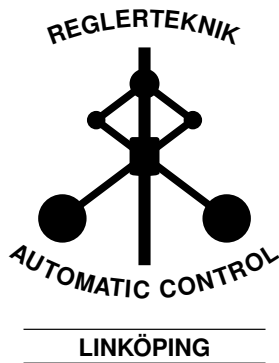


Linköping studies in science and technology. Thesis.
No. 1426

Initialization Methods for System Identification

Christian Lyzell



Division of Automatic Control
Department of Electrical Engineering
Linköping University, SE-581 83 Linköping, Sweden
<http://www.control.isy.liu.se>
lyzell@isy.liu.se

Linköping 2009

This is a Swedish Licentiate's Thesis.

Swedish postgraduate education leads to a Doctor's degree and/or a Licentiate's degree.

A Doctor's Degree comprises 240 ECTS credits (4 years of full-time studies).

A Licentiate's degree comprises 120 ECTS credits,
of which at least 60 ECTS credits constitute a Licentiate's thesis.

Linköping studies in science and technology. Thesis.

No. 1426

Initialization Methods for System Identification

Christian Lyzell

lyzell@isy.liu.se

www.control.isy.liu.se

Department of Electrical Engineering

Linköping University

SE-581 83 Linköping

Sweden

ISBN 978-91-7393-477-0

ISSN 0280-7971

LiU-TEK-LIC-2009:34

Copyright © 2009 Christian Lyzell

Printed by LiU-Tryck, Linköping, Sweden 2009

To my better two thirds Carolin and Tuva

Abstract

In the system identification community a popular framework for the problem of estimating a parametrized model structure given a sequence of input and output pairs is given by the prediction-error method. This method tries to find the parameters which maximize the prediction capability of the corresponding model via the minimization of some chosen cost function that depends on the prediction error. This optimization problem is often quite complex with several local minima and is commonly solved using a local search algorithm. Thus, it is important to find a good initial estimate for the local search algorithm. This is the main topic of this thesis.

The first problem considered is the regressor selection problem for estimating the order of dynamical systems. The general problem formulation is difficult to solve and the worst case complexity equals the complexity of the exhaustive search of all possible combinations of regressors. To circumvent this complexity, we propose a relaxation of the general formulation as an extension of the nonnegative garrote regularization method. The proposed method provides means to order the regressors via their time lag and a novel algorithmic approach for the ARX and LPV-ARX case is given.

Thereafter, the initialization of linear time-invariant polynomial models is considered. Usually, this problem is solved via some multi-step instrumental variables method. For the estimation of state-space models, which are closely related to the polynomial models via canonical forms, the state of the art estimation method is given by the subspace identification method. It turns out that this method can be easily extended to handle the estimation of polynomial models. The modifications are minor and only involve some intermediate calculations where already available tools can be used. Furthermore, with the proposed method other a priori information about the structure can be readily handled, including a certain class of linear gray-box structures. The proposed extension is not restricted to the discrete-time case and can be used to estimate continuous-time models.

The final topic in this thesis is the initialization of discrete-time systems containing polynomial nonlinearities. In the continuous-time case, the tools of differential algebra, especially Ritt's algorithm, have been used to prove that such a model structure is globally identifiable if and only if it can be written as a linear regression model. In particular, this implies that once Ritt's algorithm has been used to rewrite the nonlinear model structure into a linear regression model, the parameter estimation problem becomes trivial. Motivated by the above and the fact that most system identification problems involve sampled data, a version of Ritt's algorithm for the discrete-time case is provided. This algorithm is closely related to the continuous-time version and enables the handling of noise signals without differentiations.

Acknowledgments

First of all, I would like to thank my supervisor Professor Lennart Ljung and my co-supervisor Dr. Martin Enqvist. You have shown a great deal of patience in guiding me towards the results presented in this thesis and it would almost surely not have been the same without your support. Especially Martin, I learn a lot from our discussions and I am most grateful for your attention to details and the time you have given me.

Professor Lennart Ljung, as the head of the Division of Automatic Control, and our administrator Ulla Salaneck deserve extra gratitude for their immense work in managing our group. I feel most comfortable in your guidance down the *dark slippery slope*¹.

I would also like to thank Professor Torkel Glad and Dr. Jacob Roll for their guidance and co-authoring some of the papers, which now constitutes the foundation of this thesis. Furthermore, working with Dr. Roland Tóth, Dr. Peter Heuberger and Professor Paul Van den Hof at the Delft Center for Systems and Control is a breeze and I hope our collaboration will continue in the future.

This work has been supported by the Swedish Research Council which is gratefully acknowledged.

Writing a thesis, even if it is only for a licentiate degree, can be cumbersome and quite often frustrating. Therefore, help in different shapes from our computer gurus Dr. Gustaf Hendeby and Lic. Henrik Tidefelt are much appreciated. Your knowledge in Linux, T_EX, C++ and computer graphics is substantial and your willingness to explain is even more impressive. This thesis is written in a L^AT_EX template constructed by Gustaf and all the figures have been generated via the languages of `shapes` and `mpscatter` written by Henrik.

In a project such as this, it sometimes happens that incomprehensible formulations magically appear in the text. Luckily, I am surrounded by friends and colleagues with eyes like eagles and I am most grateful to Lic. Daniel Ankelhed, Rikard Falkeborn, Fredrik Lindsten, Lic. Henrik Ohlsson, Dr. Umut Orguner, Daniel Petersson, Lic. Henrik Tidefelt and Lic. Johanna Wallén for proofreading various parts of this thesis.

Without my friends, within and outside this group, I would have been lost. The time we spend on “tjöta”, sharing a pint or even a balcony, is what life is all about. Every time I get to beat you at Guitar Hero, I feel that I am a little bit better than you, which is quite a satisfactory feeling ☺. Seriously, you rock!

Finally, I would like to thank the loves of my life Carolin and Tuva. Please, bare with me. I can change, I promise. . .

¹We fell only once, which is a feat in itself (ERNSI, 2009).

Contents

| | | |
|----------|---------------------------------------|-----------|
| 1 | Introduction | 1 |
| 1.1 | Research Motivation | 1 |
| 1.2 | Contributions | 6 |
| 1.3 | Thesis Outline | 7 |
| 2 | Linear Regression Problems | 9 |
| 2.1 | The Least-Squares Estimator | 9 |
| 2.2 | Instrumental Variables | 10 |
| 2.3 | Regressor Selection | 11 |
| 2.4 | Selection Criteria | 15 |
| 3 | System Identification | 17 |
| 3.1 | Introduction | 17 |
| 3.2 | Model Structures | 18 |
| 3.2.1 | Linear Time-Invariant | 18 |
| 3.2.2 | Linear Parameter-Varying | 21 |
| 3.2.3 | Nonlinear Time-Invariant | 22 |
| 3.2.4 | General Theory | 22 |
| 3.3 | Instrumental Variables | 24 |
| 3.4 | Subspace Identification | 28 |
| 3.4.1 | Discrete Time | 28 |
| 3.4.2 | Continuous Time | 34 |
| 3.5 | An Algebraic Approach | 36 |
| 3.6 | Model Validation | 38 |

| | | |
|----------|-------------------------------------------------------------------|-----------|
| 4 | The Nonnegative Garrote in System Identification | 41 |
| 4.1 | Introduction | 41 |
| 4.2 | Problem Formulation | 45 |
| 4.3 | Applications to ARX Models | 46 |
| 4.4 | Applications to LPV-ARX Models | 50 |
| 4.5 | Discussion | 55 |
| 4.A | Parametric Quadratic Programming | 56 |
| 5 | Utilizing Structure Information in Subspace Identification | 59 |
| 5.1 | Introduction | 59 |
| 5.2 | OE Models | 60 |
| 5.2.1 | Discrete Time | 61 |
| 5.2.2 | Continuous Time | 67 |
| 5.3 | ARMAX Models | 69 |
| 5.4 | Special Gray-Box Models | 71 |
| 5.5 | Discussion | 75 |
| 6 | Difference Algebra and System Identification | 77 |
| 6.1 | Introduction | 77 |
| 6.2 | Algebraic Concepts | 79 |
| 6.2.1 | Signal Shifts | 79 |
| 6.2.2 | Polynomials | 80 |
| 6.2.3 | Systems of Polynomials | 84 |
| 6.3 | Identifiability | 88 |
| 6.4 | Identification Aspects | 90 |
| 6.5 | Discussion | 93 |
| | Bibliography | 95 |

Notation

Symbols, Operators and Functions

| | |
|------------------|--------------------------------------------------------------------------------------------------------------|
| \mathbb{N} | the set of natural numbers ($0 \in \mathbb{N}$) |
| \mathbb{Z} | the set of integers |
| \mathbb{Z}_+ | the set of positive integers |
| \mathbb{R} | the set of real numbers |
| \in | belongs to |
| $A \subset B$ | A is a subset of B |
| $A \setminus B$ | set difference, $\{x \mid x \in A \wedge x \notin B\}$ |
| \triangleq | equal by definition |
| \preceq | component-wise inequality (for vectors), negative semidefiniteness (for a matrix A with $A \preceq 0$) |
| $\arg \min f(x)$ | value of x that minimizes $f(x)$ |
| q | the shift operator, $qu(t) = u(t + 1)$ |
| $(x(t))_{t=0}^M$ | the sequence $x(0), x(1), \dots, x(M)$ |
| $\ x\ _1$ | $\sum_{i=1}^n x_i $ |
| $\ x\ _2$ | $(\sum_{i=1}^n x_i^2)^{1/2}$ |
| θ | parameter vector |
| $u(t)$ | input signal at time t |
| $y(t)$ | output signal at time t |

Abbreviations and Acronyms

| | |
|-------|------------------------------------------------------------|
| AIC | Akaikes Information Criterion |
| ARX | AutoRegressive (system) with eXternal input |
| ARMAX | AutoRegressive Moving Average (system) with eXternal input |
| IV | Instrumental Variables |
| FS | Forward Stepwise regression |
| LTI | Linear Time-Invariant (system) |
| LPV | Linear Parameter-Varying (system) |
| LS | Least-Squares |
| MDL | Minimum Description Length |
| NARX | Nonlinear AutoRegressive (system) with eXternal input |
| NFIR | Nonlinear Finite Impulse Response (system) |
| NNG | NonNegative Garrote |
| OE | Output Error (system) |
| OCF | Observer Canonical Form |
| PEM | Prediction Error Method |
| QP | Quadratic Programming |
| SID | Subspace IDentification |

1

Introduction

The art of modeling is an inherent nature of the human being and plays a major role throughout our lives. From the moment of our birth, empirical data of the surrounding environment are gathered, either through our own experiences or via the experience of others. The data are then used to construct models, which can help us predict the outcome of our actions in different situations. Some models protect us from danger, while others help us to plan our actions to get the desired outcome. As an example, consider the problem of traveling from one location to another, where there is a constraint on the time of arrival. Usually, there are several choices of transportation, for example, one can go by car or take a bus. When using the car, one needs to consult a map to get an idea of which roads to choose and how much time is needed to cover the distance. Here, the map represents a model of the terrain and the road network available.

It is important to differentiate the model from the *system* itself, that is, the system is the reality that the model tries to explain. In the traveling example, the system is the true terrain, while the map is a two dimensional model of the system with finite resolution.

System identification is a subset of *mathematical modeling*, which treats the subject of modeling of *dynamical systems* from empirical data. In the following section, we will try to form a basic understanding of the problems involved when trying to model a system from measured data.

1.1 Research Motivation

The fundamental concept of *system identification* can be described as follows: given some *observations* of a *system*, find a mathematical model that explains the observations as *accurately* as possible, and thus yielding a valid model of the system itself. The observations usually consist of a collection of measurements of *input* $u(t)$ and *output* $y(t)$ signals, respectively:

$$Z^N = (u(t), y(t))_{t=1}^N. \quad (1.1)$$

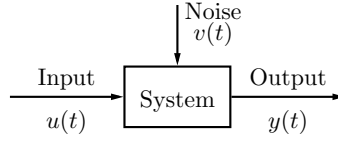


Figure 1.1: A schematic description of the signals involved in the system identification process.

In most real world processes, not all variables which affect the output of the system are measured. Such unmeasured signals $v(t)$ are referred as *disturbances* or *noise*. Figure 1.1 shows a schematic diagram of the signals involved in the identification process.

A common method in system identification is the *prediction-error method* (PEM) (see, for instance, Ljung, 1999). In this method, one tries to find the parameter vector θ which best describes the data in the sense that some cost function is minimized

$$\hat{\theta}_N = \arg \min_{\theta \in \mathbb{R}^n} V_N(\varepsilon(t, \theta), Z^N). \quad (1.2)$$

A common choice of the cost function is the quadratic cost function

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t|t-1, \theta))^2, \quad (1.3)$$

where $\hat{y}(t|t-1, \theta)$ is the predictor of the output defined by the chosen model structure. Even though it is often quite straightforward to formulate the optimization problem (1.2) of the parameter estimation problem, it turns out that it can sometimes be quite tricky to find the global optimum.

An important subproblem of system identification is to find the model with the lowest complexity, within some model set, which describes a given set of data sufficiently well according to some criterion (1.2). There are several reasons for considering this problem. One reason is that even though a higher model complexity will yield a better adaptation to the data used for the estimation, it might be that the model is adapting too well. Thus, the model does not properly represent the system itself, only the data that is used for the estimation. Also, a higher model complexity usually means a higher computational cost, both in time and in memory. Hence, a model with lower complexity might be preferred to one with higher complexity if the loss in the ability to describe the data is not too great. The model order selection problem is in several ways a difficult problem and is best described by considering a simple example:

Example 1.1

Consider the simple linear regression model

$$y(t) = \varphi(t)^T \theta + e(t), \quad (1.4)$$

where $y(t)$ and $\varphi(t)$ are known entities, $e(t)$ is an unknown disturbance, and $\theta \in \mathbb{R}^n$ is the unknown parameter vector that we want to estimate. For the special case of a white noise disturbance $e(t)$, the predictor of the output of the model structure (1.4) is given by

$$\hat{y}(t|t-1, \theta) = \varphi(t)^T \theta. \quad (1.5)$$

A common measure of the complexity for linear regression models is the number of parameters used. Thus, it is interesting to consider the problem of finding the least-squares estimate of θ using only $k \leq n$ parameters can be formulated as

$$\begin{aligned} & \underset{\theta \in \mathbb{R}^n}{\text{minimize}} && \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi(t)^T \theta)^2 \\ & \text{subject to} && \text{card}\{i \in \mathbb{N} \mid \theta_i \neq 0, 1 \leq i \leq n\} \leq k, \end{aligned} \quad (1.6)$$

where the cardinality operator card returns the number of elements in the set. In addition, to find the most appropriate model order, the problem (1.6) needs to be solved for all $1 \leq k \leq n$. The optimization problem (1.6) can be shown to be NP-hard (see, for instance, Amaldi and Kann, 1998). Instead of solving (1.6), one could try to estimate all possible combinations of the regressors and choose the combination that yields the best prediction of the data. It is easy to see that the total number of combinations is given by

$$\sum_{k=1}^n \binom{n}{k} = 2^n - 1, \quad (1.7)$$

which grows quite rapidly with the number of possible regressors n . Thus, there is a need for algorithms to select model order with lower computational complexity.

There exists a wide variety of methods to handle the difficulties described above, but many of these methods do not take the properties of dynamical systems into account. A selection of these methods will be presented in a following chapter where also the modifications in the dynamical case will be discussed.

Now, let us consider a direct application of the PEM on a special class of linear models. A common model structure is the *output error* (OE) model structure, which is defined by the input-output relationship

$$y(t) = \frac{B_p(q)}{F_p(q)} u(t) + e(t), \quad (1.8)$$

where $y(t)$ is the output, $u(t)$ the input and $e(t)$ is the measurement noise. The polynomials $B_p(q)$ and $F_p(q)$ are given by

$$B_p(q) = b_1 q^{-n_k} + \dots + b_{n_b} q^{-n_k - n_b + 1} \quad (1.9a)$$

$$F_p(q) = 1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f}, \quad (1.9b)$$

where $n_a, n_b, n_k \in \mathbb{N}$ and q is the shift operator, that is, $qu(t) = u(t+1)$. The predictor of the output defined by the OE model structure is

$$\hat{y}(t|t-1, \theta) = \frac{B_p(q)}{F_p(q)} u(t), \quad (1.10)$$

where the unknown parameters in θ consist of the coefficients to the numerator and denominator polynomials. The problem of estimating the parameters from (1.2), given a set of data (1.1), using the quadratic loss function (1.3), may turn out to be have several local minima.

Example 1.2: (Söderström, 1975)

Let the system be given by the OE model

$$y(t) = \frac{q^{-1}}{(1 - 0.7q^{-1})^2} u(t), \quad (1.11)$$

and let the input be generated by

$$u(t) = (1 - 0.7q^{-1})^2 (1 + 0.7q^{-1})^2 v(t), \quad (1.12)$$

where $v(t)$ is white Gaussian noise with zero mean and unit variance. Now, let us try to fit a model

$$\hat{y}(t|t-1, \theta) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1} + a_2 q^{-2}} u(t), \quad (1.13)$$

to the data $Z^N = (y(t), u(t))_{t=1}^N$ generated by (1.11) and (1.12). This is done by minimizing the loss function as in (1.3). It can be shown (Söderström, 1975) that the minimization with respect to b_1 can be done separately and an analytic solution as a function of the values of a_1 and a_2 can be found. Hence, without loss of generality, we need only to consider the parameter values of a_1 and a_2 when plotting the level curves of the cost function (1.3). Such a contour plot, for the case $N \rightarrow \infty$, is given in Figure 1.2.

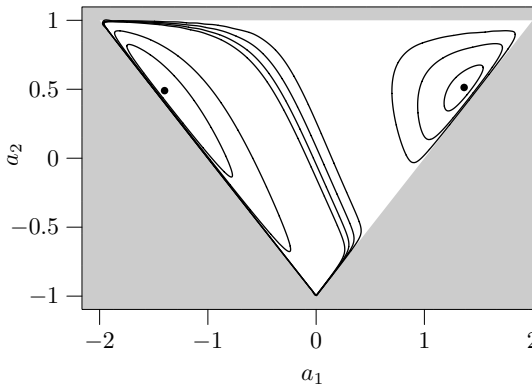


Figure 1.2: A contour plot of the cost function (1.3) for different values of the parameters a_1 and a_2 . The triangle is the region of stability.

Here we notice that there are two local minima, the global minimum in $b_1 = 1$, $a_1 = -1.4$, $a_2 = 0.49$ and a local one in $b_1 = -0.23$, $a_1 = 1.367$, $a_2 = 0.513$. Thus, it is important to find a good initial estimate to the optimization problem (1.2), if one wish to apply a local search algorithm, to be able to find the true parameter values. It is worth noting that the non-uniqueness of the minima is due to the special choice of the input signal (1.12) and is not an inherent property of the system (1.11) itself.

Even for the seemingly simple model structure in the example above, it turns out that the

PEM formulation of the parameter estimation problem can be quite difficult to solve. To this end, a lot of effort has been put into finding good initial estimates which lie close to the global optimum, from where one can start a local search. For the OE model structure, and other closely related structures, there is still ongoing research to find good initial estimates and we will return to this problem at a later stage.

Now, let us consider an even simpler model structure:

Example 1.3

Let the system that we want to model be given by the input-output relation

$$y(t) = \theta_0 u(t) + \theta_0^2 u(t-1), \quad (1.14)$$

where $\theta_0 = 0.9$ and let the input be a sum of sines $u(t) = \sin(t) + \sin(t/3)$. Plotting the cost function (1.3) with

$$\hat{y}(t|t-1, \theta) = \theta u(t) + \theta^2 u(t-1), \quad (1.15)$$

for some different values of θ yields the curve in Figure 1.3.

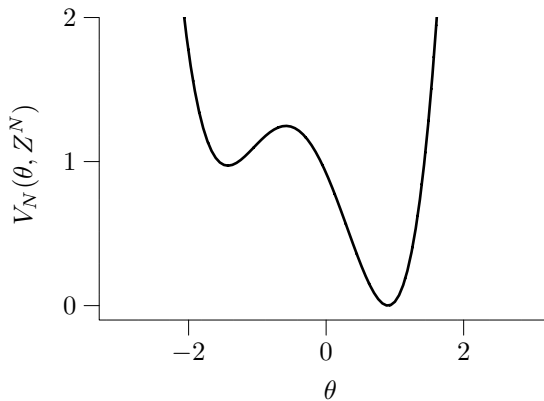


Figure 1.3: The cost function (1.3) for the system (1.14) with the predictor (1.15) for some different parameter values θ .

There are two local minima present, the global minimum in $\theta = 0.9$ and a local in $\theta = -1.4279$. Thus, also for this simple example, it is important to have a good initial estimate to find the true global minimum.

The difficulties that were found in the example above are due to the square on the unknown parameter. Fortunately, the model structure discussed above has a favorable structure, that is, only polynomial nonlinearities are present. This enables the use of certain recently developed algebraic methods, which can be used to rewrite (1.14) to an equivalent form where all the nonlinearities are moved from the unknown parameter to the signals involved. Thus, instead of minimizing a fourth order polynomial, one only needs to find the unique minimum of a second order polynomial. The mentioned algebraic techniques will be discussed later in the thesis.

1.2 Contributions

The main contributions of this thesis are new methods to approach the problems discussed in the section above. Recent developments in the statistical learning community have opened up for new efficient algorithms for the model selection problem. These algorithms also have interesting implications for the model order selection of linear dynamic regression models:

C. Lyzell, J. Roll, and L. Ljung. The use of nonnegative garrote for order selection of ARX models. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008

This method can be modified to also include the structural selection problem, that is, the problem of choosing between several possible grouped regressors. These modifications have been developed to include a slightly more general model class in:

R. Tóth, C. Lyzell, M. Enqvist, P. Heuberger, and P. Van den Hof. Order and structural dependence selection of LPV-ARX models using a nonnegative garrote approach. In *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009

The initialization of linear polynomial model structures is a mature topic in system identification and several methods exist. In the last two decades, a new algorithm for estimation of linear state-space models has been developed. By utilizing the structure of certain canonical state-space forms, this method can be altered to work also as an initialization method for certain linear polynomial models, such as the OE model in Example 1.2, which is the topic of:

C. Lyzell, M. Enqvist, and L. Ljung. Handling certain structure information in subspace identification. In *Preprints of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009a

The tools of differential algebra have shown to be quite useful when analyzing different system identification aspects of certain continuous-time model structures. It turns out that a model structure containing only polynomial nonlinearities is globally identifiable if and only if it can be written as a linear regression model. If these methods were available for discrete-time equivalent model structures, this would imply that the parameter estimation problem in Example 1.3 could be rewritten as an optimization problem with only one minimum. It turns out that some of the results can be generalized, with slight alterations, to handle discrete-time model structures:

C. Lyzell, T. Glad, M. Enqvist, and L. Ljung. Identification aspects of Ritt's algorithm for discrete-time systems. In *Preprints of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009b

Not included published material is:

C. Lyzell and G. Hovland. Verification of the dynamics of the 5-DOF Gantry-Tau parallel kinematic machine. In *Proceedings of Robotics and Applications and Telematics*, Würzburg, Germany, August 2007

1.3 Thesis Outline

This thesis is structured as follows: In Chapter 2, the fundamental linear regression model is reviewed, which includes the regression selection problems discussed in Example 1.1. Chapter 3 concerns the system identification framework and different model structures and estimation methods are reviewed. In Chapter 4 a novel algorithmic contribution to the model selection and the structural selection problem in the dynamical case is described. The initialization of different linear polynomial model structures is analyzed in Chapter 5, while Chapter 6 presents an algebraic framework for analyzing system identification aspects of discrete-time model structures with polynomial nonlinearities.

2

Linear Regression Problems

The concept of *linear regression* goes back to the work of Gauss (1809) on the motion of heavenly bodies and is still today a widely used tool in the field of applied science. The linear regression model is the simplest type of a *parametric model* and is described by the relationship

$$y(t) = \varphi^T(t)\theta + v(t). \quad (2.1)$$

Here, $y(t)$ is a measurable quantity called the *observations* or the *output*, $\varphi(t)$ is a known vector of *independent variables* or *regressors* and θ is a vector of unknown parameters. The signal $v(t)$ is called *disturbance* or *noise* and represents the *error* that might occur when measuring the regressed variable $y(t)$. In this chapter, we will review different methods for estimating the parameter vector θ in (2.1) and also the problem of determining which regressors are important for describing the data. For further details on the subjects at hand, see, for instance, Draper and Smith (1981), Casella and Berger (2002) or Hastie et al. (2009) and the references therein.

2.1 The Least-Squares Estimator

How does one find a reliable estimate of the parameter vector θ in (2.1) given a sequence of observations $y(t)$ and regressors $\varphi(t)$ where the variable t ranges from 1 to N ? Assuming that the noise is a white Gaussian process with zero mean, which is uncorrelated with the regressors, it can be shown that the *least-squares* (LS) estimator

$$\hat{\theta}_N^{\text{LS}} = \arg \min_{\theta \in \mathbb{R}^n} \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t)\theta)^2, \quad (2.2)$$

is a *statistically efficient* estimator (see, for example, Casella and Berger, 2002). The difference $y(t) - \varphi^T(t)\theta$ is known as the *residual* and represents the remaining unmodeled

behavior of the data. By introducing the notation

$$R_N \triangleq \frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t), \quad (2.3a)$$

$$f_N \triangleq \frac{1}{N} \sum_{t=1}^N \varphi(t) y(t), \quad (2.3b)$$

the solution to the unconstrained quadratic optimization problem (2.2) can be written as

$$\hat{\theta}_N^{\text{LS}} = R_N^{-1} f_N, \quad (2.4)$$

given that the matrix R_N has full rank. Determining the estimate $\hat{\theta}_N^{\text{LS}}$ directly via the formula (2.4) is not numerically sound and (2.2) should in practice be solved using a more efficient algorithm (see, for instance, Golub and Van Loan, 1996).

In some applications, one might have prior knowledge of the values of certain parameters or there might exist a linear relationship between some of them, that is, the parameters satisfy some linear equality constraints $A\theta = b$. Adding these constraints to the optimization problem (2.2) results in an *equality constrained least-squares* (LSE) problem

$$\begin{aligned} & \underset{\theta \in \mathbb{R}^{n_\theta}}{\text{minimize}} && \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t)\theta)^2, \\ & \text{subject to} && A\theta = b. \end{aligned} \quad (2.5)$$

A simple solution to this class of optimization problems is to eliminate the linear equality constraints by an appropriate change of coordinates. If A has full rank, a possible coordinate change can be found by determining the QR decomposition with column pivoting

$$A^T \Pi = (Q_1 \quad Q_2) \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad (2.6a)$$

and make the change of variables

$$\theta = Q_1 R^{-T} \Pi^T b + Q_2 \tilde{\theta}. \quad (2.6b)$$

This coordinate change eliminates the equality constraints and only an ordinary least-squares problem in the variable $\tilde{\theta}$ remains to be solved. See, for example, Nocedal and Wright (2006) for further details on the handling of linear equality constraints.

2.2 Instrumental Variables

Let the system have the linear regression structure

$$y(t) = \varphi^T(t)\theta_0 + v(t). \quad (2.7)$$

The least-squares estimator (2.4), using the notation in (2.3), becomes

$$\hat{\theta}_N^{\text{LS}} = R_N^{-1} f_N = \theta_0 + R_N^{-1} \left(\frac{1}{N} \sum_{t=1}^N \varphi(t) v(t) \right).$$

Now, if the regressors and the noise are independent and R_N^{-1} is a finite matrix, then it holds that $E \hat{\theta}_N^{\text{LS}} = \theta_0$, that is, the least-squares estimator will be *unbiased*.

When the regressors and the noise are correlated, it might happen that the least-squares estimator is *biased* and therefore needs to be modified. To this end, the *instrumental variables* (IV) method was proposed in Reiersøl (1941). A thorough treatment of the method is given in Söderström and Stoica (1980) and the fundamentals can be found in, for instance, Ljung (1999) and Söderström and Stoica (1989). The basic idea of the IV method is to find variables $\zeta(t)$, which are uncorrelated with the noise $v(t)$, so that the estimator

$$\hat{\theta}_N^{\text{IV}} = \left(\frac{1}{N} \sum_{t=1}^N \zeta(t) \varphi^T(t) \right)^{-1} \left(\frac{1}{N} \sum_{t=1}^N \zeta(t) y(t) \right), \quad (2.8)$$

is asymptotically unbiased. The variables $\zeta(t)$ are often referred to as *instruments*. The *variance optimal* instruments depend on the true system (see, for example, Söderström and Stoica, 1980), which is generally not known beforehand. Thus, it may be quite difficult to choose appropriate instruments and different methods for approximating the optimal instruments have been designed (see, for example, the IV4 method in Söderström and Stoica, 1980). It is clear that the least-squares estimator is a special case of IV when choosing the instruments as $\zeta(t) = \varphi(t)$.

2.3 Regressor Selection

An important problem when dealing with linear regression models is the selection of regressors, that is, finding the regressors that are significant for describing the data. There exists a wide variety of methods for regressor selection, some of which are described in Hastie et al. (2009). Recent developments can be found in Hesterberg et al. (2008).

A brute force way of selecting the most significant regressors is to estimate all possible combinations and then select the best one according to some criterion. The selection criterion should take both the fit to data and the number of parameters into account. This kind of method is often referred to as *all possible regressors* and efficient implementations can be found in, for example, Furnival and Wilson (1974) or in Niu et al. (1996). A huge disadvantage of the above strategy is that the computational complexity grows quite rapidly with the number of regressors and it is therefore not applicable in cases with a large number of possible regressors, as is often the case in biological and medical applications. If a large number of regressors are available, a more computationally tractable solution to the regressor selection problem is to use *regularization* methods.

One of the first regularization methods proposed in the statistical community was presented in Hoerl and Kennard (1970) and is referred to as *ridge regression*

$$\hat{\theta}_N^{\text{ridge}} \triangleq \arg \min_{\theta \in \mathbb{R}^{n_\theta}} \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t)\theta)^2 + \frac{\lambda}{N} \|\theta\|_2^2. \quad (2.9)$$

The nonnegative variable λ is called the *regularization parameter*, which balances the need for a good fit to data and the size of the estimated parameter values. For λ equal to zero, only the fit to data will be considered and the solution equals the least-squares estimate (2.4). As λ increases, the focus shifts from having a good fit to data to having small parameter values and in the limit all the parameter values will be zero.

In Tibshirani (1996), a slightly different regularization method was proposed, referred to as *lasso regression*

$$\hat{\theta}_N^{\text{lasso}} \triangleq \arg \min_{\theta \in \mathbb{R}^{n_\theta}} \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t)\theta)^2 + \frac{\lambda}{N} \|\theta\|_1. \quad (2.10)$$

Even though (2.9) and (2.10) might appear similar, their solutions have quite different properties. To illustrate this difference, rewrite (2.9) and (2.10) as

$$\begin{aligned} &\text{minimize}_{\theta \in \mathbb{R}^{n_\theta}} \quad \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t)\theta)^2 \\ &\text{subject to} \quad \|\theta\|_p^p \leq s, \end{aligned} \quad (2.11)$$

where $s \geq 0$ and $p \in \{1, 2\}$, respectively, where there is a one-to-one correspondence between the variables s and λ . In Figure 2.1, the feasible set and the cost function in (2.11) are illustrated for a problem with two parameters.

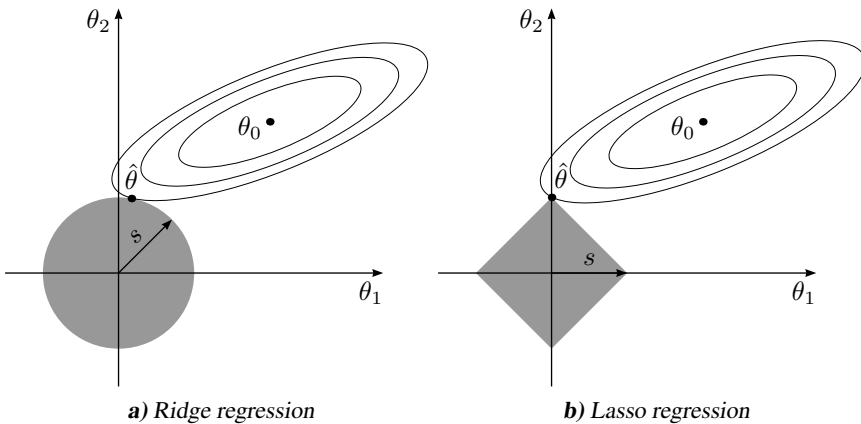


Figure 2.1: The feasible set (shaded area) and the level curves of (2.11) for ridge (to the left) and lasso (to the right) regression. The true parameter values are denoted by θ_0 and the estimate by $\hat{\theta}$.

In the ridge regression case, the spherical shape of the feasible set allows both parameters to be nonzero at the optimum. This is a drawback of the method in the sense that even though the parameter values will shrink towards zero as the parameter s decreases, most of them will be nonzero until s is very small. Thus, it may be difficult to decide which regressors are more important than the others (see, for instance, Hastie et al., 2009).

