initial state. Until now all subspace methods proposed in the literature suffer from this bias. As discussed in this thesis, this bias does not need to be a problem, if the subspace methods are only used to generate initial estimates for optimization-based methods. However, it would of course be nice to avoid the nonlinear optimization completely, like in linear subspace identification where the models obtained are often already very accurate and do not need to be improved by nonlinear optimization.

Dimension Reduction in Subspace Identification

The dimension of the data matrices in LPV subspace identification can be a major problem. The number of rows in the data matrices grows exponentially with the block size. Since the block size has to be at least equal to the system order, higher-order systems require a ridiculous amount of computer memory. To reduce the dimension of the data matrices, it is suggested that only the most dominant rows are used and the other ones discarded. Recall that the data matrices are used in a least-squares problem to approximate the state sequence. In this setting, selecting the most dominant rows boils down to solving a subset-selection problem. Subset selection can be carried out efficiently with an RQ factorization. It is possible to perform the selection row by row such that there is no need to store the entire data matrix into the memory. Since certain rows are discarded, an approximation error is introduced. Simulations show that often a surprisingly large dimension reduction can be achieved that does not deteriorate the model performance much; for example, using 180 out of 180000 rows results in only 5% performance loss.

The row selection procedure is an effective way of reducing the dimensions of the data matrices and it makes the LPV subspace identification practically feasible. However, the large memory requirements are partly converted into large computation times. The derivation of more efficient algorithms is a topic for future research. Possible directions towards more efficient algorithms are the use of higher-order statistics or cumulants as in the work of Tsoulkas, Koukoulas, and Kalouptsidis (1999) on bilinear system identification, and the use of more efficient implementation techniques like multichannel embedding as used in Volterra system identification (Glentis, Koukoulas, and Kalouptsidis 1999).

Identification Based on Nonlinear Optimization

LPV systems can also be identified with an optimization-based approach that minimizes the prediction error. An optimization-based method can deal with LPV systems that have affine parameter dependence in both the state and output equation; it can even deal with fractional parameter dependence as shown by Lee and Poolla (1999). However, in this thesis the discussion is limited to affine parameter dependence.

The success of an optimization-based method depends to a large extend on the initial starting point and on the parameterization used. The subspace method can provide a good initial starting point. Although subspace identification assumes a time-invariant output equation, it can compute an approximate model for the parameter-varying case. Simulations confirm that this approximation yields good initial starting points for the nonlinear optimization. An alternative way of generating an initial starting point has been proposed by Lee and Poolla (1999) and is based on using LTI subspace methods with an extended input. This procedure makes use of a rough approximation to the LPV system and does not always yield a good initial point. Therefore, in general the LPV subspace methods are to be preferred. However, the LPV subspace methods are much more computationally demanding, thus if the trick with the extended input works, calculation time can be saved.

The issue of choosing an appropriate parameterization of the LPV system can

be circumvented as discussed by Lee and Poolla (1999). It is possible to use a full parameterization and at each iteration determine the active parameters and only update these parameters. Such an approach has numerical advantages over canonical parameterizations. It can be used with output measurement noise as well as with a special type of process noise.

To conclude, subspace identification and optimization-based identification for LPV systems form a perfect combination. On the one hand, subspace identification provides the optimization-based approach with a good initial starting point. On the other hand, subspace identification yields an approximate model —because of neglecting the initial state, and possibly because of the dimension reduction— which can be improved by the optimization-based method if its performance is not satisfactory.

Since LPV system identification is still in its infancy, there is plenty of room for future research in both subspace methods as well as in nonlinear optimization-based methods. In particular, methods for recursive identification and closed-loop identification are of interest.

Part II

Bilinear State-Space Systems

Chapter 8

Introduction

Nonlinear models come in a variety of complexity. A nonlinear model with a simple structure is the bilinear state-space system. It is a simple nonlinear extension of a linear system: the evolution of the state of the bilinear system does not only depend on the input and the state, but also on the product between the input and the state. This simple extension makes the bilinear system more general than linear systems, and yet less complex than general nonlinear approximators like Volterra series and neural networks. Bilinear systems are therefore easier to analyze, and a considerable body of theoretical results has been obtained over the years (for an overview see for example Mohler 1973; Bruni, Dipillo, and Koch 1974; Mohler and Kolodziej 1980; Brewer 1987; Mohler 1991).

When considering bilinear state-space systems, it is important to distinguish between continuous-time systems and discrete-time systems, because they have different approximating properties. Exact discretization of a continuous-time bilinear system destroys the bilinear structure (Dunoyer et al. 1997). A discrete-time bilinear system can only approximate a continuous-time bilinear system, for example, by application of a forward Euler approximation.

A continuous-time bilinear system is described by the following two equations:

$$\frac{d}{dt}x_t = Ax_t + F(u_t \otimes x_t) + Bu_t,$$
$$y_t = Cx_t + Du_t,$$

where $x_t \in \mathbb{R}^n$ is the state, $u_t \in \mathbb{R}^m$ the input, and $y_t \in \mathbb{R}^\ell$ the output. Several real-life systems inherently show bilinear dynamic behavior in continuous time. Population models and compartment models are bilinear. These models can be used to describe a wide variety of phenomena in the fields of physics, biology, medicine, and ecology (Mohler 1973; Brewer 1987). Examples discussed by Mohler (1973) include nuclear fission, heat transfer, biochemical reactions controlled by enzyme-catalysts, respiratory chemostat, cardiovascular regulator, thermoregulation of the human body, economic growth, and urban dynamics. Some other examples are: a distillation column (España and Landau 1975, 1978), nuclear magnetic resonance

(Brillinger 1990, 1994), nuclear rocket propulsion (Mohler 1991), and a DC motor (Daniel-Berhe and Unbehauen 1996).

Continuous-time bilinear systems are universal approximators: any continuous causal nonlinear system can be approximated arbitrarily well by a bilinear system within any bounded time interval (Mohler and Kolodziej 1980; Chen and Billings 1989). A continuous-time bilinear system can be obtained by Carleman linearization of a continuous-time nonlinear system (Svoronos, Stephanopoulos, and Aris 1980; Rugh 1981). Carleman linearization is a technique in which the nonlinear functions that describe the nonlinear state-space system are approximated by finite Taylor series expansions, and the state of the system is extended with higher-order products of the states. Bilinear state-space systems can approximate a large class of nonlinear systems, because they are realizations of finite Volterra series with time-invariant separable kernels (Isidori and Ruberti 1973; Rugh 1981; Crouch 1987).

This part of the thesis will focus on discrete-time bilinear state-space systems. A discrete-time bilinear system is described by

$$x_{k+1} \;=\; Ax_k + F(u_k \otimes x_k) + Bu_k, \tag{8.1}$$
$$y_k \;=\; Cx_k + Du_k, \tag{8.2}$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ the input, and $y_k \in \mathbb{R}^\ell$ the output. The reader should observe that this bilinear system can in fact be viewed as a special case of the LPV system described in Part I. The LPV system (3.1)–(3.2) on page 26 with a linear time-invariant output equation can be converted into a bilinear system by setting $p_k = u_k$, $A_0 = A$, $B_0 = B$, and $A_i = F_i$, $B_i = 0$ for $i = 1, 2, \ldots, m$, with $F = [F_1, F_2, \ldots, F_m]$. As a consequence, the identification methods in Part I can also be used for discrete-time bilinear systems. This part of the thesis discusses some subtle issues that arise when the identification methods in Part I are applied to bilinear systems, and it presents some new methods that exploit the specific structure of the discrete-time bilinear system.

It is important to observe that discrete-time bilinear systems do not have the universal approximation property. Hence, it is impossible to accurately approximate every discrete-time nonlinear system with a discrete-time bilinear system. Carleman linearization of a discrete-time nonlinear system does not result in a bilinear system, but in a *state-affine system* (Rugh 1981, p. 254). A single-input state-affine system is given by

$$z_{k+1} \;=\; \sum_{i=0}^{M-1} A_i u_k^i z_k + \sum_{i=1}^{M} B_i u_k^i,$$
$$y_k \;=\; \sum_{i=0}^{M-1} C_i u_k^i z_k + \sum_{i=1}^{M} D_i u_k^i,$$

where z_k is the state, and M is the degree of approximation. Typically, the dimension of the state z_k is $M \cdot n$, where n is the state dimension of the nonlinear system that is approximated. State-affine systems are universal approximators (Diaz and Desrochers 1988). The bilinear system is only a special case of the state-affine system ($M = 2$, $B_2 = 0$, $C_1 = 0$, $D_2 = 0$). Like the bilinear system, the state-affine

system is also a special case of the LPV system described in Part I. It is not difficult to see that by setting

$$
p_k = \begin{bmatrix} u_k \\ u_k^2 \\ \vdots \\ u_k^{M-1} \end{bmatrix},
$$

the LPV system (2.1)–(2.2) on page 14 includes the state-affine system. Therefore, the identification methods in Part I are also suitable for the identification of state-affine systems, but this is not further discussed in this thesis.

Although discrete-time bilinear systems are not universal approximators, they still represent a fairly general class of nonlinear systems: any discrete-time finite Volterra series expansion with time-invariant separable kernels can be realized by a discrete-time bilinear state-space system (Rugh 1981; Monin and Salut 1996). This is discussed in some detail in Section 8.1 below.

The main motivation for studying discrete-time bilinear systems is their simplicity compared with other nonlinear models like, for example, neural networks. If a linear system does not approximate the real-life system sufficiently accurately, the next step would be to try a bilinear model. Bilinear models can represent certain nonlinear behavior, and have been successfully applied to several real-life problems. For example, Marchi, dos Santos Coelho, and Rodrigues Coelho (1999) compared the performance of linear, bilinear, Volterra, and neural network models for an experimental nonlinear fan-plate process. They showed that the bilinear model performs much better than the linear model, and is only slightly worse than the neural network models, which are computationally far more complex.

The remainder of this chapter consists of six sections. Section 8.1 briefly discusses the realization of finite Volterra series expansion by a discrete-time bilinear system. In Section 8.2 input-output descriptions of bilinear state-space systems are derived. An overview of identification methods for bilinear systems is given in Section 8.3. Sections 8.4 and 8.5 introduce some concepts and assumptions that are needed in the forthcoming chapters. Finally, the organization of this part of the thesis is described in Section 8.6.

8.1   Bilinear Realization of Volterra Series Expansion

Volterra series expansions have been widely used to describe the input-output behavior of nonlinear dynamical systems (Rugh 1981). A discrete-time nonlinear system defined by a finite Volterra series expansion has the following input-output description:

$$
y_k = \sum_{i=1}^{d} \sum_{\tau_1=0}^{k} \sum_{\tau_2=0}^{\tau_1} \cdots \sum_{\tau_i=0}^{\tau_{i-1}} h_i(k, \tau_1, \tau_2, \ldots, \tau_i) u_{\tau_1} u_{\tau_2} \cdots u_{\tau_i}, \tag{8.3}
$$

where d is the *degree* of the Volterra series expansion, and $h_i(k, \tau_1, \tau_2, \ldots, \tau_i)$ are the *kernels*. Note that by taking $d = 1$, the Volterra series expansion becomes an

impulse response description of a linear system, that is,

$$y_k = \sum_{\tau=0}^{k} h(k, \tau) u_\tau.$$

A Volterra kernel $h_i(k, \tau_1, \tau_2, \ldots, \tau_i)$ is called *time-invariant* if

$$h_i(k, \tau_1, \tau_2, \ldots, \tau_i) = \overline{h}_i(k - \tau_1, \tau_1 - \tau_2, \ldots, \tau_{i-1} - \tau_i).$$

A time-invariant Volterra kernel $\overline{h}_i(\tau_1, \tau_2, \ldots, \tau_i)$ is called *separable* if

$$\overline{h}_i(\tau_1, \tau_2, \ldots, \tau_i) = \sum_{r=1}^{q} \psi_{r1}(\tau_1) \psi_{r2}(\tau_2) \cdots \psi_{ri}(\tau_i),$$

for a finite integer q. For time-invariant separable kernels, the following important result can be stated:

Theorem 8.1 *The finite Volterra series expansion (8.3) is realizable by a finite-order bilinear state-space system (8.1)–(8.2) if the kernels $h_i(k, \tau_1, \tau_2, \ldots, \tau_i)$ are time-invariant and separable.*

The proof has been given by several authors, for example Rugh (1981) and Monin and Salut (1996).

8.2   Input-Output Representations

Assuming that the initial state of the bilinear system equals zero, recursive substitution of the state equation (8.1) yields the following input-output description (Inagaki and Mochizuki 1984):

$$
\begin{aligned}
y_k \;=\;& Du_k + \sum_{j=0}^{k-1} CA^j B u_{k-j-1} \\
&+ \sum_{j=0}^{k-1} \sum_{h=0}^{k-1} \sum_{i=1}^{m} CA^j F_i A^h B [u_{k-j-1}]_i u_{k-j-h-2} \\
&+ \sum_{j=0}^{k-1} \sum_{h=0}^{k-1} \sum_{f=0}^{k-1} \sum_{g=1}^{m} \sum_{i=1}^{m} CA^j F_i F_g A^h B [u_{k-j-1}]_i [u_{k-j-f-2}]_g u_{k-j-f-h-3} \\
&+ \cdots
\end{aligned}
\tag{8.4}
$$

Note that this description is in fact a Volterra series expansion.

The derivation of an input-output description from the bilinear state-space system is much easier for systems in *phase-variable* form. A bilinear system in phase-

variable form has the following system matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & & & & 1 \\ a_n & a_{n-1} & a_{n-2} & \cdots & a_1 \end{bmatrix}, \quad F = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ f_n & f_{n-1} & \cdots & f_1 \end{bmatrix},$$

$$B = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}^T, \quad C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \quad D = 0.$$

By successive elimination of the states, the input-output description can be derived; it equals

$$y_k = \sum_{i=1}^{n} c_i u_{k-i} + u_{k-n} \sum_{i=1}^{n} (f_i y_{k-i} + d_i u_{k-i}) + \sum_{i=1}^{n} a_i y_{k-i},$$

where

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} := \begin{bmatrix} 1 & & & & \\ -a_1 & 1 & & & \\ -a_2 & -a_1 & 1 & & \\ \vdots & \vdots & & \ddots & \\ -a_{n-1} & -a_{n-2} & \cdots & -a_1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

$$\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} := \begin{bmatrix} 0 & & & & \\ -b_1 & 0 & & & \\ -b_2 & -b_1 & 0 & & \\ \vdots & \vdots & & \ddots & \\ -b_{n-1} & -b_{n-2} & \cdots & -b_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

Some authors consider bilinear input-output descriptions directly, that is, they do not derive the input-output description from a bilinear state-space system, but they postulate an input-output description in bilinear form. An example of such a bilinear input-output description is

$$y_k = \sum_{i=1}^{q_1} a_i y_{k-i} + \sum_{i=1}^{q_2} b_i u_{k-i} + \sum_{i=1}^{q_3} \sum_{j=0}^{q_4} c_{ij} y_{k-i} u_{k-j}. \tag{8.5}$$

The relation between this bilinear input-output description and the bilinear state-space system is however not immediately clear.

8.3   Overview of Identification Methods

The identification of continuous-time and discrete-time bilinear systems calls for different kinds of methods. Since this thesis focuses on discrete-time systems, only some key references are given for continuous-time system identification; the overview for discrete-time systems is more elaborate. An overview of some early

methods for continuous-time bilinear system identification was given by Bruni, Dipillo, and Koch (1974) and by Mohler and Kolodziej (1980). An overview of both continuous-time and discrete-time methods up to 1987 was given by Fnaiech and Ljung (1987).

8.3.1   Continuous-Time Identification Methods

The response of a continuous-time bilinear system is obtained by integrating the differential equation that describes its dynamics; this results in an equation describing the relation between the current output and past values of the input through several integrals. Bilinear system identification then boils down to estimating the system parameters in this equation. A popular approach for bilinear system identification is to approximate the input-output equation with certain basis functions, such that the parameter estimation problem is reduced to an algebraic problem (see for example Paraskevopoulos, Tsirikos, and Arvanitis 1994). Several types of basis functions have been used for this purpose: Walsh functions (Karanam, Frick, and Mohler 1978; Jha, Saxena, and Rajamani 1992), block-pulse functions (Cheng and Hsu 1982; Dai and Sinha 1992), Chebyshev polynomials (Liu and Shih 1984), Legendre polynomials (Hwang and Chen 1986), and Hartley-based modulating functions (Daniel-Berhe and Unbehauen 1998a,b).

Another approach to continuous-time bilinear system identification was described by Brillinger (1990) and is based on second- and third-order spectral procedures.

8.3.2   Discrete-Time Identification Methods

Identification methods for discrete-time bilinear systems in input-output form differ in the way they treat the noise that acts on the system. Let z_k denote the noise-free output of the system, then the bilinear system can be described by

$$z_k = \sum_{i=1}^{q_1} a_i z_{k-i} + \sum_{i=1}^{q_2} b_i u_{k-i} + \sum_{i=1}^{q_3} \sum_{j=0}^{q_4} c_{ij} z_{k-i} u_{k-j}, \tag{8.6}$$

$$y_k = z_k + v_k, \tag{8.7}$$

where v_k is a white-noise disturbance. A basic problem is that the noise-free output z_k is not measurable (Fnaiech and Ljung 1987). A similar problem occurs in linear system identification (Ljung 1999).

Based on how the noise is treated, the identification methods can be divided into two main classes: the *equation-error methods* and the *output-error methods*. In the equation-error method the influence of v_k is simply neglected; a model of the type (8.5) is estimated. The output-error methods take the noise term into account and estimate a model of the form (8.6)–(8.7). Only the output-error methods yield unbiased parameter estimates. However, it involves a nonlinear estimation procedure that is computationally extensive, and that may have several local minima. Statistical properties of the output-error methods, like consistency and asymptotic efficiency, were discussed by Svoronos, Stephanopoulos, and Aris (1981) and Chen, Zinober, and Ruan (1996). Because of the problems associated with the

nonlinear optimization in the output-error methods, the equation-error approach, which boils down to solving a linear least-squares problem, is also popular.

A second distinction can be made between *recursive* or *on-line* identification methods and *off-line* identification methods. The recursive methods update the estimates of the parameters at every time instant; the off-line methods collect a batch of data samples and use the complete batch to estimate the parameters.

Recursive identification methods for bilinear systems can be derived from recursive methods for linear system identification as described by Fnaiech and Ljung (1987); this leads to the recursive least-squares (RLS), the extended least-squares, the recursive instrumental variable, and the recursive prediction-error method. The extended least-squares and the recursive prediction-error method are of the output-error type; the RLS and the recursive instrumental variable method are of the equation-error type. Other recursive equation-error type methods include the least-mean-square (LMS), and the conjugate gradient method (Bose and Chen 1995). The LMS algorithm is the less computationally extensive method, but is outperformed by the RLS method with respect to speed of convergence. The conjugate gradient method outperforms the RLS method in speed of convergence and is often also computationally more efficient. Several methods were proposed to reduce the bias of the recursive equation-error approach, by modifying the least-squares cost function, for example: improved least-squares (Zhu and Leung 1999), and modified least-squares (Meddeb, Tourneret, and Castanie 2000). Dai and Sinha (1989) proposed a robust RLS method that can successfully handle data containing outliers for input-output systems derived from phase-variable state-space descriptions. They extended their method to use instrumental variables to avoid biased estimates (Dai and Sinha 1991b).

Besides the bias issue, convergence of the recursive methods is an important subject. Several authors have worked on proving convergence, among them are: Zhao and Lu (1991a,b), Dai and Sinha (1989, 1991b), and Liu and Wang (1993).

A variety of methods were proposed for off-line identification of bilinear systems. Baheti, Mohler, and Spang III (1980) used second-order correlations for a phase-variable state-space bilinear system with binary and ternary input signals, to define a least-squares problem involving the parameters of the bilinear system. Inagaki and Mochizuki (1984) estimated the kernels in the Volterra series expansion (8.4) and constructed a state-space realization with the techniques of Isidori (1973). A nonlinear optimization-based maximum-likelihood method was presented by Gabr and Rao (1984). An instrumental variable approach was described by Ahmed (1986). Han, Kim, and Sung (1996) discussed an extended generalized total least-squares method. Ralston and Boashash (1997) suggested using bandlimited regression, to reduce the bias and to deal with outliers in the off-line equation-error approach for input-output bilinear systems. To avoid the nonlinear optimization problem in the output-error approaches for input-output bilinear systems, Meddeb, Tourneret, and Castanie (1998a) used Bayesian inference in combination with a Gibbs sampler to simulate the parameter a posteriori probability density function. Cerone (1995) discussed the identification of input-output bilinear systems with bounded errors. He derived bounds for the identified parameters. However, as argued by Belforte (1992), the proved result only allows for parameter overbounding; tight parameter bounds can only be obtained in a fairly

restrictive case. Robust off-line identification methods based on Huber's minmax principle were derived by Dai, Sinha, and Puthenpura (1989), and Dai and Sinha (1991a). A completely different approach was taken by Tsoulkas, Koukoulas, and Kalouptsidis (1999) who used cross-cumulants information to determine the parameters of a noise-free input-output bilinear system driven by a white-noise input.

All the methods described above are only suitable for time-invariant bilinear systems; Meddeb, Tourneret, and Castanie (1998b) proposed methods for time-varying bilinear input-output systems.

Recently, subspace identification methods were proposed for the off-line identification of bilinear state-space systems. An overview of these methods is presented at the beginning of Chapter 9.

The methods described above deal with bilinear systems in input-output form or in phase-variable state-space form, except for the subspace methods, and the method of Inagaki and Mochizuki (1984), which deal with the more general state-space description (8.1)–(8.2). Verdult, Verhaegen, and Chou (1999) described a nonlinear optimization-based identification method that also deals with the general state-space description; this method is discussed in detail in Chapter 11.

8.4 System Analysis

This section presents some concepts that are used in the forthcoming chapters.

8.4.1 Similarity Transformations

In Subsection 2.3.1 it was shown that the LPV state-space system representation is not unique: several choices of the state sequence exist that result in the same input-output behavior. Obviously, since the bilinear system is a special case of the LPV system, the same holds for the bilinear system. For any nonsingular similarity transformation T, the system

$$\begin{aligned} \overline{x}_{k+1} &= A_T \overline{x}_k + F_T(u_k \otimes \overline{x}_k) + B_T u_k, \\ y_k &= C_T \overline{x}_k + D_T u_k, \end{aligned}$$

with the transformed state $\overline{x}_k = T^{-1}x_k$ and the system matrices: $A_T = T^{-1}AT$, $F_T = [T^{-1}F_1 T, T^{-1}F_2 T, \ldots, T^{-1}F_m T]$, $B_T = T^{-1}B$, $C_T = CT$, and $D_T = D$; has the same input-output behavior as the system (8.1)–(8.2).

8.4.2 Stability

The concept of stability for the bilinear system is similar to the concept of stability for the LPV system as discussed in Subsection 2.3.2.

Definition 8.1 *The bilinear state equation*

$$x_{k+1} = Ax_k + F(u_k \otimes x_k), \tag{8.8}$$

is called uniformly exponentially stable if for every finite constant $\beta > 0$ there exist a finite constant $\gamma > 0$ and a constant λ, $0 \leq \lambda < 1$ such that for every $j \leq k$, every x_j, and every u_k satisfying

$$\sup_k \|u_k\|_2 \leq \beta,$$

the corresponding solution satisfies

$$\|x_k\|_2 \leq \gamma \lambda^{k-j} \|x_j\|_2.$$

Definition 8.2 *The bilinear state-space system (8.1)–(8.2) is called uniformly bounded-input, bounded-output stable if there exists a finite constant η such that for any j and any input signal u_k the corresponding response y_k for $x_j = 0$ satisfies*

$$\sup_{k \geq j} \|y_k\|_2 \leq \eta \sup_{k \geq j} \|u_k\|_2.$$

Corollary 2.1 on page 21 and Corollary 2.2 on page 21 are adapted for the bilinear system as follows:

Corollary 8.1 *The bilinear state equation (8.8) is uniformly exponentially stable if given a finite constant $\beta > 0$ there exist a finite constant $\gamma > 0$ and a constant λ, $0 \leq \lambda < 1$ such that*

$$\left\| \prod_{t=j}^{k-1} \left(A + \sum_{i=1}^{m} [u_t]_i F_i \right) \right\|_2 \leq \gamma \lambda^{k-j},$$

for all k, j such that $k > j$, and for all u_k such that

$$\sup_k \|u_k\|_2 \leq \beta.$$

Corollary 8.2 *If the bilinear state equation (8.8) is uniformly asymptotically stable, then the bilinear system (8.1)–(8.2) is uniformly bounded-input, bounded-output stable.*

8.4.3   Asymptotic Stationarity and Covariances

In general, the state sequence of the bilinear system (8.1)–(8.2) driven by a stochastic input signal u_k, will not be stationary, that is, its mean and covariance can vary in time. If the bilinear system is driven by a white-noise sequence, conditions can be derived for which the mean and covariance of the state sequence become asymptotically constant over time; these conditions are given in the following theorem:

Theorem 8.2 *If the input u_k of the bilinear system (8.1)–(8.2) is a zero-mean white-noise sequence, and if all the eigenvalues of the matrix A and of the matrix*

$$A \otimes A + \sum_{i=1}^{m} \mathbf{E}\left[[u_k]_i^2\right] F_i \otimes F_i$$

have a magnitude that is smaller than one, then

$$\begin{array}{rclcl} \mathbf{E}[x_k] & = & \mathbf{E}[x_{k+1}] & = & 0, \\ \mathbf{E}[x_k x_k^T] & = & \mathbf{E}[x_{k+1} x_{k+1}^T] & = & P, \end{array}$$

where $\mathbf{E}[\cdot]$ *denotes statistical expected value, and* P *satisfies*

$$P = APA^T + \sum_{i=1}^{m} \mathbf{E}\Big[[u_k]_i^2\Big] F_i P F_i^T + B\mathbf{E}[u_k u_k^T]B^T.$$

The proof of this theorem is given by Priestley (1988, p. 59–63). This theorem will be used in Chapter 11.

8.4.4 Observability

Several notions of observability exist for bilinear systems, because observability depends on the input sequence. Different observability concepts have been discussed by Hara and Furuta (1977). Here, only one type of observability is introduced: the one that is commonly used to define minimal state-space realizations of bilinear systems. This type of observability is called *observability with some input* by Hara and Furuta (1977). Its definition is based on the indistinguishability of the state.

Definition 8.3 *(Rugh 1981, p. 169) A state x of the bilinear system (8.1)–(8.2) is called indistinguishable (from 0) if the response y_k with $x_0 = x$ is identical to the response with $x_0 = 0$ for some input u_k.*

Observability is defined as follows:

Definition 8.4 *(Rugh 1981, p. 169) The bilinear system (8.1)–(8.2) is called observable if there are no indistinguishable states.*

Observability of the bilinear system can be judged from its system matrices, because of the following result:

Theorem 8.3 *The bilinear system (8.1)–(8.2) is observable if and only if*

$$\mathrm{rank}(\mathcal{O}_n) = n,$$

where

$$\mathcal{O}_j = \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_j \end{bmatrix},$$

and

$$Q_0 = C, \qquad Q_j = \begin{bmatrix} Q_{j-1}A \\ Q_{j-1}F_1 \\ \vdots \\ Q_{j-1}F_m \end{bmatrix} \quad (j = 1, 2, \dots).$$

This theorem can be proven in different ways. A proof based on invariant subspaces was given by Isidori and Ruberti (1973) and Rugh (1981, p. 170); an alternative proof was presented by Hara and Furuta (1977).

8.5   Assumptions

The forthcoming chapters present several identification methods for bilinear systems subject to disturbances. Process noise w_k and measurement noise v_k act on the system in the following way:

$$
\begin{aligned}
x_{k+1} &= Ax_k + F(u_k \otimes x_k) + Bu_k + w_k, \\
y_k &= Cx_k + Du_k + v_k.
\end{aligned}
$$

The disturbances w_k and v_k are assumed to be sequences of independent Gaussian distributed random variables with zero-mean and bounded second- and fourth-order moments. The goal is to estimate the system matrices A, B, C, D, and F up to a similarity transformation from a finite number of measurements of the input u_k and the output y_k.

Throughout this part of the thesis it is assumed that the bilinear system is uniformly bounded-input, bounded-output stable in the sense of Definition 8.2. It is also assumed that the bilinear system is observable as in Definition 8.4. In Chapters 9 and 10, a somewhat stronger assumption is made that the linear part of the bilinear system is observable; in other words, that the linear system A, B, C, D constructed from the bilinear system is observable. Note that this implies that the bilinear system is also observable in the sense of Definition 8.4.

In Chapter 9 it is assumed that the bilinear system is driven by a deterministic input signal u_k, which is such that for all $j \in \mathbb{Z}$

$$
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} u_k u_{k+j}^T = R_u(j),
$$

with $R_u(j)$ a bounded matrix that is positive definite for $j = 0$.

In Chapters 10 and 11 the input signal u_k is assumed to be a stochastic signal, such that for all $j \in \mathbb{Z}$

$$
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} u_k u_{k+j}^T = \mathbf{E}[u_k u_{k+j}^T], \quad \text{w. p. } 1,
$$

with $\mathbf{E}[u_k u_{k+j}^T]$ a bounded matrix that is positive definite for $j = 0$, and where $\mathbf{E}[\cdot]$ denotes statistical expected value, and 'w. p. 1' is used to mean 'with probability 1'.

In all chapters it assumed that the disturbances v_k and w_k are statistically independent of the input u_k; therefore, for all $j \in \mathbb{Z}$

$$
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} u_k v_{k+j}^T = 0, \quad \text{w. p. } 1.
$$

A similar expression holds for w_k.

Because it is assumed that the bilinear system is bounded-input, bounded-output stable, it also holds that for all $j \in \mathbb{Z}$

$$
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} y_k y_{k+j}^T = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \mathbf{E}[y_k y_{k+j}^T], \quad \text{w. p. } 1,
$$

with the implication that these limits exist. A formal proof of this result is based on Theorem 2.3 from the book by Ljung (1999, p. 43). In addition, the following holds:

$$\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} y_k v_{k+j}^T = 0, \quad \text{w. p. } 1,$$

for all $j > 0$; again a similar expression holds for w_k.

8.6   Outline of Part II

This part of the thesis deals with identification of discrete-time bilinear state-space systems. Chapter 9 describes subspace identification methods. The derivation of these methods is based on the relation between the LPV system discussed in Part I and the bilinear system. The discussion heavily draws from the derivation of LPV subspace identification methods presented in Chapter 3. As with the LPV subspace identification methods, the bilinear identification methods suffer from the curse of dimensionality; the number of rows in the data matrices can be enormous. To circumvent this problem, the subset-selection procedure described in Chapter 4 can be applied to the bilinear subspace identification methods. An alternative way to circumvent the dimensionality problem is given in Chapter 10. There, it is shown that if the bilinear system is driven by a white-noise sequence, the properties of this white-noise sequence can be exploited to obtain an approximate model of the bilinear system. This approximate model can be further improved using optimization-based identification techniques. Such techniques for the bilinear system are presented in Chapter 11. In this chapter, first the projected gradient search method for the LPV system in Chapter 5 is adapted for the bilinear system; second, a new optimization-based procedure derived from the principle of separable least-squares is described. The performance of the different identification methods that are discussed, is compared in Chapter 12 using Monte-Carlo simulations. In addition, in this chapter, the identification methods are used to estimate a bilinear model of a high-purity distillation column. Finally, Chapter 13 contains the conclusions and some recommendations for further research.

Chapter 9

Subspace Identification

S ubspace identification is an attractive method for the estimation of multivariable state-space systems from input-output data. Subspace identification has already been discussed in this thesis in Chapter 3 for LPV systems. At the beginning of that chapter, a brief overview of subspace systems for linear and certain nonlinear systems has been given. It was mentioned that bilinear systems can also be identified with subspace methods. This chapter is concerned with subspace identification for bilinear systems.

The first subspace methods that were developed for the identification of bilinear systems were based on the assumption that the input to the system is a white-noise sequence. Favoreel, De Moor, and Van Overschee (1996, 1997a,b, 1999) derived such subspace methods from stochastic realization theory for bilinear systems described by Desai (1986). These methods can handle white process and measurement noise in a consistent way. A different approach was taken by Verdult et al. (1998a,b). They developed a method for white-noise inputs based on modeling part of the unknown state by the output measurements and approximating the bilinear system as a linear system with an extended input. This method is described in more detail in Chapter 10.

Although white-noise inputs provide a good excitation of the dynamical system, in several applications the input cannot be taken equal to a white-noise sequence; it might be that only sum-of-sine inputs or step signals with a minimum duration are allowable or possible. Therefore, bilinear subspace identification methods that can deal with more general input signals are of interest. This chapter focuses on bilinear subspace methods for general inputs.

Favoreel and De Moor (1998) described a bilinear subspace method that deals with more general inputs (see also Favoreel 1999). However, they assumed a particular type of noise model. For more general noise conditions it will yield biased estimates. A more important drawback of the method is that the number of rows in the data matrices grows exponentially with the order of the system (recall that the same problem occurs in LPV subspace identification). To overcome the curse of dimensionality, Verdult and Verhaegen proposed to use approximate subspace identification methods to find an initial estimate of the model, and subsequently

refine this model using a nonlinear optimization-based prediction-error method. The initial approximation of the model can either be obtained by applying a bilinear subspace method developed for white-noise inputs (Verdult and Verhaegen 1999), or by using subset-selection techniques —like in Chapter 4— to reduce the dimension of the data matrices in the method of Favoreel (Verdult and Verhaegen 2001b). Chen and Maciejowski recently developed alternative bilinear subspace methods for general inputs; first, only for deterministic systems (Chen and Maciejowski 2000d), later also for combined deterministic and stochastic systems (Chen and Maciejowski 2000a,b,c; Chen, Maciejowski, and Cox 2001). Their methods deal with white process and measurement noise in a consistent way; this means that their noise model is more general than the one used by Favoreel. However, these methods also suffer from the curse of dimensionality.

This chapter describes subspace identification methods for bilinear systems subject to general inputs. These methods are based on approximating the state from input-output data, just like the LPV subspace methods. Once an approximation of the state is available, the system matrices follow by solving a least-squares problem. The general framework for LPV subspace identification developed in Chapter 3 can be used to derive bilinear subspace methods, because the bilinear system can be viewed as a special case of the LPV system (see Chapter 8). The results presented in this chapter draw heavily from the discussion on LPV methods in Chapter 3. Therefore, the readers are encouraged to read that chapter first.

This chapter starts with the derivation of the data equations for bilinear subspace identification in Section 9.1. In Section 9.2 several identification methods are derived based on the general framework in Chapter 3. In this section it is also shown how the bilinear subspace methods of Chen and Maciejowski (2000b) fit into this framework. Section 9.3 discusses subspace identification for the special innovations model used by Favoreel (1999).

9.1   Data Equations

The bilinear system under consideration is the system (8.1)–(8.2) with a white process noise w_k acting on the state equation, and a white measurement noise v_k acting on the output, that is,

$$
\begin{aligned}
x_{k+1} &= Ax_k + F(u_k \otimes x_k) + Bu_k + w_k, & (9.1) \\
y_k &= Cx_k + Du_k + v_k. & (9.2)
\end{aligned}
$$

The data equations for this bilinear system are derived by exploiting its relation with the LPV system (3.1)–(3.2); the time-varying parameter equals the input ($p_k = u_k$), and the system matrices are: $A_0 = A$, $B_0 = B$, and $A_i = F_i$, $B_i = 0$ for $i = 1, 2, \ldots, m$, with $F = [F_1, F_2, \ldots, F_m]$.

First, the data matrices have to be defined. The data matrix for the input signal is defined as follows:

$$
U_k \quad := \quad [u_k, u_{k+1}, \ldots, u_{k+N-1}] \in \mathbb{R}^{m \times N}, \tag{9.3}
$$

$$U_{j|j} \quad := \quad U_j \in \mathbb{R}^{m \times N}, \tag{9.4}$$

$$U_{k+j|j} \quad := \quad \begin{bmatrix} U_{k+j} \\ U_{k+j-1|j} \\ U_{k+j} \odot U_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((m+1)^{k+1}-1) \times N}. \tag{9.5}$$

Note that compared with the input data matrix for the LPV system (3.9) on page 28, the block row $P_{k+j} \odot U_{k+j} = U_{k+j} \odot U_{k+j}$ has been omitted; it can be removed, because $B_i = 0$ for $i = 1, 2, \ldots, m$.

The other data matrices for the bilinear system are also slightly different from the data matrices in Section 3.1; they are given by

$$X_k \quad := \quad [x_k, x_{k+1}, \ldots, x_{k+N-1}] \in \mathbb{R}^{n \times N}, \tag{9.6}$$

$$X_{j|j} \quad := \quad \begin{bmatrix} X_j \\ U_j \odot X_j \end{bmatrix} \in \mathbb{R}^{(m+1)n \times N}, \tag{9.7}$$

$$X_{k+j|j} \quad := \quad \begin{bmatrix} X_{k+j-1|j} \\ U_{k+j} \odot X_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((m+1)^{k+1})n \times N}, \tag{9.8}$$

$$W_k \quad := \quad [w_k, w_{k+1}, \ldots, w_{k+N-1}] \in \mathbb{R}^{n \times N}, \tag{9.9}$$

$$W_{j|j} \quad := \quad W_j \in \mathbb{R}^{n \times N}, \tag{9.10}$$

$$W_{k+j|j} \quad := \quad \begin{bmatrix} W_{k+j} \\ W_{k+j-1|j} \\ U_{k+j} \odot W_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((m+1)^{k+1}-1)n/m \times N}. \tag{9.11}$$

From Lemma 3.2 on page 28 the following result can be derived:

Corollary 9.1 *Given the matrices (9.3) and (9.6)–(9.8). The relation between X_j and $X_{k+j|j}$ is*

$$X_{k+j|j} = \begin{bmatrix} X_j \\ U_{k+j|j}^p \odot X_j \end{bmatrix}, \tag{9.12}$$

where

$$U_{j|j}^p \quad := \quad U_j \in \mathbb{R}^{m \times N},$$

$$U_{k+j|j}^p \quad := \quad \begin{bmatrix} U_{k+j-1|j}^p \\ U_{k+j}(1, :) \\ U_{k+j}(1, :) \odot U_{k+j-1|j}^p \\ \vdots \\ U_{k+j}(m, :) \\ U_{k+j}(m, :) \odot U_{k+j-1|j}^p \end{bmatrix} \in \mathbb{R}^{((m+1)^{k+1}-1) \times N},$$

and $U_k(i, :)$ denotes the ith row of the matrix U_k.

The superscript p in $U_{k+j|j}^p$ refers to the fact that $U_{k+j|j}^p$ equals $U_{k+j|j}$ up to a permutation of the rows.

The data equation for the state of a bilinear system follows from Lemma 3.3 on page 30.

Corollary 9.2 *Given the matrices (9.3)–(9.11). For the bilinear system (9.1)–(9.2) it holds that*

$$X_{k+j} = \Delta_k^x X_{k+j-1|j} + \Delta_k^u U_{k+j-1|j} + \Delta_k^w W_{k+j-1|j}, \qquad (9.13)$$

where

$$
\begin{aligned}
\Delta_1^x &:= [A, F], \\
\Delta_k^x &:= [A\Delta_{k-1}^x, F_1\Delta_{k-1}^x, \ldots, F_m\Delta_{k-1}^x], \\
\Delta_1^u &:= B, \\
\Delta_k^u &:= [\Delta_1^u, A\Delta_{k-1}^u, F_1\Delta_{k-1}^u, \ldots, F_m\Delta_{k-1}^u], \\
\Delta_1^w &:= I_n, \\
\Delta_k^w &:= [\Delta_1^w, A\Delta_{k-1}^w, F_1\Delta_{k-1}^w, \ldots, F_m\Delta_{k-1}^w].
\end{aligned}
$$

Lemma 3.4 on page 31 together with the data matrices (3.15)–(3.20) provide the data equation involving the output data matrix.

Corollary 9.3 *Given the matrices (9.3)–(9.11), and (3.15)–(3.20). For the bilinear system (9.1)–(9.2) it holds that*

$$\mathcal{Y}_{k+j|j} = H_k^x X_{k+j-1|j} + H_k^u U_{k+j-1|j} + H_k^w W_{k+j-1|j} + G_k^u U_{k+j} + \mathcal{V}_{k+j|j}, \quad (9.14)$$

where

$$
\begin{aligned}
H_1^x &:= \begin{bmatrix} CA & CF_1 & \cdots & CF_m \\ C & 0 & \cdots & 0 \end{bmatrix}, \\
H_k^x &:= \begin{bmatrix} CA\Delta_{k-1}^x & CF_1\Delta_{k-1}^x & \cdots & CF_m\Delta_{k-1}^x \\ H_{k-1}^x & 0 & \cdots & 0 \end{bmatrix}, \\
H_1^u &:= \begin{bmatrix} CB \\ D \end{bmatrix}, \\
H_k^u &:= \begin{bmatrix} CB & CA\Delta_{k-1}^u & CF_1\Delta_{k-1}^u & \cdots & CF_m\Delta_{k-1}^u \\ G_{k-1}^u & H_{k-1}^u & 0 & \cdots & 0 \end{bmatrix}, \\
H_1^w &:= \begin{bmatrix} C \\ 0 \end{bmatrix}, \\
H_k^w &:= \begin{bmatrix} C & CA\Delta_{k-1}^w & CF_1\Delta_{k-1}^w & \cdots & CF_m\Delta_{k-1}^w \\ 0 & H_{k-1}^w & 0 & \cdots & 0 \end{bmatrix}, \\
G_k^u &:= \begin{bmatrix} D \\ 0 \end{bmatrix},
\end{aligned}
$$

and Δ_k^x, Δ_k^u and Δ_k^w are as in Corollary 9.2.

Equation (9.14) combined with (9.12) yields

$$
\begin{aligned}
\mathcal{Y}_{k+j|j} = \ & \Gamma_k X_j + \widetilde{H}_k^x (U_{k+j-1|j}^p \odot X_j) + H_k^u U_{k+j-1|j} + G_k^u U_{k+j} \\
& + H_k^w W_{k+j-1|j} + \mathcal{V}_{k+j|j},
\end{aligned}
$$

with Γ_k given by (3.22) on page 33 and $\widetilde{H}_k^x$ given by (3.23). This equation has the same structure as the generalized data equation (3.25) on page 33. Because of

this similarity, the basic ideas underlying bilinear subspace identification are the same as the ideas underlying LPV subspace identification. Therefore, the results presented in Chapter 3 on LPV subspace identification can be adapted to suit the bilinear system.

Since for bilinear systems $p_k = u_k$, it is more natural to write the generalized data equation (3.32) on page 35 as

$$Y = \Phi_X Z_X + \Phi_{U \odot X} Z_{U \odot X} + \Phi_U U + \Phi_V Z_V + \Xi_X + \Xi_{U \odot X}, \qquad (9.15)$$

where obviously, $\Phi_{P \odot X}$ has been replaced by $\Phi_{U \odot X}$, $Z_{P \odot X}$ by $Z_{U \odot X}$, and $\Xi_{P \odot X}$ by $\Xi_{U \odot X}$.

9.2   Identification Methods

Subspace identification methods for bilinear systems can be derived directly from the methods for LPV systems discussed in Chapter 3. However, some subtle differences exist. This is illustrated below by deriving a two-block and a three-block method similar to the methods in Subsections 3.3.1 and 3.3.2. In addition, it is pointed out how the four-block methods of Chen and Maciejowski (2000a,b,c) fit into the general framework for bilinear and LPV subspace identification.

The methods described below can be implemented using RQ factorizations as discussed in Section 3.4. To deal with the enormous dimension of the data matrices, the subset-selection techniques in Chapter 4 can again be applied.

9.2.1   Two-Block Methods

A two-block method for the bilinear system (9.1)–(9.2) can be based on its LPV counterpart in Subsection 3.3.1. This directly leads to the following theorem:

Theorem 9.1 *Given the bounded-input, bounded-output stable system (9.1)–(9.2) with bounded sequences u_k, w_k, and v_k, the data equations of Corollaries 9.1, 9.2, and 9.3, and the generalized data equation (9.15):*

$$Y = \Phi_X Z_X + \Phi_{U \odot X} Z_{U \odot X} + \Phi_U U + \Phi_V Z_V + \Xi_X + \Xi_{U \odot X},$$

with the following substitutions:

$$
\begin{aligned}
Y &\leftarrow \mathcal{Y}_{2k|k}, \\
Z_X &\leftarrow U_{k-1|0}, \\
Z_{U \odot X} &\leftarrow U^p_{2k-1|k} \odot U_{k-1|0}, \\
U &\leftarrow \begin{bmatrix} U_{2k-1|k} \\ U_{2k} \end{bmatrix}, \\
Z_V &\leftarrow \begin{bmatrix} W_{2k-1|k} \\ \mathcal{V}_{2k|k} \\ W_{k-1|0} \\ U^p_{2k-1|k} \odot W_{k-1|0} \end{bmatrix}.
\end{aligned}
$$

Then

$$\lim_{N \to \infty} \frac{1}{N} Z_V \Omega^T = 0 \quad with \quad \Omega := \begin{bmatrix} Z_X \\ Z_{U \odot X} \\ U \end{bmatrix}$$

holds with probability one, and the approximation errors are

$$\begin{aligned}
\Xi_X &\leftarrow \Gamma_k \Delta_k^x X_{k-1|0}, \\
\Xi_{U \odot X} &\leftarrow (\widetilde{H}_k^x \otimes \Delta_k^x)(U_{2k-1|k}^p \odot X_{k-1|0}),
\end{aligned}$$

with Γ_k given by (3.22) and $\widetilde{H}_k^x$ given by (3.23).
  If the matrix Ω has full row rank, the matrix

$$\Phi := \begin{bmatrix} \Phi_X & \Phi_{U \odot X} & \Phi_U \end{bmatrix}$$

can be estimated as

$$\widehat{\Phi} = Y \Omega^T \left(\Omega \Omega^T \right)^{-1},$$

and $(\widehat{\Phi} - \Phi)\Omega$ does not depend on the noise term Z_V with probability one for $N \to \infty$.

