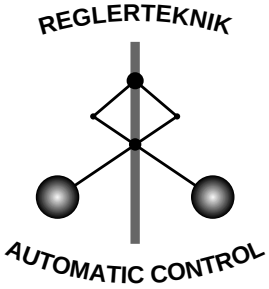
Linköping Studies in Science and Technology
Thesis No. 810

Quality Estimation of Approximate Models

Fredrik Tjärnström



Division of Automatic Control
Department of Electrical Engineering
Linköpings universitet, SE–581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
Email: fredrikt@isy.liu.se

Linköping 2000

Quality Estimation of Approximate Models

Department of Electrical Engineering,
Linköpings universitet,
SE–581 83 Linköping,
Sweden.

ISBN 91-7219-671-8
ISSN 0280-7971
LiU-TEK-LIC-2000:06

Printed by UniTryck, Linköping, Sweden 2000

To Malin

Abstract

This thesis discusses three different topics: model error modeling, bootstrap, and model reduction. These subjects may at first sight seem to be quite far away from each other. However, there are some connections between them, the most important one being uncertainty estimation.

Model error modeling is actually a tool for model validation. The idea is to construct a model of the model errors that are present in the nominal model, and present them in an easily interpreted way. When the error models are linear, we prefer to present the result in the frequency domain. We discuss different ways of estimating such models, as well as how the "size" of such models should be presented and interpreted. Examples illustrate how some model errors could be accepted although they may be large. This is partly in contrast with traditional model validation tools, that more have the character of telling whether we have any model errors or not.

In some situations it is very difficult to calculate the uncertainties present in an estimate. One would therefore like to repeat the experiment several times to get better knowledge about it. Bootstrap mimics this, since it simulates new data from the original sample and thus makes it possible to repeat a similar experiment again. We describe how bootstrap can be used in a system identification experiment. The most interesting results are that we are able to estimate the variance error of undermodeled models and that it is possible to construct several confidence regions where we are in control of the simultaneous confidence degree (this is, regions which *all* cover their respective parameters with a certain confidence degree).

The last chapter is focused on quantifying the variance reduction that occurs in model reduction. We specifically look at L_2 model reduction and show that estimating the model in two steps, first a high order model which is then subjected to L_2 model reduction, in some situations give the same variance as estimating the model directly. We also show that it might even be better to estimate the model in two steps in some specific cases. From the calculations of these results it also follows that L_2 model reduction is optimal in reducing the variance of the estimate.

Acknowledgments

First of all, I would like to thank my supervisor Prof. Lennart Ljung for the excellent guidance throughout the work on this thesis. His enormous knowledge and intuition in the field of system identification has given me lots of ideas and inspiration.

I also would like to thank Lennart for letting me join the Automatic Control group here at Linköping University in the first place. All the members together form an extremely skilful and industrious group of people. Some of them deserve extra credit. All you people who have taken your time to discuss and explain things for me that I have not really understood, thank you all. Mattias Olofsson who kept the computer system running, Niclas Bergman and Anders Stenman who maintained the XEmacs and LaTeX installations. Moreover, Ulla Salaneck deserves a lot of gratitude for being helpful and patient with all my (and my colleagues') questions regarding all sorts of practical things.

Several people have helped me more specifically during this work. Dr. Anders Stenman, Ola Härkegård and Jacob Roll read earlier versions of the manuscript and gave me lots of valuable comments. Dr. Dietmar Bauer gave insightful comments on the method to obtain simultaneous confidence degree using bootstrap. Dr. Liang-Liang Xie and Jacob Roll endured a lot of interruption from me discussing the proof of Theorem 6.1. Your comments and ideas simplified and helped a lot.

This work was supported by the graduate school ECSEL, which is gratefully acknowledged.

I would like to thank my parents, Margareta and Gunnar, and my sister Mari for their love and support, and my dog Chica for *always* being happy to see me, no matter in what mood I am.

Finally, I would like to thank Malin for giving me love, support and encouragement every day of the year. I love you.

Linköping, February 2000

Fredrik Tjärnström

Contents

Notation

Symbols

$(A)_{i,j}$	the (i, j)th element of A.
$(A)_{\cdot,j}$	the jth column of A.
$e(t)$	disturbance variable at time t (usually white noise).
$G(q)$	transfer function from u to y.
$G(q,\theta)$	transfer function parameterized by θ.
$\hat{G}_N(q)$	transfer function estimate, $\hat{G}_N(q) = G(q,\hat{\theta}_N)$.
$G_0(q)$	"true" transfer function from u to y for a given system.
$H(q)$	transfer function from e to y.
P_θ	covariance matrix of θ.
$R_s(k)$	$\bar{\mathrm{E}}s(t)s^T(t-k)$.
$R_{sw}(k)$	$\bar{\mathrm{E}}s(t)w^T(t-k)$.
$u(t)$	input variable at time t.
$U_N(\omega)$	Fourier transform of $\{u(1), \ldots, u(N)\}$.
$v(t)$	disturbance variable at time t (usually filtered white noise).
$V_N(\theta)$	criterion function to be minimized.
$\mathbf{w}$	weighting vector, see (2.65).
x^*	bootstrap resamples of x.

$y(t)$	output variable at time t.
$\hat{y}(t\|\theta)$	predictor of $y(t)$ given θ and Z^{t-1}.
Z^t	set of input and output data up to time t.
$\Delta(q)$	model error transfer function, $G(q) - \hat{G}(q)$.
$\varepsilon(t, \theta)$	prediction errors, $\varepsilon(t, \theta) = y(t) - \hat{y}(t\|\theta)$.
θ	parameter vector of dimension d.
$\hat{\theta}_N$	estimate of θ using N data points.
θ^*	limiting estimate of θ.
θ_0	"true" parameter value.
$\Phi_u(\omega)$	spectrum of the signal $u(t)$.
$\Phi_{yu}(\omega)$	cross-spectrum between the signals $y(t)$ and $u(t)$.
$\varphi(t)$	regression vector at time t.
$\Psi(t, \theta)$	gradient of $y(t\|\theta)$ with respect to θ.
$\mathbb{R}$, $\mathbb{C}$	sets of real and complex numbers.
$\mathrm{AsN}(\mu, P)$	asymptotic normal distribution with mean μ and covariance P.
$\mathrm{Prob}(x \leq C)$	probability that the random variable x is less than C.
I_θ	confidence interval for θ.

Operators

q^{-1}	delay operator, $q^{-1}u(t) = u(t-1)$.
$\mathrm{E}\, x$	expectation of the random variable x.
$\bar{\mathrm{E}}\, f(t)$	$\lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \mathrm{E}\, f(t)$
$\mathrm{Var}(x)$, $\mathrm{Cov}(x)$	variance/ covariance matrix of the random vector x.
$\dim \theta$	dimension of the column vector θ.

Abbreviations and Acronyms

ARX	AutoRegressive with eXternal input.
ARMAX	AutoRegressive Moving Average with eXternal input.
BJ	Box-Jenkins.
ETFE	Empirical Transfer Function Estimate.
i.i.d.	Independent and Identically Distributed.
LTI	Linear Time Invariant.
MEM	Model Error Model.
MOD	Model On Demand.
OE	Output Error.
PDF	Probability Density Function.

1

Introduction

This chapter gives an outline of the thesis to introduce the different topics treated in the chapters. It also contains a presention of the main contributions of this work.

1.1 Thesis Outline

This thesis is divided into six chapters, including this first one.

Chapter 2 contains background material in the areas of system identification and model validation. The only part that is non-standard is the part describing local polynomial modeling. This background material is needed in Chapter 4, where we use this modeling tool to obtain unprejudiced model error models.

In Chapter 3 we review some basic facts about confidence regions. The important part is the understanding of how the construction of high dimensional confidence regions relates to the construction of one-dimensional confidence intervals.

The concept of model error modeling as a tool for model validation is discussed to some extent in Chapter 4. We explain why we think that linear model error models should be presented in the frequency domain and discuss how non linear model error models can be used. The chapter ends with a large simulation study of the performance of different linear model error models.

Bootstrap is a relatively new area in statistics. Since its start in Efron (1979) the number of applications has steadily increased. In Chapter 5 we show how bootstrap can be used in system identification. Its wide use is shown by the application

to variance estimation of undermodeled models. The chapter contains several simulation studies to show the performance of the methods.

The last chapter discusses how the variance of identified models influences the reduced models. We specialize to L_2 model reduction, but from the discussion it should be clear that the same approach also works for other reduction techniques. The most interesting result is that when estimating a model in two steps, first estimation of a high order model and then reduction to low order, we will end up with the same variance as the direct identification of the low order would give. From this it also follows that L_2 model reduction gives maximal variance reduction of the high order estimate.

1.2 Contributions

The main contributions of this thesis are the following:

- Using local polynomial modeling and non-parametric frequency domain identification in model error modeling (Section 4.4).
- The idea of using bootstrap to estimate the variance error in models with unmodeled dynamics (Section 5.2.2).
- Describing how bootstrap can be used to obtain confidence intervals with simultaneous confidence degree (Section 5.3.3).
- The result that L_2 model reduction to undermodeled FIR-models produces models with lower variances than direct estimation (Section 6.4).
- Showing that L_2 reduced models meets the Cramér-Rao bound in the case of no undermodeling. This also shows that it is optimal in minimizing the variance of the high order estimate (Section 6.5).

Parts of the results in this thesis have been published earlier:

Tjärnström, F. and Ljung, L. (1999). Estimating the variance in case of undermodeling using bootstrap. In *Proceedings of the 38th IEEE Conference on Decision and Control*, Pheonix, Arizona, USA.

Tjärnström, F. and Forssell, U. (1999). Comparison of methods for probabilistic uncertainty bounding. In *Proceedings of the 38th IEEE Conference on Decision and Control*, Pheonix, Arizona, USA.

Tjärnström, F. and Ljung, L. (1999). Minimizing the variance of transfer function estimates. In Yakubovich, V. and Fradkov, A., editors, *6th Saint Petersburg Symposium on Adaptive Systems Theory*, volume 1, pages 194–200, S:t Petersburg, Russia.

2

System Identification and Model Validation

System identification deals with the problem of estimating models of dynamical systems from input-output data and prior information. There are many approaches to how the identification phase should be performed. Traditionally, the two major ways have been black-box parametric identification and non-parametric frequency domain identification. From this other approaches have emerged, like subspace identification and local polynomial identification. In this chapter the basis for such techniques will be discussed (except subspace identification). The chapter will also include a discussion on model validation. Model validation deals with the problem of telling whether or not a given model could have generated data from a given system, which makes it an extremely important subject. The material presented in this chapter is standard and could be skipped by readers familiar with the topics.

2.1 Parametric Identification

Dynamical systems in general is a quite wide class of systems, so in order fulfill the goal of obtaining good estimates we need to specialize to certain model structures (usually linear time-invariant ones). These models are described by a vector of parameters, which is adjusted so that the model mimics the systems behavior as much as possible. The quality of these models is naturally determined by their ability to describe the underlying system. It is therefore important to extract information from the parameter estimates to obtain an uncertainty description of the estimated

model. These are the topics that will be discussed in this section. We present the basic concepts, and some results that are needed in this thesis. For more thorough treatments see Ljung (1999b), or Söderström and Stoica (1989).

2.1.1 Prediction Error Methods

We will throughout the thesis denote the input signal by $u(t)$ and the output signal by $y(t)$. The input-output data collected up to time t will be denoted Z^t, i.e.,

$$Z^t = \{u(1), y(1), \dots, u(t), y(t)\}. \tag{2.1}$$

Usually N will be the total number of measurements. We assume that Z^N is generated according to

$$y(t) = G_0(q)u(t) + v(t), \tag{2.2}$$

where $G_0(q)$ is a linear time-invariant system, usually referred to as the "true system". q is the discrete-time shift operator, i.e., $qu(t) = u(t+1)$ and $v(t)$ is some noise acting upon the system. It will be assumed that

$$v(t) = H_0(q)e(t), \tag{2.3}$$

where $H_0(q)$ is some inversely stable monic filter, and $\operatorname{Var} e(t) = \lambda$.

The models we fit to data will be parameterized by a d-dimensional real-valued parameter vector θ, i.e.,

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t). \tag{2.4}$$

The choice of parameterization is taken either as a rational transfer function or as a state space description, where the latter may be desirable in the MIMO (Multiple Input Multiple Output) case. Different choices of transfer functions will be discussed in Section 2.1.4.

A differentiable mapping of the space of parameters, $\theta \in \mathbb{R}^d$, to the space of parameterized models like in (2.4) is called a *model structure* $\mathcal{M}$. The model we get for a particular choice of parameters, θ, will be denoted $\mathcal{M}(\theta)$. This leads us to define the notion of equality of models. We say that two models $\mathcal{M}(\theta_1)$ and $\mathcal{M}(\theta_2)$ are *equal* if and only if

$$\begin{cases} G(e^{i\omega},\theta_1) = G(e^{i\omega},\theta_2) \\ H(e^{i\omega},\theta_1) = H(e^{i\omega},\theta_2) \end{cases}, \text{ for almost all } \omega. \tag{2.5}$$

The one-step-ahead predictor of $y(t)$ given measurements up to time $t-1$ is given by

$$\hat{y}(t|\theta) = H^{-1}(q,\theta)G(q,\theta)u(t) + \left(1 - H^{-1}(q,\theta)\right)y(t). \tag{2.6}$$

Note that the predictor only depends on $y(s),\ s < t$, since H is monic. This results in a prediction error

$$\begin{aligned}\varepsilon(t,\theta) =& y(t) - \hat{y}(t|\theta) \\ =& H^{-1}(q,\theta)\left(y(t) - G(q,\theta)u(t)\right).\end{aligned} \tag{2.7}$$

To ensure stability of the predictions we see that we need restrict θ in such a way that the transfer functions from u and y to $\hat{y}$ are stable. Therefore we define

$$D_{\mathcal{M}} = \{\theta \in \mathbb{R}^d \,|\, H^{-1}(q,\theta)G(q,\theta) \text{ and } H^{-1}(q,\theta) \text{ are stable}\} \tag{2.8}$$

This leads us to define a *model set* as

$$\mathcal{M}^* = \{\mathcal{M}(\theta) \mid \theta \in D_{\mathcal{M}}\} \tag{2.9}$$

In order to judge the models performance on the data set, Z^N, we define the loss function

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} l\left(\varepsilon(t,\theta)\right). \tag{2.10}$$

Here we use a general norm $l(\cdot)$. The most commonly used one is the quadratic

$$l(\varepsilon) = \frac{1}{2}\varepsilon^2, \tag{2.11}$$

which will be used throughout the rest of this thesis. Other norms might be useful, e.g., with respect to robustness against outliers (Ljung, 1999b).

It is natural to choose the estimate of θ as the minimizer of (2.10). This model is the best predictor of future outputs (with respect to the dataset used)

$$\begin{aligned}\hat{\theta}_N &= \arg\min_{\theta\in D_{\mathcal{M}}} V_N(\theta) \\ &= \arg\min_{\theta\in D_{\mathcal{M}}} \frac{1}{2N}\sum_{t=1}^{N}\varepsilon^2(t,\theta),\end{aligned} \tag{2.12}$$

i.e., we use prediction error methods (PEM). We denote the estimated transfer functions by $\hat{G}_N(q) = G(q,\hat{\theta}_N)$, $\hat{H}_N(q) = H(q,\hat{\theta}_N)$.

2.1.2 Convergence Properties

To describe the limit properties of the estimates (as $N \to \infty$) we need to establish some additional notation. Let $\bar{\mathrm{E}}$ denote

$$\bar{\mathrm{E}}\, f(t) = \lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N}\mathrm{E}\, f(t), \tag{2.13}$$

i.e., averaging over both the stochastic and deterministic parts of its arguments. Moreover, we let the limiting loss function be defined by

$$\bar{V}(\theta) = \bar{\mathrm{E}}\frac{1}{2}\varepsilon^2(t,\theta). \tag{2.14}$$

The basic result is then (Ljung, 1999b, Chapter 8) that under weak conditions

$$\hat{\theta}_N \to \theta^* = \arg\min_{\theta \in D_{\mathcal{M}}} \bar{V}(\theta), \text{ as } N \to \infty. \tag{2.15}$$

That is, $\hat{\theta}_N$ converges to the best model provided by the model class. (If the minimizer of $\bar{V}(\theta)$ is not unique , the convergence will be to some value in the set of minimizers.)

If the "true system" belongs to the model class, i.e., there exists some θ_0 such that $G(e^{i\omega}, \theta_0) = G_0(e^{i\omega})$ and $H(e^{i\omega}, \theta_0) = H_0(e^{i\omega})$ for almost all ω, we get that the limiting estimate actually corresponds to the true system. The estimate is then said to be *unbiased*. For convenience we will denote "the true system"=$\mathcal{S}$.

2.1.3 Statistical Properties of the Estimates

To be able to *validate* the estimated model we need to have expressions for the distributions of the estimates. Most expressions are based on the central limit theorem, in one form or another. We once again refer to Ljung (1999b) for the details.

We start with some notations. Let $f'(\theta_1)$ denote the $1 \times d$ dimensional matrix being the derivative of $f(\theta)$ evaluated at $\theta = \theta_1$. As in Section 2.1.2 we let θ^* be the limiting estimate. We also need to introduce the concept of *identifiability*. We say that a model structure is *globally identifiable* at θ^* if

$$\mathcal{M}(\theta) = \mathcal{M}(\theta^*) \Rightarrow \theta = \theta^*. \tag{2.16}$$

Assuming global identifiability and some other weak conditions it now holds that

$$\sqrt{N}(\hat{\theta}_N - \theta^*) \in As N(0, P_\theta) \tag{2.17}$$

$$P_\theta = \left[\bar{V}''(\theta)\right]^{-1} Q \left[\bar{V}''(\theta)\right]^{-1} \tag{2.18}$$

$$Q = \lim_{N\to\infty} N\,\mathrm{E}\left\{[V_N'(\theta^*)][V_N'(\theta^*)]^T\right\}. \tag{2.19}$$

If $\mathcal{S} \in \mathcal{M}$, the expression for the covariance matrix simplifies considerably

$$P_\theta = \lambda_0 \left[\bar{\mathrm{E}}\Psi(t,\theta_0)\Psi^T(t,\theta_0)\right]^{-1} \tag{2.20}$$

$$\Psi(t,\theta_0) = -\frac{d}{d\theta}\varepsilon(t,\theta)\Big|_{\theta=\theta_0}. \tag{2.21}$$

From this we also get that covariance matrix can easily be estimated from Z^N

$$\hat{P}_\theta = \hat{\lambda}_N \left[\frac{1}{N} \sum_{t=1}^{N} \Psi(t, \hat{\theta}_N) \Psi^T(t, \hat{\theta}_N) \right]^{-1} \tag{2.22}$$

$$\hat{\lambda}_N = \frac{1}{N} \sum_{t=1}^{N} \varepsilon^2(t, \hat{\theta}_N). \tag{2.23}$$

If $\mathcal{S} \notin \mathcal{M}$, i.e., we have *unmodeled* dynamics, it becomes much more difficult to calculate Q in (2.19). Solutions to this and related problems are given in Hjalmarsson (1993), Larssen (1992), and Pötscher and Prucha (1997). The difficulty lies in that the natural estimate of Q,

$$\hat{Q} = [V_N'(\hat{\theta}_N)][V_N'(\hat{\theta}_N)]^T, \tag{2.24}$$

will be zero by the definition of $\hat{\theta}_N$ as the minimizer of the loss-function in (2.12) (Hjalmarsson and Ljung, 1992; Ljung, 1999b).

We also mention that if the model order, n, tends to infinity we have the following result

$$\frac{1}{N} P(\omega) \approx \frac{n}{N} \frac{1}{2} \Phi_v(\omega) \begin{bmatrix} 1/\Phi_u(\omega) & 0 \\ 0 & 1/\Phi_u(\omega) \end{bmatrix} \text{ as } N, n, \frac{N}{n} \to \infty, \tag{2.25}$$

where $P(\omega)$ is the covariance matrix for $\left[\operatorname{Re} \hat{G}_N(e^{i\omega}) \;\; \operatorname{Im} \hat{G}_N(e^{i\omega}) \right]^T$ (Ljung, 1985b; Forssell, 1998). That is, confidence ellipsoids in the Nyquist plot tend to circles with a radius determined by the signal-to-noise ratio.

2.1.4 Choice of Parameterization

In this section we will present the most common parameterizations used for rational transfer functions. Different types of state space parameterizations can be found in, e.g., McKelvey (1995), and Maciejowski and Ober (1988), but they will not be discussed here.

The most general structure can be written

$$A(q)y(t) = \frac{B(q)}{F(q)} u(t) + \frac{C(q)}{D(q)} e(t), \tag{2.26}$$

where A, C, D, and F are monic polynomials in q^{-1} and

$$B(q) = b_1 q^{-n_k} + \ldots + b_{n_b} q^{-n_b - n_k + 1}. \tag{2.27}$$

Structure	Polynomials
FIR	B
ARX	$A,\ B$
ARMAX	$A,\ B,\ C$
OE	$B,\ F$
BJ	$B,\ C,\ D,\ F$

Table 2.1 *Common model structures, the polynomials not included are all equal to one.*

From this general structure we get the substructures in Table 2.1.

When talking about, say, an OE-model with four f-parameters and three b-parameters and two delays, we will denote it an OE(3,4,2)-model. The digits are presented in "alphabetic order" corresponding to the polynomials. (This is in the same spirit as in Ljung (1997b).)

The maybe most important one of the model structures in Table 2.1 is the ARX (AutoRegressive with eXternal input) structure. This is mainly due to the two following reasons.

1. It is fast and easy to estimate and there exist no local minima.

2. It is capable of approximating any linear system arbitrarily well, provided that the model order is high enough. (Ljung, 1985a).

We give these results below.

As seen from above, the ARX structure has the following parameterization:

$$G(q,\theta) = \frac{B(q)}{A(q)} = \frac{b_1 q^{-n_k} + \cdots + b_{n_b} q^{-n_k-n_b+1}}{1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}}, \tag{2.28}$$

$$H(q,\theta) = \frac{1}{A(q)} = \frac{1}{1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}}. \tag{2.29}$$

The simplicity of this structure is seen from that we can write the one step ahead predictor as

$$\hat{y}(t|\theta) = \varphi^T(t)\theta \tag{2.30}$$

$$\varphi(t) = \left[-y(t-1)\cdots -y(t-n_a)\ \ u(t-n_k)\cdots u(t-n_k-n_b+1)\right]^T \tag{2.31}$$

$$\theta = \left[a_1 \cdots a_{n_a}\ \ b_1 \cdots b_{n_b}\right]^T. \tag{2.32}$$

Now the minimization in (2.12) will be a standard least-squares problem, with the solution

$$\hat{\theta}_N = \left[\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\varphi(t)y(t). \tag{2.33}$$

The implementation is, however, usually done using QR-factorizations (Ljung, 1999b, 1997b). This give rise to fast and numerically stable solutions.

If we take $n_a = n_b = n$ and $n_k = 1$, we have the other appealing property (see (Ljung, 1999b)):

$$\hat{G}_N(e^{i\omega}) = \frac{\hat{B}_N(e^{i\omega})}{\hat{A}_N(e^{i\omega})} \to G_0(e^{i\omega}) \quad \text{uniformly in } \omega \text{ as } N \gg n \to \infty, \tag{2.34}$$

$$\hat{H}_N(e^{i\omega}) = \frac{1}{\hat{A}_N(e^{i\omega})} \to H_0(e^{i\omega}) \quad \text{uniformly in } \omega \text{ as } N \gg n \to \infty. \tag{2.35}$$

This means that ARX models can approximate any linear system arbitrarily well, just use a high enough model order. This property could be useful for model validation purposes. More on this subject can be found in Section 4.2. It can also be useful as a modeling tool itself; estimate a high-order ARX model and reduce it to an appropriate structure using some model reduction technique, e.g., balanced reduction (see Chapter 6).

2.2 Non-Parametric Frequency Domain Identification

In contrast to the parametric approach just given, we will here present a non-parametric alternative. Non-parametric means that we do not try to estimate any parametric models that describe the data. Instead we estimate the frequency response at a set of frequencies. This estimate is a very raw and crude estimate and need to be smoothed in order to be useful. This smoothing operation is achieved by weighting nearby estimates together to reduce the variance in the estimates. Here we actually need to specify one parameter that describes the weighting function, but the method is anyhow said to be non-parametric.

2.2.1 The Empirical Transfer Function Estimate

The underlying assumptions about the data generation is the same as in Section 2.1, i.e.,

$$y(t) = G_0(q)u(t) + v(t).$$

We start by introducing the Fourier transforms of the input and the output

$$U_N(\omega) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} u(t) e^{-i\omega t}, \tag{2.36}$$

$$Y_N(\omega) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} y(t) e^{-i\omega t}. \tag{2.37}$$

From these we define the empirical transfer function estimate (ETFE) in the following way

$$\hat{\hat{G}}_N(e^{i\omega}) = \frac{Y_N(\omega)}{U_N(\omega)}. \tag{2.38}$$

That is, we estimate the frequency response by the ratio of the output and input Fourier transforms. This seems reasonable, since it gives a correct estimate in a noise free environment (provided that there is some input energy at that frequency).

This estimate is known to be a very crude estimate of the true system, G_0. One can show that the ETFE has the following asymptotic properties (Ljung, 1999b, Lemma 6.1)

$$\mathrm{E}\, \hat{\hat{G}}_N(e^{i\omega}) = G_0(e^{i\omega}) + \rho^{(1)}(N), \tag{2.39}$$

$$\mathrm{Var}\, \hat{\hat{G}}_N(e^{i\omega}) = \frac{1}{|U_N(\omega)|^2} \left[\Phi_v(\omega) + \rho^{(2)}(N) \right], \tag{2.40}$$

where $\Phi_v(\omega)$ is the noise spectrum, and

$$\rho^{(i)}(N) \to 0 \text{ as } N \to \infty, \ i = 1, 2. \tag{2.41}$$

That is, the ETFE is an asymptotically unbiased estimate of the true system, but the variance of the estimate tends to the noise to signal ratio. To decrease the, usually big, variance the estimate has to be smoothed. Note that the results (2.39-2.40) are derived under first order approximations, see Guillaume et al. (1996) for details.