In the lasso regression case, the diamond shape of the feasible set forces the less important parameter to be exactly zero. The lasso method generally releases one parameter at a time as s increases until all parameters are nonzero, which makes the regression selection problem easier. The formulation (2.11) with $p = 1$ can easily be rewritten as a convex quadratic optimization problem by introducing slack variables for the ℓ_1 norm constraint (Tibshirani, 1996). This makes it a simple problem to solve for any given λ , but it is not clear how to solve the problem efficiently for all nonnegative λ .

An important contribution to the regularized regression problem was presented in the seminal paper by Efron et al. (2004), where a new way of describing how the regressors should be selected was presented. The algorithm is called *least angle regression* (LARS) and is easiest explained via the *forward stepwise* (FS) regression method. The FS method is quite simple and builds a model sequentially by adding one parameter at a time. At each time step, it identifies the regressor that is most correlated with the unmodeled response (the residual) and then updates the least-squares parameter estimate accordingly.

The LARS method uses a similar strategy. At each time step, LARS locates the regressor that is most correlated with the residual. But, instead of fitting the parameter completely in the least-squares sense, it only fits the parameter as much that another unused regressor gets an equal correlation with the residual. This process is repeated, adding one regressor at a time, until the full linear regression model is retrieved. To illustrate the LARS procedure, let us consider a simple example.

Example 2.1

Let us consider a linear regression model (2.1) involving two parameters

$$y(t) = \varphi_1(t)\theta_1 + \varphi_2(t)\theta_2, \quad (2.12)$$

where we for simplicity assumed noise-free output. Introducing the vector of stacked outputs

$$\mathbf{y} \triangleq (y(1) \quad y(2) \quad \cdots \quad y(N))^T,$$

and similarly for φ_1 and φ_2 , then (2.12) may be written in vector form as

$$\mathbf{y} = \varphi_1\theta_1 + \varphi_2\theta_2. \quad (2.13)$$

In the following, assume (without loss of generality) that the vectors \mathbf{y} , φ_1 and φ_2 have zero mean. Furthermore, assume the regressor vectors φ_1 and φ_2 have been scaled to have unit norm and that φ_1 has a higher correlation with \mathbf{y} than φ_2 does. Here, by a *predictor* we mean

$$\hat{\mathbf{y}} = \varphi_1\hat{\theta}_1 + \varphi_2\hat{\theta}_2,$$

for some estimates $\hat{\theta}_1$ and $\hat{\theta}_2$ of the parameters θ_1 and θ_2 , respectively. The least-squares predictor of (2.13) using both regressors is illustrated in Figure 2.2 by $\hat{\mathbf{y}}_2$. The best least-squares predictor when only one regressor may be used, in this case φ_1 , is given by the predictor $\hat{\mathbf{y}}_1$ in Figure 2.2.

The FS regression method starts with both parameters equal to zero, that is, in the predictor denoted $\hat{\mathbf{y}}_0$ in Figure 2.2. Since φ_1 has the highest correlation with \mathbf{y} of the two available regressors, the FS method first adds φ_1 to the model which results in the predictor $\hat{\mathbf{y}}_1$. Then, φ_2 is added and the estimate is updated resulting in $\hat{\mathbf{y}}_2$. The FS solution path is illustrated by the solid line in Figure 2.2.

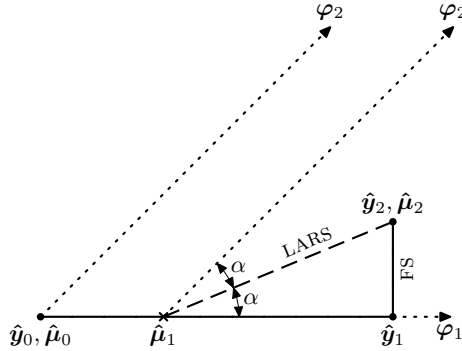


Figure 2.2: The solution path of the FS method (solid line) and the LARS procedure (dashed line) for the simple two regressors model (2.12).

The LARS has a similar solution path as the one given by the FS method. It starts with both parameters equal to zero, that is, the point denoted $\hat{\mu}_0$ in Figure 2.2. In the same way as for the FS method, the regressor φ_1 is added to the model. Instead of fitting this estimate completely, LARS changes the value of the corresponding parameter so that the predictor moves continuously towards the least-squares predictor \hat{y}_1 until the regressor φ_2 has as high correlation with the residual $\mathbf{y} - \varphi_1 \hat{\theta}_1$ as φ_1 does. This happens when the predictor denoted $\hat{\mu}_1$ in Figure 2.2 is reached. Finally, the regressor φ_2 is added to the model and the parameter values changes continuously until the least-squares predictor $\hat{\mu}_2 = \hat{y}_2$ is found. The solution path of the LARS algorithm is depicted as the dashed line in Figure 2.2. For further examples and illustrations of the LARS algorithm, the reader is referred to Efron et al. (2004) and Hastie et al. (2009).

By the description of the LARS algorithm and the example above, one notices that the corresponding solution path is piecewise affine, that is, the resulting estimate $\hat{\theta}_N^{\text{LARS}}$ is a piecewise affine function, see also Figure 2.2. One of the main results in Efron et al. (2004) showed that the lasso regression can be solved via the LARS algorithm if the step length is restricted (see also Hastie et al., 2009). In addition to providing an efficient algorithm for finding the entire solution path of the lasso method, this implies that the solution to the lasso regularization problem (2.10) is piecewise affine in the regularization parameter. In other words, the solution path $\hat{\theta}_N^{\text{lasso}}(\lambda)$ is a piecewise affine function of the parameter λ .

Another interesting regularization method is the *nonnegative garrote* (NNG), which was presented in Breiman (1995)

$$\begin{aligned} & \underset{w \in \mathbb{R}^{n_\theta}}{\text{minimize}} && \frac{1}{N} \sum_{t=1}^N \left(y(t) - (\varphi(t) \odot \hat{\theta}_N^{\text{LS}})^T w \right)^2 + \frac{\lambda}{N} \sum_{i=1}^{n_\theta} w_i \\ & \text{subject to} && w \succeq 0, \end{aligned} \quad (2.14)$$

where \odot denotes componentwise multiplication and \succeq denotes componentwise inequality. This method differs from the previous methods in that it uses the least-squares esti-

mate (2.4) as a starting value, and then finds the *importance weights* w of the parameters instead of manipulating the parameter values directly. In Yuan and Lin (2006) it was shown that also the NNG regularization problem (2.14) has a piecewise affine solution path and an efficient algorithm similar to the LARS method was presented.

The algorithm used in Efron et al. (2004) can be generalized to a larger class of regularization problems and the optimization algorithms used are referred to as *parametric optimization*, see, for example, Rosset and Zhu (2007) or Roll (2008).

The regularization methods presented above yield a collection of parameter estimates along the solution path and somehow one must decide which of these estimates to use. This is the topic of the next section.

2.4 Selection Criteria

In the estimation and selection methods discussed so far, the cost function

$$V_N(\theta) \triangleq \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t)\theta)^2 \quad (2.15)$$

has been used to determine estimates of the parameters. One drawback of this cost function is that it only takes the fit to data into consideration when selecting the model and thus rewards higher order structures with more degrees of freedom. Thus, a different kind of cost function is needed, which also considers the number of parameters used to achieve the fit to prevent over-fitting.

To this end, several criteria have been proposed in the literature, some of which can be found in, for instance, Ljung (1999) and Hastie et al. (2009). In this thesis, two different selection criteria will be used. The first one is an approximation of the *Akaike information criteria* (AIC) (see Akaike, 1969) which has the form

$$W_N^{\text{AIC}}(\theta) \triangleq V_N(\theta) \left(1 + \frac{2 \dim \theta}{N} \right). \quad (2.16)$$

where the \dim operator yields the number of nonzero elements of a vector. The AIC criteria has a tendency to select higher order models and a different criterion called the *minimum description length* (MDL) (see Rissanen, 1978) can be used instead

$$W_N^{\text{MDL}}(\theta) \triangleq V_N(\theta) \left(1 + \frac{\dim \theta \log N}{N} \right), \quad (2.17)$$

The primary use of these two criteria in this thesis is to automatize the selection of which parameter estimate to use out of the entire solution path given by the regularization methods presented in the previous section. As an example, consider the piecewise affine solution path $\hat{\theta}_N^{\text{asso}}(\lambda)$ to the regularization problem (2.10). The MDL choice is the given by minimizing $W_N^{\text{MDL}}(\hat{\theta}_N^{\text{asso}}(\lambda))$ for all $\lambda \geq 0$. Due to the piecewise affine solution path of the lasso method, it suffices to find the minimum of $W_N^{\text{MDL}}(\hat{\theta}_N^{\text{asso}}(\lambda))$ for a finite number of points, namely the values of λ for which the local affine representation of the solution path changes.

3

System Identification

The subject of system identification, which already has been touched upon in the introductory chapters, is the art of modeling dynamical systems given a collection of measurements of the input and output signals of the system. Here, a more thorough treatment of the concepts needed in later chapters will be given. The topics presented here only form a minor subset of the vast subject that is system identification. To be able to grasp the diversity of the topic and to get a more complete picture of its essence one should explore the contents of the de facto standard textbooks Ljung (1999) or Söderström and Stoica (1989).

3.1 Introduction

System identification is the art of modeling dynamical systems from a given set of data

$$Z^N = (u(t), y(t))_{t=1}^N, \quad (3.1)$$

where u and y are the input and the output signals, respectively, of the system that should be modeled, see also Figure 1.1. A common assumption is that the system can be well approximated using some parametrized model structure which may represent the prior knowledge of the system. The choice of parametrization is highly dependent on the application at hand, where experience and system knowledge are vital to decide an appropriate parametrization. Once the model structure has been decided upon, one can adapt a model to the measurement by minimizing some criterion

$$\hat{\theta}_N = \arg \min_{\theta \in \mathcal{D}} V_N(\theta, Z^N), \quad (3.2)$$

where the unknown parameter vector θ represents the parametrization of the model structure. Thus, there are mainly two choices that need to be made: an appropriate parametrization of the model and a cost function which reflects the application in which the model

will be used. In automatic control applications, the model is often used to predict the output of the system given the current state and input signal. This is represented by choosing a cost function on the form

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^N \ell(L(q)\varepsilon(t, \theta)), \quad (3.3)$$

where $\ell(\cdot)$ is some convex function and $L(q)$ is a filter which removes unwanted properties in the data. The quantity

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|t-1, \theta), \quad (3.4)$$

is called the *prediction error* and $\hat{y}(t|t-1, \theta)$ is the one-step-ahead *predictor* representing the model of the system. The method of finding the predictor which best describes a set of data (3.1) by minimizing the criterion (3.3) is commonly referred to as the *prediction-error method* (PEM). The cost function (3.3) is quite general and it is not uncommon to choose a less complex criterion. The choice in this thesis is the quadratic cost function (1.3), that is,

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^N \varepsilon(t, \theta)^2. \quad (3.5)$$

This choice is probably the most widely used special case of the general cost function (3.3). The main motivation for its use is the simplicity, both in terms of deriving theoretical properties of the resulting estimator and in terms of implementing simple/practical algorithms for adapting models to data. For a thorough description of the PEM and its properties, see Ljung (1999) and the references therein.

In the sections that follow, some examples of common choices of model parametrization will be given. Furthermore, as was pointed out in the introductory chapter, since the PEM often involves solving a non-convex optimization problem, some methods for finding an initial estimate which lies close to the optimal will be presented. Finally, the chapter is concluded with a collection of methods for validating estimated models.

3.2 Model Structures

In this section, some of the commonly used model parametrizations will be presented. Each choice of parametrization incorporates some knowledge about the system, for example, where the noise enters the system. Each assumption inflicts some structure to the estimation problem (3.3), which should be utilized to increase the efficiency of the optimization routine. The section is concluded with a mathematical formalization of the concepts introduced.

3.2.1 Linear Time-Invariant

A widely used assumption in system identification is that the system at hand is *linear time-invariant* (LTI):

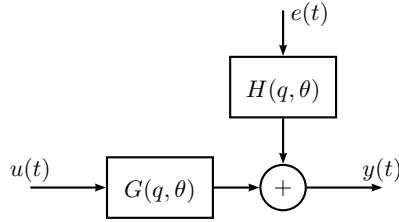


Figure 3.1: A common representation of LTI systems.

Definition 3.1. A system is *linear* if its output response of a linear combination of inputs is the same linear combination of the output responses of the individual inputs. It is said to be *time invariant* if its response to a certain input signal does not depend on absolute time.

A common representation of LTI systems is given by the class of linear *transfer-function* models

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t), \quad (3.6)$$

where q is the forward shift operator, that is, $qu(t) = u(t + 1)$. Here, $y(t)$ is an n_y dimensional vector of outputs, $u(t)$ is an n_u dimensional vector of inputs, and $e(t)$ is the noise signal with appropriate dimensions, respectively. Furthermore, the transfer-functions $G(q, \theta)$ and $H(q, \theta)$ are rational functions in q and the coefficients are given by the elements of the parameter vector θ . In this presentation, we will assume that $n_y = n_u = 1$. The predictor associated with the output of (3.6) is given by (see, for example, Ljung, 1999, page 80)

$$\hat{y}(t|t-1, \theta) \triangleq H^{-1}(q, \theta)G(q, \theta)u(t) + (1 - H^{-1}(q, \theta))y(t), \quad (3.7)$$

where $H^{-1}(q, \theta) \triangleq 1/H(q, \theta)$. The model structure (3.6) is quite general and some special cases deserve attention. A simple case of (3.6) is the ARX model structure, that is,

$$A_p(q)y(t) = B_p(q)u(t) + e(t), \quad (3.8)$$

where

$$A_p(q) = 1 + a_1q^{-1} + \cdots + a_{n_a}q^{-n_a}, \quad (3.9a)$$

$$B_p(q) = b_1q^{-n_k} + \cdots + b_{n_b}q^{-n_k-n_b+1}. \quad (3.9b)$$

In the case when no input is present, that is $B_p(q) = 0$ in (3.8), the model structure is referred to as an AR model. The ARX model structure (3.8) can be generalized by assuming that the noise is described by a *moving average* process, which results in the ARMAX model structure

$$A_p(q)y(t) = B_p(q)u(t) + C_p(q)e(t), \quad (3.10)$$

where

$$C_p(q) = 1 + c_1q^{-1} + \cdots + c_{n_c}q^{-n_c}.$$

An important special case of (3.6) is the *output error* (OE) model structure given by

$$y(t) = \frac{B_p(q)}{F_p(q)}u(t) + e(t). \quad (3.11)$$

The predictors of the output given by the model structures (3.8)-(3.11) can be derived using (3.7) with straightforward manipulations.

Example 3.1: (Estimation of ARX models)

Let us consider finding the PEM estimate (3.3) with the cost function (3.5) for the ARX model structure (3.8) where we, without loss of generality, assume that $n_k = 1$. Since $G(q, \theta) = B_p(q)/A_p(q)$ and $H(q, \theta) = 1/A_p(q)$, the predictor (3.7) of the output can be written as

$$\hat{y}(t|t-1, \theta) = B_p(q)u(t) + (1 - A_p(q))y(t) = \varphi(t)^T \theta, \quad (3.12)$$

where

$$\varphi(t) \triangleq (-y(t-1) \quad \cdots \quad -y(t-n_a) \quad u(t-1) \quad \cdots \quad u(t-n_b))^T, \quad (3.13)$$

$$\theta \triangleq (a_1 \quad \cdots \quad a_{n_a} \quad b_1 \quad \cdots \quad b_{n_b})^T. \quad (3.14)$$

Thus, for the ARX model structure (3.8) with the quadratic cost function (3.5), the PEM estimate coincides with the LS estimate (2.2) and the simple methods presented in Section 2.1 can be used.

The development in the automatic control community during the 1960s led to that a different representation of LTI models became popular, namely the *state-space model*

$$x(t+1) = A(\theta)x(t) + B(\theta)u(t) + w(t), \quad (3.15a)$$

$$y(t) = C(\theta)x(t) + D(\theta)u(t) + v(t). \quad (3.15b)$$

Here, $x(t)$ is the auxiliary *state* vector of dimension n_x and the white signals $w(t)$ and $v(t)$ represent the *process* and *measurement* noise, respectively. The best linear predictor associated with the output of (3.15) is given by (see, for example, Ljung, 1999, page 98)

$$\begin{aligned} \hat{y}(t|t-1, \theta) &= C(\theta)(qI - A(\theta) + K(\theta)C(\theta))^{-1} \\ &\quad \times \left((B(\theta) - K(\theta)D(\theta))u(t) + K(\theta)y(t) \right), \end{aligned} \quad (3.16)$$

where I denotes the identity matrix of appropriate dimensions and $K(\theta)$ is the Kalman gain. Using the residual $\varepsilon(t, \theta) = y(t) - \hat{y}(t|t-1, \theta)$, one may rewrite the predictor (3.16) on state-space form as

$$\hat{x}(t+1|t, \theta) = A(\theta)\hat{x}(t|t, \theta) + B(\theta)u(t) + K(\theta)\varepsilon(t, \theta), \quad (3.17a)$$

$$y(t) = C(\theta)\hat{x}(t|t, \theta) + D(\theta)u(t) + \varepsilon(t, \theta). \quad (3.17b)$$

Now, assume that the system agrees with (3.15) for some $\theta = \theta_0$. Then the corresponding residual $\varepsilon(t, \theta_0)$ is a white noise sequence with zero mean coinciding with the *innovation process* of the system and (3.17) may be represented by

$$\hat{x}(t+1|t, \theta) = A(\theta)\hat{x}(t|t, \theta) + B(\theta)u(t) + K(\theta)e(t), \quad (3.18a)$$

$$y(t) = C(\theta)\hat{x}(t|t, \theta) + D(\theta)u(t) + e(t), \quad (3.18b)$$

where $e(t)$ is the innovation process. The representation (3.18) is referred to as the *innovations form* of (3.15). There exist several canonical state-space representations of the linear transfer-function structures given by (3.6). An example of a representation, which will be utilized in a later chapter, is the *observer canonical form* (OCF), which for the ARMAX case (3.10) with $n_k = 1$ can be written as

$$x(t+1) = \begin{pmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n-1} & 0 & 0 & \dots & 1 \\ -a_n & 0 & 0 & \dots & 0 \end{pmatrix} x(t) + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix} u(t) + \begin{pmatrix} c_1 - a_1 \\ c_2 - a_2 \\ \vdots \\ c_{n-1} - a_{n-1} \\ c_n - a_n \end{pmatrix} e(t), \quad (3.19a)$$

$$y(t) = (1 \quad 0 \quad 0 \quad \dots \quad 0) x(t) + e(t). \quad (3.19b)$$

The unknown parameters in the matrices (3.19) correspond directly to the coefficients of the polynomials (3.10). For the OE model structure (3.11), there is no process noise, and thus $K(\theta) = 0$ in (3.18). For a complete description of the representation of LTI models and the estimation of such, see, for instance, Ljung (1999) or Söderström and Stoica (1989).

3.2.2 Linear Parameter-Varying

The model structures presented so far have all belonged to the class of LTI models, that is, the coefficients of the polynomials in (3.6) and the matrices (3.15) have all been static. A direct generalization of these model structures is obtained if one allows the polynomials and matrices to depend on a known time-varying parameter $p : \mathbb{R} \rightarrow \mathbb{P}$, where $\mathbb{P} \subset \mathbb{R}$ is the *scheduling space*. These model structures are called *linear parameter-varying* (LPV).

The LPV-ARX model structure is defined in the SISO case as

$$\mathcal{A}(p, q)y(t) = \mathcal{B}(p, q)u(t) + e(t), \quad (3.20)$$

where

$$\begin{aligned} \mathcal{A}(p, q) &= 1 + a_1(p)q^{-1} + \dots + a_{n_a}(p)q^{-n_a}, \\ \mathcal{B}(p, q) &= b_1(p)q^{-n_k} + \dots + b_{n_b}(p)q^{-n_k - n_b + 1}, \end{aligned}$$

and the coefficient functions $a_i, b_j : \mathbb{P} \rightarrow \mathbb{R}$ have a static dependence on the measured variable p . Introduce

$$(\phi_1(p) \quad \dots \quad \phi_{n_g}(p)) \triangleq (a_1(p) \quad \dots \quad a_{n_a}(p) \quad b_1(p) \quad \dots \quad b_{n_b}(p)),$$

with $n_g \triangleq n_a + n_b$. A common assumption is that each of the functions ϕ_i is linearly parametrized as

$$\phi_i(p) = \theta_{i0} + \sum_{j=1}^{s_i} \theta_{ij} \psi_{ij}(p),$$

where $(\theta_{ij})_{i=1, j=0}^{n_g, s_i}$ are unknown parameters and $(\psi_{ij})_{i=1, j=1}^{n_g, s_i}$ are basis functions chosen by the user. In this case, straightforward definitions of the parameter vector and the regression vector allows us to rewrite (3.20) as a linear regression (2.1). Thus, the LPV-ARX model structure can be estimated from data

$$Z^N = (u(t), p(t), y(t))_{t=1}^N, \quad (3.21)$$

using the simple tools presented in Section 2.1.

The LPV state-space model (LPV-SS) form used in this thesis is given by

$$x(t+1) = A(p, \theta)x(t) + B(p, \theta)u(t) + K(p, \theta)e(t), \quad (3.22a)$$

$$y(t) = C(p, \theta)x(t), \quad (3.22b)$$

where the matrices depend on the scheduling variable p . The relationship between the LPV-ARX (3.20) and the LPV-SS (3.22) models is quite involved and details are given in Tóth (2008), where also an overview of different methods for the estimation of LPV models is given.

3.2.3 Nonlinear Time-Invariant

In later chapters, the initialization of nonlinear model structures will be considered. A simple example of such a structure is the nonlinear ARX model structure (NARX) given by the relation

$$y(t) = g(\varphi(t), \theta) + e(t), \quad (3.23)$$

where g is some function and a special case was considered in Example 1.3. The regressor vector $\varphi(t)$ is a vector containing information about old inputs and outputs, for example, the elements of $\varphi(t)$ consist of old input and output components. The most general model structure that is going to be considered in this thesis is the nonlinear state-space model

$$x(t+1) = f(x(t), u(t); \theta), \quad (3.24a)$$

$$y(t) = h(x(t), u(t); \theta) + e(t), \quad (3.24b)$$

where f and h are nonlinear polynomial functions. The identification of nonlinear models is a vast field where the utilization of structure is vital for a successful estimation of the parameters. Details and references to different methods are given in Ljung (1999).

3.2.4 General Theory

In the sections above, we have used the concept of *model structures* quite freely without formalizing its meaning. This is often sufficient when dealing with practical aspects of system identification, but can be quite limiting when analyzing certain theoretical properties. For example, consider the concept of *identifiability* for a model structure, which, loosely speaking, means that, the parameters can be determined uniquely from a model defined by the model structure.

Example 3.2: (Identifiability)

Consider the model structure defined by the difference equation

$$y(t) = \theta^2 u(t-1) + e(t), \quad (3.25)$$

where $e(t)$ is white noise and the parameter $\theta \in \mathbb{R}$ is free. Now, the model given by

$$y(t) = u(t-1) + e(t), \quad (3.26)$$

lies in the set of models given by (3.25) for all $\theta \in \mathbb{R}$. However, when trying to determine the parameter which gives the model (3.26), one finds that both $\theta = 1$ and $\theta = -1$ result in the same model. Thus, the model structure (3.25) is not identifiable.

The example above indicates that the notion of identifiability is important, since it determines how a model structure may be used in practice. If the model is only to be used for predicting the output of the system, then the identifiability of the model structure is not important. For instance, in the example above, it does not matter which parameter value one chooses since both values yield the same predicted output. On the other hand, if the parameters represent some physical entity one must be careful when deriving conclusions about the system, since a positive parameter value may have a totally different interpretation than a negative one.

To be able to analyze the identifiability of model structures, it is necessary to have a well defined framework. Furthermore, the framework needs to be general so that all kinds of model structures, both linear and nonlinear, can be analyzed in the same way.

The presentation given below follows Ljung (1999, Sections 4.5 and 5.7) quite closely and we start by defining the concept of a model. In the following, let Z^t denote a data set (3.1) containing input and output pairs from time 1 up until time t .

Definition 3.2. A predictor model m of a dynamical system is a sequence of functions

$$g_m : \mathbb{R} \times \mathbb{R}^{n_y(t-1)} \times \mathbb{R}^{n_u(t-1)} \rightarrow \mathbb{R}^{n_y}, \quad t \in \mathbb{Z}_+,$$

which predicts the output $y(t)$ from past data, that is,

$$\hat{y}(t|t-1) = g_m(t, Z^{t-1}).$$

As an intermediate step to finding a rigorous definition of the notion of a model structure, we need the following concept.

Definition 3.3. A model set \mathcal{M}^* is a collection of predictor models

$$\mathcal{M}^* = \{g_{m_\alpha} \mid \alpha \in \mathcal{A}\},$$

where \mathcal{A} is some index set.

The predictor associated with (3.25) is easily seen to be $\hat{y}(t|t-1, \theta) = \theta^2 u(t-1)$, assuming that the noise $e(t)$ is white with zero mean. According to the definition,

$$\mathcal{M}^* = \{\theta^2 u(t-1) \mid \theta \in \mathbb{R}\},$$

is an example of a model set.

Definition 3.4. A *model structure* \mathcal{M} is a differentiable mapping from a connected open subset $\mathcal{D}_{\mathcal{M}} \subset \mathbb{R}^{n_{\theta}}$ to a model set \mathcal{M}^* , such that the gradients of the predictor functions exist and are stable.

The definition above includes a technical requirement that the gradients of the predictor functions are *stable* (see, for instance, Definition 5.1 in Khalil, 2002) which is used to prove convergence and consistency properties of the PEM. This property is not essential for this presentation and the reader is referred to Ljung (1999, Chapter 8).

Now, let us once again consider (3.25). Then

$$\mathcal{M}^* = \{\theta^2 u(t-1) \mid \theta \in \mathbb{R}\},$$

is the corresponding model set. Define the mapping \mathcal{M} as $\theta \mapsto \theta^2 u(t-1)$, which maps the parameter θ to the elements of the model set \mathcal{M}^* . Since $\mathcal{D}_{\mathcal{M}} = \mathbb{R}$ is an open and connected set, and the gradient of the predictor function is given by

$$\frac{d}{d\theta} \hat{y}(t|t-1, \theta) = 2\theta u(t-1),$$

it is clear that \mathcal{M} constitutes a model structure. Thus, the opening statement in Example 3.2 is valid if one refers *the model structure defined by (3.25)* to \mathcal{M} .

We are now ready to define the concept of identifiability.

Definition 3.5. A model structure \mathcal{M} is *globally identifiable* at $\theta^* \in \mathcal{D}_{\mathcal{M}}$ if $\mathcal{M}(\theta) = \mathcal{M}(\theta^*)$ for some $\theta \in \mathcal{D}_{\mathcal{M}}$ implies that $\theta = \theta^*$.

The strict formulation of the meaning of the equality $\mathcal{M}(\theta) = \mathcal{M}(\theta^*)$ in the definition above is quite technical and the reader is referred to Ljung (1999) for details. For the simple examples used in this thesis it suffices to treat the equality as meaning that the predictor models are effectively the same.

In Example 3.2 we hinted that the model structure \mathcal{M} generated by (3.25) is not globally identifiable. This is in alignment with the definition above, since

$$\mathcal{M}(-1) = \mathcal{M}(1),$$

for all $t \in \mathbb{Z}_+$. The definition of a model structure is quite tedious to use and we will for the sake of convenience often refer to expressions like (3.25) as model structures.

3.3 Instrumental Variables

In the introduction of this chapter, the basics of the PEM for fitting a parametrized model structure to a sequence of input and output pairs (3.1) was presented as the solution to a, in general, non-convex optimization problem (see also Example 1.2). Thus, if the initial parameters for the local search optimization routine are chosen poorly, this means that one cannot guarantee that the globally optimal parameter values are found. Here, we will present a popular method for finding good initial estimates called the *instrumental variables* (IV) method (see Söderström and Stoica, 1980). The basics of this method for

linear regression models have already been given in Section 2.2 and here we will restrict ourselves to the estimation of single-input single-output LTI transfer-function models

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t), \quad (3.27)$$

where $G(q, \theta)$ and $H(q, \theta)$ are rational functions in the forward shift operator q and the coefficients are given by the elements of the parameter vector θ (see Section 3.2.1). Furthermore, we assume that $e(t)$ is white Gaussian noise with zero mean and that the system lies in the model structure defined by (3.27), that is, the system agrees with (3.27) for some choice $\theta = \theta_0$. If the deterministic part of (3.27) is given by

$$G(q, \theta) = \frac{B_p(q, \theta)}{A_p(q, \theta)},$$

then we can rewrite (3.27) as

$$A_p(q, \theta)y(t) = B_p(q, \theta)u(t) + A_p(q, \theta)H(q, \theta)e(t). \quad (3.28)$$

By introducing $\varphi(t)$ as in (3.13), we may write (3.28) as

$$y(t) = \varphi(t)^T \eta + v(t), \quad (3.29)$$

where the parameter vector η only contains the parameters involved in $G(q, \theta)$ and the noise contribution has been collected in

$$v(t) \triangleq A_p(q, \theta)H(q, \theta)e(t). \quad (3.30)$$

Now, consider the problem of estimating the parameters η given a dataset (3.1) generated by the system, that is, the dataset agrees with (3.27) for the choice $\theta = \theta_0$. If one could find an $\zeta(t)$ such that

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \zeta(t)v(t) = 0, \quad (3.31)$$

then it would, asymptotically as $N \rightarrow \infty$, hold that

$$\frac{1}{N} \sum_{t=1}^N \zeta(t)y(t) = \frac{1}{N} \sum_{t=1}^N \zeta(t)\varphi(t)^T \eta.$$

and an estimate of η could be found via

$$\hat{\eta}_N^{\text{IV}} = \arg \min_{\eta \in \mathbb{R}^{r_\eta}} \left\| \frac{1}{N} \sum_{t=1}^N \zeta(t)(y(t) - \varphi(t)^T \eta) \right\|_2^2. \quad (3.32)$$

The elements of $\zeta(t)$ are referred to as the *instrumental variables* and the ordinary least-squares estimate of η is retrieved by choosing $\zeta(t) = \varphi(t)$ in (3.32). So, how should the instrumental variables $\zeta(t)$ be chosen in this case? By considering (3.30), one notices that

$$v(t) = A_p(q, \theta_0)H(q, \theta_0)e(t),$$

for some realization of $e(t)$. Thus, the optimal instruments must depend on the system which is what we are trying to estimate. Also, one notices that (3.32) is not general enough and that one would want to include a filter which attenuates the noise of the residual $\varepsilon(t, \theta) = y(t) - \varphi(t)^T \eta$. This leads to the *extended IV* approach which finds

$$\hat{\eta}_N^{\text{EIV}} = \arg \min_{\eta \in \mathbb{R}^{n_\eta}} \left\| \frac{1}{N} \sum_{t=1}^N \zeta(t) L(q) (y(t) - \varphi(t)^T \eta) \right\|_Q^2, \quad (3.33)$$

for some user defined choice of filter $L(q)$ and weighting matrix Q . It can be shown (see, for example, Ljung, 1999) that the optimal choices, under the assumption that the system agrees with (3.27) for some $\theta = \theta_0$ and that $e(t)$ is white Gaussian noise with zero mean, is given by

$$\zeta^{\text{opt}}(t) = L^{\text{opt}}(q) \begin{pmatrix} -G(q, \theta_0)u(t-1) & \cdots & \\ & -G(q, \theta_0)u(t-n_a) & u(t-1) & \cdots & u(t-n_b) \end{pmatrix}^T,$$

where the filter $L^{\text{opt}}(q)$ is chosen as

$$L^{\text{opt}}(q) = \frac{1}{A_p(q, \theta_0)H(q, \theta_0)},$$

and Q^{opt} as the identity matrix of appropriate dimension. These particular choices are quite intuitive since $L^{\text{opt}}(q)v(t) = e(t)$ and the instrumental variables are built up from the noise free outputs and old inputs which are uncorrelated with $e(t)$.

Since the optimal IV method requires knowledge of the system, it is not feasible in practice and some approximations have to be made. To this end, a multistep IV algorithm called IV4 was invented which step by step removes the stochastic part of the data by generating and applying appropriate instrumental variables (see, for instance, Söderström and Stoica, 1980). The IV4 method is summarized in Algorithm 1.

Algorithm 1 IV4

Given: A dataset (3.1) and the model parameters $\{n_a, n_b, n_k\}$.

1) Rewrite (3.27) as a linear regression as in (3.28) - (3.29) and estimate θ via

$$\hat{\eta}_N^{(1)} = \arg \min_{\eta \in \mathbb{R}^{n_\eta}} \sum_{t=1}^N (y(t) - \varphi(t)^T \eta)^2,$$

and denote the corresponding transfer-function by $\widehat{G}_N^{(1)}(q) = \widehat{B}_N^{(1)}(q)/\widehat{A}_N^{(1)}(q)$.

2) Generate the instruments

$$\begin{aligned} x^{(1)}(t) &= \widehat{G}_N^{(1)}(q)u(t), \\ \zeta^{(1)}(t) &= (-x^{(1)}(t-1) \cdots -x^{(1)}(t-n_a) u(t-1) \cdots u(t-n_b))^T, \end{aligned}$$

and determine the IV estimate using these instruments

$$\hat{\eta}_N^{(2)} = \arg \min_{\eta \in \mathbb{R}^{n_\eta}} \left\| \frac{1}{N} \sum_{t=1}^N \zeta^{(1)}(t) (y(t) - \varphi(t)^T \eta) \right\|_2^2.$$

Denote the corresponding transfer-function by $\widehat{G}_N^{(2)}(q) = \widehat{B}_N^{(2)}(q) / \widehat{A}_N^{(2)}(q)$.