9.2.2 Three-Block Methods

The LPV three-block method in Subsection 3.3.2 implies that a three-block method for the bilinear system can be obtained by making the following substitutions:

$$\begin{aligned}
Y &\leftarrow \mathcal{Y}_{3k|2k}, \\
Z_{A,IV} &\leftarrow \begin{bmatrix} Y_k \\ U_k \\ U_{k-1|0} \end{bmatrix}, \\
Z_{X,R} &\leftarrow Y_{2k}, \\
Z_{X,IV} &\leftarrow \begin{bmatrix} U_{2k} \\ U_{2k-1|k} \end{bmatrix}, \\
Z_{U \odot X} &\leftarrow \begin{bmatrix} U_{3k-1|2k}^p \odot Y_{2k} \\ U_{3k-1|2k}^p \odot U_{2k} \\ U_{3k-1|2k}^p \odot U_{2k-1|k} \end{bmatrix}, \\
U &\leftarrow \begin{bmatrix} U_{3k-1|2k} \\ U_{3k} \end{bmatrix}.
\end{aligned}$$

However, to be able to use Identification Procedure II on page 55 the matrix

$$Z_{IV} = \begin{bmatrix} Z_{A,IV} \\ Z_{X,IV} \\ Z_{U \odot X} \\ U \end{bmatrix}$$

must have full row rank. With the substitutions given above, this matrix never has full row rank, because the matrix $U_{3k-1|2k}^p \odot U_{2k}$ contains duplicate rows, and

because the matrices $U^p_{3k-1|2k} \odot U_{2k}$ and $U_{3k-1|2k}$ have some rows in common. Chen and Maciejowski (2000d) were the first to recognize and solve this problem. The solution is simply to remove the double rows, and use different substitutions for the matrices $Z_{U \odot X}$ and U. Because the overlap between $Z_{U \odot X}$ and U has to be removed, it is more convenient to treat them as one matrix instead of two separate ones. By merging $Z_{U \odot X}$ and U, and removing the duplicate rows, the matrices $\Phi_{U \odot X}$ and Φ_U in the generalized data equation (9.15) are also changed; however, this does not affect the state estimation procedure, since that is based on an estimate of the row space of the matrix $\Phi_X Z_X$.

To remove the duplicate rows, the following data matrices are introduced:

$$U^r_{j|j} := \begin{bmatrix} U_j(1,:) \odot U_j(1:m,:) \\ U_j(2,:) \odot U_j(2:m,:) \\ \vdots \\ U_j(m,:) \odot U_j(m,:) \end{bmatrix} \in \mathbb{R}^{(m+1)m/2 \times N}, \qquad (9.16)$$

$$U^r_{k+j|j} := \begin{bmatrix} U^r_{k+j-1|j} \\ U_{k+j} \odot U^r_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((m+1)^{k+1})m/2 \times N}. \qquad (9.17)$$

The matrix $U^r_{k+j|j}$ contains only those rows of $U_{k+j|j} \odot U_j$ that do not occur in $U_{k+j|j}$.

The next lemma shows which rows can be removed from the matrix

$$\begin{bmatrix} U_{k+j|j} \odot U_j \\ U_{k+j|j} \end{bmatrix}$$

without changing its row space.

Lemma 9.1 *Given the matrices (9.3)–(9.5), (9.16), and (9.17), it holds that*

$$\mathrm{range}\left(\begin{bmatrix} U_{k+j|j} \odot U_j \\ U_{k+j|j} \end{bmatrix}^T \right) = \mathrm{range}\left(\begin{bmatrix} U^r_{k+j|j} \\ U_{k+j|j} \end{bmatrix}^T \right).$$

Proof: The proof follows an induction argument. For $k = 0$, it is easy to verify that (9.1) holds. Now, suppose that (9.1) holds for $k = i$, then

$$\mathrm{range}\left(\begin{bmatrix} U_{i+1+j|j} \odot U_j \\ U_{i+1+j|j} \end{bmatrix}^T \right) = \mathrm{range}\left(\begin{bmatrix} U_{i+j|j} \odot U_j \\ U_{i+j|j} \\ U_{i+1+j} \odot \begin{bmatrix} U_{i+j|j} \odot U_j \\ U_{i+j|j} \end{bmatrix} \\ U_{i+1+j} \odot U_j \\ U_{i+1+j} \end{bmatrix}^T \right)$$

$$= \mathrm{range}\left(\begin{bmatrix} U^r_{i+j|j} \\ U_{i+j|j} \\ U_{i+1+j} \odot \begin{bmatrix} U^r_{i+j|j} \\ U_{i+j|j} \end{bmatrix} \\ U_{i+1+j} \end{bmatrix}^T \right)$$

$$= \mathrm{range}\left(\begin{bmatrix} U^r_{i+1+j} \\ U_{i+1+j|j} \end{bmatrix}^T \right),$$

since $\mathrm{range}(U_{i+1+j} \odot U_j)$ is contained in $\mathrm{range}(U_{i+1+j} \odot U_{i+j|j})$; thus (9.1) also holds for $k = i + 1$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

This result allows the formulation of the three-block method for bilinear systems.

Theorem 9.2 *Given the bounded-input, bounded-output stable system (9.1)–(9.2) with bounded sequences u_k, w_k, and v_k; the data equations of Corollaries 9.1, 9.2, and 9.3; the data matrices (9.16) and (9.17); and the generalized data equation (9.15):*

$$Y = \Phi_X Z_X + \Phi_{U \odot X} Z_{U \odot X} + \Phi_U U + \Phi_V Z_V + \Xi_X + \Xi_{U \odot X},$$

with

$$Z_X := \begin{bmatrix} Z_{X,R} \\ Z_{X,IV} \end{bmatrix},$$

and with the following substitutions:

$$
\begin{aligned}
Y &\leftarrow \mathcal{Y}_{3k|2k}, \\
Z_{A,IV} &\leftarrow \begin{bmatrix} Y_k \\ U_k \\ U_{k-1|0} \end{bmatrix}, \\
Z_{X,R} &\leftarrow Y_{2k}, \\
Z_{X,IV} &\leftarrow \begin{bmatrix} U_{2k} \\ U_{2k-1|k} \end{bmatrix}, \\
\begin{bmatrix} Z_{U \odot X} \\ U \end{bmatrix} &\leftarrow \begin{bmatrix} U^p_{3k-1|2k} \odot Y_{2k} \\ U^p_{3k-1|2k} \odot U_{2k-1|k} \\ U^r_{3k-1|2k} \\ U_{3k-1|2k} \\ U_{3k} \end{bmatrix}, \\
Z_V &\leftarrow \begin{bmatrix} W_{3k-1|2k} \\ \mathcal{V}_{3k|2k} \\ W_{2k-1|k} \\ V_{2k} \\ U^p_{3k-1|2k} \odot W_{2k-1|k} \\ U^p_{3k-1|2k} \odot V_{2k} \end{bmatrix}.
\end{aligned}
$$

Then

$$\lim_{N \to \infty} \frac{1}{N} Z_V Z_{IV}^T = 0 \quad \text{with} \quad Z_{IV} := \begin{bmatrix} Z_{A,IV} \\ Z_{X,IV} \\ Z_{U \odot X} \\ U \end{bmatrix}$$

holds with probability one, and the approximation errors are

$$
\begin{aligned}
\Xi_X &\leftarrow \Gamma_k (I - MC) \Delta_k^x X_{2k-1|k}, \\
\Xi_{U \odot X} &\leftarrow \left(\widetilde{H}_k^x \otimes (I - MC) \Delta_k^x \right) (U^p_{3k-1|2k} \odot X_{2k-1|k}),
\end{aligned}
$$

with Γ_k given by (3.22) and $\widetilde{H}_k^x$ given by (3.23).

If the matrix ΩZ_{IV}^T has full row rank, the matrix

$$\Phi := \begin{bmatrix} \Phi_X & \Phi_{U \odot X} & \Phi_U \end{bmatrix}$$

can be estimated as

$$\widehat{\Phi} = Y Z_{IV}^T Z_{IV} \Omega^T \left(\Omega Z_{IV}^T Z_{IV} \Omega^T \right)^{-1},$$

and $(\widehat{\Phi} - \Phi)\Omega$ does not depend on the noise term Z_V with probability one for $N \to \infty$.

Two-stage three-block methods similar to the ones in Subsections 3.3.3 and 3.3.4 can also be derived for bilinear subspace methods. The derivation is rather straightforward and will be omitted here.

9.2.3 Four-Block Methods

Chen and Maciejowski (2000a,b,c) have developed bilinear subspace identification methods that are based on four sets of data matrices. This section shows how these four-block methods fit into the general framework in Chapter 3. The four-block methods are instrumental variable methods that use two set of data matrices to construct instrumental variables. The data matrices starting at 0 and at k, where k is the block size, are used to construct the instrumental variables. The data matrices starting at k and $2k$ are used to approximate the state. The data matrices starting at $3k$ are used to identify the system.

These methods use a different data matrix to store the output signals. This matrix is defined by

$$
\begin{align}
Y_k &:= [y_k, y_{k+1}, \ldots, y_{k+N-1}] \in \mathbb{R}^{\ell \times N}, \tag{9.18}\\
Y_{j|j} &:= Y_j \in \mathbb{R}^{\ell \times N}, \tag{9.19}\\
Y_{k+j|j} &:= \begin{bmatrix} Y_{k+j} \\ Y_{k+j-1|j} \\ U_{k+j} \odot Y_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((m+1)^{k+1}-1)\ell/m \times N}. \tag{9.20}
\end{align}
$$

The following lemma presents the data equation that expresses such an output data matrix as a linear combination of data matrices constructed from the state, the input and the disturbances. Note that the later data matrices are constructed in the same way as the data matrices for the two-block and three-block method discussed above.

Lemma 9.2 *Given the matrices (9.3)–(9.11), and (9.18)–(9.20). For the bilinear system (9.1)–(9.2) it holds that*

$$Y_{k+j-1|j} = \mathcal{L}_k^x X_{k+j-1|j} + \mathcal{L}_k^u U_{k+j-1|j} + \mathcal{L}_k^w W_{k+j-1|j} + \mathcal{L}_k^v V_{k+j-1|j},$$

where

$$\mathcal{L}_1^x := \begin{bmatrix} C & 0 \end{bmatrix},$$

$$\mathcal{L}_k^x := \begin{bmatrix} C\Delta_{k-1}^x & 0 & \cdots & 0 \\ \mathcal{L}_{k-1}^x & 0 & \cdots & 0 \\ 0 & \mathcal{L}_{k-1}^x & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{L}_{k-1}^x \end{bmatrix},$$

$$\mathcal{L}_1^u := D,$$

$$\mathcal{L}_k^u := \begin{bmatrix} D & C\Delta_{k-1}^u & 0 & \cdots & 0 \\ 0 & \mathcal{L}_{k-1}^u & 0 & \cdots & 0 \\ 0 & 0 & \mathcal{L}_{k-1}^u & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathcal{L}_{k-1}^u \end{bmatrix},$$

$$\mathcal{L}_1^w := 0_{\ell \times n},$$

$$\mathcal{L}_k^w := \begin{bmatrix} 0 & C\Delta_{k-1}^w & 0 & \cdots & 0 \\ 0 & \mathcal{L}_{k-1}^w & 0 & \cdots & 0 \\ 0 & 0 & \mathcal{L}_{k-1}^w & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathcal{L}_{k-1}^w \end{bmatrix},$$

$$\mathcal{L}_1^v := I_\ell,$$

$$\mathcal{L}_k^v := \begin{bmatrix} I_\ell & 0 & 0 & \cdots & 0 \\ 0 & \mathcal{L}_{k-1}^v & 0 & \cdots & 0 \\ 0 & 0 & \mathcal{L}_{k-1}^v & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathcal{L}_{k-1}^v \end{bmatrix},$$

and Δ_k^x, Δ_k^u, and Δ_k^w are as in Corollary 9.2.

This lemma has been proved by Favoreel, De Moor, and Van Overschee (1996) and by Chen and Maciejowski (2000b). The proof is based on an induction argument similar to the one used in the proof of Lemma 3.4.

If the matrix $X_{k+j-1|j}$ in the output equation of Lemma 9.2 is replaced by equation (9.12) on page 129, the resulting output equation has the same structure as the generalized data equation (9.15). Therefore, the basic ideas in Section 3.2 can again be applied.

The four-block methods of Chen and Maciejowski (2000a,b,c) are based on Lemma 9.2. They are two-stage methods. The two-stage state approximation is used to determine the matrix Z_X; the matrix $Z_{U \odot X}$ is based on the approximation of the state by

$$X_{k+j} \approx M(Y_{k+j} - DU_{k+j}) + (I - MC)\Delta_k^u U_{k+j-1|j}.$$

The choices of $M = 0$ and $M \neq 0$ in this equation lead to two different methods, similar to the two-stage methods described in Subsections 3.3.3 and 3.3.4.

The method that corresponds to the choice of $M \neq 0$, consists of making the following substitutions in the generalized data equation; it will be called the *four-block method*

$$Y \quad \leftarrow \quad Y_{4k-1|3k},$$

$$Z_{A,IV} \quad \leftarrow \quad \begin{bmatrix} Y_{2k-1|k} \\ U_{2k-1|k} \\ U^r_{2k-1|k} \\ U^p_{2k-1|k} \odot Y_k \\ U^p_{2k-1|k} \odot U_{k-1|0} \end{bmatrix},$$

$$Z_{X,R} \quad \leftarrow \quad Y_{3k-1|2k},$$

$$Z_{X,IV} \quad \leftarrow \quad \begin{bmatrix} U_{3k-1|2k} \\ U^r_{3k-1|2k} \\ U^p_{3k-1|2k} \odot Y_{2k} \\ U^p_{3k-1|2k} \odot U_{2k-1|k} \end{bmatrix},$$

$$\begin{bmatrix} Z_{U \odot X} \\ U \end{bmatrix} \quad \leftarrow \quad \begin{bmatrix} U_{4k-1|3k} \\ U^r_{4k-1|3k} \\ U^p_{4k-1|3k} \odot Y_{3k} \\ U^p_{4k-1|3k} \odot U_{3k-1|2k} \end{bmatrix}.$$

The method that corresponds to the choice of $M = 0$ will be called the *four-block method with reduced dimensions*; it consists of making the following substitutions in the generalized data equation:

$$Y \quad \leftarrow \quad Y_{4k-1|3k},$$

$$Z_{A,IV} \quad \leftarrow \quad \begin{bmatrix} Y_{2k-1|k} \\ U_{2k-1|k} \\ U^p_{2k-1|k} \odot U_{k-1|0} \end{bmatrix},$$

$$Z_{X,R} \quad \leftarrow \quad Y_{3k-1|2k},$$

$$Z_{X,IV} \quad \leftarrow \quad \begin{bmatrix} U_{3k-1|2k} \\ U^p_{3k-1|2k} \odot U_{2k-1|k} \end{bmatrix},$$

$$\begin{bmatrix} Z_{U \odot X} \\ U \end{bmatrix} \quad \leftarrow \quad \begin{bmatrix} U_{4k-1|3k} \\ U^p_{4k-1|3k} \odot U_{3k-1|2k} \end{bmatrix}.$$

Both these methods can be implemented by Identification Procedure II on page 55. Note that duplicate rows of $Z_{U \odot X}$ and U have been removed using the result of Lemma 9.1.

The four-block methods of Chen and Maciejowski (2000b) are based on the substitutions described above, but instead of Identification Procedure II, a slightly different procedure to obtain the state sequence is used. Recall from Subsection 3.2.2 that the state approximation is based on an SVD of the product ΓX. Instead of approximating this product as

$$\Gamma X \approx \Phi_X Z_X,$$

it can also be approximated as

$$\Gamma X \approx Y - (\Phi_{U \odot X} Z_{U \odot X} + \Phi_U U).$$

This later idea is exploited by Chen and Maciejowski. They estimate $\Phi_{U \odot X}$ and Φ_U, instead of Φ_X. Their method is based on combining equation (3.26) on page 34 with (9.15):

$$Y = \Gamma X + \Phi_{U \odot X} Z_{U \odot X} + \Phi_U U + \Phi_V Z_V - V_X + \Xi_{U \odot X}.$$

Multiplying both sides with the instrumental variable matrix

$$Z_{IV} = \begin{bmatrix} Z_{A,IV} \\ Z_{X,IV} \\ Z_{P \odot X} \\ U \end{bmatrix}$$

yields

$$\lim_{N \to \infty} \frac{1}{N} Y Z_{IV}^T = \lim_{N \to \infty} \frac{1}{N} \left(\Gamma X Z_{IV}^T + \Phi_{U \odot X} Z_{U \odot X} Z_{IV}^T + \Phi_U U Z_{IV}^T + \Xi_{U \odot X} Z_{IV}^T \right).$$

With the assumption that the matrix

$$\Omega = \begin{bmatrix} Z_X \\ Z_{U \odot X} \\ U \end{bmatrix}$$

has full row rank, the term $\Gamma X Z_{IV}^T$ can be removed by performing a projection on the orthogonal complement of its row space; the RQ factorization

$$Z_X Z_{IV}^T = \begin{bmatrix} \overline{R} & 0 \end{bmatrix} \begin{bmatrix} \overline{Q}_1 \\ \overline{Q}_2 \end{bmatrix} \tag{9.21}$$

can be used to obtain

$$\lim_{N \to \infty} \frac{1}{N} Y Z_{IV}^T \overline{Q}_2^T = \lim_{N \to \infty} \frac{1}{N} \left(\Phi_{U \odot X} Z_{U \odot X} Z_{IV}^T \overline{Q}_2^T + \Phi_U U Z_{IV}^T \overline{Q}_2^T + \Xi_{U \odot X} Z_{IV}^T \overline{Q}_2^T \right).$$

From which the matrices $\Phi_{U \odot X}$ and Φ_U are estimated as

$$\begin{bmatrix} \widehat{\Phi}_{U \odot X} & \widehat{\Phi}_U \end{bmatrix} = Y Z_{IV}^T \overline{Q}_2^T \left(\begin{bmatrix} Z_{U \odot X} \\ U \end{bmatrix} Z_{IV}^T \overline{Q}_2^T \right)^\dagger. \tag{9.22}$$

The products

$$Z_X Z_{IV}^T, \quad Y Z_{IV}^T, \quad \begin{bmatrix} Z_{U \odot X} \\ U \end{bmatrix} Z_{IV}^T, \tag{9.23}$$

can be computed efficiently with the RQ factorization of Step 1 of Identification Procedure II on page 55.

The procedure described above is computationally less demanding than Identification Procedure II. The reduced computational complexity is due to the fact

that the computation of the pseudo inverse of Step 2 of Identification Procedure II is replaced by the RQ factorization (9.21) and the computation of a pseudo inverse of a smaller dimensional matrix (9.22). This will be illustrated for the four-block method with reduced dimensions that was described above. For simplicity it is assumed that the system under consideration is square ($\ell = m$).

First, the number of floating point operations (flops) involved in the RQ factorization (9.21) and the computation of the pseudo inverse in (9.22) is derived. This gives a good estimate of the total number of flops involved; the operations to form the matrix products (9.23) can be neglected. Computing the R factor of a p by q matrix ($p < q$) by a Householder RQ factorization requires $2(qp^2 - p^3/3)$ flops; computing the Q factor by backward accumulation requires $4(q^2p - qp^2 + p^3/3)$ flops (Golub and Van Loan 1996, p. 225). Thus, the RQ factorization of the $(d_k^2 + 2d_k)$ by $(3d_k^2 + 4d_k)$ matrix $Z_X Z_{IV}^T$ in (9.21) involves $(92/3)d_k^6 + 140d_k^5 + 208d_k^4 + (304/3)d_k^3$ flops, where $d_k = (m+1)^{k+1} - 1$. The pseudo inverse in (9.22) is computed using an SVD. An SVD of a p by q matrix ($p < q$) requires $6qp^2 + 20p^3$ floating point operations (Golub and Van Loan 1996, p. 254). The computation of this pseudo inverse of a $(d_k^2 + d_k)$ by $2(d_k^2 + d_k)$ matrix involves $32(d_k^2 + d_k)^3$ flops. Therefore, the total number of flops for the procedure outlined above is:

$$\frac{188}{3}d_k^6 + 236d_k^5 + 304d_k^4 + \frac{400}{3}d_k^3. \tag{9.24}$$

Next, the number of flops for Step 2 of Identification Procedure II is derived. This step involves the computation of the pseudo inverse of a $(2d_k^2 + 3d_k)$ by $(3d_k^2 + 4d_k)$ matrix; using an SVD it involves

$$6(3d_k^2 + 4d_k)(2d_k + 3d_k)^2 + 20(2d_k^2 + 3d_k)^3 = 232d_k^6 + 1032d_k^5 + 1530d_k^4 + 756d_k^3$$

flops. Comparison with (9.24) shows that for large d_k, Step 2 of Identification Procedure II requires much more flops than the procedure based on (9.21) and (9.22).

9.2.4 Discussion

Because the bilinear subspace identification methods are derived within the general framework for LPV subspace identification in Chapter 3, the discussion given in Subsection 3.3.5 is applicable. The identification methods differ in the way they approximate the state sequence, and this results in differences in accuracy and in the dimensions of the data matrices involved. The methods can be implemented by the RQ factorization as discussed in Section 3.4. For the methods discussed, the total number of rows in the RQ factorization is given below. In these expressions d_k is defined as $d_k := (m+1)^{k+1} - 1$.

- Two-block method (Subsection 9.2.1, Identification Procedure I):

$$d_k^2 + 2d_k + m + (k+1)\ell.$$

- Three-block method (Subsection 9.2.2, Identification Procedure II):

$$d_k^2 + d_k(\ell + 3) + (d_k + 1)\frac{m}{2} + 3m + +2\ell + (k+1)\ell.$$

Table 9.1: Total number of rows in the RQ factorization for the identification methods discussed in Subsections 9.2.1, 9.2.2, and 9.2.3, as a function of the block size k in the case that $m = \ell = 1$.

Identification method	Procedure	$k = 1$	$k = 2$	$k = 3$	$k = 4$
Two-block	I	20	70	264	1 034
Three-block	II	37	101	323	1 149
Four-block	II	78	264	924	3 396
Four-block red.	II	54	210	810	3 162

Table 9.2: Total number of rows in the RQ factorization for the identification methods discussed in Subsections 9.2.1, 9.2.2, and 9.2.3, as a function of the block size k in the case that $m = \ell = 2$.

Identification method	Procedure	$k = 1$	$k = 2$	$k = 3$	$k = 4$
Two-block	I	86	736	6 570	59 060
Three-block	II	127	849	6 899	60 037
Four-block	II	315	2 421	20 403	179 325
Four-block red.	II	240	2 184	19 680	177 144

- Four-block method (Subsection 9.2.3, Identification Procedure II):

$$3d_k^2 + 3d_k \left(\frac{\ell}{m} + \ell + 1 \right) + 3(d_k + 1) \frac{m}{2}.$$

- Four-block method with reduced dimensions (Subsection 9.2.3, Identification Procedure II):

$$3d_k^2 + 3d_k \left(\frac{\ell}{m} + 1 \right).$$

Tables 9.1 and 9.2 illustrate these expressions. Note that in the four-block methods the matrix Y is a data matrix that also includes products of time-lagged values of the output. As with the LPV identification methods, the number of rows in the RQ factorization grows very rapidly with the block size k. For large dimensions, alternative methods are desirable, like the subset-selection method described in Chapter 4.

9.3 Identification for Innovation-Type Noise Models

In Section 3.5 subspace identification methods for innovation-type LPV systems were derived. It is straightforward to use the results of this section, to derive subspace identification methods for bilinear systems in which the process noise

equals

$$w_k = \begin{bmatrix} K_0 & K_1 & \cdots & K_m \end{bmatrix} \begin{bmatrix} v_k \\ u_k \otimes v_k \end{bmatrix}.$$

These methods will not be derived here; instead, attention is focused on bilinear systems of the form

$$\begin{align}
x_{k+1} &= Ax_k + F(u_k \otimes x_k) + Bu_k + Kv_k \tag{9.25} \\
y_k &= Cx_k + Du_k + v_k, \tag{9.26}
\end{align}$$

where, with a slight abuse of notation, K is used instead of K_0. The reason for focusing on this particular type of bilinear system is that the bilinear subspace identification method of Favoreel (1999) deals with such a system. What follows is a summary of this method that clarifies how this method fits into the general framework presented in Chapter 3.

Note that the system (9.25)–(9.26) can easily be converted into a one-step-ahead predictor by substitution of $v_k = y_k - Cx_k - Du_k$ into (9.25):

$$\begin{align}
x_{k+1} &= (A - KC)x_k + F(u_k \otimes x_k) + (B - KD)u_k + Ky_k \\
y_k &= Cx_k + Du_k + v_k.
\end{align}$$

The data equations for the system (9.25)–(9.26) follow from Corollary 3.3 and Lemma 3.6 on page 59 and the definitions

$$Y_{j|j} := \begin{bmatrix} Y_j \\ U_j \odot Y_j \end{bmatrix} \in \mathbb{R}^{(m+1)\ell \times N}, \tag{9.27}$$

$$Y_{k+j|j} := \begin{bmatrix} Y_{k+j} \\ Y_{k+j-1|j} \\ U_{k+j} \odot Y_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((m+1)^{k+1}-1)\ell/m \times N}, \tag{9.28}$$

$$V_{j|j} := \begin{bmatrix} V_j \\ U_j \odot V_j \end{bmatrix} \in \mathbb{R}^{(m+1)\ell \times N}, \tag{9.29}$$

$$V_{k+j|j} := \begin{bmatrix} V_{k+j} \\ V_{k+j-1|j} \\ U_{k+j} \odot V_{k+j-1|j} \end{bmatrix} \in \mathbb{R}^{((m+1)^{k+1}-1)\ell/m \times N}. \tag{9.30}$$

The data equations are presented in the following two corollaries:

Corollary 9.4 *Given the matrices (3.15), (9.3)–(9.8), (9.27), and (9.28). For the bilinear system (9.25)–(9.26) it holds that*

$$X_{k+j} = \overline{\Delta}_k^x X_{k+j-1|j} + \overline{\Delta}_k^u U_{k+j-1|j} + \overline{\Delta}_k^y Y_{k+j-1|j}, \tag{9.31}$$

where

$$\begin{align}
\overline{\Delta}_1^x &:= [(A - KC), F_1, \ldots, F_m], \\
\overline{\Delta}_k^x &:= [(A - KC)\overline{\Delta}_{k-1}^x, F_1\overline{\Delta}_{k-1}^x, \ldots, F_m\overline{\Delta}_{k-1}^x], \\
\overline{\Delta}_1^u &:= B - KD, \\
\overline{\Delta}_k^u &:= [\overline{\Delta}_1^u, (A - KC)\overline{\Delta}_{k-1}^u, F_1\overline{\Delta}_{k-1}^u, \ldots, F_m\overline{\Delta}_{k-1}^u], \\
\overline{\Delta}_1^y &:= K, \\
\overline{\Delta}_k^y &:= [\overline{\Delta}_1^y, (A - KC)\overline{\Delta}_{k-1}^y, F_1\overline{\Delta}_{k-1}^y, \ldots, F_m\overline{\Delta}_{k-1}^y].
\end{align}$$

Corollary 9.5 *Given the matrices (3.15)–(3.18), (9.3)–(9.8), (9.29), and (9.30). For the bilinear system (9.25)–(9.26) it holds that*

$$\mathcal{Y}_{k+j|j} = H_k^x X_{k+j-1|j} + H_k^u U_{k+j-1|j} + H_k^v V_{k+j-1|j} + G_k^u U_{k+j} + G_k^v V_{k+j}, \quad (9.32)$$

where

$$
\begin{aligned}
H_1^v &:= \begin{bmatrix} CK \\ I_\ell \end{bmatrix}, \\
H_k^v &:= \begin{bmatrix} CK & CA\Delta_{k-1}^v & CF_1\Delta_{k-1}^v & \cdots & CF_s\Delta_{k-1}^e \\ G_{k-1}^v & H_{k-1}^v & 0 & \cdots & 0 \end{bmatrix}, \\
G_k^v &:= \begin{bmatrix} I_\ell \\ 0 \end{bmatrix}, \\
\Delta_1^v &:= K, \\
\Delta_k^v &:= [\Delta_1^v, A\Delta_{k-1}^v, F_1\Delta_{k-1}^v, \dots, F_m\Delta_{k-1}^v],
\end{aligned}
$$

and H_k^x, H_k^u, and G_k^u are as in Corollary 9.3.

Based on Corollary 9.4, the state sequence can be approximated as

$$X_{k+j} \approx \overline{\Delta}_k^u U_{k+j-1|j} + \overline{\Delta}_k^y Y_{k+j-1|j}.$$

From this approximation a two-block method can be derived similar to the one presented in Subsection 3.5.2.

Theorem 9.3 *Given the bounded-input, bounded-output stable system (9.25)–(9.26) with bounded sequences u_k and v_k, the data equations of Corollaries 9.1, 9.4, and 9.5, and the generalized data equation (9.15):*

$$Y = \Phi_X Z_X + \Phi_{U \odot X} Z_{U \odot X} + \Phi_U U + \Phi_V Z_V + \Xi_X + \Xi_{U \odot X},$$

with the following substitutions:

$$
\begin{aligned}
Y &\leftarrow \mathcal{Y}_{2k|k}, \\
Z_X &\leftarrow \begin{bmatrix} U_{k-1|0} \\ Y_{k-1|0} \end{bmatrix}, \\
Z_{U \odot X} &\leftarrow U_{2k-1|k}^p \odot \begin{bmatrix} U_{k-1|0} \\ Y_{k-1|0} \end{bmatrix}, \\
U &\leftarrow \begin{bmatrix} U_{2k-1|k} \\ U_{2k} \end{bmatrix}, \\
Z_V &\leftarrow \begin{bmatrix} V_{2k-1|k} \\ V_{2k} \end{bmatrix}.
\end{aligned}
$$

Then

$$\lim_{N \to \infty} \frac{1}{N} Z_V \Omega^T = 0 \quad \text{with} \quad \Omega := \begin{bmatrix} Z_X \\ Z_{U \odot X} \\ U \end{bmatrix}$$

Table 9.3: Total number of rows in the RQ factorization for the two-block identification method described in Section 9.3 as a function of the block size k.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$m = \ell = 1$	44	182	744	3 018
$m = \ell = 2$	158	1 438	13 050	117 866

holds with probability one, and the approximation errors are

$$\Xi_X \quad \leftarrow \quad \Gamma_k \overline{\Delta}_k^x X_{k-1|0},$$
$$\Xi_{U \odot X} \quad \leftarrow \quad (\widetilde{H}_k^x \otimes \overline{\Delta}_k^x)(U_{2k-1|k}^p \odot X_{k-1|0}),$$

with Γ_k given by (3.22) and $\widetilde{H}_k^x$ given by (3.23).

If the matrix Ω has full row rank, the matrix

$$\Phi := \begin{bmatrix} \Phi_X & \Phi_{U \odot X} & \Phi_U \end{bmatrix}$$

can be estimated as

$$\widehat{\Phi} = Y\Omega^T \left(\Omega\Omega^T\right)^{-1},$$

and $(\widehat{\Phi} - \Phi)\Omega$ does not depend on the noise term Z_V with probability one for $N \to \infty$.

This theorem shows that a two-block subspace identification method for the bilinear system (9.25)–(9.26) can be based on Identification Procedure I on page 53. This subspace method is basically the one described by Favoreel (1999). The only difference is that Favoreel uses the matrix $U_{2k|k}$ instead of

$$\begin{bmatrix} U_{2k-1|k} \\ U_{2k} \end{bmatrix}, \tag{9.33}$$

for the data matrix U. This unnecessarily increases the number of rows, since terms like $U_{2k} \odot U_{2k-1|k}$, which are not needed, are included in $U_{2k|k}$.

For the subspace method described above, the total number of rows in the RQ factorization of Identification Procedure I equals

$$(d_k^2 + d_k)\left(\frac{\ell}{m} + 1\right) + d_k + m + (k+1)\ell.$$

Table 9.3 illustrates this expression, and can be used for comparison with the other identification methods in Tables 9.1 and 9.2 on page 140.

Chen and Maciejowski (2000b) argue that their four-block methods have a reduced computational complexity compared to the two-block method of Favoreel (1999). At first sight, this seems strange, because the number of rows in the RQ factorization is much larger for the four-block method than for the two-block method. However, the total number of floating point operations is dominated by the computation of the pseudo inverse that is needed to compute the state sequence. Chen and Maciejowski (2000b) use the procedure discussed at the end of Subsection 9.2.3. The pseudo inverse in this procedure requires less floating point operations than the inverse used in the two-block method.

Chapter 10

Subspace Identification for White-Noise Input Signals

Subspace identification for bilinear systems as discussed in the previous chapter suffers from the curse of dimensionality: the dimensions of the data matrices involved grow exponentially with the order of the system. To obtain subspace algorithms that can be used for practical applications, the dimensions need to be reduced, and consequently an approximation has to be made. One way to reduce the dimensions is to use the subset-selection technique discussed in Chapter 4. Although that technique has been described for LPV subspace identification, it can also be applied to bilinear subspace identification, because of the similarities between these two subspace methods (see Section 9.1).

This chapter discusses an alternative way to overcome the curse of dimensionality in bilinear subspace identification. It is assumed that the input driving the bilinear system is a zero-mean white-noise signal. The term 'white noise' means that the vector input signal consists of statistically independent scalar sequences that are independently identically distributed random variables in time. The special properties of such an input signal can be exploited to obtain consistent estimates of the matrices A and C of the bilinear system (9.1)–(9.2), up to a similarity transformation. The main idea to estimate A and C is to approximate the bilinear system (9.1)–(9.2) by a linear system with an extended input, and use an errors-in-variables subspace algorithm that deals with the approximation errors. The method is therefore called the *input-extension method*. The white-noise assumption on the input is crucial for the success of this method. The remaining system matrices B, D, and F can be estimated by solving some linear least-squares problems. However, the estimates of these matrices are biased. Therefore, it is proposed to improve the bilinear model using nonlinear optimization-based identification techniques discussed in Chapter 11. Note that to successfully use such an optimization-based method, a good starting point is of paramount importance; the method presented in this chapter provides such a starting point.

The input-extension method for bilinear systems can be modified to deal with systems that have some quadratic terms in addition to the bilinear term in the state

equation. Such an extension was described by Schrempf and Del Re (2001a,b) who call this type of systems *extended bilinear systems*.

This chapter is organized as follows: In Section 10.1 the main idea of extending the input is described. Section 10.2 contains the subspace algorithm for estimating the matrices A and C. Section 10.3 is devoted to the estimation of the matrices B, D, and F.

10.1   Extending the Input

The nonlinearity of the bilinear system (9.1)–(9.2) is due to the term $u_k \otimes x_k$. The main difficulty in identifying the bilinear system stems from the fact that x_k is unknown. By extending the input of the bilinear system, part of the term $u_k \otimes x_k$ can be described by input and output data. Without loss of generality it is assumed that the matrix C of the bilinear system has full row rank, and thus $\ell \leq n$; this simply means that there are no redundant output measurements. The main idea is to decompose each matrix F_i as

$$F_i = F_i^{(1)}C + F_i^{(2)}\widetilde{C}, \tag{10.1}$$

with $\widetilde{C} \in \mathbb{R}^{(n-\ell) \times n}$ such that the matrix $[C^T \ \widetilde{C}^T]$ has full rank. Each row of the matrix F_i is written as a linear combination of the rows of the matrices C and $\widetilde{C}$. The decomposition (10.1) can be used to write equation (9.1) as

$$\begin{aligned}
x_{k+1} &= Ax_k + \begin{bmatrix} B & F^{(1)} & -F_D^{(1)} \end{bmatrix} \begin{bmatrix} u_k \\ u_k \otimes y_k \\ u_k \otimes u_k \end{bmatrix} \\
&\quad + F_{\widetilde{C}}^{(2)}(u_k \otimes x_k) - F^{(1)}(u_k \otimes v_k) + w_k,
\end{aligned} \tag{10.2}$$

where

$$\begin{aligned}
F^{(1)} &:= \begin{bmatrix} F_1^{(1)} & F_2^{(1)} & \cdots & F_m^{(1)} \end{bmatrix}, \\
F_D^{(1)} &:= \begin{bmatrix} F_1^{(1)}D & F_2^{(1)}D & \cdots & F_m^{(1)}D \end{bmatrix}, \\
F_{\widetilde{C}}^{(2)} &:= \begin{bmatrix} F_1^{(2)}\widetilde{C} & F_2^{(2)}\widetilde{C} & \cdots & F_m^{(2)}\widetilde{C} \end{bmatrix}.
\end{aligned}$$

By introducing the extended input signal

$$\widetilde{u}_k := \begin{bmatrix} u_k \\ u_k \otimes y_k \\ u_k \otimes u_k \end{bmatrix},$$

and the matrix

$$\widetilde{B} := \begin{bmatrix} B & F^{(1)} & -F_D^{(1)} \end{bmatrix},$$

equation (10.2) becomes

$$x_{k+1} = Ax_k + \widetilde{B}\widetilde{u}_k + F_{\widetilde{C}}^{(2)}(u_k \otimes x_k) - F^{(1)}(u_k \otimes v_k) + w_k. \tag{10.3}$$

If the bilinear system has an F matrix such that $F_{\widetilde{C}}^{(2)} = 0$, the term $u_k \otimes x_k$ vanishes, and this equation equals the state equation of a linear system with an extended input $\widetilde{u}_k$, and with a process noise that depends on the input due to the term $u_k \otimes v_k$. This results in correlation between the process noise and the input. The bilinear system with $F_{\widetilde{C}}^{(2)} = 0$ can be identified using linear identification techniques that can deal with this correlation. If $F_{\widetilde{C}}^{(2)} \neq 0$, the bilinear system can be approximated by a linear system with an extended input $\widetilde{u}_k$. The error that is introduced by neglecting the bilinear term that corresponds to $F_{\widetilde{C}}^{(2)}$ should be dealt with appropriately. First, this error is analyzed.

The output associated with (10.3) can be written as

$$
\begin{aligned}
y_k &= CA^{k-j}x_j + C\sum_{\tau=j}^{k-1} A^{k-\tau-1}\Big(\widetilde{B}\widetilde{u}_\tau - F^{(1)}(u_\tau \otimes v_\tau) + w_\tau\Big) \\
&\quad + Du_k + v_k + \xi_k,
\end{aligned} \tag{10.4}
$$

where $j \leq k$ and ξ_k is an error due to neglecting the term $F_{\widetilde{C}}^{(2)}(u_k \otimes x_k)$. This error can be expressed as

$$
\xi_k := C\mathcal{P}(u_j, \ldots, u_{k-1})x_j + C\sum_{\tau=j}^{k-1} \mathcal{P}(u_\tau, \ldots, u_{k-1})\Big(\widetilde{B}\widetilde{u}_\tau - F^{(1)}(u_\tau \otimes v_\tau) + w_\tau\Big),
$$

where the term $\mathcal{P}(u_j, \ldots, u_{k-1})$ is used to denote a linear combination of the entries of $u_j, \ldots, u_{k-1}$ and products of these entries.

Theorem 10.1 *If the bilinear system (9.1)–(9.2) is driven by a zero-mean white-noise input signal, the error term ξ_k in equation (10.4) has a zero mean and satisfies*

$$
\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \xi_{k+s}u_k^T = 0, \quad \text{w. p. 1,} \quad \text{for } s \geq 0, \tag{10.5}
$$

$$
\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \xi_{k+s}y_k^T = 0, \quad \text{w. p. 1,} \quad \text{for } s > 0. \tag{10.6}
$$

Proof: Without loss of generality, ξ_k can be written as

$$
\xi_k := C\mathcal{P}(u_{k-1})x_{k-1} = CF_{\widetilde{C}}^{(2)}(u_{k-1} \otimes x_{k-1}).
$$

Because u_k is a zero-mean sequence, ξ_k is also a zero-mean sequence. Because u_k is a white-noise sequence,

$$
\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} x_{k+s}u_k^T = 0, \quad \text{w. p. 1,} \quad \text{for } s \leq 0,
$$

$$
\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} x_{k+s}u_k^T \neq 0, \quad \text{w. p. 1,} \quad \text{for } s > 0,
$$

and therefore,

$$\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} (u_{k+s-1} \otimes x_{k+s-1}) u_k^T \;=\; 0, \quad \text{w. p. 1}, \quad \text{for } s \geq 0,$$

$$\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} (u_{k+s-1} \otimes x_{k+s-1}) x_k^T \;=\; 0, \quad \text{w. p. 1}, \quad \text{for } s > 0.$$

Equation (10.6) easily follows from $y_k = Cx_k + Du_k + v_k$. □

10.2   Estimating the Matrices A and C

The 'linear' system (10.3), (10.4) of the previous section can be identified with a subspace identification method for linear time-invariant systems. The MOESP subspace method of Chou and Verhaegen (1997) for the errors-in-variables problem can deal with the special type of disturbances that act on the state equation (10.3), if the input signal is a zero-mean white-noise sequence. For such input signals, the method yields consistent estimates of the matrices A and C.

To explain the estimation of the matrices A and C by this subspace method, the data sequences are stored in block Hankel matrices. The block Hankel matrix for the output signal y_k is

$$Y_{j,s,N} := \begin{bmatrix} y_j & y_{j+1} & \cdots & y_{j+N-1} \\ y_{j+1} & y_{j+2} & \cdots & y_{j+N} \\ \vdots & \vdots & \ddots & \vdots \\ y_{j+s-1} & y_{j+s} & \cdots & y_{j+N+s-2} \end{bmatrix}. \tag{10.7}$$

The sequences $\widetilde{u}_k$, v_k, ξ_k, and $\widetilde{w}_k := w_k - F^{(1)}(u_k \otimes v_k)$ are stored in a similar way into the matrices $\widetilde{U}_{j,s,N}$, $V_{j,s,N}$, $\Xi_{j,s,N}$, and $\widetilde{W}_{j,s,N}$. In terms of these data matrices, equation (10.4) becomes

$$Y_{j,s,N} = \Gamma_s X_{j,N} + H_s \widetilde{U}_{j,s,N} + G_s \widetilde{W}_{j,s,N} + V_{j,s,N} + \Xi_{j,s,N}, \tag{10.8}$$

where

$$\Gamma_s \;:=\; \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix}, \qquad X_{j,N} := [x_j, x_{j+1}, \ldots, x_{j+N-1}],$$

$$H_s \;:=\; \begin{bmatrix} \widetilde{D} & 0 & 0 & \cdots & 0 \\ C\widetilde{B} & \widetilde{D} & 0 & \cdots & 0 \\ CA\widetilde{B} & C\widetilde{B} & \widetilde{D} & & 0 \\ \vdots & & \ddots & \ddots & \\ CA^{s-2}\widetilde{B} & CA^{s-3}\widetilde{B} & \cdots & C\widetilde{B} & \widetilde{D} \end{bmatrix},$$

$$
G_s \quad := \quad
\begin{bmatrix}
0 & 0 & 0 & \cdots & 0 \\
C & 0 & 0 & \cdots & 0 \\
CA & C & 0 & & 0 \\
\vdots & & \ddots & \ddots & \\
CA^{s-2} & CA^{s-3} & \cdots & C & 0
\end{bmatrix},
$$

and $\widetilde{D} = [D, 0]$. The subspace method is based on estimating the column space of Γ_s and using the structure of Γ_s to compute the matrices A and C up to a similarity transformation. The next theorem shows that the column space of the extended observability matrix Γ_s can be obtained by using 'past' input and output values as instrumental variables, and computing an RQ factorization. It is similar to Theorem 3 in the paper by Chou and Verhaegen (1997) on errors-in-variables subspace identification.

Theorem 10.2 *Consider the bilinear system (9.1)–(9.2) driven by a zero-mean white-noise input signal that is independent of the disturbances v_k and w_k. Let $Y_{j,s,N}$ be a data matrix constructed from y_k as in equation (10.7), and let $U_{j,s,N}$ be constructed in a similar way from the input sequence u_k, and $\widetilde{U}_{j,s,N}$ from the extended input sequence*

$$
\widetilde{u}_k := \begin{bmatrix} u_k \\ u_k \otimes y_k \\ u_k \otimes u_k \end{bmatrix}.
$$

Given the RQ factorization

$$
\begin{bmatrix}
\widetilde{U}_{r+1,s,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix} \\
Y_{r+1,s,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix}
\end{bmatrix}
=
\begin{bmatrix}
R_{11} & 0 \\
R_{21} & R_{22}
\end{bmatrix}
\begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix},
\tag{10.9}
$$

where $r = (1 + m)s$, the following holds:

$$
\lim_{N\to\infty} \frac{1}{\sqrt{N}} R_{22} = \lim_{N\to\infty} \frac{1}{\sqrt{N}} \Gamma_s X_{r+1,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix} Q_2^T, \quad \text{w. p. 1.}
\tag{10.10}
$$

Proof: Because the white-noise disturbances v_k and w_k are independent of the input:

$$
\lim_{N\to\infty} \frac{1}{N} \widetilde{W}_{r+1,s,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix} = 0, \quad \text{w. p. 1,}
$$

$$
\lim_{N\to\infty} \frac{1}{N} V_{r+1,s,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix} = 0, \quad \text{w. p. 1.}
$$

From Theorem 10.1 it follows that

$$
\lim_{N\to\infty} \frac{1}{N} \Xi_{r+1,s,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix} = 0, \quad \text{w. p. 1.}
$$

These results together with equation (10.8) and the RQ factorization (10.9) yield

$$\lim_{N\to\infty} \frac{1}{\sqrt{N}} Y_{r+1,s,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix} Q_2^T = \lim_{N\to\infty} \frac{1}{\sqrt{N}} \Gamma_s X_{r+1,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix} Q_2^T$$

$$+ \lim_{N\to\infty} \frac{1}{\sqrt{N}} H_s R_{11} Q_1 Q_2^T$$

$$= \lim_{N\to\infty} \frac{1}{\sqrt{N}} \Gamma_s X_{r+1,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix} Q_2^T,$$

with probability one, and also

$$Y_{r+1,s,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix} Q_2^T = R_{22}.$$

Combining these two expressions completes the proof.   □

From this theorem it follows that the columns of the matrix R_{22} are linear combinations of the columns of the matrix Γ_s. Thus, the column space of Γ_s is contained in the column space of R_{22}. If the matrix

$$\lim_{N\to\infty} \frac{1}{\sqrt{N}} X_{r+1,N} \begin{bmatrix} U_{1,r,N}^T & Y_{1,r,N}^T \end{bmatrix} (Q_2)^T$$

has rank n, and if $s > n$, the column space of R_{22} equals the column space of Γ_s. Empirical evidence suggests that this rank condition holds if the input signal is a white-noise sequence. Thus, R_{22} can be used to determine the matrices A and C. To determine the column space of R_{22}, the following SVD is used:

$$R_{22} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} V^T,$$

where $U_1 \in \mathbb{R}^{\ell s \times n}$ and $\Sigma_1 \in \mathbb{R}^{n \times n}$. The number of dominant singular values in Σ_1 indicates the order of the bilinear system. The matrices A and C are computed up to a similarity transformation T, based on the relation

$$U_1 = \Gamma_s T = \begin{bmatrix} CT \\ CTT^{-1}AT \\ \vdots \\ CT\left(T^{-1}AT\right)^{s-1} \end{bmatrix} = \begin{bmatrix} C_T \\ C_T A_T \\ \vdots \\ C_T A_T^{s-1} \end{bmatrix}.$$

The matrix $C_T = CT$ equals the first ℓ rows of U_1, that is, $C_T = U_1(1 : \ell, :)$. The matrix $A_T = T^{-1}AT$ is computed by solving the overdetermined equation

$$U_1(1 : (s-1)\ell, :)A_T = U_1(\ell + 1 : s\ell, :),$$

which has a unique solution, because of the condition $s > n$.