2.2.2 Smoothing the ETFE

The reason for the bad variance properties of this estimate is that when forming the ETFE we do not take into account that there might be a relation between different frequencies. To take advantage of this information, it is natural to weigh estimates at different frequencies together to *smooth* the estimate.

$$\hat{G}_N(e^{i\omega}) = \frac{\sum_k w_k(\omega) \hat{\hat{G}}_N(e^{i\omega_k})}{\sum_k w_k(\omega)}. \tag{2.42}$$

This smoothing operation reduces the variance of the estimate, but introduces a bias. We shall give the results for a standard choice of weights, namely when the weights are chosen as the inverse of the variances of the observations. Let the smoothed estimate, $\hat{G}_N(e^{i\omega})$, be defined by

$$\hat{G}_N(e^{i\omega}) = \frac{\int_{-\pi}^{\pi} W_\gamma(\omega-\xi)|U_N(\xi)|^2 \hat{\hat{G}}_N(e^{i\xi})d\xi}{\int_{-\pi}^{\pi} W_\gamma(\omega-\xi)|U_N(\xi)|^2 d\xi}. \tag{2.43}$$

Here $W_\gamma(\omega)$ is a frequency window whose (inverse) width is defined by γ. One usually characterizes the window by the following numbers

$$\int_{-\pi}^{\pi} W_\gamma(\xi)d\xi = 1, \quad \int_{-\pi}^{\pi} \xi W_\gamma(\xi)d\xi = 0, \quad \int_{-\pi}^{\pi} \xi^2 W_\gamma(\xi)d\xi = M(\gamma)$$
$$\int_{-\pi}^{\pi} |\xi|^3 W_\gamma(\xi)d\xi = C_3(\gamma), \quad \int_{-\pi}^{\pi} W_\gamma^2(\xi)d\xi = \frac{1}{2\pi}\bar{W}(\gamma) \tag{2.44}$$

Using these one can show that the following asymptotic expressions (Ljung, 1999b, Chapter 6) hold

$$\mathrm{E}\,\hat{G}_N(e^{i\omega}) - G_0(e^{i\omega}) = M(\gamma)\left[\frac{1}{2}G_0''(e^{i\omega}) + G_0'(e^{i\omega})\frac{\Phi_u'(\omega)}{\Phi_u(\omega)}\right] + O(C_3(\gamma)) + O(1/\sqrt{N}), \tag{2.45}$$

$$\mathrm{E}\left|\hat{G}_N(e^{i\omega}) - \mathrm{E}\,\hat{G}_N(e^{i\omega})\right|^2 = \frac{1}{N}\bar{W}(\gamma)\frac{\Phi_v(\omega)}{\Phi_u(\omega)} + o(\bar{W}(\gamma)/N). \tag{2.46}$$

where prime denotes differentiation with respect to ω. It can also be shown that the real and imaginary parts of $\hat{G}_N(e^{i\omega})$ are asymptotically uncorrelated and have variances equal to half of that of (2.46). Furthermore, the estimates are asymptotically uncorrelated at different frequencies.

Using these expressions it is possible to compute an optimal choice (in mean square error sense) of the *bandwidth* of the window, $\gamma(\omega)$. The optimal choice is, as can be realized from the expressions above, dependent upon unknown quantities like the derivative of $G(e^{i\omega})$, which makes it difficult to compute. We also note that the optimal choice of γ will depend on ω, but for practical reasons one often choose a fixed bandwidth. The final choice is then done by visual inspection of the estimate. A typical approach is to start with $\gamma = N/20$ and increase γ until enough details can be seen. This will make it difficult to obtain good fits throughout the frequency axis. This is exemplified in Stenman et al. (1999).

One common choice of window function is the Hamming window. This is also the window used in the used in the routine `spa` in the System Identification Toolbox

in MATLAB. It is described in the time domain by

$$w_\gamma(\tau) = \frac{1}{2}\left(1 + \cos\frac{\pi\tau}{\gamma}\right),\ 0 \le |\tau| \le \gamma. \tag{2.47}$$

The most important (approximate) expressions for this window are

$$M(\gamma) \approx \frac{\pi^2}{2\gamma^2}, \quad \bar{W}(\gamma) \approx 0.75\gamma. \tag{2.48}$$

2.2.3 Spectral Analysis

Closely related to the ETFE is the concept of spectral analysis. Here the unsmoothed estimate coincides with the ETFE. The difference lies in the smoothing procedure. In spectral analysis the smoothing operation is performed in the time domain using the covariance and cross-covariance functions ($\hat{R}_u(\tau)$ and $\hat{R}_{yu}(\tau)$). These are finally transformed to frequency domain using the discrete Fourier transform, giving $\hat{\Phi}_u(\omega)$ and $\hat{\Phi}_{yu}(\omega)$. The spectral estimate is then obtained through

$$\hat{G}_N(e^{i\omega}) = \frac{\hat{\Phi}_{yu}(\omega)}{\hat{\Phi}_u(\omega)}. \tag{2.49}$$

See Ljung (1999b) for details.

2.3 Local Polynomial Modeling

In this section we will discuss how local polynomial models can be used to smooth noisy measurements. The strength of this approach is the fact that it is local. This makes it possible to, e.g., estimate nonlinear systems using local linear models. See (Stenman, 1999) for several applications in this area. We will start to introduce the subject with a review some important aspects of the theory. More complete treatments can be found in Wand and Jones (1995), and Fan and Gijbels (1996). Applications to frequency domain identification is discussed in Stenman et al. (1999).

2.3.1 Local Polynomial Regression

To solve the smoothing problem, we assume that the noisy data is generated according to

$$Y_i = m(X_i) + e_i,\ i = 1, \ldots, N \tag{2.50}$$

where the $\{e_i\}_{i=1}^N$ is an i.i.d. sequence with zero mean and variances σ_i^2 and $m(\cdot)$ is some unknown function. Since the measurements are noisy we would like to perform some kind of averaging operation on the measurements in order to reduce their variance. The idea is to take advantage of the possible dependence between *response variables*, Y_i, originating from closely lying X_is. This leads to an estimate of the *regression function*, m, at a point x:

$$\hat{m}(x) = \sum_{i=1}^{N} W_i(x) Y_i. \tag{2.51}$$

Here the weights $W_i(x)$ are chosen as to put more weight on the Y_is that origin from X_is closest to x. There are several ways to chose these weights, like nearest neighbor Fan and Gijbels (1996) and direct estimation methods Stenman (1999). We will discuss how they can be calculated using kernel methods.

The Local Polynomial Model

One fundamental difference from global regression estimation is that we are not interested in finding an explicit form for the regression function, m. The aim is instead to approximate $m(\cdot)$ by fitting a polynomial locally around a point x. Since the model is only adapted around a small area of x we will expect a good fit in this area. (Think of Taylor expansions around that point.) As a matter a fact we seldom explicitly use the estimated model, we mostly just use the estimate at x.

Thus, we model the regression function, $m(\cdot)$, by a pth degree polynomial

$$m(X_i) = \beta_0 + \beta_1(X_i - x) + \ldots + \beta_p(X_i - x)^p \tag{2.52}$$

locally around x. Furthermore, let the estimate of $\boldsymbol{\beta} = \begin{pmatrix}\beta_0 & \ldots & \beta_p\end{pmatrix}^T$ be given by the minimization of

$$V_x(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left(Y_i - \sum_{j=0}^{p} \beta_j (X_i - x)^j \right)^2 \frac{K_h(X_i - x)}{\sigma_i^2}, \tag{2.53}$$

where $K_h(\cdot) = \frac{1}{h} K(\frac{\cdot}{h})$, and $K(\cdot)$ is a *kernel function* and h is its *bandwidth*. The extra weighting provided by the inverse of the variances in the loss function is optional. When this is extra information is used (and known) the criterion is similar to maximum-likelihood estimation, otherwise it is a weighted prediction error criteria. Either way, from the minimization above we get the estimate of $m(x)$ (using bandwidth h)

$$\hat{m}_h(x) = \hat{\beta}_0. \tag{2.54}$$

Obtaining Locality

The locality of the estimate is obtained using weighting given by the kernel. The use of kernel functions is also closely related to probability density function estimation. Therefore, $K(\cdot)$ is traditionally normalized according to

$$\int K(u)du = 1, \quad \int uK(u)du = 0. \tag{2.55}$$

Two common kernels are the *tricube* kernel

$$K(u) = \frac{70}{81}(1 - |u|^3)_+^3 \tag{2.56}$$

and the *Epanechnikov* kernel

$$K(u) = \frac{3}{4}(1 - u^2)_+. \tag{2.57}$$

Here $(\cdot)_+$ denotes the positive part. The latter kernel is known to have some asymptotic mean square error optimality, but it suffers from the discontinuities at ± 1. On the other hand, the tricube kernel has continuous derivatives in the closed interval $[-1, 1]$. This has the positive effect of reducing *leakage* effects. This is one of the reasons it is the default choice in standard softwares like LOWESS Cleveland (1979), LOESS Cleveland and Devlin (1988), and LOCFIT Loader (1997).

2.3.2 The Choice of Bandwidth

The most problematic and computer intensive part in local polynomial regression is the choice of the local bandwidth, h. The computation time is significantly reduced if a global bandwidth is chosen, but unfortunately, quite a lot of flexibility is lost when using fixed bandwidths (compare Section 4.3). We will show how one can achieve local bandwidths by minimizing some well designed criterion. There are several such criteria like cross-validation, generalized cross-validation, Akaike's information criterion (AIC), and final prediction error (FPE). The criterion that we will use is a local generalized version of Mallows C_p statistic Cleveland and Loader (1994)

$$\begin{aligned} C_x(h) = &\sum_{i=1}^{N} \frac{K_h(X_i - x)}{\sigma_i^2 \operatorname{tr}(\mathbf{W})} \left(Y_k - \hat{m}(X_i)\right)^2 \\ &- 1 + \alpha \frac{\operatorname{tr}((\mathbf{X}^T\mathbf{W}\mathbf{V}\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{W}^2\mathbf{V}\mathbf{X}))}{\operatorname{tr}(\mathbf{W})}. \end{aligned} \tag{2.58}$$

Here $\mathbf{V} = \text{diag}(1/\sigma_1^2, \ldots, 1/\sigma_N^2)$, $\mathbf{X}$ is a Vandermonde matrix of size $N \times (p+1)$

$$\mathbf{X} = \begin{pmatrix} 1 & (X_1 - x) & \ldots & (X_1 - x)^p \\ \vdots & \vdots & & \vdots \\ 1 & (X_N - x) & \ldots & (X_N - x)^p \end{pmatrix}, \tag{2.59}$$

and $\mathbf{W} = \text{diag}(K_h(X_1 - x), \ldots, K_h(X_N - x))$. The parameter $\alpha \geq 2$ is an extra tuning parameter. If $\alpha = 2$ we have the standard local C_p criterion. On the other hand, by increasing α we also increase the penalty on the variance (given by the last term in (2.58)).

The local optimal bandwidth at x is given by

$$h_{opt} = \arg\min_h C_x(h). \tag{2.60}$$

Note that $C_x(h)$ does not only depend on h explicitly through $K_h(\cdot)$, it also depends on h implicitly through the estimate $\hat{m}(X_i)$.

To find the optimal choice of h at a certain point x some sort of numerical search has to be performed. The following approach, which is also used in the LOCFIT implementation, is suggested by Loader (1997).

ALGORITHM 2.1 Bandwidth selection

1. Start by fitting a local model, m, and compute the goodness of fit (2.58) using a very small bandwidth h_0.
2. Increase the bandwidth exponentially using

   $$h_{i+1} = (1 + \frac{0.3}{d}) \cdot h_i,$$

   where $d = \dim X_j$. Compute $\hat{m}$ and $C_x(h_{i+1})$. Repeat until the $C_x(h)$ has started to increase "significantly".
3. If necessary perform a finer search around the minimizer found in the previous step.

❑

2.3.3 Computing the Estimate

The choice of the (weighted) 2-norm in the minimization of Equation (2.53) makes the estimate $\hat{\boldsymbol{\beta}}$ easy to compute. In fact, the solution can be expressed explictly since it becomes a standard least-squares problem. The explicit solution makes

analysis much more tractable. Results on this can be found in the references in the beginning of the section.

Using matrix representation we can rewrite (2.53) in a more convenient form. Defining

$$\mathbf{y} = \begin{pmatrix} Y_1 & \dots & Y_N \end{pmatrix}^T, \tag{2.61}$$

the loss function (2.53) can be written

$$V_x(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T \mathbf{W}\mathbf{V}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}), \tag{2.62}$$

with solution $\hat{\boldsymbol{\beta}}$ given by

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T \mathbf{W}\mathbf{V}\mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{W}\mathbf{V}\mathbf{y}. \tag{2.63}$$

From this we get that

$$\hat{m}_h(x) = \hat{\beta}_0 = \mathbf{e}_1^T \left(\mathbf{X}^T \mathbf{W}\mathbf{V}\mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{W}\mathbf{V}\mathbf{y}, \tag{2.64}$$

where $\mathbf{e}_1$ is the first column of the $p \times p$ unit matrix. We see that the estimate is actually a weighted sum of the measurements, which is in correspondence with (2.51). The weights are the elements of the row vector

$$\mathbf{w}^T(x) = \mathbf{e}_1^T \left(\mathbf{X}^T \mathbf{W}\mathbf{V}\mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{W}\mathbf{V} \tag{2.65}$$

The whole estimation procedure now ends up in the following:

1. At the point of interest, x, choose the degree, p, of the polynomial to fit, the shape of the kernel, $K(u)$, and the goodness of fit criterion, $C_x(h)$.

2. Estimate the model using the bandwidth obtained from Algorithm 2.1.

2.3.4 Pointwise Confidence Intervals

In many applications of local polynomial regression, there is a great interest in a description of the uncertainty in the estimate, not only the point estimate itself. Uncertainty measures for this estimation technique is in fact quite easy to calculate, since the estimate is linear in the observations (2.64). We have (provided $\hat{m}_h(x)$ is

an unbiased estimate of $m(x)$)

$$\begin{aligned}\operatorname{Var}\hat{m}_h(x) =& \mathrm{E}(\hat{m}_h(x) - \mathrm{E}\,\hat{m}_h(x))^2 \\ =& \mathrm{E}\left(\sum_{i=1}^{N} W_i(x)e_i\right)^2 = \sum_{i=1}^{N} W_i^2(x)\sigma_i^2 \\ \approx& \sigma^2(x)\cdot\sum_{i=1}^{N} W_i^2(x) = \sigma^2(x)\cdot\mathbf{w}^T(x)\mathbf{w}(x) \\ =& \sigma^2(x)\mathbf{e}_1^T\left(\mathbf{X}^T\mathbf{W}\mathbf{V}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{W}^2\mathbf{V}^2\mathbf{X}\left(\mathbf{X}^T\mathbf{W}\mathbf{V}\mathbf{X}\right)^{-1}\mathbf{e}_1. \end{aligned} \tag{2.66}$$

The approximation involved is justified by the heavier weighting around the estimation point.

Moreover, if the model errors e_i are Gaussian distributed we get approximate pointwise confidence intervals for $m(x)$

$$I_{m(x)} = \left(\hat{m} - \Phi(1-\alpha/2)\cdot\sigma\cdot||\mathbf{w}||, \hat{m} + \Phi(1-\alpha/2)\cdot\sigma\cdot||\mathbf{w}||\right). \tag{2.67}$$

2.3.5 Noise Variance Estimation

When computing confidence intervals from the estimate, $\hat{m}_h(x)$, we will need a estimate of the noise variance, $\sigma^2(x)$. This is obtained as the normalized prediction errors

$$\hat{\sigma}^2(x) = \frac{\sum_i w_i(x)\left(Y_i - \hat{m}(X_i)\right)^2}{\operatorname{tr}(\mathbf{W}) - \operatorname{tr}\left((\mathbf{X}^T\mathbf{W}\mathbf{V}\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{W}^2\mathbf{V}\mathbf{X})\right)}. \tag{2.68}$$

2.4 Model Validation

Model validation deals with the problem of trying to tell if the model could have produced the data measured from the system. If we cannot come up with convincing evidence of the opposite, we accept the model, or rather, we do *not falsify* it. Therefore a more correct name would be to talk about model invalidation, i.e., we never accept a model as being correct, we only reject the obviously incorrect ones. The aim in model validation research is therefore to come up with stronger model validation tests that falsify more and more models.

Typically the model validation question is split up into subquestions of different degree of generality:

- Is the underlying "true system" described by the model?
- Does the model describe the system accurately enough for my purposes?

- If my model is invalidated, what types of errors does it have?

To answer these questions one has to compute some sort of measure that describes important aspects of the model/system. Depending on the noise assumptions this leads to different descriptions of the errors that are present in the model. We will try to point out that for some measures we end up with the same types of tests, irrespective of noise assumptions.

2.4.1 Residual Analysis vs Statistical Model Validation

Most classical model validation tests are based on *residual analysis*. The residuals are computed using some model $\hat{G}$ of the system

$$\varepsilon(t) = L(q)(y(t) - \hat{G}(q)u(t)), \tag{2.69}$$

where $L(q)$ is a possible pre-filter which, e.g., could be taken as the inverse noise model. The tests performed on the residuals are then based on one or several of the following measures of the residuals:

- The maximal absolute value of the residuals

$$\max_{1 \le t \le N} |\varepsilon(t)|$$

- The mean of the residuals

$$\bar{\varepsilon} = \frac{1}{N} \sum_{t=1}^{N} \varepsilon(t)$$

- The variance of the residuals

$$\frac{1}{N} \sum_{t=1}^{N} (\varepsilon(t) - \bar{\varepsilon})^2$$

- The mean square error of the residuals

$$\frac{1}{N} \sum_{t=1}^{N} \varepsilon^2(t)$$

- Whiteness of the residuals

$$\frac{1}{N} \sum_{t=1}^{N} [\varepsilon(t-1) \dots \varepsilon(t-M)]^T \, \varepsilon(t)$$

- The number of sign changes of the residuals (based on the assumption that the residuals are white and come from a symmetric distribution)
- Correlation between residuals and inputs

$$\frac{1}{N}\sum_{t=1}^{N}[u(t-M_1)\dots u(t-M_2)]^T\,\varepsilon(t)$$

These measures are thresholded to check whether they are "too large" or not. It is important to clarify that most of these measures are meaningful from both probabilistic and deterministic viewpoints. The difference is not in the *statistics*, it is in *how* the thresholds are determined. If we incorporate a probabilistic view to residual analysis we refer to this as *statistical model validation*. In deterministic residual analysis the threshold is determined on the basis of prior knowledge, assumptions, or some ad hoc procedure. In statistical model validation, on the other hand, one computes the threshold based on some probabilistic criterion. The close connection between residual analysis and statistical model validation is thoroughly discussed in Ljung and Hjalmarsson (1995).

This fact can be illustrated by the following discussion. Some commonly used types of model validation tests are based on worst case assumptions on the noise like: "max $|\varepsilon(t)|$ cannot be larger than δ if the model is to be validated". This is often the assumption in unknown-but-bounded or set membership characterizations. Such a test is obviously in strong connection with the measure under the first point in this list above. This test could also be done in a probabilistic manner. Then it would be rephrased like: "The residuals will all have a magnitude less than δ with a probability of 95%".

The two last items in the list will be examined more closely in the next two sections. But before we go into that we mention that there are other ideas to model validation. One of these is based on a splitting of the model error in two parts

$$\varepsilon(t) = \tilde{f}(u(t)) + v(t), \tag{2.70}$$

where the first part is due to unmodeled dynamics and the other part is due to noise. See the papers by Smith and Doyle (1992), Kosut (1995) and Poolla et al. (1994) among others as starting points for a study on that subject. Briefly, the idea is to try to compromise between the uncertainties from unmodeled dynamics and those from disturbances. The formulation used resembles quite closely with the ones used in robust control. Note that these approaches mostly *do not* assume independence between the noise and the input. This is somewhat contradicting the general idea about noise: Noise is something that does not change if the input is changed. If the noise changes it contains unmodeled dynamics. That is, we are arguing that a

model validation test needs to check whether or not the noise is independent (or uncorrelated, which is a weaker condition) of the input.

2.4.2 A Whiteness Test

To check for whiteness of the residuals we compute the covariance estimate

$$\hat{R}_\varepsilon^N(\tau) = \frac{1}{N}\sum_{t=1}^{N-\tau}\varepsilon(t)\varepsilon(t-\tau),$$

where $\varepsilon(t) = \varepsilon(t, \hat{\theta}_N)$. If $\varepsilon(t)$ is a white-noise sequence with variance λ one can show that

$$\sqrt{N}\begin{bmatrix}\hat{R}_\varepsilon^N(1)\\ \vdots \\ \hat{R}_\varepsilon^N(M)\end{bmatrix} \in \mathrm{AsN}(0, \lambda^2 \cdot I).$$

This gives that (assuming $N \gg M$)

$$\frac{N}{\left(\hat{R}_\varepsilon^N(0)\right)^2}\sum_{\tau=1}^{M}\left(\hat{R}_\varepsilon^N(\tau)\right)^2 \in \chi^2(M). \tag{2.71}$$

One could also perform a more graphically oriented test, where $\hat{R}_\varepsilon^N(\tau)$ is plotted together with two straight lines located at $\pm\hat{\lambda}\cdot\Phi((1+\alpha)/2)$, where $\Phi(x) = P(X \leq x)$ for $X \in \mathrm{N}(0, 1)$. If the covariance function lies 'well outside" these two lines we say that there exists some correlation at that specific lag. Note, however, that this test does not have simultaneous confidence degree of level α. It is only possible to say that the simultaneous confidence degree is greater than or equal to $1 - M \cdot \alpha$ using Bonferroni's inequality, see Section 3.2.

2.4.3 A Cross-Correlation Test

An expression similar to that for the whiteness test is obtained when testing for independence between the residuals and the inputs (Ljung, 1999b). Let $M = M_2 - M_1 + 1$ and form

$$r_M^N = \frac{1}{\sqrt{N}}\sum_{t=1}^{N}\varepsilon(t)\begin{bmatrix}u(t-M_1)\\ \vdots \\ u(t-M_2)\end{bmatrix}. \tag{2.72}$$

If $\varepsilon(t)$ and $u(t)$ are independent sequences we have that

$$[r_M^N]^T P^{-1}[r_M^N] \in \chi^2(M), \tag{2.73}$$

where the (k, l)th element of P is given by

$$P_{(k,l)} = \sum_{\tau=-\infty}^{\infty} R_\varepsilon(\tau) R_u(\tau - k + l). \tag{2.74}$$

It is also possible to construct a graphically oriented test of this statistic in the same spirit as above. Then one utilizes that

$$\sqrt{N}\hat{R}_{\varepsilon u}^N(\tau) \in \text{AsN}(0, P_1) \tag{2.75}$$

$$P_1 = \sum_{\tau=-\infty}^{\infty} R_\varepsilon(\tau) R_u(\tau). \tag{2.76}$$

The rest follows from the discussion above.

Example 2.1

We illustrate a typical graphical model validation test with Figure 2.1. The upper plot shows the whiteness test described in Section 2.4.2 and the lower plot shows the cross-correlation test described in this section. The nominal model and underlying system is irrelevant for the discussion. We see that there is no evidence of the residuals being non-white, but there is some significant cross-correlation between the residuals and the input signal, especially at lag $\tau = 5$. The nominal model will therefore be falsified by this model validation test. ❑

This example does not only show how the tests are performed and interpreted, but it also shows the weakness of this kind of test. We do not get any feeling for which type of errors that are present. It could be the case that the model is falsified by the tests, but the errors present are virtually harmless for the intended model use.

2.4.4 Other Types of Tests

Since we throughout the thesis assume that the measurement noise acting on the system will be described by a stochastic process, we naturally end up with statistical model validation tests based on standard confidence intervals and hypothesis tests. This means that asymptotic distributions for other statistics associated with the model can be calculated. This includes the step response, the poles and zeros of the system, and the frequency function of the system. A test could then be performed

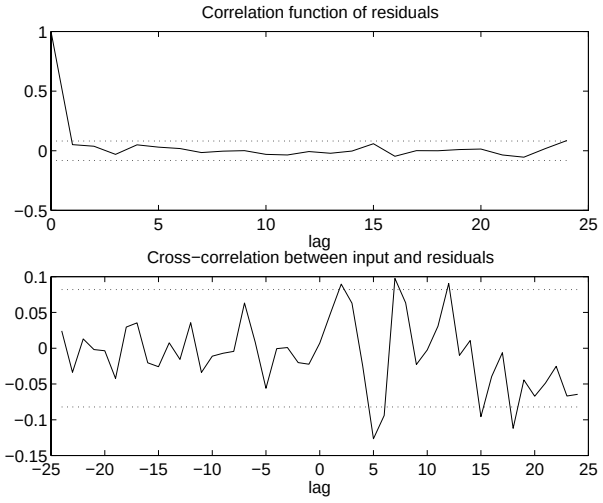


Figure 2.1 *Results from a typical model validation test. The upper plot shows a whiteness test of the residuals and the lower plot shows a cross-correlation test between the residuals and the inputs.*

by plotting the confidence regions calculated from the nominal model together with plots of the statistic calculated from fresh validation data.

The calculation of the distributions for these kind of statistics is based on a linear approximation of the mapping from the parameter distribution given by (2.17-2.19) to the statistic of interest. This mapping is usually referred to as Gauss' approximation formula. It states that if $\hat{\theta}_N$ is sufficiently close to $\theta^* = \mathrm{E}\,\hat{\theta}_N$, we can make the approximation

$$\mathrm{Cov}\; f(\hat{\theta}) \approx \left[f'(\theta^*)\right] P_\theta \left[f'(\theta^*)\right]^T \approx \left[f'(\hat{\theta}_N)\right] P_\theta \left[f'(\hat{\theta}_N)\right]^T. \tag{2.77}$$

Furthermore, if the higher order derivatives of f are small, this approximation will be good, and if $\hat{\theta}_N$ is asymptotically Gaussian distributed, so will $f(\hat{\theta}_N)$.