3) Let

$$\hat{w}_N^{(2)} = \widehat{A}_N^{(2)}(q)y(t) - \widehat{B}_N^{(2)}(q)u(t).$$

Postulate an AR process of order $n_a + n_b$, that is,

$$L(q)\hat{w}_N^{(2)} = e(t),$$

and estimate it using the LS method. Denote the result by $\widehat{L}_N(q)$.

4) Generate the instruments

$$x^{(2)}(t) = \widehat{G}_N^{(2)}(q)u(t),$$

$$\zeta^{(2)}(t) = \widehat{L}_N(q) \left(-x^{(2)}(t-1) \cdots -x^{(2)}(t-n_a) u(t-1) \cdots u(t-n_b) \right)^T,$$

and determine the final estimate

$$\hat{\eta}_N^{\text{IV}} = \arg \min_{\eta \in \mathbb{R}^{n_\eta}} \left\| \frac{1}{N} \sum_{t=1}^N \zeta^{(2)}(t) \widehat{L}_N(q) (y(t) - \varphi(t)^T \eta) \right\|_2^2.$$

This algorithm is implemented in the *system identification toolbox* for MATLAB (see Ljung, 2009) and is used to find initial estimates for the PEM. For an extensive coverage of the IV method and its properties the reader is referred to, for example, Söderström and Stoica (1980). The IV4 method only finds an estimate of the deterministic part of (3.27), that is, no noise model $H(q, \theta)$ is estimated. If a noise model is sought, additional steps are needed. As an example, consider the estimation of an ARMAX model

$$y(t) = \frac{B_p(q)}{A_p(q)} u(t) + \frac{C_p(q)}{A_p(q)} e(t), \quad (3.34)$$

given a dataset (3.1). Then one can take the following steps to find an estimate.

1) Apply the IV4 algorithm to find an estimate of the deterministic part

$$G(q, \hat{\theta}) = \frac{\widehat{B}_N(q)}{\widehat{A}_N(q)}.$$

2) Find the residual

$$\hat{w}_N(t) = \widehat{A}_N(q)y(t) - \widehat{B}_N(q)u(t).$$

- 3) Postulate an AR process $L(q)\hat{w}_N(t) = e(t)$ and estimate it using the least-squares method. Denote the result by $\hat{L}_N(q)$.
- 4) Determine an estimate of the innovations by $\hat{e}_N(t) = \hat{L}_N(q)\hat{w}_N(t)$ and find the least-squares estimate of $C_p(q)$ via the relationship

$$\hat{w}_N(t) - \hat{e}_N(t) = (C_p(q) - 1)\hat{e}_N(t) + \tilde{e}(t),$$

where the noise contribution has been collected in $\tilde{e}(t)$.

The steps 3)–4) constitutes a method that is often called *Durbin's method* and was first presented in Durbin (1959). For different methods of estimating the stochastic part of (3.27), see Stoica and Moses (2005).

In this section we have shown how to utilize the IV method to find estimates of transfer-function models (3.27). The case when one is interested in obtaining estimates of state-space models is a bit more involved and for that a different method, which utilizes the IV philosophy, is better suited.

3.4 Subspace Identification

In this section, a presentation of the *subspace identification* (SID) approach to the estimation of state-space models will be given. The method is mainly based on linear algebraic techniques, where projections are used to find the range and null spaces of certain linear mappings (represented by matrices) which reveal information about the matrices in the state-space model. This also enables efficient implementations based on numerical linear algebra (see, for instance, Golub and Van Loan, 1996). A thorough treatment of the SID framework can be found in Van Overschee and De Moor (1996a) and a nice introduction is given in Verhaegen and Verdult (2007).

The first part of this section deals with the estimation of the discrete-time state-space models, where our presentation is based on Verhaegen and Verdult (2007) and Viberg et al. (1997). A different view of SID in the discrete-time case can be found in Jansson and Wahlberg (1996) and Savas and Lindgren (2006). Next, a short description of a SID method for continuous-time state-space models, similar to the discrete-time case, is given. It is based on frequency-domain techniques and a full description of the algorithm can be found in McKelvey and Akçay (1994) and Van Overschee and De Moor (1996b).

The results in this presentation are not new and the reader who is already familiar with SID might just want to browse the parts titled “*estimating $B(\theta)$ and $D(\theta)$* ” and “*estimating $K(\theta)$* ” which are important in a later chapter.

3.4.1 Discrete Time

Consider the discrete-time state-space model (3.15). For notational convenience, we will often not explicitly write out the dependence of the unknown parameter vector θ . Furthermore, we will assume that the system matrix $A(\theta)$ is Hurwitz and that the system operates

in open loop. Now, recursive application of (3.15) yields

$$y(t + \tau) = CA^\tau x(t) + \sum_{i=0}^{\tau-1} CA^{\tau-i-1} Bu(t+i) + Du(t+\tau) + \sum_{i=0}^{\tau-1} CA^{\tau-i-1} w(t+i) + v(t+\tau). \quad (3.35)$$

Introduce the vector of α stacked future outputs

$$y_\alpha(t) \triangleq (y(t)^T \quad y(t+1)^T \quad \cdots \quad y(t+\alpha-1)^T)^T, \quad (3.36)$$

where α is a user-defined prediction horizon. Similarly, define the vectors of stacked inputs, process and measurement noise as $u_\alpha(t)$, $w_\alpha(t)$, and $v_\alpha(t)$, respectively. In this notation, (3.35) can be expressed as

$$y_\alpha(t) = \Gamma_\alpha x(t) + \Phi_\alpha u_\alpha(t) + \Psi_\alpha w_\alpha(t) + v_\alpha(t), \quad (3.37)$$

where

$$\Gamma_\alpha \triangleq \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{\alpha-1} \end{pmatrix}, \quad \Phi_\alpha \triangleq \begin{pmatrix} D & 0 & \cdots & 0 \\ CB & D & & \\ \vdots & & \ddots & \\ CA^{\alpha-2}B & CA^{\alpha-3}B & \cdots & D \end{pmatrix}, \quad (3.38)$$

and

$$\Psi_\alpha \triangleq \begin{pmatrix} 0 & 0 & \cdots & 0 \\ C & 0 & & \\ CA & C & \ddots & \\ \vdots & & & \\ CA^{\alpha-2} & CA^{\alpha-3} & \cdots & 0 \end{pmatrix}. \quad (3.39)$$

The matrix Γ_α in (3.38) is referred to as the *extended observability matrix*. By stacking the data, (3.37) can be written in matrix form as

$$Y_{\beta,\alpha,N} = \Gamma_\alpha X_{\beta,N} + \Phi_\alpha U_{\beta,\alpha,N} + \Psi_\alpha W_{\beta,\alpha,N} + V_{\beta,\alpha,N}, \quad (3.40)$$

where we introduced the notation

$$Y_{\beta,\alpha,N} \triangleq (y_\alpha(\beta) \quad y_\alpha(\beta+1) \quad \cdots \quad y_\alpha(\beta+N-1)), \quad (3.41)$$

with $U_{\beta,\alpha,N}$, $W_{\beta,\alpha,N}$, and $V_{\beta,\alpha,N}$ defined accordingly, and

$$X_{\beta,N} \triangleq (x(\beta) \quad x(\beta+1) \quad \cdots \quad x(\beta+N-1)). \quad (3.42)$$

The parameter β is a design variable, whose interpretation will become clear in the ensuing paragraphs. Equation (3.40) is often referred to as the *data equation* and constitutes the foundation of modern approaches to SID (see, for instance, Van Overschee and De Moor, 1996a).

Estimating $A(\theta)$ and $C(\theta)$

In this presentation, we are going to make use of the IV framework for SID given in Viberg et al. (1997). The basic idea is to isolate the extended observability matrix Γ_α in (3.40) via appropriate matrix projections explained below. Once an estimate of Γ_α has been found, it is straightforward to create estimates of $A(\theta)$ and $C(\theta)$ via the relationship (3.38). First, the operator

$$\Pi_{\bar{U}_{\beta,\alpha,N}}^\perp \triangleq I - U_{\beta,\alpha,N}^T (U_{\beta,\alpha,N} U_{\beta,\alpha,N}^T)^{-1} U_{\beta,\alpha,N}, \quad (3.43)$$

projects the row space of a matrix onto the orthogonal complement of the row space of the matrix $U_{\beta,\alpha,N}$. Thus, it holds that $U_{\beta,\alpha,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp = 0$ and multiplying (3.40) by $\Pi_{\bar{U}_{\beta,\alpha,N}}^\perp$ from the right yields

$$Y_{\beta,\alpha,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp = \Gamma_\alpha X_{\beta,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp + \Psi_\alpha W_{\beta,\alpha,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp + V_{\beta,\alpha,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp. \quad (3.44)$$

Secondly, to eliminate the noise influence in (3.44), we need to find a matrix Z_β such that

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} W_{\beta,\alpha,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp Z_\beta^T &= 0, \\ \lim_{N \rightarrow \infty} \frac{1}{N} V_{\beta,\alpha,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp Z_\beta^T &= 0, \end{aligned} \quad (3.45)$$

and that the matrix

$$\lim_{N \rightarrow \infty} \frac{1}{N} X_{\beta,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp Z_\beta^T$$

has full rank n_x . Equation (3.44) would then, asymptotically, be equivalent to

$$Y_{\beta,\alpha,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp Z_\beta^T = \Gamma_\alpha X_{\beta,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp Z_\beta^T. \quad (3.46)$$

Hence, the column-space of Γ_α could then be determined from the matrix on the left hand side of (3.46). Under certain mild assumptions on the input signal and the noise contribution, it is shown in Viberg et al. (1997) that

$$Z_\beta = \begin{pmatrix} U_{0,\beta,N} \\ Y_{0,\beta,N} \end{pmatrix}, \quad (3.47)$$

where

$$U_{0,\beta,N} = (u_\beta(0) \quad u_\beta(1) \quad \cdots \quad u_\beta(N-1)), \quad (3.48a)$$

$$Y_{0,\beta,N} = (y_\beta(0) \quad y_\beta(1) \quad \cdots \quad y_\beta(N-1)), \quad (3.48b)$$

fulfills these requirements. Thus, the user-defined parameter β defines the number of old inputs and outputs that are going to be used as instrumental variables. The choice (3.47) is quite intuitive since the i th column of the matrices $W_{\beta,\alpha,N}$ and $V_{\beta,\alpha,N}$ only contains white noise terms from time $t = \beta + i - 1$ and onwards, so that any Z_β^T where the i th row is constructed from data prior to $t = \beta + i - 1$ will satisfy (3.45).

Now, as noted earlier, the extended observability matrix Γ_α can be estimated by the column-space of the matrix $Y_{\beta,\alpha,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp Z_\beta^T$. This subspace can be found via the *singular value decomposition* (SVD)

$$Y_{\beta,\alpha,N} \Pi_{\bar{U}_{\beta,\alpha,N}}^\perp Z_\beta^T = (U_1 \quad U_2) \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}, \quad (3.49)$$

where the $\alpha n_y \times n_x$ matrix U_1 determines a basis for the column-space (see, for example, Golub and Van Loan, 1996). Thus, an estimate of Γ_α is given by $\widehat{\Gamma}_\alpha = U_1$. Finally, from the relation (3.38) one can find estimates

$$\widehat{C} = \widehat{\Gamma}_{\alpha,1}, \quad \widehat{A} = \begin{pmatrix} \widehat{\Gamma}_{\alpha,1} \\ \vdots \\ \widehat{\Gamma}_{\alpha,\alpha-1} \end{pmatrix}^{-1} \begin{pmatrix} \widehat{\Gamma}_{\alpha,2} \\ \vdots \\ \widehat{\Gamma}_{\alpha,\alpha} \end{pmatrix}, \quad (3.50)$$

where

$$\widehat{\Gamma}_\alpha = \begin{pmatrix} \widehat{\Gamma}_{\alpha,1} \\ \vdots \\ \widehat{\Gamma}_{\alpha,\alpha} \end{pmatrix},$$

has been partitioned into α blocks of sizes $n_y \times n_x$.

Estimating $B(\theta)$ and $D(\theta)$

Assume now that the estimates \widehat{A} and \widehat{C} have been found according to (3.50). Setting $t = 0$ in (3.35) and replacing the variable τ in the resulting equation with t yields

$$y(t) = CA^t x(0) + \sum_{i=0}^{t-1} CA^{t-i-1} Bu(i) + Du(t) + n(t), \quad (3.51)$$

where the noise contributions have been collected in $n(t)$. Equivalently, (3.51) can be rewritten as

$$\begin{aligned} y(t) = CA^t x(0) + \left(\sum_{i=0}^{t-1} u(i)^T \otimes CA^{t-i-1} \right) \text{vec}(B) \\ + \left(u(t)^T \otimes I_{n_y} \right) \text{vec}(D) + n(t), \end{aligned} \quad (3.52)$$

where \otimes is the Kronecker product and the vec -operator stacks the columns of the argument in a column vector. Thus, since the noise $n(t)$ is uncorrelated with the input $u(t)$ due to the open loop assumption, the matrices B and D can be estimated via the least-squares problem

$$\arg \min_{\theta} \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi(t)^T \theta)^2, \quad (3.53)$$

where we have introduced $\theta = (x(0)^T \text{vec}(B)^T \text{vec}(D)^T)^T$ and

$$\varphi(t)^T = \left(\widehat{C} \widehat{A}^t \sum_{i=0}^{t-1} u(i)^T \otimes \widehat{C} \widehat{A}^{t-i-1} \quad u(t)^T \otimes I_{n_y} \right).$$

Estimating $K(\theta)$

Assume now that an innovation model (3.18) is to be estimated. Using the methodology presented above, estimates \hat{A} , \hat{B} , \hat{C} and \hat{D} may be found. Thus, what remains to be estimated is $K(\theta)$. To this end, we are going to use the approach presented in Van Overschee and De Moor (1996a) with the notation given in Haverkamp (2001). If an estimate of the state sequence $X_{\beta,N}$ was at hand, this would be a simple task. In fact, estimates of the process noise and the measurement noise are then given by

$$\begin{pmatrix} \widehat{W}_{\beta,1,N-1} \\ \widehat{V}_{\beta,1,N-1} \end{pmatrix} = \begin{pmatrix} \widehat{X}_{\beta+1,N} \\ Y_{\beta,1,N-1} \end{pmatrix} - \begin{pmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{pmatrix} \begin{pmatrix} \widehat{X}_{\beta,N-1} \\ U_{\beta,1,N-1} \end{pmatrix}, \quad (3.54)$$

which in turn yields estimates of the noise covariances

$$\begin{pmatrix} \widehat{Q} & \widehat{S} \\ \widehat{S}^T & \widehat{R} \end{pmatrix} = \frac{1}{N} \begin{pmatrix} \widehat{W}_{\beta,1,N} \\ \widehat{V}_{\beta,1,N} \end{pmatrix} \begin{pmatrix} \widehat{W}_{\beta,1,N} \\ \widehat{V}_{\beta,1,N} \end{pmatrix}^T. \quad (3.55)$$

Finally, an estimate of the Kalman filter gain $K(\theta)$ is given by the solution to the corresponding Riccati equation

$$\widehat{K} = (\widehat{A}\widehat{P}\widehat{C}^T + \widehat{S})(\widehat{C}\widehat{P}\widehat{C}^T + \widehat{R})^{-1}, \quad (3.56a)$$

$$\widehat{P} = \widehat{A}\widehat{P}\widehat{A}^T + \widehat{Q} - (\widehat{A}\widehat{P}\widehat{C}^T + \widehat{S})(\widehat{C}\widehat{P}\widehat{C}^T + \widehat{R})^{-1}(\widehat{A}\widehat{P}\widehat{C}^T + \widehat{S})^T. \quad (3.56b)$$

The difficult part of estimating $K(\theta)$ lies in reconstructing the state sequences $\widehat{X}_{\beta,N}$ and $\widehat{X}_{\beta+1,N}$. Now, assume that estimates \hat{A} , \hat{B} , \hat{C} and \hat{D} have been found via (3.50) and (3.53), respectively. With $w(t) = Ke(t)$ and $v(t) = e(t)$, the data equation (3.40) becomes

$$Y_{\beta,\alpha,N} = \widehat{\Gamma}_\alpha X_{\beta,N} + \widehat{\Phi}_\alpha U_{\beta,\alpha,N} + \widehat{\Psi}_\alpha E_{\beta,\alpha,N}, \quad (3.57)$$

where $\widehat{\Gamma}_\alpha$ and $\widehat{\Phi}_\alpha$ are as in (3.38) with the system matrices replaced by their respective estimate and

$$\widehat{\Psi}_\alpha \triangleq \begin{pmatrix} I_{n_y} & 0 & \dots & 0 \\ CK & I_{n_y} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{\alpha-2}K & \dots & CK & I_{n_y} \end{pmatrix}. \quad (3.58)$$

Introduce the instrumental variables as

$$\tilde{Z}_\beta \triangleq \begin{pmatrix} U_{\beta,\alpha,N} \\ U_{0,\beta,N} \\ Y_{0,\beta,N} \end{pmatrix}, \quad (3.59)$$

with the corresponding projection matrix

$$\Pi_{\tilde{Z}_\beta} = \tilde{Z}_\beta^T (\tilde{Z}_\beta \tilde{Z}_\beta^T)^{-1} \tilde{Z}_\beta. \quad (3.60)$$

Multiplying (3.57) by $\Pi_{\tilde{Z}_\beta}$ from the right and taking the limit as $\beta \rightarrow \infty$ yields

$$\lim_{\beta \rightarrow \infty} Y_{\beta,\alpha,N} \Pi_{\tilde{Z}_\beta} = \lim_{\beta \rightarrow \infty} \left(\widehat{\Gamma}_\alpha X_{\beta,N} \Pi_{\tilde{Z}_\beta} + \widehat{\Phi}_\alpha U_{\beta,\alpha,N} \Pi_{\tilde{Z}_\beta} + \widehat{\Psi}_\alpha E_{\beta,\alpha,N} \Pi_{\tilde{Z}_\beta} \right), \quad (3.61)$$

where $\widehat{\Gamma}_\alpha$ and $\widehat{\Phi}_\alpha$ have been constructed from the estimates \widehat{A} , \widehat{B} , \widehat{C} and \widehat{D} according to (3.38), respectively. Since the columns of $E_{\beta,\alpha,N}(t)$ are independent of the rows of \widetilde{Z}_β^T , the last term of the right hand side of (3.61) will disappear. Also, since $U_{\beta,\alpha,N}$ is completely spanned by $\Pi_{\widetilde{Z}_\beta}$, it holds that

$$\lim_{\beta \rightarrow \infty} U_{\beta,\alpha,N} \Pi_{\widetilde{Z}_\beta} = \lim_{\beta \rightarrow \infty} U_{\beta,\alpha,N},$$

and (3.61) may be reduced to

$$\lim_{\beta \rightarrow \infty} Y_{\beta,\alpha,N} \Pi_{\widetilde{Z}_\beta} = \widehat{\Gamma}_\alpha \lim_{\beta \rightarrow \infty} X_{\beta,N} \Pi_{\widetilde{Z}_\beta} + \lim_{\beta \rightarrow \infty} \widehat{\Phi}_\alpha U_{\beta,\alpha,N}. \quad (3.62)$$

Thus, to find an estimate of $X_{\beta,N}$, we need to be able to say something about

$$\lim_{\beta \rightarrow \infty} X_{\beta,N} \Pi_{\widetilde{Z}_\beta}.$$

To this end, insert (3.18b) into (3.18a) resulting in

$$\hat{x}(t+1) = \bar{A}\hat{x}(t) + \bar{B}u(t) + Ky(t), \quad (3.63)$$

where we have introduced

$$\bar{A} \triangleq \widehat{A} - K\widehat{C} \quad \text{and} \quad \bar{B} \triangleq \widehat{B} - K\widehat{D}.$$

Now, recursive application of (3.63) yields

$$\widehat{X}_{\beta,N} = \bar{A}^\beta \widehat{X}_{0,N} + \mathcal{C}_{u,\beta} U_{0,\beta,N} + \mathcal{C}_{y,\beta} Y_{0,\beta,N}, \quad (3.64)$$

where

$$\begin{aligned} \mathcal{C}_{u,\beta} &\triangleq (\bar{A}^{\beta-1} \bar{B} \quad \bar{A}^{\beta-2} \bar{B} \quad \dots \quad \bar{B}), \\ \mathcal{C}_{y,\beta} &\triangleq (\bar{A}^{\beta-1} K \quad \bar{A}^{\beta-2} K \quad \dots \quad K). \end{aligned}$$

Hence, if \bar{A} is stable then $\bar{A}^\beta \widehat{X}_{0,N}$ will disappear as $\beta \rightarrow \infty$ and $\widehat{X}_{\beta,N}$ is completely spanned by $\Pi_{\widetilde{Z}_\beta}$ in (3.60). This implies that

$$\lim_{\beta \rightarrow \infty} X_{\beta,N} \Pi_{\widetilde{Z}_\beta} = \lim_{\beta \rightarrow \infty} X_{\beta,N}.$$

Thus, (3.62) can, finally, be written as

$$\lim_{\beta \rightarrow \infty} Y_{\beta,\alpha,N} \Pi_{\widetilde{Z}_\beta} = \widehat{\Gamma}_\alpha \lim_{\beta \rightarrow \infty} X_{\beta,N} + \lim_{\beta \rightarrow \infty} \widehat{\Phi}_\alpha U_{\beta,\alpha,N}$$

and an estimate of $X_{\beta,N}$, for large β , may be found by

$$\widehat{X}_{\beta,N} = \widehat{\Gamma}_\alpha^{-1} \left(Y_{\beta,\alpha,N} \Pi_{\widetilde{Z}_\beta} - \widehat{\Phi}_\alpha U_{\beta,\alpha,N} \right). \quad (3.65)$$

Thus, there are several steps involved to find an estimate of the Kalman gain $K(\theta)$. First, one needs to reconstruct the state sequences $X_{\beta,N}$ and $X_{\beta+1,N}$ to get estimates of the covariance matrices (3.55). Finally, one needs to solve the Riccati equation (3.56) for \widehat{K} .

Summary

The estimation of a complete state-space innovations model (3.18), as described in the sections above, involves several steps. First, the matrices $A(\theta)$ and $C(\theta)$ are estimated via (3.50). Then, these estimates are used to estimate the matrices $B(\theta)$ and $D(\theta)$ via (3.53). Finally, an estimate of $K(\theta)$ is found by the Riccati equation (3.56). The steps are summarized in Algorithm 2.

Algorithm 2 SID

Given: A dataset (3.1), the model order n_x and the user-defined prediction and instrumental variables horizons $\{\alpha, \beta\}$.

- 1) Construct the Hankel matrices $U_{\beta, \alpha, N}$ and $Y_{\beta, \alpha, N}$ as in (3.41). Also, find the corresponding projection matrix $\Pi_{U_{\beta, \alpha, N}}^\perp$ via (3.43) and construct the instruments Z_β^T as in (3.47).
 - 2) Find the column space of $Y_{\beta, \alpha, N} \Pi_{U_{\beta, \alpha, N}}^\perp Z_\beta^T$ via (3.49) as an estimate $\hat{\Gamma}_\alpha$ of the extended observability matrix (3.38).
 - 3) Determine the estimates \hat{A} and \hat{C} of $A(\theta)$ and $C(\theta)$, respectively, via (3.50).
 - 4) Determine the estimates \hat{B} and \hat{D} of $B(\theta)$ and $D(\theta)$, respectively, via (3.53).
 - 5) Reconstruct the state sequences $X_{\beta, N}$ and $X_{\beta+1, N}$ via (3.65) using the instruments defined in (3.59) and $\hat{\Phi}_\alpha$ as in (3.38) with the system matrices replaced by the corresponding estimates.
 - 6) Estimate the covariance matrices via (3.55) and find an estimate \hat{K} of $K(\theta)$ by solving the corresponding Riccati equation (3.56).
-

The precise effects of the user-defined prediction and instrumental variables horizons α and β are not yet fully understood, other than that they must be greater than n_x . For efficient implementations and suggestions of different design choices, the reader is referred to Van Overschee and De Moor (1996a) and the references therein.

3.4.2 Continuous Time

So far we have only considered estimating discrete-time state-space models (3.15) from sampled data (3.1). In this section, we will deal with the identification of continuous-time state-space models

$$\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t), \quad (3.66a)$$

$$y(t) = C(\theta)x(t) + D(\theta)u(t) + e(t), \quad (3.66b)$$

where the output $y(t)$, the input $u(t)$ and the noise $e(t)$ are assumed to be differentiable. Furthermore, the input and the noise are assumed to be uncorrelated. The presentation below is based on frequency domain techniques presented in McKelvey and Akçay (1994). The resulting algorithm is quite similar to the discrete-time SID method presented above.

Consider the continuous-time state-space model (3.66). Now, by introducing

$$y_\alpha(t) = \left(y(t) \quad \frac{d}{dt}y(t) \quad \cdots \quad \frac{d^{\alpha-1}}{dt^{\alpha-1}}y(t) \right)^T,$$

similarly for $u(t)$ and $e(t)$, it holds that

$$y_\alpha(t) = \Gamma_\alpha x(t) + \Phi_\alpha u_\alpha(t) + e_\alpha(t). \quad (3.67)$$

Here, Γ_α and Φ_α are the same matrices as in the time-discrete case (3.38). Applying the continuous-time Fourier transform to (3.67) yields

$$W_\alpha(\omega) \otimes Y(\omega) = \Gamma_\alpha X(\omega) + \Phi_\alpha W_\alpha(\omega) \otimes U(\omega) + W_\alpha(\omega) \otimes E(\omega), \quad (3.68)$$

where $Y(\omega)$, $X(\omega)$, $U(\omega)$ and $E(\omega)$ are the Fourier transforms of the signals $y(t)$, $x(t)$, $u(t)$ and $e(t)$, respectively and

$$W_\alpha \triangleq (1 \quad i\omega \quad \cdots \quad (i\omega)^{\alpha-1})^T.$$

Assume that we have samples of the Fourier transforms of the input and output signals at frequencies $(\omega_k)_{k=1}^M$. Introduce

$$Y_{\beta,\alpha,M} \triangleq (W_\alpha(\omega_\beta) \otimes Y(\omega_\beta) \quad \cdots \quad W_\alpha(\omega_M) \otimes Y(\omega_M)), \quad (3.69)$$

and similarly for $U_{\beta,\alpha,M}$ and $E_{\beta,\alpha,M}$. Now, (3.68) can be written as

$$Y_{\beta,\alpha,M} = \Gamma_\alpha X_{\beta,M} + \Phi_\alpha U_{\beta,\alpha,M} + E_{\beta,\alpha,M}, \quad (3.70)$$

with

$$X_{\beta,M} \triangleq (X(\omega_\beta) \quad \cdots \quad X(\omega_M)).$$

Equation (3.70) is quite similar to the discrete-time data equation (3.40). The only significant difference is that the elements of the matrices in (3.70) are in general complex numbers and certain modifications have to be made. Define the projection matrix

$$\Pi_{U_{\beta,\alpha,M}}^\perp \triangleq I - U_{\beta,\alpha,M}^H (U_{\beta,\alpha,M} U_{\beta,\alpha,M}^H)^{-1} U_{\beta,\alpha,M}, \quad (3.71)$$

where superscript H denotes the complex conjugate transpose of a matrix. Multiplying (3.70) with (3.71) from the right yields

$$Y_{\beta,\alpha,M} \Pi_{U_{\beta,\alpha,M}}^\perp = \Gamma_\alpha X_{\beta,M} \Pi_{U_{\beta,\alpha,M}}^\perp + E_{\beta,\alpha,M} \Pi_{U_{\beta,\alpha,M}}^\perp \quad (3.72)$$

Now, it can be shown that

$$Y_{\beta,\alpha,M} \Pi_{U_{\beta,\alpha,M}}^\perp U_{1,\beta,M}^H = \Gamma_\alpha X_{\beta,M} \Pi_{U_{\beta,\alpha,M}}^\perp U_{1,\beta,M}^H, \quad (3.73)$$

holds asymptotically as $M \rightarrow \infty$, where the matrix $U_{1,\beta,M}$ is the instrumental variables matrix (see, for example, McKelvey and Akçay, 1994). In a similar fashion as in the discrete-time case one can now find the column space of Γ_α via the matrix $G \triangleq Y_{\beta,\alpha,M} \Pi_{U_{\beta,\alpha,M}}^\perp U_{1,\beta,M}^H$ (see McKelvey and Akçay, 1994)

$$(\operatorname{Re}(G) \quad \operatorname{Im}(G)) = (U_1 \quad U_2) \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}, \quad (3.74)$$

where the operators Re and Im find the real and imaginary parts of a complex number, respectively. An estimate $\hat{\Gamma}_\alpha$ of Γ_α is now found by U_1 and the matrices $A(\theta)$ and $C(\theta)$ can be estimated as in (3.50).

To find estimates of $B(\theta)$ and $D(\theta)$, one only needs to solve

$$\arg \min_{B,D} \sum_{t=1}^M \left| Y(\omega_t) - (D + \hat{C}(i\omega_t I - \hat{A})^{-1} B) U(\omega_t) \right|^2. \quad (3.75)$$

For an efficient implementation of the frequency domain SID method for estimating continuous-time state-space models (3.66), the reader is referred to McKelvey and Akçay (1994). These problems are often ill-conditioned and for higher order systems certain care is needed when implementing an algorithm (see, for instance, Van Overschee and De Moor, 1996b).

The SID algorithms presented in this section and in the previous one have been proved to work well in practice and have become de facto standard for initializing the PEM when estimating state-space models given a dataset. One drawback with these methods is that it is difficult to include prior information about the system into the model. For example, physically motivated state-space models often contain elements in the system matrices that are known beforehand. This limitation will be discussed in a later chapter.

3.5 An Algebraic Approach

In the well-cited paper by Ljung and Glad (1994), a new framework for analyzing the identifiability of nonlinear model structures were introduced, based on the techniques of differential algebra. The procedure used in Ljung and Glad (1994) is called Ritt's algorithm and provides means to simplify a system of differential equations via algebraic manipulations and differentiations of the signals involved. For instance, the main theorem in Ljung and Glad (1994) states that any nonlinear model structure

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), \theta), \\ y(t) &= h(x(t), u(t), \theta), \end{aligned} \quad (3.76)$$

where f and h are polynomials, is globally identifiable if and only if it can be rewritten as a linear regression model

$$\tilde{f}(u(t), y(t), \dot{u}(t), \dot{y}(t), \dots) = \tilde{g}(u(t), y(t), \dot{u}(t), \dot{y}(t), \dots) \theta, \quad (3.77)$$

for some polynomials \tilde{f} and \tilde{g} . The implications of this result for system identification is twofold. First of all, instead of analyzing the properties of a globally identifiable nonlinear model structure (3.76), which may be quite difficult, one can work with an equivalent linear regression model (3.77), which significantly simplifies the analysis. Second, since an algorithm that performs the transformation of the nonlinear model to an equivalent linear regression model is provided, this also implies that the estimation of the parameters is trivial once the transformation has been made.

The price to be paid for using Ritt's algorithm to transform the nonlinear model structure (3.76) into the linear regression model (3.77) is the introduction of derivatives of the

signals $u(t)$ and $y(t)$. Thus, the framework proposed in Ljung and Glad (1994) requires noise-free data and means to find the exact derivatives of the signals involved. Let us consider a simple example to illustrate the steps involved in the manipulations.

Example 3.3

Consider the following continuous-time model structure

$$y(t) = \theta u(t) + \theta^2 \dot{u}(t), \quad (3.78)$$

where the dot operator denotes differentiation with respect to time t . As was noted in the discrete-time case depicted in Example 1.3, applying the PEM directly to the model structure (3.78) may result in a non-convex optimization problem, see Figure 1.3. Instead, let us try to apply Ritt's algorithm, as described in Ljung and Glad (1994). Via a series of differentiations and polynomial divisions, the algorithm provides a linear regression model

$$y(t)\ddot{u}(t) - \dot{y}(t)\dot{u}(t) = (u(t)\ddot{u}(t) - \dot{u}^2(t))\theta, \quad (3.79)$$

plus an additional relationship between the input and output signals that must be fulfilled for all $t \in \mathbb{R}$

$$\begin{aligned} y(t)\dot{u}^4(t) - u(t)\dot{u}^3(t)\dot{y}(t) - \dot{u}^3(t)\dot{y}^2(t) - u(t)y(t)\dot{u}^2(t)\ddot{u}(t) \\ + u^2(t)\dot{u}(t)\dot{y}(t)\ddot{u}(t) + 2y(t)\dot{u}^2(t)\dot{y}(t)\ddot{u}(t) - y^2(t)\dot{u}(t)\ddot{u}^2(t) = 0. \end{aligned} \quad (3.80)$$

Thus, to estimate the unknown parameter θ in (3.78), given a sequence of input and output data, one can use (3.79) to get an initial estimate for the PEM which guarantees, in the noise-free case and with exact derivatives, that the global optimum will be found.