10.3   Estimating the Matrices B, D, and F

The previous subsection showed that a subspace identification method can be used to obtain unbiased estimates of the matrices A and C. This section shows

how the remaining matrices B, D, and F can be estimated. Unfortunately, unbiased estimates of B, D, and F are difficult to obtain, even with white-noise input sequences. Therefore, it is suggested that the procedures outlined below are used to obtain initial estimates of the matrices B, D, and F, and that these estimates are improved with the nonlinear optimization-based methods that will be discussed in Chapter 11.

Using the property $\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B)$, the output equation (10.4) can be rewritten as

$$
\begin{aligned}
y_k \;=\; & CA^{k-j}x_j + \left(\sum_{\tau=j}^{k-1} \widetilde{u}_\tau^T \otimes CA^{k-\tau-1} \right) \text{vec}(\widetilde{B}) + (u_k^T \otimes I_\ell)\text{vec}(D) \\
& + C\sum_{\tau=j}^{k-1} A^{k-\tau-1}\Big(w_\tau - F^{(1)}(u_\tau \otimes v_\tau)\Big) + v_k + \xi_k.
\end{aligned} \tag{10.11}
$$

This equation shows that y_k depends linearly on $\widetilde{B}$ and D. To estimate the matrices $\widetilde{B}$ and D, their entries are collected in the parameter vector θ as follows:

$$
\theta := \begin{bmatrix} \text{vec}(\widetilde{B}) \\ \text{vec}(D) \end{bmatrix} = \begin{bmatrix} \text{vec}(B) \\ \text{vec}\left(F^{(1)}\right) \\ -\text{vec}\left(F_D^{(1)}\right) \\ \text{vec}(D) \end{bmatrix}.
$$

For k ranging from j to $j + N - 1$, equation (10.11) can be written as

$$
Y_j^T = \Gamma_N x_j + \Phi_j \theta + \Psi_j + V_j^T + \Xi_j^T,
$$

where

$$
\begin{aligned}
Y_j \;&:=\; [y_j, y_{j+1}, \ldots, y_{j+N-1}], \\
V_j \;&:=\; [v_j, v_{j+1}, \ldots, v_{j+N-1}], \\
\Xi_j \;&:=\; [\xi_j, \xi_{j+1}, \ldots, \xi_{j+N-1}], \\
\Phi_j \;&:=\; \begin{bmatrix} 0 & u_j^T \otimes I_\ell \\ \widetilde{u}_j^T \otimes C & u_{j+1}^T \otimes I_\ell \\ \vdots & \vdots \\ \sum_{\tau=j}^{j+N-2} \widetilde{u}_\tau^T \otimes CA^{N-1} & u_{j+N-1}^T \otimes I_\ell \end{bmatrix}, \\
\Psi_j \;&:=\; \begin{bmatrix} 0 \\ C\widetilde{w}_j \\ \vdots \\ C\sum_{\tau=j}^{j+N-2} A^{N-1}\widetilde{w}_\tau \end{bmatrix}.
\end{aligned}
$$

The parameter vector θ is estimated by solving the least-squares problem

$$
\min_\theta \|Y_j^T - \Phi_j \theta\|_2^2.
$$

The solution is given by

$$\widehat{\theta} = \left(\Phi_j^T \Phi_j \right)^{-1} \Phi_j^T Y_j^T. \tag{10.12}$$

From this estimate the matrices B, D, and $F^{(1)}$ can be derived. If $F_{\widetilde{C}}^{(2)} \neq 0$, also an estimate of the matrix $F_{\widetilde{C}}^{(2)}$ is needed to form the matrix F. The following procedure can be used to obtain an initial estimate of the complete F matrix:

1. Compute an estimate of the state sequence by simulating the equation

$$\widehat{x}_{k+1} = A\widehat{x}_k + \widetilde{B}\widetilde{u}_k.$$

2. Compute an estimate of $F_{\widetilde{C}}^{(2)}$ from the relation

$$
\begin{aligned}
y_k \;\approx\; & CA^{k-j}\widehat{x}_j + C\sum_{\tau=j}^{k-1} A^{k-\tau-1}\left(\widetilde{B}\widetilde{u}_\tau - F^{(1)}(u_\tau \otimes v_\tau) + w_\tau \right) \\
& + C\sum_{\tau=j}^{k-1} A^{k-\tau-1} F_{\widetilde{C}}^{(2)}(u_k \otimes \widehat{x}_k) + Du_k + v_k.
\end{aligned}
$$

   This can be done by solving a linear least-squares problem similar to the one used for computing $\widetilde{B}$ and D.

3. The estimate of F follows from

$$F = \left[\begin{array}{cccc} F_1^{(1)}C & F_2^{(1)}C & \cdots & F_m^{(1)}C \end{array} \right] + F_{\widetilde{C}}^{(2)}.$$

Unfortunately, the estimates of B, D, and F obtained by this procedure are biased. Nevertheless, they are suitable to initialize a nonlinear optimization-based identification procedure, like the ones that will be described in Chapter 11. Since the estimates of the matrices A and C are unbiased, in principle only the matrices B, D, and F need to be optimized. Such an approach was described by Verdult et al. (1998a). However, to make the identification procedure robust with respect to a violation of the white-noise input assumption, it is recommended that all the system matrices are optimized. This is the approach that will be followed in Chapter 11.

Chapter 11

Identification Based on Nonlinear Optimization

Bilinear systems can be identified using subspace methods. However, these methods only yield approximate models. The methods discussed in Chapter 9 are based on an approximation of the state sequence. Moreover, in practice the dimensions of the data matrices involved in these methods can be enormous and must be reduced. A reduction of the dimensions can be obtained by applying the subset-selection techniques in Chapter 4, but this results in an additional approximation. The subspace method discussed in Chapter 10 yields unbiased estimates of the matrices A and C of the bilinear system (9.1)–(9.2), only if the system is driven by a zero-mean white-noise input signal. The other system matrices can only be estimated approximately, because of a bias. For nonwhite input signals also the matrices A and C are biased.

The performance of the approximate model obtained by one of the bilinear subspace identification methods might not be satisfactory. It would be desirable to have a method available that can be used to iteratively refine the model to improve its performance. Nonlinear optimization techniques can be used for this purpose. A cost function is formulated that depends on the system matrices of the bilinear model, and that decreases when the model approaches the true system. Such a cost function is often nonlinear, and can have several local minima. An iterative optimization algorithm is used to minimize the cost function with respect to the system matrices. Most optimization algorithms can get stuck in local minima, and therefore not reach the global optimum. Exactly where the algorithms end up very much depends on the initialization of the system matrices. An already good initial estimate of the model steers the algorithm to the global optimum or a local optimum that is close to it. Therefore, a good initial model is of paramount importance; the previously discussed subspace methods can be used to obtain such an initial model.

Subspace methods and nonlinear optimization-based methods should not be viewed as competing identification methods, but rather as complementary tools that in combination allow to tackle real-life identification problems. Combined,

the methods benefit from each other: the approximate subspace model provides a good initial starting point for the nonlinear optimization-based method that iteratively improves the model. In Part I, a similar combination of methods for the identification of LPV system has been discussed.

This chapter describes two nonlinear optimization-based methods for bilinear systems. The first method is described in Section 11.1, and is a special case of the projected gradient search method for LPV system identification that appeared in Chapter 5. The second method is based on the principle of separable least-squares, and is described in detail in Section 11.2.

11.1   Projected Gradient Search

In Chapter 8 it was shown that the bilinear system (8.1)–(8.2) can be regarded as a special case of the LPV system (3.1)–(3.2); $p_k = u_k$, $A_0 = A$, $B_0 = B$, and $A_i = F_i$, $B_i = 0$ for $i = 1, 2, \ldots, m$, with $F = [F_1, F_2, \ldots, F_m]$. Therefore, the optimization-based identification method in Chapter 5 can also be used to identify the bilinear system

$$x_{k+1} = Ax_k + F(u_k \otimes x_k) + Bu_k, \tag{11.1}$$
$$y_k = Cx_k + Du_k + v_k, \tag{11.2}$$

which is disturbed by a white-noise sequence v_k at the output. The advantage of the method in Chapter 5 is that the bilinear system does not need to be parameterized in a special way. The parameters to be optimized are simply the entries of the system matrices. To avoid ill-conditioning and nonuniqueness of the state-space representation, the parameters that change the cost function are determined numerically at each iteration of the optimization algorithm, and only these parameters are updated. A detailed description of this procedure can be found in Chapter 5.

When no process noise is present, as in (11.1)–(11.2), identification can be carried out by minimizing the output error, given by

$$J_N(\theta) := \sum_{k=1}^{N} \|y_k - \hat{y}_k(\theta)\|_2^2, \tag{11.3}$$

where y_k is the output of the system, $\hat{y}_k(\theta)$ the output of the model, and θ is a set of parameters that describe the system matrices of the model. The projected gradient search method in Section 5.1 can be used to minimize this output error. The special structure of the bilinear system gives rise to some parameters that are zero by definition and can be dealt with by introducing an appropriate selection matrix Ξ as discussed at the end of Subsection 5.1.1.

For a special type of process noise:

$$x_{k+1} = Ax_k + F(u_k \otimes x_k) + Bu_k + Kv_k,$$
$$y_k = Cx_k + Du_k + v_k,$$

it is easy to derive the one-step-ahead predictor:

$$
\begin{aligned}
x_{k+1} &= (A - KC)x_k + F(u_k \otimes x_k) + (B - KD)u_k + Ky_k, & (11.4) \\
y_k &= Cx_k + Du_k + v_k, & (11.5)
\end{aligned}
$$

and the prediction-error method in Section 5.2 can be used. Note that for more general types of process noise, formulation of the predictor requires knowledge of the noise model.

11.2   Separable Least-Squares

Separable least-squares (SLS) is a method described by Golub and Pereyra (1973) to solve nonlinear optimization problems with a particular structure. It deals with problems in which the cost function to be minimized depends on a number of parameters that can be separated into two disjoint sets: a set of parameters that enters the cost function in a linear way and another set of parameters that enters in a nonlinear way. The parameters that enter in a linear way can be eliminated from the problem, resulting in a modified nonlinear optimization problem with a reduced dimension of the parameter space. When this modified nonlinear optimization problem has been solved, the linear parameters follow from a linear least-squares problem.

   SLS has been used for the identification of linear state-space systems and for nonlinear Wiener state-space systems by Bruls et al. (1999) and Haverkamp (2001). It can also be used for the identification of bilinear systems, as will be explained below.

   This section is divided into four subsections. Subsection 11.2.1 shows that the output of the bilinear system can be written such that it depends linearly on the system matrices B and D, and nonlinearly on the matrices A, C, and F. Separation into linear and nonlinear entering matrices provides the basis for applying SLS to the minimization of the output error (11.3); this is the topic of Subsection 11.2.2. Subsection 11.2.3 discusses the parameterization of the system matrices. Finally, in Subsection 11.2.4 it is shown how to deal with process noise.

11.2.1   Bilinear Output Equation

The bilinear system (11.1)–(11.2) can also be viewed as a quasi-linear system with a time-varying A matrix; this matrix is given by

$$
A_k := A + \sum_{i=1}^{m}[u_k]_i F_i. \tag{11.6}
$$

With this interpretation, recursive substitution of the state equation yields an expression for the output at time instant k:

$$
y_k = C \prod_{h=j}^{k-1} A_h x_j + \sum_{\tau=j}^{k-1} C \left(\prod_{h=\tau+1}^{k-1} A_h \right) B u_\tau + Du_k + v_k,
$$

where

$$\prod_{h=j}^{k} A_h := \left\{ \begin{array}{ll} A_k A_{k-1} \cdots A_{j-1} A_j & j \leq k, \\ 1 & j > k. \end{array} \right.$$

Using the property $\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B)$, this expression can also be written as

$$\begin{aligned} y_k &= C \prod_{h=j}^{k-1} A_h x_j + \left(\sum_{\tau=j}^{k-1} u_\tau^T \otimes C \prod_{h=\tau+1}^{k-1} A_h \right) \mathrm{vec}(B) \\ &\quad + (u_k^T \otimes I_\ell)\mathrm{vec}(D) + v_k. \end{aligned} \tag{11.7}$$

This equation illustrates that if A, C, and F are fixed, the output y_k depends linearly on the matrices B and D; conversely, if B and D are fixed, the output y_k depends nonlinearly on A, C, and F.

To apply SLS, the output y_k given by (11.7) needs to be written as a function of a set of nonlinear parameters, and a set of linear ones. An obvious choice for the linear parameters is the collection of all entries of the matrices B and D. Thus, the linear parameters, denoted by θ_ℓ, are given by

$$\theta_\ell := \left[\begin{array}{c} \mathrm{vec}(B) \\ \mathrm{vec}(D) \end{array} \right].$$

The nonlinear parameters describe how y_k depends on A, C, and F; they are denoted by θ_n. The choice of θ_n depends on the parameterization used for the matrices A, C, and F; and is discussed in Subsection 11.2.3. First, it is shown how the concept of separable least-squares can be applied.

11.2.2 Solving the Optimization Problem

The bilinear system (11.1)–(11.2) is identified from a finite number of measurements of the input u_k and the output y_k, by minimizing the output error, which is given by

$$J_N(\theta_n, \theta_\ell) := \sum_{k=j}^{j+N-1} \|y_k - \hat{y}_k(\theta_n, \theta_\ell)\|_2^2,$$

where y_k is the output of the system to be identified, $\hat{y}_k(\theta_n, \theta_\ell)$ the output of the model, and θ_ℓ and θ_n are the parameters that describe the system matrices of the model.

For k ranging from j to $j + N - 1$, equation (11.7) can be written as

$$Y_j^T = \Gamma_j(\theta_n)x_j + \Phi_j(\theta_n)\theta_\ell + V_j^T, \tag{11.8}$$

where

$$\begin{aligned} Y_j &:= [y_j, y_{j+1}, \ldots, y_{j+N-1}], & (11.9) \\ V_j &:= [v_j, v_{j+1}, \ldots, v_{j+N-1}], & (11.10) \end{aligned}$$

$$
\Gamma_j(\theta_n) \quad := \quad
\begin{bmatrix}
C \\
CA_j \\
\vdots \\
C \displaystyle\prod_{h=j}^{j+N-2} A_h
\end{bmatrix},
\tag{11.11}
$$

$$
\Phi_j(\theta_n) \quad := \quad
\begin{bmatrix}
0 & & u_j^T \otimes I_\ell \\
u_j^T \otimes C & & u_{j+1}^T \otimes I_\ell \\
\vdots & & \vdots \\
\displaystyle\sum_{\tau=j}^{j+N-2} u_\tau^T \otimes C \displaystyle\prod_{h=\tau+1}^{j+N-2} A_h & & u_{j+N-1}^T \otimes I_\ell
\end{bmatrix}.
\tag{11.12}
$$

Now the output error becomes

$$
J_N(\theta_n, \theta_\ell) = ||Y_j^T - \Phi_j(\theta_n)\theta_\ell||_2^2.
$$

It is linear in θ_ℓ and nonlinear in θ_n. Because of this structure, the principle of SLS can be applied to minimize it. This principle is summarized in the following theorem:

Theorem 11.1 *(Golub and Pereyra 1973) Given the matrices (11.9) and (11.12). Assume that the matrix $\Phi_j(\theta_n)$ has constant rank over an open set $\mathcal{S} \subset \mathbb{R}^p$, with p the dimension of θ_n. Let*

$$
J_N(\theta_n, \theta_\ell) \quad := \quad ||Y_j^T - \Phi_j(\theta_n)\theta_\ell||_2^2
\tag{11.13}
$$

$$
\widetilde{J}_N(\theta_n) \quad := \quad ||Y_j^T - \Phi_j(\theta_n)\left(\Phi_j^T(\theta_n)\Phi_j(\theta_n)\right)^{-1}\Phi_j^T(\theta_n)Y_j^T||_2^2.
\tag{11.14}
$$

(a) If $\widehat{\theta}_n \in \mathcal{S}$ is a minimizer of $\widetilde{J}_N(\theta_n)$ and

$$
\widehat{\theta}_\ell = \left(\Phi_j^T(\widehat{\theta}_n)\Phi_j(\widehat{\theta}_n)\right)^{-1}\Phi_j^T(\widehat{\theta}_n)Y_j^T,
\tag{11.15}
$$

then $(\widehat{\theta}_n, \widehat{\theta}_\ell)$ is a minimizer of $J_N(\theta_n, \theta_\ell)$.

(b) If $(\widehat{\theta}_n, \widehat{\theta}_\ell)$ is a minimizer $J_N(\theta_n, \theta_\ell)$ for $\theta_n \in \mathcal{S}$, then $\widehat{\theta}_n$ is a minimizer of $\widetilde{J}_N(\theta_n)$ in $\mathcal{S}$ and $\widetilde{J}_N(\widehat{\theta}_n) = J_N(\widehat{\theta}_n, \widehat{\theta}_\ell)$. Furthermore, if there is a unique $\widehat{\theta}_\ell$ among the minimizing pairs of $J_N(\theta_n, \theta_\ell)$, then $\widehat{\theta}_\ell$ must satisfy (11.15).

This theorem tells us that minimizing equation (11.13) is equivalent to minimizing (11.14) and subsequently using its minimizer to compute θ_ℓ from (11.15). The minimization of (11.13) and of (11.14) are nonlinear optimization problems, which have to be solved numerically. The advantage of using SLS is that (11.14) has a reduced dimension of the parameter space, which results in a reduced computational load.

The next theorem shows that if the estimated nonlinear parameters equal the real ones, that is, if the global minimum of (11.14) is attained, and if the system is driven by a zero-mean white-noise input, then the estimated linear parameters are asymptotically unbiased.

Theorem 11.2 *Given the bilinear system (11.1)–(11.2) and the optimization problems of Theorem 11.1. Assume that u_k is a zero-mean white-noise sequence that is statistically independent of the zero-mean white-noise sequence v_k, and of the initial state x_j. Assume also that $\widehat{\theta}_n = \theta_n$. If the matrix*

$$\lim_{N \to \infty} \frac{1}{N} \Phi_j^T(\theta_n) \Phi_j(\theta_n) \tag{11.16}$$

has full rank with probability one, and if the bilinear system is such that the eigenvalues of the matrix A and of the matrix

$$A \otimes A + \sum_{i=1}^{m} \mathbf{E}\Big[[u_k]_i^2\Big] F_i \otimes F_i \tag{11.17}$$

have a magnitude that is smaller than one, the parameters $\widehat{\theta}_\ell$ calculated from (11.15) satisfy:

$$\lim_{N \to \infty} \widehat{\theta}_\ell - \theta_\ell = 0, \quad \text{w. p. 1.}$$

Proof: With equation (11.8) it follows that

$$\Phi_j^T(\theta_n) \Phi_j(\theta_n)(\widehat{\theta}_\ell - \theta_\ell) = \Phi_j^T(\theta_n)\Big(\Gamma_j(\theta_n)x_j + V_j^T\Big).$$

Hence, it needs to be shown that

$$\lim_{N \to \infty} \frac{1}{N} \Phi_j^T(\theta_n) V_j^T = 0, \quad \text{w. p. 1,}$$

$$\lim_{N \to \infty} \frac{1}{N} \Phi_j^T(\theta_n) \Gamma_j(\theta_n) x_j = 0, \quad \text{w. p. 1.} \tag{11.18}$$

The first equation is almost trivial, because u_k is independent of v_k. The second equation involves somewhat more work.

For ease of notation the matrix $Z_k \in \mathbb{R}^{\ell \times mn}$ is introduced:

$$
\begin{aligned}
Z_k \; &:= \; \big[z_k^{11}, z_k^{21}, \ldots, z_k^{n1}, \ldots, z_k^{1m}, z_k^{2m}, \ldots, z_k^{nm}\big] \\
&:= \; \sum_{\tau=j}^{k-1} \Big[C \prod_{h=\tau+1}^{k-1} A_h[u_\tau]_1, \ldots, C \prod_{h=\tau+1}^{k-1} A_h[u_\tau]_m\Big] \\
&= \; \sum_{\tau=j}^{k-1} u_\tau^T \otimes C \prod_{h=\tau+1}^{k-1} A_h.
\end{aligned}
$$

The signal z_k^{pq} ($p = 1, \ldots, n$, $q = 1, \ldots, m$) can be viewed as the output of a bilinear system of the form

$$
\begin{aligned}
x_{k+1}^{pq} &= A x_k^{pq} + F(u_k \otimes x_k^{pq}) + e_p[u_k]_q, \\
z_k^{pq} &= C x_k^{pq},
\end{aligned}
$$

where the initial condition equals $x_j^{pq} = 0$, and the n dimensional vector e_p is zero everywhere except for the pth entry, which equals one. Now the left-hand side of

equation (11.18) can be written as

$$\frac{1}{N}\Phi_j^T(\theta_n)\Gamma_j(\theta_n)x_j = \frac{1}{N}\sum_{k=j}^{j+N-1}\left[\begin{array}{c} Z_k^T C x_k^0 \\ u_k \otimes C x_k^0 \end{array}\right],\qquad(11.19)$$

where

$$x_k^0 = C\prod_{h=j}^{k-1}A_h x_j.\qquad(11.20)$$

The signal x_k^0 represents the evolution of the state due to the initial condition x_j. Since x_k^0 does not depend on u_k, the second block row of the right-hand side of equation (11.19) goes to zero. The first block row of the right-hand side of equation (11.19) also goes to zero. To see this, note that the block rows of $Z_k^T C x_k^0$ can be written as $(x_k^{pq})^T C^T C x_k^0$, which is just the weighted sum of products between the elements of x_k^{pq} and x_k^0. Since the system is driven by a white-noise sequence, and the eigenvalues of the matrix A and the matrix (11.17) have magnitudes less than one, the covariance of the state sequence does not depend on time (see Theorem 8.2 on page 123); therefore, the matrix

$$M := \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\mathbf{E}[x_k^{pq}(x_k^0)^T]$$

satisfies

$$\begin{aligned} M &= AMA^T + F\left(\mathbf{E}[u_k u_k^T]\otimes M\right)F^T \\ &= AMA^T + \sum_{i=1}^{m}\mathbf{E}\left[[u_k]_i^2\right]F_i M F_i^T, \end{aligned}$$

where use has been made of the following facts: $\mathbf{E}[u_k w_k^T] = 0$, $\mathbf{E}[u_{k+1}(x_k^0)^T] = 0$, $\mathbf{E}[u_{k+1}(u_k\otimes x_k^0)^T] = 0$, $\mathbf{E}[u_{k+1}(x_k^{pq})^T] = 0$, and $\mathbf{E}[u_{k+1}(u_k\otimes x_k^{pq})^T] = 0$.

Using the property $\mathrm{vec}(ABC) = (C^T\otimes A)\mathrm{vec}(B)$, the last equation can also be written as

$$\left(I_{n^2} - A\otimes A - \sum_{i=1}^{m}\mathbf{E}\left[[u_k]_i^2\right]F_i\otimes F_i\right)\mathrm{vec}(M) = 0.$$

This equation has a unique solution $\mathrm{vec}(M)$ that equals the zero vector, because the eigenvalues of the matrix (11.17) have magnitudes less than one. Thus, it follows that

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\mathbf{E}[x_k^{pq}(x_k^0)^T] = \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}x_k^{pq}(x_k^0)^T = 0,\quad\text{w. p. 1.}$$

Hence, equation (11.19) goes to zero with probability one for $N\to\infty$.   □

11.2.3　Parameterizing the System Matrices

For SLS optimization of the output error of the bilinear system (11.1)–(11.2), the matrices A, C, and F need to be parameterized by a set of parameters θ_n. As discussed above, these matrices are estimated by solving a nonlinear optimization from which the matrices B and D have been eliminated. To apply the SLS method, the parameterization of A, C, and F must be independent of the matrices B and D (see Theorem 11.1). In addition, the parameterization should be unique, and numerically robust, that is, not sensitive to small perturbations. The often used phase-variable parameterization that was discussed in Section 8.2 is unique, but far from numerically robust; it is an extension of a canonical form for linear systems, which is known to be very sensitive to small perturbations of the parameters (McKelvey 1995).

At present, no suitable parameterization of the bilinear system exists. Therefore, it seems reasonable to use a numerically robust parameterization for linear systems to parameterize the matrices A and C, and by lack of a better choice, to fully parameterize F. Such a parameterization removes the nonuniqueness due to similarity transformations. Any remaining ill-conditioning should be taken care of by an appropriate algorithm for the numerical optimization, like the Levenberg-Marquardt method (Moré 1978; Coleman, Branch, and Grace 1999), which is based on a regularized Hessian.

Several choices exist to parameterize the matrices A and C in a numerical robust way. Two parameterizations that have been used for SLS identification of linear systems are the tridiagonal parameterization and the output-normal parameterization (Haverkamp 2001). Below, only the latter parameterization is briefly explained.

The output-normal parameterization (Hanzon and Peeters 1997; Peeters, Hanzon, and Olivi 1999) describes the matrices A and C with $n\ell$ parameters. Since the matrix F is fully parameterized, the total number of parameters in θ_n is $n\ell + n^2 m$. To parameterize the matrices A and C in output-normal form, the first step is to apply a similarity transformation T such that the transformed matrices $A_T = T^{-1}AT$ and $C_T = CT$ satisfy $A_T^T A_T + C_T^T C_T = I_n$. The next step is to perform an orthogonal similarity transformation Q such that the matrix

$$\begin{bmatrix} C_T \\ C_T A_T \end{bmatrix} Q$$

is lower triangular with positive entries on the diagonal. Finally, the transformed pair is parameterized as follows:

$$\begin{bmatrix} C_T Q \\ Q^{-1} A_T Q \end{bmatrix} = T_1 T_2 \cdots T_n \begin{bmatrix} 0 \\ I_n \end{bmatrix},$$

with

$$T_k = \begin{bmatrix} I_{k-1} & & \\ & U_k & \\ & & I_{n-k} \end{bmatrix} \in \mathbb{R}^{(n+\ell) \times (n+\ell)},$$

$$U_k = \begin{bmatrix} -s_k & S_k \\ r_k & s_k^T \end{bmatrix},$$

$$r_k = \sqrt{1 - s_k^T s_k},$$

$$S_k = I_\ell - \frac{1 - r_k}{s_k^T s_k} s_k s_k^T,$$

where s_k contains ℓ parameters. The $n\ell$ parameters that describe A and C are thus $s_1, s_2, \ldots, s_n$. A detailed description of this procedure is given by Haverkamp (2001).

11.2.4 Dealing with Process Noise

The SLS method can also be used when the state equation is disturbed by white process noise w_k, as follows:

$$x_{k+1} = Ax_k + F(u_k \otimes x_k) + Bu_k + w_k, \tag{11.21}$$

$$y_k = Cx_k + Du_k + v_k. \tag{11.22}$$

In the presence of process noise, equation (11.7) becomes

$$y_k = C\prod_{h=j}^{k-1} A_h x_j + \left(\sum_{\tau=j}^{k-1} u_\tau^T \otimes C \prod_{h=\tau+1}^{k-1} A_h \right) \text{vec}(B)$$

$$+ (u_k^T \otimes I_\ell)\text{vec}(D) + C\sum_{\tau=j}^{k-1} \left(\prod_{h=\tau+1}^{k-1} A_h \right) w_\tau + v_k, \tag{11.23}$$

and equation (11.8) becomes

$$Y_j^T = \Gamma_j(\theta_n)x_j + \Phi_j(\theta_n)\theta_\ell + \Psi_j(\theta_n) + V_j^T, \tag{11.24}$$

where

$$\Psi_j(\theta_n) := \begin{bmatrix} 0 \\ Cw_j \\ \vdots \\ C\sum_{\tau=j}^{j+N-2} \left(\prod_{h=\tau+1}^{j+N-2} A_h \right) w_\tau \end{bmatrix}.$$

The additional term $\Psi_j(\theta_n)$ can cause some problems. Since $\Psi_j(\theta_n)$ depends on u_k, it can be correlated with the regressor $\Phi_j(\theta_n)$. Correlation between $\Psi_j(\theta_n)$ and $\Phi_j(\theta_n)$ leads to a biased estimate of the linear parameters θ_ℓ, if they are estimated with equation (11.15). For the special case where the input u_k is a zero-mean white-noise sequence that is independent of w_k and v_k, an unbiased estimate of θ_ℓ can be obtained by solving (11.15).

Theorem 11.3 *Given the bilinear system (11.21)–(11.22) and the optimization problems of Theorem 11.1. Assume that u_k is a zero-mean white-noise sequence that is statistically independent of the zero-mean white-noise sequences v_k and w_k, and of the initial state x_j. Assume also that $\widehat{\theta}_n = \theta_n$. If the matrix*

$$\lim_{N \to \infty} \frac{1}{N} \Phi_j^T(\theta_n) \Phi_j(\theta_n) \tag{11.25}$$

has full rank with probability one, and if the bilinear system is such that the eigenvalues of the matrix A and of the matrix

$$A \otimes A + \sum_{i=1}^m \mathbf{E}\Big[[u_k]_i^2 \Big] F_i \otimes F_i \tag{11.26}$$

have a magnitude that is smaller than one, the parameters $\widehat{\theta}_\ell$ calculated from (11.15) satisfy:

$$\lim_{N \to \infty} \widehat{\theta}_\ell - \theta_\ell = 0, \quad \text{w. p. 1.}$$

Proof: With equation (11.24) it follows that

$$\Phi_j^T(\theta_n) \Phi_j(\theta_n)(\widehat{\theta}_\ell - \theta_\ell) = \Phi_j^T(\theta_n)\Big(\Gamma_j(\theta_n)x_j + \Psi_j(\theta_n) + V_j^T \Big).$$

Hence, it needs to be shown that

$$\lim_{N \to \infty} \frac{1}{N} \Phi_j^T(\theta_n) V_j^T \;=\; 0, \quad \text{w. p. 1,} \tag{11.27}$$

$$\lim_{N \to \infty} \frac{1}{N} \Phi_j^T(\theta_n) \Gamma_j(\theta_n) x_j \;=\; 0, \quad \text{w. p. 1,} \tag{11.28}$$

$$\lim_{N \to \infty} \frac{1}{N} \Phi_j^T(\theta_n) \Psi_j(\theta_n) \;=\; 0, \quad \text{w. p. 1.} \tag{11.29}$$

The first equation is almost trivial, because u_k is independent of v_k. Equation (11.28) was already derived in the proof of Theorem 11.2.

To proof of (11.29), the matrix Z_k that was introduced in the proof of Theorem 11.2 is used. With the matrix Z_k, the left-hand side of equation (11.29) can be written as

$$\frac{1}{N} \Phi_j^T(\theta_n) \Psi_j(\theta_n) = \frac{1}{N} \sum_{k=j}^{j+N-1} \left[\begin{array}{c} Z_k^T C x_k^s \\ u_k \otimes C x_k^s \end{array} \right], \tag{11.30}$$

where

$$x_k^s := \sum_{\tau=j}^{k-1} \prod_{h=\tau+1}^{k-1} A_h w_\tau. \tag{11.31}$$

The signal x_k^s is the part of the state of the bilinear system due to the process noise; it satisfies $x_{k+1}^s = A x_k^s + F(u_k \otimes x_k^s) + w_k$. The remaining part of the proof of (11.29) goes along the same lines as the proof of (11.28) with x_k^0 replaced by x_k^s. $\qquad\square$

When the input is not a white-noise sequence, the linear parameters obtained from (11.15) are biased. Unbiased estimates can be obtained by applying an instrumental variable approach (Söderström and Stoica 1983). The available input data set is split into two halves: the part with later time indices is used to construct instruments and the other part is used to construct the regressors and regressed variables. Instead of (11.14), the problem

$$\min_{\theta_n} ||Y_j^T - \Phi_j(\theta_n)\Big(\Phi_{j+N}^T(\theta_n)\Phi_j(\theta_n)\Big)^{-1}\Phi_{j+N}^T(\theta_n)Y_j^T||_2^2 \tag{11.32}$$

is solved and the linear parameters are computed from

$$\widehat{\theta}_\ell = \Big(\Phi_{j+N}^T(\widehat{\theta}_n)\Phi_j(\widehat{\theta}_n)\Big)^{-1}\Phi_{j+N}^T(\widehat{\theta}_n)Y_j^T. \tag{11.33}$$

The following theorem shows that in this way unbiased estimates are obtained:

Theorem 11.4 *Given the bilinear system (11.21)–(11.22) and the optimization problem (11.32). Assume that u_k is a zero-mean random sequence that is statistically independent of the zero-mean white-noise sequences v_k and w_k, and of the initial state x_j. Assume also that $\widehat{\theta}_n = \theta_n$. If the matrix*

$$\lim_{N\to\infty}\frac{1}{N}\Phi_{j+N}^T(\theta_n)\Phi_j(\theta_n) \tag{11.34}$$

has full rank with probability one, and if the bilinear system bounded-input, bounded-output stable, the parameters $\widehat{\theta}_\ell$ calculated from (11.33) satisfy:

$$\lim_{N\to\infty}\widehat{\theta}_\ell - \theta_\ell = 0, \quad \text{w. p. 1.}$$

Proof: With equation (11.24) it follows that

$$\Phi_{j+N}^T(\theta_n)\Phi_j(\theta_n)(\widehat{\theta}_\ell - \theta_\ell) = \Phi_{j+N}^T(\theta_n)\Big(\Gamma_j(\theta_n)x_j + \Psi_j(\theta_n) + V_j^T\Big).$$

Hence, it needs to be shown that

$$\lim_{N\to\infty}\frac{1}{N}\Phi_{j+N}^T(\theta_n)V_j^T = 0, \quad \text{w. p. 1,} \tag{11.35}$$

$$\lim_{N\to\infty}\frac{1}{N}\Phi_{j+N}^T(\theta_n)\Psi_j(\theta_n) = 0, \quad \text{w. p. 1,} \tag{11.36}$$

$$\lim_{N\to\infty}\frac{1}{N}\Phi_{j+N}^T(\theta_n)\Gamma_j(\theta_n)x_j = 0, \quad \text{w. p. 1.} \tag{11.37}$$

The first equation is almost trivial, because u_k is independent of v_k.

The left-hand side of equation (11.36) can be written as

$$\frac{1}{N}\Phi_{j+N}^T(\theta_n)\Psi_j(\theta_n) = \frac{1}{N}\sum_{k=j}^{j+N-1}\left[\begin{array}{c}\left(\sum_{\tau=j+N}^{k+N-1} u_\tau^T \otimes C \prod_{h=\tau+1}^{k+N-1} A_h\right)^T Cx_k^s \\ u_{k+N} \otimes Cx_k^s\end{array}\right], \tag{11.38}$$

with x_k^s as in (11.31) on page 162. The signal x_k^s is the part of the state of the bilinear system due to the process noise. Equation (11.38) goes to zero with probability one for $N \to \infty$, because for $k < j + N$, x_k^s does not depend on the inputs $u_{j+N}, u_{j+N+1}, \ldots, u_{k+2N-1}$.

The left-hand side of equation (11.37) can be written as

$$
\frac{1}{N}\Phi_{j+N}^T(\theta_n)\Gamma_j(\theta_n)x_j = \frac{1}{N}\sum_{k=j}^{j+N-1}\left[\begin{array}{c}\left(\sum_{\tau=j+N}^{k+N-1} u_\tau^T \otimes C \prod_{h=\tau+1}^{k+N-1} A_h\right)^T Cx_k^0 \\ u_{k+N} \otimes Cx_k^0\end{array}\right],
$$
(11.39)

with x_k^0 as in (11.20) on page 159. The signal x_k^0 represents the evolution of the state due to the initial condition x_j. Equation (11.39) goes to zero with probability one for $N \to \infty$, because for $k < j + N$, x_k^0 does not depend on the inputs $u_{j+N}, u_{j+N+1}, \ldots, u_{k+2N-1}$.   □

The suggestion that may follow from this theorem is that when the input signal is not a white-noise sequence, the instrumental variable approach should be used to obtain the system matrices. However, in practice this will not always yield the best results. Even if the input is nonwhite, it may be better to solve (11.14) and (11.15) instead of (11.32) and (11.33). Reason being that it is often easier to satisfy the rank condition (11.25) than the condition (11.34). In other words, the inverse matrix in (11.15) is often better conditioned than the inverse matrix in (11.33). Therefore, using (11.33) can result in a large variance error, leading to a loss of accuracy. The bias resulting from (11.14) and (11.15) for nonwhite inputs can be very small if the system and the input are such that the magnitudes of the eigenvalues of the matrix (11.26) are smaller than one. The proof of Theorem 11.3 shows that in this case, the bias that remains is only due to the second block row of the right-hand side of equation (11.30). To conclude, it might be better to accept a small bias when (11.14) and (11.15) are solved, than to obtain unbiased results with (11.32) and (11.33) at the price of a large variance error. In Subsection 12.1.1 these claims are illustrated with some simulation examples.

Another approach to identify a system subject to process noise, is to minimize the prediction error instead of the output error. For this approach a predictor needs to be derived; in general, this requires knowledge of the noise model. However, for a special type of process noise, namely $w_k = Kv_k$, the one-step-ahead predictor equals (11.4)–(11.5), as discussed in Section 11.1. For this special case, the SLS optimization can be applied to the predictor (11.4)–(11.5), and a result similar to Theorem 11.2 can be derived. Problems can occur if the predictor becomes unstable during the optimization. Empirical evidence suggests that this only happens occasionally (Fnaiech and Ljung 1987). To completely avoid stability problems, considerably more complex procedures are needed that use the time-varying Kalman filter as discussed by Fnaiech and Ljung (1987).

Chapter 12

Examples

The identification methods for bilinear systems that were described in the previous chapters are now applied to several examples. First, in Section 12.1 Monte-Carlo simulations are presented for a single input, single output (SISO) bilinear system to compare the performance of the different methods for both white-noise and nonwhite-noise input signals. Next, in Section 12.2 the identification methods are used to determine a bilinear model of a high-purity distillation column.

12.1 Simulations

This section contains Monte-Carlo simulations for identification of a bilinear system of the form (9.1)–(9.2). To keep the computational load of the simulations low, a simple SISO system is used. The system is borrowed from Favoreel, De Moor, and Van Overschee (1996), and its system matrices are:

$$
A = \begin{bmatrix} 0 & 0.5 \\ -0.5 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix},
$$

$$
C = \begin{bmatrix} 1 & 1 \end{bmatrix}, \qquad D = 2,
$$

$$
F = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.3 \end{bmatrix}.
$$

To evaluate the performance of the different identification methods, the eigenvalues of the estimated A and F matrices are used. As a second measure of the performance, the response of the estimated models is computed using a fresh data set, that is, a data set that was not used for estimating the models. As a performance measure for this response, the variance-accounted-for (VAF) is used; the VAF is defined by (6.1) on page 88.
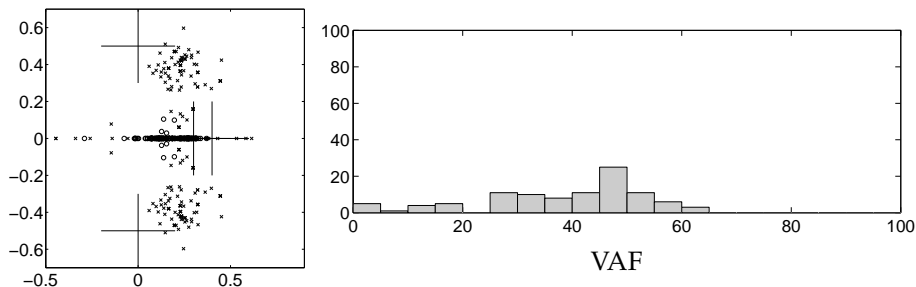
Figure 12.1: Quality of the initial bilinear models. *Left:* Eigenvalues of the matrices A (*crosses*) and F (*circles*) of the models in the complex plane for 100 experiments. The big crosses correspond to the eigenvalues of the real system. *Right:* Histogram of VAF values (%) of the 100 models obtained on a fresh data set. The range of VAF values from 0 to 100% is divided into bins of 5%. For each bin, it is shown how many data sets out of the total 100 resulted in VAF values that fall into that bin.

12.1.1 Instrumental Variables in Separable Least-Squares

To evaluate the use of instrumental variables in the SLS identification method in Section 11.2, the bilinear system was simulated 100 times, with a nonwhite random input sequence that was generated as follows:

$$u_k = (0.75 + 1.05q^{-1} + 0.15q^{-2})\mu_k + 4H(q)\nu_k,$$

where μ_k and ν_k are zero-mean Gaussian white-noise sequences of unit variance, q denotes the shift operator, and $H(q)$ is a second-order low-pass Butterworth filter with a cut-off frequency of one-fifth of the sample frequency. To ensure a good excitation of the bilinear term, the variance of the input signal was normalized to 1.6. In each simulation different realizations of the disturbance signals w_k and v_k acted on the system. The disturbance signal v_k was generated as a white-noise sequence with variance equal to 0.03; the signal w_k was derived from v_k as Kv_k with the matrix K equal to $[\sqrt{2/3}, \sqrt{1/3}]^T$. For each simulation 1000 samples were generated and used to estimate the system matrices with the SLS identification method with and without instrumental variables. Recall that SLS without instrumental variables is given by (11.14) and (11.15) on page 157, and SLS with instrumental variables by (11.32) and (11.33) on page 163.

The nonlinear optimizations involved in the SLS methods were solved numerically using the Levenberg-Marquardt iterative method (Moré 1978; Coleman, Branch, and Grace 1999). They were initialized by estimating a bilinear model using the subset-selection procedure in Chapter 4 in combination with the two-block identification method in Section 9.3. The block size of the data matrices equaled $k = 4$. The subset selection was carried out such that only half of the rows of the matrix Z_X was selected, and that out of the 221 rows of $Z_{P\odot X}$ and U, only 100 rows were selected. Figure 12.1 shows the estimated eigenvalues and the VAF values of the initial bilinear models.

Figure 12.2 shows the estimated eigenvalues and the VAF values on a fresh data set for the models obtained from SLS without instrumental variables. Fig-
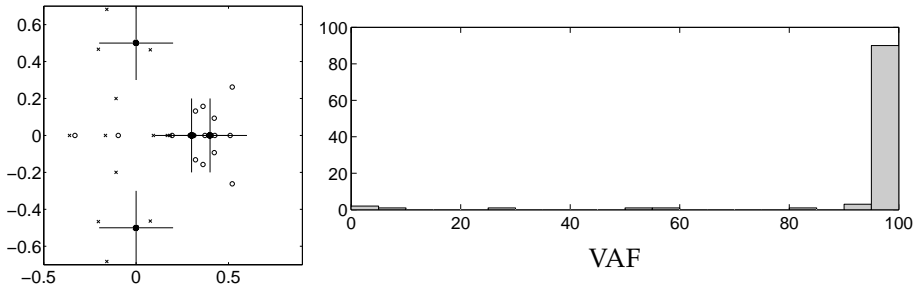
Figure 12.2: Quality of the bilinear models obtained by SLS without instrumental variables. See Figure 12.1 for an explanation of the quantities shown.
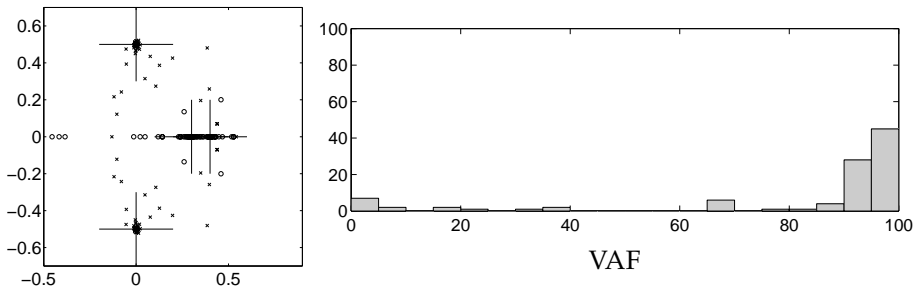
Figure 12.3: Quality of the bilinear models obtained by SLS with instrumental variables. See Figure 12.1 for an explanation of the quantities shown.

ure 12.3 shows the same quantities for the models obtained from SLS with instrumental variables. The outliers in the graphs are a sign that the nonlinear optimization problem sometimes got stuck in a local minimum. Because the input signal is a nonwhite random sequence, it is expected that the method that uses instrumental variables yields the best results (see Subsection 11.2.4 for an explanation). However, the figures show the contrary: although the input is nonwhite, the SLS optimization without instrumental variables yields better results. This is because the matrix $\Phi_j^T(\theta_n)\Phi_j(\theta_n)$ used in SLS without instrumental variables (11.15) is usually better conditioned than the matrix $\Phi_{j+N}^T(\theta_n)\Phi_j(\theta_n)$ used in SLS with instrumental variables (11.33). Figure 12.4 compares the condition numbers of these matrices and confirms this assertion. The conclusion that follows is that the use of instrumental variables does not necessarily improve the performance of the estimated models.