If we, for instance, are interested in uncertainty bounds for the frequency function we get from (2.77)

$$\mathrm{Cov}\begin{bmatrix}\mathrm{Re}\,\hat{G}_N(e^{i\omega}) \\ \mathrm{Im}\,\hat{G}_N(e^{i\omega})\end{bmatrix} \approx \frac{1}{N} P(\omega) = \begin{bmatrix}\mathrm{Re}\,\hat{G}'_\theta(e^{i\omega},\hat{\theta}_N) \\ \mathrm{Im}\,\hat{G}'_\theta(e^{i\omega},\hat{\theta}_N)\end{bmatrix} P_\theta \begin{bmatrix}\mathrm{Re}\,\hat{G}'_\theta(e^{i\omega},\hat{\theta}_N) \\ \mathrm{Im}\,\hat{G}'_\theta(e^{i\omega},\hat{\theta}_N)\end{bmatrix}^T. \tag{2.78}$$

We also know that the real and imaginary are jointly normally distributed. This can be used to construct confidence regions for the frequency function in the Nyquist diagram.

3

A Note on Confidence Regions

In this chapter we will discuss some basic facts about confidence regions. The most important topic in this section is the discussion regarding the concept *simultaneous confidence degree*. Section 3.2 contains some discussion on how a single confidence region relates to a set of confidence intervals calculated from the same data, how these relations can be used, and how the intervals could be interpreted.

3.1 Confidence Intervals and Regions

In the discussion to follow we will assume that we have N independent observations, $\mathbf{x} = (x_1, \ldots, x_N)$, of a stochastic variable X having a distribution function, $F_X(x)$, depending on some unknown d-dimensional parameter vector, θ, i.e., $F_X(x) = F_X(x|\theta)$. From the observations we aim at constructing a *point estimate* of θ. This estimate will be a function of the observations

$$\hat{\theta}_N = \hat{\theta}_N(\mathbf{x}). \tag{3.1}$$

The quality of this estimate realtes to whether or not it converges to the true parameter value θ as the number of observations increases (to infinity), i.e., whether

$$\hat{\theta}_N \to \theta, \quad \text{as } N \to \infty \tag{3.2}$$

or not. If (3.2) holds, we say that the estimate is *consistent*. This means that the distribution of $\hat{\theta}_N$ converges to a point distribution. When discussing the quality

of point estimates we are also concerned with the rate at which the distribution converges. One measure of this is how the width of a confidence interval for the parameter estimate depends on the number of observations, N. For a single parameter one forms a *confidence interval* of (confidence) degree α as

$$\bar{\theta}_1(\mathbf{x}) < \theta < \bar{\theta}_2(\mathbf{x}), \tag{3.3}$$

where $\bar{\theta}_1$ and $\bar{\theta}_2$ are chosen such that repeated constructions of the interval (from new observations) include the value θ a portion α of the time. This can be rephrased as that the interval covers the true parameter value with probability α. It is important to note that the parameter value θ is fixed and the interval is stochastic (since it depends on the observations). We illustrate the construction with a simple example.

Example 3.1

Assume that $\mathbf{x} = (x_1, \ldots, x_N)$ are independent observations from a stochastic variable X with normal distribution with unknown mean, θ, and variance 1. The point estimate of θ is the arithmetic mean

$$\hat{\theta}_N = \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

From this we get that the estimate is normally distributed according to

$$\hat{\theta}_N \in \mathrm{N}(\theta, \frac{1}{N}),$$

which means that

$$\mathrm{Prob}(\hat{\theta}_N - \frac{1.96}{\sqrt{N}} < \theta < \hat{\theta}_N + \frac{1.96}{\sqrt{N}}) = 0.95.$$

This gives an interval with 95% confidence degree of the form

$$I_\theta = (\hat{\theta}_N - \frac{1.96}{\sqrt{N}}, \hat{\theta}_N + \frac{1.96}{\sqrt{N}}).$$

❑

In this example the length of the interval shrinks at a rate inversely proportional to the square root of N. This is typical for most estimates, but could of course be quite different.

For a vector θ of parameters one constructs confidence regions instead of intervals. These will typically be of the form

$$\{\theta \mid \mathcal{P}(\theta) \leq C_\alpha\}, \tag{3.4}$$

where $\mathcal{P}$ depends on the distribution of $\hat{\theta}_N$ and C_α is chosen such that the confidence degree equals α. One should note that there are several ways of choosing the limits of the region. The most common way is to choose the limits such that the region becomes the smallest possible (measured in the 2-norm), situated around the point estimate. Under the assumption that $\hat{\theta}_N$ is normally distributed with covariance matrix P the region is therefore commonly chosen as

$$\{\theta \mid (\theta - \hat{\theta}_N)^T P^{-1} (\theta - \hat{\theta}_N) \leq C_\alpha\}. \tag{3.5}$$

The limit setting value C_α is in this case defined through $\text{Prob}(X \leq C_\alpha) = \alpha$, $X \in \chi^2(d)$. This is used several times in the thesis.

3.2 Simultaneous Confidence Degree

When constructing confidence regions in dimensions higher than 3, it becomes difficult to illustrate the resulting regions in diagrams. One therefore has to project the results to intervals or regions in lower dimensions. This is typically the case when one constructs confidence bands for the amplitude curve in a system identification application. Here one constructs several intervals at different frequencies and connects these intervals with straight lines to form a confidence band. The problem that occurs in such projections is that they, in almost every case, involve some approximations or some conservatism.

Before discussing the solution to high-dimensional problems, we present the well known Bonferonni inequality (Manoukian, 1986). Let the scalar components of the parameter vector be denoted θ_i and the intervals corresponding to them I_{θ_i}, with $i = 1, \ldots, d$. Then the simultaneous confidence degree is

$$\begin{aligned} \text{Prob}(\theta_i \in I_{\theta_i} \; \forall \, i) &= 1 - \text{Prob}(\theta_i \notin I_{\theta_i} \text{ some } i) \\ &\geq 1 - \sum_{i=1}^{d} \text{Prob}(\theta_i \in I_{\theta_i}) = 1 - \sum_{i=1}^{d} (1 - \alpha_i). \end{aligned} \tag{3.6}$$

From this it also follows that the bound is tight if all intervals are independent. The main use of the inequality is to design individual confidence intervals such that they together obtain a predefined simultaneous confidence degree. However, it is important to note that this bound can be very conservative if the intervals are

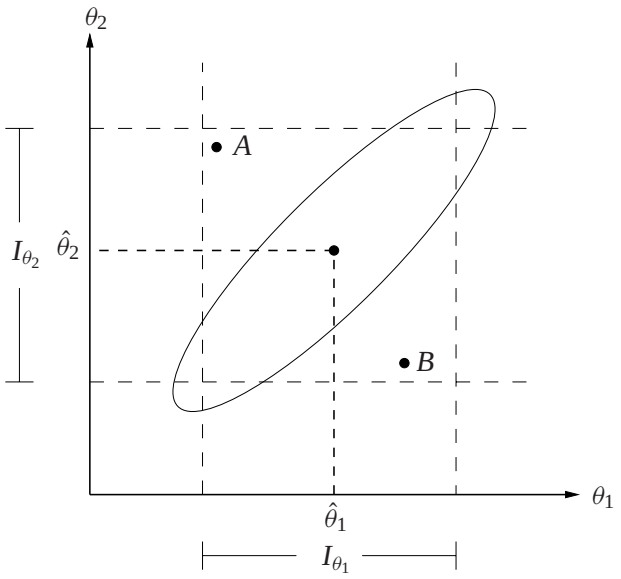


Figure 3.1 *Illustration of simultaneous confidence degree.*

heavily dependent or if many intervals are constructed. For example, constructing 30 intervals, each of degree 99%, will give a simultaneous confidence degree of at least 70%. In most applications this is of little use.

Figure 3.1 illustrates the basic problem with interpreting and constructing confidence intervals that should be of a certain simultaneous confidence degree. Assume that two parameters θ_1 and θ_2 have been estimated from data and that the estimates are normally distributed. A 90% confidence region for the two parameters calculated as in (3.5) is depicted as the ellipse in Figure 3.1. Using only the diagonal elements of the covariance matrix one obtains the intervals I_{θ_1} and I_{θ_2} (also included in the figure) of degree 95% each. Using the Bonferroni inequality (3.6) we can guarantee a simultaneous degree of at least 90%. The depicted elliptical confidence region follows the level curves of the 2-dimensional normal distribution and therefore give a good description of where the true values are. If we however try to interpret the two confidence intervals in a simultaneous manner, we could be led to the conclusion that the points A and B are probable points for the true θ-value, which they obviously are not. The rectangular area defined by the two intervals is still a valid confidence region of confidence degree at least 90%, but it does not provide any information about level curves of the probability density function. When using individual intervals to interpret a distribution of a high dimensional vector, care must therefore be taken. The intervals should serve more as guidance and illustration of the result than a description of probability mass of

the statistic. For a more thorough discussion on this subject, see Draper and Smith (1998). In Draper (1995) there is a discussion about the ratio of the volumes of ellipitical confidence regions and rectangular regions. A comparison between the Bonferroni method and a method using projections of confidence regions can be found in Nickerson (1994).

When constructing several individual intervals to be interpreted in a simultaneous way, it is easy to understand that they should be constructed as orthogonal projections of an axis parallel rectangular (or hyper rectangular) onto the axes. In this way we make no approximations in going from the confidence region to the individual intervals. The only problem that remains to be solved is how this region can be calculated. In the general case this is extremely difficult and it is therefore seldom used. These regions can however be approximately calculated when using resampling techniques such as bootstrap. This will be discussed in Section 5.3.3.

4

Model Error Modeling

Areas like model validation, identification for control, control oriented model validation have in the last decade attracted a lot of interest from researchers from wide areas in the control community. The reason for this is that one has realized that there is a lack of tools to assess and describe the errors that are present in the models. The problem with standard methods like those presented in Section 2.4 is that they suffer from the fact that the validation results are hard to interpret, see Example 2.1. Model error modeling tries to solve this problem by visualizing the model errors in the frequency domain. In this chapter we will discuss different approaches to estimate the model error and we will also try to describe what we think are the strengths and the weaknesses of the approaches.

4.1 The Concept

The concept of *model error modeling* (MEM) was introduced in Ljung (1997a) and Ljung (1998), and was further discussed in Ljung (1999a). The idea behind the concept is to explicitly estimate the model error and its uncertainty. This in order to find out if and where the model errors are significant. MEM is closely related to standard model validation tests, but there is one important difference. The aim is to present the model errors in a control oriented fashion, and MEM tries to achieve this by displaying the models deficiencies in the frequency domain.

The basic setup in model error modeling is that we are given a nominal model

$\hat{G}$ and measurements Z^N (validation data) of some underlying system G_0. Irrespectively of how the model was obtained, we aim at answering the question:

Can we cope with the model errors that are present in this model?

To answer this question we compute the simulation error (on the fresh validation data set Z^N)

$$\varepsilon(t) = y(t) - \hat{G}(q)u(t). \tag{4.1}$$

Assuming that $y(t)$ is generated from a system where the disturbances enters additively we end up in the same split of the residuals as in (2.70). Moreover, if we assume that system that has generated $y(t)$ is a linear time-invariant (LTI) system, $G_0(q)$, we get

$$\varepsilon(t) = G_0(q)u(t) + v(t) - \hat{G}(q)u(t) = \Delta(q)u(t) + v(t). \tag{4.2}$$

The rather bold assumptions of an underlying LTI system with additive noise may be somewhat naive in a real world situation, but it makes it easier to illustrate the model error modeling idea and it is also a starting point for this work. (In the next section we discuss how non-linearities, like in (2.70), can be taken care of.)

Model error modeling is all about explictly estimating the model error, $\Delta(q)$, and its uncertainty. These two estimates together provide the information about the size of the model error that is needed. This becomes extremely useful if the errors are presented in the frequency domain. We illustrate this with an example.

Example 4.1

In this example we have computed a model error model on the same data that was used in Example 2.1 (the way the MEM was estimated is immaterial for this example). Figure 4.1 shows how the model errors could be displayed using the MEM concept. The lower plot shows the estimated model error (dashed line) and the uncertainty in that estimate (shaded area). The upper plot shows the true system (solid line), the nominal model (dashed line), and the uncertainty area from model error added to the nominal model (shaded). From this figure we can draw the following conclusions about the nominal model:

- In the frequency range $0-0.06$ rad/s, there is no evidence of model errors. This conclusion can be drawn since the uncertainty contains the zero amplitude in this region, i.e., the model error does not significantly deviate from zero.
- Between 0.06 and 1.5 rad/s there is a substantial model error which could be as high as 10.

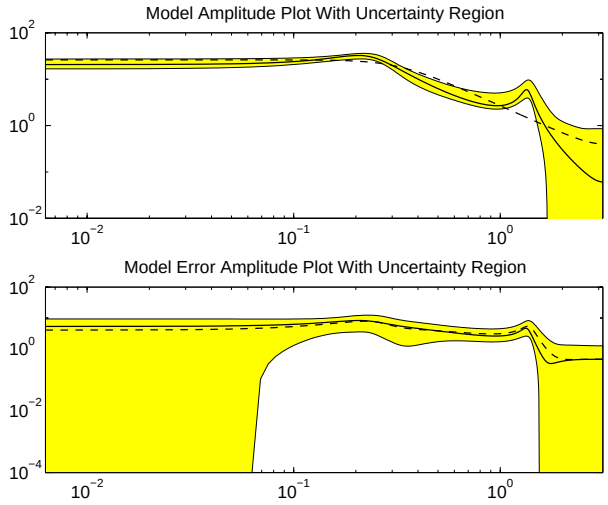


Figure 4.1 *Lower part: The estimated model error model (dashed), the true model error (solid) and the uncertainty region for the estimate. Upper part: True system (solid), nominal model (dashed) and the uncertainty from the model error model.*

- If the bandwidth of the closed loop system is supposed to be lower than, say, 0.8 rad/s the nominal model may still be useful despite the model errors. It is all up to the control designer to decide whether or not the model errors are too big to build a successful controller.

This should be compared with the results in Example 2.1. Here the conclusion was that the model was falsified because of statistical evidence that $\hat{R}_{\varepsilon u}(5) \neq 0$. From this it is pretty obvious that the conclusions we are able to make using the MEM perspective tells us much more about the deficiencies in the nominal model. ❑

This simple example leads us to the following claims:

1. MEMs provide a more control oriented visualization of the model errors.
2. MEMs make it possible to safely use falsified models.
3. MEMs allow us to use simple nominal models. The uncertainty may, however, be described by much more complicated structures.

We state the first two claims since the model error model gives us bounds on the model error in the frequency domain. Most robust control design techniques,

such as H_∞ and μ (Zhou et al., 1994), are based on this type of information. If the control designer can construct a good controller for all models in the uncertainty region, it does not matter that nominal model is falsified. What matters is simply if the model is good enough for our purposes.

These thoughts differs somewhat from traditional model validation, where one has put all effort in trying to falsify and non-falsify models. One has spent less time in thinking of the importance of describing the type of model errors present. All these ideas lead us to the third claim, which might be even more controversial. Traditionally, the basic scientific principle has been to use the simplest unfalsified model. Here we would like to rephrase this:

> *Use the simplest model, with model errors of size and character that you can live with.*

This seems like a reasonable suggestion when thinking of Example 4.1. The reason why this principle has not been used previously might be the difficulty in getting an estimate of the total size of the model error (including both the bias error and the variance error). When only looking at the nominal model, we will get little information about the possible bias errors, and as described in Section 2.1.3 it is also more complicated to estimate the variance error in case of undermodeling. The idea behind the third claim can be pushed to its limit using the following thought experiment (adopted from Ljung (1999a)): Suppose that in a system identification experiment we use an input signal to a linear system containing only two sinusoids. The simplest unfalsified model would then be a second order model. It is even impossible to invalidate this model, irrespective of the underlying linear system. We therefore claim that it is not necessary that the first unfalsified model is the best one to use.

We also mention that recently the MEM idea was picked up by Garulli and Reinelt (1999). Here the MEM concept is cast into a set membership framework. All ideas are the same in this setup, except that the MEM results should be interpreted as exact deterministic bounds. One problem that occurs in the set membership framework, and that does not occur in the traditional framework, is that *a priori* knowledge of the maximum value of the residuals is necessary. This is typically unknown since we do not know the model errors. This has then to be solved by some trial and error guesses. Related to this discussion is the survey by Reinelt et al. (1999), where the authors compare MEM, stochastic embedding, and set membership identification from their model error describing properties.

The problem with MEM is that there is no easy solution on how to estimate the model error. We will here discuss three approaches. The first one uses high order parametric models and follows the ideas in Ljung (1999a). The second

approach uses non-parametric frequency function estimation methods like the *empirical transfer function estimate* (ETFE) discussed in Section 2.2. Finally, the third approach is based on *local polynomial regression*, see Section 2.3. Here one uses local models to smooth the ETFE. The potential benefit with this approach is that it is more or less automatic, i.e., no parameters have to be chosen.

4.2 A Parametric Approach

In this section we will try to give some guidelines regarding the choice of model structures and model orders for parametric model error models. Another important aspect is how to quantify the size of the model error. This is quite straightforward in the linear case, but becomes a more open question when using non-linear structures.

4.2.1 Choice of Model Structure

The choice of model structure selection for MEMs is not an easy problem. The only known attempt to present something in this direction was given in Ljung (1999a). The idea in this article is basically to use a rich model structure like the Box-Jenkins (BJ) structure (Section 2.1.4) in combination with a non-linear model based on, e.g., neural nets, to model the model error. This guideline is not very precise and it is probably hard to give a more exact answer to this problem. From the discussion to follow it will however be argued for the use of rather high order models to ensure unbiased estimates.

Non-linear models can be modeled in many different ways. We propose the use of a FIR-type non-linear structure to detect the presence of non-linear effects in the input-output data. Let the model be described by

$$\varepsilon(t) = f(u(t), \dots, u(t-M+1), \eta) + v(t), \tag{4.3}$$

where η represents the parameterization of the model and M corresponds to the number of lagged inputs used in the model structure. M could probably be taken to a relatively small number, like in the order 3 – 7, if we only aim at detecting the presence of non-linearities. If we also would like to use the estimate to access what types of non-linearities that are present, this number should be adopted for optimal fit, and lagged outputs should be included in the structure.

As already pointed out it is important to use a rich model structure that can capture the unmodeled dynamics well. It is also clear that the order has to be high, as to extract deficiencies in the nominal model, like resonances. Since the model error is built up from both the underlying system and the nominal model

$$\Delta = G_0 - \hat{G}, \tag{4.4}$$

we find that the model error structure will be more complicated and have higher order than the nominal model itself. A guideline will therefore be to choose a BJ model structure of order *at least* twice the order of the nominal model.

We also mention that the MEM has to be validated itself to be of any use. This is to ensure that it does not contain any bias error. It is clear that if the MEM contains bias we are back to the problem we wanted to solve in the beginning, i.e., getting a full description of the model error.

Finally, we note that in the actual identification phase it is important that the linear model is estimated before the non-linear model is estimated. This to separate the linear part from the non-linear one.

4.2.2 The Size of the Error Model

No matter how the MEM is constructed it is important to measure the size of the model (in our framework we are actually interested in some probability measures of the size uncertainty). For linear parametric models this will impose no problems. Uncertainty descriptions are provided from the discussions in Sections 2.1.3 and 2.4. As already demonstrated in the previous section, the linear models are preferably displayed in the Bode plot. In this way we automatically get some sort of engineering feeling for the errors. We also have the natural interpretation of a model being unfalsified in a certain frequency range when the models uncertainty covers the zero amplitude.

Things become a little bit more complicated when estimating non-linear structures. The size of the non-linearity could of course be judged by checking whether or not the estimated parameters are significantly different from zero, using some hypothesis test. This will then be in correspondence with the standard system identification view. It is however also possible to measure the size of the non-linearity by calculating its sup-norm (or other well-defined norms)

$$\|\hat{f}\|_\infty = \sup_{u(t),\dots,u(t-M+1)} \frac{|f(u(t),\dots,u(t-M+1),\hat{\eta})|^2}{\sum_{k=0}^{M-1} u^2(t-k)}. \tag{4.5}$$

This norm has then to be thresholded in some way. The choice of threshold is however not obvious. One idea could be to map the confidence region for $\hat{\eta}$ to a interval for $\|f(u,\hat{\eta})\|_\infty$ and check if this interval is close to cover zero or not.

4.2.3 A Suggestion

To summarize, it is clear that there are some non-easy problems to solve when using parametric model error models. This makes it hard to give any exact guidelines

regarding the choice of the model order. In Ljung (1999a) the author suggests the following "default choice" (if no specific prior information is available).

$$\varepsilon(t) = \sum_{k=0}^{20} b_k u(t-k) + g_{NN}(u(t), \ldots, u(t-5), \eta) + \frac{1 + c_1 q^{-1} + \ldots + c_4 q^{-4}}{1 + d_1 q^{-1} + \ldots + d_4 q^{-4}} \tag{4.6}$$

Here g_{NN} is a neural network black box model of NFIR-type (Haykin, 1994; Sjöberg, 1995).

4.3 The Non-Parametric Approach

To get around the problems of model structure and model order selection in parametric MEM, one might be to try non-parametric frequency domain identification methods. We will describe how the smoothed ETFE (Section 2.2) can be used to estimate the model errors. This has the potential benefit of being free from structural selection problems, but on the other hand involves a smoothing problem.

4.3.1 Bandwidth Selection and Variance Properties

The assumption we base the modeling on is the split of the residuals in an LTI part originating from the input and an additive noise part

$$\varepsilon(t) = \Delta(q)u(t) + v(t). \tag{4.7}$$

The MEM is estimated as a smoothed version of the fraction of the Fourier transforms of the residuals, $\varepsilon(t)$, and the input, $u(t)$. For model error modeling purposes this approach involves the problem of choosing an appropriate γ^{-1}. As already been noted in (4.4), the model error could very well be a more complicated structure. For instance, a small bias in the estimate of a resonance frequency will make the model error contain both a resonance peak and a resonance dip. This will make it necessary to choose a small frequency window, i.e., a large γ, in order not to decrease the frequency resolution too much. This will in turn give noisy estimates in regions where the model error is "flat". Because of this, the final choice of γ^{-1} will probably be slightly larger than in standard system identification. An initial bandwidth choice could then be $\gamma \approx N/10$.

The uncertainty description of the non-parametric MEM follows easily from the results in Section 2.2.2. The variance of the estimate is obtained from

$$\mathrm{E}\left|\hat{\Delta}(e^{i\omega}) - \mathrm{E}\,\hat{\Delta}(e^{i\omega})\right|^2 \approx \frac{1}{N}\bar{W}(\gamma)\frac{\Phi_v(\omega)}{\Phi_u(\omega)}. \tag{4.8}$$

Here everything is known except the noise spectrum, $\Phi_v(\omega)$. It can be estimated by

$$\hat{\hat{\Phi}}_v(\omega_k) = \frac{1}{N}\left|Y_N(\omega_k) - \hat{\Delta}(e^{i\omega_k})U_N(\omega_k)\right|^2. \tag{4.9}$$

If the estimate of $\Phi_v(\omega)$ is too noisy, it can be smoothed using some techniques described in Section 4.2 or in Section 4.4.

4.3.2 Connection to Cross-Correlation Tests

It may seem like classical cross-correlation tests and model error modeling are two rather different subjects. There is however a quite strong connection between the two. We will illustrate this by showing how non-parametric frequency domain methods relate to the validation test given in Section 2.4.3. When constructing test statistics for the cross-correlation between residuals and inputs we had that if $\varepsilon(t)$ and $u(t)$ are independent

$$r_M^N = \frac{1}{\sqrt{N}}\sum_{t=1}^{N}\varepsilon(t)\begin{bmatrix}u(t-M_1)\\ \vdots \\ u(t-M_2)\end{bmatrix} = \sqrt{N}\begin{bmatrix}\hat{R}_{\varepsilon u}^N(1)\\ \vdots \\ \hat{R}_{\varepsilon u}^N(M)\end{bmatrix}$$

is normally distributed with mean value zero and covariance matrix P, where the (k,l)th element is given by (2.74). Note here that this assumption is the same as assuming that $\Delta(e^{i\omega}) \equiv 0$ in the model

$$\varepsilon(t) = \Delta(q)u(t) + v(t).$$

This is the fundamental link between the model validation tests and MEM.

From r_M^N we can form the estimate of the cross-spectrum between the residuals and the inputs as

$$\hat{\Phi}_{\varepsilon u}(\omega) = \sum_{k=M_1}^{M_2}\hat{R}_{\varepsilon u}(k)e^{-i\omega k} = \frac{1}{\sqrt{N}}W^* r_M^N, \tag{4.10}$$

where

$$W = \begin{bmatrix}e^{i\omega M_1} & \dots & e^{i\omega M_2}\end{bmatrix}^T. \tag{4.11}$$

Here $(\cdot)^*$ denotes transpose and complex conjugate. The ETFE estimate is then formed by dividing the cross-spectrum with the input spectrum, i.e.,

$$\hat{\Delta}(e^{i\omega}) = \frac{\Phi_{\varepsilon u}(\omega)}{\Phi_u(\omega)} = \frac{\Xi(\omega)}{U(\omega)}, \tag{4.12}$$

where $\Xi(\omega)$ is the Fourier transform of the residuals. It now follows from (2.39-2.40) that

$$\hat{\Delta}(e^{i\omega}) \in \text{AsN}(0, \frac{\Phi_v(\omega)}{\Phi_u(\omega)}). \tag{4.13}$$

Hence we come to the conclusion that the cross-correlation test in Section 2.4.3 has a very strong connection to the test that the MEM non-parametric frequency domain performs. We therefore would like to recommend that non-parametric MEM should be prefered compared to classical cross-correlation tests. The tests are more or less equivalent, but the interpretation of the results differs a lot, cf. Examples 4.1 and 2.1.

4.4 Estimating the Model Error Model Using Local Polynomial Regression

We shall here point out how local polynomial regression can be used to obtain an estimate of the model error model. It is all a matter of mapping the framework in Section 2.3 into a frequency domain setting. The methodology is the same as the one used in Stenman (1999, Chapter 6) to smooth the ETFE. The difference is that we here smooth the ETFE of the model error, $\Delta(e^{i\omega})$.