The linear regression (3.79) can be derived in quite a simple manner by eliminating the nonlinear term θ^2 via some elementary algebra. First, differentiate (3.78) with respect to t , remembering that θ is constant, which results in

$$\dot{y}(t) = \theta\dot{u}(t) + \theta^2\ddot{u}(t). \quad (3.81)$$

By comparing (3.78) with (3.81) one notices that multiplying (3.78) by $\ddot{u}(t)$ and (3.81) with $\dot{u}(t)$ and subtracting the results, yields the same result as in (3.79). These simple algebraic manipulations is basically what Ritt's algorithm does and the multiplication and subtraction scheme used above forms what we know as polynomial division.

In the example above one notices that by making use Ritt's algorithm to eliminate the nonlinearities associated with θ in (3.78), one increases the complexity of the interactions between the signals $y(t)$, $u(t)$ and the corresponding derivatives. This may not be a problem in the noise-free case, but if a noise signal $e(t)$ was added to (3.78) it would also undergo the nonlinear transformations. In the general case, this implies that products between the noise $e(t)$, the input $u(t)$, the output $y(t)$, and their time derivatives would appear in the linear regression model. This may complicate the estimation problem significantly, since the regressor variables and the noise signal may become correlated, and Ljung and Glad (1994) does not consider the case when noise is present. In a later chapter we will discuss the generalization of these results to discrete time. Besides making it possible to analyze discrete-time nonlinear systems, this opens up the possibility of dealing with noise.

The methods of differential algebra used here was first formalized by Ritt (1950) and then further developed by Kolchin (1973). These methods were much later introduced into the automatic control community by Diop (1991). Further details on the use of differential algebraic methods in automatic control problems can be found in Fliess and Glad (1994) and algorithmic details are given in Glad (1997).

3.6 Model Validation

An important part of the system identification cycle is to ensure the validity of the model that has been estimated. To this end, several methods have been developed (see, for example, Ljung, 1999), and in this section we will only cover a small subset. For further discussions and references regarding the subject of model validation, the reader is referred to the books by Ljung (1999) and Söderström and Stoica (1989).

Many of the validation tests in system identification make, in some way, use of the residual

$$\varepsilon(t, \hat{\theta}_N) = y(t) - \hat{y}(t|\hat{\theta}_N),$$

where the parameter estimate $\hat{\theta}_N$ has been found using some method. A simple, yet effective validation procedure, is the *cross-validation* method. Here, the predicted output of the model for a different dataset, referred to as *validation* data, than the dataset used when estimating the model, referred to as *estimation* data, is compared to the measured output. Different criteria have been proposed to quantify the closeness of the predicted and the measured output. A popular choice is the *model fit* value

$$\text{fit} = 100 \left(1 - \frac{\|\varepsilon(t, \hat{\theta}_N)\|_2}{\|y(t) - \bar{y}\|_2} \right), \quad (3.82)$$

which gives the relative performance increase, in percent, of using the estimated model compared to just using the mean \bar{y} of the output as a predictor. The model fit (3.82) generally depends on the amount of noise in the data and more noise usually means a lower model fit. Therefore it is recommended that one also considers the graph containing the predicted and the measured output side by side and takes the prediction performance into account when determining if the model is valid or not.

Another simple test is given by the *cross-correlation* method. This method is based on the covariance between the residuals and past inputs

$$\hat{R}_{\varepsilon u}^N(\tau) \triangleq \frac{1}{N} \sum_{t=1}^N \varepsilon(t) u(t - \tau). \quad (3.83)$$

A large value of the covariance (3.83) for some time shift τ indicates that the corresponding shifted input is important for describing the data and should be incorporated into the model. This is usually found out by a simple χ^2 hypothesis test (see Ljung, 1999). It can also be informative to consider the correlation among the residuals themselves

$$\hat{R}_{\varepsilon \varepsilon}^N(\tau) \triangleq \frac{1}{N} \sum_{t=1}^N \varepsilon(t) \varepsilon(t - \tau). \quad (3.84)$$

A large value of the correlation (3.84) for some time shift $\tau \neq 0$ indicates that some part of $\varepsilon(t, \hat{\theta}_N)$ could have been predicted from past data. Thus, the measured output $y(t)$ could have been predicted better. This effect can also be detected by a simple χ^2 hypothesis test (see Ljung, 1999)

If several models, within the same model set, with different orders have been estimated one should not only take the adaptation to data into account when deciding which of the models to use, but also the complexity of the model. To this end, several measures that besides taking the fit to data into account also penalizes the number of parameters used to achieve this fit. In this thesis we will only make use of the AIC and MDL selection criteria, which have already been presented in Section 2.4.

The methods described above are quite simple but have shown to work well in practice. It is important to note that the model validation methods usually focus on detecting one particular deficiency of the estimated models and therefore it is important to use several methods when testing their validity. In this thesis, model validation will not play a major role and the methods described in this section are sufficient for our purpose.

4

The Nonnegative Garrote in System Identification

Regressor and structure selection of linear regression models have been thoroughly researched in the statistical community for some time. Different regressor selection methods have been proposed, such as the ridge and lasso regression methods, see Section 2.3. Especially the lasso regression has won popularity because of its ability to set less important parameters exactly to zero. However, these methods were not developed with dynamical systems in mind, where the regressors are ordered via the time lag. For this end, a modified variant of the nonnegative garrote (NNG) method (Breiman, 1995) will be analyzed.

4.1 Introduction

An important subproblem of system identification is to find the model with the *lowest complexity*, within some model set, which describes a given set of data *sufficiently well*. There are several reasons for considering this problem. One reason is that even though a higher model complexity will yield a better adaptation to the data used for the estimation, it might be that the model is adapting too well. Thus, the model does not properly represent the system itself, only the data that is used for the estimation. Another aspect is the possible constraints on the computational resources available in certain time-critical applications. A higher model complexity usually means a higher computational cost, both in time and in memory. Hence, a model with lower complexity might be preferred to one with higher complexity if the loss in the ability to describe the data is not too great.

In the statistical community, this problem is often referred to as the *regressor selection* problem, for which several solutions have been proposed in the literature (see Chapter 2 for a selection of these methods). For linear regression models, the complexity is defined as the total number of parameters used. This is a good measure of the complexity when there is no dynamical dependence between the regressors. For dynamical systems, such as the ARX model structure (3.8), the complexity is not defined only by the number of

parameters used, but also via the *order* of the model. Here, the order is defined by the maximum of the highest time lag of the input and the highest time lag of the output used in the model. Choosing the model which describes the data sufficiently well given a number of parameters is not uniquely defined and we have already presented several criteria which consider both the adaptation to data and the number of parameters used in Section 2.4.

Now, let us consider using the lasso and the NNG regressor selection methods presented in Section 2.3 for estimating the order of an ARX model

$$A_p(q)y(t) = B_p(q)u(t) + e(t), \quad (4.1)$$

given measurements from a real-life process. This model structure may be represented as a linear regression

$$y(t) = \varphi(t)^T \theta + e(t), \quad (4.2)$$

where

$$\begin{aligned} \varphi(t) &\triangleq (-y(t-1) \quad \cdots \quad -y(t-n_a) \quad u(t-1) \quad \cdots \quad u(t-n_b))^T, \\ \theta &\triangleq (a_1 \quad \cdots \quad a_{n_a} \quad b_1 \quad \cdots \quad b_{n_b})^T, \end{aligned}$$

and the use of the regularization methods presented in Section 2.3 is straightforward.

Example 4.1: (Fan and Plate)

Let us consider using the lasso (2.10) and the NNG (2.14) regularization methods to estimate the order of a “fan and plate” process illustrated in Figure 4.1. The system consists

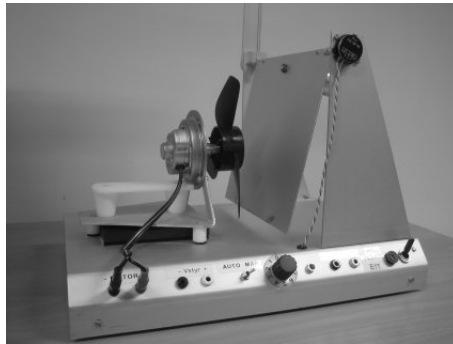


Figure 4.1: The “Fan and Plate” process

of a fan connected to an electrical motor and a hinged plate which is able to swing with low friction. The electrical motor is driven by an electrical current which forces the fan to generate wind gusts. The wind then hits the plate which starts to move accordingly. This system is known to be well-behaved from an identification point of view and can be well approximated by an ARX model (4.1). The input $u(t)$ is the voltage driving the fan and the output $y(t)$ is the angle of the plate. Data has been collected using a random binary telegraph signal (see Ljung, 1999) as input, switching between 2 V and 6 V with a probability of 0.08 and the sampling time was set to $T_s = 0.04$ s. In total, 1000 data points have been collected where a part is illustrated in Figure 4.2. The data is split into two parts, the

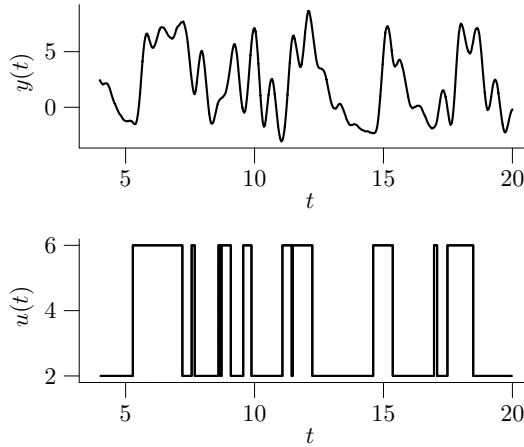


Figure 4.2: Parts of the data collected from the fan and plate process. The lower plot shows the input signal and the upper plot the corresponding output.

first two thirds has been used as estimation data and the remaining for validation.

As discussed in Section 2.3, it can be shown that the solution paths of the lasso (2.10) and the NNG (2.14) are piecewise affine functions of the penalizing parameter λ . This property enables efficient algorithms for finding the complete solution path and the values of λ where the local affine representation of the solution changes will be referred to as *breakpoints* (see Figure 2.2). The solution paths are illustrated in Figure 4.3, where the lasso is given in the left column and the NNG in the right. The first row shows the values of the model fit (3.82) for the corresponding method evaluated on validation data, with the indices of the breakpoints on the horizontal axis. The second row depicts which parameters in the $A_p(q)$ polynomial in (4.1) that are nonzero and similarly for the last row where the nonzero coefficients of $B_p(q)$ are represented.

For the lasso method (the first column of Figure 4.3), the model fit is good up to a certain point and the number of nonzero parameters used to achieve the fit is small, especially for the $A_p(q)$ polynomial illustrated in the second row. In fact, minimizing the MDL model order selection criterion $W_N^{\text{MDL}}(\hat{\theta}_N^{\text{lasso}}(\lambda))$ in (2.17) evaluated on estimation data with respect to λ , chooses the 88th breakpoint which corresponds to using only five nonzero coefficients in the $A_p(q)$ polynomial and eight in $B_p(q)$. Unfortunately, this choice corresponds to using the model orders $n_a = 20$, $n_b = 8$ and $n_k = 0$. Thus, few nonzero parameters does not automatically imply a low model order.

The NNG method (the second column of Figure 4.3) behaves similarly as the lasso and the MDL choice is the 39th breakpoint. This corresponds to $\lambda = 0.2311$ which results in the model orders $n_a = 17$, $n_b = 3$ and $n_k = 1$. Thus, the NNG detects that there is a possible time delay in the system, but fails to yield a lower order model.

The example above shows that the statistical definition of model complexity does not take the order of dynamical systems into account, that is, the regularization methods presented in Section 2.3 only minimizes the number of parameters used and does not consider the

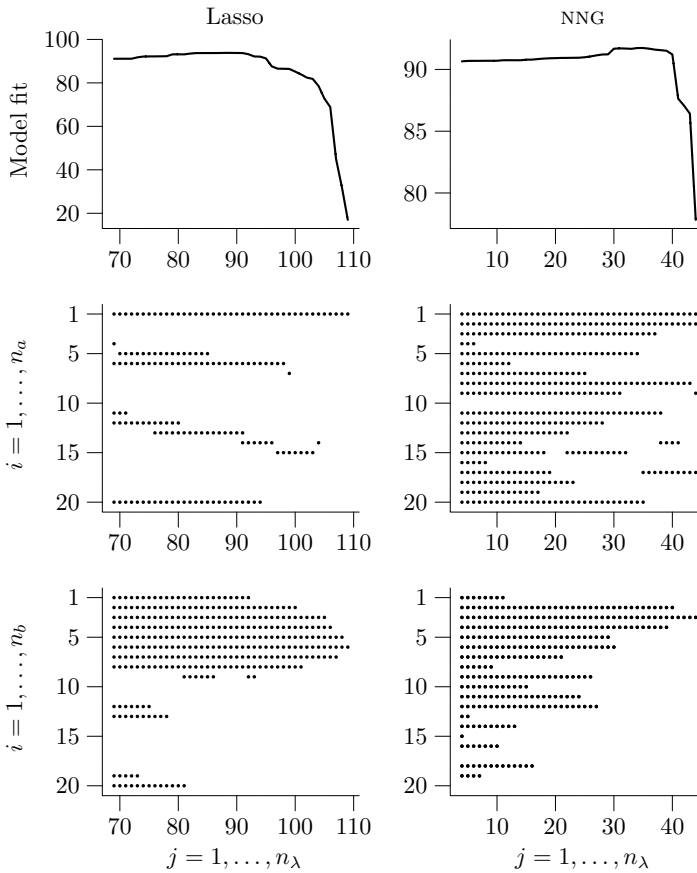


Figure 4.3: The lasso and the NNG solution paths for the fan and plate data, respectively. The first row shows the values of the model fit for the corresponding methods evaluated on validation data with the indices of the breakpoints on the horizontal axis. The second row illustrates which coefficients of the $A_p(q)$ polynomial that are nonzero and similarly for $B_p(q)$ in the last row.

order of the time lags. To take such structural information into account one needs to incorporate an ordering of the time lags into the regularization problem. For the lasso method, this is difficult to achieve since one works directly with the absolute values of the regressors. The NNG allows ordering of the regressors according to their importance instead of their absolute values which is appealing. The possibility to incorporate structural information into the NNG for linear regression models was also mentioned in Yuan et al. (2007), but no solution was given.

The chapter is organized as follows: First we start by formulating the NNG as an optimization problem where general linear inequality constraints can be handled. Then a modification of the NNG is proposed to handle the ordering of ARX models via the time lag. The proposed method is evaluated on two different sets of data and the result

is compared to that of an exhaustive search among all possible regressors. This follows with a generalization of the proposed method for order selection of the LPV-ARX model structure.

4.2 Problem Formulation

The main focus in this chapter is to incorporate structural information into the regressor selection problem of linear regression models

$$y(t) = \varphi(t)^T \theta + e(t). \quad (4.3)$$

Here, the regression vector $\varphi(t)$ is a vector valued, possibly nonlinear, function of old input and output data. This description contains a wide class of model structures, for instance, the ARX and the LPV-ARX model structures presented in Chapter 3. When selecting regressors for dynamical systems, there may be a natural or desired ordering of the regressors, for example, in the ARX case the regressors are ordered via their time lag.

Let us now turn our attention to the formulation of the NNG problem

$$\begin{aligned} & \underset{w \in \mathbb{R}^{n_\theta}}{\text{minimize}} && \frac{1}{N} \sum_{t=1}^N \left(y(t) - (\varphi(t) \odot \hat{\theta}_N^{\text{LS}})^T w \right)^2 + \frac{\lambda}{N} \sum_{i=1}^{n_\theta} w_i \\ & \text{subject to} && w \succeq 0, \end{aligned} \quad (4.4)$$

where \odot denotes element-wise multiplication and $\hat{\theta}_N^{\text{LS}}$ is the least-squares estimate of the parameters. By stacking the output $y(t)$ and the regressors $\varphi(t)^T$ into matrices Y and Φ , respectively, one can rewrite the NNG (4.4) as

$$\begin{aligned} & \underset{w \in \mathbb{R}^{n_\theta}}{\text{minimize}} && \|Y - \Phi \hat{\Theta}_N^{\text{LS}} w\|_2^2 + \lambda \mathbf{1}^T w \\ & \text{subject to} && w \succeq 0, \end{aligned} \quad (4.5)$$

where the scaling factor $1/N$ has been removed and we have introduced

$$\hat{\Theta}_N^{\text{LS}} \triangleq \text{diag}(\hat{\theta}_N^{\text{LS}}).$$

To incorporate structural orderings of the regressors, we exchange the simple bound constraints $w \succeq 0$ in (4.5) for the more general linear inequality constraint $Aw \preceq b$, which may, of course, contain simple bounds. Expanding the matrix norm in (4.5) yields

$$\begin{aligned} & \underset{w \in \mathbb{R}^{n_\theta}}{\text{minimize}} && \frac{1}{2} w^T Q w + f^T w + \lambda \mathbf{1}^T w \\ & \text{subject to} && A w \preceq b, \end{aligned} \quad (4.6)$$

where $Q = 2\hat{\Theta}_N^{\text{LS}} \Phi^T \Phi \hat{\Theta}_N^{\text{LS}}$ and $f = -2\hat{\Theta}_N^{\text{LS}} \Phi^T Y$. The modified formulation (4.6) of the original NNG method (4.4) using general linear inequality constraints will in the following be denoted by MNNG. The problem (4.6) belongs to the special class of optimization problems referred to as *parametric quadratic programs* (PQPs) (see, for instance, Gould, 1991). These problems have a piecewise affine solution path, that is, the solution $w(\lambda)$ to (4.6) is a piecewise affine function of the parameter λ . This enables efficient methods for finding the complete solution path and an algorithm for the strictly convex case, when Q is positive definite with full rank, is given in Appendix 4.A.

4.3 Applications to ARX Models

As previously mentioned, the regressors in the ARX model structure (3.8) are naturally ordered by their time lag. The original NNG method (4.4) does not take such orderings into consideration. It just sets the weights of the less important regressors low, not considering their order. To be able to penalize higher order lags first, we will use the more general formulation (4.6) by adding constraints on the weights. For ARX models, these constraints are chosen as

$$1 \geq w_1 \geq w_2 \geq \dots \geq w_{n_a} \geq 0 \quad (4.7a)$$

$$1 \geq w_{n_a+1} \geq w_{n_a+2} \geq \dots \geq w_{n_a+n_b} \geq 0. \quad (4.7b)$$

In this presentation we have chosen to include the upper bound that all weights should be less than or equal to one. There is no real theoretical motivation for this choice and the effect of this upper bound is small in the examples that follows. This inclusion of the constraints (4.7) into the MNNG problem (4.6) provides a natural extension of the NNG method for order selection of ARX models in system identification, which ensures ordering of time lags. Note that the ordering (4.7) lets the selection algorithm choose independently between using old inputs or old outputs to describe the data. In other words, the ordering (4.7) lets the coefficients in the zero polynomial and the pole polynomial be chosen independently. This yields an automatic order selection and a natural way to choose the importance between input lag and output lag.

Let us first evaluate the proposed method on a simple example where the system has an ARX structure.

Example 4.2: (ARX system)

As a first evaluation of the proposed modification of the NNG regressor selection method, let the system have the ARX structure

$$A_p(q)y(t) = B_p(q)u(t) + e(t), \quad (4.8a)$$

where $e(t)$ is zero mean white Gaussian noise with unit variance and

$$A_p(q) = 1 - 1.25q^{-1} + 0.4375q^{-2} - 0.3594q^{-3} + 0.1719q^{-4} + 0.3125q^{-5} \\ - 0.2764q^{-6} + 0.1360q^{-7} - 0.0769q^{-8} + 0.0137q^{-9}, \quad (4.8b)$$

$$B_p(q) = 1 + 0.25q^{-1} - 0.25q^{-2}. \quad (4.8c)$$

Thus, the true model order is given by $n_a = 9$, $n_b = 3$ and $n_k = 0$. For data collection, the system (4.8) is simulated using a white Gaussian noise input $u(t)$ with zero mean and unit variance. The data is then split into two separate sets, each of length $N = 1000$, one for estimation and one for validation.

In the *system identification toolbox* for MATLAB there are functions for the estimation of the order of an ARX model, namely `struc`, `arxstruc` and `selstruc` (see Ljung, 2009). These commands implement an exhaustive search among all specified model orders. Here we let n_a and n_b be chosen independently from the set $\{1, 2, \dots, 20\}$ where the time delay $n_k = 0$ has been fixed. The result, where the model orders have been chosen by the minimum value of the cost function on the validation dataset, is given in the

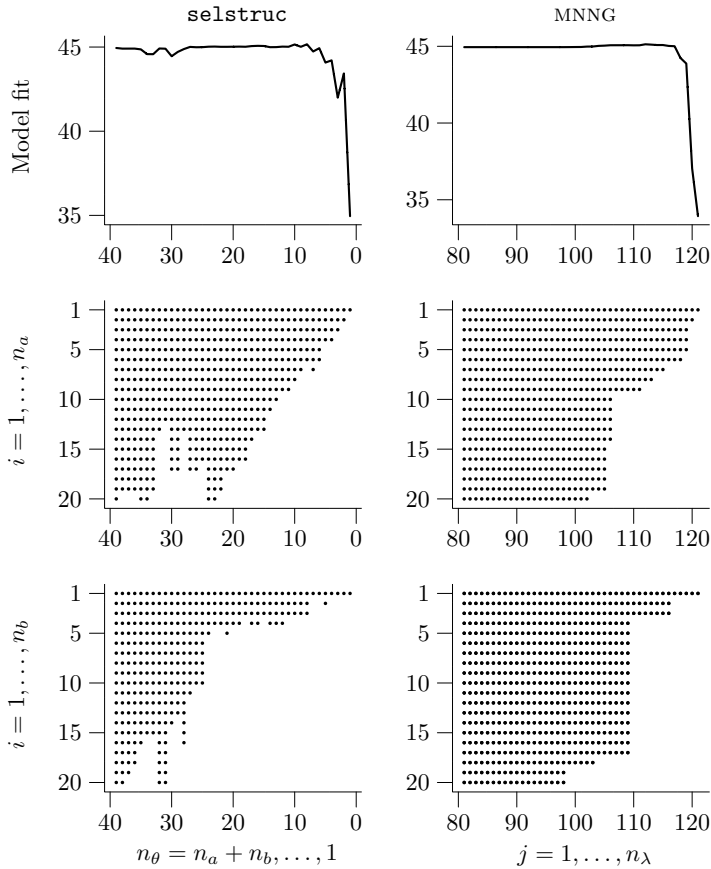


Figure 4.4: The resulting solution paths using the exhaustive search implemented in the system identification toolbox for MATLAB and the modified NNG method for data generated by the ARX system (4.8). The first row shows the model fit values evaluated on validation data for the corresponding method. The second row illustrates which coefficients of the $A_p(q)$ polynomial that are nonzero and similarly for $B_p(q)$ in the last row.

left column of Figure 4.4. The first row illustrates the model fit values (3.82) evaluated on validation data with the different model orders depicted on the horizontal axis. The middle row shows which coefficients that are nonzero in the pole polynomial $A_p(q)$ and similarly on the last row for the zero polynomial $B_p(q)$. The MDL choice (2.17) of model order for the exhaustive search implementation is given by $n_a = 8$ and $n_b = 3$ when evaluated on estimation data.

Now, for the MNNG method, the initial least-squares parameter estimate $\hat{\theta}_N^{LS}$ is found by solving (2.2) for the order $n_a = 20$, $n_b = 20$, and $n_k = 0$. Due to the presence of noise, all estimated parameter values will be nonzero. Solving (4.6) for all $\lambda \geq 0$ under the proposed constraints (4.7) and determining the model fit for validation data

results in the right column of Figure 4.4. Here, the horizontal axis depicts the indices of the breakpoints where the local affine representation of the piecewise solution path as a function of λ changes. We notice that as λ increases, more and more parameter values become exactly zero, until a certain point where only one parameter in the zero polynomial $B_p(q)$ remains. The MDL choice (2.17) yields, for this particular realization of input and noise signals, the same model order ($n_a = 8, n_b = 3$) as the exhaustive search implementation.

In the example above we saw that the proposed modification of the NNG performed equally well when searching for the order of an ARX model as the exhaustive search implemented in the system identification toolbox for MATLAB. Neither of the two methods proposed the true model order, even though $n_a = 9$ and $n_b = 3$ clearly exist as an element of the solution path in both methods. This implies that the last parameter of the pole polynomial $A_p(q)$ does not yield any significant contribution to the model fit and may thus be disregarded.

Now, let us return to the fan and plate process used in the introductory example.

Example 4.3: (Example 4.1 revisited)

Consider once again the fan and plate system in Example 4.1. Using once again the exhaustive search implementation, letting n_a and n_b be chosen independently from the set $\{1, 2, \dots, 20\}$ where $n_k = 1$ has been fixed, results in the left column of Figure 4.5. As before, the model orders have been chosen by the minimum value of the cost function determined on validation data. The horizontal axis depicts the total number of parameters used in the corresponding model. The maximal model fit (3.82) is seen to correspond to choosing $n_a = 4$ and $n_b = 2$, which coincides with the MDL choice (2.17) when evaluated on estimation data.

Now, let us consider using the NNG (4.4) with the proposed linear inequality constraints (4.7). Solving (4.6) for all possible $\lambda \geq 0$ yields the solution path illustrated in the right column of Figure 4.5. The model fit values have been determined for validation data with the indices of the breakpoints of the piecewise affine solution path on horizontal axis.

Comparing this result with the ordinary NNG solution given in the right column of Figure 4.3, one notices that the sparsity pattern of the coefficients has changed. The gaps in the coefficient list of the ordinary NNG has disappeared at the cost of using more parameters in the MNNG method. Even though the number of parameters has increased, the model order has decreased. Here, the maximal model fit value for the MNNG method corresponds to an ARX model with $n_a = 9$ and $n_b = 4$ which coincides with the MDL choice. Thus, we have managed to decrease the model order chosen by the NNG method by adding the linear inequality constraints (4.7) on the weights w when solving (4.6). Unfortunately, this choice uses twice as many parameters than what is really needed (compare with the result using an exhaustive search).

The example above indicates that the proposed modification of the NNG may decrease the model order chosen by the method. One drawback is that the chosen model order, at least in this example, is much higher than really needed. On the plus side is that the proposed method scales much better than the exhaustive search when the number of

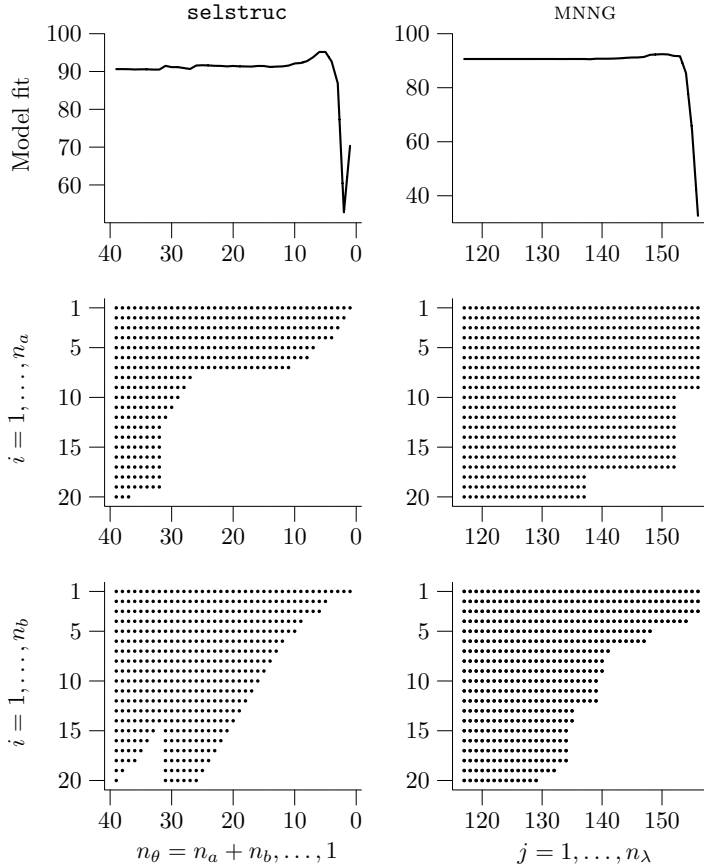


Figure 4.5: The resulting solution paths using the exhaustive search implemented in the system identification toolbox for MATLAB and the MNNG method for the fan and plate data in Example 4.1. The first row shows the model fit values for the corresponding methods evaluated on validation data. The second row illustrates which coefficients of the $A_p(q)$ polynomial that are nonzero and similarly for $B_p(q)$ in the last row.

parameters increases. The computational complexity of the MNNG method increases only linearly as the number of parameters increases while the complexity of the exhaustive search increases quadratically. This should yield a significant computational advantage for the NNG compared to the exhaustive search for high initial model orders or when multiple inputs and outputs are present.

Remark 4.1. It has recently been brought to the author's attention that the NNG can be seen as a special case of the *adaptive lasso* (see, for instance, Zou, 2006) for which consistency and other desirable statistical properties have been provided. What implications this has for the modifications proposed for dynamical systems is not clear and it remains as a topic for future research.

4.4 Applications to LPV-ARX Models

Let us consider using the MNNG for order selection of LPV-ARX models, see Section 3.2.2. This problem adds an extra dimension to the selection problem, compared to the ordinary ARX case, since it is also important to select which basis functions that should be included in the model. The problem of choosing which basis functions to incorporate into the model is referred to as *structure* selection.

The LPV-ARX model structure is defined in the SISO case as

$$\mathcal{A}(p, q)y(t) = \mathcal{B}(p, q)u(t) + e(t), \quad (4.9)$$

where

$$\begin{aligned} \mathcal{A}(p, q) &= 1 + a_1(p)q^{-1} + \dots + a_{n_a}(p)q^{-n_a}, \\ \mathcal{B}(p, q) &= b_1(p)q^{-n_k} + \dots + b_{n_b}(p)q^{-n_k - n_b + 1}, \end{aligned}$$

and the coefficient functions $a_i, b_j : \mathbb{P} \rightarrow \mathbb{R}$ have a static dependence on the measured variable p . Introduce

$$(\phi_1(p) \quad \dots \quad \phi_{n_g}(p)) \triangleq (a_1(p) \quad \dots \quad a_{n_a}(p) \quad b_1(p) \quad \dots \quad b_{n_b}(p)),$$

with $n_g \triangleq n_a + n_b$. Here, we assume that each of the functions ϕ_i may be linearly parametrized as

$$\phi_i(p) = \theta_{i0} + \sum_{j=1}^{s_i} \theta_{ij} \psi_{ij}(p),$$

where $(\theta_{ij})_{i=1, j=0}^{n_g, s_i}$ are unknown parameters and $(\psi_{ij})_{i=1, j=1}^{n_g, s_i}$ are basis functions. The LPV-ARX model structure can be written as a linear regression model (4.2) with

$$\begin{aligned} \varphi(t) \triangleq & \left(-y(t-1) \quad -\psi_{11}(p(t))y(t-1) \quad \dots \quad -\psi_{1s_1}(p(t))y(t-1) \right. \\ & \left. -y(t-2) \quad \dots \quad -\psi_{n_a s_{n_a}}(p(t))y(t-n_a) \quad u(t) \quad \dots \right)^T, \quad (4.10) \\ \theta \triangleq & (\theta_{1,0} \quad \dots \quad \theta_{1,s_1} \quad \theta_{2,0} \quad \dots \quad \theta_{n_g, s_{n_g}})^T. \end{aligned}$$

From the structure of the linear regression model above, one notices that the regressors are *grouped*. That is, each time shifted signal $x(t-i)$, where $x(t)$ denotes either input $u(t)$ or output $y(t)$, is involved with $s_i + 1$ regressors, where s_i is the number of basis functions ψ_{ij} , $j = 1, \dots, s_i$, within the group. The groups correspond to the scheduling variable dependent coefficients $a_i(p(t))$ and $b_j(p(t))$ of the polynomials $\mathcal{A}(p, q)$ and $\mathcal{B}(p, q)$, respectively.

As in the ARX case, we want to use as few time lags of the input $u(t)$ and the output $y(t)$ as possible. This can be accomplished by adding the constraints

$$\begin{aligned} \sum_{j=0}^{s_1} w_{1j} &\geq \sum_{j=0}^{s_2} w_{2j} \geq \dots \geq \sum_{j=0}^{s_{n_a}} w_{n_a j}, \\ \sum_{j=0}^{s_{n_a+1}} w_{(n_a+1)j} &\geq \dots \geq \sum_{j=0}^{s_{n_g}} w_{n_g j}, \end{aligned} \quad (4.11)$$

to the NNG method (4.4). This choice is similar to the constraints on the weights in the ARX case, see (4.7), but now with grouped variables. There are several possible choices of constraints, where (4.11) is only one of them. Another possible constraint is that the minimum value of the weights in one group should be greater than the maximum value of the following. This choice is more restrictive than (4.11) and probably not suitable in this application. The effect of different choices is not known and is a subject for future research.

Now, let us evaluate the proposed method in some simulations. Unfortunately, it seems to be difficult to find data from real-life LPV systems and the evaluation has to be done on simulated data.