12.1.2 Comparison of Identification Methods Using White-Noise Inputs

The bilinear system described above was used to compare the performance of the following subspace identification methods:

- the two-block method in Section 9.3; this method corresponds to the method
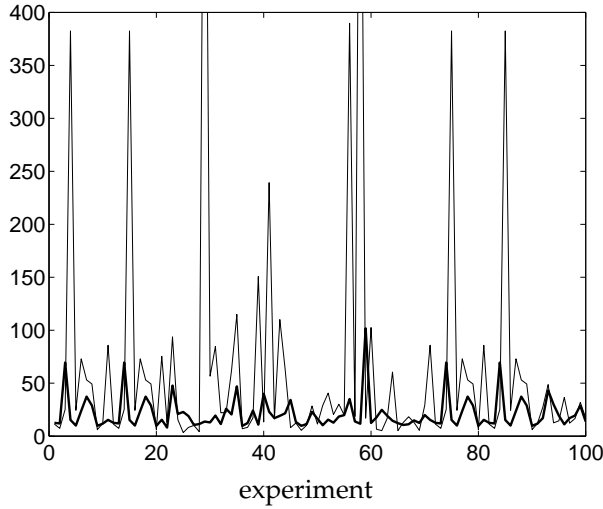
Figure 12.4: The condition number of the matrix $\Phi_j^T(\theta_n)\Phi_j(\theta_n)$ used in SLS without instrumental variables (*thick line*), and of the matrix $\Phi_{j+N}^T(\theta_n)\Phi_j(\theta_n)$ used in SLS with instrumental variables (*thin line*); for each simulation experiment.

introduced by Favoreel (1999);

- the four-block method in Subsection 9.2.3; this method uses a similar approximation of the state as the methods of Chen and Maciejowski (2000b);

- the four-block method with reduced dimensions in Subsection 9.2.3; this method uses a similar approximation of the state as the methods of Chen and Maciejowski (2000b);

- the input-extension method in Chapter 10; this method should work well if the input is a white-noise sequence;

- the subset-selection procedure in Chapter 4 in combination with the two-block method in Section 9.3.

The bilinear system was simulated 100 times; each simulation consists of 1200 samples. To ensure a good excitation of the bilinear term, the variance of the input signal was normalized to 2. In each simulation, different realizations of the disturbance signals w_k and v_k acted on the system. The disturbance signal v_k was generated as a white-noise sequence with variance equal to 0.03; the signal w_k was derived from v_k as Kv_k with the matrix K equal to $[\sqrt{2/3}, \sqrt{1/3}]^T$.

The block size for the input-extension method was taken equal to $s = 3$. The block size for the other subspace methods was taken equal to $k = 2$. Note that these choices of block size are consistent, because as a result the matrix $Y_{j,s,N}$ used in the input-extension method equals the matrix $\mathcal{Y}_{k+j|j}$ used in the other methods. In the subset-selection procedure, 5 rows were selected out of the 28 rows of Z_X; the total number of rows in $Z_{P\odot X}$ and U was first reduced from 92 to
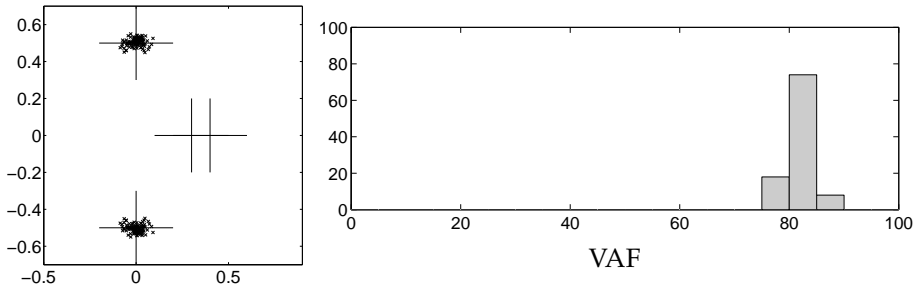
Figure 12.5: Quality of the linear models obtained by subspace identification, for a white-noise input. *Left:* Eigenvalues of the matrix A (*crosses*) of the models in the complex plane for 100 experiments. The big crosses correspond to the eigenvalues of the matrices A and F of the real bilinear system. *Right:* Histogram of VAF values (%) of the 100 models obtained on a fresh data set. The range of VAF values from 0 to 100% is divided into bins of 5%. For each bin, it is shown how many data sets out of the total 100 resulted in VAF values that fall into that bin.

88, by discarding all rows involving more than six multiplications; then from the remaining 88 rows, 17 rows were selected.

In addition to the bilinear subspace methods, a linear subspace method was used to identify a linear time-invariant system. The performance of the estimated linear systems provides a reference point. The linear subspace method that was used, was the PI-MOESP method of Verhaegen (1993) with a block size equal to 3. Figure 12.5 shows the estimated eigenvalues of the A matrix of the linear model, along with the VAF values obtained on a fresh data set.

The results of the different bilinear models are presented in Figures 12.6–12.10. These figures show the estimated eigenvalues of the A and F matrix, and the VAF values obtained on a fresh data set. Clearly, the two-block method has a superior performance, followed by the input-extension method and the subset-selection method. Although the performance of the four-block methods is quite good, it is not as good as the performance of the other methods. A possible explanation is the fact that the number of samples is small compared to the number of rows used in the RQ factorization of the four-block methods. Note however that the performance of the four-block methods depends to large extend on the particular choice of instrumental variables that is used. Therefore, one should be careful extrapolating these results to the methods of Chen and Maciejowski (2000b), which use a completely different set of instrumental variables. It can be concluded that the approximate methods, the input-extension method and the subset-selection method give quite a good performance, even better than the four-block methods for white-noise input signals.

In addition to the subspace methods, also the optimization-based methods in Chapter 11 were evaluated. Both the projected gradient search and the SLS method were tested using the same 100 simulations as were used for the subspace methods. To test the influence of the initial model, both the input-extension method and the subset-selection method were used to generate initial models for the optimization-based methods. This results in four different optimization-based
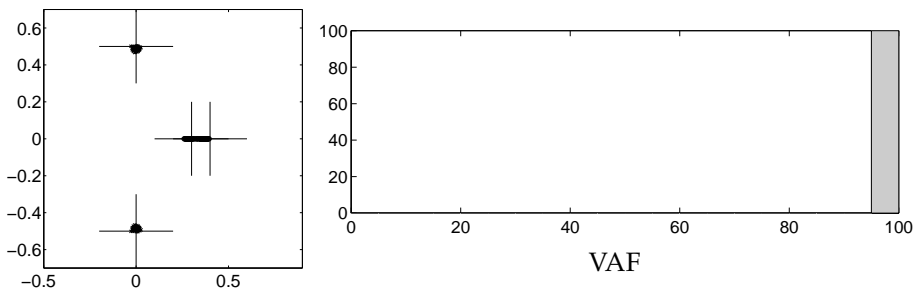
Figure 12.6: Quality of the bilinear models obtained by the two-block method in Section 9.3, for a white-noise input. *Left:* Eigenvalues of the matrices *A* (*crosses*) and *F* (*circles*) of the models in the complex plane for 100 experiments. The big crosses correspond to the eigenvalues of the real system. *Right:* Histogram of VAF values (%) of the 100 models obtained on a fresh data set. The range of VAF values from 0 to 100% is divided into bins of 5%. For each bin, it is shown how many data sets out of the total 100 resulted in VAF values that fall into that bin.
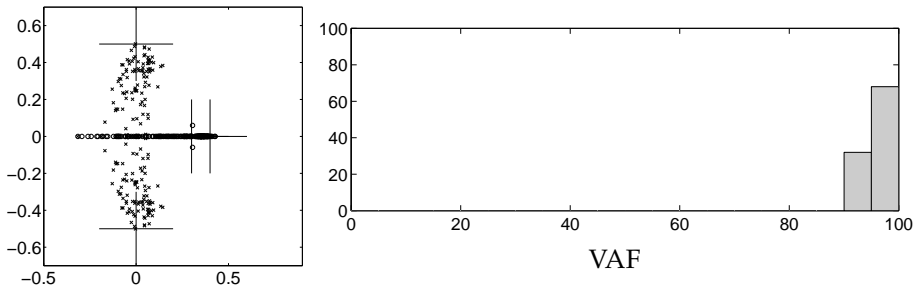
Figure 12.7: Quality of the bilinear models obtained by the four-block method in Subsection 9.2.3 for a white-noise input. See Figure 12.6 on page 170 for an explanation of the quantities shown.
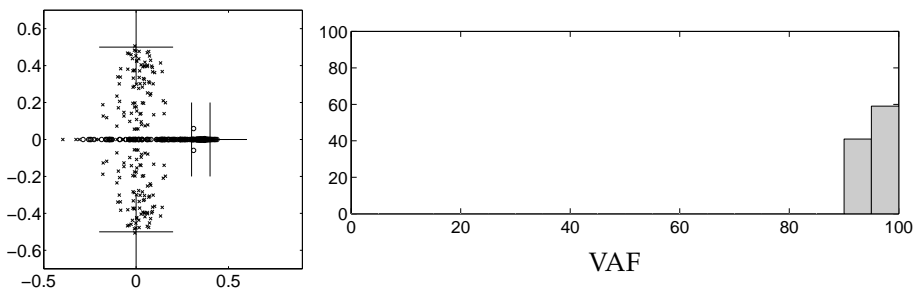
Figure 12.8: Quality of the bilinear models obtained by the four-block method with reduced dimensions in Subsection 9.2.3 for a white-noise input. See Figure 12.6 on page 170 for an explanation of the quantities shown.
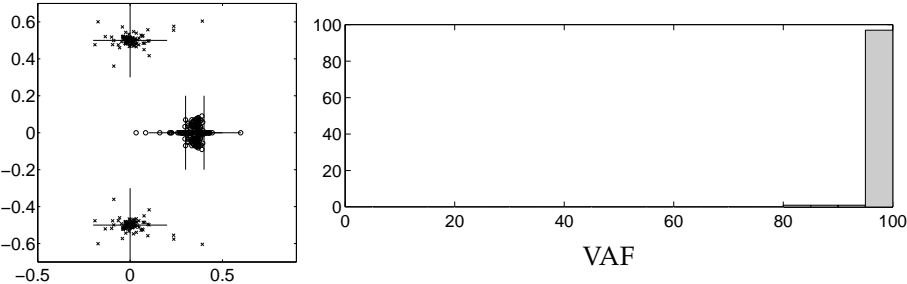
Figure 12.9: Quality of the bilinear models obtained by the input extension method in Chapter 10 for a white-noise input. See Figure 12.6 on page 170 for an explanation of the quantities shown.
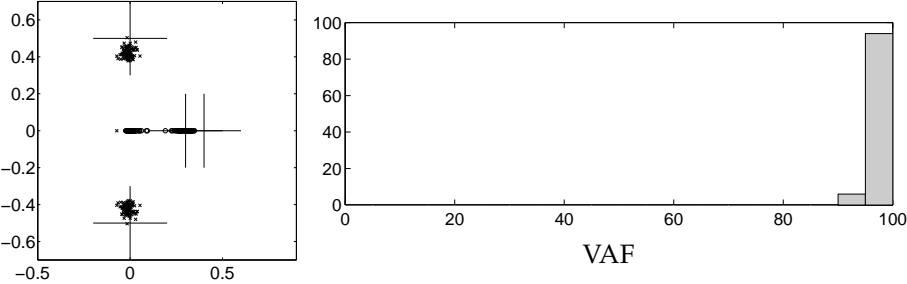
Figure 12.10: Quality of the bilinear models obtained by the subset-selection procedure in Chapter 4 in combination with the two-block method in Section 9.3 for a white-noise input. See Figure 12.6 on page 170 for an explanation of the quantities shown.
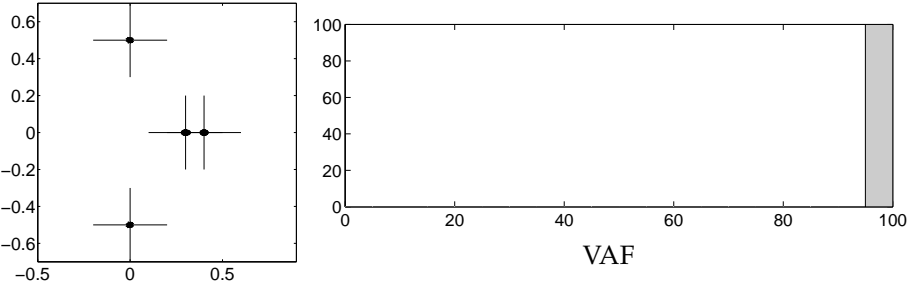
Figure 12.11: Quality of the bilinear models obtained by the optimization-based methods in Chapter 11 for a white-noise input. See Figure 12.6 on page 170 for an explanation of the quantities shown.

identification methods. The models estimated by the four methods were almost identical; the difference in VAF values was only 10^{-10}, which can be attributed to round-off errors. Figure 12.11 shows the estimated eigenvalues of the A and F matrix, and the VAF values obtained on a fresh data set. Although the methods all ended up with the same model, the number of iterations to achieve these results differed considerably. The projected gradient search method needed 12 iterations when initialized with the input-extension method, and 14 iterations when initialized with the subset-selection method. The SLS method needed 36 iterations when initialized with the input-extension method, and 56 iterations when initialized with the subset-selection method. The computational complexity of one iteration is dominated by the simulation of the bilinear system to compute the cost function, and is thus about the same for the projected gradient search and the SLS method. Therefore, the projected gradient search method is preferred to the SLS method: it needs less than half the number of iterations, and it is less sensitive to the initial estimate used.

12.1.3   Comparison of Identification Methods Using Nonwhite-Noise Inputs

To test the influence of the whiteness of the input signal, the experiments in Subsection 12.1.2 were repeated with an input signal generated as

$$u_k = (0.9 + 0.7q^{-1} + 0.1q^{-2})\mu_k + 4H(q)\nu_k,$$

where μ_k is a zero-mean Gaussian white-noise sequence with a variance of 1.5, ν_k is a zero-mean Gaussian white-noise sequence with unit variance, q denotes the shift operator, and $H(q)$ is a second-order low-pass Butterworth filter with a cut-off frequency of $1/20$ of the sample frequency. To ensure a good excitation of the bilinear term, the variance of the input signal was normalized to 2.

Figures 12.12–12.17 present the results of the subspace identification methods. Comparison of these results with the results for white-noise inputs in Figures 12.5–12.10 reveals that all methods suffer from a performance degradation. This may be explained from the fact that the white-noise signal provides a better excitation of the dynamics of the bilinear system than the nonwhite input signal does. Nevertheless, the performance of the methods in the nonwhite-noise case is still acceptable.

The results for the four nonlinear optimization-based methods differed considerably; they are shown in Figures 12.18–12.21, and are significantly different from the white-noise input case, where the results of the optimization-based methods were almost identical. In most cases, the optimization-based methods significantly improve the bilinear model. However, occasionally the results are very bad. This is either attributable to the initial model being unstable, to the model becoming unstable during iteration, or to hitting a local minimum of the cost function. The input-extension subspace method resulted in an initial unstable bilinear model in 13 out of 100 experiments. For the subset-selection subspace method this only happened once. Obviously, the unstable models could not be used to initialize the nonlinear optimization-based methods. For these cases, the initial models were used to generate the Figures 12.18–12.21. The figures show that initialization
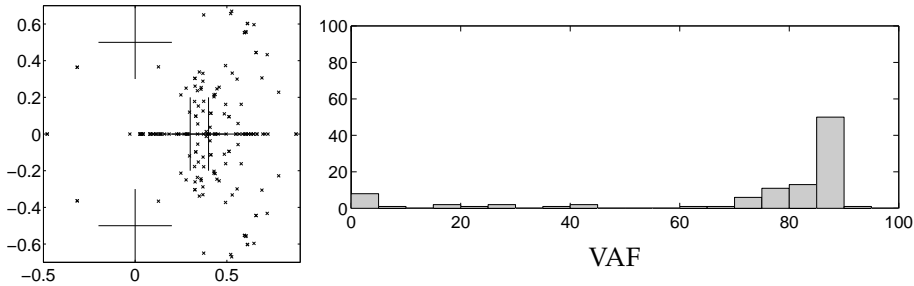
Figure 12.12: Quality of the linear models obtained by subspace identification, for a nonwhite-noise input. *Left:* Eigenvalues of the matrix A (*crosses*) of the models in the complex plane for 100 experiments. The big crosses correspond to the eigenvalues of the matrices A and F of the real bilinear system. *Right:* Histogram of VAF values (%) of the 100 models obtained on a fresh data set. The range of VAF values from 0 to 100% is divided into bins of 5%. For each bin, it is shown how many data sets out of the total 100 resulted in VAF values that fall into that bin.
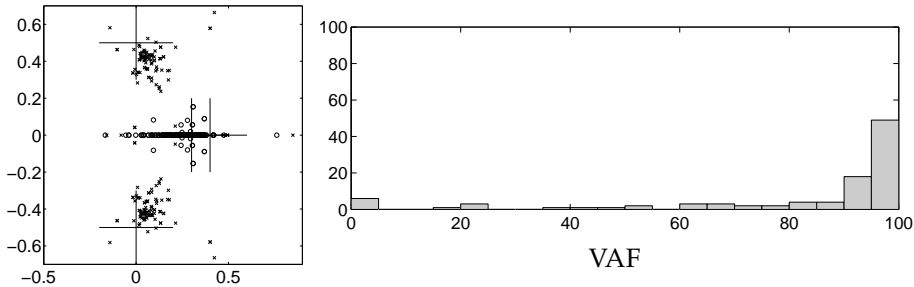
Figure 12.13: Quality of the bilinear models obtained by the two-block method in Section 9.3, for a nonwhite-noise input. *Left:* Eigenvalues of the matrices A (*crosses*) and F (*circles*) of the models in the complex plane for 100 experiments. The big crosses correspond to the eigenvalues of the real system. *Right:* Histogram of VAF values (%) of the 100 models obtained on a fresh data set. The range of VAF values from 0 to 100% is divided into bins of 5%. For each bin, it is shown how many data sets out of the total 100 resulted in VAF values that fall into that bin.
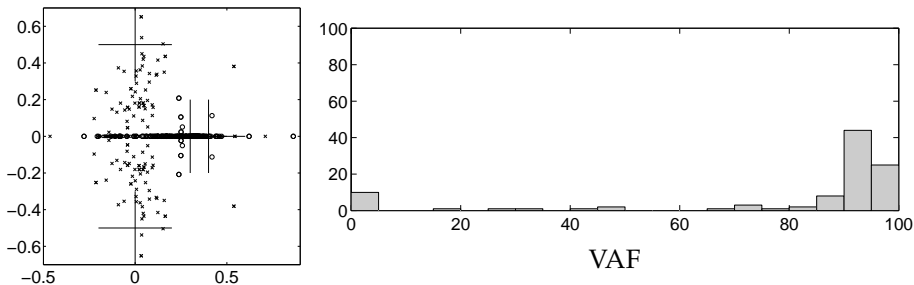
Figure 12.14: Quality of the bilinear models obtained by the four-block method in Subsection 9.2.3 for a nonwhite-noise input. See Figure 12.13 on page 173 for an explanation of the quantities shown.
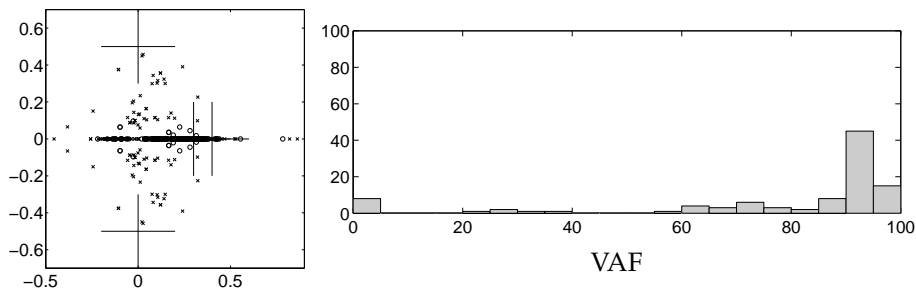
Figure 12.15: Quality of the bilinear models obtained by the four-block method with reduced dimensions in Subsection 9.2.3 for a nonwhite-noise input. See Figure 12.13 on page 173 for an explanation of the quantities shown.
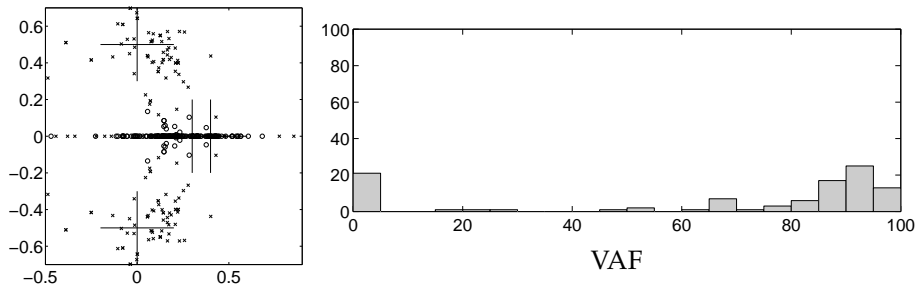
Figure 12.16: Quality of the bilinear models obtained by the input extension method in Chapter 10 for a nonwhite-noise input. See Figure 12.13 on page 173 for an explanation of the quantities shown.
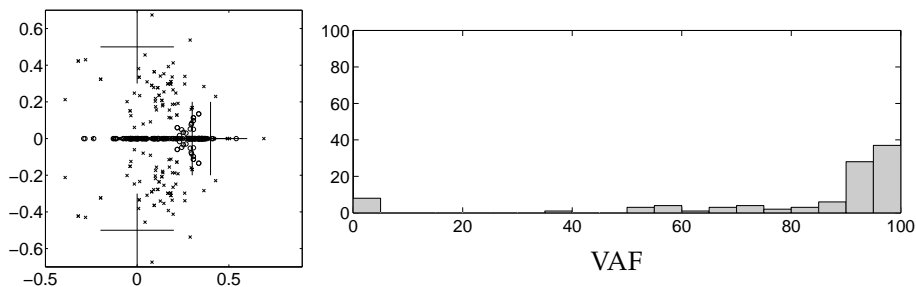
Figure 12.17: Quality of the bilinear models obtained by the subset-selection procedure in Chapter 4 in combination with the two-block method in Section 9.3 for a nonwhite-noise input. See Figure 12.13 on page 173 for an explanation of the quantities shown.
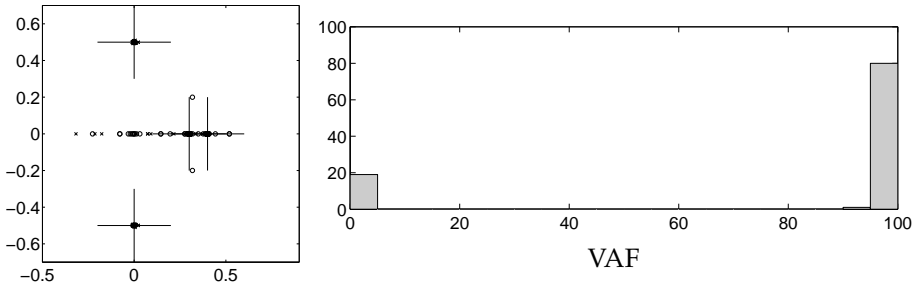
Figure 12.18: Quality of the bilinear models obtained by the SLS method initialized with the input-extension method for a nonwhite-noise input. See Figure 12.13 on page 173 for an explanation of the quantities shown.
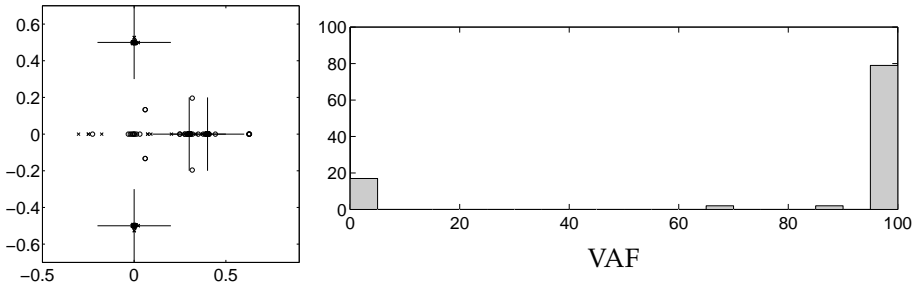
Figure 12.19: Quality of the bilinear models obtained by the projected gradient search method initialized with the input-extension method for a nonwhite-noise input. See Figure 12.13 on page 173 for an explanation of the quantities shown.
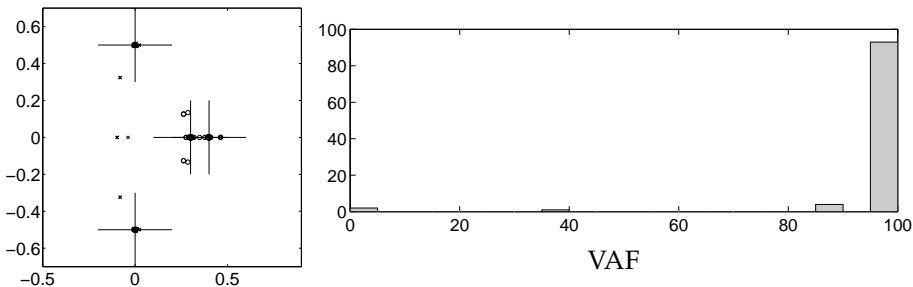
Figure 12.20: Quality of the bilinear models obtained by the SLS method initialized with the subset-selection method for a nonwhite-noise input. See Figure 12.13 on page 173 for an explanation of the quantities shown.
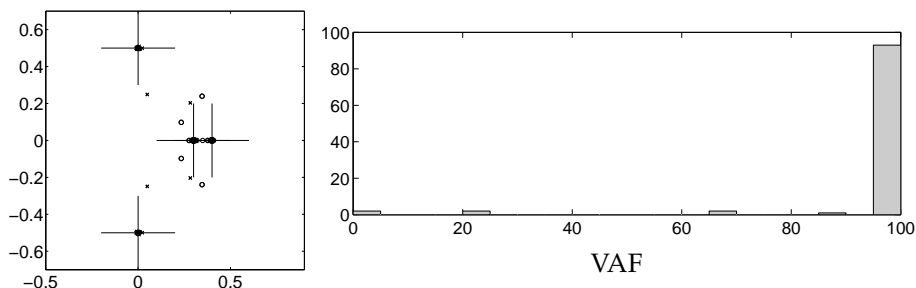
Figure 12.21: Quality of the bilinear models obtained by the projected gradient search method initialized with the subset-selection method for a nonwhite-noise input. See Figure 12.13 on page 173 for an explanation of the quantities shown.

with the model obtained from the subset-selection procedure yields slightly better results than initialization by the model obtained from the input-extension method. Therefore, the subset-selection method is the preferred method for initialization.

The number of iterations to achieve the results shown in Figures 12.18–12.21 differed considerably. The projected gradient search method needed 54 iterations when initialized with the input-extension method, and 67 iterations when initialized with the subset-selection method. The SLS method needed 55 iterations when initialized with the input-extension method, and 77 iterations when initialized with the subset-selection method. The number of iterations needed is significantly increased compared with the white-noise input case. The projected gradient search method is preferred, because it needs slightly less iterations than the SLS method when initialized with the subset-selection method.

12.2   Identification of a High-Purity Distillation Column

Distillation columns are commonly used in the chemical process industry to separate a mix of several components. Often, the goal is to extract the components with a very high purity. The purer the product needs to be, the more nonlinear the dynamic behavior of the distillation column. When the column is operated at different set-points to extract products of different purity, a nonlinear model of the column is required to describe the behavior at these set-points, and the transitions between them.

Besides nonlinearity, distillation columns also suffer from a directionality phenomenon that can cause ill-conditioning. A binary distillation column, that is, a column that has two output components, has a high gain direction that corresponds to one component becoming purer, and the other component becoming impurer. The low gain direction corresponds to both products becoming purer. Obviously, the low gain direction is of interest, if both products are to be as pure as possible. However, this direction is difficult to identify. It is possible to determine it by performing closed-loop experiments as, for example, discussed by Chou et al. (2000) and Bloemen (2002). In this section, the directionality problem

is not addressed, and the column is identified from open-loop data. This setting is applicable if only one product is required to be as pure as possible, since then only the high gain direction is of interest.

España and Landau (1975, 1978) have argued that under some assumptions, a distillation column can be accurately modeled by a low-order bilinear state-space model. This section shows that accurate bilinear models of a high-purity distillation column can be estimated using subspace and optimization-based bilinear identification methods.

A nonlinear simulation model of a binary distillation column was used to generate input and output data. The simulation model that was used is a 39 stage column with a reboiler and a condenser; it is 'column A' presented by Skogestad and Morari (1988) as a benchmark, and it is available from the Internet at `http://www.chembio.ntnu.no/users/skoge/`. The main assumptions for the column are: binary component, constant pressure, negligible vapor hold-up, total condenser, equimolal flow, and vapor-liquid equilibrium at all stages with constant relative volatility. The model was used with two inputs: reflux flow and boil-up flow; and two outputs: impurity at the top stage and impurity at the bottom stage. The feed rate and feed composition were kept constant. To stabilize the column, the LV configuration was used (Skogestad and Morari 1988).

The distillation column was simulated with a prolonged white-noise input signal; where prolonged means that the amplitude of the signal stays constant for at least 5 subsequent sample instances. The sampling time was 2 minutes. Additive white noise was added to the output, such that the signal to noise ratio became 20 dB.

After detrending the input and output signals, 2500 samples were used to identify bilinear models using different methods:

- the two-block subspace method in Section 9.3;

- the input-extension method in Chapter 10;

- the subset-selection method in Chapter 4 in combination with the two-block subspace method in Section 9.3;

- the SLS method in Section 11.2;

- the projected gradient search method in Section 11.1.

The block size for the input-extension method was taken equal to $s = 8$. To keep the dimensions of the data matrices within reasonable limits, the block size for the other subspace methods was taken equal to $k = 2$. In the subset-selection procedure, 15 rows were selected out of the 156 rows of Z_X; the total number of rows in $Z_{P \odot X}$ and U was first reduced from 1278 to 1086, by discarding all rows involving more than six multiplications; then from the remaining 1086 rows, 21 rows were selected. The nonlinear optimizations of the SLS and the projected gradient search method were numerically solved using the Levenberg-Marquardt iterative method (Moré 1978; Coleman, Branch, and Grace 1999). After some trial and error, it was decided that a second-order model yields the best performance.

The performance of the estimated models was evaluated by computing the VAF on a fresh data set of 2500 samples. The VAF values obtained for the different

Table 12.1: VAF values (%) computed for a fresh data set for the bilinear models of the distillation column.

	Two-block subspace	Input extension	Subset selection	SLS	Projected gradient
Output 1	79.9	86.0	92.4	94.8	94.8
Output 2	64.9	83.3	86.9	94.3	94.3

Table 12.2: VAF values (%) computed for a fresh data set for the linear and Wiener models of the distillation column.

	Linear	Wiener	SLS Wiener
Output 1	81.8	96.2	99.2
Output 2	78.5	90.6	98.8

identification methods are given in Table 12.1. The subset-selection method yields the best model among the subspace methods. As expected, the optimization-based methods yield the models having the best performance.

For comparison, the following models were estimated:

- a linear time-invariant model, with the PI-MOESP subspace method of Verhaegen (1993);

- a nonlinear Wiener model, with the subspace method of Westwick and Verhaegen (1996);

- a nonlinear Wiener model, with the SLS method of Bruls et al. (1999), initialized with the Wiener model obtained by subspace identification.

The Wiener model is a linear time-invariant system with a static nonlinearity at the output. Because the nonlinearity of the distillation column has a saturating characteristic, it is well suited to model distillation columns. Chou et al. (2000) describe an example of closed-loop identification of Wiener models for distillation columns.

The block sizes in the subspace methods were taken equal to 8. The nonlinear optimization for the SLS method was again solved using the Levenberg-Marquardt iterative method. Table 12.2 shows the VAF values obtained for the linear and Wiener models on a fresh data set. The bilinear models obtained from the optimization-based methods perform much better than the linear model. Their performance is comparable to the Wiener model obtained by subspace identification. The Wiener model obtained by SLS has the best performance. Figure 12.22 compares the outputs of the best linear, Wiener, and bilinear model with the real outputs of the column. Although the bilinear models give an accurate description of the dynamics of the distillation column, the Wiener model is slightly better and is therefore the preferred model.
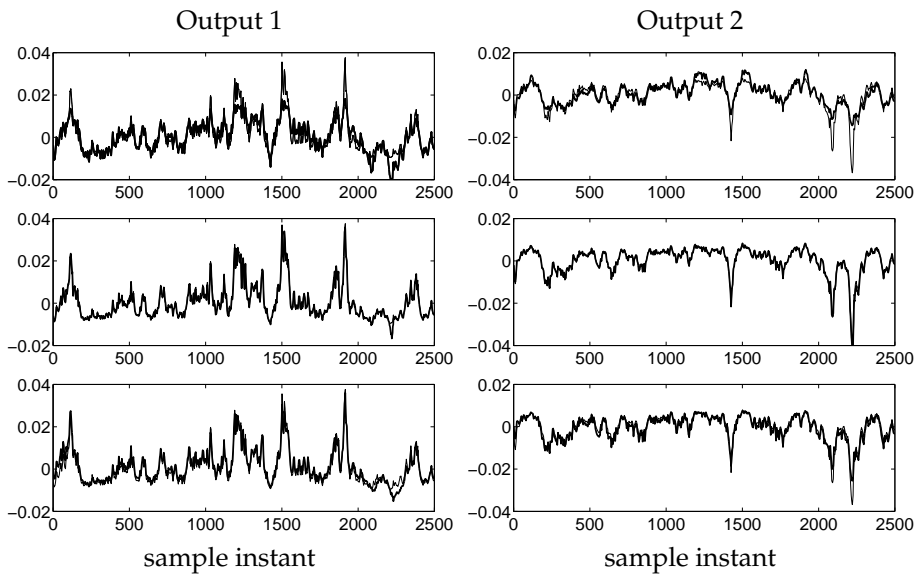
Figure 12.22: Outputs of the distillation column (*thin lines*) and the outputs of the linear model (*top, thick line*), the Wiener model obtained by SLS (*middle, thick line*), and the bilinear model obtained by SLS (*bottom, thick line*). The *left* graph shows the impurity of the top product; the *right* graph shows the impurity of the bottom product.

Chapter 13

Conclusions and Recommendations for Part II

Bilinear system identification has been discussed in the second part of this thesis. This chapter summarizes the most important results, and provides some suggestions for further research.

Subspace Identification

The bilinear system can be viewed as a special case of the LPV system discussed in Part I by taking the time-varying parameter p_k equal to the input signal. Consequently, the subspace methods for LPV systems discussed in Chapter 3 can also be used for identification of bilinear systems. The data matrices involved are constructed in a slightly different way, but the basic ideas stay the same. The key point is again the approximation of the state sequence. Different choices for this approximation are possible, leading to different algorithms, like the two-block, three-block, and four-block methods. The subspace algorithms that appeared in the literature before, the ones described by Favoreel (1999) and by Chen and Maciejowski (2000b), also fit in this framework.

Because of the similarity between the LPV subspace methods and the bilinear subspace methods, the conclusions for LPV subspace identification given in Chapter 7 are also applicable to bilinear subspace identification, and they are not repeated here. The same holds for the recommendations for further research. Like LPV subspace identification, bilinear subspace identification is in its infancy, so there is plenty of room for further research.

One peculiarity of the bilinear subspace methods is that, in the three-block and four-block methods, certain combinations of data matrices can lead to a repetition of rows; such a repetition did not show up in the LPV subspace methods. This repetition is undesirable, because it leads to a violation of certain rank conditions. This problem can be solved by simply reconstructing the data matrices such that no repeated rows occur.

Although the data matrices involved in bilinear subspace methods have a slightly smaller number of rows than the ones used in LPV subspace methods, the dimensions of the data matrices can still be a problem. As with the LPV methods, the number of rows of the data matrices grows exponentially with the block size of these data matrices. Since the block size should be larger than the order of the system, this can lead to enormous matrices for higher-order systems. In Chapter 4 a subset-selection technique has been described that selects only the most dominant rows of the data matrices used in LPV subspace identification. This subset-selection method reduces the dimensions drastically, at the expense of introducing an approximation error. The subset selection can also be used in bilinear subspace, to make it practically feasible.

An alternative way to overcome the dimensionality problem, is to excite the bilinear system by a white-noise input signal and exploit its special properties. For white-noise inputs, part of the bilinear term can be modeled by second-order products between the inputs and outputs. The bilinear system can then be approximated by a linear system with an extended input that consists of the original input plus these second-order products. The method is therefore called the input-extension method. The approximation errors can be dealt with by a suitable choice of instrumental variables in an errors-in-variables linear subspace identification technique. In this way, consistent estimates of the system matrices A and C can be obtained. The remaining system matrices can be estimated by solving a set of linear least-squares problems. However, the estimates of these matrices are biased, even for white-noise inputs. Therefore, it is desirable to improve the bilinear model obtained in this way by nonlinear optimization-based methods.

The different subspace identification methods were compared using extensive Monte-Carlo simulations. All methods yield acceptable results for white-noise input signals. For nonwhite input signals, the performance of all the methods slightly deteriorates, and is occasionally very bad. It might be that the white-noise signal excites the bilinear system better. Further research addressing the difference in performance for white-noise and nonwhite-noise signals is desirable. Overall, the two-block method —originally introduced by Favoreel (1999)— outperforms all the other subspace methods. However, since it suffers from the curse of dimensionality, it can only be applied to low-order systems with few inputs and outputs. The input-selection method and the subset-selection method do not suffer from this dimensionality problem, and yield models with a reasonable performance. The performance of these approximate methods is even better than the four-block methods described in this thesis. Note that these four-block methods use different instrumental variables than the ones proposed by Chen and Maciejowski (2000b), and consequently the performance can differ. These observations lead to the following recommendation: if the dimensions of the system allow it, use the two-block subspace method; if not, use the input extension or subset-selection method, and if desirable, improve the obtained model using a nonlinear optimization-based method.

Identification Based on Nonlinear Optimization

Bilinear system identification can also be carried out by minimizing the prediction error with respect to the system matrices. This minimization problem is nonlinear

and has to be solved numerically. Two prediction-error approaches have been described in this part of the thesis.

The first method exploits the fact that the bilinear system is a special case of the LPV system discussed in Part I. The projected gradient search algorithm for LPV system identification in Chapter 5 can be adapted to suit the bilinear identification problem. The advantage of this method is that no special parameterization is needed for the bilinear system. The projected gradient search method uses a full parameterization; at each minimization step it determines the active parameters, and only updates these parameters.

The second method is based on the principle of separable least-squares (SLS), described by Golub and Pereyra (1973). The prediction-error cost function for bilinear system identification depends linearly on the system matrices B and D, and nonlinearly on the matrices A, C, and F. The principle of separable least-squares states that the matrices B and D can be eliminated from the cost function, such that it only depends on A, C, and F. When this modified cost function has been minimized, the obtained estimates of A, C, and F can be used to compute the matrices B and D by solving a linear least-squares problem. The advantage of this approach is that the dimension of the parameter space is reduced, and that only the matrices A, C, and F need to be parameterized.

The SLS method needs a unique and numerically reliable parameterization of the system matrices A, C, and F. At present, no suitable parameterization for bilinear systems has been described in the literature. In this thesis it is proposed to parameterize the matrices A and C with an output-normal parameterization that is used in linear system identification, and to parameterize F by all its entries. Simulations showed that this parameterization is suitable for solving the identification problem. Nevertheless, its use is not yet theoretically justified, and thus the parameterization of bilinear state-space system remains a topic for future research.

In the presence of white process noise, the SLS method yields unbiased results if the input is a white-noise sequence. For nonwhite input signals, biased results are obtained. To avoid such a bias, SLS should be applied to the predictor, or instrumental variables should be used. However, both methods are not without problems. The predictor can become unstable during the optimization, as already discussed by Fnaiech and Ljung (1987). The instrumental variable approach requires a certain rank condition to hold. Simulation results showed that this rank condition is not always satisfied, leading to a loss of accuracy. The simulations also showed that the bias resulting from the 'standard' SLS method is usually very small; therefore the use of instrumental variables is not recommended.

Several Monte-Carlo simulations were performed to evaluate the nonlinear optimization-based methods. Both the input-extension subspace method and the subset-selection subspace method were used to generate bilinear models to initialize the numerical optimization procedure. For white-noise input signals, the SLS and projected gradient search methods yield identical results, regardless of the initialization used. For nonwhite-noise inputs, initialization with the subset-selection subspace method is preferred, since the input-extension method sometimes yields unstable models that are not suitable for initializing the nonlinear optimization. In all cases, the projected gradient search algorithm requires fewer

iterations than the SLS method and is therefore the preferred method.

Summing up, the subset-selection subspace method combined with the projected gradient search method forms a powerful tool to tackle bilinear identification problems. The subset-selection subspace method avoids the curse of dimensionality in bilinear subspace identification, and it provides an approximate model with a reasonable performance that is suitable for the initialization of the projected gradient search method. The projected gradient search method avoids the issue of parameterizing the bilinear system and it improves the initial bilinear model in an iterative fashion.

Part III

Local Linear State-Space Systems

Chapter 14

Introduction

Modeling and control of complex nonlinear systems is a challenging task. Nonlinear effects can no longer be neglected to be able to meet the specifications imposed on today's complex control systems. Unfortunately, when complexity increases, our ability to deal with it and understand it rapidly decreases. A common engineering way to deal with complexity is the divide-and-conquer strategy: decompose the complex problem into several subproblems that are easier to solve.

Based on the divide-and-conquer strategy, a method to model complex nonlinear systems has arisen. It is based on partitioning the whole operating range of the nonlinear system into multiple, smaller operating regimes and modeling the system for each of these regimes. The task of finding a complex global model for the system is thus replaced by determining local models that are less complex, and subsequently combining these local models into a global description for the system. This multiple model approach is often referred to as *operating regime decomposition* (Murray-Smith and Johansen 1997b). The rationale behind the approach is that the interactions between the relevant system phenomena are less complex locally than globally.

From the divide-and-conquer point of view, it is desirable to choose the local models such that they are less complex than the global nonlinear model. It is expected that the simpler the local models, the more models and thus the more operating regimes are needed to describe the global system sufficiently accurately. A trade-off has to be made, because too simple local models lead to an explosion of the number of operating regimes needed. Although it is possible to use nonlinear local models (see for example Pottmann, Unbehauen, and Seborg 1993), a common choice is to use local linear models. The main reasons for this choice are that a firm theory for linear systems has been developed over the years, that linear models are easy to understand, and that they are widely used by engineers.

Local linear models have proven their validity in different kinds of engineering problems. Some successful applications of local linear models are:

- Identification and model-based predictive control of the batch fermentation

of glucose to gluconic acid (Foss, Johansen, and Sørensen 1995; Johansen and Foss 1995; Azimzadeh, Galán, and Romagnoli 2001).

- Identification of a coal pulverizer used in coal-fired power stations (Cao, Rees, and Feng 1997).

- Modeling and control of electrically stimulated muscle for cardiac assistance from skeletal muscles and for functional electrical stimulation of limb muscle in paralyzed patients (Gollee and Hunt 1997; Gollee et al. 1997).

- Identification and adaptive predictive temperature control of a cross-flow heat exchanger (Nelles, Hecker, and Isermann 1997; Fischer, Nelles, and Isermann 1998).

- Identification and model-based predictive control of a pH neutralization process in a continuous stirred tank reactor (Townsend et al. 1998; Zhang and Morris 1999).

- Identification and model-based predictive control of a semibatch reactor for free-radical polymerization of polymethyl methacrylate (Lakshmanan and Arkun 1999).

- Identification of the combustion process in diesel engines for advanced engine control systems (Hafner et al. 2000).

- Identification of longitudinal vehicle dynamics and high-precision speed control of a Mercedes–Benz medium-size lorry for the development of intelligent cruise control systems (Hunt et al. 1996, 2000).

- Identification of a continuous stirred tank bioreactor (De Bruin and Roffel 1996; Cao, Rees, and Feng 1997; Verdult, Ljung, and Verhaegen 2001).

This part of the thesis focuses on the modeling of a nonlinear system by operating regime decomposition. Each operating regime is modeled by a local linear model. A weighted combination of the local models is used to obtain a global description of the nonlinear system. The weighting functions can be interpreted as model validity functions: they indicate which model or combination of models is active for a certain operating regime of the system. A more precise description of the weighted combination of local linear models is provided below in Section 14.1. Figure 14.1 illustrates the concept of operating regime decomposition.

The weights that combine the local linear models can also be interpreted as fuzzy membership functions: they indicate the degree of validity of a certain operating regime. With this interpretation, the local linear model description is equivalent to the widely used Takagi-Sugeno fuzzy model (Takagi and Sugeno 1985). Fuzzy modeling is also an operating regime decomposition method that uses local linear models. An overview of fuzzy modeling can be found in the Ph. D. thesis of Babuška (1997) and in the overview papers by Babuška and Verbruggen (1996, 1997). A precise study of the relation between Takagi-Sugeno fuzzy models and the local linear model description has been performed by Hunt, Haas, and Murray-Smith (1996) for the case of normalized radial basis functions as weighting/membership functions.
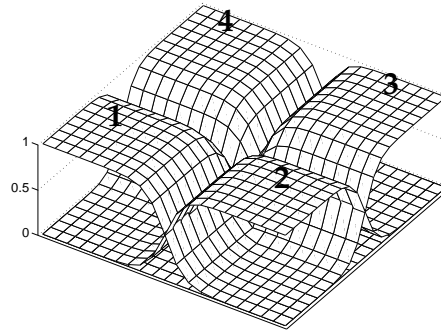
Figure 14.1: Example of operating regime decomposition. The horizontal plane represents the operating range of the system under consideration. The operating range is partitioned into four operating regimes. In each operating regime the system is described by a different local model. The weighting functions that are shown, indicate the validity of each of the local models. In each operating regime only one model is active. The transitions between the operating regimes are smooth.

The remainder of this chapter is organized as follows: In Section 14.1 a precise description of the local linear model structure is given for both state-space and input-output descriptions. An overview of the literature on local linear model approaches for identification is provided in Section 14.2. Some properties of the local linear state-space description that will be studied in this part, are given in Section 14.3. Section 14.4 summarizes the assumptions made throughout the following chapters. Finally, Section 14.5 presents a detailed outline of the remaining part of the thesis.

14.1   Local Linear Models for Nonlinear Systems

As mentioned above, a weighted combination of linear models can be used to approximate a nonlinear system. The linear models can be interpreted as linearizations of the nonlinear system in a number of different operating points, the weights combine these models to describe the transitions between the operating points. It is possible to apply the local modeling approach to both state-space and input-output models. Although many authors have considered identification of local linear input-output models —as will be clear from Section 14.2— the input-output models will only be briefly discussed. The main focus is on state-space models.

Why are state-space models considered? Although input-output models can approximate a broad class of nonlinear systems, they are not suitable for many dynamic analysis and control methods. Many existing nonlinear control methods are based on state-space systems. Furthermore, state-space systems are more attractive for dealing with multivariable inputs and outputs. As argued by Rivals and Personnaz (1996), and already mentioned in Chapter 1, state-space systems are likely to require fewer parameters, especially for multivariable systems. One

of the reasons is that the number of time lagged inputs and outputs M used in the input-output description

$$y_k = h(y_{k-1}, y_{k-2}, \ldots, y_{k-M}, u_k, u_{k-1}, \ldots, u_{k-M}),$$

with h a smooth map, satisfies $n \leq M \leq 2n+1$ with n the order of the state-space realization of this system, provided it exists (Stark et al. 1997; Stark 1999).