The benefit with this approach is that local polynomial modeling is the one of the most unprejudiced ways of modeling data. We essentially only assume that we have noisy measurements of some smooth function. In the model error modeling case this function is the frequency function of the model error. The smoothness of this curve is therefore ensured by the physical interpretation of the problem. We belive that this is the most natural way of solving the difficult "order selection" problem which occurs in all modeling approaches. It is essential to note that we have much information about the system in this case, since we do not assume any knowledge of model orders or model structures. This will lead to larger uncertainties in the estimates than if a parametric model had been used.

The underlying model assumption (2.50) will in this setting be

$$\hat{\hat{\Delta}}(e^{i\omega_k}) = \Delta(e^{i\omega_k}) + e_k. \tag{4.14}$$

This is, we have noisy measurements of the model error at a set of evenly spread frequencies

$$Y_k = \hat{\hat{\Delta}}(e^{i\omega_k}) \in \mathbb{C}, \quad X_k = \omega_k = \frac{2\pi k}{N} \in \mathbb{R}, \quad k = 1, \ldots, n \tag{4.15}$$

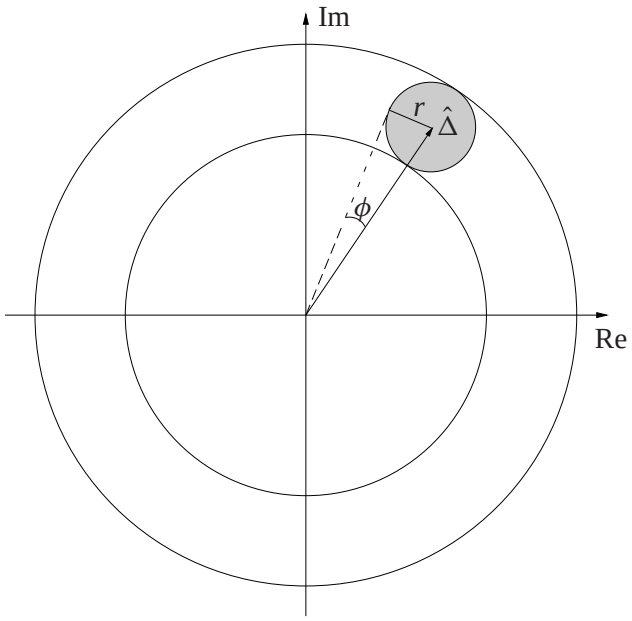


Figure 4.2 *Confidence region for the estimate $\hat{\Delta}(e^{i\omega})$ (shaded area). $r(\omega)$ and $\phi(\omega)$ shows how this region can be mapped to confidence intervals for the amplitude and phase curves. The two-dimensional torus shows the actual confidence region for amplitude curve, when this mapping is utilized.*

where $n = \lfloor (N-1)/2 \rfloor$ ($\lfloor \cdot \rfloor$ denotes flooring). We only use half of the frequencies since the ETFE is hermitian (except at frequencies 0 and π which are skipped). Moreover, the noise e_k is a zero mean complex-valued disturbance with variance

$$\sigma_k^2 = \frac{\Phi_v(\omega_k)}{|U_N(\omega_k)|^2} \tag{4.16}$$

where v and u stems from Equation (4.2). Once again we estimate $\Phi_v(\omega)$ as described by (4.9). In this expression, the unknown $\hat{G}(e^{i\omega_k})$ must be estimated. This can be done using the non-parametric methods described in Section 2.2 using a small bandwidth.

In model error modeling the focus is set on the uncertainty in the estimated model error. The uncertainty regions can be calculated using the techniques described in Section 2.3.4. However, we can give some more details about the distributions in this special case. The difference in constructing confidence intervals for the smoothed ETFE (or model error model in this particular case) compared to the general local smoothed curve lies in the fact that $\hat{\Delta}(e^{i\omega_k})$ is asymptotically

complex normally distributed. Since the real and imaginary parts of $\hat{\Delta}(e^{i\omega_k})$ are jointly normal and independent with variance

$$\text{Var Re}\,\hat{\Delta}(e^{i\omega_k}) = \text{Var Im}\,\hat{\Delta}(e^{i\omega_k}) = \frac{\sigma^2\|\mathbf{w}\|^2}{2} \tag{4.17}$$

we get

$$\frac{2\cdot\left|\hat{\Delta}(e^{i\omega_k}) - \Delta(e^{i\omega_k})\right|^2}{\|\mathbf{w}\|^2} \in \chi^2(2), \tag{4.18}$$

with $\mathbf{w}$ defined by (2.65). The variance in the data is unknown but can be estimated using (2.68). Constructing a 95% confidence interval for $\hat{\Delta}$ yields

$$|\hat{\Delta}(e^{i\omega}) - \Delta(e^{i\omega})| \le \sqrt{\frac{\chi^2_{0.95}(2)}{2}}\cdot\sigma(\omega)\|\mathbf{w}(\omega)\| = r(\omega). \tag{4.19}$$

This region can be mapped to approximate confidence bands for the amplitude and phase curve. In this translation, some conservatism has to be incorporated, since we are mapping a two-dimensional object to a one-dimensional one. The mapping is explained by Figure 4.2.

We see that

$$\phi(\omega) = \arcsin\frac{r(\omega)}{\hat{\Delta}(e^{i\omega})}, \tag{4.20}$$

so the approximate intervals for the amplitude and phase becomes

$$I_{|\Delta(e^{i\omega})|} = \left(|\hat{\Delta}(e^{i\omega})| - r(\omega), |\hat{\Delta}(e^{i\omega})| + r(\omega)\right) \tag{4.21}$$

$$I_{\arg\Delta(e^{i\omega})} = \left(\arg\hat{\Delta}(e^{i\omega}) - \phi(\omega), \arg\hat{\Delta}(e^{i\omega}) + \phi(\omega)\right). \tag{4.22}$$

From Figure 4.2 we see that these intervals can be quite conservative. If we only study the confidence interval for the amplitude, we do not take the phase into account at all. This interval will therefore cover more systems than what was intended. Similar results are obtained for the phase.

4.4.1 Handling Non-Linearities

The local polynomial modeling approach can easily be extended to handle non-linearities. This is done in Stenman (1999, Chapters 5 and 7). This concept is named Model On Demand (MOD). The idea is that we build local models on-line

when they are needed. The reason for this can be that system is time-varying or that it is hard to build global models of th e system. In order to achieve these local models (local mainly in the regressor space, but also in time if needed) one identifies the quantities in (2.50) with the following *dynamic* data generating equation

$$y(t) = m(\varphi(t)) + v(t) \tag{4.23}$$

Here $\varphi(t)$ are the so called *regressor vectors* consisting of lagged inputs and outputs. The local modeling problem now aims at choosing the weights in order to minimize the mean square prediction error, i.e., estimate the best predictor

$$\hat{y}(t) = \sum_{k=1}^{N} W_k(\varphi(t))y(k) \tag{4.24}$$

using weights chosen to minimize

$$\mathrm{E}(y(t) - \hat{y}(t))^2. \tag{4.25}$$

One important difference from the smoothing of the ETFE is that the regressor variables now are vectorized. This will force us to introduce other measures of distance. One natural way to do this is to introduce some distance function $d(\cdot,\cdot)$ which measures the distance between regressor points, $\varphi(k)$, and the point where the model will be fitted, $\varphi(t)$. Common choices are

$$d(\varphi(k), \varphi(t)) = \|\varphi(k) - \varphi(t)\|_M, \tag{4.26}$$

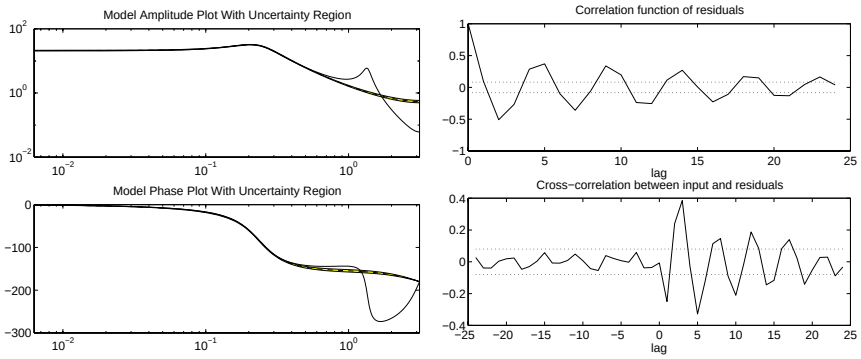
where M is a positive (semi-) definite matrix. The weights are then given by

$$w_k(x) = K\left(\frac{d(\varphi(k), \varphi(t))}{h}\right). \tag{4.27}$$

In an off-line situation like model error modeling, it is possible to estimate new low order local model at every sample time. This will give us a possibility to capture both time-varying and non-linear effects. Measuring the reduction of the size of the residuals from the nominal model to the size of the residuals from this model error model will be one quantification of the non-linear unmodeled dynamics which can be used for MEM purposes.

4.5 Model Error Modeling: An Example

Let us illustrate the different model error modeling approaches presented with an example adopted from (Ljung, 1999a). Let data be generated from the following



(a) Upper plot shows the amplitude curve for the true system (solid) and the nominal model (dashed), the lower plot shows the phase curve. The shaded area is the confidence region calculated using (2.22-2.23) and Gauss' approximation formula.

(b) A whiteness test for the residuals (upper plot) and a cross-correlation test between residuals and inputs (lower plot) for the estimated nominal model (4.29).

Figure 4.3 *Results from the estimation of the nominal model.*

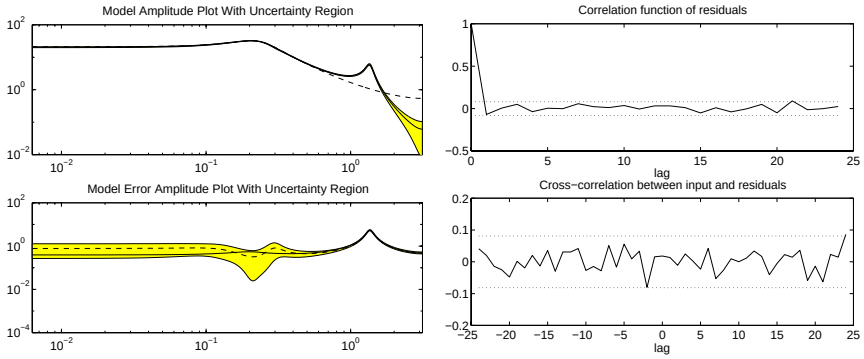
fourth order system

$$\begin{aligned} y(t) &= \frac{B(q)}{F(q)}u(t) + e(t), \\ B(q) &= q^{-1} + 0.5q^{-2}, \\ F(q) &= 1 - 2.2q^{-1} + 2.42q^{-2} - 1.87q^{-3} + 0.7225q^{-4}. \end{aligned} \tag{4.28}$$

The input, $u(t)$, chosen as a Gaussian noise sequence of length $N = 1000$, and the disturbance, $e(t)$, as an additive Gaussian noise sequence with zero mean and variance 1. Validation data was simulated using the same settings. A nominal model is estimated using a second order output-error (OE) model structure. The estimation is performed using the system identification toolbox in MATLAB (Ljung, 1997b). From this we obtain a nominal model

$$\hat{G}(q) = \frac{1.5293q^{-1} - 0.4364q^{-2}}{1 - 1.7913q^{-1} + 0.8431q^{-2}}. \tag{4.29}$$

Since the nominal model is estimated using a too low model order we suspect that this model will be falsified by a model validation test as described in Section 2.4. That this is also the case, is shown by Figure 4.3(b). We then know that the confidence regions calculated from this model will not be correct. Figure 4.3(a)



(a) The lower plot shows the estimated parametric OE(6,6,1) MEM (dashed line), the true model error (solid line), and a 95% uncertainty region obtained from the estimate (shaded area). The upper plot shows the true system (solid line), the nominal model (dashed line), and the uncertainty region obtained by adding the uncertainty region from the MEM to the nominal model.

(b) A whiteness test for the residuals (upper plot) and a cross-correlation test between residuals and inputs (lower plot) for the estimated parametric model error model.

Figure 4.4 *Results from the estimation of the parametric model error model.*

shows the results. The explanation to this is that the model uncertainty estimates are based on the assumption that $\mathcal{M} \in \mathcal{S}$ and therefore use (2.22-2.23) to estimate the parameter uncertainties.

The structures of the true system and the nominal model makes the model error have an OE(6,6,1) structure. Estimating a parametric model error model of this particular structure will then have a variance that reaches the Cramèr-Rao lower bound. This will act as a measure of what is possible to achieve using model error models. Note however that we reach this lower bound with the prior information of the correct model structure. Figure 4.4(a) shows the obtained model error estimate (dashed line, lower plot) and the uncertainty in that estimate. This plot shows that there is a significant model error throughout the frequency axis. The amplitude of the error is low for frequencies below 1 rad/s, and the nominal model misses a resonance peak at about 1.5 rad/s. Figure 4.4(b) shows the traditional model validation test of the model error model. It is seen that the parametric error model is unfalsified by this test.

To see how well one can do with nonparametric methods we smoothed the ETFE

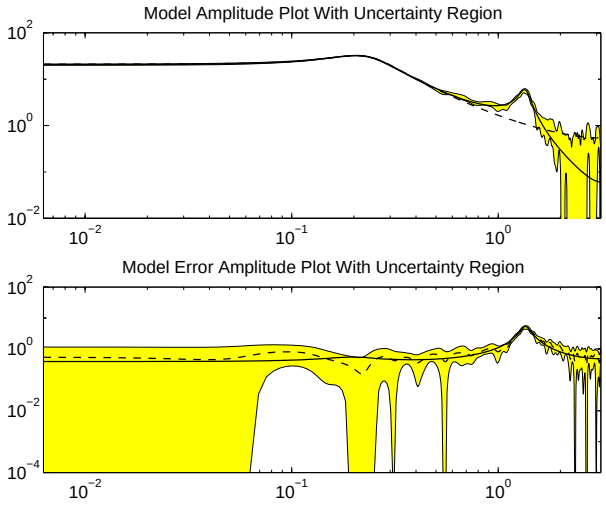
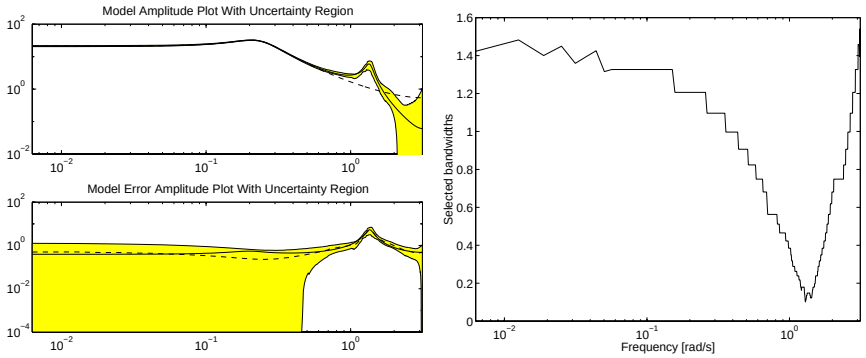


Figure 4.5 *The lower plot shows the estimated non-parametric MEM using* $\gamma = 100$ *(dashed line), the true model error (solid line), and a* 95% *uncertainty region obtained from the estimate (shaded area). The upper plot shows the true system (solid line), the nominal model (dashed line), and the uncertainty region obtained from the MEM. The uncertainty region is obtained by adding the uncertainty region from the MEM to the nominal model.*

using a Hamming window with $\gamma = N/10 = 100$. The obtained results are shown in Figure 4.5. The estimate is quite noisy as expected. The difference compared to the parametric model is that this test says that the low frequency response of the nominal model seems to be close to the true response. The regions are also larger throughout the frequency axis. This comes as no surprise since this model contains much less prior information than the parametric model error model.

In the local polynomial approach we use a tricube kernel (2.56) and a polynomial model of degree 2. The bandwidth selection criterion is the generalized local version of Mallows C_p (2.58) with $\alpha = 5$. A high value of α is used in order to avoid spurious effects in the noise. The MEM result is shown in Figure 4.6(a) and the selected bandwidths are shown in Figure 4.6(b). The result is that the nominal model is unfalsified for frequencies lower than 0.2 rad/s. We also see that uncertainty regions are even larger in this case compared to the two previous ones. Studying the choice of bandwidths we see that selection procedure works exactly as suspected. The bandwidth is significantly smaller at the resonance peak than in, e.g., the "flat" low frequency region.



(a) The lower plot shows the estimated local polynomial MEM (dashed line), the true model error (solid line), and a 95% uncertainty region obtained from the estimate (shaded area). The upper plot shows the true system (solid line), the nominal model (dashed line), and the uncertainty region obtained by adding the uncertainty region from the MEM to the nominal model.

(b) The selected bandwidths for the local polynomial model.

Figure 4.6 *Results from the estimation of the local polynomial model.*

Figure 4.7 shows the different estimates of the model error, together with the true model error. The parametric model and the local polynomial model behave similarly, and the non-parametric model is a lot noisier.

A final comparison that could be made is looking at the model error description we would get if the nominal model used had the correct structure. This is shown in Figure 4.8(a). In Figure 4.8(b) we see the result of a standard model validation test. The uncertainty regions obtained are significantly lower than all of the previous ones, but this situation is ideal one and will in every case produce the smallest possible uncertainty regions.

This example shows that we will get the smallest possible confidence regions if we estimate a correct nominal model in the first place. This comes as no surprise from both an intuitive and a information based perspective. The example also shows that the local polynomial approach can produce very smooth and accurate estimates of the model error. The confidence regions will however be a little bit larger since the method uses very little prior information. We will once again point out the increase in interpretability when using model error models compared to standard cross-correlation tests. See Figures 4.3(b) and 4.6(a).

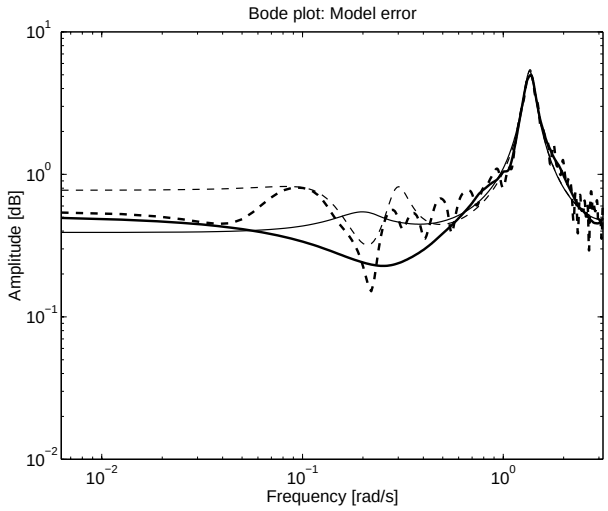
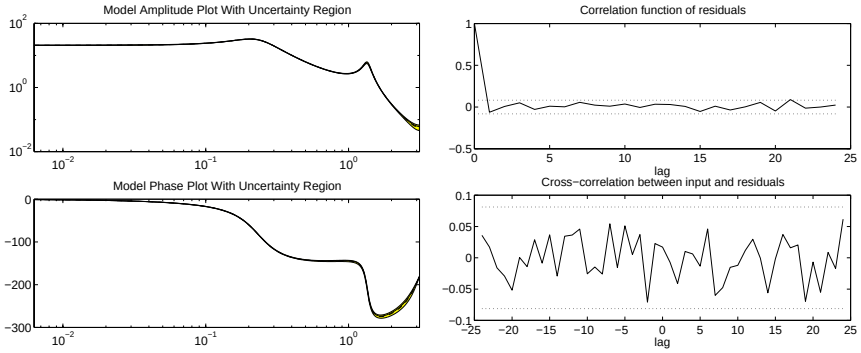


Figure 4.7 *Different estimates of the model error. Solid thin line – true model error, solid thick line – local polynomial error model, thin dashed line – parametric error model, thick dashed line non-parametric error model.*

4.6 Conclusions

We have in this chapter discussed different approaches to model error modeling. The subject is somewhat philosophical, since it does not have a unambiguous solution. The subject is still important and the discussion highlights some aspects model error modeling that have not been much discussed. It is certainly a subject that needs to be penetrated further.

The conclusions that can be drawn from this chapter are that if we know the correct model order and model structure we should *always* estimate a nominal model of this structure and order to get as much information as possible about the underlying system. If we do not have this information or we are unable to use a high order model in an application or just want to use another nominal model (obtained in any way) the MEM concept can be a very useful tool to find the deficiencies in the nominal model. If a linear MEM is sought the result is preferably displayed in the frequency domain, in the non-linear case it is somewhat more difficult display the MEM information but it is still possible. The local polynomial and the MOD approaches to model error modeling are very interesting since they represent, in some sense, the most unprejudiced way of estimating models. This will make it possible for us to forget all about the model structure and bandwidth selection problems that occur in the other approaches.



(a) Upper plot shows the amplitude curve for the true system (solid) and the nominal model (dashed), the lower plot shows the phase curve.

(b) A whiteness test for the residuals (upper plot) and a cross-correlation test between residuals and inputs (lower plot) for the estimated nominal model (4.29).

Figure 4.8 *Results from the estimation of the parametric nominal model with correct model order.*

5

Bootstrap

Most ways of characterizing the uncertainties in identified models are based on explicit asymptotic expressions like (2.17), (2.25), and (2.77). Since they are asymptotic in the length of the data record and/or the model order, their validity might not be fulfilled in every case. Fortunately, these these expressions seems to hold rather accurately already at quite modest data lengths, i.e., for $N \gtrsim 200$. Still there are situations where only short data records are available and the asymptotic results are not valid. There are also situations where we have not even been able to calculate approximate distributions of the estimates. To solve such problems we have looked at bootstrap methods to see if and where they could be applicable in system identification problems.

5.1 What is Bootstrap?

Bootstrap was introduced by Efron (1979) as a method to calculate accuracy measures of a statistic by simulations. The development of bootstrap has together with computer speed improvements made it possible to calculate uncertainty regions for a wide class of problems. Bootstrap is still in focus for a lot of research in the statistical community, and much work is spent on extending bootstrap to more complicated problems than it was originally designed for. Work has also been spent on improving bootstrap to construct more accurate confidence regions. Introductions to bootstrap can be found in Efron and Tibshirani (1993), Davison and Hinkley

(1997), Hjorth (1994), and Politis (1998). In Zoubir and Boashash (1998) a survey of signal processing applications is given. Applications to subspace identification can be found in Lovera (1997). Aronsson et al. (1998) shows how bootstrap can be used in connection with adaptive control.

The main idea behind bootstrap is that we aim at getting to a situation where we could perform Monte Carlo simulations to judge the uncertainty in the estimate we have. This in contrast of going through tedious or unfeasible calculations describing the uncertainties. To give an intuitive feeling for the bootstrap idea we will sketch the idea in a rather non-formal way.

The setup can be described as follows. Let $\mathbf{x} = (x_1, \ldots, x_N)$ be an independent, identically distributed (i.i.d.) sample from a stochastic variable X with distribution function F. We would like to estimate a statistic, $\tau = \tau(F)$, associated with the distribution F. This statistic could, for instance, be the parameter λ describing the exponential probability density distribution (PDF)

$$f(x) = \frac{1}{\lambda} e^{-\frac{x}{\lambda}}. \tag{5.1}$$

As an estimator of τ we will use T, i.e., $\hat{\tau} = T(\mathbf{x})$. This estimator will also be used to find the accuracy of $\hat{\tau}$.

Describing the accuracy of the estimate $\hat{\tau}$ would be easy to solve if F was known. Then B new samples from F, $\mathbf{x}_1, \ldots, \mathbf{x}_B$, could be generated in a Monte Carlo fashion and $\hat{\tau}_1 = T(\mathbf{x}_1), \ldots, \hat{\tau}_B = T(\mathbf{x}_B)$ could be calculated. From this, the empirical probability density function for $\hat{\tau}$ and an approximate confidence region for τ are easily constructed.

Since F is unknown, the simple and natural idea is to estimate F with the empirical distribution estimate

$$\hat{F}(x) = \frac{\#\{x_i \leq x\}}{N}, \tag{5.2}$$

where $\#\{x_i \leq x\}$ denotes the number of x_is less than or equal to x. From this we generate (bootstrap) resamples, $\mathbf{x}_j^* = (x_1^*, \ldots, x_N^*)$, $j = 1, \ldots, B$ and calculate $\hat{\tau}_1^*, \ldots, \hat{\tau}_B^*$. (Stars indicate that we are working with bootstrapped samples.) The empirical distribution for $\hat{\tau}$ is estimated with

$$\hat{F}_\tau(t) = \frac{\#\{\hat{\tau}_j^* \leq t\}}{B}. \tag{5.3}$$

From this we can easily construct approximate level α confidence intervals for τ.

The key point in using the bootstrap method is to generate resamples from an i.i.d. sample. Given such a sample and a smooth statistic, $\tau = \tau(F)$, bootstrap will generally work.

Extensions of bootstrap methods to dependent data structures, e.g., time-series, have been proposed by several authors, e.g., Freedman (1984), Bose (1988), and Nordgaard (1995). They propose different strategies to construct i.i.d. data sets associated with the measured data which can be bootstrapped. We will here illustrate one of the simplest ideas.

Assume that data, $y(1), \ldots, y(N)$, is generated by an n'th order AR-process

$$A(q)y(t) = e(t), \quad A(q) = 1 + a_1 q^{-1} + \cdots + a_n q^{-n}, \tag{5.4}$$

where $e(t)$ is white noise with zero mean. We estimate $A(q)$ by standard least-squares (2.33) and compute the residuals

$$\varepsilon(t) = \hat{A}(q)y(t). \tag{5.5}$$

Now $\varepsilon(t)$ is approximately a white noise sequence from which we can generate resamples and simulate "new" output signals according to

$$y_j^*(t) = \frac{1}{\hat{A}(q)}\varepsilon_j^*(t), \;\; j = 1, \ldots, B. \tag{5.6}$$

Here $\{\varepsilon_j^*(t)\}_{t=1}^N$, $j = 1, \ldots, B$ are drawn from the distribution $\hat{F}_\varepsilon$, constructed as in (5.2). From the simulated outputs we construct the reestimates $\hat{A}_1^*, \ldots, \hat{A}_B^*$ of $\hat{A}$. In the case the parameters $a_1, \ldots, a_n$ represent the statistics of interest, we have $\hat{\tau}_j^* = \hat{A}_j^*$, $j = 1, \ldots, B$, and the estimator $\hat{\tau} = T(y(1), \ldots, y(N))$ is represented by the mapping given by the minimization (2.33). Other statistics, such as the poles, can be computed from the estimated $\hat{A}_j^*$.