Example 4.4: (LPV-SS system)

As a first evaluation of the proposed modification of the NNG regressor selection method, let the system be given by the LPV-SS model

$$\begin{aligned} x(t+1) &= \begin{pmatrix} 0 & p(t) \\ 1 & p(t) \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u(t) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} e(t), \\ y(t) &= \begin{pmatrix} 1 & 0 \end{pmatrix} x(t). \end{aligned} \quad (4.12)$$

Here, we assume that the scheduling variable $p(t)$ is contained within the closed interval $[-0.4, 0.4]$ and that $e(t)$ is white Gaussian noise with zero mean and unit variance. The *frozen* poles of the system, when letting p vary within the interval $[-0.4, 0.4]$ are given in the upper left corner of Figure 4.6. Using the method for transforming an LPV-SS model to an equivalent LPV-ARX model described in Tóth (2008), the model (4.12) may be rewritten as

$$y(t) - p(t-1)y(t-1) - p(t-1)y(t-2) = u(t-1) + e(t-1). \quad (4.13)$$

Thus, the true model order is given by $n_a = 2$, $n_b = 1$ and $n_k = 1$.

For data collection, the system (4.12) is simulated using a white Gaussian noise input $u(t)$ with zero mean and unit variance and a uniformly distributed scheduling variable $p(t)$ in the interval $[-0.4, 0.4]$. The data is then split into two separate sets, each of length $N = 5000$, one for estimation and one for validation. In the initial least-squares estimate, we let $n_a = 5$, $n_b = 5$ and the basis functions be given by

$$\psi_{i1}(p) = p(t), \quad \psi_{i2}(p) = p(t-1), \quad \psi_{i3}(p) = p(t-2), \quad \psi_{i4}(p) = p(t-3).$$

The result of running the NNG with the additional constraints (4.7) is shown in Figure 4.6 where the model fit in the upper right corner has been determined on validation data. The two figures in the last row of Figure 4.6 show which parameters θ in (4.10) corresponding to the coefficients in the $\mathcal{A}(p, q)$ and the $\mathcal{B}(p, q)$ polynomials that are nonzero, respectively. The estimated coefficients of the $\mathcal{A}(p, q)$ and the $\mathcal{B}(p, q)$ polynomials for the maximal model fit corresponds to the 83rd breakpoint and are given by

$$\begin{aligned} a_1(p(t)) &= -0.9999p(t-1), & a_2(p(t)) &= -0.9385p(t-1), \\ b_0(p(t)) &= -0.0099 - 0.0435p(t-1), & b_1(p(t)) &= 0.9932 - 0.0001p(t-2), \end{aligned}$$

and $b_2(p(t)) = -0.0642p(t-3)$, where the remaining coefficients are zero. This is quite close to the system (4.13). The exceptions are that the polynomials b_0 , b_2 and the

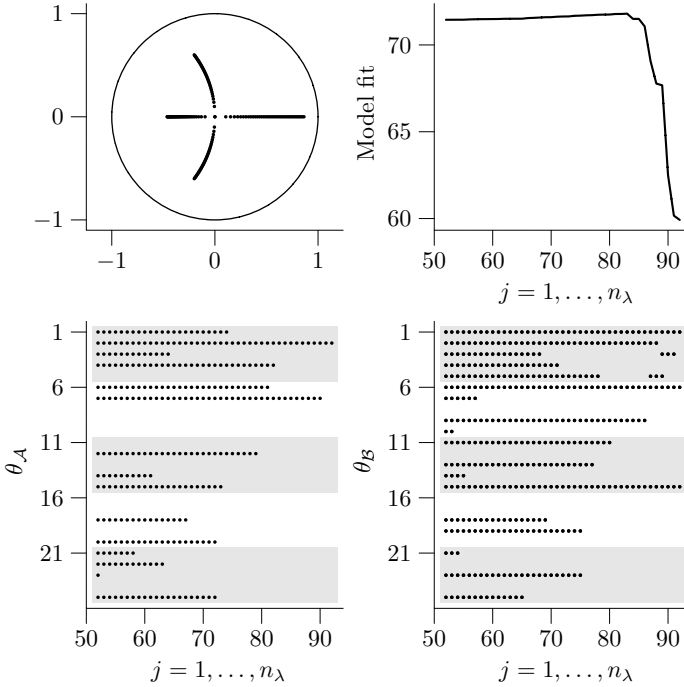


Figure 4.6: The result of the MNNG for the LPV-SS system given by (4.12). The upper left corner depicts the frozen poles of the system (4.12) when the scheduling variable $p(t) \in [-0.4, 0.4]$ varies. In the upper right corner the model fit for the estimated models evaluated on validation are shown. The lower left corner illustrates which coefficients of the $A_p(q)$ polynomial that are nonzero and similarly for $B_p(q)$ in the lower right corner.

coefficient in the b_1 polynomial corresponding to the basis function $p(t - 2)$ should be zero. This is an effect of the choice of ordering (4.11) of the weights of the regressors which inflicts that the sum of the weights for the b_0 coefficient should be greater than the sum for b_1 . Since b_1 plays an important role in the system (4.13), the sum of the weights for this coefficient must be large which in turn forces the sum of the weights for the b_0 coefficient to be large.

The example above indicates that the MNNG method might be useful for finding the order of the system and also the structural dependence, that is, which basis functions to use in the model. It should be noted that the good results in the example are due to that the system lies in the model set and the fact that the noise level is quite low compared to the strength of the output signal.

Now, let us evaluate the proposed method on a more complex problem where we make use of nonlinear basis functions.

Example 4.5: (LPV-ARX system)

As a second evaluation of the proposed modification of the NNG regressor selection method, let the system be given by the LPV-ARX model

$$\mathcal{A}(q, p)y = \mathcal{B}(q, p)u + e, \quad (4.14a)$$

where

$$\begin{aligned} \mathcal{A}(q, p) &= 1 + (0.24 + 0.1p)q^{-1} - (0.1\sqrt{-p} - 0.6)q^{-2} + 0.3 \sin(p)q^{-3} \\ &\quad + (0.17 + 0.1p)q^{-4} + 0.3 \cos(p)q^{-5} - 0.27q^{-6} \\ &\quad + (0.01p)q^{-7} - 0.07q^{-8} + 0.01 \cos(p)q^{-9}, \\ \mathcal{B}(q, p) &= 1 + (1.25 - p)q^{-1} - (0.2 + \sqrt{-p})q^{-2}. \end{aligned} \quad (4.14b)$$

Here, we assume that the scheduling variable $p(t)$ is contained within the interval $[-2\pi, 0]$ and that $e(t)$ is white Gaussian noise with zero mean and variance 0.1. Thus, the true model order is given by $n_a = 9$, $n_b = 3$ and $n_k = 0$. The frozen poles of the system, when letting p vary in the interval $[-2\pi, 0]$ are given in the upper left corner of Figure 4.7.

For data collection, the system (4.14) is simulated using a white Gaussian noise input $u(t)$ with zero mean and unit variance and a uniformly distributed scheduling variable $p(t)$ in the interval $[-2\pi, 0]$. The data is then split into two separate sets, each of length $N = 5000$, one for estimation and one for validation. In the initial least-squares estimate, we let $n_a = 9$, $n_b = 3$ and the basis functions be given by

$$\psi_{i1}(p) = p, \quad \psi_{i2}(p) = \sqrt{-p}, \quad \psi_{i3}(p) = \sin(p), \quad \psi_{i4}(p) = \cos(p).$$

Thus, the system is in the model set. The result of running the NNG with the additional constraints (4.7) is shown in Figure 4.7 where the model fit in the upper right corner has been determined on validation data. The two columns in the last row of Figure 4.7 show which parameters θ in (4.10) corresponding to the coefficients in the $\mathcal{A}(p, q)$ and the $\mathcal{B}(p, q)$ polynomials that are nonzero, respectively. The estimated coefficients for the maximal model fit and the one suggested by MDL contains far too many nonzero parameter values to be presented here. Instead, let us consider the 145th breakpoint where the model fit starts to decrease significantly. The parameter values of the $\mathcal{A}(p, q)$ polynomial are then given by

$$\begin{aligned} a_1(p) &= 0.2107 + 0.0807p, & a_2(p) &= 0.5501 - 0.0683\sqrt{-p}, \\ a_3(p) &= 0.0083\sqrt{-p} + 0.3045 \sin p, & a_4(p) &= 0.1153 + 0.0837p, \\ a_5(p) &= 0.2955 \cos p, & a_6(p) &= -0.2643, \\ a_7(p) &= 0.0026p - 0.0048\sqrt{-p}, & a_8(p) &= -0.0562, \end{aligned}$$

where the remaining coefficients are zero. This is close to the system (4.14) with the exceptions of some small parameter values that should not be there (for example the coefficient in the $a_3(p)$ polynomial corresponding to the basis function $\sqrt{-p}$) and the lack of a_9 . The latter is due to that it lies at the end of (4.11), and is therefore penalized

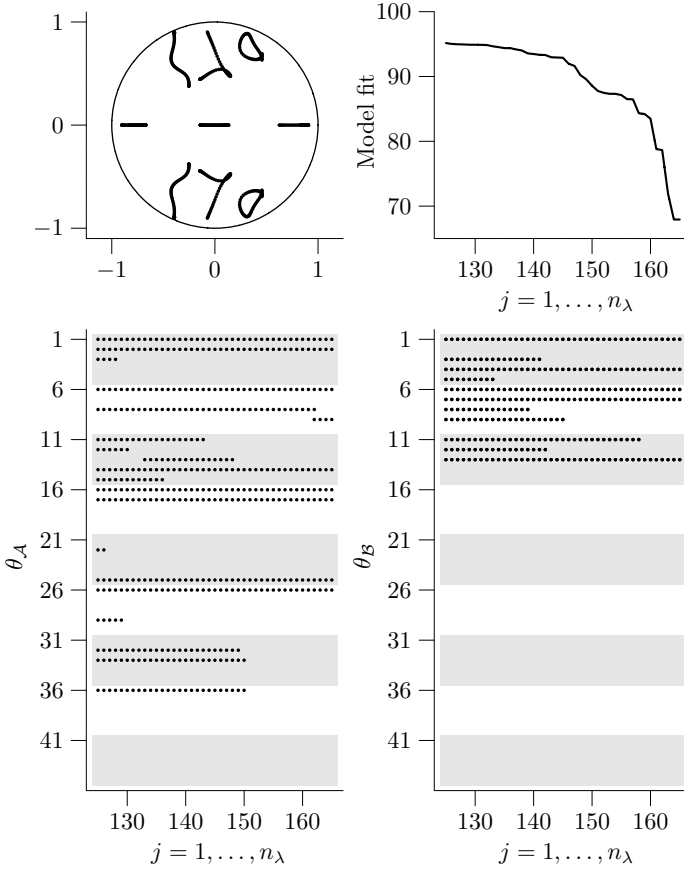


Figure 4.7: The result of the MNNG for the LPV-SS system given by (4.14). The upper left corner depicts the frozen poles of the system (4.14) when the scheduling variable $p(t) \in [-2\pi, 0]$ varies. In the upper right corner the model fit for the estimated models evaluated on validation data are shown. The lower left corner illustrates which coefficients of the $A_p(q)$ polynomial that are nonzero and similarly for $B_p(q)$ in the lower right corner.

more than the others. The corresponding estimate of the $\mathcal{B}(p, q)$ polynomial is also quite close to the system and is given by

$$\begin{aligned} b_0(p) &= 0.9873 - 0.0024 \sin p, \\ b_1(p) &= 1.2309 - 1.0136p + 0.0001 \sin p, \\ b_2(p) &= -0.3319 - 0.7094\sqrt{-p}. \end{aligned}$$

Here we notice that certain parameters are present that should not be there, for example the parameter in the $b_0(p)$ polynomial corresponding to $\sin p$.

The two examples for the LPV-ARX case presented above indicate that the proposed mod-

ification of the NNG may be used successfully when determining which basis functions that should be used to describe the data. Also, the example suggests that the choice of ordering (4.11) might not be optimal since it does not find the system even in such simple examples. Finding a better ordering instead of (4.11) remains as future work.

4.5 Discussion

In this chapter a method for order and structural dependence selection of ARX and LPV-ARX models was introduced. The method is a modified variant of the NNG method introduced by Breiman (1995), where constraints on the weights are added according to the natural ordering of the regressors in ARX models. Different examples were given, both on data from simulations and from a real-life system, and the results look promising. The choice of ordering in the LPV-ARX is still an open question and further analysis of the effects of different choices should be analyzed.

The regularization parameter λ in (4.6) was used to penalize both the pole and the zero polynomials, thus letting the ordering of the output lags and the input lags be independent. Another possibility is to make use of two penalizing parameters λ_y and λ_u , one for weights on the coefficients of the pole polynomial and one for the weights on the coefficients of the zero polynomial. This idea would need to make use of the methods in *multi-parametric quadratic programming* and although they are not as simple as the one presented for a single parameter, efficient algorithms exist (see, for instance, Tøndel et al., 2003). The proposed method is easily extended to the multivariable case. Here, one could still use one regularization parameter λ and it would be interesting to observe the behavior of the proposed method in this context.

The order and structure selection problem of LPV-ARX models is a special case of the related NARX problem and the possibility of extending the ideas to the NARX case should be considered in future work. Furthermore, it would also be interesting to include the *instrumental variables* framework into the regression selection algorithms presented in this chapter. This would enable automatic order selection of, for example, output error models.

Finally, the classical NNG method as presented by Breiman (1995) has a close relationship with the adaptive lasso (see, for example, Zou, 2006) for which some desirable statistical properties have been proved. The consequences for the MNNG approach proposed here is not clear and should be investigated.

Appendix

4.A Parametric Quadratic Programming

The nature of the PQP problem is quite similar to the ordinary QP problem, for which the *active-set method* is a common choice (see, for instance, Nocedal and Wright, 2006). Thus, it is only natural to incorporate these ideas into the PQP algorithms as much as possible. In this section, we will derive the formulas for the PQP which is a special case of the more general setting presented in Roll (2008). The resulting algorithm can also be found in Gould (1991) and Stein et al. (2008). A different perspective of the solution to the PQP problem is given in Romanko (2004). From here on, we will assume convexity, that is, the matrix Q is nonsingular and positive definite.

The algorithm derived below is based on the fact that the solution to (4.6) is a continuous piecewise affine function of λ with a finite number of breakpoints (see, for example, Roll, 2008). Thus, a simple procedure to find the solution path $w(\lambda)$ for all $\lambda \geq 0$, is to first find the initial solution $w(0)$. Then one determines the direction for which the solution decreases the most as a function of λ and continues in this direction until one hits a constraint. The solution is then updated and a new direction is found. The procedure is repeated until λ reaches infinity. This simple idea is the foundation of the algorithm derived below.

The Lagrangian (Boyd and Vandenberghe, 2004) associated with (4.6) is

$$\mathcal{L}(w, \mu) = \frac{1}{2}w^T Qw + f^T w + \lambda \mathbf{1}^T w + \mu^T (Aw - b), \quad (4.15)$$

where μ is a vector of Lagrangian multipliers. This yields the Karush-Kuhn-Tucker (KKT) conditions (Boyd and Vandenberghe, 2004):

$$Qw + f + \lambda \mathbf{1} + A^T \mu = 0 \quad (4.16a)$$

$$Aw - b \preceq 0 \quad (4.16b)$$

$$\mu_i (A_i w - b_i) = 0, \quad i \in \mathcal{I} \quad (4.16c)$$

$$\mu \succeq 0, \quad \lambda \geq 0, \quad (4.16d)$$

where \mathcal{I} is the index set of the inequality constraints. Now, let $\mathcal{W} \subset \mathcal{I}$ denote the set of

*active*¹ constraints. Solving (4.16) is then equivalent to solving

$$\begin{pmatrix} Q & A_{\mathcal{W}}^T \\ A_{\mathcal{W}} & 0 \end{pmatrix} \begin{pmatrix} w \\ \mu_{\mathcal{W}} \end{pmatrix} = \begin{pmatrix} -f \\ b_{\mathcal{W}} \end{pmatrix} + \lambda \begin{pmatrix} -\mathbf{1} \\ 0 \end{pmatrix}. \quad (4.17)$$

Until now, no consideration of the effect of the parameter λ has been taken. Differentiation with respect to λ yields

$$\begin{pmatrix} Q & A_{\mathcal{W}}^T \\ A_{\mathcal{W}} & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial w}{\partial \lambda} \\ \frac{\partial \mu_{\mathcal{W}}}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} -\mathbf{1} \\ 0 \end{pmatrix}. \quad (4.18)$$

The question that remains to be answered is how to use (4.17) and (4.18) to get an efficient algorithm. Remembering the initial discussion, the initial solution is found by solving (4.17) for $\lambda = 0$, where the active set \mathcal{W} is found, if not apparent, by a Phase I linear program (Nocedal and Wright, 2006). The search direction is clearly found via (4.18), so the final detail is to find the step length $\delta\lambda \geq 0$. There are, in principal, two different ways of hitting a constraint. Firstly, the updated w should be feasible, that is,

$$A_i \left(w + \frac{\partial w}{\partial \lambda} \delta\lambda \right) = b_i, \quad i \in \mathcal{I}, \quad (4.19)$$

which is always the case for $i \in \mathcal{W}$. Thus, we should find the smallest $\delta\lambda \geq 0$ satisfying (4.19) for some $i \notin \mathcal{W}$. Furthermore, since (4.19) can be rewritten as

$$A_i \frac{\partial w}{\partial \lambda} \delta\lambda = b_i - A_i w \geq 0,$$

it must additionally hold that $A_i \frac{\partial w}{\partial \lambda} > 0$. Secondly, there might be a constraint that becomes inactive. This can only happen when there is an $i \in \mathcal{W}$ such that $\frac{\partial \mu_i}{\partial \lambda} < 0$. In this case, one should find the smallest $\delta\lambda \geq 0$ satisfying

$$\mu_i + \frac{\partial \mu_i}{\partial \lambda} \delta\lambda = 0, \quad (4.20)$$

for some $i \in \mathcal{W}$. Finally, the step length is given by the smallest $\delta\lambda \geq 0$ such that both (4.19) and (4.20) are fulfilled. The results are gathered in Algorithm 3.

Algorithm 3 Parametric Quadratic Programming

Given: A convex parametric quadratic program (4.6).

- 1) *Initialization:* Let $\lambda = 0$ and find the solution to (4.6) via a Phase I linear programming. Set $\mathcal{W} = \mathcal{W}_0$, $w = w_0$ and $\mu_{\mathcal{W}} = \mu_{\mathcal{W}_0}$ where \mathcal{W}_0 , w_0 and $\mu_{\mathcal{W}_0}$ are the results from the linear program. Finally, let $S = \{(\lambda, w)\} = \{(0, w_0)\}$.
- 2) *Directions:* Solve (4.18) for $\frac{\partial w}{\partial \lambda}$ and $\frac{\partial \mu_{\mathcal{W}}}{\partial \lambda}$.
- 3) *Step length:* Find the minimal $\delta\lambda \geq 0$ satisfying one of the following:

¹The correct notion of \mathcal{W} is the *working* set, which in general is a subset of the active set. The primary function of the working set, instead of using the full active set, is to avoid *primal degeneracy* in the implementation of the algorithm, see Nocedal and Wright (2006) for details.

- i) If $A_i(w + \frac{\partial w}{\partial \lambda} \delta \lambda) = b_i$ and $A_i \frac{\partial w}{\partial \lambda} > 0$ for some $i \notin \mathcal{W}$, then $\mathcal{W} \leftarrow \mathcal{W} \cup \{i\}$.
- ii) If $\mu_i + \frac{\partial \mu_i}{\partial \lambda} \delta \lambda = 0$ and $\frac{\partial \mu_i}{\partial \lambda} < 0$ for some $i \in \mathcal{W}$, then $\mathcal{W} \leftarrow \mathcal{W} \setminus \{i\}$.

If no feasible solution $\delta \lambda \geq 0$ exists, set $\delta \lambda \leftarrow \infty$.

- 4) *Update:* Set $\lambda \leftarrow \lambda + \delta \lambda$, $w \leftarrow w + \frac{\partial w}{\partial \lambda} \delta \lambda$ and $\mu_{\mathcal{W}} \leftarrow \mu_{\mathcal{W}} + \frac{\partial \mu_{\mathcal{W}}}{\partial \lambda} \delta \lambda$. Add new pair $S \leftarrow \{S, (\lambda, w)\}$.
- 5) *Termination:* **Stop** if $\lambda = \infty$, else continue from step 2.

It is worth noting that the index set \mathcal{W} used in Algorithm 3, does not necessarily contain all active constraints. This becomes clear when several constraints become active at once. Then only one constraint is added to \mathcal{W} , while the active set contains them all. This technicality prohibits the addition of linear dependent constraints to the reduced KKT system (4.17), so that it always has full rank.

For the particular choice of ordering (4.7), the initial solution to (4.6) is easy to find. Let $\lambda = 0$, then (4.6) is an ordinary least square problem with inequality constraints and the obvious solution is given by $w = \mathbf{1}$. To find $\mu_{\mathcal{W}}$ we need to solve (4.16). Since

$$\begin{aligned} Qw + f &= 2\hat{\Theta}_N^{\text{LS}} \Phi^T \Phi \hat{\Theta}_N^{\text{LS}} \mathbf{1} - 2\hat{\Theta}_N^{\text{LS}} \Phi^T Y = 2\hat{\Theta}_N^{\text{LS}} \Phi^T (\Phi \hat{\Theta}_N^{\text{LS}} \mathbf{1} - Y) \\ &= 2\hat{\Theta}_N^{\text{LS}} \Phi^T (\Phi \hat{\theta}_N^{\text{LS}} - Y) = 0, \end{aligned} \quad (4.21)$$

where the last equality follows from the fact that the regression matrix Φ is orthogonal to the residuals $\Phi \hat{\theta}_N^{\text{LS}} - Y$, one finds that $\mu_{\mathcal{W}} = 0$.

5

Utilizing Structure Information in Subspace Identification

The prediction-error approach to parameter estimation of linear time-invariant models often involves solving a non-convex optimization problem (as discussed in Section 1.1). It may therefore be difficult to guarantee that the global optimum will be found. A common way to handle this problem is to find an initial estimate, hopefully lying in the region of attraction of the global optimum, using some other method. The prediction-error estimate can then be obtained by a local search starting at the initial estimate. In this chapter, a new approach for finding an initial estimate of polynomial linear models utilizing structure and the subspace method is presented. The polynomial models are first written on the observer canonical state-space form, where the specific structure is later utilized, rendering least-squares estimation problems with linear equality constraints.

5.1 Introduction

The estimation of linear time-invariant polynomial models (3.6), given a sequence of input and output pairs (3.1), is a classical problem in system identification. Several methods exist, where the PEM (Section 3.1) or the IV (Section 3.3) are common choices. Especially, the PEM methods have shown to have desirable statistical properties and have also been proved to work in practice. One drawback with the PEM is that the procedure typically involves solving a non-convex optimization problem (see Example 1.2). It is therefore important to have a good initial estimate of the parameters, from which the PEM estimate can be found by a local search. To this end, several different methods have been developed which, in one way or another, utilize the structure of the problem. For instance, the *system identification toolbox* (Ljung, 2009) in MATLAB uses a variant of the multi-step IV algorithm presented in Section 3.3 as the choice of initialization method for the PEM for a wide variety of linear model structures.

In the last three decades, the *subspace identification* (SID) methods (see Section 3.4) have been developed and become an important set of tools for estimating state-space mod-

els. These methods have been proved to yield reliable estimates in a numerically stable and efficient manner. In this chapter, we are going to consider how to incorporate certain problem specific structure information into the SID methods. Consider, for example, using a SID method as an initialization method when estimating the linear transfer-function model structures described in Section 3.2.1.

Example 5.1

Assume that one would like to find an OE model (3.11), given a data set (3.1), with two parameters in the numerator and three in the denominator

$$y(t) = \frac{b_1q^{-1} + b_2q^{-2}}{1 + a_1q^{-1} + a_2q^{-2} + a_3q^{-3}}u(t) + e(t). \quad (5.1)$$

Writing the above model in the *observer canonical form* (OCF) (3.19) yields

$$x(t+1) = \begin{pmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{pmatrix} x(t) + \begin{pmatrix} b_1 \\ b_2 \\ 0 \end{pmatrix} u(t), \quad (5.2a)$$

$$y(t) = (1 \ 0 \ 0) x(t) + e(t). \quad (5.2b)$$

Using a SID method will lead to a third order state-space model in some unknown basis. This implies that, by converting the resulting state-space model to the OCF, the third element of the $B(\theta)$ will in general not be zero. Thus, to use a SID method to find an initial estimate for the PEM, it would be desirable to incorporate the knowledge of the structure of the $B(\theta)$ matrix into the SID scheme.

Motivated by the example above, the goal of this chapter is to explore the possibilities of incorporating structure information into the SID scheme. A theoretical analysis of the use of SID methods to estimate (unstructured) ARMAX models is given in Bauer (2005, 2009).

Even though the main focus of this chapter is on the discrete-time problem formulation, the continuous-time case follows readily if one substitutes the use of a discrete-time SID method for a continuous-time equivalent, such as the ones presented in, for instance, McKelvey and Akçay (1994) and Van Overschee and De Moor (1996b).

The structure of this chapter is as follows: Firstly, we are going to consider the estimation of OE models, both in discrete time and continuous time. Then follows the estimation of discrete-time ARMAX models, for which the OE estimation is an intermediate step. Finally, the ideas are further developed to handle certain gray-box model structures.

5.2 OE Models

In this section and the following, the SID algorithm, described in Section 3.4, will be modified to handle the identification of general OE and ARMAX models. The method for OE models is described first, since it constitutes an intermediate step when estimating an ARMAX model. To avoid notational difficulties, only SISO systems are initially discussed.

5.2.1 Discrete Time

To illustrate the basic idea of using a SID method for estimating a discrete-time transfer-function OE model (3.11), let us return to the example in the introduction.

Example 5.2: (Example 5.1 revisited)

Assume that one wants to estimate an OE model (5.1) or its equivalent (5.2) using the SID method presented in Section 3.4.1. After the first three steps in Algorithm 2, estimates \hat{A} and \hat{C} of the $A(\theta)$ and $C(\theta)$ matrices, respectively, have been derived in some unknown basis. If these estimates could be transformed into the OCF (5.2), one would know that the last element of $B(\theta)$ should be zero. This can be achieved in several ways and probably the simplest way is to remove the column in the regression matrix, corresponding to the parameter that should be zero, before solving the least-squares problem (3.53). In this way, the remaining parameters are estimated as though this parameter truly was zero.

A different approach, yielding the same result, is to add a linear equality constraint to the least-squares problem (3.53) and solve it with, for instance, the method presented in Section 2.1. This method will be illustrated below since it is more general and remains valid if the known parameter is nonzero.

Thus, the question is how to transform the estimates \hat{A} and \hat{C} to the OCF without knowledge of the remaining matrices in the state-space equation. This is simply done by determining the characteristic equation of \hat{A} , which can be done with the following MATLAB pseudo-code:

$$\hat{a} = \text{poly}(\hat{A}), \quad \hat{A}_{\text{OCF}} = [-\hat{a}(2 : n_a + 1)^T, \text{eye}(n_a, n_a - 1)], \quad (5.3)$$

and $\hat{C}_{\text{OCF}} = \text{eye}(1, n_a)$, where n_a is the degree of the pole polynomial.

The next step is to find an estimate of the $B(\theta)$ matrix via the least-squares problem (3.53). Since the \hat{A} and \hat{C} matrices now are in the OCF (5.2), we know that the third element of the $B(\theta)$ matrix is zero. Thus, we add the linear equality constraint

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} B \\ D \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

when solving the least-squares problem (3.53) and a solution is attained using the method presented in Section 2.3 or with the `lsqlin` in the MATLAB optimization toolbox. Note that the same result is achieved by removing the column of the regression matrix corresponding to the parameter that should be zero before solving (3.53).

It is known that the transformation of a state-space model to these kind of canonical forms is not numerically sound, since the similarity transform may be badly conditioned. Here, we do not need to explicitly find the transformation matrix and its inverse, since the structure of the problem is so simple. Furthermore, even though the problem of finding the eigenvalues of a matrix is numerically ill-conditioned, using the eigenvalues to find the coefficients of the characteristic equation is within the round-off error of the matrix in question. Thus, the method used above to transform the estimated matrices to the OCF is numerically stable. This does not, of course, change the fact that the OCF is extremely sensitive to perturbations in the parameter values. For example, a small perturbation of the last coefficient a_{n_a} may yield a big change in the eigenvalues.

The method used in the example above is easily seen to be valid whenever $n_b \leq n_a$, where the only difference is the number of known zeros in the $B(\theta)$ and $D(\theta)$ matrices in the final step. But what if one needs a model with $n_b > n_a$?

Example 5.3: (OE model with $n_b > n_a$)

Consider an discrete-time OE model with $n_b = 5$, $n_a = 2$ and $n_k = 3$:

$$y(t) = \frac{b_1q^{-3} + b_2q^{-4} + b_3q^{-5} + b_4q^{-6} + b_5q^{-7}}{1 + a_1q^{-1} + a_2q^{-2}}u(t) + e(t).$$

Rewriting the above equation to OCF (3.19) with delayed input yields

$$x(t+1) = \begin{pmatrix} -a_1 & 1 & 0 & 0 & 0 \\ -a_2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} x(t) + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} u(t-2),$$

$$y(t) = (1 \ 0 \ 0 \ 0 \ 0) x(t) + e(t).$$

This means that if one wants to estimate this model, given data, using a SID method, then one needs to estimate a fifth order state-space model. Furthermore, one needs to be able to constrain some of the elements of $A(\theta)$ to be zero, that is, the coefficients of the characteristic polynomial. To the author's knowledge, no existing subspace method can handle such constraints and the estimates of the coefficients \hat{a}_i in the characteristic equation of $A(\theta)$ will in general all be nonzero.

A possible way to solve this problem would be to estimate the $A(\theta)$ matrix, transform it to OCF and just truncate the characteristic polynomial by setting unwanted coefficients to zero. This might work well in some cases, but is probably a bad idea in the case of undermodeling.

Here, we are going to propose a different solution by introducing *virtual inputs*. Let us rewrite the original equation by splitting the rational expression into two separate terms

$$\begin{aligned} y(t|t-1, \theta) &= \frac{b_1 + b_2q^{-1} + b_3q^{-2} + b_4q^{-3} + b_5q^{-4}}{1 + a_1q^{-1} + a_2q^{-2}}q^{-3}u(t) \\ &= \frac{b_1 + b_2q^{-1} + b_3q^{-2}}{1 + a_1q^{-1} + a_2q^{-2}}q^{-3}u(t) + \frac{b_4 + b_5q^{-1}}{1 + a_1q^{-1} + a_2q^{-2}}q^{-6}u(t). \end{aligned}$$

Now, polynomial division of the rational expressions yields

$$\begin{aligned} y(t|t-1, \theta) &= \left(b_1 + \frac{(b_2 - b_1a_1)q^{-1} + (b_3 - b_1a_2)q^{-2}}{1 + a_1q^{-1} + a_2q^{-2}} \right) u(t-3) \\ &\quad + \left(b_4 + \frac{(b_5 - b_4a_1)q^{-1} + (-b_4a_2)q^{-2}}{1 + a_1q^{-1} + a_2q^{-2}} \right) u(t-6), \end{aligned}$$

which can be written as

$$\begin{aligned} x(t+1) &= \begin{pmatrix} -a_1 & 1 \\ -a_2 & 0 \end{pmatrix} x(t) + \begin{pmatrix} b_2 - b_1a_1 & b_5 - b_4a_1 \\ b_3 - b_1a_2 & -b_4a_2 \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}, \\ y(t) &= (1 \ 0) x(t) + (b_1 \ b_4) \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} + e(t), \end{aligned} \tag{5.4}$$

where $u_1(t) \triangleq u(t-3)$ and $u_2(t) \triangleq u(t-6)$. Thus, the original model can now be estimated as a second order state-space model with two input signals instead of one. The constraints on the characteristic polynomial have been implicitly taken care of.

It is worth noting that other choices of virtual inputs are possible, for example, one could have chosen $u_1(t) \triangleq u(t-2)$, $u_2(t) \triangleq u(t-4)$ and $u_3(t) \triangleq u(t-6)$. In this way, the polynomial divisions performed above would not be necessary and the $D(\theta)$ matrix in the state-space model (5.4) would become zero. The latter choice of virtual inputs is probably preferable when $n_k > 0$, but the former choice, resulting in (5.4), is also valid when $n_k = 0$.

Hence, after this initial transformation, we can proceed as before. First, the estimates \hat{A} and \hat{C} are found and transformed into OCF by the same method as in Example 5.3. Now, since the estimate of a_2 is known, it is clear that the last element of $B(\theta)$ in (5.4) is related to the last element of $D(\theta)$. This relation may be enforced by adding the constraint

$$(0 \ 0 \ 0 \ 1 \ 0 \ -\hat{a}_2) \begin{pmatrix} \text{vec } B \\ \text{vec } D \end{pmatrix} = 0,$$

where the vec -operator stacks the columns of the argument in a row vector, to the least-squares problem (3.53).