Despite these advantages, the major drawback of state-space systems is that they are recurrent (Nerrand et al. 1993). It is well-known that the identification of recurrent models can be troublesome (Haykin 1999; Nerrand et al. 1994). It is shown in Chapter 15 that some of the problems associated with identification of recurrent networks can be overcome for the local linear model description by exploiting its special structure.

The discussion focuses on local linear models that are smoothly interpolated by weighting functions to describe a global model. Other approaches exist in which switching between local models is used (for example Narendra, Balakrishnan, and Ciliz 1995; Schott and Bequette 1997), instead of smooth interpolation. In these approaches only one model is active at a certain time instant, not a combination of models, as can be the case with interpolation. Approaches in which no global description is created have also been described (for example Skeppstedt, Ljung, and Millnert 1992; Stenman 1999). Such approaches only estimate the local models and do not bother about combining them into a global model. None of these approaches is discussed any further.

14.1.1   State-Space Models

A nonlinear dynamical state-space system is described by

$$
\begin{aligned}
x_{k+1} &= f(x_k, u_k), & (14.1)\\
y_k &= g(x_k, u_k), & (14.2)
\end{aligned}
$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ the input, $y_k \in \mathbb{R}^\ell$ the output, and f and g are continuous functions with certain smoothness properties. The idea is to approximate this system with local linear models in a certain bounded region of the space $\mathbb{R}^{n+m}$ of all possible states and inputs. This region is called the *operating range* of the system. It contains the values of the state and input that are considered relevant to the model. The operating range is divided into a certain number of subsets in which a local linear model is believed to be a reasonable approximation of the system. These subsets are called the *operating regimes* of the system. For example, in a chemical plant different operating regimes can correspond to the production of different products, or to different modes of operation like start-up and shut-down. The global approximation of the system is obtained by smooth interpolation between the operating regimes.

To characterize the operating regime in which the system is working, the *scheduling vector* ϕ_k is introduced. It is a function of the state and the input, that is, $\phi_k = \psi(x_k, u_k)$. The name stems from the fact that it schedules the system from one operating regime to the other.

The system (14.1)–(14.2) is divided into s distinct operating regimes as follows:

$$x_{k+1} \quad = \quad \sum_{i=1}^{s} p_i(\phi_k) f(x_k, u_k),$$

$$y_k \quad = \quad \sum_{i=1}^{s} p_i(\phi_k) g(x_k, u_k),$$

where $p_i(\phi_k)$ are weighting functions with a local support that satisfy

$$0 \le p_i(\phi_k) \le 1, \quad \sum_{i=1}^{s} p_i(\phi_k) = 1, \qquad i = 1, 2, \ldots, s. \tag{14.3}$$

For the ith operating regime, the corresponding weighting function $p_i(\phi_k)$ is close to one; the other weighting functions are close to zero. In the transitions between the operating regimes, the weights smoothly vary from approximately zero to approximately one and vice versa (see Figure 14.1 on page 189).

In every operating regime an *operating point* is chosen around which the function f is linearized. Let the operating point for the ith regime be $(x^{(i)}, u^{(i)})$, then a first-order Taylor series expansion is

$$f(x_k, u_k) \approx f(x^{(i)}, u^{(i)}) + A_i \cdot (x_k - x^{(i)}) + B_i \cdot (u_k - u^{(i)}),$$

with

$$A_i \quad := \quad \frac{\partial f}{\partial x_k^T}(x^{(i)}, u^{(i)}),$$

$$B_i \quad := \quad \frac{\partial f}{\partial u_k^T}(x^{(i)}, u^{(i)}).$$

The constant terms for each local model are collected and called the *offsets*

$$O_i := f(x^{(i)}, u^{(i)}) - A_i x^{(i)} - B_i u^{(i)}.$$

An approximation of (14.1) with local linear models is

$$x_{k+1} \approx \sum_{i=1}^{s} p_i(\phi_k)(A_i x_k + B_i u_k + O_i).$$

A similar first-order Taylor expansion can be made for the function g in each operating regime; this yields

$$y_k \approx \sum_{i=1}^{s} p_i(\phi_k)(C_i x_k + D_i u_k + P_i),$$

where

$$C_i \quad := \quad \frac{\partial g}{\partial x_k^T}(x^{(i)}, u^{(i)}),$$

$$D_i \quad := \quad \frac{\partial g}{\partial u_k^T}(x^{(i)}, u^{(i)}),$$

$$P_i \quad := \quad g(x^{(i)}, u^{(i)}) - C_i x^{(i)} - D_i u^{(i)}.$$

Note that the approximation of g has been based on the same operating points and regimes as used for f. This is not necessary; they can be different, but keeping the same operating regimes facilitates the interpretation of the local linear model structure.

Now a model for the system (14.1)–(14.2) is

$$\widehat{x}_{k+1} = \sum_{i=1}^{s} p_i(\phi_k)(A_i\widehat{x}_k + B_i u_k + O_i), \qquad (14.4)$$

$$\widehat{y}_k = \sum_{i=1}^{s} p_i(\phi_k)(C_i\widehat{x}_k + D_i u_k + P_i). \qquad (14.5)$$

This model is a collection of local linear approximations. It has been shown by Johansen, Shorten, and Murray-Smith (2000) that this model is a universal approximator to the system (14.1)–(14.2), in the sense that for a sufficiently large number of local models an arbitrarily small modeling error can be achieved on a compact subset of the state and input space. Based on this property it is also possible to view the system (14.4)–(14.5) as a black-box model for the system (14.1)–(14.2); it is not necessary to view the local models as local linearizations of the nonlinear system. If the weights are parameterized with some basis functions, for example, radial basis functions, the global model (14.4)–(14.5) can be viewed as a neural network with a particular structure.

Although the interpretation of local linearizations is attractive, it should be exercised with care, as pointed out by Shorten et al. (1999), Leith and Leithhead (1998, 1999), and Johansen, Shorten, and Murray-Smith (2000). The main problem is that in regions of the operating space where not one model dominates, but where multiple models are interpolated to provide an accurate description of the system, the dynamics of the local models are not directly related to the dynamics of the nonlinear system to be approximated.

It might be better to view (14.4)–(14.5) as a neural network with a particular structure or as a linear parameter-varying system in which the time-varying parameters are the weighting functions. The latter interpretation is given in Section 14.3.

14.1.2   Input-Output Models

The local linear modeling approach can also be used to approximate an input-output representation of a nonlinear dynamic system, like

$$y_k = h(y_{k-1}, y_{k-2}, \ldots, y_{k-L}, u_k, u_{k-1}, \ldots, u_{k-M}).$$

Such input-output descriptions are widely used in the identification of nonlinear systems (Sjöberg et al. 1995). It is commonly known as the NARMAX model (Chen and Billings 1989). The input-output description can also be approximated by local linear models (Billings and Voon 1987; Johansen and Foss 1993, 1995). Similar to the state-space approach, a division into s operating regimes can be made:

$$y_k = \sum_{i=1}^{s} p_i(\phi_k) h(y_{k-1}, y_{k-2}, \ldots, y_{k-L}, u_k, u_{k-1}, \ldots, u_{k-M}).$$

Linearization in the operating points

$$\left(y_{k-1}^{(i)}, y_{k-2}^{(i)}, \ldots, y_{k-L}^{(i)}, u_k^{(i)}, u_{k-1}^{(i)}, \ldots, u_{k-M}^{(i)}\right), \quad i = 1, 2, \ldots, s$$

yields

$$y_k \approx \sum_{i=1}^{s} p_i(\phi_k) \Big(\mathcal{B}_i(q) u_k - \mathcal{A}_i(q) y_k + \mathcal{O}_i \Big),$$

with $\mathcal{O}_i \in \mathbb{R}^\ell$ the offsets, and

$$
\begin{aligned}
\mathcal{A}_i(q) &:= \sum_{i=1}^{L} \alpha_i q^{-i}, \\
\mathcal{B}_i(q) &:= \sum_{i=0}^{M} \beta_i q^{-i},
\end{aligned}
$$

where $\alpha_i \in \mathbb{R}^{\ell \times \ell}$, $\beta_i \in \mathbb{R}^{\ell \times m}$, and q is the shift operator: $q^{-n} u_k = u_{k-n}$.

Note that the identification of these local input-output models is fundamentally different from the identification of the state-space model (14.4)–(14.5). Of course the local linear input-output models can be converted to local state-space models, but there is no guarantee that these local state-space models use the same coordinates for the state vector. Therefore, the states of these local models cannot be combined directly as in (14.4)–(14.5). The only way to combine their states is by stacking them into one big state vector. Of course the model obtained in this way is not observable and not controllable. Model reduction is not trivial, because the model is nonlinear: the system matrices depend on the scheduling vector.

14.2 Overview of Identification Methods

A comprehensive introduction to multiple model approaches can be found in the book edited by Murray-Smith and Johansen (1997b). The first chapter of this book (Johansen and Murray-Smith 1997) contains an extensive review of the literature related to multiple model approaches and operating regime decomposition up to 1997. It provides an overview of local methods for learning and modeling and methods for structure identification. It also contains an overview of control methods, including gain scheduling, multiple model adaptive control, fuzzy logic control, and global model-based control strategies like nonlinear model-based predictive control.

The current section highlights some contributions to the field that appeared after 1996 and are not mentioned in the book of Murray-Smith and Johansen (1997b).

Banerjee et al. (1997) presented an approach to determine the weights that combine several local state-space models into a global description of the nonlinear system. The weights are treated as parameters of the global model, and their time variation is treated as a random walk process. The estimation of the weights and the state of the global model is carried out by a moving-horizon Bayesian estimator that can deal with constraints on the weights. A similar approach was

presented by Verdult and Verhaegen (2001a). They used an iterative procedure to estimate both the weights and the local models. One step of this iterative procedure consists of estimating the weights and the state of the global model using the current estimates of the local models; the other step consists of updating the local models using the current estimate of the weights.

Cao, Rees, and Feng (1997) proposed a fuzzy local linear input-output model different from the classical Takagi-Sugeno fuzzy model. It can be written as a global linear parameter-varying system in which the time-varying parameters are the fuzzy membership functions. They argued that this system has some advantages for control design, because it can easily be converted into an LPV state-space system. They discussed the identification of this model structure in which the operating regimes are determined by fuzzy clustering. A modified fuzzy clustering algorithm was proposed that takes into account the dynamic structure of the model.

Another approach to fuzzy input-output modeling was discussed by Boukhris, Mourot, and Ragot (1999). Instead of using fuzzy clustering, the membership functions are parameterized. The parameters describing the membership functions and the parameters describing the local models are determined by solving a nonlinear optimization problem. They suggested a modification of the Lipschitz quotient of He and Asada (1993) to determine the order of the local models.

Nelles, Hecker, and Isermann (1997) developed a method called LOLIMOT (LOcal LInear MOdel Trees) to determine the structure of a local linear input-output model description from empirical data. It is a tree algorithm that in combination with subset regression techniques determines a partition in operating regimes, the number of local models, and the orders and dead-times of the local models. It was applied for modeling a heat exchanger (Fischer, Nelles, and Isermann 1998) and the combustion process of a diesel engine (Hafner et al. 2000). The main drawback of the LOLIMOT algorithm is that it only allows operating regime decompositions parallel to the scheduling variables. This restriction is removed in a related tree algorithm presented by Ernst (1998) that uses hinging hyperplanes (Breiman 1993; Pucar and Sjöberg 1998) to construct local models.

Shorten et al. (1999), Johansen, Shorten, and Murray-Smith (2000) and Leith and Leithhead (1998, 1999) pointed out that the interpretation of local linear model dynamics should be done with care, it is not as straightforward as it might initially appear. Leith and Leithhead (1998, 1999) proposed to use velocity-based linearizations as an alternative to the conventional local linearization approach. Their approach facilitates the dynamic analysis, but is only valid for continuous-time systems.

Several authors have studied the identification of local linear models in combination with model-based predictive control, among them are: Fischer, Nelles, and Isermann (1998); Townsend et al. (1998); Zhang and Morris (1999); Roubos et al. (1999); Lakshmanan and Arkun (1999); and Azimzadeh, Galán, and Romagnoli (2001). Choi and Farrell (2000) used local linear models to identify and model uncertainty on-line in a nonlinear adaptive control method. Hunt et al. (2000) discussed the identification of local linear models and a generalized gain-scheduling approach for high-precision speed control of a vehicle that was implemented in real-time and experimentally verified.

Most authors dealt with the identification of local input-output models. In this way they circumvent some problems that arise when looking at state-space models of the form (14.4)–(14.5). These problems are discussed in detail in Chapter 15 and possible solutions are given there. One important problem that arises with state-space models is that it is not possible to determine them uniquely from input and output data. Reason being that by applying a similarity transformation to the state, a whole set of state-space systems can be created that have the same input-output behavior (see also Subsection 2.3.1). This issue can be solved by choosing a special parameterization for the state-space system, such that it can be uniquely determined. This is the approach followed by most authors. However, such parameterizations are often not numerically reliable. Therefore, in Chapter 15 a different approach is taken to deal with the nonuniqueness of the state-space system.

14.3   System Analysis

The local linear model structure

$$x_{k+1} \;=\; \sum_{i=1}^{s} p_i(\phi_k)(A_i x_k + B_i u_k + O_i), \qquad (14.6)$$

$$y_k \;=\; \sum_{i=1}^{s} p_i(\phi_k)(C_i x_k + D_i u_k + P_i), \qquad (14.7)$$

is closely related to the LPV system that was discussed in Part I of this thesis:

$$x_{k+1} \;=\; A_0 x_k + B_0 u_k + \sum_{i=1}^{s} [p_k]_i (A_i x_k + B_i u_k),$$

$$y_k \;=\; C_0 x_k + D_0 u_k + \sum_{i=1}^{s} [p_k]_i (C_i x_k + D_i u_k).$$

The system (14.6)–(14.7) is an LPV system, which has as input $[u_k^T, 1]^T$, and has time-varying parameters $[p_k]_i = p_i(\phi_k)$, $i = 1, 2, \ldots, s$, and system matrices $A_0 = 0$, $B_0 = 0$, $C_0 = 0$ and $D_0 = 0$.

Although the local linear model structure can be written as an LPV system, the time-varying parameter is assumed to be unknown, and can depend on the input and the state. In contrast, the time-varying parameters of the LPV systems considered in Part I were assumed to be known and independent of the state and the output. This is the main difference between the identification problem discussed in Part I and the identification problem discussed in the forthcoming chapters.

Stability of the system (14.6)–(14.7) can be defined in the same way as for the LPV systems in Part I. The definitions below are adopted from Subsection 2.3.2.

Definition 14.1 *The state equation*

$$x_{k+1} = \sum_{i=1}^{s} p_i(\phi_k) A_i x_k, \qquad (14.8)$$

with $p_i(\phi_k)$ satisfying (14.3) and arbitrary vectors ϕ_k is called uniformly exponentially stable if there exist a finite constant $\gamma > 0$ and a constant λ, $0 \le \lambda < 1$ such that for every $j \le k$ and every x_j the corresponding solution satisfies

$$\|x_k\|_2 \le \gamma \lambda^{k-j} \|x_j\|_2.$$

Definition 14.2 *The system (14.6)–(14.7) with $p_i(\phi_k)$ satisfying (14.3) and arbitrary vectors ϕ_k is called uniformly bounded-input, bounded-output stable if there exists a finite constant η such that for any j and any input signal u_k the corresponding response y_k for $x_j = 0$ satisfies*

$$\sup_{k \ge j} \|y_k\|_2 \le \eta \sup_{k \ge j} \|u_k\|_2.$$

In general, it is difficult to derive stability results for the system (14.6)–(14.7), because the scheduling vector ϕ_k depends on the state. For the special case where the scheduling vector ϕ_k only depends on the input of the system, the stability results of Corollaries 2.1 and 2.2, discussed in Subsection 2.3.2 for LPV systems, can be used for the system (14.6)–(14.7). For convenience these results are reformulated below.

Corollary 14.1 *The linear state equation (14.8) with $p_i(\phi_k)$ satisfying (14.3) and $\phi_k = \psi(u_k)$ is uniformly exponentially stable if there exist a finite constant $\gamma > 0$ and a constant λ, $0 \le \lambda < 1$ such that*

$$\left\| \prod_{t=j}^{k-1} \left(\sum_{i=1}^{s} p_i(\phi_t) A_i \right) \right\|_2 \le \gamma \lambda^{k-j},$$

for all u_k, and all k, j such that $k > j$.

Corollary 14.2 *If the state equation (14.8) with $p_i(\phi_k)$ satisfying (14.3) and $\phi_k = \psi(u_k)$ is uniformly exponentially stable, then the corresponding system (14.6)–(14.7) is uniformly bounded-input, bounded-output stable.*

14.4 Assumptions

The subsequent chapters deal with the identification of the model (14.4)–(14.5) from measurements of the inputs and the outputs. A standing assumption is that the system (14.6)–(14.7) is uniformly bounded-input, bounded-output stable and observable (note that observability can be defined in the same way as for the LPV system in Part I). It is also assumed that any stochastic disturbances acting on the system are statistically independent of the input signal and that these disturbances, as well as the input signal, are bounded.

It is assumed that the structure of the model (14.4)–(14.5) is known, that is, the number of local models and the order of the models are known. It is also assumed that it is known which variables to use as the scheduling vector ϕ_k. Structure determination of the local linear model description is a challenging problem. To solve it, the use of prior information on the nonlinear system is essential (Johansen 1994; Johansen and Foss 1995; Johansen and Murray-Smith 1997). Structure identification will not be addressed; only some directions for further research are given in Chapter 17.

14.5   Outline of Part III

This part consists of four chapters, including this one. Chapter 15 discusses an identification method for local linear state-space models based on nonlinear optimization. This method uses fully parameterized local linear models. It deals with the nonuniqueness of the state-space system by only updating the parameters in directions that change the cost function. Section 16 illustrates the method with several examples. Conclusions and recommendations for further research are found in Chapter 17.

Chapter 15

Identification Based on Nonlinear Optimization

Local linear modeling is one of the many possibilities to approximate a nonlinear dynamical system. It is an attractive approach, because the local models can be interpreted as linearizations of the nonlinear system in different operating points. By making a weighted combination of these linear models, one hopes to describe the complete nonlinear behavior sufficiently accurately (Murray-Smith and Johansen 1997b). In this chapter, the goal is to determine a weighted combination of linear models from a finite number of measurements of the input and output of the system. This is the so-called black-box identification approach (Sjöberg et al. 1995). Black-box identification of local linear model structures has been studied mainly for input-output systems and for state-space systems of which the full state vector is measured (an overview of the literature is given in Section 14.2). The case where only part of the state is measured is of course of more general interest. This chapter discusses the identification of local linear state-space models where the state is not measured directly, only an observable linear combination of some of the states is available as an output.

The discussion focuses on systems of the form

$$x_{k+1} \quad = \quad \sum_{i=1}^{s} p_i(\phi_k)\Big(A_i x_k + B_i u_k + O_i\Big), \tag{15.1}$$

$$y_k \quad = \quad C x_k + v_k, \tag{15.2}$$

where s is the number of local models, $x_k \in \mathbb{R}^n$ the unknown state, $u_k \in \mathbb{R}^m$ the input, $y_k \in \mathbb{R}^\ell$ the output, $p_i(\phi_k) \in \mathbb{R}$ the weighting for the ith model, and $v_k \in \mathbb{R}^\ell$ an unknown white-noise sequence that is independent of u_k. The weighting vectors p_i are unknown functions of the scheduling vector $\phi_k \in \mathbb{R}^q$. This scheduling vector corresponds to the operating point of the system, typically it will depend on the input and the state, that is,

$$\phi_k = \psi(x_k, u_k),$$

199

with $\psi : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^q$, which is assumed to be a known, possibly nonlinear function. It is also assumed that the number of local models and the order of the system are known.

It is common practice in local linear modeling to use normalized radial basis functions for the weights $p_i(\phi_k)$. Let the ith radial basis function be equal to

$$r_i(\phi_k; c_i, w_i) = \exp\left(-(\phi_k - c_i)^T \operatorname{diag}(w_i)^2 (\phi_k - c_i)\right),$$

where c_i is called the *center* and w_i is called the *width* of the ith radial basis function, then the ith weight is taken as

$$p_i(\phi_k; c_i, w_i) = \frac{r_i(\phi_k; c_i, w_i)}{\sum\limits_{j=1}^{s} r_j(\phi_k; c_j, w_j)}. \tag{15.3}$$

The normalization ensures that the weights are always between zero and one, and that they sum up to one at every time instant. Normalization is common practice in local linear modeling, although some subtle problems can arise, as discussed by Shorten and Murray-Smith (1997). They show that with normalization, the weights no longer need to decrease monotonically as the distance from their center increases. This means that the weights can 'reactivate', that is, have more than one region of significant activity. It is important to realize that with some minor modifications, the identification method presented below can also be used in case the weights are radial basis functions that have not been normalized.

The goal is to determine from a finite number of measurements of the input u_k and output y_k, the matrices A_i, B_i, O_i, and C that describe the local models; and the centers c_i and widths w_i that describe the radial basis functions. To determine these parameters, the following cost function is minimized:

$$J_N(\theta, c, w) := \sum_{i=1}^{N} \|y_k - \hat{y}_k(\theta, c, w)\|_2^2 = E_N(\theta, c, w)^T E_N(\theta, c, w),$$

where

$$E_N(\theta, c, w)^T = \left[\left(y_1 - \hat{y}_1(\theta, c, w)\right)^T \ \left(y_2 - \hat{y}_2(\theta, c, w)\right)^T \ \cdots \ \left(y_N - \hat{y}_N(\theta, c, w)\right)^T\right],$$

the vectors c and w consist of the centers c_i and w_i, respectively; θ is a set of parameters that describes the system matrices of the local models, A_i, B_i, O_i, and C; and $\hat{y}_k(\theta, c, w)$, is the output of the model

$$x_{k+1}(\theta, c, w) = \sum_{i=1}^{s} p_i(\phi_k; c, w)\left(A_i(\theta)x_k(\theta, c, w) + B_i(\theta)u_k + O_i(\theta)\right), \tag{15.4}$$

$$\hat{y}_k(\theta, c, w) = C(\theta)x_k(\theta, c, w). \tag{15.5}$$

The minimization of J_N is a nonlinear, nonconvex problem, because of the complicated dependence of J_N on the parameters θ, c, and w. In the neural network literature, solving this optimization problem is referred to as *training* the

model. A commonly used technique to train recurrent networks is the Levenberg-Marquardt method (Moré 1978; Coleman, Branch, and Grace 1999), which is an iterative procedure that updates the parameters as follows:

$$
\begin{bmatrix} \theta^{(i+1)} \\ c^{(i+1)} \\ w^{(i+1)} \end{bmatrix} = \begin{bmatrix} \theta^{(i)} \\ c^{(i)} \\ w^{(i)} \end{bmatrix} - \mu^{(i)} H_N\left(\lambda^{(i)}, \theta^{(i)}, c^{(i)}, w^{(i)}\right)^{-1}
$$
$$
\times \Psi_N^T\left(\theta^{(i)}, c^{(i)}, w^{(i)}\right) E_N\left(\theta^{(i)}, c^{(i)}, w^{(i)}\right), \quad (15.6)
$$

where $\mu^{(i)}$ is the step size,

$$
H_N(\lambda, \theta, c, w) := \lambda I + \Psi_N^T(\theta, c, w)\Psi_N(\theta, c, w)
$$

is the approximate Hessian, $\lambda^{(i)}$ is the Levenberg-Marquardt regularization parameter and

$$
\Psi_N(\theta, c, w) := \begin{bmatrix} \dfrac{\partial E_N(\theta, c, w)}{\partial \theta^T} & \dfrac{\partial E_N(\theta, c, w)}{\partial c^T} & \dfrac{\partial E_N(\theta, c, w)}{\partial w^T} \end{bmatrix}
$$

is the gradient.

The model (15.4)–(15.5) can be viewed as a recurrent state-space neural network (Haykin 1999; Narendra and Parthasarathy 1990) with a special structure. Since recurrent networks are difficult to train, special structures have been suggested that can be exploited to facilitate training. An example of this is the radial basis function network proposed by Gan and Danai (2000). The model that is considered in this chapter, is another attempt to exploit special structure.

Training of the recurrent model (15.4)–(15.5) is difficult, because of three main reasons.

First, the state-space representation is not unique; there exist different state-space models that have the same input-output behavior. For the model (15.4)–(15.5), it is readily seen that the same input-output behavior is obtained by taking $\widetilde{x}_k = T^{-1}x_k$ (T nonsingular), $\widetilde{A}_i = T^{-1}A_iT$, $\widetilde{B}_i = T^{-1}B_i$, $\widetilde{O}_i = T^{-1}O_i$ and $\widetilde{C} = CT$. One way to deal with this nonuniqueness is to use an appropriate parameterization that makes the state-space representation unique. However, such representations may have bad numerical properties; therefore, in Section 15.1, it is proposed to use fully parameterized system matrices and a method is described that deals with the similarity transformation T during training. At each iteration of the training process, this method determines the directions in the parameter space that do not change the cost function, and does not update the parameters in this direction by using an appropriate projection for the gradient in (15.6). In this way the active parameters, the ones that change the cost function, are determined from the data; this leads to better numerical conditioning of the optimization algorithm. The method is based on ideas used for identification of linear and linear parameter-varying state-space systems by McKelvey and Helmersson (1997), Lee and Poolla (1999), and Verdult and Verhaegen (2001c). It is similar to the LPV identification method in Chapter 5.

The second reason that makes training difficult, is the complicated dependence of the gradient on the past data. It follows from (15.4)–(15.5) that the gradient at

time instant k with respect to θ, c, and w depends on the entire past of the state sequence x_k. This dynamic dependence may lead to stability problems during training. Sections 15.2 and 15.3 show that for certain special choices of the scheduling vector ϕ_k, attractive training methods exist that do not suffer from stability problems.

   If the model is used to approximate a nonlinear system, there is a third reason that makes training difficult. The original system can be approximated by different models of the type (15.4)–(15.5), equally well. For the model that is considered in this chapter, a simple example is to shift the operating points. The nonuniqueness of the model can lead to ill-conditioning during training. This problem has been discussed by Murray-Smith and Johansen (1997a) and Shorten et al. (1999). Remedies are local learning (Murray-Smith and Johansen 1997a; Bottou and Vapnik 1992), regularization and early stopping (Haykin 1999; Sjöberg and Ljung 1995; Johansen 1996, 1997). Although we do not discuss these techniques in this chapter, they can all be applied to train the model (15.4)–(15.5).

   This chapter consists of four sections. Section 15.1 deals with the problem of nonuniqueness of the state-space representation. Sections 15.2 and 15.3 are devoted to some special scheduling cases where the structure of the system can be exploited to facilitate the nonlinear optimization associated with the identification procedure. All these sections deal with the case of measurement noise at the output; in Section 15.4 the extension for a special kind of process noise is discussed.

15.1   Projected Gradient Search

This section presents a projected gradient search identification method for the system (15.4)–(15.5). For ease of notation, the following matrices are introduced:

$$
\begin{aligned}
\overline{A} &:= \begin{bmatrix} A_1 & A_2 & \dots & A_s \end{bmatrix}, \\
\overline{B} &:= \begin{bmatrix} B_1 & B_2 & \dots & B_s \end{bmatrix}, \\
\overline{O} &:= \begin{bmatrix} O_1 & O_2 & \dots & O_s \end{bmatrix}, \\
\overline{C} &:= \begin{bmatrix} C & 0 & \dots & 0 \end{bmatrix}.
\end{aligned}
$$

The matrix

$$
\Theta := \begin{bmatrix} \overline{A} & \overline{B} & \overline{O} \\ \overline{C} & 0 & 0 \end{bmatrix}
$$

is used to define the parameters θ that describe the local linear models as

$$
\theta := \Xi \, \mathrm{vec}(\Theta),
$$

where Ξ is a selection matrix such that the entries of Θ that are zero by definition (including the zeros in $\overline{C}$) do not appear in θ. Since θ contains all the entries of the matrices $\overline{A}$, $\overline{B}$, $\overline{O}$, and C, it is clear that a full parameterization is used for the system matrices that describe the local linear models. With this approach, it is important to deal with similarity transformations of the state in an appropriate way. Since such transformations do not change the input-output behavior of the system, they can cause trouble with the update rule (15.6). Below it is explained

how to avoid such problems by using a projected gradient search algorithm. The discussion is similar to the one in Chapter 5.

The similarity map $S_\Theta : T \in \mathbb{R}^{n \times n}, \det(T) \neq 0 \to \mathbb{R}^{(n+\ell) \times (n+m+1)s}$, which is defined as

$$S_\Theta(T) := \begin{bmatrix} T^{-1} & 0 \\ 0 & I_\ell \end{bmatrix} \begin{bmatrix} \overline{A} & \overline{B} & \overline{O} \\ \overline{C} & 0 & 0 \end{bmatrix} \begin{bmatrix} T_s & 0 & 0 \\ 0 & I_{ms} & 0 \\ 0 & 0 & I_s \end{bmatrix}, \qquad (15.7)$$

with T_s a block diagonal matrix with s copies of T along the diagonal, can be used to characterize the similarity transformations. Taking a certain model Θ, all models with the same input-output behavior are given by

$$\mathcal{I}_\Theta := \left\{ \overline{\Theta} \,|\, \overline{\Theta} = S_\Theta(T), \ \det(T) \neq 0 \right\}.$$

This set is called the *indistinguishable set at* Θ, because it contains all the models that cannot be distinguished from Θ by looking at their input-output behavior.

If the similarity map S_Θ is locally one-to-one at I_n, that is, if $S_\Theta(T) = \Theta$ implies $T = I_n$, the connected component of $\mathcal{I}_\Theta$,

$$\mathcal{I}_\Theta^+ := \left\{ \overline{\Theta} \,|\, \overline{\Theta} = S_\Theta(T), \ \det(T) > 0 \right\},$$

is a differentiable manifold (Lee and Poolla 1999). The tangent space of this manifold at the point Θ contains the directions along which a change in the parameters θ does not influence the value of the cost function $J_N(\theta, c, w)$ (see Figure 5.1 on page 79). It does not make sense to change the parameters in these directions during the iterative optimization. Therefore, the update rule (15.6) is modified such that the parameters θ are not changed along these directions. For this modification, the tangent space of the manifold $\mathcal{I}_\Theta^+$ needs to be determined.

To determine the tangent space of $\mathcal{I}_\Theta^+$, it is assumed that the similarity map is locally one-to-one. The following lemma provides some possibly conservative conditions on the system matrices that guarantee this:

Lemma 15.1 *Let the observability matrix for A_i and C be given by*

$$\mathcal{O}_{n,i} := \begin{bmatrix} C \\ C A_i \\ C A_i^2 \\ \vdots \\ C A_i^{n-1} \end{bmatrix}.$$

If the matrix $\mathcal{O}_{n,i}$ has full rank for some $i \in \{1, 2, \ldots, s\}$, then $S_\Theta(T) = \Theta$ implies $T = I_n$.

The proof of this lemma is similar to the proof of Lemma 5.1 on page 79, and appeared in a different form in the work of Lee and Poolla (1999). It can also be shown that $S_\Theta(T) = \Theta$ implies $T = I_n$ if the observability condition for A_i and C is replaced with a controllability condition for A_i and B_i.

The tangent space of the manifold $\mathcal{I}_\Theta^+$ at a certain point determined by T, is a linear approximation to the manifold $\mathcal{I}_\Theta^+$ at that particular point. It can hence be

determined by linearizing the similarity map $S_\Theta(T)$ (compare equation (5.10) on page 80):

$$
\begin{aligned}
S_\Theta(T) &\approx S_\Theta(I_n + T) - S_\Theta(I_n) \\
&\approx \left[\begin{array}{c} \overline{A} \\ \hline \overline{C} \end{array} \right] \left[\begin{array}{cc} T_s & 0_{n \times (m+1)s} \end{array} \right] - \left[\begin{array}{c} T \\ 0_{\ell \times n} \end{array} \right] \left[\begin{array}{ccc} \overline{A} & \overline{B} & \overline{O} \end{array} \right], \quad (15.8)
\end{aligned}
$$

where the last expression is obtained by approximating $(I_n + T)^{-1}$ by $I_n - T$ and neglecting all second-order terms. This equation can also be written as

$$
S_\Theta(T) \approx \sum_{i=1}^{s} \left[\begin{array}{c} \overline{A} \\ \hline \overline{C} \end{array} \right] \Pi_i^T T \left[\begin{array}{cc} \Pi_i & 0_{n \times (m+1)s} \end{array} \right] - \left[\begin{array}{c} I_n \\ 0_{\ell \times n} \end{array} \right] T \left[\begin{array}{ccc} \overline{A} & \overline{B} & \overline{O} \end{array} \right],
$$

where

$$
\Pi_i := \left[\begin{array}{ccc} 0_{n \times (i-1)n} & I_n & 0_{n \times (s-i)n} \end{array} \right];
$$

or as

$$
\mathrm{vec}\left(S_\Theta(T) \right) \approx M_\Theta \mathrm{vec}(T), \quad (15.9)
$$

where

$$
M_\Theta := \sum_{i=1}^{s} \left[\begin{array}{c} \Pi_i^T \\ 0_{(m+1)s \times n} \end{array} \right] \otimes \left[\begin{array}{c} \overline{A}\Pi_i^T \\ \overline{C}\Pi_i^T \end{array} \right] - \left[\begin{array}{c} \overline{A}^T \\ \overline{B}^T \\ \overline{O}^T \end{array} \right] \otimes \left[\begin{array}{c} I_n \\ 0_{\ell \times n} \end{array} \right],
$$

and the property $\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B)$ has been used.

Equation (15.9) shows that the tangent space of the manifold $\mathcal{I}_\Theta^+$ at the point Θ equals the column space of the matrix M_Θ. Since the left null space of the matrix M_Θ equals the orthogonal complement of the tangent space of the manifold $\mathcal{I}_\Theta^+$ at the point Θ, the left null space contains the directions in which the parameters should be changed to obtain a change in the cost function $J_N(\theta, c, w)$. In general, the matrix M_Θ has full rank, as shown by the next lemma, which is similar to Lemma 5.2 on page 80.

Lemma 15.2 *The similarity map S_Θ, given by (15.7), is locally one-to-one, that is, the equality $S_\Theta(I_n + T) = \Theta$ implies $T = 0$, if and only if the matrix M_Θ has full rank.*

Note that it is sufficient to look at ΞM_Θ, because the zero entries in Θ do not appear in θ. Hence, if $\overline{\theta} = \Xi \mathrm{vec}(S_\Theta(T))$, then

$$
\overline{\theta} \approx \Xi M_\Theta \mathrm{vec}(T).
$$

The singular value decomposition

$$
\Xi M_\Theta = \left[\begin{array}{cc} U_1(\theta) & U_2(\theta) \end{array} \right] \left[\begin{array}{c} \Sigma(\theta) \\ 0 \end{array} \right] V^T(\theta) \quad (15.10)
$$

can be used to decompose every parameter vector θ into two parts:

$$
\theta = U_1(\theta)U_1^T(\theta)\theta + U_2(\theta)U_2^T(\theta)\theta,
$$

where the first part corresponds to the directions that do not influence the cost function (column space of ΞM_Θ), and the second part to the directions that change the value of the cost function (left null space of ΞM_Θ). Based upon this observation, the update rule for the parameters (15.6) is changed such that the update is restricted to the directions that change the cost function:

$$
\begin{bmatrix} \theta^{(i+1)} \\ c^{(i+1)} \\ w^{(i+1)} \end{bmatrix} = \begin{bmatrix} \theta^{(i)} \\ c^{(i)} \\ w^{(i)} \end{bmatrix} - \mu^{(i)} U_2(\theta) \widetilde{H}_N \left(\lambda^{(i)}, \theta^{(i)}, c^{(i)}, w^{(i)} \right)^{-1}
$$
$$
\widetilde{\Psi}_N^T \left(\theta^{(i)}, c^{(i)}, w^{(i)} \right) E_N \left(\theta^{(i)}, c^{(i)}, w^{(i)} \right),
$$

where

$$
\begin{aligned}
\widetilde{H}_N(\lambda, \theta, c, w) &:= \lambda I + \widetilde{\Psi}_N^T(\theta, c, w) \widetilde{\Psi}_N(\theta, c, w), \\
\widetilde{\Psi}_N(\theta, c, w) &:= \Psi_N(\theta, c, w) U_2(\theta).
\end{aligned}
$$

Since the matrix U_2 depends on the current parameters $\theta^{(i)}$, the singular value decomposition (15.10) must be computed at each iteration. However, this is not so demanding compared with the computation of the gradient.

15.2 Scheduling with the Input

As mentioned before, the recurrent nature of the model (15.4)–(15.5) can cause instability of the gradient calculations. This section shows that for the special case where the scheduling vector of the model (15.4)–(15.5) equals the input, $\phi_k = u_k$, such instability problems do not arise. For the case $\phi_k = u_k$, the recurrent nature of the model with respect to the weights disappears. The model is still recurrent in the state, but the weights p_i do not depend on the state and can be computed independently. Once the weights have been computed, the state sequence follows by simulating a linear parameter-varying system, where the weights $p_i(u_k)$ are the time-varying parameters:

$$
x_{k+1} = A(p_k)x_k + B(p_k)u_k + O(p_k), \tag{15.11}
$$

with $p_k := [p_1(u_k), p_2(u_k), \ldots, p_s(u_k)]$,

$$
A(p_k) := \sum_{i=1}^{s} A_i p_i(u_k),
$$

and similar definitions for $B(p_k)$ and $O(p_k)$. This greatly simplifies the gradient calculations, and hence the complexity of the training procedure.

Gradient calculation for the identification of linear parameter-varying systems has been described by Lee and Poolla (1999). It appears that the gradient of $E_N(\theta, c, w)$ with respect to θ can be obtained by simulating a set of linear time-varying state equations (see also Subsection 5.1.2). Note that to compute this

gradient, $\partial\widehat{y}_k/\partial\theta_j$ needs to be computed for every parameter θ_j, and every time instant $k = 1, 2, \ldots, N$. From the output equation (15.5) it follows that

$$\frac{\partial\widehat{y}_k}{\partial\theta_j} = C\mathcal{X}_k^{\theta_j} + \frac{\partial C}{\partial\theta_j}\,x_k, \tag{15.12}$$

with

$$\mathcal{X}_k^{\theta_j} := \frac{\partial x_k}{\partial\theta_j}, \tag{15.13}$$

which can be obtained for $k = 1, 2, \ldots, N$ by simulating the state equation

$$\mathcal{X}_{k+1}^{\theta_j} = \sum_{i=1}^{s} p_i(u_k)\left(A_i\mathcal{X}_k^{\theta_j} + \frac{\partial A_i}{\partial\theta_j}\,x_k + \frac{\partial B_i}{\partial\theta_j}\,u_k + \frac{\partial O_i}{\partial\theta_j}\right), \tag{15.14}$$

where x_k is the state of (15.11).

The gradients with respect to the parameters c and w that describe the radial basis functions are also easy to compute. The gradient for the center c_j of the jth basis function is obtained from

$$\frac{\partial\widehat{y}_k}{\partial c_j} = C\mathcal{X}_k^{c_j},$$

where $\mathcal{X}_k^{c_j}$ is obtained by simulating

$$\mathcal{X}_{k+1}^{c_j} = \sum_{i=1}^{s} p_i(u_k)A_i\mathcal{X}_k^{c_j} + \frac{\partial p_i(u_k)}{\partial c_j}\left(A_ix_k + B_iu_k + O_i\right), \tag{15.15}$$

with

$$\frac{\partial p_i(u_k)}{\partial c_j} = \begin{cases} \left(p_j(u_k) - p_j(u_k)^2\right)\text{diag}(w)(u_k - c_j) & \text{for} \quad i = j \\ -p_i(u_k)p_j(u_k)\text{diag}(w)(u_k - c_j) & \text{for} \quad i \neq j. \end{cases}$$

The expression for the gradient with respect to w is similar.

For successful training, the dynamic equations of the model (15.4) and the ones involved in the gradient calculation, that is, (15.14) and (15.15), have to be stable. The stability depends on both $p_i(\phi_k)$ and A_i ($i = 1, 2, \ldots, s$). Conditions for bounded-input, bounded-output stability are given in the following lemma:

Lemma 15.3 *If the system (15.4)–(15.5) with $\phi_k = u_k$ and $p_i(\phi_k)$ given by (15.3) is bounded-input, bounded-output stable, and for any q*

$$\sup_{k \geq q} \|u_k\|_2 < \infty,$$

then $\mathcal{X}_k^{\theta_j}$ given by (15.14), satisfies

$$\sup_{k \geq q} \|\mathcal{X}_k^{\theta_j}\|_2 < \infty,$$

for all j. If in addition

$$\sup_{k \geq q} \left\| \frac{\partial p_i(u_k)}{\partial c_j} \right\| < \infty,$$

for all j, then $\mathcal{X}_k^{c_j}$ given by (15.15), satisfies

$$\sup_{k \geq q} \| \mathcal{X}_k^{c_j} \|_2 < \infty,$$

for all j.

Proof: Application of Corollary 14.2 on page 196 to the system (15.4)–(15.5) shows that

$$\sup_{k \geq q} \| x_k \|_2 < \infty.$$

Since x_k and u_k are bounded and act as inputs in the state equations (15.14) and (15.15), application of Corollary 14.2 shows that both $\mathcal{X}_k^{\theta_j}$ and $\mathcal{X}_k^{c_j}$ are also bounded. □

From this lemma it follows that if the matrices A_i ($i = 1, 2, \ldots, s$) are such that the model (15.4)–(15.5) is bounded-input, bounded-output stable, then the recursions (15.14) and (15.15) for calculating the gradients are also bounded-input, bounded-output stable. Along the same lines it can be shown that the recursions for calculating the gradients with respect to the widths w are also bounded-input, bounded-output stable. If the training is initiated with a bounded-input, bounded-output stable model, it is unlikely that during the iterative optimization the system matrices A_i are modified such that the global system becomes unstable, because the parameters θ are determined using a descent method that always decreases the cost function. Therefore, the gradient calculations will be stable during training.

The optimization-based identification procedure described above needs an initial estimate of the model parameters. Because the optimization problem is nonlinear and recurrent, the choice of this initial estimate can influence the outcome considerably. A natural way to initialize the model (15.4)–(15.5) is to estimate a global linear state-space model and take all the local models equal to this linear model. Efficient subspace identification methods can be used for this (Verhaegen 1994; Van Overschee and De Moor 1996). Next, the radial basis functions are distributed uniformly over the range of the input u_k. In this way, the initial model equals the linear model and hence, the performance of the local linear models on the training data is always better than the performance of the global linear model. Using linear models for initialization has been proposed and motivated by Sjöberg (1997) and is also used by other authors, for example, Gan and Danai (2000).

Note that the training method outlined above, can easily be extended to deal with the case where $\phi_k = h(u_k, u_{k-1}, \ldots, u_{k-M})$ for some known nonlinear map h and integer M.

15.3   Scheduling with the Input and Output

Compared with the previous section, a more general model is obtained if the scheduling vector for (15.4)–(15.5) is taken equal to

$$\phi_k := \begin{bmatrix} u_k \\ Cx_k \end{bmatrix}. \tag{15.16}$$

The reason for considering Cx_k is that the output y_k is an estimate of this quantity. Actually, in the noise-free case we have $y_k = Cx_k$, hence the scheduling vector can be taken equal to

$$\phi_k := \begin{bmatrix} u_k \\ y_k \end{bmatrix}, \tag{15.17}$$

and the training procedure described in Section 15.2 can be applied. However, this training procedure is not appropriate if there is noise, as illustrated by Nerrand et al. (1994). Because $y_k - Cx_k = v_k \neq 0$, there will be a difference between the training based on (15.16) and training based on (15.17). To analyze the difference between these two training methods, the parameters θ, c and w are fixed, and the difference between the state sequences is derived. The state sequence with the scheduling vector (15.16) is given by

$$x_{k+1} = \sum_{i=1}^{s} p_i(u_k, Cx_k)\Big(A_i x_k + B_i u_k + O_i\Big), \tag{15.18}$$

and the state sequence obtained by using (15.17) instead, is given by

$$\widetilde{x}_{k+1} = \sum_{i=1}^{s} p_i(u_k, y_k)\Big(A_i \widetilde{x}_k + B_i u_k + O_i\Big). \tag{15.19}$$

The state error $\varepsilon_k := x_k - \widetilde{x}_k$ satisfies

$$\varepsilon_{k+1} = \sum_{i=1}^{s} p_i(u_k, y_k) A_i \varepsilon_k + \mu_k, \tag{15.20}$$

with

$$\mu_k = \sum_{i=1}^{s} \Big(p_i(u_k, Cx_k) - p_i(u_k, y_k)\Big)\Big(A_i x_k + B_i u_k + O_i\Big).$$

The state error ε_k is bounded provided that the systems (15.18) and (15.19) are bounded-input, bounded-output stable. The state error will not go to zero as time increases, because the system (15.20) is driven by the signal μ_k. The magnitude of μ_k depends on the difference between $p_i(u_k, Cx_k)$ and $p_i(u_k, y_k)$, and thus on the noise level. Therefore, it is expected that a lower noise level yields a smaller state error ε_k.

Note that the above analysis is for fixed parameters. Since the parameters are changed during training, based on the value and gradient of the cost function, the estimated models obtained from the two procedures can differ considerably, and they frequently do in practice.

Summing up, in the presence of noise, training with (15.17) is different from training with (15.16). Certain models can be very sensitive to the noise, so that even very small noise levels, like numerical errors arising from finite word lengths calculations, cause problems. The appropriate training method is the one based on (15.16). Unfortunately, this training procedure is not without problems.

For training based on (15.16), the gradient with respect to the parameters θ can again be computed by simulating a dynamic system, but now this system is no longer linear with respect to the state. The gradient can be computed from (15.12) with $\mathcal{X}_k^{\theta_j}$ satisfying

$$
\begin{aligned}
\mathcal{X}_{k+1}^{\theta_j} \;=\; & \sum_{i=1}^{s} p_i(u_k, Cx_k) \left(A_i \mathcal{X}_k^{\theta_j} + \frac{\partial A_i}{\partial \theta_j} x_k + \frac{\partial B_i}{\partial \theta_j} u_k + \frac{\partial O_i}{\partial \theta_j} \right) \\
& + \sum_{i=1}^{s} \frac{\partial p_i(u_k, Cx_k)}{\partial \theta_j} \left(A_i x_k + B_i u_k + O_i \right).
\end{aligned}
$$

Linearity is destroyed, because $\partial p_i(u_k, Cx_k)/\partial \theta_j$ depends on $\mathcal{X}_k^{\theta_j}$, which was defined in (15.13). Therefore, stability of the dynamic equation of the model (15.4) no longer implies stability of the dynamic equations involved in the gradient calculations. When the gradient calculations become unstable during training, the optimization algorithm breaks down.