Proofs of the validity of this method can be found in Freedman (1984) and Bose (1988). In Freedman (1984) it is also shown that the same method works in case of ARX-models. Here data is assumed to be collected in open-loop and the control signal is assumed to be white noise.

5.2 Bootstrap and System Identification

From a system identification perspective the main interest in bootstrap lies in its possible ability to estimate the uncertainty in some difficult problems such as:

1. Describing the uncertainty in models with unmodeled dynamics.
2. Estimating confidence regions with simultaneous confidence degree.
3. Describing the uncertainty in subspace based identified models.

The problem with unmodeled dynamics, i.e., $\mathcal{S} \notin \mathcal{M}$, is often called the problem of undermodeling. It has historically been very difficult to find efficient methods to estimate uncertainty regions for statistics of models that are undermodeled. See Hjalmarsson (1993, Chapter 5), Hjalmarsson and Ljung (1992), Larssen (1992), and Pötscher and Prucha (1997). We will show how bootstrap and the concept model error modeling together can be used to solve this problem.

The perhaps most useful uncertainty description is confidence bands for the frequency function. These bounds are traditionally calculated using Gauss' approximation formula (2.77). In this section we will look at some ideas to directly estimate the distribution of $\hat{G}_N(e^{i\omega})$, without requiring any of the approximations involved in (2.77). These bounds are of even more interest if they are calculated with a given simultaneous confidence degree. This second problem is discussed in Section 5.3.3.

The third issue has been a problem with the successful subspace identification methods (Van Overschee and De Moor, 1996; Viberg, 1995). However, a potential solution to this problem is discussed by Lovera (1997). These ideas are very similar to those discussed in this thesis.

5.2.1 The General Idea

The idea on how to use bootstrap in system identification is clearly inspired by the one used for autoregressive models and time-series data in Section 5.1. We simply estimate the system parameters with a maximum likelihood estimator and compute the residuals from the resulting model. If these residuals pass a traditional whiteness test, we use them as the source for constructing bootstrap resamples.

In Section 5.1 we calculated the residuals to eliminate the dependence between the measurements $\{y(t)\}_{t=1}^{N}$. This idea will also work in presence of a deterministic input signal. We must however remember that the inputs are deterministic and therefore ordered. This means that when taking bootstrap resamples we cannot change the order of the input samples, $u(t),\ t = 1, \ldots, N$.

If the true system belongs to the model class, i.e., if $\mathcal{S} \in \mathcal{M}$, we know that as $N \to \infty$ we have $\hat{G}_N \to G_0$ and $\hat{H}_N \to H_0$. This also means that the residuals we compute

$$\varepsilon(t, \hat{\theta}_N) = \hat{H}_N^{-1}(q)\big(y(t) - \hat{G}_N(q)u(t)\big), \tag{5.7}$$

will converge to the "true" noise sequence $\{e(t)\}_{t=1}^{N}$. Since these are i.i.d. by assumption, it will make them good candidates for bootstrap resampling. In practice we do not know the correct model order and we have to check the whiteness of the residuals and the independence between the residuals and the input by the tests

described in Section 2.4. If the residuals cannot be shown to be non-white, we use them to draw bootstrap resamples from. If they are non-white the model is too simple and the bootstrap method will fail. The solution to this problem is to choose a more flexible model structure and/or a higher model order. The variance error of the low order model can however be estimated using bootstrap as wil be described in the next section.

Given the residuals we estimate their distribution function. The easiest method is to determine the empirical distribution function

$$\hat{F}_\varepsilon(x) = \frac{\#\{\varepsilon \leq x\}}{N}. \tag{5.8}$$

This estimate could be refined using kernel smoothing, see (Wand and Jones, 1995). In this chapter we will however only use the straightforward estimate (5.8).

The approximate distribution function (5.8) can be used to draw resampled residuals from, i.e., each noise sample is drawn with replacement from the residuals (5.7). New bootstrapped output data sequences can be generated from the residuals according to

$$y_j^*(t) = \hat{G}_N(q)u(t) + \hat{H}_N(q)\varepsilon_j^*(t). \tag{5.9}$$

This will give the bootstrapped input-output data sets

$$Z_j^{N\,*} = \{y_j^*(1),\ u(1),\ \ldots,\ y_j^*(N),\ u(N)\},\ \ j = 1, \ldots, B. \tag{5.10}$$

From each of the B generated output sequences we reestimate the system and noise models, $\hat{G}_N$ and $\hat{H}_N$. The B reestimated parameter vectors can be used to approximate the distribution function for the estimated parameters, $F_{\hat{\theta}_N}$. Confidence regions for other statistics, such as the poles and zeros of the system, can of course be computed. Simply estimate these from the B bootstrapped input-output data sets, $Z_j^{N\,*}$, and construct the empirical PDFs for these statistics.

The whole procedure above can be summarized in the following algorithm:

ALGORITHM 5.1 Bootstrap resampling in system identification

1. Estimate $G(q)$ and $H(q)$ and compute the residuals:
   $\varepsilon(t, \hat{\theta}_N) = \hat{H}_N^{-1}(q)(y(t) - \hat{G}_N(q)u(t))$
2. Check the whiteness of the residuals using for instance the test in Section 2.4.2.
3. Compute the empirical distribution of the residuals:
   $\hat{F}_\varepsilon(x) = \dfrac{\#\{\varepsilon \leq x\}}{N}$

4. Generate B resampled noise sequences from $\hat{F}_\varepsilon(x)$:
   $\left\{\varepsilon_j^*(t)\right\}_{t=1}^{N}$, $j = 1, \ldots, B$
5. Simulate the new outputs:
   $y_j^*(t) = \hat{G}_N(q)u(t) + \hat{H}_N(q)\varepsilon_j^*(t)$, $j = 1, \ldots, B$
6. Estimate $\hat{\theta}_j^*$ from the resampled data:
   $Z_j^{N\,*} = \{y_j^*(1),\ u(1),\ \ldots,\ y_j^*(N),\ u(N)\}$, $j = 1, \ldots, B$.

❑

As seen from the algorithm above, bootstrap automatically gives us another feature. We do not have to go through any more approximations to construct confidence regions for other statistics. This is in contrast to the situation we had in Section 2.4, where we had to take an extra step via Gauss' approximation formula (2.77). We calculate $\hat{\tau}_j^* = T(Z_j^{N\,*})$, $j = 1, \ldots, B$, for any statistic associated with the model. This could be the frequency response, the poles and zeros or the step response. How the confidence regions for such statistics could be approximated will be described in Section 5.3.

5.2.2 Undermodeling

Algorithm 5.1 in the previous section will not work in case of undermodeling, i.e., $\mathcal{S} \notin \mathcal{M}$. The reason for this is that the unmodeled dynamics will be present in the calculated residuals (5.7). To get rid of this influence we mix the ideas in Algorithm 5.1 with the concept of MEM in Chapter 4.

We start by estimating the system with a fixed low-order model with parameter vector estimate, $\hat{\theta}_N$, and calculate the residuals from that model

$$\zeta(t) = \hat{H}_N^{-1}(q)(y(t) - \hat{G}_N(q)u(t)). \tag{5.11}$$

It might seem stupid to use a too "small" model, but it might very well be the case that we want to use this low order model to, e.g., base a regulator design on, whether or not it is a exact description of the systems dynamic behavior.

If this model actually is too small to describe the true system, we will have a dependence between $\zeta(t)$ and $u(t)$ due to the unmodeled dynamics, i.e., we have a relation

$$\zeta(t) = \tilde{G}(q)u(t) + \tilde{H}(q)e(t) \tag{5.12}$$

To get rid of this dependence we estimate the model error, $\{\tilde{G}, \tilde{H}\}$, with, e.g., a high order ARX-model, giving $\{\hat{\tilde{G}}, \hat{\tilde{H}}\}$. This model will be an unbiased estimate

of the model error if the model order is chosen high enough (Section 2.1.4). The residuals from this model will therefore be approximately an i.i.d. sequence

$$\varepsilon(t) = \hat{\tilde{H}}_N^{-1}(q)(\zeta(t) - \hat{\tilde{G}}_N(q)u(t)). \tag{5.13}$$

The empirical distribution estimate is computed

$$\hat{F}_\varepsilon(x) = \frac{\#\{\varepsilon \leq x\}}{N}. \tag{5.14}$$

From this distribution we draw bootstrap resamples, i.e., we construct B pseudo-noise sequences $\{\varepsilon_j^*(t)\}_{t=1}^N$, $j = 1, \ldots, B$, which are taken as driving noise inputs to the system

$$\begin{aligned} y_j^*(t) =& \hat{G}_N(q)u(t) + \hat{H}_N(q)\zeta_j^*(t) \\ =& \left(\hat{G}_N(q) + \hat{H}_N(q)\hat{\tilde{G}}_N(q)\right)u(t) + \hat{H}_N(q)\hat{\tilde{H}}_N(q)\varepsilon_j^*(t). \end{aligned} \tag{5.15}$$

These outputs will have approximately the same statistical properties as the measured outputs, $y(t)$. This means that if we reestimate the fixed, low order model from the "bootstrapped" outputs, $\{y_j^*(t)\}_{t=1}^N$, $j = 1, \ldots, B$, they will also have the same statistical properties as $\hat{\theta}_N$.

Algorithm 5.1 now has to be modified according to the following:

ALGORITHM 5.2 Bootstrap resampling in case of undermodeling

1. Estimate $G(q)$ and $H(q)$ and compute the residuals:
   $\zeta(t, \hat{\theta}_N) = \hat{H}_N^{-1}(q)(y(t) - \hat{G}_N(q)u(t))$
2. Estimate the model error (from $u(t)$ to $\zeta(t)$) with a higher-order ARX model and compute the residuals from this second model:
   $\varepsilon(t) = \hat{\tilde{H}}_N^{-1}(q)(\zeta(t) - \hat{\tilde{G}}_N(q)u(t))$
3. Check the whiteness of the residuals using for instance the test in Section 2.4.2.
4. Compute the empirical distribution of the residuals:
   $\hat{F}_\varepsilon(x) = \dfrac{\#\{\varepsilon \leq x\}}{N}$
5. Generate B resampled noise sequences from $\hat{F}_\varepsilon(x)$:
   $\left\{\varepsilon_j^*(t)\right\}_{t=1}^N$, $j = 1, \ldots, B$
6. Simulate the new outputs in two steps:
   $\zeta_j^*(t) = \hat{\tilde{G}}_N(q)u(t) + \hat{\tilde{H}}_N(q)\varepsilon_j^*(t)$
   $y_j^*(t) = \hat{G}_N(q)u(t) + \hat{H}_N(q)\zeta_j^*(t)$, $j = 1, \ldots, B$

7. Estimate $\hat{\theta}_j^*$ from the resampled data:
   $Z_j^{N*} = \{y_j^*(1),\ u(1),\ \ldots,\ y_j^*(N),\ u(N)\},\ j = 1, \ldots, B.$

❑

Note that this algorithm only estimates the variance errors in the model $\hat{G}_N$. An estimate of the bias error is however given by the model error model (and could thus be used as a bias correction). Connections to the model error modeling concept in Chapter 4 should be obvious.

5.3 Constructing Uncertainty Regions

In Section 5.2 we discussed methods to obtain bootstrap estimates of the system that was being modeled. We will in this section discuss how these estimates can be used to construct approximate uncertainty regions for a given statistics. We will specifically show how we can construct uncertainty regions for $\hat{G}_N(e^{i\omega})$ using either estimated covariance matrices or simulation based methods. The problem of constructing uncertainty regions of simultaneous confidence degree will also be discussed. The reason for looking at the frequency response is that it will be used in the examples in Section 5.4.

The confidence regions constructed will be based on the assumption that the statistics we estimate will be asymptotically normally distributed. This assumption will be necessary to motivate the use of the inverse of the covariance matrix as a measure in the probability space that is regarded. A motivation for the use of this is the asymptotic results (2.17), (2.25), and (2.77).

5.3.1 Using Estimated Covariance Matrices

With estimated covariance matrices, $P(e^{i\omega_k})$, for the real and imaginary parts of the frequency response we can construct approximate confidence regions in the Nyquist plot at frequency ω_k. We utilize that

$$\begin{pmatrix} \mathrm{Re}[\hat{G}_N(e^{i\omega_k})] \\ \mathrm{Im}[\hat{G}_N(e^{i\omega_k})] \end{pmatrix} \in \mathrm{AsN}\left(\begin{pmatrix} \mathrm{Re}[G_0(e^{i\omega_k})] \\ \mathrm{Im}[G_0(e^{i\omega_k})] \end{pmatrix}, P(e^{i\omega_k}) \right). \tag{5.16}$$

This result can be motivated from (2.17) and the use of Gauss' approximation formula (2.77). As already mentioned in Section 2.4.4 we have that all smooth functions of the parameters θ will be asymptotically normally distributed. This will in turn show that this discussion easily can be adapted for other statistics.

Subtracting the expected value ($= G_0(e^{i\omega_k})$) and multiplying the result with $P(e^{i\omega_k})^{-1/2}$ will make the new variables mutually independent and normally distributed with expected values equal to zero and variances equal to one. Finally, squaring these variables and summing them together will give a $\chi^2(2)$ distributed variable. Elliptical uncertainty regions centered around the nominal estimate $\hat{G}_N(e^{i\omega_k})$ will then be given by

$$\begin{pmatrix} \mathrm{Re}[G_0(e^{i\omega_k}) - \hat{G}_N(e^{i\omega_k})] \\ \mathrm{Im}[G_0(e^{i\omega_k}) - \hat{G}_N(e^{i\omega_k})] \end{pmatrix}^T \left(\frac{1}{N} \cdot P(e^{i\omega_k}) \right)^{-1} \times \begin{pmatrix} \mathrm{Re}[G_0(e^{i\omega_k}) - \hat{G}_N(e^{i\omega_k})] \\ \mathrm{Im}[G_0(e^{i\omega_k}) - \hat{G}_N(e^{i\omega_k})] \end{pmatrix} \leq C_\alpha, \tag{5.17}$$

where C_α is defined through $\mathrm{Prob}(X \leq C_\alpha) = \alpha$, $X \in \chi^2(2)$. $P(e^{i\omega_k})$ can be estimated using (2.20) and (2.78), from (2.25) or by using bootstrap methods.

5.3.2 Using Monte Carlo Simulations/Bootstrap Resampling

This section will describe how confidence regions can be constructed using a set of estimates of the system. The idea is the same as used in Davison and Hinkley (1997, Section 5.8). For notational ease we start by defining

$$\hat{G}_j^*(e^{i\omega}) = G(e^{i\omega}, \hat{\theta}_j^*). \tag{5.18}$$

Suppose that we are given B estimates, $\hat{G}_j^*(e^{i\omega})$, $j = 1, \ldots, B$, of the true system, $G_0(e^{i\omega})$. These estimates can either be obtained from Monte Carlo simulations or a bootstrap procedure. The estimates will be used to form approximate confidence regions for the frequency function in the Nyquist plot. The regions will naturally be constructed from the "closest to the mean" estimates and can be achieved in the following way. Denote the mean and the covariance of the estimates at frequency ω_k by

$$\bar{G}_B(e^{i\omega_k}) = \frac{1}{B} \sum_{j=1}^{B} \hat{G}_j^*(e^{i\omega_k}) \tag{5.19}$$

and

$$P_B(e^{i\omega_k}) = \frac{1}{B} \sum_{j=1}^{B} \begin{pmatrix} \mathrm{Re}[\hat{G}_j^*(e^{i\omega_k}) - \bar{G}_B(e^{i\omega_k})] \\ \mathrm{Im}[\hat{G}_j^*(e^{i\omega_k}) - \bar{G}_B(e^{i\omega_k})] \end{pmatrix} \begin{pmatrix} \mathrm{Re}[\hat{G}_j^*(e^{i\omega_k}) - \bar{G}_B(e^{i\omega_k})] \\ \mathrm{Im}[\hat{G}_j^*(e^{i\omega_k}) - \bar{G}_B(e^{i\omega_k})] \end{pmatrix}^T, \tag{5.20}$$

respectively. To order these estimates, we sort them in increasing distance from the bootstrap mean (5.19), measured in the norm given by inverse of the estimated covariance matrix (5.20). (This is the correct measure if the level curves of the probability density function are elliptical, which is the case for jointly normal distributed variables.) Next, construct the convex hull of the $B \cdot \alpha$ points, closest to the mean. This convex hull will represent an approximate confidence region of degree α for the frequency function at ω_k.

A drawback with bootstrap methods is that it is hard to construct reliable confidence regions with a confidence degree higher than 95%. This is due to the fact that we do not get an accurate description of the tails of the distribution of the residuals when using this technique. This is discussed in the literature and there are ways to partly overcome this problem with more advanced techniques. See Efron (1987).

5.3.3 Obtaining Simultaneous Confidence Degree

In estimating a system from input-output data it is of great importance to come up with a tight and reliable uncertainty description. This uncertainty is preferably displayed in the frequency domain since it provides useful information to, e.g., a control designer. The simplest (and probably most used) way to create such description is to construct confidence regions with a certain confidence degree at, e.g., a specific frequency for the frequency function. In order to make a confidence band out of this, the procedure is repeated at several frequencies and the regions are connected to produce a band for the entire frequency function. It should be noted that the different regions are dependent through the estimate of a parametric model and it is thus not possible to say much about the simultaneous confidence degree. As described in Section 3.2 the confidence regions constructed with this method (and the method in the previous section) will have lower simultaneous confidence degree than the individual regions. This should be clear from the discussion about Bonferroni's inequality (3.6). The lower bound given by this inequality is $1-d\cdot(1-\alpha)$, where d is the number of confidence regions and α is the confidence degree for one single region. In situations where a guaranteed simultaneous confidence degree is wanted, the parameters α and d should be chosen according to this. However, constructing too many intervals will force the designer to choose α very small and hence also make the bands (possibly) more conservative.

One exception to this basic procedure is given in Vuerinckx et al. (1998). Here the authors look at uncertainty descriptions for the poles and zeros of the system in a simultaneous manner. In the article there is no simulation study to evaluate the actual coverage probability of the intervals, but the approach looks interesting. In Politis et al. (1992) a method to construct simultaneous confidence bands for the spectra and cross-spectra of stationary weakly dependent time series is presented.

Their approach relies on the so called "block bootstrapping" method. Block bootstrapping essentially means that the we resample blocks of the time series instead of single measurements. This type of blocking method will not work in system identification related problems since the control signal contains deterministic components which do not fulfill the weakly dependent assumption. The simulations in the article shows quite good agreement with the theory.

The discussion above points out the problem with the methods described in Sections 5.3.1 and 5.3.2. These methods simply do not take the dependence between different frequency regions into account. Fortunately, this can be taken care of quite easily using bootstrap.

In Section 3.2 we pointed out that the confidence region in a high dimensional space should be rectangularly shaped if we want to project the region to low dimensional spaces. This can be quite difficult in a parametric setting, but is actually a relatively easy task in a bootstrap situation. Since we aim at constructing a rectangular box in a d-dimensional space, we must choose a norm measuring the distance between points in a suitable way. After appropriate scaling and translation of the estimates the l_∞-norm is the natural choice since it measures the longest orthogonal distance to the sides of the rectangle. In this way we do not take any cross-correlation between estimates into account (just as we want to). This can be compared to the previous section where the l_2-norm weighted with the inverse covariance matrix was used to construct approximate ellipsoids along the Nyquist curve. The use of the l_2-norm in this case is very natural since it is based on an underlying normality assumption of the estimates.

We will now formalize this idea. Assume that we estimate some d-dimensional parameter vector θ and aim at constructing d single confidence intervals with a simultaneous confidence degree α. B bootstrap estimates of θ are calculated and will be used to construct the intervals. Denote these estimates with

$$\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*. \tag{5.21}$$

Let the elements of $\hat{\theta}_k^*$ be denoted

$$\begin{pmatrix} \hat{\theta}_{1,k}^* & \cdots & \hat{\theta}_{d,k}^* \end{pmatrix}^T, \quad k = 1, \ldots, B, \tag{5.22}$$

and the estimated means and standard deviations of the elements of $\hat{\theta}$ by

$$\bar{\theta}_{j,\cdot}^* = \frac{1}{B} \sum_{k=1}^{B} \hat{\theta}_{j,k}^*, \quad j = 1, \ldots, d, \tag{5.23}$$

$$\hat{\sigma}_j^* = \sqrt{\frac{1}{B-1} \sum_{k=1}^{B} \left(\hat{\theta}_{j,k}^* - \bar{\theta}_{j,\cdot}^* \right)}, \quad j = 1, \ldots, d. \tag{5.24}$$

Normalize and translate the estimates of the different elements such that they all have unit variance and zero mean

$$\tilde{\theta}^*_{j,k} = \frac{\hat{\theta}^*_{j,k} - \bar{\theta}^*_{j,\cdot}}{\hat{\sigma}^*_j}. \tag{5.25}$$

This will give all elements in θ equal influence.

Define the distances from the origin to the bootstrap estimates as

$$q_k = \|\tilde{\theta}^*_k\|_\infty = \max(|\tilde{\theta}^*_{1,k}|, \ldots, |\tilde{\theta}^*_{d,k}|), \quad k = 1, \ldots, B. \tag{5.26}$$

Order the estimates in increasing distance to the mean

$$\hat{\theta}^*_{(1)}, \ldots, \hat{\theta}^*_{(B)}, \tag{5.27}$$

where the number between the parentheses denotes the order.

From the ordered set of estimates we pick the α-fraction closest to the mean. That is, the confidence region should be built from

$$\hat{\theta}^*_{(1)}, \ldots, \hat{\theta}^*_{(\lceil B\cdot\alpha \rceil)}, \tag{5.28}$$

where $\lceil\cdot\rceil$ denotes ceiling. The boundaries

$$b_1^{low}, b_1^{high}, \ldots, b_d^{low}, b_d^{high} \tag{5.29}$$

of the rectangular region will be decided from

$$b_j^{low} = \min(\hat{\theta}^*_{j,(1)}, \ldots, \hat{\theta}^*_{j,(\lceil B\cdot\alpha \rceil)}), \quad j = 1, \ldots, d, \tag{5.30}$$

$$b_j^{high} = \max(\hat{\theta}^*_{j,(1)}, \ldots, \hat{\theta}^*_{j,(\lceil B\cdot\alpha \rceil)}), \quad j = 1, \ldots, d. \tag{5.31}$$

These boundaries also form the single confidence intervals (that should be interpreted simultaneously) according to

$$I_{\theta_j} = (b_j^{low}, b_j^{high}), \quad j = 1, \ldots, d. \tag{5.32}$$

The choice of how many estimates that should be included in the confidence region actually gives the user some freedom of choice. With the choice $\lceil B \cdot \alpha \rceil$ we have $B \cdot (1-\alpha)$ of the estimates strictly outside the region and $B \cdot \alpha - 2 \cdot d$ estimates strictly inside the region. This is because we have one estimate on each boundary of the region. It should also be clear that changing the number of estimates that decides the boundaries of region to $\lceil B \cdot \alpha \rceil + 2 \cdot d$ gives $B \cdot \alpha$ estimates strictly inside the region and $B \cdot (1-\alpha) - 2 \cdot d$ estimates lying strictly outside the region. We recommend the choice of using $\lceil B \cdot \alpha \rceil + d$ as a standard choice (lying between the other two), but all choices in the interval $[\, \lceil B \cdot \alpha \rceil, \lceil B \cdot \alpha \rceil + 2 \cdot d \,]$ are valid. The choices could be preferably be labeled

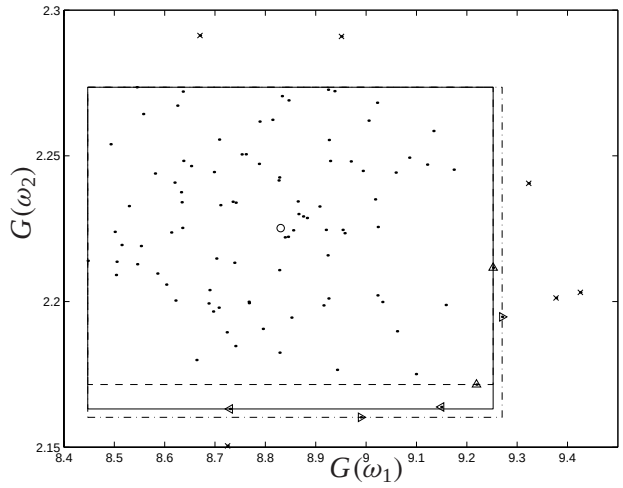


Figure 5.1 *Illustration of the freedom in choosing the rectangular simultaneous confidence region in the* 2*-dimensional case. The* $M = 100$ *dots illustrates different bootstrap estimates. The circle shows the mean of the estimates, the right rectangles (*$\triangleright$*) shows the 94th and 95th estimate lying furthest from the mean, the rectangles (*$\triangleleft$*) shows the 92th and 92th estimate lying furthest from the mean, and the rectangles (*$\triangle$*) shows the 90th and 91th estimate lying furthest from the mean. The dashed rectangle is obtained if we let* $10\% = 10$ *of the estimates lie strictly outside the region. The dash-dotted rectangle has on the other hand* $90\% = 90$ *of the estimates strictly inside the region. A compromise between these two is the solid rectangle which has* $90\% - 2 = 88$ *estimates lying inside and* $10\% - 2 = 8$ *estimates lying outside the region.*

- $\lceil B \cdot \alpha \rceil$: The aggressive choice (giving the smallest regions)
- $\lceil B \cdot \alpha \rceil + 2 \cdot d$: The conservative choice (giving the largest regions)
- $\lceil B \cdot \alpha \rceil + d$: The inbetween choice (a compromise between the two others)

The difference with these three choices will vanish as B tends to infinity, but can be significant when B is finite. When $d/B \ll 1$ the choice is more or less immaterial. One could however argue that the regions are reliable only when all choices gives approximately the same regions. The different choices are illustrated for the case $d = 2$ in Figure 5.1.

Finally we remark that, if the dimension of the parameter vector is high, we will need a substantial amount of reestimates of θ to accurately describe the probability

density function $f_{\hat{\theta}}$. This could mean that for d in the range of $30-50$ we would probably need B at least as high as 10000.