From the simple example above, one can see how the general algorithm works. In fact, consider a general OE model

$$y(t) = \frac{b_1 q^{-n_k} + \dots + b_{n_b} q^{-n_k - n_b + 1}}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}} u(t) + e(t).$$

Then the number of virtual inputs is $n_u = \lceil n_b / (n_a + 1) \rceil$ and the total number of parameters in $B(\theta)$ and $D(\theta)$ is $n_p = n_u n_a + n_u$, where $n_l = n_p - n_b$ parameters are fixed by a linear constraint. The predictor can be rewritten as

$$\hat{y}(t|t-1, \theta) \triangleq \frac{b_1 q^{-n_k} + \dots + b_{n_b} q^{-n_k - n_b + 1}}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}} u(t) = \sum_{k=1}^{n_u} G_k(q) u_k(t), \quad (5.5a)$$

where virtual inputs $u_k(t) \triangleq u(t - n_k - (k-1)(n_a + 1))$ have been introduced. With

$$m_k \triangleq (k-1)(n_a + 1),$$

the transfer-functions are given by

$$G_k(q) = b_{m_k+1} + \frac{P_k(q)}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}}, \quad k = 1, \dots, n_u, \quad (5.5b)$$

where

$$P_k(q) = \sum_{i=1}^{n_a} (b_{m_k+i+1} - a_i b_{m_k+1}) q^{-i}, \quad k = 1, \dots, n_u - 1, \quad (5.5c)$$

$$P_{n_u}(q) = \sum_{i=1}^{n_a - n_l} (b_{m_{n_u}+i+1} - a_i b_{m_{n_u}+1}) q^{-i} + \sum_{i=n_a - n_l + 1}^{n_a} (-a_i b_{m_{n_u}+1}) q^{-i}. \quad (5.5d)$$

The first sum in (5.5d) is left out if $n_a = n_l$. Writing the above transfer-function form into the OCF (3.19), the columns of the $B(\theta)$ matrix are given by

$$B_k(\theta) = \begin{pmatrix} b_{m_k+2} - a_1 b_{m_k+1} \\ b_{m_k+3} - a_2 b_{m_k+1} \\ \vdots \\ b_{m_k+n_a+1} - a_{n_a} b_{m_k+1} \end{pmatrix}, \quad (5.6a)$$

for columns $k = 1, \dots, n_u - 1$ and the last column can be written as

$$B_{n_u}(\theta) = \begin{pmatrix} b_{m_{n_u}+2} - a_1 b_{m_{n_u}+1} \\ \vdots \\ b_{m_{n_u}+n_a-n_l+1} - a_{n_a-n_l} b_{m_{n_u}+1} \\ -a_{n_a-n_l+1} b_{m_{n_u}+1} \\ \vdots \\ -a_{n_a} b_{m_{n_u}+1} \end{pmatrix}. \quad (5.6b)$$

Finally, one gets

$$D(\theta) = (b_{m_1+1} \quad \dots \quad b_{m_{n_u}+1}). \quad (5.7)$$

Assume now that we have estimates \hat{A} and \hat{C} , which have been transformed into the OCF (3.19), that is,

$$\hat{A} = (-\hat{a} \quad I_{n_a \times (n_a-1)}), \quad \hat{C} = I_{1 \times n_a}, \quad (5.8)$$

where \hat{a} is the vector containing the estimated coefficients of the pole polynomial.

The linear equality constraints when estimating $B(\theta)$ (5.6) and $D(\theta)$ (5.7) in the least-squares problem (3.53) can be expressed as

$$\begin{pmatrix} (0_{n_l \times (n_u n_a - n_l)} & I_{n_l \times n_l}) & (0_{n_l \times (n_u - 1)} & -\hat{a}_l) \end{pmatrix} \begin{pmatrix} \text{vec } B(\theta) \\ \text{vec } D(\theta) \end{pmatrix} = 0_{n_l \times 1}, \quad (5.9)$$

where $0_{m \times n}$ denotes the zero matrix of dimensions $m \times n$ and

$$\hat{a}_l \triangleq (\hat{a}_{n_a - n_l + 1} \quad \dots \quad \hat{a}_{n_a})^T.$$

It is worth noting that there is some freedom in choosing the virtual inputs and the particular choice above is just one way of dealing with the case when n_k might be zero.

Now, let us return to the previous example and see how the general formulas work.

Example 5.4: (Example 5.3 revisited)

Consider once again the discrete-time OE model with $n_b = 5$, $n_a = 2$ and $n_k = 3$:

$$y(t) = \frac{b_1 q^{-3} + b_2 q^{-4} + b_3 q^{-5} + b_4 q^{-6} + b_5 q^{-7}}{1 + a_1 q^{-1} + a_2 q^{-2}} u(t) + e(t).$$

Now, let us use the general formulas (5.5) and (5.9) to find the linear equality constraints. It holds that $n_u = \lceil n_b / (n_a + 1) \rceil = 2$, and we should introduce the virtual inputs

$$u_k(t) \triangleq u(t - n_k - (k - 1)(n_a + 1)), \quad k = 1, \dots, n_u,$$

that is $u_1(t) = u(t-3)$ and $u_2(t) = u(t-6)$. The total number of parameters in $B(\theta)$ and $D(\theta)$ is given by $n_p = n_u n_a + n_u = 6$ and the number of linear constraints is $n_l = n_p - n_b = 1$. Now, applying the problem specific parameters $\{n_a, n_b, n_k, n_l\}$ to the general predictor (5.5) yields

$$y(t|t-1, \theta) = \left(b_1 + \frac{(b_2 - b_1 a_1)q^{-1} + (b_3 - b_1 a_2)q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} \right) u_1(t) \\ + \left(b_4 + \frac{(b_5 - b_4 a_1)q^{-1} + (-b_4 a_2)q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} \right) u_2(t),$$

which coincides with the result in Example 5.3 and may be rewritten as (5.4). The linear constraints (5.9) become

$$\left(\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & -\hat{a}_2 \end{pmatrix} \right) \begin{pmatrix} \text{vec } B(\theta) \\ \text{vec } D(\theta) \end{pmatrix} = 0,$$

which equals the choice made in Example 5.3. Thus, (5.5) and (5.9) are direct generalizations of the choices made in Example 5.3.

The ideas presented so far for the initialization of a discrete-time OE model are summarized in Algorithm 4.

Algorithm 4 OE estimation using SID.

Input: A data set (3.1) and the model parameters $\{n_a, n_b, n_k\}$.

1. Introduce the virtual inputs as described in (5.5a).
2. Find estimates \hat{A} and \hat{C} of the $A(\theta)$ and $C(\theta)$ matrices of order n_a via the Steps 1-3 in Algorithm 2.
3. Transform the estimates \hat{A} and \hat{C} to the OCF (3.19) using (5.3).
4. Estimate the $B(\theta)$ and $D(\theta)$ matrices via (3.53) with equality constraints (5.9).
5. Identify the polynomials $F_p(q)$ and $B_p(q)$ in (3.11) with the elements of the estimated matrices \hat{A} , \hat{B} and \hat{D} , respectively, using (3.19) and (5.6)-(5.7).

To evaluate the algorithm, let us consider a Monte Carlo study of the estimation of a discrete-time OE model.

Example 5.5

Let the true system be given by (3.11) with

$$B_p(q) = q^{-2} - 0.4834q^{-3} - 0.2839q^{-4} - 0.02976q^{-5} \\ F_p(q) = 1 + 0.7005q^{-1} - 0.2072q^{-2} \quad (5.10)$$

and let $e(t)$ be white Gaussian noise with unit variance and zero mean. For the Monte Carlo study, output data $y(t)$ have been generated with $M = 1000$ realizations of a zero

mean white Gaussian process, with unit variance and length 1000, as input $u(t)$. The noise also has a different realization for each Monte Carlo run. The result is given in Table 5.1. The results using the proposed subspace method are denoted by SID, and IV denotes the results using the standard initialization method in the *system identification toolbox* (Ljung, 2009) in MATLAB, that is, the IV estimate is found by

$$\text{oe}(\text{data}, [\text{nb}, \text{nf}, \text{nk}], ' \text{maxiter}', 0).$$

The first two columns (BIAS) give a measure of the bias of the parameter estimates

$$\widehat{\text{Bias}} \hat{\theta} \triangleq \frac{1}{M} \left| \sum_{i=1}^M (\theta_0 - \hat{\theta}_i) \right|. \quad (5.11)$$

The last two columns (VAR) present estimates of the variances of the parameter estimates

$$\widehat{\text{Var}} \hat{\theta} \triangleq \frac{1}{M-d} \sum_{i=1}^M (\hat{\theta}_i - \bar{\hat{\theta}})^2, \quad (5.12)$$

where d is the number of parameters and $\bar{\hat{\theta}}$ is the mean value of parameter estimates.

For this example, the SID method yields lower variance when estimating the numerator polynomial $B_p(q)$ than the traditional IV estimator. One notices that some outliers were present in the IV estimates of the $B_p(q)$ polynomial which is the cause of the high variance values. On the other hand, the IV estimator yields lower bias than the proposed SID method for some of the parameters.

Table 5.1: Monte Carlo analysis of the BIAS estimate (5.11) and the VAR estimate (5.12) when estimating the parameters in the model (5.10).

| | BIAS | | VAR | |
|-------|--------|--------|--------|----------|
| | SID | IV | SID | IV |
| b_1 | 0.0026 | 0.0233 | 0.0013 | 0.2223 |
| b_2 | 0.0667 | 0.1660 | 0.2266 | 176.9809 |
| b_3 | 0.1055 | 0.0684 | 0.0648 | 15.3032 |
| b_4 | 0.0862 | 0.0810 | 0.0739 | 29.9095 |
| a_1 | 0.0642 | 0.0491 | 0.2275 | 0.2035 |
| a_2 | 0.0042 | 0.0455 | 0.1808 | 0.1777 |

Empirical studies by the author have shown that the proposed method works quite well in general and, more often than not, yields a higher model fit (3.82) on validation data than the IV method. Occasionally, the method breaks down, but in these cases it is the underlying SID method that fails to estimate the matrix $A(\theta)$ properly.

Even though only *single-input single-output* notation was used throughout the section, it is clear that the method also works for *multiple-input single-output* (MISO) systems. This is due to the use of the OCF which may be used for representing MISO systems. Since *multiple-input multiple-output* (MIMO) systems can be seen to consist of n_y MISO systems, the above procedure may also be used to get an initial estimate of MIMO systems.

5.2.2 Continuous Time

Finally, let us evaluate the proposed method when estimating a continuous-time OE model from discrete-time data. For this end, a frequency-domain SID method is needed, for instance, the method presented in Section 3.4.2. This algorithm differs only slightly from the time-domain methods and the order in which one estimates the system matrices is the same. Thus, the method developed in the discrete-time case applies readily.

Example 5.6

Let the true system be given by the Rao and Garnier (2002) test system

$$y(t) = \frac{-6400p + 1600}{p^4 + 5p^3 + 408p^2 + 416p + 1600}u(t),$$

where p is the (ordinary) differential operator. This system is non-minimum phase, it has one fast oscillatory mode with relative damping 0.1 and one slow mode with relative damping 0.25. The system can be written on the OCF as

$$\dot{x}(t) = \begin{pmatrix} -5 & 1 & 0 & 0 \\ -408 & 0 & 1 & 0 \\ -416 & 0 & 0 & 1 \\ -1600 & 0 & 0 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 0 \\ -6400 \\ 1600 \end{pmatrix} u(t),$$

$$y(t) = (1 \ 0 \ 0 \ 0) x(t) + e(t).$$

For data collection, the system is sampled with a sampling time of $T_s = 10$ ms and simulated with a random binary input of length $N = 1023$ and white Gaussian noise of unit variance is added to the output, which gives a signal to noise ratio of 7.35 dB. The data is then Fourier transformed into the frequency domain and pre-filtered before estimation where only the data lying in the frequency window $[0, 30]$ rad/s are kept, resulting in a data set of length $M = 512$. To get a continuous-time model, the sampling-time is assumed to be zero. This will introduce some bias in the estimates, but since the sampling rate is rather high this effect will be negligible.

The standard initial estimator in MATLAB, which will be denoted by IV, is determined via the command

$$\text{oe}(\text{data}, [2, 4], ' \text{focus}', [0, 30], ' \text{maxiter}', 0)$$

Now, the SID estimates of the $A(\theta)$ and $C(\theta)$ matrices can be determined by

$$\text{model} = \text{n4sid}(\text{data}, 4, ' \text{maxiter}', 0, ' \text{focus}', [0, 30])$$

and transformed into the OCF via

$$\begin{aligned} \mathbf{a} &= \text{poly}(\text{model.A}) \\ \mathbf{A} &= [-\mathbf{a}(2 : \text{end})', \text{eye}(\text{na}, \text{na} - 1)] \\ \mathbf{C} &= \text{eye}(1, \text{na}) \end{aligned}$$

Now, the regressor matrix for estimating $B(\theta)$ is created by

```

H = data.Y ./ data.U
y = [real(H); imag(H)]
for k = 1 : M
    Phi(i, :) = C * inv(i * data.Frequency(k) * eye(na) - A)
end
Phi = [real(Phi); imag(Phi)]

```

where we have scaled (3.75) with $U(i\omega)$. Since the first two elements of $B(\theta)$ should be zero, the estimate of $B(\theta)$ is found by

$$B = [0; 0; \text{Phi}(:, [3, 4]) \backslash y]$$

The validation of the methods is given in Figure 5.1 where the amplitude and phase are

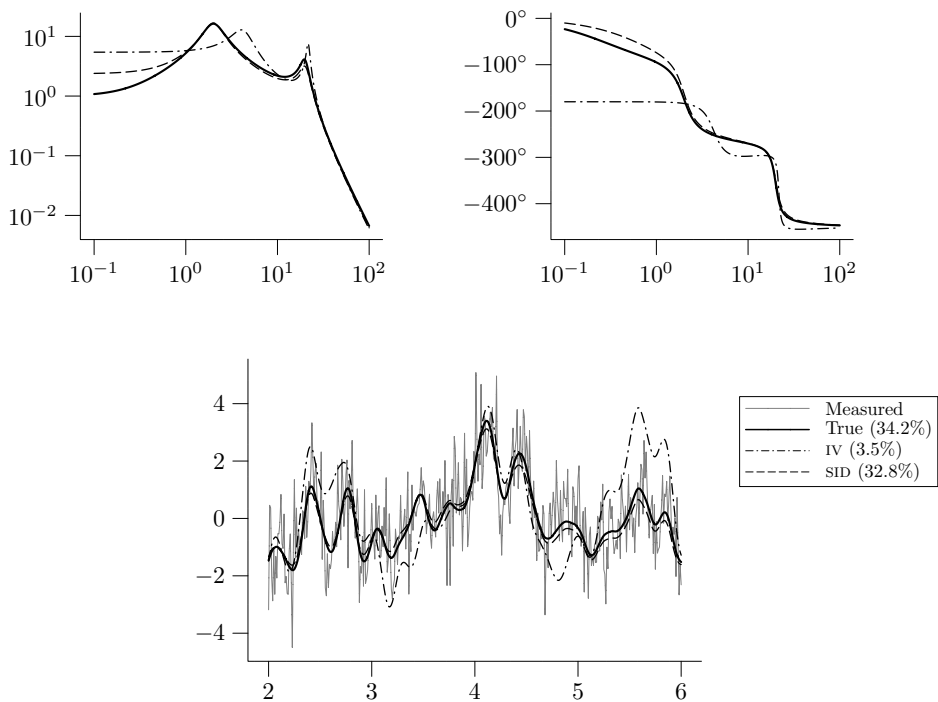


Figure 5.1: Top: The amplitude (left) and the phase (right) plot of the true system and the estimates given by the SID and the IV methods. Bottom: A simulation on validation data of the true system and the estimated models (zoomed) with the corresponding model fit values (3.82) in parenthesis.

illustrated for the different estimates. Also, a cross-validation simulation is given. One notices that the SID method captures the two modes correctly, but the static gain estimate is biased. The IV estimate does not perform as well and it has some difficulties estimating the phase.

Further information on different continuous-time identification methods using sampled data can be found in Garnier and Wang (2008) and the references therein. It is worth mentioning that, even though the IV (using oe with maxiter equal to zero) seems to perform poorly compared to the SID method in the example above, the estimate is good enough to be used as an initial starting value for the PEM, which converges to the global optimum in just a few iterations.

5.3 ARMAX Models

To find an initial estimate of the ARMAX model (3.10), the only thing that remains, after using the proposed procedure for the OE model identification, is to estimate the Kalman gain $K(\theta)$ in (3.15), which has the form (3.19):

$$K(\theta) = \begin{pmatrix} c_1 - a_1 \\ \vdots \\ c_{n_c} - a_{n_c} \\ -a_{n_c+1} \\ \vdots \\ -a_{n_a} \end{pmatrix}. \quad (5.13)$$

Here, we must have $n_c \leq n_a$, which limits the flexibility of the noise model. Now, the estimation of $K(\theta)$ requires an estimate of the process and measurement noises. For this end, the state matrix $\hat{X}_{\beta,N}$ in (3.40) is reconstructed as outlined in Section 3.4.1. Once the estimates of the system matrices and the state sequence have been found, the process noise and the measurement noise can be approximated by the residuals in (3.54), that is,

$$\begin{pmatrix} \widehat{W} \\ \widehat{V} \end{pmatrix} = \begin{pmatrix} \widehat{X}_{\beta+1,N} \\ \widehat{Y}_{\beta,1,N-1} \end{pmatrix} - \begin{pmatrix} \widehat{A} & \widehat{B} \\ \widehat{C} & \widehat{D} \end{pmatrix} \begin{pmatrix} \widehat{X}_{\beta,N-1} \\ \widehat{U}_{\beta,1,N-1} \end{pmatrix}. \quad (5.14)$$

As mentioned in Section 3.4.1, the common way to estimate $K(\theta)$ is to form the covariance matrices of (5.14) and solve the corresponding Riccati equation. In this presentation, we are going to make use of the fact that our estimated model is on the OCF. From (3.15) and (3.18) it holds that

$$K(\theta)\widehat{V} = \widehat{W}, \quad (5.15)$$

where $K(\theta)$ has the structure (5.13). Since, at this point, estimates \hat{a}_i of the elements a_i in (5.13) already have been determined, the least-squares estimate of $K(\theta)$ can be found via (5.15) with equality constraints

$$(0_{(n_a-n_c) \times n_c} \quad I_{(n_a-n_c) \times (n_a-n_c)}) K(\theta) = -\hat{a}_{n_c+1:n_a}, \quad (5.16)$$

where

$$\hat{a}_{n_c+1:n_a} = (\hat{a}_{n_c+1} \ \dots \ \hat{a}_{n_a})^T.$$

This yields in turn an estimate of the $C_p(q)$ polynomial in (3.10) as shown in (3.19). The proposed procedure for the estimation of ARMAX models is summarized in Algorithm 5.

Algorithm 5 ARMAX estimation using SID.

Given: A data set (3.1) and the model parameters $\{n_a, n_b, n_c, n_k\}$. *Limitation:* $n_c \leq n_a$.

1. Find estimates of the system matrices \hat{A} , \hat{B} , \hat{C} and \hat{D} by Algorithm 4.
 2. Reconstruct the state matrix $X_{\beta, N}$, see Section 3.4.1.
 3. Find estimates \hat{W} and \hat{V} of the residuals from (5.14).
 4. Estimate $K(\theta)$ by solving (5.15) in a least-squares sense with the equality constraints (5.16).
 5. Transform the estimated state-space model to the transfer-function form (3.10).
-

Now, let us consider a simple example.

Example 5.7

Assume that one wants to estimate an ARMAX model (3.10)

$$y(t) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} u(t) + \frac{1 + c_1 q^{-1}}{1 + a_1 q^{-1} + a_2 q^{-2}} e(t),$$

which can be rewritten as

$$\begin{aligned} x(t+1) &= \begin{pmatrix} -a_1 & 1 \\ -a_2 & 0 \end{pmatrix} x(t) + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} u(t) + \begin{pmatrix} c_1 - a_1 \\ -a_2 \end{pmatrix} e(t), \\ y(t) &= (1 \ 0) x(t) + e(t). \end{aligned}$$

The use of Algorithm 5 is straightforward and the equality constraint in step 4 is given by

$$(0 \ 1) K(\theta) = -\hat{a}_2,$$

where \hat{a}_2 is found in the first step of the algorithm.

Thus, the estimation procedure for the ARMAX model structure is only a minor extension of the discrete-time OE identification algorithm given in Section 5.2. Now, let us compare the results using Algorithm 5 and the IV based estimator given by

$$\text{armax}(\text{data}, [\text{na}, \text{nb}, \text{nc}, \text{nk}], 'maxiter', 0)$$

on some simulated data.

Example 5.8

Let the true system be given by (3.10) with

$$\begin{aligned} A_p(q) &= 1 - 0.06353q^{-1} + 0.006253q^{-2} + 0.0002485q^{-3} \\ B_p(q) &= 1 - 0.8744q^{-1} - 0.3486q^{-2} + 0.331q^{-3} \\ C_p(q) &= 1 + 0.3642q^{-1} \end{aligned}$$

and let $e(t)$ be white Gaussian noise with unit variance and zero mean. For the Monte Carlo analysis, output data $y(t)$ have been generated with $M = 1000$ realizations of a zero mean white Gaussian process, with unit variance and length 1000, as input $u(t)$. The noise also has a different realization for each Monte Carlo run. The result is given in Table 5.2 and the two methods seem to have comparable performance, where the proposed SID method only has a slight advantage in the bias and the variance.

Table 5.2: Monte Carlo analysis of the BIAS estimate (5.11) and the VAR estimate (5.12) when estimating the parameters in Example 5.8.

| | BIAS | | VAR | |
|-------|--------|--------|--------|--------|
| | SID | IV | SID | IV |
| a_1 | 0.0137 | 0.0329 | 0.0101 | 0.0447 |
| a_2 | 0.0025 | 0.0100 | 0.0016 | 0.0047 |
| a_3 | 0.0040 | 0.0072 | 0.0004 | 0.0021 |
| b_1 | 0.0000 | 0.0001 | 0.0001 | 0.0001 |
| b_2 | 0.0137 | 0.0367 | 0.0102 | 0.0560 |
| b_3 | 0.0083 | 0.0190 | 0.0027 | 0.0153 |
| b_4 | 0.0018 | 0.0157 | 0.0036 | 0.0125 |
| c_1 | 0.0141 | 0.0141 | 0.0102 | 0.0213 |

The results in the example above are quite typical for the ARMAX case. It seems, to the author's experience, that the IV based initialization method (implemented in, Ljung, 2009) for ARMAX models often works better than the corresponding method for OE models. Thus, the benefit of using the proposed SID method, compared to the IV method, seems to be greater in the OE case than in the ARMAX case.

5.4 Special Gray-Box Models

The ideas presented so far have been based on a coordinate change of the state-space model to the OCF, which works well for certain black-box model structures. But what if a different structure is present in the state-space form, like the ones in linear gray-box models?

Example 5.9: (DC motor)

A simple example of a gray-box structure is given by the state-space model

$$\dot{x}(t) = \begin{pmatrix} 0 & 1 \\ 0 & \theta_1 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \theta_2 \end{pmatrix} u(t) \quad (5.18a)$$

$$y(t) = (1 \ 0) x(t) + e(t), \quad (5.18b)$$

which is often used to represent a DC motor (see, for instance, Ljung, 1999, page 95). The states x_1 and x_2 represent the angle and the angular velocity, respectively, of the motor shaft and the parameters θ_1 and θ_2 contain the information about values of the resistance, inductance, friction, and the inertia of the motor.

Now, assume that estimates \hat{A} and \hat{C} of the matrices $A(\theta)$ and $C(\theta)$ in (5.18) have been found using some continuous-time SID method:

$$\hat{A} = \begin{pmatrix} \hat{a}_{11} & \hat{a}_{12} \\ \hat{a}_{21} & \hat{a}_{22} \end{pmatrix}, \quad \hat{C} = (\hat{c}_{11} \ \hat{c}_{12}).$$

Then, asymptotically, there is a similarity transform T satisfying

$$\begin{pmatrix} \hat{a}_{11} & \hat{a}_{12} \\ \hat{a}_{21} & \hat{a}_{22} \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & \theta_1 \end{pmatrix} \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix}^{-1},$$

$$(\hat{c}_{11} \ \hat{c}_{12}) = (1 \ 0) \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix}^{-1}.$$

Multiplying the above relation with T^{-1} from the left yields

$$\begin{pmatrix} \hat{a}_{11}\tilde{t}_{11} + \hat{a}_{21}\tilde{t}_{12} & \hat{a}_{12}\tilde{t}_{11} + \hat{a}_{22}\tilde{t}_{12} \\ \hat{a}_{11}\tilde{t}_{21} + \hat{a}_{21}\tilde{t}_{22} & \hat{a}_{12}\tilde{t}_{21} + \hat{a}_{22}\tilde{t}_{22} \end{pmatrix} = \begin{pmatrix} \tilde{t}_{21} & \tilde{t}_{22} \\ \tilde{t}_{21}\theta_1 & \tilde{t}_{22}\theta_1 \end{pmatrix},$$

$$(\hat{c}_{11} \ \hat{c}_{12}) = (\tilde{t}_{11} \ \tilde{t}_{12}),$$

where \tilde{t}_{ij} denotes the elements of T^{-1} . The equations not containing θ_1 become

$$\begin{aligned} \hat{a}_{11}\tilde{t}_{11} + \hat{a}_{21}\tilde{t}_{12} - \tilde{t}_{21} &= 0 \\ \hat{a}_{12}\tilde{t}_{11} + \hat{a}_{22}\tilde{t}_{12} - \tilde{t}_{22} &= 0 \\ \tilde{t}_{11} &= \hat{c}_{11} \\ \tilde{t}_{12} &= \hat{c}_{12} \end{aligned}$$

which can easily be solved for \tilde{t}_{11} , \tilde{t}_{12} , \tilde{t}_{21} and \tilde{t}_{22} . Thus, a similarity transform, which takes the estimates \hat{A} and \hat{C} as close as possible to the gray-box structure represented by (5.18) may be found, unless the system of equations is under-determined.

Inspired by the example above, let us consider a more general case. Here, let $A(\theta)$ and $C(\theta)$ represent the gray-box structure of system we would like to estimate. By using some SID method, estimates \hat{A} and \hat{C} of $A(\theta)$ and $C(\theta)$, respectively, can be found. Asymptotically, as the number of data points tends to infinity, the matrix estimates are

related with the structured matrices via a similarity transform T from the true values, that is, it holds that

$$\widehat{A} = TA(\theta)T^{-1}, \quad \widehat{C} = C(\theta)T^{-1}. \quad (5.19)$$

If one would be able to determine the transform T , then one could find the parameters θ by solving some system of equations. Multiplying (5.19) with T from the right yields

$$\widehat{A}T = TA(\theta), \quad \widehat{C}T = C(\theta).$$

This may be written as

$$\begin{pmatrix} I \otimes \widehat{A} - A(\theta)^T \otimes I \\ I \otimes \widehat{C} \end{pmatrix} \text{vec} T = \begin{pmatrix} 0 \\ \text{vec} C(\theta) \end{pmatrix}, \quad (5.20)$$

where \otimes denotes the Kronecker product. For certain linear gray-box structures, where all the unknown parameters of $A(\theta)$ lie in one row or one column and all the elements of $C(\theta)$ are known, the equation (5.20) can be solved for T .

Example 5.10: (OCF)

In the Sections 5.2 and 5.3, dealing with the initialization of the OE and the ARMAX model structures, the OCF (3.19) was used as a state-space representation of the transfer-function models. Instead of determining the eigenvalues of the \widehat{A} matrix to transform the estimated state-space model to the “gray-box” structure (3.19), one can determine the similarity transform via (5.20) to achieve the same result. Doing this, the left hand side of (5.20) becomes

$$\begin{pmatrix} \widehat{A} + a_1 I & a_2 I & a_3 I & \dots & a_{n_a-1} I & a_{n_a} I \\ -I & \widehat{A} & 0 & \dots & 0 & 0 \\ 0 & -I & \widehat{A} & \dots & 0 & 0 \\ 0 & 0 & -I & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -I & \widehat{A} \\ \widehat{C} & 0 & 0 & \dots & 0 & 0 \\ 0 & \widehat{C} & 0 & \dots & 0 & 0 \\ 0 & 0 & \widehat{C} & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \widehat{C} & 0 \\ 0 & 0 & 0 & \dots & 0 & \widehat{C} \end{pmatrix} \text{vec} T,$$

and the right hand side is a vector with all elements equal to zero except the $n_a^2 + 1$ element that is equal to one. Thus, it is only the first n_a equations that depend on the unknown parameters. The remaining number of equations sum up to n_a^2 , which is equal to the number of unknowns in T . Thus, (5.20) is solvable for T whenever the lower part of the matrix above has full rank.

The procedure for finding T may work when $A(\theta)$ only has one column depending on the unknown parameters. If $A(\theta)$ has all parameters in one row one should try to solve (5.19) for T^{-1} instead, which is done by multiplying the equation with T^{-1} from the left and then vectorizing the result with respect to T^{-1} .

We are now ready to give a procedure to find an initial estimate of the special gray-box structures discussed above. Assume that a dataset (3.1) and a gray-box structure of

order n_x have been given. Furthermore, assume that all unknown parameters in $A(\theta)$ are in either one row or one column.

- 1) Find estimates \hat{A} and \hat{C} of the $A(\theta)$ and $C(\theta)$ matrices of order $n_x \times n_x$ and $n_y \times n_x$, respectively, where $n_x = \dim(x)$ and $n_y = \dim(y)$, via (3.50).
- 2) Find the similarity transform T that takes the estimates \hat{A} and \hat{C} as close as possible to the gray-box structure via (5.20) and transform the estimates $\hat{A} \leftarrow T\hat{A}T^{-1}$ and $\hat{C} \leftarrow \hat{C}T^{-1}$.
- 3) Solve the least-squares problem (3.53) with equality constraints on the known elements in the $B(\theta)$ and $D(\theta)$ matrices.
- 4) Reconstruct the state matrix $X_{\beta,N}$ via (3.65).
- 5) Re-estimate the system matrices according to

$$\arg \min_{A(\theta), B(\theta), C(\theta), D(\theta)} \left\| \begin{pmatrix} \hat{X}_{\beta+1, N} \\ Y_{\beta+1, \alpha, N} \end{pmatrix} - \begin{pmatrix} A(\theta) & B(\theta) \\ C(\theta) & D(\theta) \end{pmatrix} \begin{pmatrix} \hat{X}_{\beta, N-1} \\ U_{\beta, \alpha, N-1} \end{pmatrix} \right\|^2,$$

with equality constraints on all known elements.

The last step of the procedure ensures that the constraints on the elements in the $A(\theta)$ matrix are, in some way, taken into account. Let us test the above procedure in a simple example.

Example 5.11

Let the true system be given by

$$x(t+1) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \theta_1 & 0 & \theta_2 \end{pmatrix} x(t) + \begin{pmatrix} \theta_3 \\ 0 \\ \theta_4 \end{pmatrix} u(t) \quad (5.21a)$$

$$y(t) = (1 \ 0 \ 0) x(t) + e(t) \quad (5.21b)$$

with $\theta = (-0.3, 0.3, 0.1, -0.1)^T$. Here, $e(t)$ is white Gaussian noise of unit variance. For the Monte Carlo analysis, output data $y(t)$ have been generated with $M = 500$ realizations of a zero mean white Gaussian process, with unit variance and length 10000, as input $u(t)$. The parameters θ have been estimated for each realization with the SID procedure presented above. The mean value and the variance of the parameter estimates were given by

$$\bar{\hat{\theta}} = (-0.2873 \ 0.2675 \ 0.1003 \ -0.0921)^T, \quad (5.22a)$$

$$\widehat{\text{Var}} \hat{\theta} = 10^{-2} \times (0.2395 \ 0.6518 \ 0.0096 \ 0.0144)^T. \quad (5.22b)$$

If no consideration is given to the zero at the (3, 2) element in the system matrix $A(\theta)$ in (5.18), that is, the final step 5 in the proposed procedure is not performed, the mean values of the parameter estimates become

$$\bar{\hat{\theta}}_1 = -0.2049, \quad \bar{\hat{\theta}}_2 = 0.3957, \quad (5.23a)$$

and their variances

$$\widehat{\text{Var}} \tilde{\theta}_1 = 0.0293, \quad \widehat{\text{Var}} \tilde{\theta}_2 = 0.1004. \quad (5.23b)$$

This shows, as could be expected, a slight increase in variance when not taking the structure of $A(\theta)$ into account. Furthermore, there also seems to be a significant increase in the bias of the parameter estimates.

The procedure presented above is similar to the method proposed in Xie and Ljung (2002). There, an initial model of the system is first found by an unstructured subspace method and then the similarity transform is found by solving a least-squares problem. The method then alternates between solving a nonlinear parameter estimation problem and improving the similarity transform.

5.5 Discussion

In this chapter, a new algorithm for initial parameter estimation of certain linear model structures which makes use of the standard SID methods has been presented. The algorithm is valid for both discrete-time and continuous-time identification.

The proposed method might be generalized to handle Box-Jenkins models by using the method presented in Reynders and De Roeck (2009) or similar SID methods, which is a topic for future work. Furthermore, SID methods for LPV state-space models and Wiener-Hammerstein systems might benefit from the ideas presented.

The original SID methods, presented in Section 3.4, contain means to estimate the number of states needed to describe the system. This is done by analyzing the singular values of the estimated observability matrix Γ_α , that is, the relation in (3.49). With the ideas presented in this chapter, this may be extended to automatic order selection of OE and ARMAX models, where the number of parameters used in the estimated $B(\theta)$, $D(\theta)$ and $K(\theta)$ matrices may be determined by some order selection algorithm for linear regression models.