Similar problems occur in training other types of nonlinear model structures, for example, neural networks. In general, two training procedures can be distinguished. The first one is *parallel training* (Narendra and Parthasarathy 1990), also called *undirected training* by Nerrand et al. (1994). In this method, the model outputs are in a feedback loop and it is necessary to train a recurrent network. In the second procedure, called *series-parallel training* (Narendra and Parthasarathy 1990), or *directed training* (Nerrand et al. 1994), the recurrent nature is removed by replacing the feedback of the model outputs by the measured outputs. The parallel methods suffer from stability problems during training; therefore, series-parallel methods are often preferred. Although with series-parallel methods the training procedure is well-behaved, the resulting models can have a bias (Nerrand et al. 1994) or might even give a completely different response if simulated in free run.

As pointed out by Nerrand et al. (1994), a combination of parallel (undirected) and series-parallel (directed) training is also possible, resulting in *hybrid* methods. In the current discussion, training with (15.16) is a parallel method. Training with (15.17) is a hybrid method; the method is recurrent with respect to the state sequence, but not with respect to the scheduling variable. Similar arguments as given in Section 15.2 can be used to show that the hybrid method based on (15.17) does not suffer from stability problems during training and is therefore the preferred method. The drawback of this method is of course that it might perform bad in free run.

Note that similar to the extension suggested at the end of Section 15.2, the identification procedure described above can be extended to deal with a scheduling vector given by $\phi_k = h(y_k, y_{k-1}, \ldots, y_{k-L}, u_k, u_{k-1}, \ldots, u_{k-M})$ for some known nonlinear map h and integers L and M.

15.4   Dealing with Process Noise

The presented training procedures for obtaining a model of the system (15.4)–(15.5) are unbiased if the output is disturbed by white measurement noise. For correlated noise at the output, different identification methods are needed that take into account the correlation of the noise. One possible approach is to first estimate the inverse covariance matrix of the residuals of the cost function, and as a second step re-estimate the parameters using a weighted least-squares cost function, where the estimate of the inverse covariance matrix is used as the weighting. Recently, David and Bastin (2001) described a method to estimate this very large dimensional inverse covariance matrix.

For a special type of correlated output noise v_k that involves the dynamics of the deterministic part in the following way:

$$x_{k+1} = \sum_{i=1}^{s} p_i(\phi_k)\Big(A_i x_k + B_i u_k + O_i + K_i v_k\Big), \qquad (15.21)$$

$$y_k = C x_k + v_k, \qquad (15.22)$$

it is easy to derive the one-step-ahead predictor:

$$x_{k+1} = \sum_{i=1}^{s} p_i(\phi_k)\Big((A_i - K_i C)x_k + B_i u_k + O_i + K_i y_k\Big), \qquad (15.23)$$

$$\widehat{y}_k = C x_k. \qquad (15.24)$$

This one-step-ahead predictor can be trained in similar ways as discussed previously. The expressions for the gradient in the case $\phi_k = u_k$ are not difficult to derive, and again involve simulating a linear parameter-varying system. These expressions have been derived in the context of identification of linear parameter-varying systems by Verdult and Verhaegen (2001c); see also Section 5.2. As a result, the identification approach described in the previous sections can also be used to deal with special types of process noise.

Chapter 16

Examples

The identification procedure for local linear state-space models in Chapter 15 is applied to several examples that have been studied in the literature before. Section 16.1 contains several academic benchmark examples. In Section 16.2 local linear state-space systems are used to model a bioreactor.

In all the examples, the local models were initialized using a linear time-invariant state-space model estimated by the PI-MOESP subspace identification method described by Verhaegen (1994). The centers c_i and widths w_i of the weighting functions were initialized such that the weighting functions cover the operating range more or less uniformly. The scheduling variable ϕ_k, the model order n, and the number of local linear models s were determined by trial and error using the performance of the model on a fresh data set for decision. In practice, more systematic procedures are needed to determine these quantities, but such methods are outside the scope of this thesis. In the examples, both the variance-accounted-for (VAF) and mean square error (MSE) were used to evaluate the performance of the models: the VAF is defined in equation (6.1) on page 88, the MSE is simply $\text{var}(y_k - \widehat{y}_k)$, with y_k the real output and $\widehat{y}_k$ the output of the model.

16.1 Simulations

Below, three nonlinear systems are presented to illustrate the ideas presented in Chapter 15.

16.1.1 SISO System Scheduled on the Input

The first example system was described by Narendra and Parthasarathy (1990) as an example for the use of neural networks to model dynamical systems. Nie (1994) used this example to demonstrate the dynamic modeling capabilities of neuro-fuzzy networks and Boukhris, Mourot, and Ragot (1999) used it with local linear fuzzy models. The input-output description of the system is

$$y_{k+1} = \frac{y_k y_{k-1} y_{k-2} u_{k-1}(y_{k-2} - 1) + u_k}{1 + (y_{k-2})^2 + (y_{k-1})^2}.$$

For identification a multistep input signal was used; the steps in this signal had a fixed length of 10 samples and a random magnitude between -1 and 1, determined by a uniform distribution. A data set of 800 samples was generated to train the local linear models. By some trial and error it was determined that the data can be described by four third-order models scheduled on the input. The initial weights p_i and the optimized ones are shown in Figure 16.1 on page 213. To assess the quality of the model, a validation data set was generated using the input signal

$$u_k = \begin{cases} \sin(2\pi k/250) & 1 \le k \le 500, \\ 0.8\sin(2\pi k/250) + 0.2\sin(2\pi k/25) & 501 \le k \le 800. \end{cases}$$

The same validation data set was used by the previously mentioned authors. With four third-order state-space models, the VAF on this validation data equaled 99.9% and the MSE 0.0002. Figure 16.2 on page 213 shows the outputs of the original system and of the model; the model is very good, the signals are almost indistinguishable. Boukhris, Mourot, and Ragot (1999) achieved an MSE of 0.0003 with seven third-order linear input-output models: a slightly worse MSE with almost twice as much models. The neuro-fuzzy network estimated by Nie (1994) gave an MSE of 0.00028 with 34 local models: a slightly worse MSE with a significantly larger number of models. These results are much better compared with the neural network model identified by Narendra and Parthasarathy (1990). They used a two hidden-layer neural network with 20 hidden neurons in the first layer and 10 in the second layer. They did not give any MSE value, but they plotted the model output and the output of the original system in the same figure. Their figure clearly shows that the low excursions are not modeled very well (a difference of almost 0.3). It can be concluded that the proposed local linear identification approach yields the best performance among the methods described above. Furthermore, it also yields the lowest model complexity.

Next, zero-mean white noise was added to the output signal, such that the VAF between the noise-free and noisy output equals 90.8% and the MSE 0.016. This corresponds to an SNR of 10 dB. The training was influenced by the noise, as can be seen from Figure 16.3 on page 214, which compares the weights after training with noisy data with the weights obtained by training with noise-free data. The weights clearly differ. However, the performance of the model on the (noise-free) validation data set was still very good: the VAF equaled 99.8% and the MSE equaled 0.0005. Figure 16.4 on page 214 shows the outputs of the original system and of the model obtained from noisy data. Note that the error on the validation data set is much smaller than the noise added to the identification data; therefore, the performance of the model is still satisfactory.

To further address the influence of the noise, 100 identification experiments were carried out, each with a different realization of the noise. Since the previous experiment showed that the noise had quite some influence on the weights, the noise level was reduced for these experiments such that the MSE between the noisy and noise-free data was 0.0016. The MSE values of the model output on the validation data are shown in Figure 16.5 for each experiment. The figure shows that in 84 out of 100 experiments, the MSE of the model on the validation data is smaller than the MSE between the noisy and noise-free identification data. The
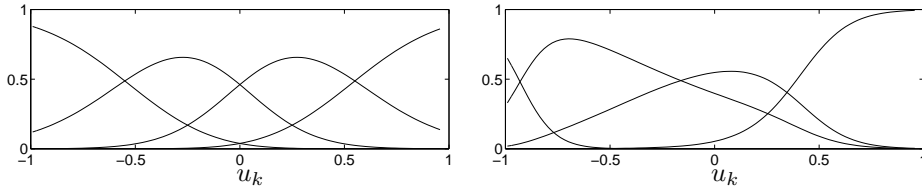
Figure 16.1: SISO example scheduled on the input: *left*, initial weights p_i as a function of the input signal; *right*, weights p_i after optimization.
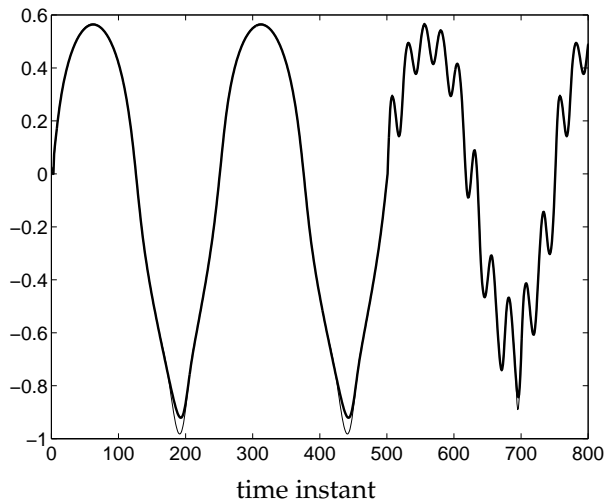
Figure 16.2: SISO example scheduled on the input: comparison between the output signal of the validation data (*thick line*) and the free-run simulation of the model (*thin line*). The lines are almost indistinguishable.
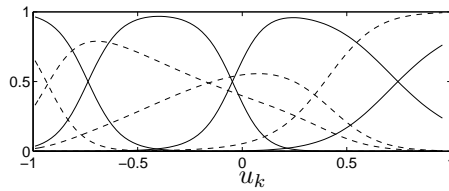
Figure 16.3: SISO example scheduled on the input, with output noise: weights p_i after optimization (*solid lines*) compared with the optimized weights p_i obtained from training without noise (*broken lines*).
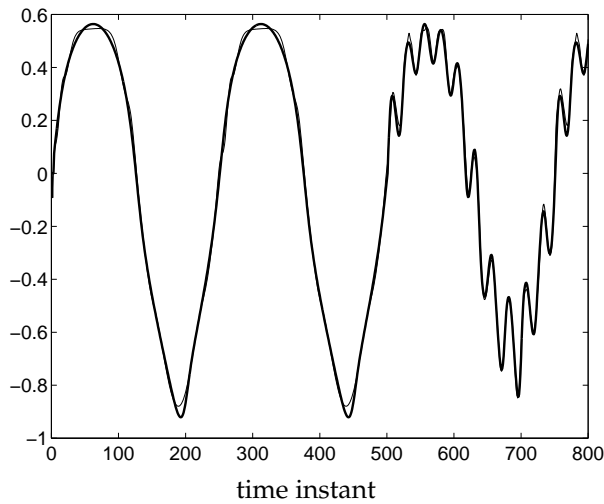
Figure 16.4: SISO example scheduled on the input, with output noise: comparison between the output signal of the validation data (*thick line*) and the free-run simulation of the model (*thin line*). The lines are almost indistinguishable.
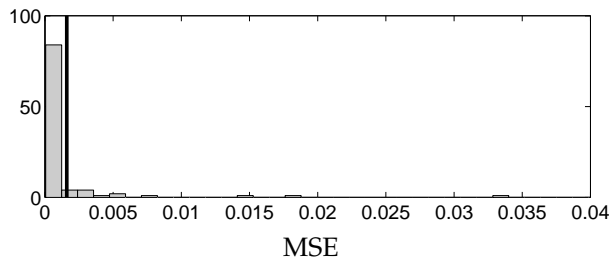
Figure 16.5: SISO example scheduled on the input, with output noise: histogram of the MSE of the free-run simulation of the model on the validation data for 100 experiments with different noise realizations. The *thick vertical line* indicates the MSE between the noisy and noise-free identification data.

mean MSE over the 100 experiments was 0.002.

16.1.2   SISO System Scheduled on the Output

The following example system was also described by Narendra and Parthasarathy (1990):

$$y_{k+1} = \frac{y_k y_{k-1}(y_k + 2.5)}{1 + (y_{k-1})^2 + (y_k)^2} + u_k.$$

Since the input enters linearly, and the nonlinearity involves only the output, it was expected that the output should be used as a scheduling variable. To generate identification data, the system was excited with a random input signal, uniformly distributed between -1 and 1. The validation data set was generated with the input

$$u_k = \sin(2\pi k/25),$$

similar to Narendra and Parthasarathy (1990). With four second-order models (42 parameters) scheduled on the output, the VAF on this validation data set equaled 99.8% and the MSE 0.01. A hybrid method based on scheduling with the measured output was used to train the models. Figure 16.6 on page 216 shows the initial and optimized weights, and Figure 16.7 shows the outputs of the original system and of the model; these signals are almost indistinguishable. These results are comparable with the two hidden-layer neural network (250 parameters) of Narendra and Parthasarathy (1990), but again with a reduced model complexity.

   Zero-mean white noise was added to the output signal, such that the VAF between the noise-free and noisy output equals 90.3% and the MSE 0.24. Figure 16.8 on page 217 compares the weights after training with noisy data with the weights obtained by training with noise-free data. They clearly differ, so the training was influenced by the noise. The VAF on the (noise-free) validation data equals 98.8% and the MSE equals 0.066. Figure 16.9 on page 217 shows the outputs of the original system and of the model obtained from noisy data. The difference between these outputs is now clearly visible, but note again that the error on the validation data is smaller than the noise added to the identification data.

   To further address the influence of the noise, 100 identification experiments were carried out, each with a different realization of the noise. The MSE values for each experiment are shown in Figure 16.10. The figure shows that in almost all cases, the MSE of the model on the validation data is smaller than the MSE between the noisy and noise-free identification data. The mean MSE over the 100 experiments was 0.11.

16.1.3   MIMO System

The following MIMO example was taken from Narendra and Parthasarathy (1992); Nie (1994) and Boukhris, Mourot, and Ragot (1999) also studied this example. The system has two inputs $u_k^{(1)}$ and $u_k^{(2)}$, and two outputs $y_k^{(1)}$ and $y_k^{(2)}$. The input-
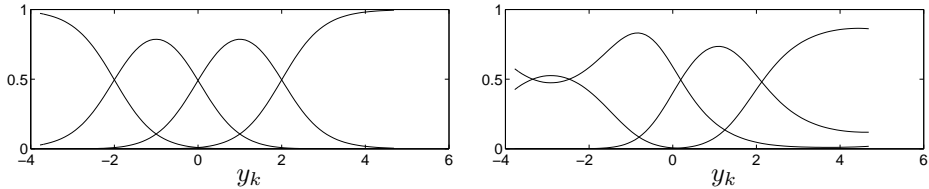
Figure 16.6: SISO example scheduled on the output: *left*, initial weights p_i as a function of the output signal; *right*, weights p_i after optimization.

time instant
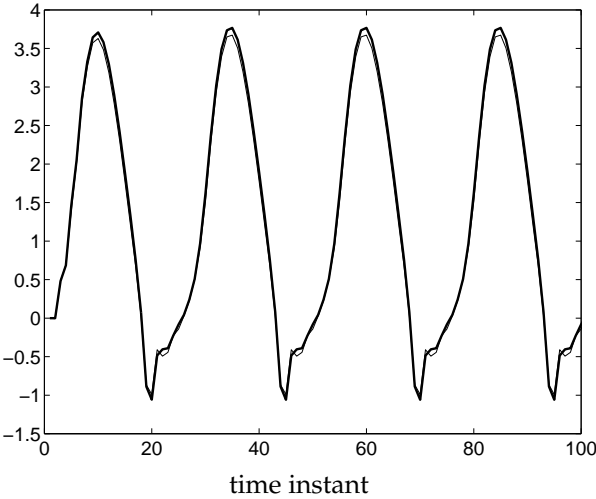
Figure 16.7: SISO example scheduled on the output: comparison between the output signal of the validation data (*thick line*) and the free-run simulation of the model (*thin line*). The lines are almost indistinguishable.
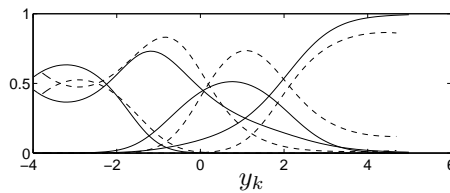
Figure 16.8: SISO example scheduled on the output, with output noise: weights p_i after optimization (*solid lines*) compared with the optimized weights p_i obtained from training without noise (*broken lines*).

time instant
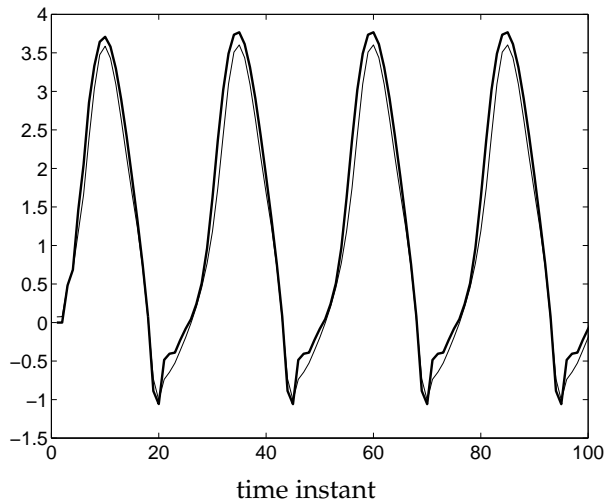
Figure 16.9: SISO example scheduled on the output, with output noise: comparison between the output signal of the validation data (*thick line*) and the free-run simulation of the model (*thin line*).

MSE
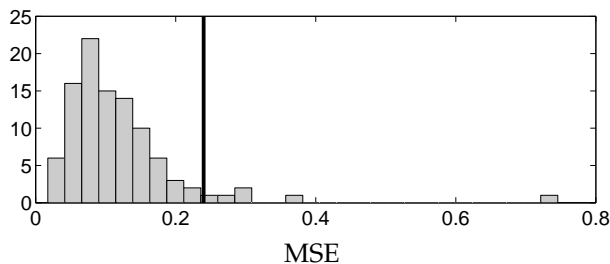
Figure 16.10: SISO example scheduled on the output, with output noise: histogram of the MSE of the free-run simulation of the model on the validation data for 100 experiments with different noise realizations. The *thick vertical line* indicates the MSE between the noisy and noise-free identification data.

output description of the system is

$$y_{k+1}^{(1)} = \frac{0.8\left(y_k^{(1)}\right)^3 + \left(u_k^{(1)}\right)^2 u_k^{(2)}}{2 + \left(y_k^{(2)}\right)^2},$$

$$y_{k+1}^{(2)} = \frac{y_k^{(1)} + y_k^{(1)} y_k^{(2)} + \left(u_{k-1}^{(1)} - 0.5\right)\left(u_k^{(2)} + 0.8\right)}{2 + \left(y_k^{(2)}\right)^2}.$$

The system was excited by taking the inputs equal to two independent random sequences of 1000 samples, uniformly distributed between -1 and 1. The resulting data set was used to train the weights and local models of the system (15.4)–(15.5). An accurate description of the system was obtained by taking four third-order models (94 parameters) scheduled on the two inputs. The initial weights p_i and the optimized ones are shown in Figure 16.11 on page 219.

Following Narendra and Parthasarathy (1992), the performance of the model was evaluated using the input signals:

$$u_k^{(1)} = \sin(2\pi k/250),$$
$$u_k^{(2)} = \cos(2\pi k/250).$$

Figure 16.12 on page 219 shows the response of the original system and the response of the model. The VAF on this validation data set was 97.9% for the first output and 99.2% for the second output; the MSE for the outputs equals 0.0003 and 0.0016, respectively. The performance of the identified local linear model structure was comparable to the performance of the fuzzy model described by Boukhris, Mourot, and Ragot (1999). They achieved MSE values of 0.00026 and 0.0011 with 20 local linear input-output models (64 parameters). The performance was much better than that of the neural network of Narendra and Parthasarathy (1992) and the fuzzy model of Nie (1994). With these models an MSE was achieved that is an order of magnitude larger.

16.2 Identification of a Bioreactor

Puskorius and Feldkamp (1994) described a bioreactor benchmark problem for the identification and control of nonlinear systems. This problem was used for the identification of fuzzy local linear models by Cao, Rees, and Feng (1997). It involves a continuous flow stirred tank reactor that is described by two coupled nonlinear differential equations. These differential equations are integrated with a forward Euler scheme using a step size of $T = 0.01$ time units; the resulting discrete-time equations are

$$x_{k+1}^{(1)} = x_k^{(1)} + T\left(-x_k^{(1)} u_k + x_k^{(1)}\left(1 - x_k^{(2)}\right)\right)\exp\left(\frac{x_k^{(2)}}{\Gamma}\right), \tag{16.1}$$

$$x_{k+1}^{(2)} = x_k^{(2)} + T\left(-x_k^{(2)} u_k + x_k^{(1)}\left(1 - x_k^{(2)}\right)\right)\exp\left(\frac{x_k^{(2)}}{\Gamma}\right)\frac{1 + \beta}{1 + \beta - x_k^{(2)}}, \tag{16.2}$$
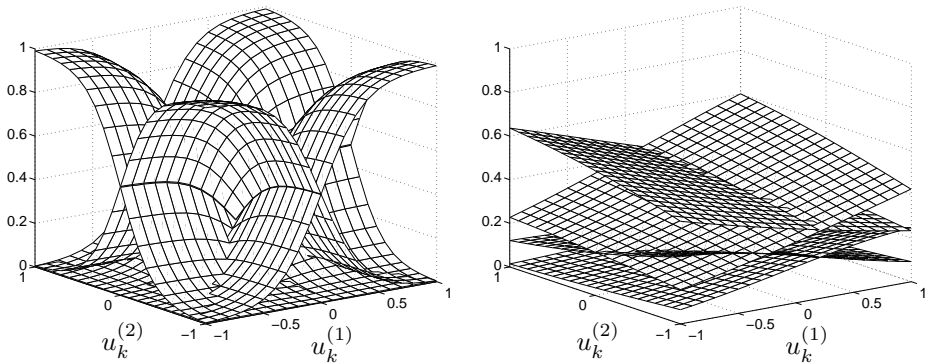
Figure 16.11: MIMO example: *left*, initial weights p_i as a function of the two input signals; *right*, weights p_i after optimization.

time instant
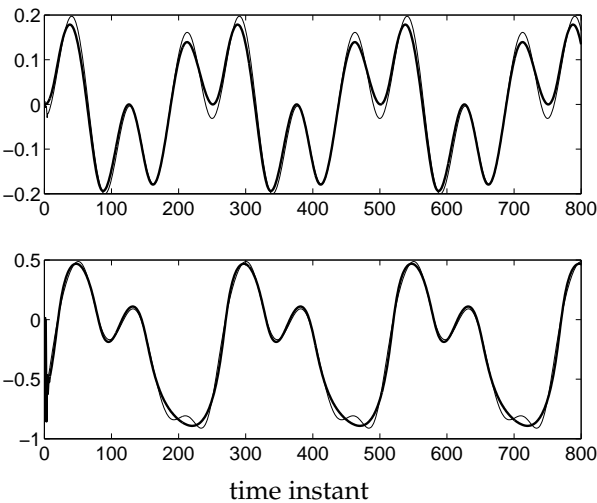
Figure 16.12: MIMO example: comparison between the two output signals of the validation data (*thick lines*) and the free-run simulation of the model (*thin lines*).
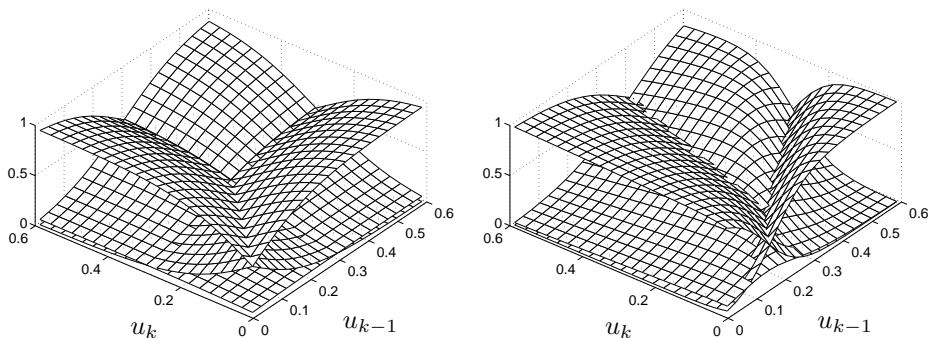
Figure 16.13: Bioreactor example: *left*, initial weights p_i as a function of u_k and u_{k-1}; *right*, weights p_i after optimization.

where $\beta = 0.02$ and $\Gamma = 0.48$. The states $x_k^{(1)}$ and $x_k^{(2)}$ are dimensionless quantities describing the cell mass and the amount of nutrients in a constant volume tank. The input u_k is the flow rate of nutrients into the tank, and equals the rate at which content is removed from the tank. Only the first state is measurable, that is, $y_k = x_k^{(1)}$.

The benchmark problem combines severe nonlinearity with a tendency to instability. However, in this thesis the modeling task was limited to the stable region. The input signal was taken equal to a multistep signal with steps of 1000 time units with a random magnitude between 0 and 0.6 determined from a uniform distribution. The equations (16.1)–(16.2) were used to generate $50\,000$ data samples. The resulting input and output data sets were resampled at $T/50$, so 1000 data samples are available for identification. The same resampling was used by Puskorius and Feldkamp (1994).

First, scheduling on the output was tried. With two first-order models a good performance was obtained if the real output, not the model output, was used as a scheduling variable. However, it is desirable to simulate the model in free run, that is, with the model output as scheduling variable. In this case the model performed very poorly. A possible explanation is that small errors in the output signal result in wrong scheduling and thus the model output diverges from the real output.

Next, the input was used for scheduling. It turned out that three third-order models scheduled on u_k and u_{k-1} gave a satisfactory performance. The initial weights p_i and the optimized ones are shown in Figure 16.13. On a fresh data set, the VAF equaled 99.1% and the MSE $4 \cdot 10^{-6}$. Figure 16.14 on page 221 shows the real output of this validation data set together with the model output. For comparison, Figure 16.15 shows the output of the linear model that was used to initialize the local models. The performance of the local linear models is comparable to the fuzzy model of Cao, Rees, and Feng (1997), who used five second-order models.
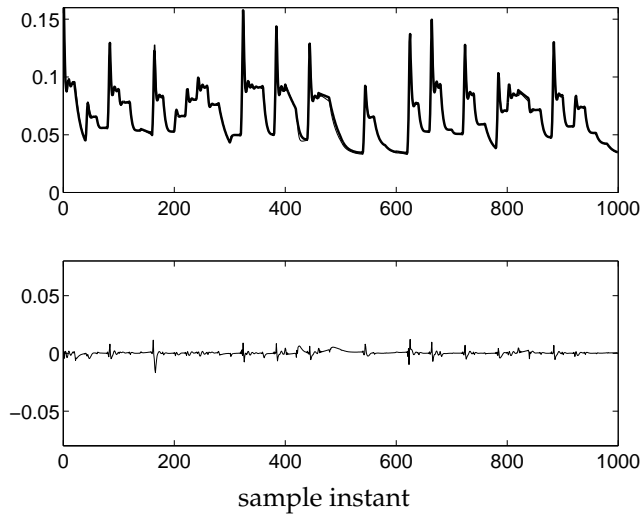
Figure 16.14: Bioreactor example: comparison between the validation data and the free-run simulation of the local linear models; The *top* figure shows the original output signal (*thick line*) and the model output (*thin line*); these signals are almost indistinguishable. The *bottom* figure shows the error between validation data and model output.
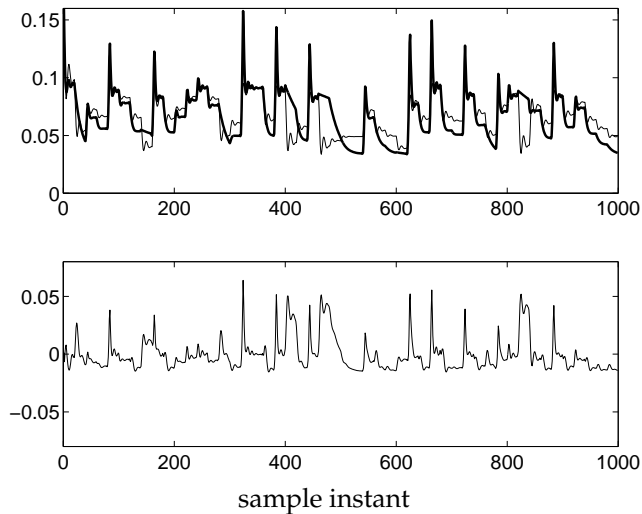
Figure 16.15: Bioreactor example: comparison between the validation data and the free-run simulation of the linear time-invariant model; The *top* figure shows the original output signal (*thick line*) and the model output (*thin line*). The *bottom* figure shows the error between validation data and model output.

Chapter 17

Conclusions and Recommendations for Part III

The final part of the thesis described a novel identification method for local linear state-space models. A summary of the most important results is given in this chapter. In addition, some suggestions for further research are made.

A nonlinear optimization-based method was presented that identifies local linear models scheduled by the input or a combination of the input and output. The method identifies the models from input and output measurements; it does not require full state measurement, and it can deal with measurement noise and certain types of process noise in a consistent way. Normalized radial basis functions are used as weighting functions that combine the local state-space models to obtain a global description of the input-output behavior. The system matrices of the local models are fully parameterized. An iterative projected gradient search method was described to identify the local models and the centers and widths of the radial basis functions. The method deals with the nonuniqueness of the fully parameterized state-space representation by calculating at every iteration the direction in which a change of parameters does not influence the input-output behavior of the model, and projecting these directions out of the parameter update.

Two special scheduling cases were discussed. When the scheduling is based only on the input signals, stability of the dynamic gradient calculations can be guaranteed, and hence the training is well-behaved. In the more general case where scheduling is also based on the output signals, two training methods are possible, each having their own advantages and disadvantages. One method is based on scheduling with the measured outputs; this method behaves well during training, but the resulting model can perform poorly when simulated in free run. The other training method that uses the model output for scheduling does not have this problem, but can suffer from stability problems during training.

To initialize the training procedure, a global linear model is estimated and used to initialize all the local models; in addition, the weighting functions are uniformly distributed over the operating range of the scheduling vector. With this initializa-

tion, the performance of the local linear models on the identification data will always be better than that of the global linear model.

Several examples that have been studied in the literature before, were used to demonstrate the potential of the proposed methods. One of these examples involves the identification of a bioreactor. It was shown that the proposed approach yields models with an accuracy that is comparable to or better than other input-output based local linear model and neural network approaches, and often at a reduced model complexity.

Only the estimation of the models was addressed in this thesis. An important problem that should be addressed in future research is structure identification. Although the structure of the model could be determined by trial and error for the examples that were presented, more sophisticated methods are needed for higher dimensional problems. Structure identification involves (Johansen and Foss 1995; Johansen and Murray-Smith 1997):

- Determination of an appropriate scheduling vector.

- Determination of the number of local models.

- Determination of the system order.

- Partitioning of the operating range.

Some of these aspects are closely linked to the particular model estimation procedure used. Below, some suggestions for structure identification are given that can be combined with the model estimation procedure in Chapter 15. Their validity still needs to be investigated.

The scheduling vector should consist of inputs and outputs and time-delayed versions of them. Selection could be done by simply trying out different combinations. This combinatorial problem can be solved using tree algorithms and it can be speeded up using some appropriate heuristics. Such a strategy has been proposed by Sugeno and Yasukawa (1993) and Johansen and Foss (1995). Since this strategy is independent of the particular model estimation procedure used, it should work with the procedure described in Chapter 15.

To determine the number of local models, two strategies have been discussed in the literature. The first strategy starts with only a few local models and increase the number of models to obtain a better model fit. In this way, a gradual refinement of the model is obtained. Tree algorithms can be used to implement this procedure, an example is the LOLIMOT method of Nelles, Hecker, and Isermann (1997). Model selection can be based on the performance on validation data. The second strategy starts with a large number of models and tries to reduce them by merging certain models. To merge the models, appropriate similarity measures are needed. An example of this strategy is the supervised fuzzy clustering method of Setnes (2000).

The order of the system can be determined by estimating models of different orders and evaluating their performance on validation data. However, alternative methods exist that are computationally less demanding. These methods determine the order directly from the input and output data, and thus circumvent the estimation of several models. One such method is based on the assumption that

the system is described by continuous nonlinear functions. It determines the order by evaluating Lipschitz quotients for different orders. The method was described by He and Asada (1993). It was also used by several other authors, including Bomberger and Seborg (1997), Bomberger (1997, Chapter 3), and Verdult, Verhaegen, and Scherpen (2000). Some refinements were proposed by Boukhris, Mourot, and Ragot (1999). Another method to determine the system order from input and output data is based on the estimation of the so-called embedding dimension that is used in time-series modeling (Kantz and Schreiber 1999; Sauer, Yorke, and Casdagli 1991). Roughly speaking, the embedding dimension equals the order of the system. False nearest neighbor techniques are used for its estimation (Kennel, Brown, and Abarbanel 1992; Schreiber 1995). The method was used for systems with inputs by several authors, including Bomberger and Seborg (1997), Rhodes and Morari (1995), and Verdult, Verhaegen, and Scherpen (2000). Rhodes (1997, Part II) performed a detailed analysis of the method in the presence of noise and suggested some improvements.

The state-space setting in Chapter 15 suggests that all the local models have the same order. However, in input-output approaches, the local models usually have different orders (Babuška and Verbruggen 1997; Johansen and Murray-Smith 1997; Nelles, Hecker, and Isermann 1997). In the state-space setting, this would mean that the state dimension varies among the local models, or that certain local models are not minimal. How to deal with different local-model orders in the state-space setting in Chapter 15 remains an open question.

Another structural issue is the partitioning of the operating range. Given the number of local models, the method described in Chapter 15 determines this partitioning. However, it needs an initial estimate. In the examples in Chapter 15, a uniform partitioning has been used to initialize the algorithm. Although this worked out fine, for higher dimensional problems more constructive methods are needed. A possible way to find an initial partitioning would be to use fuzzy clustering methods. Such methods have been widely used for estimating Takagi-Sugeno fuzzy models by numerous authors, for example Babuška (1997), De Bruin and Roffel (1996), and Cao, Rees, and Feng (1997).

Appendix

Linear and Nonlinear Subspace Identification

$\mathbf{T}$his appendix briefly reviews subspace identification for linear time-invariant systems and it provides a discussion on the extension of subspace identification to nonlinear systems.

A linear time-invariant (LTI) state-space system can be described by

$$
\begin{aligned}
x_{k+1} &= A x_k + B u_k + K v_k, & \text{(A.1)} \\
y_k &= C x_k + D u_k + v_k, & \text{(A.2)}
\end{aligned}
$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ the input, $y_k \in \mathbb{R}^\ell$ the output, and $v_k \in \mathbb{R}^\ell$ an unknown disturbance. It is assumed that v_k is a zero-mean white noise sequence. The goal is to determine the system matrices A, B, C, and D, from a finite number of measurements of the input u_k and the output y_k. It is important to realize that the input-output behavior of the system (A.1)–(A.2) does not change if the state sequence is linearly transformed, in other words, the same input-output behavior is obtained with the system

$$
\begin{aligned}
\overline{x}_{k+1} &= A_T \overline{x}_k + B_T u_k + K_T v_k, \\
y_k &= C_T \overline{x}_k + D_T u_k + v_k,
\end{aligned}
$$

with $\overline{x}_k = T^{-1} x_k$ (T is a square nonsingular matrix), $A_T = T^{-1} A T$, $B_T = T^{-1} B$, $K_T = T^{-1} K$, $C_T = C T$, and $D_T = D$. Therefore, without any additional assumptions, it is only possible to determine the system matrices A, B, C, and D up to a similarity transformation T.

As discussed at the beginning of Chapter 3, different subspace identification methods exist for linear time-invariant systems. The one described below is an N4SID type of method that estimates the state sequence up to a similarity transformation. The description of the subspace method is followed by a discussion on the development of subspace identification techniques for nonlinear systems.

The N4SID subspace identification method is based on the fact that by storing the input and output data into structured matrices, it is possible to retrieve the

state sequence up to a similarity transformation. The data sequences are stored in data matrices with a block Hankel structure. The data matrix for the output signal y_k equals

$$Y_{j,s,N} := \begin{bmatrix} y_j & y_{j+1} & \cdots & y_{j+N-1} \\ y_{j+1} & y_{j+2} & \cdots & y_{j+N} \\ \vdots & \vdots & \ddots & \vdots \\ y_{j+s-1} & y_{j+s} & \cdots & y_{j+N+s-2} \end{bmatrix}.$$

The sequences u_k and v_k are stored in a similar way into the matrices $U_{j,s,N}$ and $V_{j,s,N}$. With the system equations (A.1)–(A.2), the data matrices can be related as

$$Y_{j,s,N} = \Gamma_s X_{j,N} + H_s U_{j,s,N} + G_s V_{j,s,N}, \tag{A.3}$$

where

$$\Gamma_s := \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix}, \qquad X_{j,N} := [x_j, x_{j+1}, \ldots, x_{j+N-1}],$$

$$H_s := \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & & 0 \\ \vdots & & & \ddots & \ddots \\ CA^{s-2}B & CA^{s-3}B & \cdots & CB & D \end{bmatrix},$$

$$G_s := \begin{bmatrix} I_\ell & 0 & 0 & \cdots & 0 \\ CK & I_\ell & 0 & \cdots & 0 \\ CAK & CK & I_\ell & & 0 \\ \vdots & & & \ddots & \ddots \\ CA^{s-2}K & CA^{s-3}K & \cdots & CK & I_\ell \end{bmatrix}.$$

The goal is to recover the state sequence $X_{j,N}$ up to a similarity transformation and use it to compute the system matrices. It is also possible to recover the column space of Γ_s and use that to compute the system matrices (Verhaegen 1994), but that is not considered here. Thinking about the extension of linear subspace identification to nonlinear systems, it is expected that it is easier to come up with a method for recovering the state sequence than to determine a nonlinear equivalent for recovering the column space of Γ_s.

To recover the state sequence the influence of the 'future inputs' $U_{j,s,N}$ and the noise $V_{j,s,N}$ needs to be removed from the matrix $Y_{j,s,N}$. The MOESP type of subspace methods (Verhaegen 1994) get rid of $U_{j,s,N}$ using an orthogonal projection and get rid of $V_{j,s,N}$ using instrumental variables. The N4SID type methods proceed differently: the product $\Gamma_s X_{j,N}$ is estimated by replacing the state sequence by a linear combination of input and output data and subsequently estimating the product of the matrix Γ_s with this linear combination. A singular value decomposition of the estimated product $\Gamma_s X_{j,N}$ can then be used to recover the state sequence up to a similarity transformation. A detailed explanation follows.

The state equation (A.1) can be written as

$$x_{k+1} = (A - KC)x_k + (B - KD)u_k + Ky_k.$$

Therefore,

$$X_{s,N} = (A - KC)^{s-1}X_{1,N} + L_s Z_N, \tag{A.4}$$

where

$$
\begin{aligned}
L_s &:= \begin{bmatrix} L_s^u & L_s^y \end{bmatrix}, \\
L_s^u &:= \begin{bmatrix} (A - KC)^{s-2}(B - KD) & (A - KC)^{s-3}(B - KD) & \cdots & (B - KD) \end{bmatrix}, \\
L_s^y &:= \begin{bmatrix} (A - KC)^{s-2}K & (A - KC)^{s-3}K & \cdots & K \end{bmatrix}, \\
Z_N &:= \begin{bmatrix} U_{1,s,N} \\ Y_{1,s,N} \end{bmatrix}.
\end{aligned}
$$

Equation (A.4) shows that the state sequence $X_{s,N}$ can be expressed as a linear combination L_s of past input and output data Z_N and a nuisance term due to the initial states $X_{1,N}$. With this equation, the data equation for $i = s$ can be written as

$$Y_{s,s,N} = \Gamma_s L_s Z_N + H_s U_{s,s,N} + G_s V_{s,s,N} + \Gamma_s (A - KC)^{s-1} X_{1,N}.$$

This equation forms the basis for computing a least squares estimate of $\Gamma_s L_s$. For ease of notation define:

$$\Omega_N := \begin{bmatrix} Z_N \\ U_{s,s,N} \end{bmatrix}.$$

The normal equation for obtaining least squares estimates of $\Gamma_s L_s$ and H_s is

$$Y_{s,s,N}\Omega_N^T = \begin{bmatrix} \Gamma_s L_s & H_s \end{bmatrix} \Omega_N \Omega_N^T + G_s V_{s,s,N}\Omega_N^T + \Gamma_s(A - KC)^{s-1}X_{1,N}\Omega_N^T.$$

These least squares estimates are given by

$$\begin{bmatrix} \widehat{\Gamma_s L_s} & \widehat{H}_s \end{bmatrix} = Y_{s,s,N}\Omega_N^T(\Omega_N\Omega_N^T)^{-1}. \tag{A.5}$$

The requirement that the matrix $\Omega_N\Omega_N^T$ is invertible can be related to the notion of persistency of excitation of the input signal (Jansson and Wahlberg 1998). Here it is assumed that this matrix is invertible. Since the white noise property of v_k gives

$$\lim_{N\to\infty} \frac{1}{N}V_{s,s,N}\Omega_N^T = 0, \qquad \text{w. p. 1,}$$

it follows that

$$\lim_{N\to\infty} \begin{bmatrix} \widehat{\Gamma_s L_s} & \widehat{H}_s \end{bmatrix} = \begin{bmatrix} \Gamma_s L_s & H_s \end{bmatrix} + \Gamma_s(A - KC)^{s-1}\begin{bmatrix} \Delta_z & \Delta_u \end{bmatrix}, \qquad \text{w. p. 1,}$$

with

$$\begin{bmatrix} \Delta_z & \Delta_u \end{bmatrix} := \lim_{N\to\infty} \frac{1}{N}X_{1,N}\Omega_N^T \left(\frac{1}{N}\Omega_N\Omega_N^T\right)^{-1},$$

where Δ_z corresponds to Z_N and Δ_u corresponds to $U_{s,s,N}$. Now the product $\widehat{\Gamma_s L_s} Z_N$ can be related to the state sequence as follows:

$$
\begin{aligned}
\lim_{N \to \infty} \widehat{\Gamma_s L_s} Z_N &= \Gamma_s L_s Z_N + \Gamma_s (A - KC)^{s-1} \Delta_z Z_N \\
&= \Gamma_s X_{s,N} + \underbrace{\Gamma_s (A - KC)^{s-1} (\Delta_z Z_N - X_{1,N})}.
\end{aligned}
$$

Typically, all the eigenvalues of the matrix $A - KC$ have magnitudes smaller than one and thus, if s is chosen sufficiently large the under-braced term of the previous equation will be small. This term is therefore neglected in subspace identification. Because $\widehat{\Gamma_s L_s} Z_N \approx \Gamma_s X_{s,N}$, an approximation of the state sequence (up to a similarity transformation) can be obtained using the following singular value decomposition:

$$
\widehat{\Gamma_s L_s} Z_N = U_n \Sigma_n V_n^T,
$$

with $\Sigma_n \in \mathbb{R}^n$ and $\widehat{\Gamma_s L_s}$ given by equation (A.5). The state estimate is given by

$$
\widehat{X}_{s,N} = \Sigma_n^{1/2} V_n^T.
$$

The system matrices can now be estimated up to a similarity transformation by solving the least squares problem

$$
\min_{A_T, B_T, C_T, D_T} \left\| \begin{bmatrix} \widehat{X}_{s+1,N} \\ Y_{s,1,N} \end{bmatrix} - \begin{bmatrix} A_T & B_T \\ C_T & D_T \end{bmatrix} \begin{bmatrix} \widehat{X}_{s,N} \\ U_{s,1,N} \end{bmatrix} \right\|_F^2,
$$

where $\| \cdot \|_F$ denotes the Frobenius norm.

Subspace identification for linear time-invariant systems is based on the data equation (A.3). The structure of this data equation is essential to the success of the method. For time-varying systems the structure is destroyed, because it is not possible to relate the different columns of the matrix $Y_{s,s,N}$ to the state sequence $X_{s,N}$ as in the data equation (A.3). The extension to nonlinear systems is far from trivial, because in general it is not possible to obtain a data equation with a structure similar to (A.3). However, for certain types of nonlinear systems, it is possible to exploit the structure of the system and come up with a suitable data equation; an example is the subspace identification method for Hammerstein systems of Verhaegen and Westwick (1996). A Hammerstein system is a nonlinear system that can be decomposed as a linear time-invariant system with a static nonlinear function at the input:

$$
\begin{aligned}
x_{k+1} &= A x_k + B \mu_k, \\
y_k &= C x_k + D \mu_k, \\
\mu_k &= f(u_k),
\end{aligned}
$$

with u_k the input and f a nonlinear function. If the nonlinear function f can be represented as an polynomial in u_k, then $\mu_k = M \omega_k$, where M contains the coefficients of the polynomial and ω_k contains the polynomial terms of u_k, that is, powers of the elements of u_k and the cross products between them. Now using

ω_k as input signal, a data equation similar to (A.3) can be derived, and subspace identification can be used to determine the system matrices A, B, C, D, and M from the signals w_k and y_k. Also for Wiener systems it is possible to exploit the structure of the system to derive a data equation suitable for subspace identification (Westwick and Verhaegen 1996). This thesis discusses subspace identification for linear parameter-varying and bilinear systems; it is shown that also for these systems a data equation with a structure similar to (A.3) can be derived. Details can be found in Chapters 3 and 9.

The extension of subspace identification methods to more general nonlinear systems remains an open problem. Subspace identification is closely related to time-series embedding. The idea of time-series embedding is that it is sufficient to reconstruct the state sequence up to a nonlinear state transformation. Note the similarity with linear subspace identification, where the state is reconstructed up to a linear state transformation. Below, the idea of embedding for nonautonomous systems is briefly explained; the discussion is based on the paper by Verdult, Verhaegen, and Scherpen (2000).