5.4 Examples

We continue this chapter with some examples that hopefully will illustrate some of the benefits and drawbacks of bootstrap in the area of system identification. One should once again note that we have concentrated the examples on constructing confidence regions in the Nyquist diagram, but the methods presented in this chapter apply equally well to constructing confidence regions for other statistics.

Example 5.1 The parameter distributions

In this first example we will try to illustrate how we can improve the estimate of the parameter vector distribution, $F_{\hat{\theta}_N}$, when there is a short data record and the residuals are taken from a skew distribution.

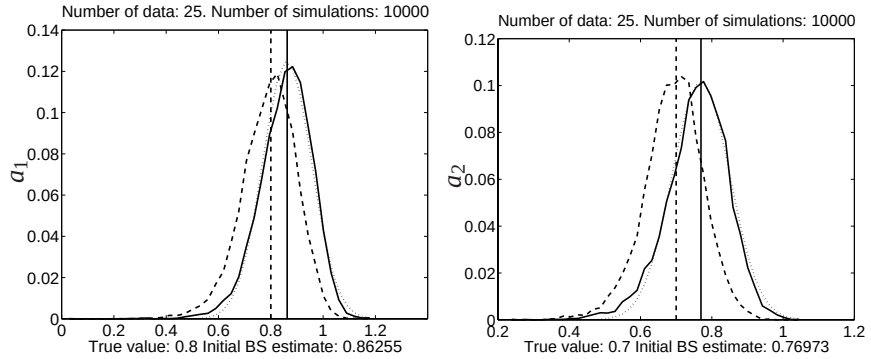
Let the true system be given by the the following ARX(2,2,1) model

$$\begin{aligned} y(t) &= \frac{B(q)}{A(q)}u(t) + \frac{1}{A(q)}e(t) \\ &= \frac{2q^{-1}+q^{-2}}{1+0.8q^{-1}+0.7q^{-2}}u(t) + \frac{1}{1+0.8q^{-1}+0.7q^{-2}}e(t), \end{aligned} \tag{5.33}$$

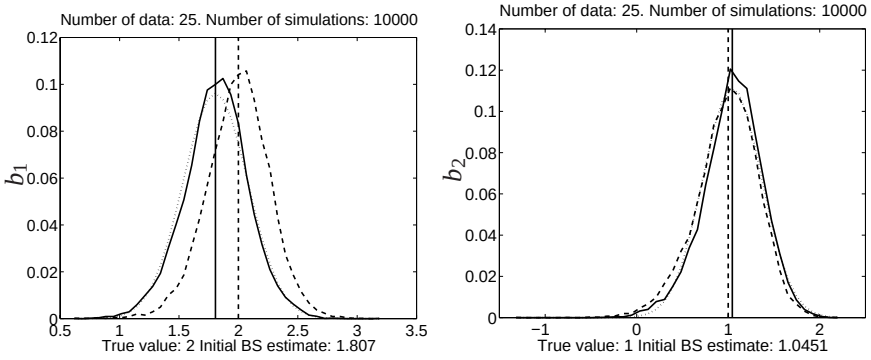
where $\{u(t)\}$ is a zero mean, Gaussian distributed white noise sequence with variance one, $\{e(t)\}$ is a zero mean white noise sequence with exponential PDF having the parameter λ equal to one. (The exponential probability density function is defined by (5.1).) This system was simulated with an input signal and noise sequence consisting of 25 samples and then estimated using the correct model structure.

Bootstrap was used to create $B = 10000$ resamples. Based on these we calculated approximate PDFs for the four parameters a_1, a_2, b_1 and b_2. Comparisons are made with Monte Carlo simulations and the approximate PDF we get from (2.17), see Figure 5.2. We also present the obtained estimates of the covariance matrix for the different methods. The estimates are denoted P_{MC}, P_{BS} and P_{ML} for Monte Carlo simulations, the bootstrap simulations and the maximum likelihood expression (2.17), respectively.

$$P_{MC} = \begin{pmatrix} 0.27 & 0.11 & 0.10 & 0.60 \\ 0.11 & 0.17 & -0.015 & 0.23 \\ 0.10 & -0.015 & 1.75 & 0.62 \\ 0.60 & 0.23 & 0.62 & 2.98 \end{pmatrix}$$



(a) PDF for a_1 (b) PDF for a_2



(c) PDF for b_1 (d) PDF for b_2

Figure 5.2 *PDFs for the parameter estimates in Example 5.1. Solid line – bootstrap, dashed line – Monte Carlo, dotted line – maximum likelihood see expression* (2.17).

$$P_{BS} = \begin{pmatrix} 0.25 & 0.11 & 0.12 & 0.49 \\ 0.11 & 0.19 & 0.0049 & 0.22 \\ 0.12 & 0.0049 & 1.83 & 0.63 \\ 0.49 & 0.22 & 0.63 & 2.63 \end{pmatrix}$$

$$P_{ML} = \begin{pmatrix} 0.22 & 0.089 & 0.053 & 0.44 \\ 0.089 & 0.16 & -0.029 & 0.17 \\ 0.053 & -0.029 & 1.86 & 0.55 \\ 0.44 & 0.17 & 0.55 & 2.60 \end{pmatrix}$$

By using a very skew PDF to generate the driving noise we have tried to construct an example where the asymptotic expression (2.17) will be less accurate. This should not affect the quality of the bootstrap estimates since this method works irrespective of the actual distribution of the noise, as long as it is white. The reason for using few data points is that we try to create a situation where the asymptotic expression (2.17) is not valid. It is worth noting that the initial estimate of G_0 and H_0 have a big influence on the bootstrap estimates. The estimation error we get as a consequence of the finite sample lengths in the initial estimate decides the centers of the PDFs for bootstrap (just as it does for the asymptotic expression). Bootstrap cannot adjust to this estimation error but it manages to find the shape of the true PDF quite well. This is most clearly seen in Figure 5.2(a), where the skewness of the Monte Carlo PDF is picked up by the bootstrap, but not (of course) by the maximum likelihood estimate. ❑

The two examples to come focus on the problem of estimating the variance error in case of undermodeling. This problem has, as already noted, not been discussed much in the literature. As mentioned in Section 2.1.3 the problem lies in the estimation of (2.19). To circumvent this problem we use the bootstrap according to Algorithm 5.2.

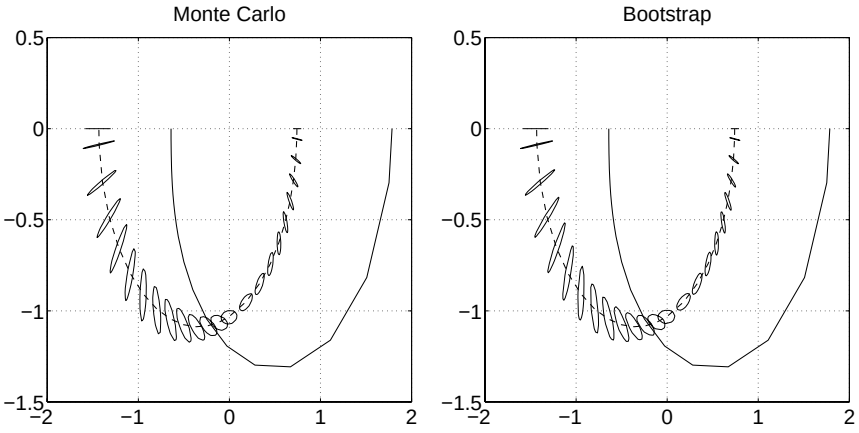
Example 5.2 Undermodeling 1

The example presented here is based on the same setup as used by Hjalmarsson and Ljung (1992). The true system is given by

$$y(t) = \frac{q^{-1}}{(1-0.2q^{-1})(1-0.3q^{-1})}u(t) + \frac{1}{(1-0.2q^{-1})(1-0.3q^{-1})(1+0.95q^{-1})}e(t), \tag{5.34}$$

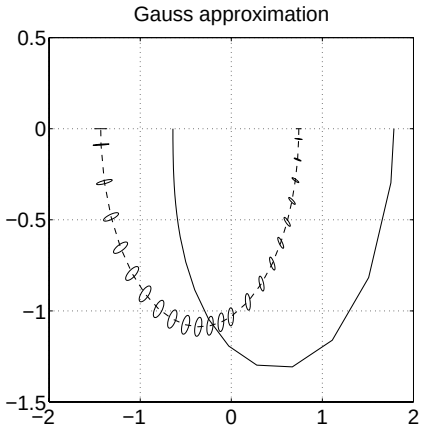
where $\{e(t)\}$ and $\{u(t)\}$ are zero mean, white Gaussian noise sequences with variances two and five, respectively. The noise has a a large amount of energy in the high frequency region, due to the pole at -0.95. To this system an ARX(1,1,1) model was fitted. This causes a large bias due to the high frequency weighting caused by noise, but the purpose of this example is only to show how the uncertainty in the estimated model can be found using bootstrap. (An estimate of the bias is however given by the model error model.)

The system was simulated with $N = 300,\ 1000,\ 5000,\ 10000$ data and $B = 2000$ resamples were made for each dataset to get the covariance estimates for the parameter vector. The model error was estimated using an ARX(15,15,0) structure



(a) Uncertainty regions using Monte Carlo simulations.

(b) Uncertainty regions using bootstrap.



(c) Uncertainty regions using (2.20) and (2.78).

Figure 5.3 *Nyquist curve with uncertainty regions using different methods described in Example 5.2. Solid line – true system.*

N	Boot-strap		Monte Carlo		ML Estimate		Hjalmarsson $\lambda = 0.99$	
300	4.3	0.2	4.9	-0.1	0.6	0.0	2.9	-1.2
	0.2	0.8	-0.1	0.8	0.0	1.6	-1.2	1.5
1000	4.9	-0.4	5.3	-0.2	0.6	0.0	2.0	-0.7
	-0.4	1.2	-0.2	1.2	0.0	1.7	-0.7	2.1
5000	6.0	-0.2	5.4	-0.3	0.6	0.0	4.0	0.2
	-0.2	1.0	-0.3	1.1	0.0	1.8	0.2	1.6
10000	5.8	-0.2	5.5	-0.1	0.6	0.0	4.6	0.3
	-0.2	0.9	-0.1	1.0	0.0	1.8	0.3	1.7

Table 5.1 *Estimates of P_θ. The bootstrap and Monte Carlo simulations are based on 2000 resamples and realizations, respectively. Note that covariance estimates in the last column are taken from (Hjalmarsson and Ljung, 1992). See the cited article for further details.*

(the same model was used in all four cases in order to avoid misunderstandings). The resulting estimates of the covariance matrices for the parameters, P_θ, are presented in Table 5.1. It summarizes the results for bootstrap, Monte Carlo simulations, and the conventional estimate (2.20). The bootstrap and the conventional estimate are based on the same realization of the noise sequence; the Monte Carlo estimates were obtained using the same input sequence as the two former, but 2000 independent runs of sequences were used as driving noise. We also give the results achieved in Hjalmarsson and Ljung (1992) in the last column. In this column one should note that we have not done any new experiments, we have just picked the values achieved in the referenced article. (We should also mention that the Monte Carlo simulations in Hjalmarsson and Ljung (1992) are not performed in the same way as here, confer the cited article for details.)

Although the comparison with Hjalmarsson and Ljung (1992) is not 100% fair, we will draw some conclusions. First the Monte Carlo estimates should be seen as being very close to the true theoretical values. The bootstrap method produces estimates that are closer to the Monte Carlo estimates than the estimates obtained in Hjalmarsson and Ljung (1992). As expected, the quality of the covariance estimates improves as N (and B) increases.

We also calculated uncertainty regions for the Nyquist curve at 24 frequencies for the dataset with $N = 5000$. This was done with Monte Carlo simulations (Figure 5.3(a)), bootstrap resampling (Figure 5.3(b)), and the conventional estimate combined with Gauss' approximation formula (Figure 5.3(c)). The methods used

to construct the uncertainty regions are described in Section 5.3.

It is evident that bootstrap works well in this case. We see that it is almost impossible to separate the results of Monte Carlo simulations from the ones obtained using bootstrap. The results could probably be even better if more effort was put on finding the best order of the model error model in all four different cases. This is a clear indication of that the suggested bootstrap procedure gives reliable variance error estimates. ❑

Example 5.3 Undermodeling 2

This example is taken from Box and Jenkins (1976, pp. 532-533). Comparisons are made with the method used in Hjalmarsson and Ljung (1992). The data set consists of 296 measurements of input gas rate and output CO_2 concentration of gas furnace. The model used is an ARX(3,1,3) (the same one as in Hjalmarsson and Ljung (1992)). Since we do not know the true system, we can not find the "correct" covariance matrix in this case by means of Monte Carlo simulations. Therefore we compare the results from bootstrap estimation with the results obtained with the method of Hjalmarsson and Ljung (1992). The bootstrap estimated covariance matrix is based on 2000 resamples and is denoted P_{BS}. We denote the covariance matrix estimated in Hjalmarsson and Ljung (1992) by P_{HL}. The results are:

$$P_{HL} = \begin{pmatrix} 10.3 & -9.6 & 1.8 & -8.3 \\ -9.6 & 9.9 & -2.4 & 7.0 \\ 1.8 & -2.4 & 0.9 & -0.9 \\ -8.3 & 7.0 & -0.9 & 7.4 \end{pmatrix}$$

$$P_{BS} = \begin{pmatrix} 1.5 & -2.3 & 0.97 & -0.63 \\ -2.3 & 3.6 & -1.6 & 0.83 \\ 0.97 & -1.6 & 0.80 & -0.31 \\ -0.63 & 0.83 & -0.31 & 0.44 \end{pmatrix}$$

The difference is significant, but it is impossible to say which one is better. However, one can argue that since the method in Hjalmarsson and Ljung (1992) seems to need a significant amount of data points to converge to the correct covariance matrix, their method will probably not be very accurate in this short data example. ❑

Example 5.4 Simultaneous confidence degree

In this example we will make a study of the different uncertainty bounding techniques described in this chapter and in Chapter 2. This includes both the method in Section 5.3.2 and the method in Section 5.3.3. As a reference we will calculate

uncertainty regions with the help of Monte Carlo simulations. Totally we have four different approaches that we will compare:

1. The maybe simplest classical expression, which is given by

$$\frac{1}{N}P(\omega) \approx \frac{n}{N}\frac{1}{2}\Phi_v(\omega)\begin{bmatrix} 1/\Phi_u(\omega) & 0 \\ 0 & 1/\Phi_u(\omega) \end{bmatrix} \text{ as } N, n, \frac{N}{n} \to \infty$$

   Against the simplicity of this expression stands the fact that it is asymptotic in the model order.

2. Probably the most used expressions, i.e., the combination of

$$P(\omega) \approx \begin{bmatrix} \text{Re}\, \hat{G}'_\theta(e^{i\omega}, \hat{\theta}_N) \\ \text{Im}\, \hat{G}'_\theta(e^{i\omega}, \hat{\theta}_N) \end{bmatrix} P_\theta \begin{bmatrix} \text{Re}\, \hat{G}'_\theta(e^{i\omega}, \hat{\theta}_N) \\ \text{Im}\, \hat{G}'_\theta(e^{i\omega}, \hat{\theta}_N) \end{bmatrix}^T,$$

   where P_θ can be estimated from (2.18-2.19).

3. The bootstrap procedure presented in Section 5.3.2. This will be called the *standard bootstrap procedure* in the rest of this example.

4. The bootstrap procedure presented in Section 5.3.3 giving simultaneous confidence degree. This will be called the *simultaneous bootstrap*.

Comparing different methods is difficult, since they are often based on different assumptions. The aim of this example is thus not to classify different methods in their ability to solve the specific problem they are exposed to. It will more serve as guide to where different methods works well, and where they do not. We do however expect that the methods that do not construct simultaneous confidence regions will produce similar results (expect method number 1, since it is asymptotic in the model order).

To illustrate these approaches we will use the following system:

$$\begin{aligned} A(q)y(t) &= B(q)u(t) + e(t) \\ A(q) &= 1 - 2.5q^{-1} + 3.3q^{-2} - 2.5q^{-3} + 1.2q^{-4} - 0.3q^{-5}, \\ B(q) &= 0.21q^{-1} + 0.35q^{-2} - 0.12q^{-3} - 0.11q^{-4} + 0.23q^{-5}, \end{aligned} \tag{5.35}$$

where $u(t)$ is a pseudo random binary signal with amplitude one and $e(t)$ is an i.i.d. Gaussian processes with zero mean and variances 0.04. A bode plot of the system is depicted in Figure 5.4. This system was simulated with $N = 1000$ data points and an ARX(5,5,1)-model was fitted to the data. Since the correct model structure and order is used we have no problem with undermodeling.

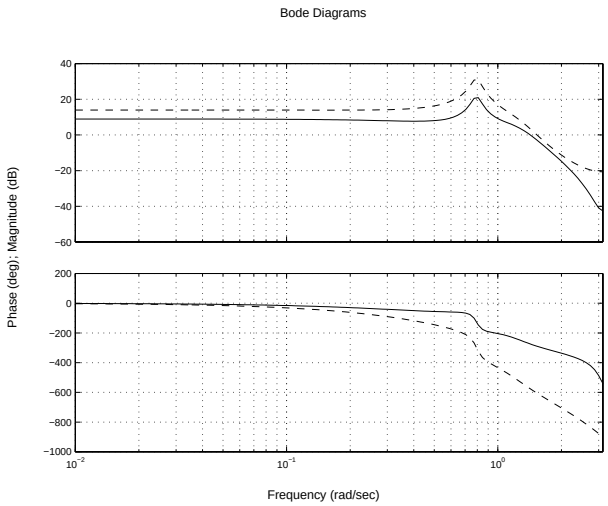


Figure 5.4 *Bode diagram of the system in Example* (5.4)*. Solid line – the system, dashed line – the noise system.*

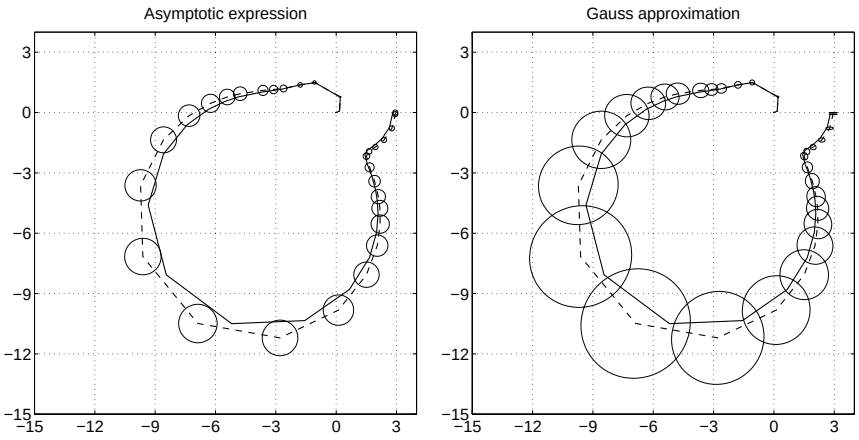
The evaluation of the results are displayed along the Nyquist curve using confidence regions calculated along a grid defined by the following 32 frequencies (in MATLAB notation):

$$\begin{aligned}\mathcal{W}_1 =&\{0.00001,\ 0.01,\ (0.1:0.1:0.6),\ 0.65,\ (0.68:0.015:0.9),\\ &0.93,\ 0.96,\ 1,\ 1.1,\ 1.2,\ 1.5,\ 2,\ 3.1\}\end{aligned}$$

(All frequencies in rad/s.)

All four different approaches are calculated and displayed in Figure 5.5 and in Figure 5.6. The construction of the bootstrap and Monte Carlo based confidence regions are based on $B = 3000$ resamples/samples. Some conclusions that can be drawn from these figures are:

- The simple expression (2.25), which is asymptotic in both the model order and N, clearly underestimates the size of the confidence regions at most frequencies (Figure 5.5(a)). This is probably because of the relatively low model order used in this example, which makes the used expression invalid.
- Gauss' approximation formula and the standard bootstrap procedure (Section 5.3.2) give results that resembles closely with the results obtained using Monte Carlo simulations. See Figures 5.5(b), 5.6(a), and 5.6(c), respectively. Looking more carefully at these figures we also see that bootstrap gives a



(a) Uncertainty regions using the asymptotic expression (2.25).

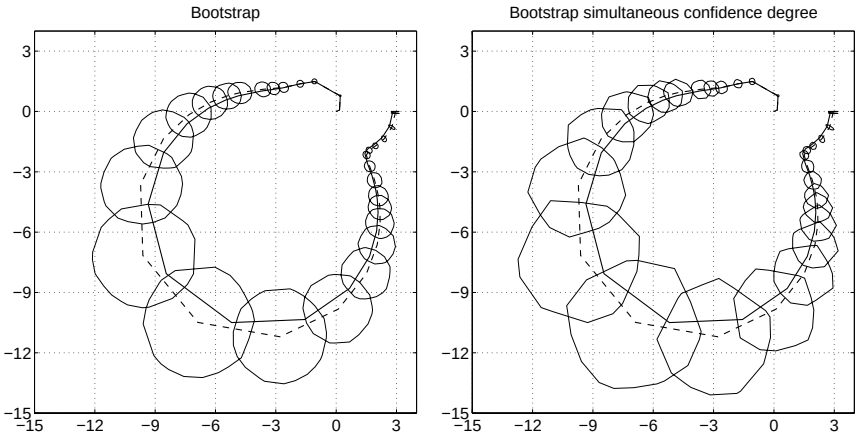
(b) Uncertainty regions using Gauss' approximation formula and the estimate (2.22).

Figure 5.5 *Nyquist curve with 95 % simultaneous confidence degree uncertainty regions using the classical expressions in Chapter 2. Solid line – true system, dashed line – estimated system.*

more accurate orientation of the regions in the low frequency range than the traditional use of Gauss' approximation formula.

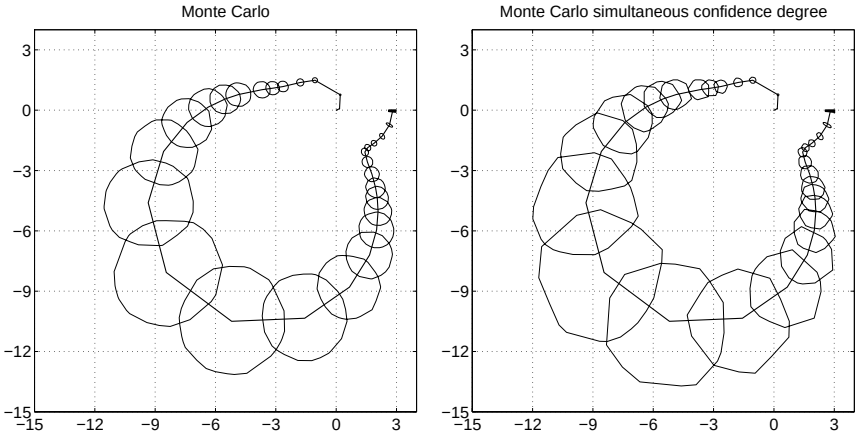
- The confidence regions built utilizing the simultaneous bootstrap are considerably larger than the others. See Figure 5.6(b) for bootstrap and Figure 5.6(d) for the Monte Carlo based results. It also looks like bootstrap manages to give a good description of the uncertainty regions even in this case, i.e., the regions have more or less the same shape size as what was achieved using Monte Carlo simulations.
- One obviously needs a substantial amount of estimates to be able to achieve "smooth" simultaneous confidence regions. This has to do with that distances are measured in a 32 dimensional space (according to the number of constructed regions). Having $B = 3000$ resamples corresponds to approximately 100 samples in each dimensions, which does not give a very precise description of the underlying distribution.

The figures shown in this example illustrate the fact that bootstrap achieves confidence regions which are very similar to the ones obtained using Monte Carlo sim-



(a) Uncertainty regions using bootstrap and the method in Section 5.3.2.

(b) Uncertainty regions using bootstrap and the method in Section 5.3.3.



(c) Uncertainty regions using Monte Carlo simulations and the method in Section 5.3.2.

(d) Uncertainty regions using Monte Carlo simulations and the method in Section 5.3.3.

Figure 5.6 *Nyquist curve with 95 % simultaneous confidence degree uncertainty regions using different methods described in Sections 5.3.2 and 5.3.3. Solid line – true system, dashed line – estimated system (where it is relevant).*

ulations. These figures do not however say anything about the confidence degree actually achieved. Therefore, we evaluate the actual quality of the simultaneous bootstrap confidence regions. This was accomplished by constructing confidence bands for the amplitude curve and evaluating how many of these bands that actually covered the true system at *every* frequency. This was done by generating 500 realizations from the system (5.35). We evaluated the results on the following frequency grid

$$\mathcal{W}_2 = \{0.00001,\ 0.01,\ (0.1 : 0.1 : 0.6),\ (0.65 : 0.05 : 0.9),\ 1,\ 1.2,\ 1.5,\ 2,\ 3.1\}.$$

We used $N = 300$ and $N = 1000$ data in each realization and constructed the confidence bands from each one of them using $B = 1000$ and $B = 3000$ resamples, respectively. The confidence degrees we wanted to achieve were $\alpha = 0.85$, 0.9, and 0.95. The obtained confidence degrees are depicted in Table 5.2. In this table we also included a comparison of three different choices of the number of samples that was included in the construction of the confidence band. Here we compared the aggressive choice, the inbetween choice, and the conservative choice. It is clear that if cannot use a high B (due to time limitations for example) we should use the conservative choice. If $d/B \ll 0.01$ the choice is more or less immaterial. The table shows that the simultaneous bootstrap seems to work really well in most cases. It is however a bit surprising that the method produces slightly worse results when the data length increases (on this example). To this we have no explanation. We also included an evaluation of the actual confidence degree obtained when choosing the confidence degrees to meet the guaranteed lower bound given by the Bonferroni inequality. The table shows that this bound is rather conservative in this case.

❑

5.5 Conclusions

We have looked at some aspects of where bootstrap could be used in system identification related problems. The proposed algorithms work fine in the demonstrated examples. Even situations where asymptotic results are valid we can find some improvements using bootstrap. This was shown in Example 5.4.

Bootstrap also make it possible to construct confidence regions with simultaneous confidence degree, which is far more difficult task using traditional methods.