The use of the OCF to represent state-space models is convenient in terms of simplicity in choosing the linear equality constraints that should be used when estimating the $B(\theta)$ and $D(\theta)$ matrices in the state-space model. The drawback is the sensitivity to small permutations in the parameters of the OCF. Thus, it would be interesting to find a different representation which does not suffer as much from this drawback. Furthermore, a different representation may enable a one-shot method for MIMO systems, instead of estimating several MISO systems.

Finally, in addition to the structure information we have incorporated into the SID method in this chapter, there might also be some other prior information that is known. For example, if the static gain of the system is known a priori, how does one incorporate such structure information?

6

Difference Algebra and System Identification

The framework of differential algebra, especially Ritt's algorithm, has turned out to be a useful tool when analyzing the identifiability of certain nonlinear continuous-time model structures (Ljung and Glad, 1994). This framework is conceptual rather than practical and it provides means to analyze complex nonlinear model structures via the much simpler linear regression models

$$y(t) = \varphi(t)^T \theta + e(t). \quad (6.1)$$

One difficulty when working with continuous-time signals is dealing with white noise signals and in Ljung and Glad (1994) these effects are ignored. In this chapter, *difference* algebraic techniques, which mimics the differential algebraic techniques, will be developed. Besides making it possible to analyze discrete-time systems, this opens up the possibility of dealing with noise.

6.1 Introduction

The analysis of nonlinearly parametrized model structures is an important branch of system identification. For instance, many physically motivated model structures fall into this category (see, for instance, Ljung, 1999). In this chapter, we will try to generalize the differential algebraic framework presented in Ljung and Glad (1994) for discrete-time systems, see also Section 3.5. Let us first consider a simple example.

Example 6.1: (Example 1.3 revisited)

Consider the problem of estimating the parameter θ in

$$y(t) = \theta u(t) + \theta^2 u(t-1) + e(t), \quad (6.2)$$

where $y(t)$ is the output, $u(t)$ is the input, and $e(t)$ is the noise. This is the same model structure as in Example 1.3. Also, by exchanging the time shift $u(t-1)$ in (6.2) with the

derivative $\dot{u}(t)$, this corresponds directly to the model structure in Example 3.3. Let us, inspired by the discussion in Example 3.3, time shift (6.2)

$$y(t-1) = \theta u(t-1) + \theta^2 u(t-2) + e(t-1). \quad (6.3)$$

By examining the equations, we see that by multiplying (6.2) by $u(t-2)$ and (6.3) by $u(t-1)$, and then subtracting the result, we obtain

$$\begin{aligned} u(t-2)y(t) - u(t-1)y(t-1) \\ = (u(t-2)u(t) - u^2(t-1))\theta \\ + u(t-2)e(t) - u(t-1)e(t-1). \end{aligned} \quad (6.4)$$

This may be written as a linear regression model if one replaces $y(t)$ in (6.1) with the left hand side of (6.4) and $\varphi(t)$ in (6.1) with the coefficient in front of θ in (6.4), respectively. Thus, instead of analyzing the properties of the model structure defined by (6.2) one may work with a linear regression model for which there are many well known properties.

Furthermore, if one formulates the estimation problem as minimizing the squared prediction error (3.5) with respect to the parameter θ using the representation (6.2) and the same signals as in Example 1.3, one notices that the optimization problem becomes non-convex, even for such a simple model structure, see Figure 6.1.

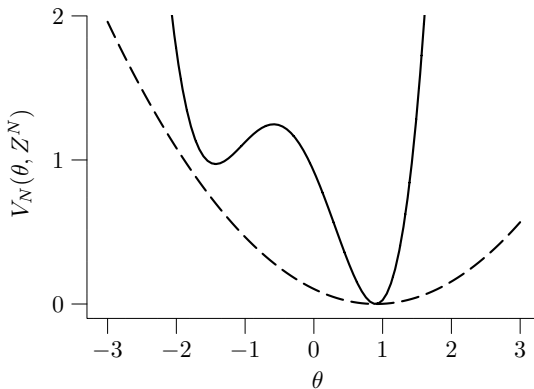


Figure 6.1: The cost function for the problem of estimating the parameter θ in (6.2) given the same data as in Example 1.3. The solid line represents using (6.2) directly and the dashed line using the reformulated description (6.4).

On the other hand, formulating the equivalent estimation problem, but now with the equivalent representation (6.4), one gets a convex optimization problem as illustrated in Figure 6.1. This motivates that, at least in this simple example, algebraic techniques may be useful in finding initial estimates for certain nonlinear model structures.

The calculations in the example above are inspired by the methods of differential algebra, especially the framework based on Ritt's algorithm presented in Ljung and Glad (1994). The main result in Ljung and Glad (1994) shows that a continuous-time model structure,

containing only polynomial nonlinearities, is globally identifiable if and only if it can be written as a linear regression. In particular, this implies that once the necessary differential algebraic manipulations have been made, an initial estimate of the unknown parameters may be found by solving a simple least-squares problem. For further references to the differential algebraic framework and its application to control theoretical problems, see Section 3.5.

In this chapter, motivated by the result presented in Ljung and Glad (1994) and the fact that most system identification problems involve sampled data, we aim to generalize these results to discrete time. Also, if one can generalize the methods of differential algebra to work for discrete-time systems, differentiation of noise signals will no longer be an issue and the noise can be manipulated as any other signal. Attempts to mimic Ritt's algorithm for discrete-time systems have already been made by Kotsios (2001), where the so-called δ -operators are introduced and different products of these are discussed. In contrast, the aim in this presentation is to generalize Ritt's algorithm with a minimum amount of changes compared to the continuous-time case.

The structure of the chapter is as follows: In Section 6.2, the basic algebraic concepts used in this chapter are presented. Furthermore, Ritt's algorithm is formulated and its finite time termination is proved. The generalization of the identifiability results presented Ljung and Glad (1994) is discussed in Section 6.3. Finally, some implications for the initialization of system identification problems are given in Section 6.4.

6.2 Algebraic Concepts

In this section, we are interested in formalizing the algebra concerning solutions to systems of polynomial equations, where the polynomials depend only on a finite number of variables (which are themselves elements of one or several time-dependent signals), that is, the solution to systems of *difference equations*. For example, the polynomial $x(t)^3 + 1$ in the variable $x(t)$ is a function f of the sequence $X_t = (x(t - \tau))_{\tau=0}^{\infty}$ which maps X_t to the polynomial $X_{t,1}^3 + 1$ (where $X_{t,1}$ is the first element in the sequence X_t). The solution to the difference equation $f = 0$, that is $x(t)^3 + 1 = 0$, is thus a sequence $(x(\tau))_{\tau=-\infty}^{\infty}$ satisfying $f(X_t) = 0$ for all t . In general, the solution to a system of difference equations will be found by algebraic manipulations involving the backward shift operator q^{-1} , which applied our example polynomial results in $q^{-1}f(X_t) = x(t+1)^3 + 1$. Thus, the time-shifted polynomial $q^{-1}f$ is a function of the second element $X_{t,2}$ of the sequence X_t in the same way as f is a function of $X_{t,1}$. Thus, the two polynomials $f(X_t)$ and $q^{-1}f(X_t)$ are considered to be different. For the sake of notational convenience, from here on the argument of the polynomials will be left out. Starting with the basics of difference algebra for time-dependent signals, we will move on to polynomials and finally an algorithm is presented for systems of difference polynomials.

6.2.1 Signal Shifts

As discussed above, we are interested in systems described by polynomials in time-dependent variables and their time shifts. The shifts will be denoted by

$$u^{(k)} \triangleq q^{-k}u(t), \quad k \in \mathbb{N}, \quad (6.5)$$

where q is the forward time shift operator and the order of the displacement is given in the parenthesis¹. To simplify the notation in the examples to come, we also introduce $\dot{u} \triangleq u^{(1)}$ and $\ddot{u} \triangleq u^{(2)}$. A fundamental concept for the algorithmic aspects of differential algebra is *ranking*. This is a *total ordering* (see, for instance, Lang, 2002) of all variables and their derivatives. In the discrete-time case it corresponds to a total ordering of all time-shifted variables.

Definition 6.1 (Ordering, Ranking). A binary operator \prec , which is a total ordering satisfying

$$(i) \quad u^{(\mu)} \prec u^{(\mu+\sigma)}$$

$$(ii) \quad u^{(\mu)} \prec y^{(\nu)} \implies u^{(\mu+\sigma)} \prec y^{(\nu+\sigma)}.$$

for all $\mu, \nu, \sigma \in \mathbb{N}$, is called an *ordering of the signals* u and y and we say that u is *ranked* lower than y if $u \prec y$.

There are many possible choices of orderings of signals. For example, let u and y be two signals. Then two possible orderings are

$$u \prec y \prec \dot{u} \prec \dot{y} \prec \ddot{u} \prec \ddot{y} \prec \dots \quad (6.6a)$$

$$u \prec \dot{u} \prec \ddot{u} \prec \dots \prec y \prec \dot{y} \prec \ddot{y} \prec \dots \quad (6.6b)$$

The latter ordering will often be written in short $u^{(\cdot)} \prec y^{(\cdot)}$. Let us turn our attention to polynomials with variables that are time-shifted signals. These polynomials will be used to represent difference equations as discussed in the introduction to this section.

6.2.2 Polynomials

As with signals, polynomials can also be time-shifted. In fact, the polynomial $f^{(\sigma)}$ is the result when all variables in the polynomial f have been shifted $\sigma \in \mathbb{N}$ time steps. Even though most of the results and definitions in this section apply to polynomials in static variables, the formulations will focus on polynomials in time-dependent variables.

To illustrate the algebraic concepts below in a simple manner, using the notation introduced in (6.5), let

$$f \triangleq \dot{u}y + \ddot{u}^3\dot{y}^2, \quad (6.7a)$$

$$g \triangleq \dot{y}^2 + \ddot{y}, \quad (6.7b)$$

$$h \triangleq u + \dot{u}^2, \quad (6.7c)$$

be polynomials in the sequences $U_t = (u(t + \tau))_{\tau=0}^{\infty}$ and $Y_t = (y(t + \tau))_{\tau=0}^{\infty}$.

To be able to order polynomials we need to find the highest ranked variable in the polynomial.

Definition 6.2 (Leader, Degree). The highest ranked time-shifted variable in a, possibly time-shifted, polynomial f is called the *leader* and is denoted by ℓ_f . The *degree* of a variable x in f is the highest exponent of x that occurs in f and is denoted by $\deg_x f$.

¹Alternatively, we may use shifts forwards in time $u^{(k)} \triangleq q^k u(t)$ for all $k \in \mathbb{N}$ and the same theory applies.

The polynomial (6.7a), with the ordering (6.6a), has the leader $\ell_f = \ddot{u}$ with $\deg_{\ddot{u}} f = 3$ and if the ordering is changed to (6.6b), the leader is given by $\ell_f = \dot{y}$ with $\deg_{\dot{y}} f = 2$. Thus, the leader depends on both the polynomial at hand and the ordering of the time-shifted signals that is used. The ranking of polynomials can now be defined.

Definition 6.3 (Ranking). Let f and g be two polynomials with leaders ℓ_f and ℓ_g , respectively. We say that f is *ranked lower* than g , denoted $f \prec g$, if either $\ell_f \prec \ell_g$ or if $\ell_f = \ell_g$ and $\deg_{\ell_f} f < \deg_{\ell_f} g$. If $\ell_f = \ell_g$ and $\deg_{\ell_f} f = \deg_{\ell_f} g$, we say that f and g have *equal ranking* and write $f \sim g$.

The polynomials f and g in (6.7), with the ordering (6.6a), have the leaders $\ell_f = \ddot{u}$ and $\ell_g = \ddot{y}$, respectively. Since $\ddot{u} \prec \ddot{y}$, according to (6.6a), it follows that $f \prec g$.

In this section, we will be dealing with the elimination of variables in time-shifted polynomials. In this context the following concept is important.

Definition 6.4 (Reduced). Let f be a polynomial with leader ℓ_f . A polynomial g is said to be *reduced* with respect to f if there is no positive time shift of ℓ_f in g and if $\deg_{\ell_f} g < \deg_{\ell_f} f$.

Using the ordering (6.6b) with the polynomials f and g in (6.7), the leaders are given by $\ell_f = \dot{y}$ and $\ell_g = \ddot{y}$, respectively. Thus, in this case, f is reduced with respect to g but not vice versa. The above concepts (ranking and reduced) are related as follows.

Lemma 6.1

Let f and g be two polynomials. If $f \prec g$ under some ordering, then f is also reduced with respect to g under that ordering.

Proof: If $f \prec g$, then either $\ell_f \prec \ell_g$ or $\ell_f = \ell_g$ and $\deg_{\ell_f} f < \deg_{\ell_g} g$. In the former case, then it follows from Definition 6.1 that $\ell_f \prec \ell_g \prec \ell_g^{(\sigma)}$ for all $\sigma \in \mathbb{Z}_+$. Thus, f does not depend on $\ell_g^{(\sigma)}$ for any $\sigma \in \mathbb{N}$ and in particular $0 = \deg_{\ell_g} f < \deg_{\ell_g} g$. Hence, f must be reduced with respect to g . The latter case follows in a similar fashion. \square

That the two concepts are not equivalent is easily seen if one chooses the ordering (6.6a) with the simple polynomials $f = y$ and $g = u$. Since f does not depend on the variable u , it holds that f is reduced with respect to g , but $f \not\prec g$ since $\ell_g \prec \ell_f$.

Before we continue providing the tool needed to reduce polynomials with respect to each other, some additional concepts of difference polynomials are needed. These will not play a major role in what follows but are used to guarantee the existence of solutions to the resulting reduced systems of polynomials.

Definition 6.5 (Separant, Initial). The *separant* S_f of a polynomial f is the partial derivative of f with respect to the leader, while the *initial* I_f is the coefficient of the highest power of the leader in f .

The polynomial f in (6.7a) have, under the ordering (6.6a), the leader $\ell_f = \ddot{u}$. This implies that the separant is given by $S_f = 3\ddot{u}^2\dot{y}^2$ and the initial $I_f = \dot{y}^2$. The tool needed to reduce polynomials with respect to each other is a variant of the standard polynomial division.

Lemma 6.2 (Pseudo-division)

Let f and g be polynomials in the variable x of degree m and n , respectively, written in the form

$$f = a_m x^m + \cdots + a_0 \quad \text{and} \quad g = b_n x^n + \cdots + b_0,$$

where $m \geq n$. Then there exist polynomials $Q \neq 0$, \bar{Q} and R such that

$$Qf = \bar{Q}g + R,$$

where $\deg_x R < n$. Furthermore, with Q given by b_n^{m-n+1} , then \bar{Q} and R are unique.

Proof: A proof can be found in Mishra (1993, p. 168). □

The suggested choice of Q in the lemma above is the initial I_g to the power of $m - n + 1$. It is also worth noting that the polynomials f and g in Lemma 6.2 may be multivariable, which implies that the coefficients of the variable x are polynomials in the remaining variables. Now, let us illustrate the concept of pseudo-division on the polynomials f and g in (6.7), where the variable in question is \dot{y} . Since $I_g = 1$, it holds that

$$f = \dot{u}^3 g + (-\ddot{u}^3 \dot{y} + \dot{u}y),$$

so that $Q = 1$, $\bar{Q} = \ddot{u}^3$ and $R = -\ddot{u}^3 \dot{y} + \dot{u}y$, which satisfies Lemma 6.2 if we notice that $\deg_{\dot{y}} R = 0$. During the algebraic simplifications, it is important that the solutions to the original system of polynomial equations are preserved. To this end, the following results show how pseudo-division can be used to eliminate variables while preserving the solution.

Lemma 6.3

Let g be a polynomial with leader ℓ_g and let f be a polynomial containing $\ell_g^{(\sigma)}$ for some $\sigma \in \mathbb{N}$. Then there exist polynomials R and Q such that

(i) R does not contain $\ell_g^{(\sigma)}$.

(ii) Every solution of $f = 0$, $g = 0$ is also a solution of $R = 0$, $g = 0$.

(iii) Every solution of $R = 0$, $g = 0$ with $Q \neq 0$ is also a solution of $f = 0$, $g = 0$.

Proof: Let m and n be the degrees of f and $g^{(\sigma)}$ as polynomials in $\ell_g^{(\sigma)}$, respectively. On one hand, if $m \geq n$, then pseudo-division according to Lemma 6.2 yields polynomials $Q_1 \neq 0$, \bar{Q}_1 and R_1 such that

$$Q_1 f = \bar{Q}_1 g^{(\sigma)} + R_1, \tag{6.8}$$

with $\deg_{\ell_g^{(\sigma)}} R_1 < n \leq m$. If R_1 still depends on the variable $\ell_g^{(\sigma)}$, then further pseudo-division yields polynomials $Q_2 \neq 0$, \bar{Q}_2 and R_2 such that

$$Q_2 g^{(\sigma)} = \bar{Q}_2 R_1 + \tilde{R}_2, \tag{6.9}$$

with $\deg_{\ell_g^{(\sigma)}} \tilde{R}_2 < \deg_{\ell_g^{(\sigma)}} R_1$. Combining (6.9) and (6.8) yields

$$Q_1 \bar{Q}_2 f = (Q_2 + \bar{Q}_1 \bar{Q}_2) g^{(\sigma)} + R_2,$$

where we have defined $R_2 \triangleq -\tilde{R}_2$. Continuing in this manner, by recursive application of Lemma 6.2, one may always construct polynomials Q , \bar{Q} and R such that

$$Qf = \bar{Q}g^{(\sigma)} + R, \quad (6.10)$$

with $\deg_{\ell_g^{(\sigma)}} R = 0$, that is, R does not depend on the variable $\ell_g^{(\sigma)}$. Now, assume that such polynomials have been constructed. Then the first statement in the lemma holds true. For the second statement, rewrite (6.10) as

$$R = Qf - \bar{Q}g^{(\sigma)}.$$

Since $g^{(\sigma)}$ is just g with all variables time shifted $\sigma \geq 0$ steps, it must hold that $g^{(\sigma)} = 0$ whenever $g = 0$. Thus, $R = 0$ whenever $f = 0$ and $g = 0$. For the final statement, rewrite (6.10), assuming $Q \neq 0$, as

$$f = \frac{1}{Q}(\bar{Q}g^{(\sigma)} + R).$$

Since $g = 0$ implies $g^{(\sigma)} = 0$, it follows that $f = 0$ whenever $R = 0$ and $g = 0$, which concludes the proof for the case when $m \geq n$. On the other hand, if $m < n$, then Lemma 6.2 yields polynomials $Q \neq 0$, \bar{Q} and R such that

$$Qg^{(\sigma)} = \bar{Q}f + R$$

with $\deg_{\ell_g^{(\sigma)}} R < m$ and similar arguments as in the former case can be applied. \square

In the proof of the Lemma 6.3 above, we see that it is possible that several pseudo-divisions are needed in the case $\sigma > 0$ to eliminate $\ell_g^{(\sigma)}$ from the polynomial f . This is the main difference between the continuous-time case and the discrete-time case. In the continuous-time case, $g^{(\sigma)}$ is affine in $\ell_g^{(\sigma)}$ (due to the product rule of differentiation) and thus only one pseudo-division is needed to eliminate $\ell_g^{(\sigma)}$ from f . In the discrete-time case, time shifting a polynomial does not change its degree with respect to a variable, that is, $\deg_{\ell_g} g = \deg_{\ell_g^{(\sigma)}} g^{(\sigma)}$ and several pseudo-divisions might be needed.

Now, the following extension of Lemma 6.3 provides (as we will see later) the main step in Ritt's algorithm.

Lemma 6.4

Let f be a polynomial which is not reduced with respect to the polynomial g . Then there exist polynomials R and Q such that

(i) *R is reduced with respect to g .*

(ii) *Every solution of $f = 0$, $g = 0$ is also a solution of $R = 0$, $g = 0$.*

(iii) *Every solution of $R = 0$, $g = 0$ with $Q \neq 0$ is also a solution of $f = 0$, $g = 0$.*

Proof: If f is not reduced with respect to g , then either f contains some positive time-shift of the leader ℓ_g of g or else f contains ℓ_g to a higher power than g .

In the latter case, Lemma 6.3 yields polynomials Q , \bar{Q} and R such that

$$Qf = \bar{Q}g + R, \quad (6.11)$$

where R is reduced with respect to g . Thus, the proof of the statements follows in the same way as in proof of Lemma 6.3.

In the former case, the polynomial f contains $\ell_g^{(\sigma_1)}$ for some $\sigma_1 \in \mathbb{Z}_+$. Lemma 6.3 then yields polynomials Q_1 , \bar{Q}_1 and R_1 satisfying

$$Q_1f = \bar{Q}_1g^{(\sigma_1)} + R_1, \quad (6.12)$$

where R_1 does not contain $\ell_g^{(\sigma_1)}$. If R_1 still contains $\ell_g^{(\sigma_2)}$ for some $\sigma_2 \in \mathbb{N}$ with $\sigma_2 < \sigma_1$, further use of Lemma 6.3 yields polynomials Q_2 , \bar{Q}_2 and R_2 satisfying

$$Q_2R_1 = \bar{Q}_2g^{(\sigma_2)} + R_2, \quad (6.13)$$

where R_2 does not contain $\ell_g^{(\sigma_2)}$. Combining (6.12) and (6.13) yields

$$Q_2Q_1f = Q_2\bar{Q}_1g^{(\sigma_1)} + \bar{Q}_2g^{(\sigma_2)} + R_2.$$

Continuing in this manner, by recursive application of Lemma 6.3, one can construct polynomials Q and R such that

$$Qf = \sum_{i=1}^n \bar{Q}_i g^{(\sigma_i)} + R, \quad (6.14)$$

for some sequence of polynomials \bar{Q}_i , $i = 1, \dots, n$, where R does not contain $\ell_g^{(\sigma)}$ for any $\sigma \in \mathbb{N}$. Thus, we have constructed a polynomial R that is reduced with respect to g . Since $g = 0$ implies that $g^{(\sigma)} = 0$ for all $\sigma \in \mathbb{N}$, the remaining statements follow in the same way as in the proof of Lemma 6.3. \square

Lemma 6.4 shows how to reduce one difference equation with respect to another while preserving the solution of both and that the main tool for achieving this is the pseudo-division concept presented in Lemma 6.2. This is important when dealing with systems of difference equations which is the topic of the following section.

6.2.3 Systems of Polynomials

Now, we are ready to define the necessary concepts for systems of polynomial difference equations, which will be represented by sets of difference polynomials. The following definition generalizes the concept of reduced polynomials.

Definition 6.6 (Auto-reduced). A set $\mathcal{A} = \{A_1, \dots, A_p\}$ of polynomials is called *auto-reduced* if all elements A_i are pairwise reduced with respect to each other. If, in addition, the polynomials A_1, \dots, A_p in the auto-reduced set \mathcal{A} are in increasing rank, then \mathcal{A} is said to be *ordered*.

Let us once again consider the polynomials f and g in (6.7), but now with the ordering (6.6a). Then it is easy to see that f and g are reduced with respect to each other and the set $\{f, g\}$ is auto-reduced. On the other hand, using the ordering (6.6b), we have already seen that g is not reduced with respect to f and the set $\{f, g\}$ is not auto-reduced. The following definition generalizes the concept of ranking to auto-reduced sets.

Definition 6.7 (Ranking). Let $\mathcal{A} = \{A_1, \dots, A_m\}$ and $\mathcal{B} = \{B_1, \dots, B_n\}$ be two ordered auto-reduced sets. Then \mathcal{A} is said to have a lower *ranking* than \mathcal{B} if either there exists an integer k such that $1 \leq k \leq \min(m, n)$ satisfying

$$\begin{aligned} A_j &\sim B_j, & j = 1, \dots, k-1 \\ A_k &\prec B_k, \end{aligned}$$

or else if $m > n$ and $A_j \sim B_j$ for all $j = 1, \dots, n$.

The ordering (6.6a) applied to the polynomials f, g and h in (6.7) yields two ordered auto-reduced sets of polynomials, namely $\mathcal{A} \triangleq \{f, g\}$ and $\mathcal{B} \triangleq \{h, g\}$. Since $h \prec f$, but not vice versa, it holds that \mathcal{B} is ranked lower than \mathcal{A} . The concept of reduced polynomials can be generalized as follows.

Definition 6.8 (Characteristic set). A *characteristic set* for a given set of time-shifted polynomials is an auto-reduced subset such that no other auto-reduced subset is ranked lower.

Using the same example of ordered auto-reduced sets $\mathcal{A} \triangleq \{f, g\}$ and $\mathcal{B} \triangleq \{h, g\}$ chosen from the set $\{f, g, h\}$ of polynomials (6.7), it is clear that \mathcal{B} is the characteristic set under the ordering (6.6a). The basic idea of Ritt's algorithm is to reduce a set of polynomials by the use of characteristic sets. In each iteration, the characteristic set is used to reduce the highest ranked polynomial not in the characteristic set to a lower ranked one. Thus, it is important to guarantee that a sequence with decreasing rank is always finite for the algorithm to terminate in a finite number of steps.

Lemma 6.5

A sequence of characteristic sets, each one ranked lower than the preceding one, can only have finite length.

The proof of this statement is a direct consequence of the following result.

Lemma 6.6

A sequence of time-shifted variables from a finite number of signals, each one ranked lower than the preceding one, can only have finite length.

Proof: Let y_1, \dots, y_p denote all the variables whose time shifts appear anywhere in the sequence. For each y_j let σ_j denote the order of the first appearing time shift. There can then be only σ_j lower time-shifted y_j in the sequence. The total number of elements is thus bounded by $\sigma_1 + \dots + \sigma_p + p$. \square

Finally, we are ready to state the elimination algorithm. Consider a system of polynomial equations on the form

$$f_1 = 0, \dots, f_n = 0, \tag{6.15}$$

where the f_i are polynomials in the variables (y, u, x, θ) and their time-shifts. The elimination procedure is given in Algorithm 6.

Algorithm 6 Ritt-Seidenberg.

Input: Two sets of polynomials $\mathcal{F} = \{f_1, \dots, f_n\}$ and $\mathcal{G} = \emptyset$ with an ordering.

Output: The updated set \mathcal{F} as a characteristic set containing the reduced polynomials and the set \mathcal{G} containing information about the separants, initials and the quotients resulting from the performed pseudo-divisions in the different steps.

- 1) Compute a characteristic set $\mathcal{A} = \{A_1, \dots, A_p\}$ of \mathcal{F} : Order the polynomials in \mathcal{F} according to their leaders, so that the first polynomial has the lowest ordered leader and initialize $\mathcal{A} \leftarrow \{f_1\}$. For all $f_i \in \mathcal{F}$, $i = 2, \dots, n$, test if $\mathcal{A} \cup \{f_i\}$ is auto-reduced and in that case update $\mathcal{A} \leftarrow \mathcal{A} \cup \{f_i\}$.
- 2) If $\mathcal{F} \setminus \mathcal{A} \neq \emptyset$, where \setminus denotes the set difference, then go to Step 4.
- 3) Add S_A, I_A for all $A \in \mathcal{A}$ to \mathcal{G} and **stop**.
- 4) Let f be the highest ranked unreduced polynomial in \mathcal{F} with respect to \mathcal{A} and apply Lemma 6.4 to get polynomials Q and R such that

$$Qf = \bar{Q}A^{(\sigma)} + R,$$

where A is the highest ordered polynomial in \mathcal{A} such that f is not reduced with respect to A . Update $\mathcal{G} \leftarrow \mathcal{G} \cup \{Q\}$.

- 5) If $R = 0$ update $\mathcal{F} \leftarrow \mathcal{F} \setminus \{f_k\}$, else $\mathcal{F} \leftarrow (\mathcal{F} \setminus \{f_k\}) \cup \{R\}$ and continue from Step 1.
-

Algorithm 6 will be referred to as Ritt's algorithm from here on. Some important properties of the proposed algorithm are given below. The proofs of the results are similar to corresponding proofs for the continuous-time case, but with the small distinction remarked upon in connection with Lemma 6.3.

Theorem 6.1

*The Algorithm will reach the **stop** after a finite number of steps.*

Proof: The only possible loop is via Step 5 to Step 1. This involves either the removal of a polynomial or its replacement with one that is reduced with respect to \mathcal{A} or has its highest unreduced time-shift removed. If R is reduced, then it is possible to construct a lower auto-reduced set. An infinite loop would thus contradict either Lemma 6.5 or Lemma 6.6. \square

Theorem 6.2

Every solution to the initial set \mathcal{F} of the algorithm is also a solution to the final set. Every solution to the final set for which the polynomials of \mathcal{G} are nonzero is also a solution of the initial set.

Proof: Follows from a repeated application of Lemma 6.4. \square

We conclude this section by repeating Example 6.1, but here the parameter nonlinearity is eliminated using Ritt's algorithm.

Example 6.2: (Example 6.1 revisited)

Consider the problem of estimating the parameter θ in the model

$$y(t) = \theta u(t) + \theta^2 u(t-1) + e(t), \quad (6.16)$$

given input and output data. This problem could be solved by defining $\tilde{\theta} = \theta^2$ and solving a least-squares problem, but let us consider using Ritt's algorithm. Define

$$f_1 \triangleq y - u\theta - \dot{u}\theta^2 - e, \quad (6.17)$$

$$f_2 \triangleq \theta - \dot{\theta}, \quad (6.18)$$

where the dot operator denotes the backward shift. Let $\mathcal{F} \triangleq \{f_1, f_2\}$ and $\mathcal{G} = \emptyset$, respectively. With the ordering

$$u^{(\cdot)} \prec y^{(\cdot)} \prec e^{(\cdot)} \prec \theta^{(\cdot)},$$

the algorithm yields:

Iteration 1. The leaders ℓ_{f_1} and ℓ_{f_2} in \mathcal{F} are θ and $\dot{\theta}$, respectively. Since $\ell_{f_1} \prec \ell_{f_2}$, f_1 is reduced with respect to f_2 . The other way around, since $\ell_{f_1} = \dot{\theta}$, f_2 is not reduced with respect to f_1 , that is, the largest auto-reduced set \mathcal{A} is given by $\{f_1\}$. The division algorithm yields $Q_1 f_2 = \bar{Q}_1 \dot{f}_1 + R_1$, with

$$R_1 = -\dot{y} + \dot{e} + \dot{u}\theta + \ddot{u}\theta\theta,$$

$Q_1 = \dot{u} + \ddot{u}\theta$, and $\bar{Q}_1 = 1$. Since R_1 still depends on $\dot{\theta}$, yet another division is needed, that is, $Q_2 f_2 = \bar{Q}_2 R_1 + R_2$, with

$$R_2 = \dot{y} - \dot{e} - \dot{u}\theta - \ddot{u}\theta^2,$$

$Q_2 = -\ddot{u}\theta$, and $\bar{Q}_2 = 1$. Putting it all together yields

$$Q f_2 = \bar{Q} \dot{f}_1 + R,$$

where $Q = Q_2 - Q_1 \bar{Q}_2$, $\bar{Q} = -\bar{Q}_1 \bar{Q}_2$, and $R = R_2$. Finally, update \mathcal{G} and set $\mathcal{F} = \{f_1, f_3\}$ where $f_3 \triangleq R$.

Iteration 2. The leaders in \mathcal{F} are now θ and θ , respectively. Thus, $\mathcal{A} = \{f_1\}$ (or $\{f_3\}$) is the characteristic set of \mathcal{F} . Division yields $\bar{Q} f_3 = Q f_1 + R$, where

$$R = -\ddot{u}y + \ddot{u}e + \dot{u}\dot{y} - \dot{u}\dot{e} + (\ddot{u}u - \dot{u}^2)\theta,$$

$\bar{Q} = \dot{u}$, and $Q = \ddot{u}$. Update \mathcal{G} and set $\mathcal{F} = \{f_1, f_4\}$ with $f_4 \triangleq R$.

Thus, one of the equations that Ritt's algorithm finds is

$$\ddot{u}y - \dot{u}\dot{y} = (\ddot{u}u - \dot{u}^2)\theta + \ddot{u}e - \dot{u}\dot{e}, \quad (6.19)$$

which coincides with the result (6.4) in Example 6.1. The algorithm continues for another two iterations, until only one element of \mathcal{F} depends on the parameter θ , but these steps are not presented here.

It is worth noting that in the first iteration, two divisions had to be made. This is one of the differences between the discrete-time and the continuous-time Ritt's algorithm. In continuous-time, the derivative always leaves a polynomial which is affine in the leader and only one division is needed in each iteration.

In Ljung and Glad (1994), the differential algebraic tools were used to derive some interesting results on the identifiability of polynomial systems. Since the discrete-time algorithm is so similar, one could expect to get the same results.

6.3 Identifiability

In Ljung and Glad (1994), necessary and sufficient conditions for global identifiability of continuous-time model structures were given (see Section 3.5). Here, we will discuss the generalization of these results for discrete-time systems. First, let us recall the definition of global identifiability (see also Section 3.2.4).

Definition 6.9. A model structure \mathcal{M} is *globally identifiable* at $\theta^* \in \mathcal{D}_{\mathcal{M}}$ if $\mathcal{M}(\theta) = \mathcal{M}(\theta^*)$ for some $\theta \in \mathcal{D}_{\mathcal{M}}$ implies that $\theta = \theta^*$.