Let X be a compact manifold of dimension n. Consider the following nonlinear state-space system:

$$
\begin{aligned}
x_{k+1} &= f(x_k, u_k), & \text{(A.6)} \\
y_k &= h(x_k), & \text{(A.7)}
\end{aligned}
$$

where $f : X \times \mathbb{R}^m \to X$ is a smooth diffeomorphism and $h : X \to \mathbb{R}$ a smooth function. Note that a smooth map f is a diffeomorphism if it is one-to-one and onto, and if the inverse map f^{-1} is also smooth. Given only measurements of y_k and u_k, the goal is to determine a state-space system having the same input-output dynamic behavior as the original system (A.6)–(A.7). It is important to realize that a unique solution to this identification problem does not exist, because the state x_k can only be determined up to an embedding $\Psi : X \to M$, with M a $q \geq n$ dimensional compact manifold. Recall that an embedding Ψ on a compact manifold X is a smooth diffeomorphism having a one-to-one derivative map $D\Psi$. In other words, an embedding is a map that does not collapse points or tangent directions (Sauer et al. 1991). The embedding Ψ can in fact be regarded as a nonlinear state transformation. It corresponds to a change of local coordinates to represent the manifold X.

Let $\xi_k := \Psi(x_k)$. The following state-space system has the same input-output dynamic behavior as the system (A.6)–(A.7):

$$
\begin{aligned}
\xi_{k+1} &= \Psi\big(f\big(\Psi^{-1}(\xi_k), u_k\big)\big) &= \overline{f}(\xi_k, u_k), \\
y_k &= h\big(\Psi^{-1}(\xi_k)\big) &= \overline{h}(\xi_k).
\end{aligned}
$$

It is important to note that the embedding Ψ does not need to be a square map. Hence, the dimension of the state vector ξ_k can be larger than the dimension of x_k.

Define the map $\Phi : X \to Z$ as follows:

$$
\begin{aligned}
\Phi_{f,h,\mu_k}(x_k) &:= \Big[h(x_k), h(f(x_k, u_k)), \ldots, h(f(\cdots f(f(x_k, u_k), u_{k+1}) \cdots), u_{k+d-2})\Big]^T \\
&= [y_k, y_{k+1}, \ldots, y_{k+d-1}]^T,
\end{aligned}
$$

where Z is a d dimensional compact manifold and $\mu_k := [u_k, u_{k+1}, \ldots, u_{k+d-2}]^T$. It has been shown by Stark et al. (1997), Stark (1999) that under some minor assumptions, for every μ_k the map Φ_{f,h,μ_k} is an embedding, provided that $d \geq 2n + 1$. This result is in fact an extension of the Takens embedding theorem for autonomous systems (Takens 1980; Sauer et al. 1991). It means that the dynamics of the system (A.6)–(A.7) can be reconstructed using only a finite number of delayed input and output measurements. Let the delay vector z_k be defined as follows:

$$z_k := [y_k, y_{k+1}, \ldots, y_{k+d-1}]^T,$$

then

$$
\begin{aligned}
z_{k+1} &= \Phi_{f,h,\mu_{k+1}}(x_{k+1}) \\
&= \Phi_{f,h,\mu_{k+1}}\Big(f(x_k, u_k)\Big) \\
&= \Phi_{f,h,\mu_{k+1}}\Big(f(\Phi^{-1}_{f,h,\mu_k}(z_k), u_k)\Big) \\
&= F(z_k, u_k, u_{k+1}, \ldots, u_{k+d-1}),
\end{aligned}
$$

where the inverse map Φ^{-1}_{f,h,μ_k} exists, because Φ_{f,h,μ_k} is an embedding. Hence, the dynamic behavior of the system is completely described by the mapping F. However, we do not end up with a state-space description of the form (A.6). Reason being that the embedding Φ_{f,h,μ_k} depends on μ_k. In other words, the nonlinear state transformation depends on a finite number of 'future' inputs. The last component of F yields a nonlinear ARX type of model describing the system:

$$y_{k+d} = F_{k+d-1}(y_k, y_{k+1}, \ldots, y_{k+d-1}, u_k, u_{k+1}, \ldots, u_{k+d-1}).$$

To arrive at a state-space model of the form (A.6)–(A.7), the dependence of the delay vector z_k on the 'future' inputs μ_k has to be removed; it is similar to the removal of $U_{j,s,N}$ in subspace identification of linear systems. The vectors z_k, $k = 0, 1, 2, \ldots$ are points in the reconstructed state space, and thus describe the dynamics of the system. The elements of the delay vector z_k can be viewed as the coordinates, which are used to reconstruct the state space. Of course, we do not want the coordinates of this space to depend on the input signal, and thus the influence of the 'future' inputs $u_k, u_{k+1}, \ldots, u_{k+d-1}$ has to be removed. It is proposed to use the following modified delay vector

$$
\begin{aligned}
\widetilde{z}_k &:= [\widetilde{y}_k, \widetilde{y}_{k+1}, \ldots, \widetilde{y}_{k+d-1}]^T \\
&= [h(x_k), h(f(x_k, 0)), \ldots, h(f(\cdots f(f(x_k, 0), 0) \cdots), 0)]^T \\
&= \widetilde{\Phi}_{f,h}(x_k).
\end{aligned}
$$

This vector $\widetilde{z}_k$ does not depend on the 'future' inputs. It can be constructed by taking the state x_k as an initial state and then simulate the system for d time steps

with the input set to zero. The delay vectors $\widetilde{z}_k$ satisfy the dynamic equation

$$
\begin{aligned}
\widetilde{z}_{k+1} &= \widetilde{\Phi}_{f,h}(x_{k+1}) \\
&= \widetilde{\Phi}_{f,h}\Big(f(x_k, u_k)\Big) \\
&= \widetilde{\Phi}_{f,h}\Big(f(\widetilde{\Phi}_{f,h}^{-1}(\widetilde{z}_k), u_k)\Big) \\
&= \widetilde{F}(\widetilde{z}_k, u_k),
\end{aligned}
$$

which is of the form (A.6). This equation shows that to make the transition from $\widetilde{z}_k$ to $\widetilde{z}_{k+1}$, the input u_k is still needed. Thus, the reconstruction coordinates are independent of the input, but the influence of u_k on the dynamics is preserved.

In the previous discussion it is assumed that the map $\widetilde{\Phi}_{f,h}$ is an embedding. No formal proof of this has been given. The proposed procedure is mainly motivated by its similarity to the linear subspace method. In order for the procedure to work, it is at least necessary that the system $\widetilde{x}_{k+1} = f(\widetilde{x}_k, 0)$ is observable (Nijmeijer 1982).

The key point of the discussion above is the construction of the modified delay vectors $\widetilde{z}_k$. Verdult, Verhaegen, and Scherpen (2000) proposed a two step procedure in which first a nonlinear input-output model of the system is estimated and as a second step this model is simulated with zero input signals to generate $\widetilde{z}_k$. It is not yet known how to construct the modified delay vectors directly from the available measurements of the input and output; this remains a topic for further research.

Summary

System identification aims at determining a dynamic model of a system directly from input and output data. It provides an alternative to the more classical way of modeling that is based on the first principles of, for example, physics, chemistry, and biology. Modeling from first principles requires a lot of expert knowledge, a considerable time investment, and often results in complex models that are not suitable for fast on-line applications. By contrast, system identification often yields compact models that can be used in real-time model-based controllers.

Over the years considerable attention has been given to the identification of linear systems. Linear systems have proven their usefulness in numerous engineering applications, and many theoretical results have been derived for the identification and control of these systems. However, most real-life systems inherently show nonlinear dynamic behavior. Consequently, the use of linear models has its limitations. When performance requirements are high, the linear model is no longer accurate enough, and nonlinear models have to be used. This motivates the development of identification methods for nonlinear systems.

In this thesis, new system identification methods are presented for three particular types of nonlinear systems: linear parameter-varying state-space systems, bilinear state-space systems, and local linear state-space systems. Although most work on nonlinear system identification deals with nonlinear input-output descriptions, this thesis focuses on state-space descriptions. State-space systems are considered, because they are especially suitable for dealing with multiple inputs and outputs, and they usually require less parameters to describe a system than input-output descriptions do. Equally important, the starting point of many nonlinear control methods is a state-space model of the system to be controlled.

Linear Parameter-Varying State-Space Systems

A linear parameter-varying (LPV) state-space system is a linear system in which the system matrices are functions of a known time-varying parameter vector. In this thesis, the following LPV systems with affine parameter dependence are considered:

$$x_{k+1} = A \left[\begin{array}{c} x_k \\ p_k \otimes x_k \end{array} \right] + B \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right],$$

$$y_k = C \left[\begin{array}{c} x_k \\ p_k \otimes x_k \end{array} \right] + D \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right],$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ the input, $y_k \in \mathbb{R}^\ell$ the output, $p_k \in \mathbb{R}^s$ the time-varying parameter vector, and $\otimes$ denotes the Kronecker product.

The LPV system can be viewed as a nonlinear system that is linearized along a time-varying trajectory determined by the time-varying parameter vector p_k. The time-varying parameter vector p_k determines the operating point of the nonlinear system. This time-varying vector is assumed to be measurable for control and identification. In many industrial applications like flight control and process control, this is indeed the case.

Controller design for LPV systems is an active research area, and many promising techniques have already emerged. By contrast, identification of LPV systems is still in its infancy. This motivates the study of new identification methods for LPV systems. In this thesis, methods are presented that allow the identification of LPV state-space systems given measurements of the input, output, and time-varying parameters.

In Chapter 3 a framework is presented that allows the derivation of several different subspace identification methods for LPV systems with a time-invariant output equation, which are disturbed by process and measurement noise. Subspace methods first appeared in the literature for linear time-invariant systems. Subspace methods are numerically robust methods that do not require a particular parameterization of the state-space system; this makes it attractive to use these methods to deal with multivariable systems.

The key to the derivation of subspace methods for LPV systems is the reconstruction of the state sequence. This reconstruction is always an approximation, because the effect of the initial state has to be neglected. In LPV subspace identification, the state sequence is approximated by a linear combination of the rows of certain structured data matrices that are constructed from measurements of the input, output, and time-varying parameter. This linear combination is unknown, but follows once a linear least-squares problem has been solved. Different identification algorithms can be derived depending on which combination of data matrices is used to approximate the state. An important tuning parameter is the block size of the data matrices. In general, increasing the block size improves the accuracy of the state approximation, but it also increases the number of rows in the data matrices exponentially. A large block size can result in enormous computation times and memory requirements. Therefore, the choice of the block size is a trade-off between accuracy and computational load. Equally important, with a large block size the effect of the finite data length cannot be neglected, and it deteriorates the estimated models.

Chapter 4 describes a subset-selection method that can be integrated into the LPV subspace methods to reduce the number of rows in the data matrices. Since the number of rows in the data matrices grows exponentially with the block size, LPV subspace identification can require a huge amount of computer memory; already for low-order multivariable systems the amount required exceeds the memory of today's personal computers. It is crucial to reduce the dimensions of the data matrices to make the method practically feasible for multivariable systems. The subset-selection method selects only the most dominant rows, and discards the other ones, at the expense of the introduction of an approximation error. This selection of rows can be implemented in an efficient way using an RQ factoriza-

tion that processes the data matrices row by row such that there is no need to store the entire data matrix into the memory. The subset-selection method partly converts the large memory requirements into large computation times; therefore, the derivation of more efficient algorithms should be investigated.

All presented LPV subspace identification methods only yield approximate models; therefore it is desirable to further improve the obtained models. Chapter 5 describes an iterative nonlinear optimization-based method for the identification of LPV systems. The LPV model obtained by subspace identification provides a good initial starting point for this nonlinear optimization. The optimization-based method is a special case of the LFT identification method proposed by Lee and Poolla (1999). It circumvents the need to choose an appropriate parameterization of the LPV system; it uses a full parameterization, and at each iteration only updates the parameters that change the cost function to be optimized. This approach is called the projected gradient search method, because it projects the gradient used in solving the optimization problem such that only the directions that change the cost function remain. It has numerical advantages over canonical parameterizations, and can be used with output measurement noise as well as with a special type of process noise. The combination of the subset-selection subspace identification method and the projected gradient search method forms a powerful identification tool that can be used to identify real-life multivariable LPV systems.

Chapter 6 contains Monte-Carlo simulations involving the different LPV identification methods. These simulations confirm that too small a block size in LPV subspace identification leads to large approximation errors. They also confirm that the block size should not be taken too large, because then the effect of the finite data length cannot be neglected, and again large approximation errors would result. In addition, the simulations show that the dimension reduction by subset selection is feasible. Often a surprisingly large dimension reduction can be achieved that does not deteriorate the model performance much; for example, using only 180 out of $180\,000$ rows results in a performance loss of only 5%. In addition, it is shown that the LPV identification methods can be used to successfully model helicopter rotor dynamics.

Bilinear State-Space Systems

The bilinear system is a simple nonlinear extension of a linear system that can be described by

$$
\begin{aligned}
x_{k+1} &= Ax_k + F(u_k \otimes x_k) + Bu_k, \\
y_k &= Cx_k + Du_k,
\end{aligned}
$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ the input, $y_k \in \mathbb{R}^\ell$ the output, and $\otimes$ denotes the Kronecker product.

The main motivation for studying discrete-time bilinear systems is their simple structure compared with other nonlinear models like, for example, neural networks. Despite their simple structure, bilinear systems can still represent interesting nonlinear behavior: every discrete-time finite Volterra series expansion with time-invariant separable kernels can be realized by a discrete-time bilinear system.

Bilinear systems have been studied extensively, and have been used successfully to solve several real-life problems.

The majority of the existing identification methods for bilinear systems deal either with input-output systems or with state-space systems in phase-variable form. Few methods deal with the general state-space description, in which no special structure of the system matrices is assumed. In this thesis, subspace and nonlinear optimization methods are discussed for the identification of general bilinear state-space systems from input and output data.

Chapter 9 presents several subspace identification methods for bilinear systems. Since the bilinear system can be regarded as a special case of the LPV system (take the time-varying parameter p_k equal to the input u_k), the general framework for LPV subspace identification presented in Chapter 3 can be used for the derivation of bilinear subspace identification methods. The subspace algorithms that appeared in the literature before, the ones described by Favoreel (1999) and by Chen and Maciejowski (2000b), also fit in this framework. This underscores the generality of the subspace identification framework presented in Chapter 3.

As with the LPV methods, the number of rows of the data matrices grows exponentially with the block size of these data matrices. Hence, a large block size results in enormous computation times and memory requirements. Again, the subset-selection technique in Chapter 4 can be used to drastically reduce the number of rows in the data matrices. It makes bilinear subspace identification practically feasible.

Chapter 10 presents an alternative way to avoid the use of data matrices having enormous dimensions. It is shown that if a white-noise input signal is used to excite the bilinear system, this system can be approximated by a linear system with an extended input that consists of the original input plus second-order products between the inputs and the outputs. An errors-in-variables subspace method for linear time-invariant systems can be used to obtain consistent estimates of the system matrices A and C. The remaining system matrices can be estimated by solving a set of linear least-squares problems; however, these estimates are biased. Because a linear subspace method can be used, the number of rows in the data matrices grows linearly with the block size instead of exponentially, as in the conventional bilinear subspace methods. This obviously leads to an enormous reduction in computational complexity.

Chapter 11 describes two nonlinear optimization-based identification methods that can be used to improve the models obtained by bilinear subspace identification. The first method exploits the fact that the bilinear system is a special case of the LPV system. The projected gradient search algorithm for LPV system identification in Chapter 5 can be adapted to suit the bilinear identification problem. The advantage of this method is that it avoids the use of a special parameterization for the bilinear system. This is a big advantage, because until now no suitable special parameterization of the bilinear system has been described in the literature. The second method is based on the principle of separable least squares (SLS), described by Golub and Pereyra (1973). Application of SLS allows the elimination of the matrices B and D from the nonlinear optimization problem. When the modified nonlinear optimization problem that only depends on A, C, and F, has been solved, the matrices B and D follow by solving a linear least-squares problem.

The advantage of this approach is that it reduces the dimension of the parameter space in the nonlinear optimization problem. The SLS approach requires a parameterization of the matrices A, C, and F. Since no suitable parameterization has been described in the literature, it is proposed to use an output-normal parameterization for A and C, and to parameterize F by all its entries. For the bilinear example systems considered in this thesis, simulations confirm that this parameterization is suitable for identification. Nevertheless, the parameterization of bilinear state-space systems remains a topic for future research. To deal with process noise, an SLS approach is discussed that uses instrumental variables. However, simulation results show that a certain rank condition for the instrumental variables is not always satisfied, leading to a loss of accuracy. The simulations also show that the bias resulting from the 'standard' SLS method is usually very small; therefore the use of instrumental variables is not recommended.

In Chapter 12 the different subspace and nonlinear optimization-based identification methods are compared using extensive Monte-Carlo simulations. All methods yield acceptable results for white-noise input signals. For nonwhite noise input signals, the performance of all the methods slightly deteriorates, and is occasionally very bad. It might be that the white-noise signal excites the bilinear system better. Further research addressing the difference in performance for white-noise and nonwhite-noise signals is desirable. In addition, in Chapter 12 it is shown that the bilinear identification methods can be used to obtain an accurate model of a high-purity distillation column if only the high gain direction of the column is of interest.

Local Linear State-Space Systems

Local linear modeling is one of the many possibilities to approximate a nonlinear dynamical system. It is based on partitioning the whole operating range of the nonlinear system into multiple, smaller operating regimes and modeling the system for each of these regimes by a linear model. By making a weighted combination of these linear models, one hopes to describe the complete nonlinear behavior sufficiently accurately. In this thesis, the following weighted combination of local linear models is considered:

$$
\begin{aligned}
x_{k+1} &= \sum_{i=1}^{s} p_i(\phi_k)\Big(A_i x_k + B_i u_k + O_i\Big), \\
y_k &= C x_k,
\end{aligned}
$$

where s is the number of local models, $x_k \in \mathbb{R}^n$ the unknown state, $u_k \in \mathbb{R}^m$ the input, $y_k \in \mathbb{R}^\ell$ the output, and $p_i(\phi_k) \in \mathbb{R}$ the weighting for the ith model. The weighting vectors p_i are unknown functions of the scheduling vector $\phi_k \in \mathbb{R}^q$. This scheduling vector corresponds to the operating point of the system; typically it will depend on the input and the state. The weighting functions can be interpreted as model validity functions: they indicate which model or combination of models is active for a certain operating regime of the system. A weighted combination of local linear models can be used to approximate a smooth nonlinear system up to arbitrary accuracy by increasing the number of local models.

Note that the local linear state-space system is closely related to the LPV system: consider the time-varying parameters of the LPV system as model validity functions. The main difference is that for the LPV system it is assumed that the time-varying parameters are known, while for the local linear model structure the weighting functions have to be determined from input and output data.

The identification of local linear model structures has been studied mainly for input-output systems and for state-space systems of which the full state vector is measured. The case where only part of the state is measured is of course of more general interest. This thesis addresses the identification of local linear state-space systems where the state is not measured directly; only an observable linear combination of some of the states is available as an output. Normalized radial basis functions are used as weighting functions that combine the local state-space models to obtain a global description of the input-output behavior. The attempt to derive subspace identification methods for local linear models has not been successful. Therefore, only an optimization-based identification approach is described.

Chapter 15 presents an optimization-based identification procedure for the local linear model structure that is considered. The system matrices of the local models are fully parameterized. An iterative projected gradient search method —similar to the one used for LPV systems— is used to identify the local models and the centers and widths of the radial basis functions. The method deals with the nonuniqueness of the fully parameterized state-space representation by first calculating at every iteration the directions in which a change of parameters does not influence the input-output behavior of the model, and subsequently projecting these directions out of the parameter update.

Because of the recurrent nature of the local linear models, the gradient calculations needed in the projected gradient search method are governed by dynamic equations. For successful identification, these gradient calculations need to be stable. The stability is discussed for two special scheduling cases. When the scheduling is based only on the input signals, stability of the dynamic gradient calculations can be guaranteed, and hence the training is well-behaved. In the more general case where scheduling is also based on the output signals, two training methods are possible, each having their own advantages and disadvantages. One training method is based on scheduling with the measured outputs; this method behaves well during training, but the resulting model can perform poorly when simulated in free run. The other training method that uses the model output for scheduling does not have this problem, but can suffer from stability problems during training.

Because the identification procedure is based on solving a nonlinear optimization problem, an initial estimate of the local linear models is required. Since no subspace identification method is available to generate an initial set of local models and weighting functions, it is proposed to estimate a global linear model, and use it to initialize all the local models; in addition, the weighting functions are uniformly distributed over the operating range of the scheduling vector. With this initialization, the performance of the local linear models on the identification data will always be better than that of the global linear model.

In Chapter 16 several examples that have been studied in the literature before

are used to demonstrate the potential of the proposed methods. One of these examples involves the identification of a bioreactor.

Only the estimation of the models is addressed in this thesis. An important problem that should be addressed in future research is structure identification of local linear state-space models. Although for the examples that were presented, the structure of the model could be determined by trial and error, more sophisticated methods are needed for higher dimensional problems.

Samenvatting

Niet-lineaire systeemidentificatie: een toestandsaanpak

Het doel van systeemidentificatie is het direct bepalen van een dynamisch model met behulp van in- en uitgangsdata. Systeemidentificatie is een alternatief voor de meer klassieke manier van modelleren, die gebaseerd is op de basiswetten van bijvoorbeeld de natuurkunde, scheikunde en biologie. Modelleren op grond van basiswetten vereist veel specialistische kennis en een aanzienlijke tijdsinvestering. Het leidt vaak tot complexe modellen die niet geschikt zijn voor on-line toepassingen. Systeemidentificatie levert daarentegen vaak compacte modellen op, die gebruikt kunnen worden in *real-time* model-gebaseerde regelaars.

De identificatie van lineaire systemen heeft de laatste jaren veel aandacht gekregen. Lineaire systemen hebben hun nut bewezen in talrijke technische toepassingen. Voor de identificatie en regeling van deze systemen zijn behoorlijk wat theoretische resultaten afgeleid. Echter, de meeste praktische systemen vertonen inherent niet-lineair dynamisch gedrag. Daarom heeft het gebruik van lineaire modellen zijn beperkingen. Wanneer hoge eisen aan de prestatie gesteld worden, is het lineaire model niet langer nauwkeurig genoeg en moeten niet-lineaire modellen gebruikt worden.

In dit proefschrift worden nieuwe identificatietechnieken gepresenteerd voor drie specifieke typen niet-lineaire systemen: lineaire parameter-variërende toestandssystemen, bilineaire toestandssystemen en lokaal lineaire toestandssystemen. Ondanks het feit dat niet-lineaire systeemidentificatie meestal gericht is op niet-lineaire ingangs-uitgangsbeschrijvingen, gaat dit proefschrift in op toestandsbeschrijvingen. Toestandsbeschrijvingen worden behandeld omdat zij uitermate geschikt zijn voor systemen met meerdere in- en uitgangen en omdat er meestal minder parameters nodig zijn om het systeem te beschrijven dan met ingangs-uitgangsbeschrijvingen. Tevens kan gesteld worden dat een groot aantal niet-lineaire regelmethoden gebaseerd is op een toestandsmodel van het te regelen systeem.

Lineaire parameter-variërende toestandssystemen

Een lineair parameter-variërend (LPV) toestandssysteem is een lineair systeem waarin de systeemmatrices functies zijn van een bekende tijdsvariërende parametervector. Dit proefschrift behandelt de volgende LPV systemen met affiene parameterafhankelijkheid:

$$
\begin{aligned}
x_{k+1} &= A \left[\begin{array}{c} x_k \\ p_k \otimes x_k \end{array} \right] + B \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right], \\
y_k &= C \left[\begin{array}{c} x_k \\ p_k \otimes x_k \end{array} \right] + D \left[\begin{array}{c} u_k \\ p_k \otimes u_k \end{array} \right],
\end{aligned}
$$

waarbij $x_k \in \mathbb{R}^n$ de toestand is, $u_k \in \mathbb{R}^m$ de ingang, $y_k \in \mathbb{R}^\ell$ de uitgang, $p_k \in \mathbb{R}^s$ de tijdsvariërende parametervector en $\otimes$ het Kronecker product.

Een LPV systeem kan beschouwd worden als een niet-lineair systeem dat gelineariseerd is langs een tijdsvariërend traject, dat bepaald wordt door de tijdsvariërende parametervector p_k. Deze parametervector p_k bepaalt het werkpunt van het niet-lineaire systeem. Aangenomen wordt dat deze tijdsvariërende vector meetbaar is voor het regelen en identificeren. Dit is het geval in veel industriële toepassingen, zoals vliegtuig- en procesregelingen.

Het ontwerpen van regelaars voor LPV systemen is een populair onderzoeksgebied en een aantal veelbelovende technieken is hier al uit voortgekomen. Dit in tegenstelling tot de identificatie van LPV systemen, die nog in de kinderschoenen staat, hetgeen een studie naar nieuwe identificatiemethoden voor LPV systemen motiveert. In dit proefschrift worden methoden gepresenteerd voor de identificatie van LPV toestandssystemen die gebruik maken van metingen van de ingang, de uitgang en de tijdsvariërende parameters.

Hoofdstuk 3 beschrijft een opzet waarmee verschillende *subspace*-identificatiemethoden voor LPV systemen met een tijdsinvariante uitgangsvergelijking afgeleid kunnen worden, rekening houdend met de aanwezigheid van proces- en meetruis. *Subspace*-methoden verschenen als eerste in de literatuur voor lineaire tijdsinvariante systemen. Het zijn numeriek betrouwbare methoden die geen speciale parameterisatie van het toestandssysteem vereisen. Dit maakt ze aantrekkelijk in het gebruik voor systemen met meerdere in- en uitgangen.

Het afleiden van *subspace*-methoden voor LPV systemen draait om de reconstructie van de toestand in de tijd. Deze reconstructie is altijd een benadering, omdat de invloed van de begintoestand verwaarloosd moet worden. In LPV *subspace*-identificatie wordt de toestand benaderd door een lineaire combinatie van de rijen van bepaalde gestructureerde datamatrices, die uit de metingen van de ingang, uitgang en tijdsvariërende parameters opgebouwd worden. De lineaire combinatie is onbekend, maar volgt uit het oplossen van een kleinste-kwadratenprobleem. Afhankelijk van de combinatie van de datamatrices die gebruikt wordt om de toestand te benaderen, kunnen verschillende identificatiealgoritmen afgeleid worden. Een belangrijke instelparameter is de blokdimensie van de datamatrices. In het algemeen leidt een vergroting van de blokdimensie tot een hogere nauwkeurigheid in de toestandsbenadering, maar zij leidt ook tot een exponentiële toename van het aantal rijen in de datamatrices. Een grote blokdimensie kan leiden tot enorme rekentijden en een grote behoefte aan computergeheugen. Daarom is de keuze van de blokdimensie een afweging tussen nauwkeurigheid en het totaal aantal berekeningen. Met een grote blokdimensie kan eveneens het effect van eindige datalengte niet meer worden verwaarloosd en verslechteren de geschatte modellen.

Hoofdstuk 4 beschrijft een subset-selectiemethode die geïntegreerd kan worden in de LPV *subspace*-methoden, om het aantal rijen in de datamatrices te re-

duceren. Omdat het aantal rijen exponentieel groeit met de blokdimensie, kan LPV *subspace*-identificatie een enorme hoeveelheid computergeheugen vereisen. Deze hoeveelheid kan al voor multivariabele systemen van een lage orde de geheugencapaciteit van de huidige generatie computers overstijgen. Het reduceren van de dimensies van de datamatrices is cruciaal voor het praktisch toepasbaar maken van de methode voor systemen met meerdere in- en uitgangen. De subset-selectiemethode selecteert alleen de meest dominante rijen en neemt de overige rijen niet mee, ten koste van het introduceren van een benaderingsfout. Deze selectiemethode kan efficiënt geïmplementeerd worden met een RQ-factorisatie die de datamatrices rij voor rij bewerkt, zodat het niet nodig is om de volledige matrix in het geheugen op te slaan. De subset-selectiemethode zet de grote eisen aan de geheugencapaciteit echter gedeeltelijk om in lange rekentijden. Daarom zou de afleiding van efficiëntere methoden nader onderzocht moeten worden.

Alle behandelde LPV *subspace*-identificatiemethoden leveren slechts een benadering op van het systeem. Daarom is een verdere verbetering van de verkregen modellen gewenst. Hoofdstuk 5 beschrijft een iteratieve methode voor de identificatie van LPV systemen, die gebaseerd is op niet-lineaire optimalisatie. Het LPV model verkregen via *subspace*-identificatie vormt een goede beginschatting voor dit niet-lineaire optimalisatieprobleem. De methode gebaseerd op optimalisatie is een speciaal geval van de LFT-identificatiemethode beschreven door Lee en Poolla (1999), die het kiezen van een geschikte parameterisatie voor het LPV systeem omzeilt. Deze methode maakt gebruik van een volledige parameterisatie en past tijdens iedere iteratie alleen de parameters aan die de te optimaliseren kostenfunctie veranderen. Deze aanpak wordt de geprojecteerde gradiëntmethode genoemd, omdat de gradiënt die gebruikt wordt bij het oplossen van het optimalisatieprobleem zo geprojecteerd wordt, dat alleen de richtingen die de kostenfunctie veranderen, overblijven. Deze aanpak heeft numerieke voordelen ten opzichte van het gebruik van canonieke parameterisaties en kan gebruikt worden met uitgangsruis, evenals met een bepaald soort procesruis. De combinatie van de subset-selectie-*subspace*-identificatiemethode en de geprojecteerde gradiëntmethode vormt een krachtig gereedschap voor de identificatie van multivariabele LPV systemen.

Hoofdstuk 6 bevat Monte-Carlo simulaties voor verschillende LPV identificatiemethoden. Deze simulaties bevestigen dat een te kleine blokdimensie bij LPV *subspace*-identificatie leidt tot grote benaderingsfouten. Tevens bevestigen ze dat de blokdimensie niet te groot gekozen moet worden, omdat het effect van eindige datalengte dan niet langer verwaarloosbaar is en opnieuw grote benaderingsfouten het gevolg zijn. Daarnaast laten de simulaties zien dat dimensiereductie met de subset-selectiemethode haalbaar is. Vaak kan een verrassend grote dimensiereductie bereikt worden, zonder dat de kwaliteit van de modellen erg achteruitgaat. De selectie van 180 rijen uit $180\,000$ levert bijvoorbeeld slechts 5% kwaliteitsverlies op. Ten slotte is getoond dat de LPV identificatiemethoden met succes gebruikt kunnen worden voor het modelleren van de dynamica van een helikopterrotor.

Bilineaire toestandssystemen

Een bilineair systeem is een eenvoudige niet-lineaire uitbreiding van een lineair

systeem, dat beschreven kan worden als:

$$
\begin{aligned}
x_{k+1} &= Ax_k + F(u_k \otimes x_k) + Bu_k, \\
y_k &= Cx_k + Du_k,
\end{aligned}
$$

waarbij $x_k \in \mathbb{R}^n$ de toestand is, $u_k \in \mathbb{R}^m$ de ingang, $y_k \in \mathbb{R}^\ell$ de uitgang en $\otimes$ het Kronecker product.

De belangrijkste motivatie voor het bestuderen van discrete-tijd bilineaire systemen is hun eenvoudige structuur in vergelijking met andere niet-lineaire modellen, zoals bijvoorbeeld neurale netwerken. Ondanks hun eenvoudige structuur kunnen bilineaire systemen toch interessant niet-lineair gedrag beschrijven: elke eindige discrete-tijd Volterrareeks met tijdsinvariante scheidbare kernen kan gerealiseerd worden door een discrete-tijd bilineair systeem. Bilineaire systemen zijn uitgebreid bestudeerd en toegepast om diverse praktijkgerichte problemen op te lossen.

De meeste bestaande identificatiemethoden voor bilineaire systemen werken met ingangs-uitgangssystemen of met toestandssystemen in *phase-variable*-vorm. Slechts enkele methoden werken met een algemene toestandsbeschrijving waarbij geen speciale structuur voor de systeemmatrices wordt aangenomen. In dit proefschrift worden *subspace*-methoden en methoden gebaseerd op niet-lineaire optimalisatie beschreven voor de identificatie van algemene bilineaire toestandssystemen, gebruik makend van in- en uitgangsdata.

Hoofdstuk 9 beschrijft verscheidene *subspace*-identificatiemethoden voor bilineaire systemen. Omdat het bilineaire systeem gezien kan worden als een speciaal geval van het LPV systeem (neem de tijdsvariënde parameter p_k gelijk aan de ingang u_k), kan de algemene opzet voor LPV *subspace*-identificatie uit hoofdstuk 3 gebruikt worden voor het afleiden van bilineaire *subspace*-identificatiemethoden. De *subspace*-algoritmen die eerder in de literatuur zijn beschreven, die van Favoreel (1999) en van Chen en Maciejowski (2000b), passen ook binnen deze opzet, wat de algemeenheid ervan benadrukt.

Zoals met de LPV methoden, groeit het aantal rijen in de datamatrices exponentieel met de blokdimensie van deze matrices. Daarom leidt een grote blokdimensie tot enorme rekentijden en een grote behoefte aan computergeheugen. Opnieuw kan de subset-selectietechniek uit hoofdstuk 4 gebruikt worden om het aantal rijen in de datamatrices drastisch te reduceren. Dit maakt bilineaire *subspace*-identificatie praktisch haalbaar.

Hoofdstuk 10 bevat een alternatieve manier om het gebruik van datamatrices met enorme dimensies te omzeilen. Het bilineaire systeem kan, indien een witte ruis als ingangssignaal gebruikt wordt, benaderd worden door een lineair systeem met een uitgebreide ingang die bestaat uit de originele ingang en tweede-orde producten tussen de in- en de uitgangen. Door een *errors-in-variables-subspace*-methode voor lineaire tijdsinvariante systemen te gebruiken, kunnen consistente schattingen van de systeemmatrices A en C verkregen worden. De overige systeemmatrices worden geschat door het oplossen van een stelsel lineaire kleinste-kwadratenvergelijkingen. Deze schattingen zijn echter niet zuiver. Vanwege het gebruik van een lineaire *subspace*-methode groeit het aantal rijen in de datamatrices lineair met de blokdimensie, in plaats van exponentieel zoals in de conventionele bilineaire *subspace*-methoden. Dit leidt uiteraard tot een enorme afname van

de benodigde rekenkracht.

Hoofdstuk 11 beschrijft twee identificatiemethoden gebaseerd op niet-lineaire optimalisatie, die gebruikt kunnen worden om de modellen verkregen met *subspace*-identificatie te verbeteren. De eerste methode benut het feit dat het bilineaire systeem een speciaal geval is van een LPV systeem. De geprojecteerde gradiëntmethode voor LPV identificatie uit hoofdstuk 5 kan geschikt gemaakt worden voor het oplossen van bilineaire identificatieproblemen. Het voordeel van deze methode is dat er geen speciale parameterisatie van het bilineaire systeem nodig is. Dit is een groot voordeel, omdat er geen geschikte parameterisatie voor bilineaire systemen in de literatuur beschreven is. De tweede methode is gebaseerd op het principe van scheidbare kleinste-kwadraten (*separable least squares* of kortweg SLS), beschreven door Golub en Pereyra (1973). Door het toepassen van SLS kunnen de matrices B en D uit het niet-lineaire optimalisatieprobleem verwijderd worden. Na het oplossen van het aangepaste niet-lineaire optimalisatieprobleem dat alleen afhangt van A, C en F, volgen de matrices B en D uit het oplossen van een kleinste-kwadratenprobleem. Het voordeel van deze aanpak is dat de dimensie van de parameterruimte in het niet-lineaire optimalisatieprobleem kleiner wordt. De SLS aanpak vereist een parameterisatie van de matrices A, C en F. Omdat een geschikte parameterisatie in de literatuur ontbreekt, wordt voorgesteld om een *output-normal*-parameterisatie voor A en C te gebruiken en om F te parameteriseren door al zijn elementen. Voor de bilineaire systemen die in dit proefschrift behandeld worden, hebben simulaties aangetoond dat deze parameterisatie bruikbaar is voor identificatiedoeleinden. Niettemin blijft de parameterisatie van bilineaire toestandssystemen een onderwerp voor toekomstig onderzoek. Om met ruis om te gaan wordt een SLS aanpak beschreven die gebruik maakt van instrumentele variabelen. Simulatieresultaten laten echter zien dat niet altijd aan een bepaalde rangconditie voor deze instrumentele variabelen voldaan wordt, wat leidt tot een verlies aan nauwkeurigheid. De simulaties laten ook zien dat de onzuiverheid in de schattingen van de 'standaard' SLS methode meestal erg klein is. Daarom wordt het gebruik van instrumentele variabelen afgeraden.

Hoofdstuk 12 vergelijkt de verschillende *subspace*-methoden en de op niet-lineaire optimalisatie gebaseerde methoden met elkaar, door middel van uitgebreide Monte-Carlo simulaties. Alle methoden leveren acceptabele resultaten op voor witte ruis ingangssignalen. Voor niet-witte ruis ingangssignalen neemt de nauwkeurigheid af. Het is mogelijk dat het witte ruissignaal het bilineaire systeem beter exciteert. Nader onderzoek naar het verschil in prestatie bij witte en niet-witte ruissignalen is gewenst. Hoofdstuk 12 illustreert ook dat de bilineaire identificatiemethoden gebruikt kunnen worden om een nauwkeurig model van een hoge-zuiverheids-destillatiekolom te verkrijgen, indien alleen de dominante versterkingsrichting van de kolom van belang is.

Lokaal lineaire toestandssystemen

Lokaal lineair modelleren is één van de vele mogelijkheden om een niet-lineair dynamisch systeem te benaderen. Het is gebaseerd op het opdelen van het volledige werkbereik van het niet-lineaire systeem in meerdere kleinere werkgebieden en het modelleren van het systeem voor elk van deze gebieden, door middel van

een lineair model. Door een gewogen combinatie van deze lineaire modellen te maken, kan het volledige niet-lineaire gedrag voldoende nauwkeurig worden beschreven. Dit proefschrift behandelt de volgende gewogen combinatie van lokale lineaire modellen:

$$
\begin{aligned}
x_{k+1} &= \sum_{i=1}^{s} p_i(\phi_k)\Big(A_i x_k + B_i u_k + O_i \Big), \\
y_k &= C x_k,
\end{aligned}
$$

waarbij s het aantal lokale modellen is, $x_k \in \mathbb{R}^n$ de onbekende toestand, $u_k \in \mathbb{R}^m$ de ingang, $y_k \in \mathbb{R}^\ell$ de uitgang en $p_i(\phi_k) \in \mathbb{R}$ de weging voor het i^{de} model. De wegingsvectoren p_i zijn onbekende functies van de planningsvector $\phi_k \in \mathbb{R}^q$. Deze planningsvector komt overeen met het werkpunt van het systeem; in het algemeen hangt deze vector af van de ingang en de toestand. De weegfuncties kunnen geïnterpreteerd worden als modelgeldigheidsfuncties: ze geven aan welk model of welke combinatie van modellen actief is voor een bepaald werkgebied van het systeem. Door het aantal lokale modellen te vergroten, kan het systeem gebruikt worden om een differentieerbaar niet-lineair systeem te benaderen tot elke gewenste nauwkeurigheid.

Het is belangrijk te beseffen dat het lokaal lineaire toestandssysteem nauw verwant is met het LPV systeem: beschouw de tijdsvariërende parameters van het LPV systeem als modelgeldigheidsfuncties. Het belangrijkste verschil is dat de tijdsvariërende parameters voor het LPV systeem bekend verondersteld worden, terwijl voor de lokale lineaire modelstructuur de weegfuncties uit in- en uitgangsdata bepaald moeten worden.

De identificatie van lokaal lineaire modelstructuren is vooral bestudeerd voor ingangs-uitgangssystemen en voor toestandssystemen waarvan de volledige toestandsvector gemeten kan worden. Het geval waarin alleen een deel van de toestand gemeten kan worden, is van meer algemeen belang. Dit proefschrift behandelt de identificatie van lokale lineaire toestandssystemen waarvan de toestand niet direct gemeten wordt; alleen een waarneembare lineaire combinatie van enkele toestanden is beschikbaar als uitgang. Genormaliseerde radiale basisfuncties worden gebruikt als de weegfuncties die de lokale toestandsmodellen combineren tot een globale beschrijving van het ingangs-uitgangsgedrag. Op dit moment is het ontwikkelen van *subspace*-identificatiemethoden voor lokaal lineaire modellen nog niet succesvol gebleken. Daarom wordt alleen een methode gebaseerd op optimalisatie beschreven.

Hoofdstuk 15 beschrijft een op optimalisatie gebaseerde identificatiemethode voor de behandelde lokaal lineaire modelstructuur. De systeemmatrices van de lokale modellen zijn volledig geparameteriseerd. Een iteratieve geprojecteerde gradiëntmethode —vergelijkbaar met de methode voor LPV systemen— wordt gebruikt om de lokale modellen en de centra en breedten van de radiale basisfuncties te identificeren. De methode gaat als volgt om met de niet-uniciteit van de volledig geparameteriseerde toestandsrepresentatie: tijdens elke iteratie worden de richtingen berekend waarin een verandering van de parameters geen invloed heeft op het ingangs-uitgangsgedrag van het model en vervolgens worden deze richtingen uit de aanpassing van de parameters geprojecteerd.

Vanwege het recurrente gedrag van de lokaal lineaire modellen worden de berekeningen van de gradiënt beheerst door dynamische vergelijkingen. Voor een succesvolle identificatie moeten deze gradiëntberekeningen stabiel zijn. Voor twee speciale gevallen van de planningsvector wordt de stabiliteit besproken. Wanneer de planningsvector alleen afhangt van de ingangssignalen, kan de stabiliteit van de gradiëntberekeningen gegarandeerd worden en verloopt de training goed. In het meer algemene geval waarbij de planningsvector ook afhangt van de uitgangssignalen zijn twee trainingsmethoden mogelijk, elk met eigen voor- en nadelen. De ene trainingsmethode is gebaseerd op het gebruik van de gemeten uitgang in de planningsvector. Voor deze methode verloopt de training goed, maar het resulterende model kan een slechte prestatie leveren wanneer het vrijlopend gesimuleerd wordt. De andere trainingsmethode die de modeluitgang voor de planningsvector gebruikt, heeft dit probleem niet, maar kan lijden onder stabiliteitsproblemen gedurende de training.

Omdat de identificatieprocedure gebaseerd is op het oplossen van een nietlineair optimalisatieprobleem, is een beginschatting van de lokaal lineaire modellen nodig. Daar er geen *subspace*-methode beschikbaar is om een initiële verzameling lokale modellen en weegfuncties te genereren, wordt voorgesteld om een globaal lineair model te schatten en alle lokale modellen hiermee te initialiseren en daarnaast de weegfuncties uniform over het werkgebied van de planningsvector te verdelen. Met deze initialisatie zal de prestatie van de lokaal lineaire modellen voor de identificatiedata altijd beter zijn dan die van het globale lineaire model.

Hoofdstuk 16 toont de potentie van de voorgestelde methoden door ze toe te passen op verschillende voorbeelden die al eerder in de literatuur verschenen. Eén van deze voorbeelden is de identificatie van een bioreactor.

Dit proefschrift behandelt alleen het schatten van de modellen. De structuuridentificatie van lokaal lineaire toestandsmodellen is een belangrijk probleem dat in toekomstig onderzoek nader bekeken dient te worden. Ondanks het feit dat voor de gepresenteerde voorbeelden de structuur van het model door herhaaldelijk proberen bepaald kon worden, zijn intelligentere methoden nodig voor problemen met grotere dimensies.

Notation and Symbols

$\mathbb{Z}$	the set of integers
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^n$	the set of real-valued n-dimensional vectors
$\mathbb{R}^{m \times n}$	the set of real-valued m by n matrices
∞	infinity
$\in$	belongs to
$=$	equal
$\approx$	approximately equal
$:=$	defined as
$\square$	end of proof
$\otimes$	Kronecker product (p. 14)
$\odot$	Khatri-Rao product (p. 27)
I_n	the n by n identity matrix
$[A]_{i,j}$	the (i,j)th entry of the matrix A
$A(i,:\,)$	the ith row of the matrix A
$A(:\,,i)$	the ith column of the matrix A
A^T	the transpose of the matrix A
A^{-1}	the inverse of the matrix A
$A^{\dagger}$	the Moore-Penrose pseudo inverse of the matrix A
$A^{1/2}$	the symmetric positive definite square root of the matrix A
$\mathrm{diag}(a_1, a_2, \ldots, a_n)$	an $n \times n$ diagonal matrix whose (i,i)th entry is a_i
$\det(A)$	the determinant of the matrix A

$\text{range}(A)$	the column space of the matrix A
$\text{rank}(A)$	the rank of the matrix A
$\text{trace}(A)$	the trace of the matrix A
$\text{vec}(A)$	a vector constructed by stacking the columns of the matrix A underneath each other; starting with the left most column
$\|A\|_2$	the 2-norm of the matrix A
$\|A\|_F$	the Frobenius norm of the matrix A (p. 35)
$[x]_i$	the ith entry of the vector x
$\text{diag}(x)$	an $n \times n$ diagonal matrix, whose (i,i)th entry is the ith entry of the n-dimensional vector x
$\|x\|_2$	the 2-norm of the vector x
$\lim$	limit
$\min$	minimum
$\max$	maximum
$\sup$	supremum (least upper bound)
$a \leftarrow b$	substitute a by b
$\mathbf{E}[\,\cdot\,]$	statistical expected value
$\text{var}(\cdot)$	variance of a quasi-stationary signal
w. p. 1	with probability one

List of Abbreviations

LFT	Linear Fractional Transformation
LMS	Least-Mean-Square
LPV	Linear Parameter-Varying
LTI	Linear Time-Invariant
MIMO	Multiple Input, Multiple Output
MOESP	Multivariable Output Error State sPace
MSE	Mean Square Error
N4SID	Numerical algorithm for Subspace IDentification
RLS	Recursive Least Squares
SISO	Single Input, Single Output
SLS	Separable Least Squares
SNR	Signal-to-Noise Ratio
SVD	Singular Value Decomposition
VAF	Variance-Accounted-For

References

Ahmed, M. S. (1986). Parameter estimation in bilinear systems by instrumental variable methods. *International Journal of Control* **44**(4), 1177–1183.

Azimzadeh, F., O. Galán and J. A. Romagnoli (2001). On-line optimal trajectory control for a fermentation process using multi-linear models. *Computers and Chemical Engineering* **25**(1), 15–26.