In Example 5.2 it is shown that bootstrap provides a nice solution to the problem of estimating the variance error of the estimates in case of undermodeling. This is also the main contribution of this chapter.

$N = 300$	$\alpha = 0.85$	$\alpha = 0.9$	$\alpha = 0.95$
$B = 1000$	0.834	0.912	0.942
$B = 3000$	0.842	0.904	0.944
Bonferroni	0.952	0.968	0.98
$N = 1000$	$\alpha = 0.85$	$\alpha = 0.9$	$\alpha = 0.95$
$B = 1000$	0.810	0.886	0.942
$B = 3000$	0.832	0.884	0.946
Bonferroni	0.948	0.958	0.974
$B = 3000$	$\lceil 0.9B \rceil$	$\lceil 0.9B \rceil + 19$	$\lceil 0.9B \rceil + 2 \cdot 19$
$N = 300$	0.896	0.904	0.916
$N = 1000$	0.876	0.886	0.900

Table 5.2 *Obtained simultaneous confidence degree for the amplitude estimate of* (5.35)*. The method used to construct these bands was bootstrap as described in Section 5.3.3. The results are based on 500 realizations of* (5.35) *using N data. B is the number of bootstrap resamples and α is the confidence degree that was wished for. (Note that $\alpha = 0.9$ in the last two rows.) The rows named Bonferroni is the actual confidence level achieved when the intervals are constructed using asymptotic normality and the lengths adjusted to meet the guaranteed simultaneous confidence level.*

As a conclusion one must say that the strength of bootstrap lies in its simplicity and wide applicability. Most computations are fast and extremely easy to implement, but its repetitive nature could make the total computational time long.

6

Model Reduction and Variance Reduction

Model reduction is all about compressing a given representation of a system. The most extreme example of this is the actual identification phase, where the "model" consisting of Z^N is mapped into a nth order parameterized one. In the standard setting described in Chapter 2 this corresponds to finding the best L_2 approximation of data (given a model class). Irrespectively how the reduction phase is performed it makes it possible to keep track of the bias errors the reduction step gives rise to. This will of course help us in the control design phase, so that stability and performance measures can be calculated. There has, however, been little discussion on how the variance of the high order estimated model maps over to the low order one. Since the variance errors strongly affects the use and interpretation of the reduced model they are in many cases at least as important as the bias errors.

We will in this chapter discuss exactly this problem, i.e., how to compute the variance of the reduced model. General formulas describing the covariance of the low order model will presented. When the reduced models are of FIR and of OE structures we also explicitly compute the covariance matrix. The result is that when the true system lies in the model class of the low order model the method is efficient, i.e., the covariance matrix meets the Cramér-Rao bound.

6.1 Model Reduction

To estimate a low order model of a system, several possibilities exist. The most obvious one is to directly estimate a lower order model from data:

$$\hat{\theta}_N = \arg\min_{\theta} \frac{1}{N} \sum_{t=1}^{N} |y(t) - G(q,\theta)u(t)|^2. \tag{6.1}$$

As known from, e.g., Ljung (1999b), the prediction/output error estimate automatically gives models that are L_2-approximations of the true system in a frequency-weighted norm, determined by the input spectrum and noise model:

$$\hat{\theta}_N \to \theta^* = \arg\min_{\theta} \int_{-\pi}^{\pi} |G_0(e^{i\omega}) - G(e^{i\omega},\theta)|^2 \Phi_u(\omega) d\omega, \tag{6.2}$$

where Φ_u is the input spectrum and G_0 is the true system. A second possibility is to estimate a higher order model, which is then subjected to model reduction to the desired order. See, e.g., Wahlberg (1989). For the model reduction step, a variety of methods could be applied, like truncating balanced state-space realizations, or applying L_2-norm reduction. The latter method means that the low order model, parameterized by η is determined as

$$\hat{\eta} = \arg\min_{\theta} \int_{-\pi}^{\pi} |\hat{G}_h(e^{i\omega}) - G(e^{i\omega},\eta)|^2 W(\omega) d\omega. \tag{6.3}$$

Here, $\hat{G}_h(q)$ is the high order (estimated) model, and W is a weighting function.

An important question is whether this reduction step also will imply a reduction of variance, i.e., if the variance of $G(e^{i\omega}, \hat{\eta})$ (viewed as random variable through its dependence of the estimate $\hat{G}_h(e^{i\omega})$) is lower than that of $\hat{G}_h(e^{i\omega})$. A second question is how this variance compares with the one obtained by the direct identification method (6.1).

The somewhat surprising answer is that (6.3) may in some cases give a lower variance than (6.1). Before treating general cases, let us first consider a simple, but still illustrating example.

Example 6.1

Consider the true system

$$y(t) = u(t-1) + 0.5u(t-2) + e(t), \tag{6.4}$$

where the input u is white noise with variance μ, and e is white noise with variance λ. We compare two ways of finding a first order model of this system. First, estimate b in the FIR model

$$\hat{y}(t|b) = bu(t-1).$$

This gives the estimate (using least squares) $\hat{b}_N$, with

$$\hat{b}_N \to \mathrm{E}\,\hat{b}_N = 1, \quad \text{as} \quad N \to \infty.$$

Note here that the expectation is taken over both u and e. This will be used throughout this chapter. The variance of $\hat{b}_N$ is computed by

$$\mathrm{E}(\hat{b}_N - 1)^2 = \mathrm{E}\left(\frac{\sum_{t=1}^{N} u(t-1)(0.5u(t-2)+e(t))}{\sum_{t=1}^{N} u^2(t-1)}\right)^2 \approx \frac{\lambda + 0.25\mu}{N \cdot \mu}.$$

The second method is to estimate a higher order model (in this case second order)

$$\hat{y}(t|b_1, b_2) = b_1 u(t-1) + b_2 u(t-1).$$

This gives the estimated transfer function

$$\hat{G}_h(q) = \hat{b}_1 q^{-1} + \hat{b}_2 q^{-2},$$

with $\hat{b}_i$ tending to their true values, and each having a variance of $\lambda/(N\mu)$. Now, subjecting $\hat{G}_h$ to the L_2 model reduction (6.3) with $W(\omega) \equiv 1$ gives the reduced model

$$\hat{G}_\ell(q) = \hat{b}_1 q^{-1}.$$

The variance of the directly estimated first order model is

$$\mathrm{Var}\,\hat{b}_1 = \frac{\lambda + 0.25\mu}{N \cdot \mu},$$

while the L_2-reduced model has

$$\mathrm{Var}\,\hat{b}_1 = \frac{\lambda}{N \cdot \mu},$$

i.e., it is strictly smaller. ❑

In this example it was strictly better to estimate the low order model, both in terms of variance and mean square error, by reducing a higher order model than to estimate it directly from data. This somewhat unexpected result can clearly only happen if the low order model structure does not contain the true system. The prediction error methods are in these cases (assume that e is normal) efficient, i.e., their variances meet the Cramèr-Rao bound if the model structure contains the true system. In those cases no other estimation method can beat the direct estimation method.

6.2 Other Approaches

Before going into the actual calculation we discuss some related approaches. Three of the few contributions that take into account that the high order model is obtained through an identification experiment when performing model reduction are Wahlberg (1987), Söderström et al. (1991), and Zhou and Backx (1993). We will shortly describe the first two.

In Söderström et al. (1991) the authors look at nested model structures. Especially they look for structures that can be embedded in larger structures which are easy to estimate, such as ARX structures. After estimating the high order structure they reduce the estimate to the low order structure in a weighted least squares sense. The method is called an *indirect prediction error method*. We illustrate the idea using the generalized least squares structure.

Assume that the low order structure is of ARMAX type (see Section 2.1.4), where the polynomials $A(q)$, $B(q)$, and $C(q)$ are of orders n_{a_1}, n_{b_1} and n_{c_1} respectively. The structure is then parameterized by

$$\eta = \begin{pmatrix} a_1 & \dots & a_{n_{a_1}} & b_1 \dots & b_{n_{b_1}} & c_1 \dots & c_{n_{c_1}} \end{pmatrix}.$$

Now we rewrite this structure to a high order ARX structure by multiplying with $C(q)$, i.e.,

$$\begin{aligned} A(q,\eta)C(q,\eta)y(t) &= B(q,\eta)C(q,\eta)u(t) + e(t) \\ &\Leftrightarrow \\ \tilde{A}(q,\theta)y(t) &= \tilde{B}(q,\theta)u(t) + e(t), \end{aligned}$$

where

$$\theta = \begin{pmatrix} \tilde{a}_1 & \dots & \tilde{a}_{n_{a_2}} & \tilde{b}_1 \dots & \tilde{b}_{n_{b_2}} \end{pmatrix}.$$

This means that there is a non linear mapping between η and θ, i.e., $\eta = F(\theta)$. Now θ can be estimates using standard least squares and η is the found by minimizing

$$\hat{\eta} = \arg\min_{\eta}(F(\eta) - \hat{\theta}_N)^T \hat{P}_\theta^{-1}(F(\theta) - \hat{\theta}_N),$$

where $\hat{P}_\theta$ is an estimate of the covariance of θ. It is shown that the statistical properties of this indirect methods is the same as for standard PEM.

Wahlberg (1987) uses an approach similar to the one in Söderström et al. (1991). First a nth order FIR model parameterized by θ is estimated and is then reduced to a lower order model $G(q, \eta)$ subject to

$$\hat{\eta} = \arg\min_{\eta}(F(\eta) - \hat{\theta}_N)^T R_N(F(\eta) - \hat{\theta}_N).$$

Here

$$F(\eta) = R_N^{-1} \sum_{t=1}^{N} G(q, \eta)u(t)\varphi(t)$$

and

$$\varphi(t) = \begin{pmatrix} u(t-1) & \dots & u(t-n) \end{pmatrix}^T$$
$$R_N = \sum_{t=1}^{N} \varphi(t)\varphi^T(t).$$

It is shown that the estimate of η is asymptotically efficient, i.e., its covariance matrix meets the Cramér-Rao bound as the FIR order, n, tends to infinity.

Note that both of these approaches coincide with L_2 model reduction if η is a linear function of θ. This is the case if η parameterize by a FIR model.

6.3 The Basic Tools

To translate the variance of one estimate $\hat{\theta}$ to another $\hat{\eta} = f(\hat{\theta})$ we use Gauss' approximation formula (2.77). To use this result to compute the variance of an L_2-reduced model, we need an (asymptotic) expression for how it depends on the higher order model. For this we return to (6.3) with more specific notation. Let the high order model be

$$G(q, \hat{\theta}), \quad \text{with } \operatorname{Cov}\hat{\theta} = P_\theta. \tag{6.5}$$

Let η parameterize a lower order model $G(q, \eta)$ and define

$$\hat{\eta}(\hat{\theta}) = \arg\min_{\eta} V(\eta, \hat{\theta}) \tag{6.6}$$

for some function V, that depends on the lower order model η and the high order, estimated, model $\hat{\theta}$. For L_2-reduction we use

$$V(\eta, \hat{\theta}) = \int_{-\pi}^{\pi} |G(e^{i\omega}, \eta) - G(e^{i\omega}, \hat{\theta})|^2 W(\omega) d\omega, \tag{6.7}$$

but the form of V is immaterial for the moment. We will assume it to be differentiable, though.

Now, since $\hat{\eta}$ minimizes $V(\eta, \hat{\theta})$, we have

$$V'_{\eta}(\hat{\eta}(\hat{\theta}), \hat{\theta}) = 0, \tag{6.8}$$

where V'_{η} denotes the partial derivative of V with respect to its first argument. Now, (6.8) by definition holds for all $\hat{\theta}$, so taking the total derivative with respect to $\hat{\theta}$ gives

$$0 = \frac{d}{d\theta} V'_{\eta}(\hat{\eta}(\hat{\theta}), \hat{\theta}) = V''_{\eta\eta} \frac{d}{d\hat{\theta}} \hat{\eta}(\hat{\theta}) + V''_{\eta\theta}$$

or

$$\frac{d}{d\hat{\theta}} \hat{\eta}(\hat{\theta}) = -[V''_{\eta\eta}]^{-1} V''_{\eta\hat{\theta}}. \tag{6.9}$$

This expression for the derivative, and Gauss' approximation formula (2.77), now give the translation of the variance of $\hat{\theta}$ to that of $\hat{\eta}$:

$$\begin{aligned} \operatorname{Cov} \hat{\eta} =& P_{\eta} \\ =& \left[[V''_{\eta\eta}(\eta^*, \theta^*)]^{-1} V''_{\eta\theta}(\eta^*, \theta^*) \right] P_{\theta} \left[[V''_{\eta\eta}(\eta^*, \theta^*)]^{-1} V''_{\eta\theta}(\eta^*, \theta^*) \right]^T, \end{aligned} \tag{6.10}$$

where

$$\theta^* = \lim_{N \to \infty} \hat{\theta}_N \tag{6.11}$$

and

$$\eta^* = \eta(\theta^*). \tag{6.12}$$

This gives us a general expression for investigating variance reduction for any reduction technique that can be written as (6.6). Especially it holds for L_2-reduced estimates (6.7).

6.4 The FIR case

In this section we will look at systems of FIR structure. We show the perhaps surprising result that estimating a high order model followed by L_2 model reduction *never* gives higher variance than directly estimating the low order model.

Suppose that our data is generated by FIR system with $d = d_1 + d_2$ parameters, i.e.,

$$\begin{aligned} y(t) =& \sum_{k=1}^{d_1} b_k u(t-k) + \sum_{k=d_1+1}^{d} b_k u(t-k) + e(t) \\ =& \eta_0^T \varphi_1(t) + \xi_0^T \varphi_2(t) + e(t) = \theta_0^T \varphi(t) + e(t), \end{aligned} \tag{6.13}$$

where e is white noise with variance λ, and u is a stationary stochastic process, independent of e, with spectrum $\Phi_u(\omega)$. The definitions of η, ξ, θ, and $\varphi(t)$ should be immediate from (6.13):

$$\eta_0 = \begin{pmatrix} b_1 \\ \vdots \\ b_{d_1} \end{pmatrix}, \quad \varphi_1(t) = \begin{pmatrix} u(t-1) \\ \vdots \\ u(t-d_1) \end{pmatrix},$$

etc. Let us also introduce the notation

$$\begin{aligned} R_{11} &= \mathrm{E}\, \varphi_1(t)\varphi_1^T(t), \quad R_{12} = \mathrm{E}\, \varphi_1(t)\varphi_2^T(t) = R_{21}^T \\ R_{22} &= \mathrm{E}\, \varphi_2(t)\varphi_2^T(t). \end{aligned} \tag{6.14}$$

The true frequency function can thus be written

$$G_0(e^{i\omega}) = \theta_0^T \begin{pmatrix} e^{-i\omega} \\ \vdots \\ e^{-di\omega} \end{pmatrix}. \tag{6.15}$$

We now seek the best L_2 approximation (in the frequency weighting norm Φ_u) of this system of order d_1:

$$\begin{aligned} \eta^* &= \arg\min_\eta \int_{-\pi}^{\pi} |G_0(e^{i\omega}) - G(e^{i\omega}, \eta)|^2 \Phi_u(\omega) d\omega \\ &= \arg\min_\eta \mathrm{E}\, |\theta_0 \varphi(t) - \eta^T \varphi_1(t)|^2, \end{aligned} \tag{6.16}$$

where the second step is Parseval's identity. Simple calculations show that the solution is

$$\eta^* = \eta_0 + R_{11}^{-1} R_{12} \xi_0. \tag{6.17}$$

Now, the least squares estimate $\hat{\eta}_N$ of order d_1 is

$$\begin{aligned}\hat{\eta}_N =& \left[\sum \varphi_1(t)\varphi_1^T(t)\right]^{-1} \sum \varphi_1(t)y(t)\\ =&\eta_0 + \left[\sum \varphi_1(t)\varphi_1^T(t)\right]^{-1} \sum \varphi_1(t)\varphi_2^T(t)\xi_0\\ &+ \left[\sum \varphi_1(t)\varphi_1^T(t)\right]^{-1} \sum \varphi_1(t)e(t),\end{aligned} \tag{6.18}$$

where the second step follows from (6.13). This gives that

$$\mathrm{E}\,\hat{\eta}_N \approx \eta^*. \tag{6.19}$$

The approximation involved concerns the indicated inverse. When N is large the law of large numbers can be applied to give the result. (A technical comment: In the definition of the estimate, one may have to truncate for close-to-singular matrices. See Appendix 9.B in Ljung (1999b) for such technicalities.) Moreover

$$\begin{aligned}\mathrm{Cov}\,\hat{\eta}_N =& \mathrm{E}(\hat{\eta}_N - \eta^*)(\hat{\eta}_N - \eta^*)^T\\ \approx& \frac{\lambda}{N} R_{11}^{-1} + \mathrm{E}\, H_N \xi_0 \xi_0^T H_N^T\end{aligned} \tag{6.20}$$

where

$$H_N = \left[\sum \varphi_1(t)\varphi_1^T(t)\right]^{-1} \left[\sum \varphi_1(t)\varphi_2^T(t)\right] - [R_{11}]^{-1} R_{12}. \tag{6.21}$$

Let us now turn to the model reduction case. We first estimate the full system of order d. That gives the estimate $\hat{\theta}_N$ with

$$\mathrm{E}\,\hat{\theta}_N = \theta_0 \tag{6.22}$$

and

$$\mathrm{Var}\,\hat{\theta}_N = P_{\hat{\theta}} = \frac{\lambda}{N}\left[\mathrm{E}\,\varphi(t)\varphi^T(t)\right]^{-1} = \frac{\lambda}{N}\begin{pmatrix} R_{11} & R_{12}\\ R_{21} & R_{22}\end{pmatrix}^{-1}, \tag{6.23}$$

with obvious partitioning according to (6.14). We insert this high order estimate into (6.7) with $W(\omega) = \Phi_u(\omega)$ and perform the model reduction (6.6).

Note that, by Parseval's relation (6.7) can also be written

$$V(\eta, \hat{\theta}) = \mathrm{E}(\eta^T\varphi_1(t) - \hat{\theta}^T\varphi(t))^2, \tag{6.24}$$

with $\varphi(t)$ constructed from u as in (6.13), where u has the spectrum $W(\omega) = \Phi_u(\omega)$. In the notation of (6.8) we have

$$\begin{aligned} V''_{\eta\eta} &= \mathrm{E}\,\varphi_1(t)\varphi_1^T(t) = R_{11} \\ V''_{\eta\hat{\theta}} &= \mathrm{E}\,\varphi_1(t)\varphi^T(t) = \mathrm{E}\,\varphi_1(t)\begin{pmatrix}\varphi_1^T(t) & \varphi_2^T(t)\end{pmatrix} \\ &= \begin{pmatrix}R_{11} & R_{12}\end{pmatrix} \end{aligned} \tag{6.25}$$

From (6.10) and (6.23) we now find that

$$\mathrm{Cov}\,\hat{\eta} = R_{11}^{-1}\begin{pmatrix}R_{11} & R_{12}\end{pmatrix} \cdot \frac{\lambda}{N}\begin{pmatrix}R_{11} & R_{12}\\ R_{21} & R_{22}\end{pmatrix}^{-1}\begin{pmatrix}R_{11}\\ R_{21}\end{pmatrix} R_{11}^{-1} = \frac{\lambda}{N}R_{11}^{-1}, \tag{6.26}$$

where the last step simply follows from the definition of an inverse matrix.

Comparing with (6.20) we see that *this variance is strictly smaller than that obtained by direct identification*, provided $\xi_0 \neq 0$, that is, the true system is of higher order than d_1. However, if the true system is of order d_1 we also find that the reduced model reaches the Cramér-Rao bound, i.e.,

$$\mathrm{Cov}\,\hat{\eta} = \frac{\lambda}{N}R_{11}^{-1}. \tag{6.27}$$

The conclusion from this is that *the variance of the reduced FIR model is never higher than the variance obtained by direct estimation*.

Comments: We could here remark that the variance reduction is related to performing the reduction step "correctly". If (6.24) is approximated by the sample sum over the same input data as used to estimate $\hat{\theta}$ it follows that the reduced estimate will always be equal to the direct one. Moreover, the variance reduction can be traced to the fact that the approximation aspect of the direct estimation method depends on the finite sample properties of u over $t = 1, \ldots, N$. If expectation is carried out only with respect to e we have

$$\mathrm{E}_e\,\hat{\eta}_N = \eta^* + H_N\xi_0$$

and this is the root of the increased variance in the direct method.

6.5 The General Case

The result that it may be advantageous to use L_2 model reduction of a high order estimated model, rather than to directly estimate a low order one is intriguing. Using the basic tools, more general situations can be investigated. Here we focus on OE model structures. We assume that the low order model structure contains the true system, i.e., we look at the no undermodeling case.

Let the underlying system be given by

$$y(t) = \sum_{k=0}^{\infty} g_k u(t-k) + e(t) = G_0(q)u(t) + e(t), \tag{6.28}$$

with the same assumptions on e and u as in (6.13). Parameterize two OE model structures $G(q,\theta)$ and $G(q,\eta)$ where $\dim\theta \geq \dim\eta$, i.e.,

$$\theta = \begin{pmatrix} b_1 & \dots & b_{n_b} & f_1 & \dots & f_{n_f} \end{pmatrix}^T \tag{6.29}$$

$$\eta = \begin{pmatrix} b_1 & \dots & b_{n_{b_0}} & f_1 & \dots & f_{n_{f_0}} \end{pmatrix}^T, \tag{6.30}$$

where $n_b \geq n_{b_0}$ and $n_f \geq n_{f_0}$. Furthermore, we assume the existence of some θ^* and a unique η^* such that

$$G(e^{i\omega},\theta^*) = G(e^{i\omega},\eta^*) = G_0(e^{i\omega}) \tag{6.31}$$

for almost all ω. Note here that the parameters η form a subset of θ. This can be written as

$$S_0^T\theta = \eta, \tag{6.32}$$

where

$$S_0 = \begin{pmatrix} e_1 & \dots & e_{n_{b_0}} & e_{n_b+1} & \dots & e_{n_b+n_{f_0}} \end{pmatrix} \tag{6.33}$$

and e_j is the jth column of the $(n_b+n_f)\times(n_b+n_f)$ identity matrix.

The gradients of $\hat{y}(t,\theta)$ and $\hat{y}(t,\eta)$ equals (see (2.21) and Section 2.1.4)

$$\begin{aligned}\Psi(t,\theta) &= \frac{d}{d\theta}G(q,\theta)u(t) = \frac{d}{d\theta}\frac{B(q,\theta)}{F(q,\theta)}u(t)\\ &= \begin{pmatrix} q^{-n_{k_0}} \\ \vdots \\ q^{-n_{k_0}-n_b+1} \\ -q^{-1}G(q,\theta) \\ \vdots \\ -q^{-n_f}G(q,\theta) \end{pmatrix}\frac{1}{F(q,\theta)}u(t)\end{aligned} \tag{6.34}$$

and

$$\Psi(t,\eta) = \frac{d}{d\eta}G(q,\eta)u(t) = \frac{d}{d\eta}\frac{B(q,\eta)}{F(q,\eta)}u(t)$$
$$= \begin{pmatrix} q^{-n_{k_0}} \\ \vdots \\ q^{-n_{k_0}-n_{b_0}+1} \\ -q^{-1}G(q,\eta) \\ \vdots \\ -q^{-n_{f_0}}G(q,\eta) \end{pmatrix} \frac{1}{F(q,\eta)}u(t) \tag{6.35}$$

By observing that

$$\frac{B(q,\theta^*)}{F(q,\theta^*)} = G_0(q) \tag{6.36}$$

we find that

$$B(q,\theta^*) = B_0(q)L(q), \text{ and } F(q,\theta^*) = F_0(q)L(q). \tag{6.37}$$

Here $L(q)$ is a monic FIR filter of length $r+1$ and

$$r = \min(n_b - n_{b_0}, n_f - n_{f_0}), \tag{6.38}$$

i.e.,

$$L(q) = 1 + l_1 q^{-1} + \ldots + l_r q^{-r} = \sum_{k=0}^{r} l_k q^{-k}, \tag{6.39}$$

where we use the convention that $l_0 = 1$. We also obviously have that

$$\frac{B(q,\eta^*)}{F(q,\eta^*)} = G_0(q) \tag{6.40}$$

Putting (6.34),(6.36), and (6.37) together gives

$$\Psi(t,\theta^*) = \begin{pmatrix} q^{-n_{k_0}} \\ \vdots \\ q^{-n_{k_0}-n_b+1} \\ -q^{-1}G_0(q) \\ \vdots \\ -q^{-n_f}G_0(q) \end{pmatrix} \frac{1}{L(q)F_0(q)}u(t). \tag{6.41}$$

In the same way we get from (6.35), and (6.40)

$$\Psi(t,\eta^*) = \begin{pmatrix} q^{-n_{k_0}} \\ \vdots \\ q^{-n_{k_0}-n_{b_0}+1} \\ -q^{-1}G_0(q) \\ \vdots \\ -q^{-n_{f_0}}G_0(q) \end{pmatrix} \frac{1}{F_0(q)} u(t). \tag{6.42}$$

Looking at these two expressions and utilizing (6.32) we get the important relation

$$\Psi(t,\eta^*) = S_0^T L(q)\Psi(t,\theta^*). \tag{6.43}$$

Let us now consider (6.7) with $W(\omega) = \Phi_u(\omega)$:

$$\begin{aligned} V(\eta,\hat{\theta}) &= \int_{-\pi}^{\pi} |G(e^{i\omega},\eta) - G(e^{i\omega},\hat{\theta})|^2 \Phi_u(\omega) d\omega \\ &= \mathrm{E}\left[(G(q,\eta) - G(q,\hat{\theta}))u(t)\right]^2 \\ &= \mathrm{E}\,\varepsilon^2(t,\eta,\hat{\theta}), \end{aligned} \tag{6.44}$$

with obvious definition of $\varepsilon^2(t,\eta,\hat{\theta})$. Note that $\hat{\theta}$ should be regarded as fixed (independent of u) in this expression. Define as before

$$\hat{\eta}_N = \arg\min_{\eta} V(\eta,\hat{\theta}_N) \tag{6.45}$$

From the discussion in Ljung (1999b, Appendix B) it follows that difference between the expected value of $\hat{\eta}_N$ and η^* (defined by (6.12)) is small for large N. So the limiting estimate of the two step method (estimation and reduction) gives approximately the same limiting estimate as the direct estimation method.