Now, since the discrete-time version of Ritt's algorithm is so similar to the continuous-time case, should we not be able to derive equivalent results concerning the identifiability of discrete-time model structures? That is, is a discrete-time model structure, only containing polynomial nonlinearities, globally identifiable if and only if it can be written as a linear regression model?

The first part of this statement is, at this point, quite easy to prove. Namely, if Ritt's algorithm results in a linear regression model, then the corresponding model structure is globally identifiable.

Theorem 6.3 (Global identifiability)

If the output of Algorithm 6 contains an expression of the form $Q\theta - P$, where the diagonal matrix Q and the vector P does not depend on θ , then θ is globally identifiable, provided that $\det Q \neq 0$ for the measured data².

Proof: Since, according to Theorem 6.2, every solution of the original equations is also a solution of the output equations, it follows that there can only be one value of θ that is consistent with the measured values, provided that $\det Q \neq 0$. \square

Now, let us consider the converse of the above statement. In the continuous-time case, the proof of this fact is based on the following property: If $f(t)$ and $g(t)$ are analytical functions satisfying $f(t)g(t) = 0$ for all $t \in \mathbb{R}$, it holds that either $f(t) = 0$ or $g(t) = 0$ for all $t \in \mathbb{R}$. Unfortunately, this property does not remain valid when the domain changes

²The requirement $\det Q \neq 0$ can be interpreted as a condition of *persistence of excitation* of the input signal (see, for instance, Ljung, 1999).

from \mathbb{R} to \mathbb{Z} . A simple example of a discrete-time signal that does not satisfy the above property is the Kronecker delta function

$$\delta(t) \triangleq \begin{cases} 1, & t = 0 \\ 0, & t \in \mathbb{Z} \setminus \{0\} \end{cases}$$

For instance, it holds that $\delta(t)\delta(t-1) = 0$ for all $t \in \mathbb{Z}$ even though $\delta(t)$ and $\delta(t-1)$ are nonzero for $t = 0$ and $t = 1$, respectively. The absence of the above property for discrete-time signals hinders a straightforward generalization of the desired identifiability result. Reasonable assumptions to be able to prove the theorem are yet to be found.

Even though we were not able to provide a general identifiability result, equivalent to that achieved in the continuous-time case, we are still able to draw some conclusions when the use of Ritt's algorithm does not result in a linear regression model.

Theorem 6.4 (Unidentifiability)

Let the ranking be given by

$$u^{(\cdot)} \prec y^{(\cdot)} \prec \theta_1^{(\cdot)} \prec \dots \prec \theta_m^{(\cdot)} \prec x^{(\cdot)}$$

where x contains any unmeasured variable. Assume that the output of the algorithm is an auto-reduced set with the following form

$$\begin{aligned} p_0(u, \dot{u}, \dots, y, \dot{y}, \dots), \dot{\theta}_1 - \theta_1, \\ p_2(u, \dot{u}, \dots, y, \dot{y}, \dots, \theta_1, \theta_2), \dots \\ p_m(u, \dot{u}, \dots, y, \dot{y}, \dots, \theta_1, \dots, \theta_m), \dots \end{aligned}$$

Furthermore, assume that there exists a solution to this set such that all polynomials in \mathcal{G} are nonzero. Then there are infinitely many values of θ compatible with the u and y of this solution, that is, the system is unidentifiable.

Proof: Fixing the values of u and y in the first polynomial p_0 to those of the given solution means that the corresponding equation is always satisfied. The parameter θ_1 can now be changed to a new arbitrary constant in the second polynomial p_2 . If this change is small enough the remaining equations can now be solved successively for the leader due to the nonvanishing of the separants (the *implicit function theorem*, see, for instance, Rudin, 1976). \square

The above theorem implies that, in this particular case, if Ritt's algorithm is not able to eliminate the time-shift for at least one of the parameters, not necessarily the first parameter, then the model structure is unidentifiable. Thus, even though we were not able to prove that every globally identifiable model structure may be rewritten as a linear regression model, this indicates that Ritt's algorithm still can be useful when analyzing the identifiability of discrete-time model structures. Ritt's algorithm has some interesting implications for the parameter estimation problem in system identification, since it, in some cases, results in a linear regression.

6.4 Identification Aspects

In this section we will discuss the use of Ritt's algorithm from a parameter estimation perspective. First we discuss some relations to other known methods.

An often used concept in system identification is the methodology of *over-parametrization* (see, for instance, Bai, 1998), which can be described as follows: in a regression problem, where some coefficients are nonlinear functions of the parameters, replace these functions with new parameters. This concept is easiest to illustrate via an example.

Example 6.3: (Over-parametrization)

Again, consider the problem of estimating the parameter θ in

$$y(t) = \theta u(t) + \theta^2 u(t-1) + e(t), \quad (6.20)$$

given a dataset (3.1). This problem can be solved by defining $\theta_1 = \theta$ and $\theta_2 = \theta^2$ and solving the linear regression by least-squares, that is, by using over-parametrization. Since one is only interested in the estimate $\hat{\theta}_1$, one can just ignore the resulting estimate $\hat{\theta}_2$ or reconcile it with the estimate $\hat{\theta}_1$. Now, let us try a different approach. Time shift (6.20)

$$y = \theta u + \theta^2 \dot{u} + e,$$

yielding

$$\dot{y} = \theta \dot{u} + \theta^2 \ddot{u} + \dot{e},$$

keeping in mind that θ is constant. Stacking the equations results the over-parameterized linear regression model

$$\begin{pmatrix} y \\ \dot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} u & \dot{u} \\ \dot{u} & \ddot{u} \end{pmatrix}}_{\triangleq \Phi} \begin{pmatrix} \theta \\ \theta^2 \end{pmatrix} + \begin{pmatrix} e \\ \dot{e} \end{pmatrix}. \quad (6.21)$$

Assuming that the matrix Φ is invertible, one can multiply (6.21) by $(\det \Phi)\Phi^{-1}$ to get

$$\begin{pmatrix} \ddot{u}y - \dot{u}\dot{y} \\ \dot{u}\dot{y} - \ddot{u}y \end{pmatrix} = (\ddot{u}u - \dot{u}^2) \begin{pmatrix} \theta \\ \theta^2 \end{pmatrix} + \begin{pmatrix} \ddot{u}e - \dot{u}\dot{e} \\ \dot{u}\dot{e} - \ddot{u}e \end{pmatrix}.$$

The first equation is exactly the result of Ritt's algorithm (see (6.19)). Thus, there seems to be a connection between over-parametrization and Ritt's algorithm. This possible connection needs to be analyzed further in future work.

Another popular method in elimination theory is the so called Gröbner basis (see, for example, Mishra, 1993), which is mainly used to solve systems of static polynomial equations. Let us consider a simple example.

Example 6.4: (Gröbner basis)

Let us, once again, consider Example 6.3. By time-shifting (6.20) and remembering that θ is constant, we get the following system of equations

$$\begin{aligned} y &= \theta u + \theta^2 \dot{u} + e, \\ \dot{y} &= \theta \dot{u} + \theta^2 \ddot{u} + \dot{e}. \end{aligned}$$

Here, we will consider the variables $(u, \dot{u}, \ddot{u}, y, \dot{y}, e, \dot{e})$ as different static entities. Now, using the Gröbner basis algorithm implemented in MATHEMATICA via the command

$$\text{GroebnerBasis}(\{y - \theta u - \theta^2 \dot{u} - e, \dot{y} - \theta \dot{u} - \theta^2 \ddot{u} + \dot{e}\}, \{\theta\}),$$

results in five different polynomials where

$$\ddot{u}y - \dot{u}\dot{y} = (\ddot{u}u - \dot{u}^2)\theta + \ddot{u}e - \dot{u}\dot{e},$$

is one of them. This is the same linear regression (6.19) as the one derived using Algorithm 6. The advantage of using the Gröbner basis framework, in place of the methods presented here, is that well tested and efficient implementations are available. The drawback is that one needs to know beforehand which variables that are present in the final result, here $(u, \dot{u}, \ddot{u}, y, \dot{y}, e, \dot{e})$. These can be found by trial and error, by adding time-shifted equations to the original until the desired result appears, but to the author's knowledge no one shot method exists. It would also be interesting to investigate if, using the same ordering, the Gröbner basis method and Ritt's algorithm in general yield the same results in cases where the variables appearing in the resulting linear regression model are known.

Now, let us consider a more complex system with two parameters using Ritt's algorithm.

Example 6.5

Consider the problem of estimating the parameters $\theta = (\theta_1, \theta_2)^T$ in

$$y + \theta_1 \dot{y} + \theta_1 \theta_2 \ddot{y} = \theta_2 u + e, \quad (6.22)$$

where the dot denotes backward shift, given input and output data. Running Ritt's algorithm on (6.22) yields the linear regression $\eta = \varphi^T \theta + \nu$ where

$$\eta = \begin{pmatrix} uy^{(3)} - \dot{u}\ddot{y} + \ddot{y}\dot{y} - y^{(3)}y \\ \dot{u}\ddot{y}\dot{y} - u\dot{y}^2 - \ddot{y}\dot{y}^2 + \dot{y}^2y \end{pmatrix}, \quad (6.23a)$$

$$\varphi = \begin{pmatrix} y^{(3)}\dot{y} - \dot{y}^2 & 0 \\ 0 & uy^{(3)}\ddot{y} - \dot{u}\dot{y}^2 + \dot{y}^2\dot{y} - y^{(3)}\ddot{y}y \end{pmatrix}, \quad (6.23b)$$

$$\nu = \begin{pmatrix} \dot{e}\ddot{y} - ey^{(3)} \\ e\dot{y}^2 - \dot{e}\ddot{y}\dot{y} + ey^{(3)}\ddot{y}\theta_2 - \dot{e}\dot{y}^2\theta_2 \end{pmatrix}. \quad (6.23c)$$

Notice that in the expression for θ_1 , there are second order products between the different signals, while in the expression for θ_2 there are exclusively third order products. This implies that it will be more difficult, in some sense, to estimate the second parameter than the first parameter. Changing the ordering of the parameters when using the algorithm will yield the opposite result.

Now, let us try to do the elimination differently. As before, time shift (6.22)

$$\dot{y} + \theta_1 \ddot{y} + \theta_1 \theta_2 y^{(3)} = \theta_2 \dot{u} + \dot{e}, \quad (6.24)$$

keeping in mind that the parameters are constant in time. Multiplying (6.22) by $y^{(3)}$, (6.24) by \ddot{y} and subtracting the results, the following holds

$$yy^{(3)} - \dot{y}\ddot{y} = \begin{pmatrix} \dot{y}^2 - \dot{y}y^{(3)} \\ uy^{(3)} - \dot{u}\ddot{y} \end{pmatrix}^T \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + ey^{(3)} - \dot{e}\ddot{y}. \quad (6.25)$$

This linear regression shares the element of the regressor vector corresponding to the first parameter with (6.23), where now only second order products in the element corresponding to the second parameter exist. This is due to the fact that Ritt's algorithm continues until it has one equation for each parameter. So the question that arises is, can Ritt's algorithm be modified to take these things into account? There are some possibilities:

- 1) What enables the calculations above is that all the parameters already occur linearly in the original equation (6.22). Thus, the only thing left to eliminate is the single parameter nonlinearity. Therefore, if the parameter nonlinearities could be ordered higher than the linear occurrences, then one could stop the algorithm prematurely to receive the linear regression given in (6.25).
- 2) Another alternative is to stop the algorithm when the first linear regression is decided, that is, the equation that determines θ_1 . Then one could estimate this parameter and feed it back into the original equation (6.22) and from there estimate θ_2

Further analysis of possible modifications of Ritt's algorithm, to tailor it for system identification, needs to be made.

Use of the elimination schemes presented above, that is, Ritt's algorithm and similar methods, yields linear regression models where the noise is deformed. So when using these equations to estimate the unknown parameters, a tool to deal with these noise transformations is needed. One such tool is the *instrumental variables* (IV) method, see Section 3.3.

Example 6.6

Once again, consider the estimation problem presented in Example 6.5. Depending on the input excitation, this problem could be solved using over-parametrization, see Example 6.3. Despite this, let us try using the result (6.25).

Trying to solve (6.25) directly via the least-squares method will yield a biased estimate of the θ parameters, since the regression vector is correlated with the noise. Thus, instruments, uncorrelated with the noise, need to be chosen.

Now, let us perform a Monte Carlo simulation study of the estimation of the parameters θ via (6.25), with the least-squares (LS) method and the IV method, respectively. The true parameters are given by $\theta = (-0.1 \quad 0.2)^T$. Let the noise and the input be independent white Gaussian processes with zero mean and unit variance. With $M = 100$ Monte Carlo runs and data lengths of 10000 yields

$$\frac{1}{M} \sum_{k=1}^M \hat{\theta}_k^{\text{LS}} = \begin{pmatrix} -0.0750 \\ 0.1999 \end{pmatrix}, \quad \frac{1}{M} \sum_{k=1}^M \hat{\theta}_k^{\text{IV}} = \begin{pmatrix} -0.0977 \\ 0.2002 \end{pmatrix}, \quad (6.26)$$

where \hat{u}^2 has been used as instruments for the first parameter and the regressor as instruments for the second parameter. As predicted, using the LS estimator, it seems that a biased estimate is obtained while the IV estimate appears to be unbiased.

So far, only measured signals, except the additive noise, have appeared. Below, the pa-

parameter estimation of a nonlinear state-space model is considered. Here, the unknown states need to be eliminated completely, and therefore they are ordered higher than any time shift of the parameters, that is, the following ordering is used

$$u^{(\cdot)} \prec y^{(\cdot)} \prec \theta_1^{(\cdot)} \prec \theta_2^{(\cdot)} \prec x^{(\cdot)}.$$

Example 6.7

Consider the problem of estimating the parameters $\theta = (\theta_1 \ \theta_2)^T$ in the nonlinear state-space model

$$x_1(t+1) = \theta_1 x_1(t) + \theta_2 x_2(t) + u^2(t), \quad (6.27a)$$

$$x_2(t+1) = x_1^2(t), \quad (6.27b)$$

$$y(t) = x_1(t) + e(t). \quad (6.27c)$$

Using Ritt's algorithm yields quite lengthy results and below we only present the linear regression equation $\eta_1 = \varphi_1 \theta_1 + \nu_1$ for the first parameter, where

$$\begin{aligned} \eta_1 &= \ddot{u}^2 \ddot{y}^2 - \dot{u}^2 (y^{(3)})^2 - \ddot{y}^2 \dot{y} + (y^{(3)})^2 y, \\ \varphi_1 &= (y^{(3)})^2 \dot{y} - \ddot{y}^3, \end{aligned}$$

and

$$\begin{aligned} \nu_1 &= (e^{(3)})^2 e - \ddot{e}^2 \dot{e} - \ddot{e}^2 \ddot{u}^2 + (e^{(3)})^2 \dot{u}^2 - 2e^{(3)} e y^{(3)} \\ &\quad - 2e^{(3)} \dot{u}^2 y^{(3)} + e (y^{(3)})^2 + 2\ddot{e} \dot{e} \ddot{y} + 2\ddot{e} \ddot{u}^2 \ddot{y} - \dot{e} \ddot{y}^2 \\ &\quad - (e^{(3)})^2 y + 2e^{(3)} y^{(3)} y + \ddot{e}^2 \dot{y} - 2\ddot{e} \dot{y} \dot{y} \\ &\quad + \left[\dot{e}^3 - (e^{(3)})^2 \dot{e} + 2e^{(3)} \dot{e} y^{(3)} - \dot{e} (y^{(3)})^2 \right. \\ &\quad \left. - 3\ddot{e}^2 \dot{y} + 3\ddot{e} \ddot{y}^2 + (e^{(3)})^2 \dot{y} - 2e^{(3)} y^{(3)} \dot{y} \right] \theta_1 \end{aligned}$$

We notice that the linear regression becomes quite complicated, but the main complexity lies in the transformed noise. This means that the main problem lies in choosing the appropriate instruments. If such instruments can be found, these equations enable a way to get an initial estimate of the first parameter, which later can be refined using the PEM.

Finding good instruments when using the resulting linear regression of Ritt's algorithm seems difficult. Thus, an automatic method for finding instruments is needed if the method should be practically applied.

6.5 Discussion

In this chapter a discrete-time version of Ritt's algorithm, similar to the one given in continuous-time, has been presented. The difference lies in the number of pseudo-divisions needed to reduce the polynomials (see the discussion following Lemma 6.3). Yet another deviation from the continuous-time case became apparent when the generalization

of the identifiability results presented in Ljung and Glad (1994) was attempted. In the discrete-time case, only parts of these results could be provided, due to the lack of certain properties of analytic functions utilized in the continuous-time case which was shown by a simple example not to hold for discrete-time signals (see Section 6.3).

In Section 6.4 certain aspects of Ritt's algorithm as a tool for finding equations that simplify the finding of initial estimates of certain nonlinear model structures were analyzed. It turns out that the generalization to discrete-time enables the possibility of dealing with noise if one is able to find instruments which can deal with the transformed noise model. Furthermore, it was shown through examples that Ritt's algorithm goes unnecessary far in the algebraic manipulations. The result is a linear regression model containing higher degree polynomials than necessary.

In the future, it would be interesting to examine the possibilities to tailor Ritt's algorithm for system identification purposes. Also, the connection between over-parametrization, Gröbner basis methods and Ritt's algorithm, as implicated in the Examples 6.3 and 6.4, needs further thought. Furthermore, if the resulting linear regression model should be used for parameter estimation, how should one choose the instruments to get unbiased estimates? Also, for which model classes can one guarantee that the application of the discrete-time Ritt's algorithm results in a linear regression model? In such cases, is it possible to determine if the least-squares estimate will be unbiased? These are all open questions for future research.

Bibliography

- H. Akaike. Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21(2):243–247, 1969.
- E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209:237–260, 1998.
- E. Bai. An optimal two-stage identification algorithm for Hammerstein-Wiener nonlinear systems. *Automatica*, 34(3):333–338, 1998.
- D. Bauer. Estimating linear dynamical systems using subspace methods. *Econometric Theory*, 21:181–211, 2005.
- D. Bauer. Estimating ARMAX systems for multivariate time series using the state approach to subspace algorithms. *Journal of Multivariate Analysis*, 100:397–421, 2009.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.
- G. Casella and R. L. Berger. *Statistical Inference*. Duxbury, 2nd edition, 2002.
- S. Diop. Elimination in control theory. *Mathematics of Control, Signals and Systems*, 4(1):17–32, 1991.
- N. Draper and H. Smith. *Applied Regression Analysis*. Wiley, 2nd edition, 1981.
- J. Durbin. Efficient estimation of parameters in moving-average models. *Biometrika*, 46:306–316, 1959.

- B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–451, 2004.
- M. Fliess and T. Glad. An algebraic approach to linear and nonlinear control. In *Essays on Control: Perspectives in the Theory and Its Applications*, volume 14 of *Progress in System and Control Theory*, pages 190–198. Birkhäuser, 1994.
- G. M. Furnival and R. W. Wilson. Regressions by leaps and bounds. *Technometrics*, 16(4):499–511, 1974.
- H. Garnier and L. Wang. *Identification of Continuous-time Models from Sampled Data*. Springer, 2008.
- K. F. Gauss. *Teoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium*. Reprinted Translation: *Theory of the Motion of the Heavenly Bodies Moving About the Sun in Conic Sections*. Dover, 1809.
- T. Glad. Solvability of differential algebraic equations and inequalities: An algorithm. In *Proceedings European Control Conference*, Brussels, Belgium, 1997.
- G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.
- N. I. M. Gould. An algorithm for large-scale quadratic programming. *IMA Journal of Numerical Analysis*, 11:299–324, 1991.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.
- B. Haverkamp. *State Space Identification, Theory and Practice*. PhD thesis, Technische Universiteit Delft, Delft, The Netherlands, 2001.
- T. Hesterberg, N. H. Choi, L. Meier, and C. Fraley. Least angle and ℓ_1 penalized regression: A review. *Statistics Surveys*, 2:61–93, 2008.
- A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.
- M. Jansson and B. Wahlberg. A linear regression approach to state-space subspace system identification. *Signal Processing*, 52:103–129, 1996.
- H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition, 2002.
- E. Kolchin. *Differential Algebra and Algebraic Groups*. Academic Press, 1973.
- S. Kotsios. An application of Ritt’s remainder algorithm to discrete polynomial control systems. *IMA Journal of Mathematical Control and Information*, 18(18):19–29, 2001.
- S. Lang. *Algebra*. Springer-Verlag, 3rd revised edition, 2002.
- L. Ljung. *System Identification: Theory for the User*. Prentice Hall, second edition, 1999.
- L. Ljung. *System Identification Toolbox, User’s Guide*. MathWorks, 7 edition, 2009.

- L. Ljung and T. Glad. On global identifiability for arbitrary model parametrizations. *Automatica*, 30(2):265–276, 1994.
- C. Lyzell and G. Hovland. Verification of the dynamics of the 5-DOF Gantry-Tau parallel kinematic machine. In *Proceedings of Robotics and Applications and Telematics*, Würzburg, Germany, August 2007.
- C. Lyzell, J. Roll, and L. Ljung. The use of nonnegative garrote for order selection of ARX models. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008.
- C. Lyzell, M. Enqvist, and L. Ljung. Handling certain structure information in subspace identification. In *Preprints of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009a.
- C. Lyzell, T. Glad, M. Enqvist, and L. Ljung. Identification aspects of Ritt’s algorithm for discrete-time systems. In *Preprints of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009b.
- T. McKelvey and H. Akçay. An efficient frequency domain state-space identification algorithm. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, pages 373–378, Lake Buena Vista, Florida, 1994.
- B. Mishra. *Algorithmic Algebra*. Texts and Monographs in Computer Science. Springer-Verlag, 1993.
- S. Niu, L. Ljung, and Å. Björck. Decomposition methods for solving least-squares parameter estimation. *IEEE Transactions on Signal Processing*, 44(11):2847–2852, 1996.
- J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.
- G. P. Rao and H. Garnier. Numerical illustrations of the relevance of direct continuous-time model identification. In *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, July 2002.
- O. Reiersøl. Confluence analysis by means of lag moments and other methods of confluence analysis. *Econometrica*, 9:1–23, 1941.
- E. Reynders and G. De Roeck. Subspace identification of (AR)ARMAX, Box-Jenkins, and generalized model structures. In *Preprints of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009.
- J. Rissanen. Modelling by shortest data description. *Automatica*, 14:175–182, 1978.
- J. F. Ritt. *Differential Algebra*. Dover Publications, Inc., 1950.
- J. Roll. Piecewise linear solution paths with application to direct weight optimization. *Automatica*, 44(11):2745–2753, 2008.
- O. Romanko. An interior point approach to quadratic and parametric quadratic optimization. Master’s thesis, McMaster University, Hamilton, Ontario, 2004.

- S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *Anal. of Statistics*, 35(3):1012–1030, 2007.
- W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition, 1976.
- B. Savas and D. Lindgren. Rank reduction and volume minimization approach to state-space subspace system identification. *Signal Processing*, 86(11):3275–3285, 2006.
- T. Söderström. On the uniqueness of maximum likelihood identification. *Automatica*, 11:193–197, 1975.
- T. Söderström and P. Stoica. *Instrumental Variable Methods for System Identification*. Springer, 1980.
- T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 1989.
- M. Stein, J. Branke, and M. Schmeck. Efficient implementation of an active set algorithm for large-scale portfolio selection. *Computers & Operations Research*, 35:3945–3961, 2008.
- P. Stoica and R. Moses. *Spectral Analysis of Signals*. Prentice Hall, 2005.
- R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.
- P. Tøndel, T. A. Johansen, and A. Bemporad. Further results on multiparametric quadratic programming. In *Conference on Decision and Control*, pages 3173–3178, Maui, Hawaii, 2003.
- R. Tóth. *Modeling and Identification of Linear Parameter-Varying Systems, an Orthonormal Basis Function Approach*. PhD thesis, Technische Universiteit Delft, Delft, The Netherlands, 2008.
- R. Tóth, C. Lyzell, M. Enqvist, P. Heuberger, and P. Van den Hof. Order and structural dependence selection of LPV-ARX models using a nonnegative garrote approach. In *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009.
- P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems*. Kluwer Academic Publishers, 1996a.
- P. Van Overschee and B. De Moor. Continuous-time frequency domain subspace system identification. *Signal Processing*, 52(5):179–194, 1996b.
- M. Verhaegen and V. Verdult. *Filtering and System Identification*. Cambridge University Press, 2007.
- M. Viberg, B. Wahlberg, and B. Ottersten. Analysis of state space system identification methods based on instrumental variables and subspace fitting. *Automatica*, 33(9):1603–1616, 1997.

- L. L. Xie and L. Ljung. Estimate physical parameters by black-box modeling. In *Proceedings of the 21st Chinese Control Conference*, pages 673–677, Hangzhou, China, August 2002.
- M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.
- M. Yuan, V. R. Joseph, and H. Zou. Structured variable selection and estimation. Technical report, Georgia Institute of Technology, Atlanta, Georgia, 2007. To appear in *Annals of Applied Statistics*.
- H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

Licentiate Theses
Division of Automatic Control
Linköping University

- P. Andersson:** Adaptive Forgetting through Multiple Models and Adaptive Control of Car Dynamics. Thesis No. 15, 1983.
- B. Wahlberg:** On Model Simplification in System Identification. Thesis No. 47, 1985.
- A. Isaksson:** Identification of Time Varying Systems and Applications of System Identification to Signal Processing. Thesis No. 75, 1986.
- G. Malmberg:** A Study of Adaptive Control Missiles. Thesis No. 76, 1986.
- S. Gunnarsson:** On the Mean Square Error of Transfer Function Estimates with Applications to Control. Thesis No. 90, 1986.
- M. Viberg:** On the Adaptive Array Problem. Thesis No. 117, 1987.
- K. Ståhl:** On the Frequency Domain Analysis of Nonlinear Systems. Thesis No. 137, 1988.
- A. Skeppstedt:** Construction of Composite Models from Large Data-Sets. Thesis No. 149, 1988.
- P. A. J. Nagy:** MaMiS: A Programming Environment for Numeric/Symbolic Data Processing. Thesis No. 153, 1988.
- K. Forsman:** Applications of Constructive Algebra to Control Problems. Thesis No. 231, 1990.
- I. Klein:** Planning for a Class of Sequential Control Problems. Thesis No. 234, 1990.
- F. Gustafsson:** Optimal Segmentation of Linear Regression Parameters. Thesis No. 246, 1990.
- H. Hjalmarsson:** On Estimation of Model Quality in System Identification. Thesis No. 251, 1990.
- S. Andersson:** Sensor Array Processing; Application to Mobile Communication Systems and Dimension Reduction. Thesis No. 255, 1990.
- K. Wang Chen:** Observability and Invertibility of Nonlinear Systems: A Differential Algebraic Approach. Thesis No. 282, 1991.
- J. Sjöberg:** Regularization Issues in Neural Network Models of Dynamical Systems. Thesis No. 366, 1993.
- P. Pucar:** Segmentation of Laser Range Radar Images Using Hidden Markov Field Models. Thesis No. 403, 1993.
- H. Fortell:** Volterra and Algebraic Approaches to the Zero Dynamics. Thesis No. 438, 1994.
- T. McKelvey:** On State-Space Models in System Identification. Thesis No. 447, 1994.
- T. Andersson:** Concepts and Algorithms for Non-Linear System Identifiability. Thesis No. 448, 1994.
- P. Lindskog:** Algorithms and Tools for System Identification Using Prior Knowledge. Thesis No. 456, 1994.
- J. Plantin:** Algebraic Methods for Verification and Control of Discrete Event Dynamic Systems. Thesis No. 501, 1995.
- J. Gunnarsson:** On Modeling of Discrete Event Dynamic Systems, Using Symbolic Algebraic Methods. Thesis No. 502, 1995.
- A. Ericsson:** Fast Power Control to Counteract Rayleigh Fading in Cellular Radio Systems. Thesis No. 527, 1995.
- M. Jirstrand:** Algebraic Methods for Modeling and Design in Control. Thesis No. 540, 1996.
- K. Edström:** Simulation of Mode Switching Systems Using Switched Bond Graphs. Thesis No. 586, 1996.
- J. Palmqvist:** On Integrity Monitoring of Integrated Navigation Systems. Thesis No. 600, 1997.
- A. Stenman:** Just-in-Time Models with Applications to Dynamical Systems. Thesis No. 601, 1997.
- M. Andersson:** Experimental Design and Updating of Finite Element Models. Thesis No. 611, 1997.
- U. Forssell:** Properties and Usage of Closed-Loop Identification Methods. Thesis No. 641, 1997.

M. Larsson: On Modeling and Diagnosis of Discrete Event Dynamic systems. Thesis No. 648, 1997.

N. Bergman: Bayesian Inference in Terrain Navigation. Thesis No. 649, 1997.

V. Einarsson: On Verification of Switched Systems Using Abstractions. Thesis No. 705, 1998.

J. Blom, F. Gunnarsson: Power Control in Cellular Radio Systems. Thesis No. 706, 1998.

P. Spångéus: Hybrid Control using LP and LMI methods – Some Applications. Thesis No. 724, 1998.

M. Norrlöf: On Analysis and Implementation of Iterative Learning Control. Thesis No. 727, 1998.

A. Hagenblad: Aspects of the Identification of Wiener Models. Thesis No. 793, 1999.

F. Tjärnström: Quality Estimation of Approximate Models. Thesis No. 810, 2000.

C. Carlsson: Vehicle Size and Orientation Estimation Using Geometric Fitting. Thesis No. 840, 2000.

J. Löfberg: Linear Model Predictive Control: Stability and Robustness. Thesis No. 866, 2001.

O. Härkegård: Flight Control Design Using Backstepping. Thesis No. 875, 2001.

J. Elbornsson: Equalization of Distortion in A/D Converters. Thesis No. 883, 2001.

J. Roll: Robust Verification and Identification of Piecewise Affine Systems. Thesis No. 899, 2001.

I. Lind: Regressor Selection in System Identification using ANOVA. Thesis No. 921, 2001.

R. Karlsson: Simulation Based Methods for Target Tracking. Thesis No. 930, 2002.

P.-J. Nordlund: Sequential Monte Carlo Filters and Integrated Navigation. Thesis No. 945, 2002.

M. Östring: Identification, Diagnosis, and Control of a Flexible Robot Arm. Thesis No. 948, 2002.

C. Olsson: Active Engine Vibration Isolation using Feedback Control. Thesis No. 968, 2002.

J. Jansson: Tracking and Decision Making for Automotive Collision Avoidance. Thesis No. 965, 2002.

N. Persson: Event Based Sampling with Application to Spectral Estimation. Thesis No. 981, 2002.

D. Lindgren: Subspace Selection Techniques for Classification Problems. Thesis No. 995, 2002.

E. Geijer Lundin: Uplink Load in CDMA Cellular Systems. Thesis No. 1045, 2003.

M. Enqvist: Some Results on Linear Models of Nonlinear Systems. Thesis No. 1046, 2003.

T. Schön: On Computational Methods for Nonlinear Estimation. Thesis No. 1047, 2003.

F. Gunnarsson: On Modeling and Control of Network Queue Dynamics. Thesis No. 1048, 2003.

S. Björklund: A Survey and Comparison of Time-Delay Estimation Methods in Linear Systems. Thesis No. 1061, 2003.

M. Gerdin: Parameter Estimation in Linear Descriptor Systems. Thesis No. 1085, 2004.

A. Eidehall: An Automotive Lane Guidance System. Thesis No. 1122, 2004.

E. Wernholt: On Multivariable and Nonlinear Identification of Industrial Robots. Thesis No. 1131, 2004.

J. Gillberg: Methods for Frequency Domain Estimation of Continuous-Time Models. Thesis No. 1133, 2004.

G. Hendeby: Fundamental Estimation and Detection Limits in Linear Non-Gaussian Systems. Thesis No. 1199, 2005.

D. Axehill: Applications of Integer Quadratic Programming in Control and Communication. Thesis No. 1218, 2005.

J. Sjöberg: Some Results On Optimal Control for Nonlinear Descriptor Systems. Thesis No. 1227, 2006.

D. Törnqvist: Statistical Fault Detection with Applications to IMU Disturbances. Thesis No. 1258, 2006.

H. Tidefelt: Structural algorithms and perturbations in differential-algebraic equations. Thesis No. 1318, 2007.

- S. Moberg:** On Modeling and Control of Flexible Manipulators. Thesis No. 1336, 2007.
- J. Wallén:** On Kinematic Modelling and Iterative Learning Control of Industrial Robots. Thesis No. 1343, 2008.
- J. Harju Johansson:** A Structure Utilizing Inexact Primal-Dual Interior-Point Method for Analysis of Linear Differential Inclusions. Thesis No. 1367, 2008.
- J. D. Hol:** Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors. Thesis No. 1370, 2008.
- H. Ohlsson:** Regression on Manifolds with Implications for System Identification. Thesis No. 1382, 2008.
- D. Ankelhed:** On low order controller synthesis using rational constraints. Thesis No. 1398, 2009.
- P. Skoglar:** Planning Methods for Aerial Exploration and Ground Target Tracking. Thesis No. 1420, 2009.
- C. Lundquist:** Automotive Sensor Fusion for Situation Awareness. Thesis No. 1422, 2009.