Babuška, Robert (1997). *Fuzzy Modeling and Identification*. Ph. D. thesis, Department of Electrical Engineering, Delft University of Technology, Delft, The Netherlands. ISBN 90-9010153-5.

Babuška, Robert and Henk Verbruggen (1996). An overview of fuzzy modeling for control. *Control Engineering Practice* **4**(11), 1593–1606.

Babuška, Robert and Henk Verbruggen (1997). Fuzzy set methods for local modelling and identification. See Murray-Smith and Johansen (1997b), Chapter 2, pp. 75–100.

Baheti, R. S., Ronald R. Mohler and H. A. Spang III (1980). Second-order correlation method for bilinear system identification. *IEEE Transactions on Automatic Control* **25**(6), 1141–1146.

Bamieh, Bassam and Laura Giarré (1999). Identification of linear parameter varying models. In *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona (December), pp. 1505–1510.

Bamieh, Bassam and Laura Giarré (2000). Identification for a general class of LPV models. In *Preprints of the IFAC Symposium on System Identification*, Santa Barbara, California (June).

Bamieh, Bassam and Laura Giarré (2001). LPV models: Identification for gain scheduling. In *Proceedings of the European Control Conference 2001*, Porto, Portugal (September).

Banerjee, A., Y. Arkun, B. Ogunnaike and R. Pearson (1997). Estimation of nonlinear systems using linear multiple models. *AIChE Journal* **43**(5), 1204–1226.

Belforte, G. (1992). Parameter identification for models with bounded errors in all variables. *Systems and Control Letters* **19**(6), 425–428.

Billings, S. A. (1980). Identification of nonlinear systems: A survey. *IEE Proceedings Part D: Control Theory and Applications* **127**(6), 272–285.

Billings, S. A., H. B. Jamaluddin and S. Chen (1992). Properties of neural networks with applications to modelling non-linear dynamical systems. *International Journal of Control* **55**(1), 193–224.

Billings, S. A. and W. S. F. Voon (1986). Correlation based model validity tests for non-linear models. *International Journal of Control* **44**(1), 235–244.

Billings, S. A. and W. S. F. Voon (1987). Piecewise linear identification of non-linear systems. *International Journal of Control* **46**(1), 215–235.

Bittanti, Sergio and Marco Lovera (1996). On the zero dynamics of helicopter rotor loads. *European Journal of Control* **2**(1), 57–68.

Bloemen, Hayco (2002). *Predictive Control Based on Black-Box State-Space Models*. Ph. D. thesis, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. ISBN 90-77017-44-5.

Bloemen, Hayco, C. T. Chou, Ton van den Boom, Vincent Verdult, Michel Verhaegen and Ton Backx (2000). Wiener MPC for high purity dual composition control of a distillation column. In *Process Control and Instrumentation 2000*, Glasgow, Scotland (July), pp. 198–203.

Bloemen, Hayco, C. T. Chou, Ton van den Boom, Vincent Verdult, Michel Verhaegen and Ton Backx (2001). Wiener model identification and predictive control for dual composition control of a distillation column. *Journal of Process Control* **11**(6), 601–620.

Bloemen, Hayco, C. T. Chou, Vincent Verdult, Ton van den Boom, Ton Backx and Michel Verhaegen (1999). Economic benefits of using nonlinear models for high purity distillation column control. Technical Report UT/TN/SCE-1999.001, Faculty of Applied Physics, University of Twente, Enschede, The Netherlands.

Bloemen, Hayco, Vincent Verdult, Ton van den Boom and Michel Verhaegen (2001). Bilinear versus linear MPC: Application to a polymerization process. Accepted for presentation at *IFAC World Congress 2002*, Barcelona, Spain (July).

Bomberger, John D. (1997). *Radial Basis Function Network Models for Process Identification*. Ph. D. thesis, Department of Chemical Engineering, University of California Santa Barbara, Santa Barbara, California.

Bomberger, John D. and Dale E. Seborg (1997). Estimation of model order from input-output data applied to radial basis function network identification. In *Preprints of the IFAC Symposium on Advanced Control of Chemical Processes*, Banff, Canada (June), pp. 31–36.

Bose, Tamal and Mei-Qin Chen (1995). Conjugate gradient method in adaptive bilinear filtering. *IEEE Transactions on Signal Processing* **43**(6), 1503–1508.

Bottou, Léon and Vladimir Vapnik (1992). Local learning algorithms. *Neural Computation* **4**(6), 888–900.

Boukhris, Anass, Gilles Mourot and José Ragot (1999). Non-linear dynamic system identification: A multi-model approach. *International Journal of Control* **72**(7/8), 591–604.

Breiman, Leo (1993). Hinging hyperplanes for regression, classification and function approximation. *IEEE Transactions on Information Theory* **39**(3), 999–1013.

Brewer, J. (1987). Bilinear equations in ecology. In Madan G. Singh (Ed.), *Systems and Control Encyclopedia: Theory, Technology, Applications*, Volume 1, pp. 414–417. Oxford: Pergamon Press.

Brewer, John W. (1978). Kronecker products and matrix calculus in system theory. *IEEE Transactions on Circuits and Systems* **25**(9), 772–781.

Brillinger, David R. (1990). A study of second- and third-order spectral procedures and maximum likelihood in the identification of a bilinear system. *IEEE Transactions on Acoustics, Speech and Signal Processing* **38**(7), 1238–1245.

Brillinger, David R. (1994). Some basic aspects and uses of higer-order spectra. *Signal Processing* **36**(3), 239–249.

Bruls, Johan, C. T. Chou, Bert Haverkamp and Michel Verhaegen (1999). Linear and non-linear system identification using separable least-squares. *European Journal of Control* **5**(1), 116–128.

Bruni, Carlo, Gianni Dipillo and Giorgio Koch (1974). Bilinear systems: An appealing class of nearly linear systems in theory and applications. *IEEE Transactions on Automatic Control* **19**(4), 334–348.

Cao, S. G., N. W. Rees and G. Feng (1997). Analysis and design for a class of complex control systems part I: Fuzzy modelling and identification. *Automatica* **33**(6), 1017–1028.

Cerone, V. (1995). Parameter bounds for a class of discrete bilinear systems from records with bounded output errors. *International Journal of Adaptive Control and Signal Processing* **9**(1), 63–70.

Chen, Huixin and Jan Maciejowski (2000a). An improved subspace identification method for bilinear systems. In *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia (December).

Chen, Huixin and Jan Maciejowski (2000b). New subspace identification methods for bilinear systems. Technical Report CUED/F-INFENG/TR.357, Department of Engineering, University of Cambridge, Cambridge, United Kingdom. Submitted to Automatica.

Chen, Huixin and Jan Maciejowski (2000c). Subspace identification of combined deterministic-stochastic bilinear systems. In *Preprints of the IFAC Symposium on System Identification*, Santa Barbara, California (June).

Chen, Huixin and Jan Maciejowski (2000d). Subspace identification of deterministic bilinear systems. In *Proceedings of the 2000 American Control Conference*, Chicago, Illinois (June).

Chen, Huixin, Jan Maciejowski and Chris Cox (2001). Unbiased bilinear subspace system identification methods. In *Proceedings of the European Control Conference 2001*, Porto, Portugal (September).

Chen, Huixin, Alan S. I. Zinober and Rongyao Ruan (1996). Strong consistency and convergence rate of parameter identification for bilinear systems. *International Journal of Control* **63**(5), 907–919.

Chen, S. and S. A. Billings (1989). Representations of non-linear systems: the NARMAX model. *International Journal of Control* **49**(3), 1013–1032.

Cheng, Bing and Ning-Show Hsu (1982). Analysis and parameter estimation of bilinear systems via block-pulse functions. *International Journal of Control* **36**(1), 53–65.

Choi, Jin Young and Jay A. Farrell (2000). Nonlinear adaptive control using networks of piecewise linear approximators. *IEEE Transactions on Neural Networks* **11**(2), 390–401.

Chou, C. T., Hayco Bloemen, Vincent Verdult, Ton van den Boom, Ton Backx and Michel Verhaegen (1998). Identification and model predictive control of high purity distillation columns using Wiener models: A preliminary report. Technical Report TUD/ITS/ET/SCE98.008, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands.

Chou, C. T., Hayco Bloemen, Vincent Verdult, Ton van den Boom, Ton Backx and Michel Verhaegen (2000). Nonlinear identification of high purity distillation columns. In *Preprints of the IFAC Symposium on System Identification*, Santa Barbara, California (June).

Chou, C. T. and Michel Verhaegen (1997). Subspace algorithms for the identification of multivariable dynamic errors-in-variables models. *Automatica* **33**(10), 1857–1869.

Chou, C. T. and Michel Verhaegen (1999). Identification of Wiener models with process noise. In *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona (December), pp. 598–603.

Coleman, Thomas, Mary Ann Branch and Andrew Grace (1999). *Optimization Toolbox For Use with Matlab: User's Guide*. Natick, Massachusetts: The MathWorks.

Crouch, P. E. (1987). Realization of finite Volterra series. In Madan G. Singh (Ed.), *Systems and Control Encyclopedia: Theory, Technology, Applications*, Volume 6, pp. 3956–3962. Oxford: Pergamon Press.

Dai, Heping and Naresh K. Sinha (1989). Robust recursive least squares method with modified weights for bilinear system identification. *IEE Proceedings Part D: Control Theory and Applications* **136**(3), 122–126.

Dai, Heping and Naresh K. Sinha (1991a). A robust off-line method for system identification: Robust iterative least squares method with modified residuals. *Journal of Dynamic Systems, Measurement, and Control* **113**(4), 597–603.

Dai, Heping and Naresh K. Sinha (1991b). Robust recursive instrumental variable method with modified weights for bilinear system identification. *IEEE Transactions on Industrial Electronics* **38**(1), 1–7.

Dai, Heping and Naresh K. Sinha (1992). Robust identification of systems using block-pulse functions. *IEE Proceedings Part D: Control Theory and Applications* **139**(3), 308–316.

Dai, Heping, Naresh K. Sinha and S. C. Puthenpura (1989). Robust combined estimation of states and parameters of bilinear systems. *Automatica* **25**(4), 613–616.

Daniel-Berhe, Sequare and Heinz Unbehauen (1996). Application of the Hartley modulating functions method for the identification of the bilinear dynamics of a DC motor. In *Proceedings of the 35th IEEE Conference on Decision and Control*, Kobe, Japan (December), pp. 1533–1538.

Daniel-Berhe, Sequare and Heinz Unbehauen (1998a). Bilinear continuous-time systems identification via Hartley-based modulating functions. *Automatica* **34**(4), 499–503.

Daniel-Berhe, Sequare and Heinz Unbehauen (1998b). State space identification of bilinear continuous-time canonical systems via batch scheme Hartley modulating functions approach. In *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, Florida (December), pp. 4482–4487.

Dasgupta, Soura, Yash Shrivastava and George Krenzer (1992). Persistent excitation in bilinear systems. *IEEE Transactions on Automatic Control* **36**(3), 305–313.

David, Benoit and Georges Bastin (2001). An estimator of the inverse covariance matrix and its application to ML parameter estimation in dynamical systems. *Automatica* **37**(1), 99–106.

de Bruin, H. A. E. and B. Roffel (1996). A new identification method for fuzzy linear models of nonlinear dynamic systems. *Journal of Process and Control* **6**(5), 277–293.

Desai, Uday B. (1986). Realization of bilinear stochastic systems. *IEEE Transactions on Automatic Control* **31**(2), 189–192.

Diaz, Hernando and Alan A. Desrochers (1988). Modeling of nonlinear discrete-time systems from input-output data. *Automatica* **24**(5), 629–641.

Dunoyer, A., L. Balmer, K. J. Burnham and D. J. G. James (1997). On the discretization of single-input single-output bilinear systems. *International Journal of Control* **68**(2), 361–372.

Ernst, Susanne (1998). Hinging hyperplane trees for approximation and identifcation. In *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, Florida (December), pp. 1266–1271.

España, M. and D. Landau (1975). Bilinear approximation of the distillation process. *Ricerche di Automatica* **6**(1), 20–46.

España, M. and D. Landau (1978). Reduced order bilinear models for distillation columns. *Automatica* **14**(4), 345–355.

Favoreel, Wouter (1999). *Subspace Methods for Identification and Control of Linear and Bilinear Systems*. Ph. D. thesis, Faculty of Engineering, K. U. Leuven, Leuven, Belgium. ISBN 90-5682-216-0. Available from Internet: `http://www.esat.kuleuven.ac.be/~sistawww/cgi-bin/pub.pl` (Retrieved April 26, 2001).

Favoreel, Wouter and Bart De Moor (1998). Subspace identification of bilinear systems. In *Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems*, Padova, Italy (July).

Favoreel, Wouter, Bart De Moor and Peter Van Overschee (1996). Subspace identification of bilinear systems subject to white inputs. Technical Report ESAT-SISTA/TR 1996-53I, Faculty of Engineering, K. U. Leuven, Leuven, Belgium. Available from Internet: `http://www.esat.kuleuven.ac.be/~sistawww/cgi-bin/pub.pl` (Retrieved April 26, 2001).

Favoreel, Wouter, Bart De Moor and Peter Van Overschee (1997a). A bilinear extension of subspace identification for systems subject to white inputs. In *Proceedings of the 1997 American Control Conference*, Albuquerque, New Mexico (June), pp. 607–611.

Favoreel, Wouter, Bart De Moor and Peter Van Overschee (1997b). Subspace identification of balanced deterministic systems subject to white inputs. In *Proceedings of the European Control Conference 1997*, Brussels, Belgium (July).

Favoreel, Wouter, Bart De Moor and Peter Van Overschee (1999). Subspace identification of bilinear systems subject to white inputs. *IEEE Transactions on Automatic Control* **44**(6), 1157–1165.

Fischer, Martin, Oliver Nelles and Rolf Isermann (1998). Adaptive predictive control of a heat exchanger based on a fuzzy model. *Control Engineering Practice* **6**(2), 259–269.

Fnaiech, Farhat and Lennart Ljung (1987). Recursive identification of bilinear systems. *International Journal of Control* **45**(2), 453–470.

Foss, Bjarne A., Tor Arne Johansen and A. V. Sørensen (1995). Nonlinear predictive control using local models: Applied to a batch fermentation process. *Control Engineering Practice* **3**(3), 389–396.

Friedmann, P. P. and T. Millott (1995). Vibration reduction in rotorcraft using active control: a comparison of various approaches. *Journal of Guidance, Control and Dynamics* **18**(4), 664–673.

Gabr, M. M. and T. Subba Rao (1984). On the identification of bilinear systems from operating records. *International Journal of Control* **40**(1), 121–128.

Gan, Chengyu and Kourosh Danai (2000). Model-based recurrent neural network for modeling nonlinear dynamic systems. *IEEE Transactions on Systems, Man, and Cybernetics—PartB: Cybernetics* **30**(2), 344–351.

Giannakis, Georgios B. and Erchin Serpedin (2001). A bibliography on nonlinear system identification. *Signal Processing* **81**(3), 533–580.

Glentis, George-Othon A., Panos Koukoulas and Nicholas Kalouptsidis (1999). Efficient algorithms for Volterra system identification. *IEEE Transactions on Signal Processing* **47**(11), 3042–3057.

Gollee, Henrik and Kenneth J. Hunt (1997). Nonlinear modelling and control of electrically stimulated muscle: a local model network approach. *International Journal of control* **68**(6), 1259–1288.

Gollee, Henrik, Kenneth J. Hunt, Nick Donaldson and J. C. Jarvis (1997). Modelling of electrically stimulated muscle. See Murray-Smith and Johansen (1997b), Chapter 3, pp. 101–120.

Golub, Gene H. and V. Pereyra (1973). The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal of Numerical Analysis* **10**(2), 413–432.

Golub, Gene H. and Charles F. Van Loan (1996). *Matrix Computations* (third ed.). Baltimore, Maryland: The Johns Hopkins University Press. ISBN 0-8018-5414-8.

Grimmett, Geoffrey R. and David R. Stirzaker (1983). *Probability and Random Processes*. Oxford: Oxford University Press. ISBN 0-19-853185-0.

Haber, R. and Heinz Unbehauen (1990). Structure identification of nonlinear dynamic systems: A survey on input/output approaches. *Automatica* **26**(4), 651–677.

Hafner, M., M. Schüler, Oliver Nelles and Rolf Isermann (2000). Fast neural networks for diesel engine control design. *Control Engineering Practice* **8**(11), 1211–1221.

Han, Seokwon, Jinyoung Kim and Koengmo Sung (1996). Extended generalized total least squares method for the identification of bilinear systems. *IEEE Transactions on Signal Processing* **44**(4), 1015–1018.

Hanzon, B. and R. L. M. Peeters (1997). Balanced parameterizations of stable SISO all-pass systems in discrete-time. Technical Report M 97-05, Department of Mathematics, Maastricht University, Maastricht, The Netherlands.

Hara, Shinji and Katsuhisa Furuta (1977). Observability for bilinear systems. *International Journal of Control* **26**(4), 559–572.

Haverkamp, Bert (2001). *State Space Identification: Theory and Practice*. Ph. D. thesis, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands.

Haykin, Simon (1999). *Neural Networks: A Comprehensive Foundation* (second ed.). Upper Saddle River, New Jersey: Prentice Hall. ISBN 0-13-273350-1.

He, Xiangdong and Haruhiko Asada (1993). A new method for identifying orders of input-output models for nonlinear dynamic systems. In *Proceedings of the 1993 American Control Conference*, San Francisco, California (June), pp. 2520–2523.

Horn, Roger A. and Charles R. Johnson (1991). *Topics in Matrix Analysis*. Cambridge, United Kingdom: Cambridge University Press. ISBN 0-521-30587-X.

Hunt, Kenneth J., Roland Haas and Roderick Murray-Smith (1996). Extending the functional equivalence of radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks* **7**(3), 776–781.

Hunt, Kenneth J., Tor Arne Johansen, Jens Kalkkuhl, Hans Fritz and Th. Göttsche (2000). Speed control design for an experimental vehicle using a generalized gain scheduling approach. *IEEE Transactions on Control Systems Technology* **8**(3), 381–395.

Hunt, Kenneth J., Jens Kalkkuhl, Hans Fritz and Tor Arne Johansen (1996). Constructive empirical modelling of longitudinal vehicle dynamics using local model networks. *Control Engineering Practice* **4**(2), 167–178.

Hwang, C. and M. Y. Chen (1986). Analysis and parameter identification of bilinear systems via shifted Legendre polynomials. *International Journal of Control* **44**(2), 351–362.

Inagaki, Makato and Hitoshi Mochizuki (1984). Bilinear system identification by Volterra kernels estimation. *IEEE Transactions on Automatic Control* **29**(8), 746–749.

Isidori, Alberto (1973). Direct construction of minimal bilinear realizations from nonlinear input-output maps. *IEEE Transactions on Automatic Control* **18**(6), 626–631.

Isidori, Alberto and Antonio Ruberti (1973). Realization theory of bilinear systems. In D. Q. Mayne and R. W. Brockett (Eds.), *Geometric Methods in System Theory*, pp. 83–130. Dordrecht, The Netherlands: D. Reidel Publishing Company. ISBN 90-277-0415-5.

Jansson, Magnus and Bo Wahlberg (1998). On consistency of subspace methods for system identification. *Automatica* **34**(12), 1507–1519.

Jha, A. N., A. S. Saxena and V. S. Rajamani (1992). Parameter estimation algorithms for bilinear and non-linear systems using Walsh functions: Recursive approach. *International Journal of Systems Science* **23**(2), 283–290.

Johansen, Tor Arne (1994). *Operating Regime Based Process Modeling and Identification*. Ph. D. thesis, Department of Engineering Cybernetics, University of Trondheim, Trondheim, Norway. Available from Internet: `http://www.itk.ntnu.no/ansatte/Johansen_Tor.Arne/public.html` (Retrieved May 18, 2001).

Johansen, Tor Arne (1996). Identification of non-linear systems using empirical data and prior knowledge: An optimization approach. *Automatica* **32**(3), 337–356.

Johansen, Tor Arne (1997). On Tikhonov regularization, bias and variance in nonlinear system identification. *Automatica* **33**(3), 441–446.

Johansen, Tor Arne and Bjarne A. Foss (1993). Constructing NARMAX models using ARMAX models. *International Journal of Control* **58**(5), 1125–1153.

Johansen, Tor Arne and Bjarne A. Foss (1995). Identification of non-linear system structure and parameters using regime decomposition. *Automatica* **31**(2), 321–326.

Johansen, Tor Arne and Roderick Murray-Smith (1997). The operating regime approach to nonlinear modelling and control. See Murray-Smith and Johansen (1997b), Chapter 1, pp. 3–72.

Johansen, Tor Arne, Robert Shorten and Roderick Murray-Smith (2000). On the interpretation and identification of dynamic Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems* **8**(3), 297–313.

Johnson, Wayne (1980). *Helicopter theory*. Princeton, New Jersey: Princeton University Press. ISBN 0-691-07917-4.

Juditsky, Anatoli, Håkan Hjalmarsson, Albert Benveniste, Bernard Delyon, Jonas Sjöberg, Lennart Ljung and Qinghua Zhang (1995). Nonlinear black-box modeling in system identification: Mathematical foundations. *Automatica* **31**(12), 1725–1750.

Kantz, Holger and Thomas Schreiber (1999). *Nonlinear Time Series Analysis*. Cambridge, United Kingdom: Cambridge University Press. ISBN 0-521-65387-8. Paperback edition with corrections.

Karanam, V. R., P. A. Frick and Ronald R. Mohler (1978). Bilinear system identification by Walsh functions. *IEEE Transactions on Automatic Control* **23**(4), 709–713.

Katayama, Tohru and Giorgio Picci (1999). Realization of stochastic systems with exogenous inputs and subspace identification methods. *Automatica* **35**(10), 1635–1652.

Kennel, Matthew B., Reggie Brown and Henry D. I. Abarbanel (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A* **45**(6), 3403–3411.

Khatri, C. G. and C. R. Rao (1968). Solutions to some functional equations and their applications to characterization of probability distributions. *Sankhya: The Indian Journal of Statistics Series A* **30**, 167–180.

Kleiman, Evgeni G. (2000). Identification of time-varying systems: A survey. In *Preprints of the IFAC Symposium on System Identification*, Santa Barbara, California (June).

Lakshmanan, N. M. and Y. Arkun (1999). Estimation and model predictive control of nonlinear batch processes using linear parameter-varying models. *International Journal of Control* **72**(7/8), 659–675.

Larimore, Wallace E. (1983). System identification, reduced-order filtering and modeling via canonical variate analysis. In *Proceedings of the 1983 American Control Conference*, San Francisco, California (June), pp. 445–451.

Lawson, Charles L. and Richard J. Hanson (1995). *Solving Least Squares Problems* (second ed.). Philadelphia, Pennsylvania: SIAM. ISBN 0-13-822585-0.

Lee, Lawton H. (1997). *Identification and Robust Control of Linear Parameter-Varying Systems*. Ph. D. thesis, Department of Mechanical Engineering, University of California at Berkeley, Berkeley, California. Available from Internet: `http://jagger.me.berkeley.edu/~lawton/pubs.html` (Retrieved April 26, 2001).

Lee, Lawton H. and Kameshwar Poolla (1996). Identification of linear parameter-varying systems via LFTs. In *Proceedings of the 35th IEEE Conference on Decision and Control*, Kobe, Japan (December), pp. 1545–1550.

Lee, Lawton H. and Kameshwar Poolla (1999). Identification of linear parameter-varying systems using nonlinear programming. *Journal of Dynamic Systems, Measurement and Control* **121**(1), 71–78.

Leith, D. J. and W. E. Leithhead (1998). Gain-scheduled and nonlinear systems: dynamic analysis by velocity-based linearization families. *International Journal of Control* **70**(2), 289–317.

Leith, D. J. and W. E. Leithhead (1999). Analytic framework for blended multiple model systems using linear local models. *International Journal of Control* **72**(7/8), 605–619.

Leontaritis, I. J. and S. A. Billings (1985a). Input-output parametric models for non-linear systems part I: deterministic non-linear systems. *International Journal of Control* **41**(2), 303–328.

Leontaritis, I. J. and S. A. Billings (1985b). Input-output parametric models for non-linear systems part II: stochastic non-linear systems. *International Journal of Control* **41**(2), 329–344.

Leontaritis, I. J. and S. A. Billings (1987). Model selection and validation methods for non-linear systems. *International Journal of Control* **45**(1), 311–341.

Liu, C. C. and Y. P. Shih (1984). Analysis and parameter estimation of bilinear systems via Chebyshev polynomials. *Journal of the Franklin Institute* **317**(6), 373–382.

Liu, G. P. and H. Wang (1993). An improved recursive identification method for bilinear systems. *International Journal of Modelling and Simulation* **13**(2), 88–91.

Ljung, Lennart (1999). *System Identification: Theory for the User* (second ed.). Upper Saddle River, New Jersey: Prentice-Hall. ISBN 0-13-656695-2.

Ljung, Lennart and Torsten Söderström (1983). *Theory and Practice of Recursive Identification*. Cambridge, Massachusetts: MIT Press. ISBN 0-262-12095-X.

Lovera, Marco (1997). *Subspace Identification Methods: Theory and Applications*. Ph. D. thesis, Department of Electronics and Information, Politecnico di Milano, Milan, Italy.

Lovera, Marco, Michel Verhaegen and C. T. Chou (1998). State space identification of MIMO linear parameter varying models. In *Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems*, Padua, Italy (July), pp. 839–842.

Marchi, Pierre Alibert, Leandro dos Santos Coelho and Antonio Augusto Rodrigues Coelho (1999). Comparitive study of parametric and structural methodologies in identification of an experimental nonlinear process. In *Proceedings of the 1999 IEEE International Conference on Control Applications*, Kohala Coast-Island of Hawaii, Hawaii (August), pp. 1062–1067.

Mazzaro, María Cecilia, Bernardo Adrían Movsichoff and Ricardo S. Sánchez Peña (1999). Robust identification of linear parameter varying systems. In *Proceedings of the 1999 American Control Conference*, San Diego, California (June), pp. 2282–2284.

McKelvey, Tomas (1995). *Identification of State-Space Models from Time and Frequency Data*. Ph. D. thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden. ISBN 91-7871-531-8. Available from Internet: `http://www.control.isy.liu.se/` (Retrieved June 27, 2001).

McKelvey, Tomas and Anders Helmersson (1997). System identification using an over-parametrized model class: Improving the optimization algorithm. In *Proceedings of the 36th IEEE Conference on Decision and Control*, San Diego, California (December), pp. 2984–2989.

Meddeb, Souad, Jean Yves Tourneret and Francis Castanie (1998a). Identification of bilinear systems using Bayesian inference. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, Washington (May), pp. 1609–1612.

Meddeb, Souad, Jean Yves Tourneret and Francis Castanie (1998b). Identification of time-varying bilinear systems. In *Proceedings of the 9th IEEE Workshop on Statistical Signal and Array Processing*, Portland, Oregon (September), pp. 160–163.

Meddeb, Souad, Jean Yves Tourneret and Francis Castanie (2000). Unbiased parameter estimation for the identification of bilinear systems. In *Proceedings of the 10th IEEE Workshop on Statistical Signal and Array Processing*, Pocono Manor, Pennsylvania (September), pp. 176–179.

Milanese, M. and A. Vicino (1991). Optimal estimation theory for dynamic systems with set-membership uncertainty: An overview. *Automatica* **27**(6), 997–1009.

Mohler, Ronald R. (1973). *Bilinear Control Processes: With Applications to Engineering, Ecology, and Medicine*. New York: Academic Press. ISBN 0-12-504140-3.

Mohler, Ronald R. (1991). *Nonlinear Systems, Volume II: Applications to Bilinear Control*. Englewood-Cliffs, New Jersey: Prentice-Hall. ISBN 0-13-623521-2.

Mohler, Ronald R. and W. J. Kolodziej (1980). An overview of bilinear system theory and applications. *IEEE Transactions on Systems, Man and Cybernetics* **10**(10), 683–688.

Monin, André and Gérard Salut (1996). I.I.R. Volterra filtering with application to bilinear systems. *IEEE Transactions on Signal Processing* **44**(9), 2209–2221.

Moonen, Marc, Bart De Moor, Lieven Vandenberghe and Joos Vandewalle (1989). On- and off-line identification of linear state-space models. *International Journal of Control* **49**(1), 219–232.

Moré, J. J. (1978). The Levenberg-Marquardt algorithm: Implementation and theory. In G. A. Watson (Ed.), *Numerical Analysis*, Volume 630 of *Lecture Notes in Mathematics*, pp. 106–116. Berlin: Springer Verlag. ISBN 0-387-08538-6.

Murray-Smith, Roderick and Tor Arne Johansen (1997a). Local learning in local model networks. See Murray-Smith and Johansen (1997b), Chapter 7, pp. 185–210.

Murray-Smith, Roderick and Tor Arne Johansen (Eds.) (1997b). *Multiple Model Approaches to Modelling and Control*. London: Taylor and Francis. ISBN 0-7484-0595-X.

Narendra, Kumpati S., Jeyendran Balakrishnan and M. Kemal Ciliz (1995). Adaptation and learning using multiple models, switching and tuning. *IEEE Control Systems Magazine* **15**(3), 37–51.

Narendra, Kumpati S. and Kannan Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks* **1**(1), 4–27.

Narendra, Kumpati S. and Kannan Parthasarathy (1992). Neural networks and dynamical systems. *International Journal of Approximate Reasoning* **6**(2), 109–131.

Nelles, Oliver, Oliver Hecker and Rolf Isermann (1997). Automatic model selection in local linear model trees (LOLIMOT) for nonlinear system identification of a transport delay process. In *Preprints of the IFAC Symposium on System Identification*, Kitakyushu, Japan (September), pp. 727–732.

Nemani, Mahadevamurty, Rayadurgam Ravikanth and Bassam Bamieh (1995). Identification of linear parametrically varying systems. In *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, Louisiana (December), pp. 2990–2995.

Nerrand, Olivier, Pierre Roussel-Ragot, Léon Personnaz and Gérard Dreyfus (1993). Neural networks and non-linear adaptive filtering: Unifying concepts and new algorithms. *Neural Computation* **5**(2), 165–197.

Nerrand, Olivier, Pierre Roussel-Ragot, Dominique Urbani, Léon Personnaz and Gérard Dreyfus (1994). Training recurrent neural networks: Why and how? An illustration in dynamical process modeling. *IEEE Transactions on Neural Networks* **5**(2), 178–184.

Nie, Junhong (1994). A neural approach to fuzzy modeling. In *Proceedings of the 1994 American Control Conference*, Baltimore, Maryland (June), pp. 2139–2143.

Nijmeijer, H. (1982). Observability of autonomous discrete time non-linear systems: A geometric approach. *International Journal of Control* **36**(5), 867–874.

Nijmeijer, H. and A. J. van der Schaft (1996). *Nonlinear Dynamical Control Systems*. New York: Springer-Verlag. ISBN 0-387-97234-X.

Nowak, Robert D. and Barry D. Van Veen (1994). Random and pseudorandom inputs for Volterra filter identification. *IEEE Transactions on Signal Processing* **42**(8), 2124–2135.

Nowak, Robert D. and Barry D. Van Veen (1995). Invertibility of higher order moment matrices. *IEEE Transactions on Signal Processing* **43**(3), 705–708.

Oku, Hiroshi, Gerard Nijsse, Michel Verhaegen and Vincent Verdult (2001). Change detection in the dynamics with recursive subspace identification. In *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, Florida (December).

Packard, A. (1994). Gain scheduling via linear fractional transformations. *Systems and Control Letters* **22**(2), 79–93.

Paraskevopoulos, P. N., A. S. Tsirikos and K. G. Arvanitis (1994). A new orthogonal series approach to state space analysis of bilinear systems. *IEEE Transactions on Automatic Control* **39**(4), 793–797.

Peeters, R. L. M., B. Hanzon and M. Olivi (1999). Balanced realizations of discrete-time stable all-pass systems and the tangential Schur algorithm. In *Proceedings of the European Control Conference 1999*, Karlsruhe, Germany (September).

Peternell, K., W. Scherrer and M. Deistler (1996). Statistical analysis of novel subspace identification methods. *Signal Processing* **52**, 161–177.

Picci, Giorgio and Tohru Katayama (1996). Stochastic realization with exogenous inputs and 'subspace-methods' identification. *Signal Processing* **52**(2), 145–160.

Pottmann, M., Heinz Unbehauen and Dale E. Seborg (1993). Application of a general multi-model approach for identification of highly nonlinear processes: A case study. *International Journal of Control* **57**(1), 97–120.

Previdi, Fabio and Marco Lovera (1999). Identification of a class of linear models with nonlinearly varying parameters. In *Proceedings of the European Control Conference 1999*, Karlsruhe, Germany (September).

Previdi, Fabio and Marco Lovera (2001). Identification of a class of nonlinear parametrically varying models. In *Proceedings of the European Control Conference 2001*, Porto, Portugal (September).

Priestley, M. B. (1988). *Non-linear and Non-stationary Time Series Analysis*. London: Academic Press. ISBN 0-12-564910-X.

Pucar, Predrag and Jonas Sjöberg (1998). On the hinge-finding algorithm for hinging hyperplanes. *IEEE Transactions on Information Theory* **44**(3), 1310–1319.

Puskorius, Gintaras V. and Lee A. Feldkamp (1994). Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. *IEEE Transactions on Neural Networks* **5**(2), 279–297.

Ralston, Jonathon C. and Boualern Boashash (1997). Identification of bilinear systems using bandlimited regression. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany (April), pp. 3925–3928.

Rhodes, Carl (1997). *Nonlinear Modeling and Identification for Process Control*. Ph. D. thesis, Department of Chemical Engineering, California Institute of Technology, Passadena, California.

Rhodes, Carl and Manfred Morari (1995). Determining the model order of nonlinear input/output systems directly from data. In *Proceedings of the 1995 American Control Conference*, Seattle, Washington (June), pp. 2190–2195.

Rivals, Isabelle and Léon Personnaz (1996). Black-box modeling with state-space neural networks. In R. Zbikowski and Kenneth J. Hunt (Eds.), *Neural and Adaptive Control Technology*, pp. 237–264. Singapore: World Scientific. ISBN 981-02-2557-1.

Roubos, J. A., S. Mollov, Robert Babuška and Henk Verbruggen (1999). Fuzzy model-based predictive control using Takagi–Sugeno models. *International Journal of Approximate Reasoning* **22**(1–2), 3–30.

Rugh, Wilson J. (1981). *Nonlinear System Theory: The Volterra/Wiener Approach*. Baltimore, Maryland: The John Hopkins University Press. ISBN 0-8018-2549-0.

Rugh, Wilson J. (1996). *Linear System Theory* (second ed.). Upper Saddle River, New Jersey: Prentice-Hall. ISBN 0-13-441205-2.

Sauer, Tim, James A. Yorke and Martin Casdagli (1991). Embedology. *Journal of Statistical Physics* **65**(3/4), 579–617.

Schott, Kevin D. and B. Wayne Bequette (1997). Multiple model adaptive control. See Murray-Smith and Johansen (1997b), Chapter 11, pp. 269–291.

Schreiber, Thomas (1995). Efficient neighbor searching in nonlinear time series analysis. *International Journal of Bifurcation and Chaos* **5**(2), 349–358.

Schrempf, Andreas and Luigi Del Re (2001a). Identification and control of nonlinear discrete-time state space models. In *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, Florida (December).

Schrempf, Andreas and Luigi Del Re (2001b). Linear in parameters identifiability of extended bilinear systems. In *Proceedings of the European Control Conference 2001*, Porto, Portugal (September).

Setnes, Magne (2000). Supervised fuzzy clustering for rule extraction. *IEEE Transactions on Fuzzy Systems* **8**(4), 416–424.

Shamma, Jeff S. and Michael Athans (1991). Guaranteed properties of gain scheduled control of linear parameter-varying plants. *Automatica* **27**(3), 559–564.

Shamma, Jeff S. and Michael Athans (1992). Gain scheduling: Potential hazards and possible remedies. *IEEE Control Systems Magazine* **12**(3), 101–107.

Shorten, Robert and Roderick Murray-Smith (1997). Side-effects of normalising basis functions in local model networks. See Murray-Smith and Johansen (1997b), Chapter 8, pp. 211–228.

Shorten, Robert, Roderick Murray-Smith, Roger Bjørgan and Henrik Gollee (1999). On the interpretation of local models in blended multiple model structures. *International Journal of Control* **72**(7/8), 620–628.

Sjöberg, Jonas (1997). On estimation of nonlinear black-box models: How to obtain a good initialization. In *Proceedings of the 1997 IEEE Workshop Neural Networks for Signal Processing VII*, Amelia Island Plantation, Florida (September), pp. 72–81.

Sjöberg, Jonas and Lennart Ljung (1995). Overtraining, regularization and searching for a minimum, with applications to neural networks. *International Journal of Control* **62**(6), 1391–1407.

Sjöberg, Jonas, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre-Yves Glorennec, Håkan Hjalmarsson and Anatoli Juditsky (1995). Nonlinear black-box modeling in system identification: A unified overview. *Automatica* **31**(12), 1691–1724.

Skeppstedt, Anders, Lennart Ljung and Mille Millnert (1992). Construction of composite models from observed data. *International Journal of Control* **55**(1), 141–152.

Skogestad, Sigurd and Manfred Morari (1988). Understanding dynamic behavior of distillation columns. *Industrial and Engineering Chemistry Research* **27**(10), 1848–1862.

Söderström, Torsten and Petre Stoica (1983). *Instrumental Variable Methods for System Identification*. New York: Springer-Verlag. ISBN 0-387-12814-X.

Stark, J. (1999). Delay embeddings for forced systems part I: Deterministic forcing. *Journal of Nonlinear Science* **9**, 255–332.

Stark, J., D. S. Broomhead, M. E. Davies and J. Huke (1997). Takens embedding theorems for forced and stochastic systems. *Nonlinear Analysis, Theory, Methods and Applications* **30**(8), 5303–5314.

Stenman, Anders (1999). *Model on Demand: Algorithms, Analysis and Applications*. Ph. D. thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden. ISBN 91-7219-450-2. Available from Internet: `http://www.control.isy.liu.se/` (Retrieved May 29, 2001).

Sugeno, Michio and Takahiro Yasukawa (1993). A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems* **1**(1), 7–31.

Svoronos, Spyros, George Stephanopoulos and Rutherford Aris (1980). Bilinear approximation of general non-linear dynamic systems with linear inputs. *International Journal of Control* **31**(1), 109–126.

Svoronos, Spyros, George Stephanopoulos and Rutherford Aris (1981). On bilinear estimation and control. *International Journal of Control* **34**(4), 651–684.

Sznaier, Mario, Cecilia Mazzaro and Tamer Inanc (1999). An LMI approach to control oriented identification of LPV systems. In *Proceedings of the 1999 American Control Conference*, San Diego, California (June), pp. 3682–3686.

Takagi, T. and Michio Sugeno (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* **15**(1), 116–132.

Takens, Floris (1980). Detecting strange attractors in turbulence. In D. A. Rand and L. S. Young (Eds.), *Lecture Notes in Mathematics*, Volume 898, pp. 366–381. Springer Verlag.

Teves, D., G. Niesl, A. Blaas and S. Jacklin (1995). The role of active control in future rotorcraft. In *Proceedings of the 21st European Rotorcraft Forum*, Saint Petersburg, Russia.

Townsend, S., G. Lightbody, M. D. Brown and G. W. Irwin (1998). Nonlinear dynamic matrix control using local models. *Transactions of the Institute of Measurement and Control* **20**(1), 47–56.

Tsoulkas, V., Panos Koukoulas and Nicholas Kalouptsidis (1999). Identification of input output bilinear systems using cumulants. In *Proceedings of the 6th IEEE International Conference on Electronics, Circuits and Systems*, Pafos, Greece (September), pp. 1105–1108.

Van Overschee, Peter and Bart De Moor (1994). N4SID: Subspace algorithms for the identification of combined deterministic and stochastic systems. *Automatica* **30**(1), 75–93.

Van Overschee, Peter and Bart De Moor (1996). *Subspace Identification for Linear Systems; Theory, Implementation, Applications*. Dordrecht, The Netherlands: Kluwer Academic Publishers. ISBN 0-7923-9717-7.

Verdult, Vincent, Lennart Ljung and Michel Verhaegen (2001). Identification of composite local linear state-space models using a projected gradient search. Technical Report LiTH-ISY-R-2359, Department of Electrical Engineering, Linköping University, Linköping, Sweden. Accepted for publication in *International Journal of Control*.

Verdult, Vincent, Marco Lovera and Michel Verhaegen (2001). Identification of linear parameter-varying state space models for helicopter rotor dynamics. Technical report, Faculty of Applied Physics, University of Twente, Enschede, The Netherlands.

Verdult, Vincent and Michel Verhaegen (1999). Subspace-based identification of MIMO bilinear systems. In *Proceedings of the European Control Conference 1999*, Karlsruhe, Germany (September).

Verdult, Vincent and Michel Verhaegen (2000a). Bilinear state space systems for nonlinear dynamical modelling. *Theory in Biosciences* **119**(1), 1–9.

Verdult, Vincent and Michel Verhaegen (2000b). Identification of multivariable linear parameter-varying systems based on subspace techniques. In *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia (December).

Verdult, Vincent and Michel Verhaegen (2000c). Subspace identification of multivariable linear parameter-varying systems. Accepted for publication in *Automatica*.

Verdult, Vincent and Michel Verhaegen (2001a). Identification of a weighted combination of multivariable state space systems from input and output data. In *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, Florida (December).

Verdult, Vincent and Michel Verhaegen (2001b). Identification of multivariable bilinear state space systems based on subspace techniques and separable least squares optimization. *International Journal of Control* **74**(18), 1824–1836.

Verdult, Vincent and Michel Verhaegen (2001c). Identification of multivariable LPV state space systems by local gradient search. In *Proceedings of the European Control Conference 2001*, Porto, Portugal (September).

Verdult, Vincent, Michel Verhaegen and C. T. Chou (1999). Identification of MIMO bilinear state space models using separable least squares. In *Proceedings of the 1999 American Control Conference*, San Diego, California (June), pp. 838–842.

Verdult, Vincent, Michel Verhaegen, C. T. Chou and Marco Lovera (1998a). Efficient and systematic identification of MIMO bilinear state space models. In *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, Florida (December), pp. 1260–1265.

Verdult, Vincent, Michel Verhaegen, C. T. Chou and Marco Lovera (1998b). Efficient subspace-based identification of MIMO bilinear state space models. In *Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling: Theory and Applications*, Leuven, Belgium (July), pp. 216–221.

Verdult, Vincent, Michel Verhaegen and Jacquelien Scherpen (2000). Identification of nonlinear nonautonomous state space systems from input-output measurements. In *Proceedings of the 2000 IEEE International Conference on Industrial Technology*, Goa, India (January), pp. 410–414.

Verhaegen, Michel (1987). The minimal residual QR-factorization algorithm for reliably solving subset regression problems. Technical Report NASA TM 100021, NASA, Ames Research Center, Moffett Field, California.

Verhaegen, Michel (1993). Subspace model identification part 3: Analysis of the ordinary output-error state-space model identification algorithm. *International Journal of Control* **56**(3), 555–586.

Verhaegen, Michel (1994). Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica* **30**(1), 61–74.

Verhaegen, Michel and Patrick Dewilde (1992). Subspace model identification part 1: The output-error state-space model identification class of algorithms. *International Journal of Control* **56**(5), 1187–1210.

Verhaegen, Michel and Vincent Verdult (2001). *Filtering and System Identification: An Introduction*. Book in preparation.

Verhaegen, Michel and David Westwick (1996). Identifying MIMO Hammerstein systems in the context of subspace model identification methods. *International Journal of Control* **63**(2), 331–349.

Westwick, David and Michel Verhaegen (1996). Identifying MIMO Wiener systems using subspace model identification methods. *Signal Processing* **52**(2), 235–258.

Zhang, Jie and A. Julian Morris (1999). Recurrent neuro-fuzzy networks for nonlinear process modeling. *IEEE Transactions on Neural Networks* **10**(2), 313–326.

Zhao, Ming-Wang and Yong-Zai Lu (1991a). Parameter identification and convergence analysis based on the least-squares method for a class of non-linear systems part 1: SISO case. *International Journal of Systems Science* **22**(1), 33–48.

Zhao, Ming-Wang and Yong-Zai Lu (1991b). Parameter identification and convergence analysis for a class of non-linear systems with time-varying random parameters. *International Journal of Systems Science* **22**(8), 1467–1476.

Zhou, Kemin, John C. Doyle and Keith Glover (1996). *Robust and Optimal Control*. Upper Saddle River, New Jersey: Prentice-Hall. ISBN 0-13-456567-3.

Zhu, Zhiwen and Henry Leung (1999). Adaptive identification of bilinear systems. In *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona (March), pp. 1289–1292.

Author Index

Bold page numbers are used when the author is the first author of the work cited. Slanted page numbers refer to the list of references.

Subject Index

Curriculum Vitae

Vincent Verdult was born on May 25, 1974 in Bergen op Zoom, the Netherlands. He received his secondary education (VWO) at the Mollerlyceum in Bergen op Zoom. From 1993 to 1997 he studied Electrical Engineering at Delft University of Technology. He graduated cum laude. His Master's thesis dealt with the detection of ventricular late potentials in multichannel ECG signals, and was the result of a close cooperation between the Systems and Control Laboratory in Delft and the Academic Medical Center in Amsterdam.

In 1997 he started working as a Ph. D. student at the department of Electrical Engineering, Delft University of Technology. With his supervisor Michel Verhaegen, he worked on identification of nonlinear state-space systems for the process industry, on a project supported by the Dutch Technology Foundation (STW). During 1997 and 1998 he took graduate courses at the Dutch Institute of Systems and Control (DISC), and received the DISC certificate.

After two years as a Ph. D. student at Delft University, he moved to the Department of Applied Physics at the University of Twente. He continued working on the project financed by the Dutch Technology Foundation for the remaining two years, to complete his Ph. D. thesis.

In 2001 he was a visiting scientist for three months at the Automatic Control group lead by Professor Lennart Ljung at the Department of Electrical Engineering, Linköping University, Sweden.

Since September 2001, he is an assistant professor at the Faculty of Information Technology and Systems, Delft University of Technology, The Netherlands. He teaches basic courses in automatic control, and his research interests are system identification, fault-detection, and signal processing.

In his spare time he likes to listen to music and to watch movies in his little home theater, as well as to travel the world while exercising his photography skills. He lives together with Madelon, and calls Delft home.