In order to calculate the variance of the reduced order model we need to derive the expressions for $V''_{\eta\eta}$ and $V''_{\eta\hat{\theta}}$ from (6.44):

$$V'_{\eta}(\eta,\hat{\theta}) = \mathrm{E}\,\Psi(t,\eta)\varepsilon(t,\eta,\hat{\theta}) \tag{6.46}$$

$$V''_{\eta\eta}(\eta,\hat{\theta}) = \mathrm{E}\,\varepsilon(t,\eta,\hat{\theta})\frac{d}{d\eta}\Psi(t,\eta) + \mathrm{E}\,\Psi(t,\eta)\Psi^T(t,\eta) \tag{6.47}$$

$$V''_{\eta\theta}(\eta,\hat{\theta}) = -\mathrm{E}\,\Psi(t,\eta)\Psi^T(t,\hat{\theta}) \tag{6.48}$$

Since both parameterizations (in η and θ) are rich enough to describe the underlying true system, we know that the residuals are $\varepsilon(t,\eta,\hat{\theta})$ are white and independent

of past inputs and past outputs. This implies that the first term in (6.47) vanishes. Evaluating the last two expressions at (η^*, θ^*) gives

$$V''_{\eta\eta}(\eta^*, \theta^*) = V''_{\eta\eta}(\eta, \hat{\theta})\Big|_{\eta=\eta^*,\ \hat{\theta}=\theta^*} = \mathrm{E}\,\Psi(t, \eta^*)\Psi^T(t, \eta^*) \tag{6.49}$$

$$V''_{\eta\theta}(\eta^*, \theta^*) = -\mathrm{E}\,\Psi(t, \eta^*)\Psi^T(t, \theta^*) \tag{6.50}$$

Estimation of the high order system $G(q, \theta)$ gives $\hat{\theta}$ with covariance

$$\mathrm{Cov}\,\hat{\theta}_N = P_\theta = \frac{\lambda}{N}\left[\mathrm{E}\,\Psi(t, \theta^*)\Psi^T(t, \theta^*)\right]^{-1} \tag{6.51}$$

Putting Equations (6.10),(6.50), and (6.51) together we find that

$$\begin{aligned}\mathrm{Cov}\,\hat{\eta} = &\left[\mathrm{E}\,\Psi(t, \eta^*)\Psi^T(t, \eta^*)\right]^{-1}\left[\mathrm{E}\,\Psi(t, \eta^*)\Psi^T(t, \theta^*)\right]\\ &\times \frac{\lambda}{N}\left[\mathrm{E}\,\Psi(t, \theta^*)\Psi^T(t, \theta^*)\right]^{-1}\\ &\times \left[\mathrm{E}\,\Psi(t, \theta^*)\Psi^T(t, \eta^*)\right]\left[\mathrm{E}\,\Psi(t, \eta^*)\Psi^T(t, \eta^*)\right]^{-1}.\end{aligned} \tag{6.52}$$

We will later show that this expression can simplified to

$$\frac{\lambda}{N}\left[\mathrm{E}\,\Psi(t, \eta^*)\Psi^T(t, \eta^*)\right]^{-1}, \tag{6.53}$$

which is the Cramér-Rao bound for the estimation of η. This can equivalently be stated as

$$\begin{aligned}&\left[\mathrm{E}\,\Psi(t, \eta^*)\Psi^T(t, \eta^*)\right] =\\ &\left[\mathrm{E}\,\Psi(t, \eta^*)\Psi^T(t, \theta^*)\right]\left[\mathrm{E}\,\Psi(t, \theta^*)\Psi^T(t, \theta^*)\right]^{-1}\left[\mathrm{E}\,\Psi(t, \theta^*)\Psi^T(t, \eta^*)\right],\end{aligned} \tag{6.54}$$

which will be used later on. In order to prove this we need some more results.

We start by giving a lemma regarding rank deficient matrices.

LEMMA 6.1

Let A be a $n \times n$-dimensional positive semidefinite symmetric matrix of rank m. Define $\tilde{A} = A + \delta I$ with $\delta > 0$. Then the following holds:

i) $\tilde{A}^{-1}A = A\tilde{A}^{-1} = I - \delta\tilde{A}^{-1}$.

ii) $\lim_{\delta\to 0}\delta^2\tilde{A}^{-1} = 0$.

Proof

i) $I = \tilde{A}^{-1}\tilde{A} = \tilde{A}^{-1}(A + \delta I) \Leftrightarrow \tilde{A}^{-1}A = I - \delta\tilde{A}^{-1}$.
The other equality follows similarly.

ii) Since A is symmetric it follows that

$$A = UDU^T, \tag{6.55}$$

with $D = \text{diag}(d_1, \ldots, d_m, 0, \ldots, 0)$ and $UU^T = U^TU = I$.
Adding δI to both sides of (6.55) gives

$$A + \delta I = U(D + \delta I)U^T.$$

Inverting both sides gives (since $U^{-1} = U^T$)

$$\tilde{A}^{-1} = U(D + \delta I)^{-1}U^T.$$

Hence we get

$$\delta^2\tilde{A}^{-1} = U\bar{D}U^T, \ \bar{D} = \text{diag}(\frac{\delta^2}{d_1 + \delta}, \ldots, \frac{\delta^2}{d_m + \delta}, \delta, \ldots, \delta).$$

From this it follows that

$$\lim_{\delta\to 0} \delta^2\tilde{A}^{-1} = U \lim_{\delta\to 0} \bar{D}U^T = U \cdot 0 \cdot U^T = 0.$$

❑

Before presenting the next lemma we introduce the notation:
$(A)_{i,j}$ = the (i, j)th element of A.
$(A)_{\cdot,j}$ = the jth column of A.
Furthermore we extend the definition of S_0 in (6.33) to

$$S_k = \begin{pmatrix} e_{k+1} & \ldots & e_{k+n_{b_0}} & e_{k+n_b+1} & \ldots & e_{k+n_b+n_{f_0}} \end{pmatrix}. \tag{6.56}$$

The covariance function of the gradient $\Psi(t, \theta^*)$ is defined as

$$R_\theta(k) = \text{E}\,\Psi(t + k, \theta^*)\Psi^T(t, \theta^*) = \text{E}\,\Psi(t, \theta^*)\Psi^T(t - k, \theta^*). \tag{6.57}$$

We are now ready to state a lemma connecting $R_\theta(k)$ to $R_\theta(0)$.

LEMMA 6.2 (PROPERTIES OF THE COVARIANCE FUNCTION)
Let $R_\theta(k)$ be given by (6.57). Then it holds that:

i) $R_\theta(k)S_0 = R_\theta(0)S_k, \quad 0 \le k \le r.$

ii) $S_0^T R_\theta(-k) = S_k^T R_\theta(0), \quad 0 \le k \le r.$

iii) $S_0^T R_\theta(k-m)S_0 = S_m^T R_\theta(0)S_k, \quad 0 \le k \le r.$

Proof

i) Studying the jth, $1 \le j \le n_b - k$, column of $R_\theta(k)$, where $0 \le k \le r$, gives

$$\begin{aligned}(R_\theta(k))_{\cdot,j} &= (\mathrm{E}\,\Psi(t,\theta^*)\Psi^T(t-k,\theta^*))_{\cdot,j} \\ &= \mathrm{E}\,\Psi(t,\theta^*)\left(q^{-n_{k_0}-k-j+1}\frac{1}{L(q)F_0(q)}u(t)\right) \\ &= \mathrm{E}\,\Psi(t,\theta^*)(\Psi^T(t,\theta^*))_{\cdot,k+j} = (R_\theta(0))_{\cdot,k+j}\end{aligned}$$

Similarly for $n_b + 1 \le j \le n_b + n_f - k$ we get

$$\begin{aligned}(R_\theta(k))_{\cdot,j} &= (\mathrm{E}\,\Psi(t,\theta^*)\Psi^T(t-k,\theta^*))_{\cdot,j} \\ &= \mathrm{E}\,\Psi(t,\theta^*)\left(-q^{-k-j+1}\frac{G_0(q)}{L(q)F_0(q)}u(t)\right) \\ &= \mathrm{E}\,\Psi(t,\theta^*)(\Psi^T(t,\theta^*))_{\cdot,k+j} = (R_\theta(0))_{\cdot,k+j}\end{aligned}$$

Now the multiplication $R_\theta(k)S_0$ picks out the first n_{b_0} rows and rows with indices between $n_b + 1$ and $n_b + n_{f_0}$ from $R_\theta(k)$, whereas $R_\theta(0)S_k$ picks out rows shifted $k+1$ steps away (relatively to S_0) from $R_\theta(0)$. This means that we pick out exactly those columns corresponding to each other by the multiplication with S_0 and S_k.

ii) Follows after transposing i)

$$S_0^T R_\theta(-k) = S_0^T R_\theta^T(k) = (R_\theta(k)S_0)^T = (R_\theta(0)S_k)^T = S_k^T R_\theta(0)$$

iii) Sketch. Observe from ii) that $S_k^T R_\theta(0)$ picks out the same rows from $R_\theta(0)$ as $R_\theta(0)S_k$ picks out columns from $R_\theta(0)$. Note also that from the special structure of $R_\theta(0)$ we have

$$R_\theta(0) = \begin{pmatrix} A_1 & A_2 \\ A_2^T & A_3 \end{pmatrix},$$

and every block A_i is constant along its diagonals, i.e.,

$$(A_i)_{l,k} = (A_i)_{l+s,k-s}, \quad j = 1, 2, 3.$$

Concentrating on block A_1 we see that $S_m^T R_\theta(0) S_k$ picks out a block of size $n_{b_0} \times n_{b_0}$ from A_1, starting at $(A_1)_{k+1,m+1}$. This block lies entirely inside A_1 since $0 \leq k, m \leq r$ and A_1 is of size $n_b \times n_b$.
Now assume that $k \geq m$. This means that $S_0^T R_\theta(k-m) S_0 = S_0^T R_\theta(0) S_{k-m}$ according to i), i.e., we pick out a block of size $n_{b_0} \times n_{b_0}$ from A_1, starting at $(A_1)_{k-m+1,1}$. Since A_1 is constant along its diagonals this is the same as picking the block from $(A_1)_{k+1,m+1}$ (which is exactly what $S_m^T R_\theta(0) S_k$ does). The same argumentation holds for A_3. ❑

Before proving that the reduced model meets the Cramér-Rao bound, we must point out that the covariance of $\hat{\theta}$ given by (6.51) is not well defined in most cases. This due to fact that $r \geq 1$ implies that θ^* is actually an r-dimensional set of limiting estimates. Therefore will P_θ be rank deficient. In order to take care of this we make a regularization. This means that we replace the original minimization problem

$$\hat{\theta}_N = \arg\min_\theta V_N(\theta), \tag{6.58}$$

with

$$\hat{\theta}_N = \arg\min_\theta V_N(\theta) + \frac{\delta}{2}\|\theta - \bar{\theta}\|_2^2, \tag{6.59}$$

for some $\bar{\theta}$ minimizing (6.58). This also implies that (6.51) will be replaced by

$$\frac{\lambda}{N}\left[\mathrm{E}\,\Psi(t,\theta^*)\Psi^T(t,\theta^*) + \delta I\right]^{-1} = (R_\theta(0) + \delta I)^{-1} = \tilde{R}_\theta^{-1}(0). \tag{6.60}$$

Here $\delta > 0$ and the last equality is the definition of $\tilde{R}_\theta^{-1}(0)$. In the proof we will then let δ tend to zero, and hence take away the effect of the regularization.

THEOREM 6.1 (REDUCED MODEL VARIANCE)
Assume that the true system is given by

$$y(t) = G_0(q)u(t) + e(t),$$

where e is white noise with variance λ and u is a stationary stochastic process independent of e, with known spectrum $\Phi_u(\omega)$. Furthermore, we assume that $G(q,\theta)$ and $G(q,\eta)$ are two model structures of OE type that both contain the true system $G_0(q)$. Let $\hat{\theta}$ minimize $V_N(\theta)$ (given by (2.10)) and $\hat{\eta}$ minimize

$$V(\eta,\hat{\theta}) = \int_{-\pi}^{\pi} |G(e^{i\omega},\eta) - G(e^{i\omega},\hat{\theta})|^2 \Phi_u(\omega) d\omega.$$

Then the asymptotic variance of $\hat{\eta}$ equals the Cramér-Rao bound, or equivalently (6.54) holds.

Proof Using (6.43) we get that

$$\begin{aligned}\mathrm{E}\,\Psi(t,\eta^*)\Psi^T(t,\theta^*) &= \mathrm{E}\,S_0^T L(q)\Psi(t,\theta^*)\Psi^T(t,\theta^*)\\ &= S_0^T \sum_{k=1}^{r} l_m\,\mathrm{E}\,\Psi(t-m,\theta^*)\Psi^T(t,\theta^*)\\ &= S_0^T \sum_{k=1}^{r} l_m R_\theta(-m) = \sum_{k=1}^{r} l_m S_m^T R_\theta(0).\end{aligned}$$

Plugging this into the right hand side of (6.54) gives

$$\begin{aligned}&\left[\mathrm{E}\,\Psi(t,\eta^*)\Psi^T(t,\theta^*)\right]\left[\mathrm{E}\,\Psi(t,\theta^*)\Psi^T(t,\theta^*)+\delta I\right]^{-1}\left[\mathrm{E}\,\Psi(t,\theta^*)\Psi^T(t,\eta^*)\right]\\ &\quad= \left(\sum_{m=0}^{r} l_m S_m^T R_\theta(0)\right)\tilde{R}_\theta^{-1}(0)\left(\sum_{k=0}^{r} l_k R_\theta(0) S_k\right)\\ &\quad= \left(\sum_{m=0}^{r} l_m S_m^T R_\theta(0)\right)\left(\sum_{k=0}^{r} l_k (I-\delta\tilde{R}_\theta^{-1}(0)) S_k\right)\\ &= \sum_{m=0}^{r}\sum_{k=0}^{r} l_m l_k S_m^T R_\theta(0) S_k - \delta\left(\sum_{m=0}^{r} l_m S_m^T R_\theta(0)\right)\left(\sum_{k=0}^{r} l_k \tilde{R}_\theta^{-1}(0) S_k\right)\\ &= \sum_{m=0}^{r}\sum_{k=0}^{r} l_m l_k S_m^T R_\theta(0) S_k - \delta\sum_{m=0}^{r}\sum_{k=0}^{r} l_m l_k S_m^T S_k + \delta^2\sum_{k=0}^{r} l_k \tilde{R}_\theta^{-1}(0) S_k.\end{aligned}$$

Here we have used Lemma 6.1 i) several times. Letting $\delta \to 0$ the last two sums vanish according to Lemma 6.1 ii). Moreover we get (using Lemma 6.2)

$$\begin{aligned}&\lim_{\delta\to 0}\left[\mathrm{E}\,\Psi(t,\eta^*)\Psi^T(t,\theta^*)\right]\left[\mathrm{E}\,\Psi(t,\theta^*)\Psi^T(t,\theta^*)+\delta I\right]^{-1}\left[\mathrm{E}\,\Psi(t,\theta^*)\Psi^T(t,\eta^*)\right]\\ &\quad= \sum_{m=0}^{r}\sum_{k=0}^{r} l_m l_k S_m^T R_\theta(0) S_k = S_0^T\sum_{m=0}^{r}\sum_{k=0}^{r} l_m l_k R_\theta(k-m) S_0\\ &\quad= S_0^T\sum_{m=0}^{r}\sum_{k=0}^{r} l_m l_k\,\mathrm{E}\,q^{-m}\Psi(t,\theta^*)q^{-k}\Psi^T(t,\theta^*) S_0\\ &\quad= \mathrm{E}\,S_0^T\sum_{m=0}^{r} l_m q^{-m}\Psi(t,\theta^*)\sum_{k=0}^{r} l_k q^{-k}\Psi^T(t,\theta^*) S_0\\ &= \mathrm{E}\,S_0^T L(q)\Psi(t,\theta^*)L(q)\Psi^T(t,\theta^*)S_0 = \mathrm{E}\,\Psi(t,\eta^*)\Psi^T(t,\eta^*).\end{aligned}$$

Or equivalently, the reduced order estimate meets the Cramér-Rao bound. ❑

From this we can draw the following conclusions:

- The reduced model has an asymptotic covariance matrix equal to the Cramér-Rao bound.
- L_2 model reduction is optimal in view of achieving the lowest possible covariance of the estimates.
- Most other model reduction techniques do not reach the Cramér-Rao bound, e.g., balanced truncation. This since in order to reach that bound, the model reduction $\eta = f(\theta)$ needs to have a structure of the derivative (6.9) equal to the one that L_2 reduction has.
- The tools presented in Section 6.3 can immediately be applied to other model reduction methods. The only problem lies in defining proper loss functions for the other reduction techniques. This need not to be obvious for methods such as balance truncation.

6.6 Conclusions

We have discussed the variance properties of L_2 model reduction. We have shown the it could be strictly better to estimate a high order FIR model and reduce it using L_2 model reduction compared to estimating the low order model directly. In the general FIR case we will at least reach the Cramér-Rao bound if the low order model is rich enough.

In the last section we showed the maybe more useful result that the reduced low order models are efficient even in more general situations like OE structures. It is also interesting to note that the calculations show that L_2 model reduction is optimal in reducing the variance of the high order estimate.

Bibliography

Aronsson, M., Arvastson, L., Holst, J., Lindoff, B., and Svensson, A. (1998). Bootstrap control. Technical Report LUFTD2/TFMS-3146-SE, Department of Mathematical Statistics, Lund Institute of Technology, Lund, Sweden.

Bose, A. (1988). Edgeworth correction by bootstrap in autoregressions. *The Annals of Statistics*, 16(4):1709–1722.

Box, G. E. P. and Jenkins, G. M. (1976). *Time Series Analysis, Forcasting and Control*. Holden-Day.

Cleveland, W. (1979). Robust locally weighted regression and smoothing scatterplots. *JASA*, 74:829–836.

Cleveland, W. and Devlin, S. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *JASA*, 83:596–610.

Cleveland, W. and Loader, C. (1994). Smoothing by local regression: Principles and methods. Technical report, AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974, USA.

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and their Applications*. Cambridge University Press.

Draper, N. and Smith, H. (1998). *Applied Regression Analysis*. John Wiley & Sons, 3rd edition.

Draper, N. R. (1995). Confidence intervals versus regions. *The Statistician*, 44(3):399–403.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7:1–26.

Efron, B. (1987). Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82(397):171–200.

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*. Chapman & Hall.

Forssell, U. (1998). Asymptotic variance expressions for identified black-box models. Technical Report LiTH-ISY-R-2089, Department of Electrical Engineering, Linköping University, Linköping, Sweden. Submitted to Systems & Control Letters.

Freedman, D. (1984). On bootstrapping two-stage least-squares estimates in stationary linear models. *The Annals of Statistics*, 12(3):827–842.

Garulli, A. and Reinelt, W. (1999). On model error modeling in set membership identification. Technical Report LiTH-ISY-R-2200, Department of Electrical Engineering, Linköping University, S-581 83 Linköping, Sweden.

Guillaume, P., Kollár, I., and Pintelon, R. (1996). Statistical analysis of nonparametric transfer function estimates. *IEEE Transactions on Instrumentation and Measurement*, 45(2):594–600.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Macmillan.

Hjalmarsson, H. (1993). *Aspects on Incomplete Modeling in System Identification*. Phd thesis 298, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Hjalmarsson, H. and Ljung, L. (1992). Estimating model variance in case of undermodeling. *IEEE Transactions on Automatic Control*, 37:1004–1008.

Hjorth, U. (1994). *Computer Intensive Statistical Methods*. Chapman & Hall.

Kosut, R. L. (1995). Uncertainty model falsification: A system identification paradigm compatible with robust control design. In *Proceedings of the 34th Conference on Decision and Control*, pages 3492–3497, New Orleans, LA.

Larssen, J. (1992). A generalization error estimate for nonlinear systems. In *Proceedings IEEE-SP Workshop*, pages 29–38.

Ljung, L. (1985a). Asymptotic properties of the least squares method for estimating transfer functions and disturbance spectra. Technical Report LiTH-ISY-R-0709, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Ljung, L. (1985b). Asymptotic variance expressions for identified black-box transfer function models. *IEEE Transactions on Automatic Control*, 30(9):834–844.

Ljung, L. (1997a). Identification, model validation, and control. Plenary address at 36th IEEE CDC, San Diego, USA.

Ljung, L. (1997b). *System Identification Toolbox – User's Guide*. The Mathworks Inc, 24 Prime Park Way, Natick, MA.

Ljung, L. (1998). Identification for control – What is there to learn? In *Proceedings of Workshop on Learning, Control and Hybrid Systems*, Bangalore, India.

Ljung, L. (1999a). Model validation and model error modeling. In Wittenmark, B. and Rantzer, A., editors, *The Åström Symposium on Control*. Studentlitteratur, Lund, Sweden.

Ljung, L. (1999b). *System Identification: Theory for the User*. Prentice-Hall, 2nd edition.

Ljung, L. and Hjalmarsson, H. (1995). System identification through the eyes of model validation. In *Proc. Third European Control Conference*, volume 3, Rome, Italy.

Loader, C. (1997). *Locfit: An Introduction*. AT&T Bell Laboratories.

Lovera, M. (1997). *Subspace Identification Methods:Theory and Applications*. Phd thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy.

Maciejowski, J. M. and Ober, R. J. (1988). Balanced parametrizations and canonical forms for system identification. In *Proceedings of the 8th IFAC Symposium on System Identification and Parameter Estimation*, volume 2, pages 989–996, Pergamon, Oxford.

Manoukian, E. B. (1986). *Modern Concepts and Theorems of Mathematical Statistics*. Springer Verlag.

McKelvey, T. (1995). *Identification of State-Space Models from Time and Frequency Data*. Phd thesis 380, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Nickerson, D. M. (1994). Construction of a conservative confidence region from projections of an exact confidence region in multiple linear regression. *The American Statistiscian*, 48(2):120–124.

Nordgaard, A. (1995). *Computer-intensive Methods for Dependent Oservations*. Phd thesis 409, Department of Mathematics, Linköping University, Linköping, Sweden.

Politis, D. N. (1998). Computer-intensive methods in statistical analysis. *IEEE Signal Processing Magazine*, 15(1):39–55.

Politis, D. N., Romano, J. P., and Lai, T. (1992). Bootstrap confidence bands for spectra and cross-spectra. *IEEE Transactions on Signal Processing*, 40(5):1206–1215.

Poolla, K., Khargonekar, P., Tikku, A., Krause, J., and Nagpal, K. (1994). A time-domain approach to model validation. *IEEE Transactions on Automatic Control*, 39(5):951–959.

Pötscher, B. M. and Prucha, I. R. (1997). *Dynamic Nonlinear Econometric Models*. Springer-Verlag.

Reinelt, W., Garulli, A., Ljung, L., Braslavsky, J. H., and Vicino, A. (1999). Model error concepts in identification for control. In *38th IEEE Conference on Decision and Control, Phoenix, AZ, USA*.

Sjöberg, J. (1995). *Non-Linear System Identification with Neural Networks*. Phd thesis 381, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Smith, R. S. and Doyle, J. C. (1992). Model validation: A connection between robust control and identification. *IEEE Transactions on Automatic Control*, 37(7):942–952.

Söderström, T. and Stoica, P. (1989). *System Identification*. Prentice-Hall International.

Söderström, T., Stoica, P., and Friedlander, B. (1991). An indirect prediction error method for system identification. *Automatica*, 27:183–188.

Stenman, A. (1999). *Model on Demand: Algorithms, Analysis and Applications*. PhD thesis, Dept of EE, Linköping University, SE-581 83, Linköping, Sweden.

Stenman, A., Gustafsson, F., Rivera, D., Ljung, L., and McKelvey, T. (1999). On adaptive smoothing of empirical transfer function estimates. In Chen, H. and Wahlberg, B., editors, *Preprints of the 14th World Congress of IFAC, Beijing, P.R. China*, volume H, pages 415–420. Elsevier Science.

Tjärnström, F. and Forssell, U. (1999). Comparison of methods for probabilistic uncertainty bounding. In *Proceedings of the 38th IEEE Conference on Decision and Control*, Pheonix, Arizona, USA.

Tjärnström, F. and Ljung, L. (1999). Estimating the variance in case of undermodeling using bootstrap. In *Proceedings of the 38th IEEE Conference on Decision and Control*, Pheonix, Arizona, USA.

Tjärnström, F. and Ljung, L. (1999). Minimizing the variance of transfer function estimates. In Yakubovich, V. and Fradkov, A., editors, *6th Saint Petersburg Symposium on Adaptive Systems Theory*, volume 1, pages 194–200, S:t Petersburg, Russia.

Van Overschee, P. and De Moor, B. (1996). *Subspace identification for linear systems*. Kluwer.

Viberg, M. (1995). Subspace-based methods for identification of linear time-invariant systems. *Automatica*, 31(12):1835–1851.

Vuerinckx, R., Pintelon, R., Schoukens, J., and Rolain, Y. (1998). Obtaing accurate confidence regions for the estimated zeros and poles in system identification problems. In *Proceedings on the 37th IEEE Conference on Decision and Control*, pages 4464–4469, Tampa, Florida, USA.

Wahlberg, B. (1987). *On the Identification and Approximation of Linear Systems*. Phd thesis 163, Department of Electrical Engineering, Linköping University.

Wahlberg, B. (1989). Model reduction of high order estimated models: The asymptotic ML approach. *Int. J. Control*, 49(1):169–192.

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*. Chapman & Hall.

Zhou, K., Glover, K., Bodenheimer, B., and Doyle, J. (1994). Mixed H_2 and H_∞ performance objectives I: Robust performance analysis. *IEEE Transactions on Automatic Control*, 39(8):1564–1574.

Zhou, Y. C. and Backx, A. C. M. P. (1993). *Identification of Multivariable Industrial Process for Simulation Diagonsis and Control*. Springer-Verlag.

Zoubir, A. M. and Boashash, B. (1998). The bootstrap and its application in signal processing. *IEEE Signal Processing Magazine*, 15(1):56–76.