Technical report from Automatic Control at Linköpings universitet

System Identification

Lennart Ljung

Division of Automatic Control

E-mail: ljung@isy.liu.se

29th June 2007

Report no.: LiTH-ISY-R-2809
Accepted for publication in Wiley Encyclopedia of Electrical and Electronics Engineering

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

Technical reports from the Automatic Control group in Linköping are available from http://www.control.isy.liu.se/publications.

Abstract

This is a survey of System Identification.

Keywords: identification

System Identification

Lennart Ljung
Department of Electrical Engineering, Linköping University
S-581 83 Linköping, Sweden. e-mail ljung@isy.liu.se

April 27, 1997

1 Introduction

The process of going from observed data to a mathematical model is fundamental in science and engineering. In the control area this process has been termed "System Identification" and the objective is then to find dynamical models (difference or differential equations) from observed input and output signals. Its basic features are however common with general model building processes in statistics and other sciences.

System Identification covers the problem of building models of systems where both when insignificant prior information is available and when the system's properties are known, up to a few parameters (physical constants). Accordingly, one talks about *black box* and *gray box* models. Among black box models there are familiar linear models such as ARX and ARMAX, and among non-linear black box models we have, e.g., Artificial Neural Networks (ANN).

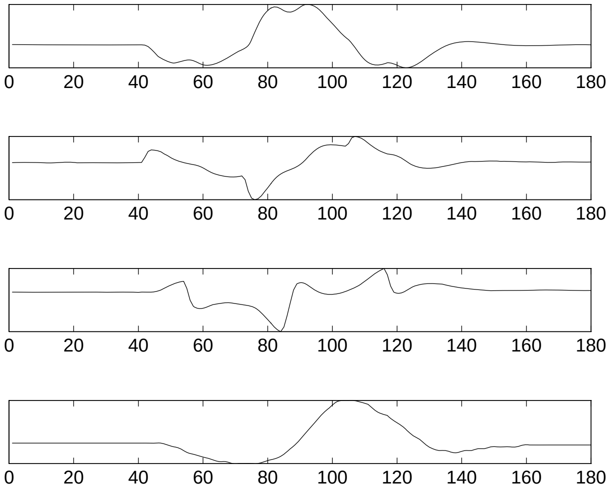

Figure 1: Results from test flights of the new Swedish aircraft JAS-Gripen, developed by SAAB Military Aircraft AB, Sweden. From above) Pitch rate. b) Elevator angle. c) Canard angle. d) Leading edge flap.

1.1 The Problem

The area of system identification begins and ends with real data. Data are required to build and to validate models. The result of the modeling process can be no better than what corresponds to the information contents in the data.

Let us take a look at two data sets:

Example 1 An unstable aircraft. *Figure 1 shows some results from test flights of the new Swedish aircraft JAS-Gripen, developed by SAAB Military Aircraft AB, Sweden. The problem is to use the information in these data to determine the dynamical properties of the aircraft for fine-tuning regulators, for simulations, and so on. Of particular interest are the aerodynamical derivatives.*

Example 2 Vessel dynamics. *Figure 2 shows data from a pulp factory.*

They are collected from one of the buffer vessels. The problem is to determine the residence time in the vessel. The pulp spends about 48 hours total in the process, and knowing the residence time in the different vessels is important in order to associate various portions of the pulp with the different chemical actions that have taken place in the vessel at different times. (The κ-number is a quality property that in this context can be seen as a marker allowing us to trace the pulp.)

So, the bottom line of these examples is that we have collected input-output data from a process or a plant, and we need to extract information from these to find out (something about) the process's dynamical properties.

1.2 Background and Literature

System Identification has its roots in standard statistical techniques and many of the basic routines have direct interpretations as well known statistical methods such as Least Squares and Maximum Likelihood. The control community took an active part in the development and application of these basic techniques to dynamic systems right after the birth of "modern control theory" in the early 1960's. Maximum likelihood estimation was applied to difference equations (ARMAX models) by [Åström and Bohlin, 1965] and thereafter a wide range of estimation techniques and model parameterizations flourished. By now, the area is well matured with established and well understood techniques. Industrial use and application of the techniques has become standard. See [Ljung, 1995] for a common software package.

The literature on System Identification is extensive. For a practical user oriented introduction we may mention [Ljung and Glad, 1994]. Texts that go deeper into the theory and algorithms include [Ljung, 1987], and [Söderström and Stoica, 1989]. A classical treatment is [Box and Jenkins, 1970].

These books all deal with the "mainstream" approach to system identification, as described in this article. In addition, there is a substantial literature on other approaches, such as "set membership" (compute all those models that reproduce the observed data within a certain given error bound), estimation of models from given frequency response measurement [Schoukens and Pintelon, 1991],

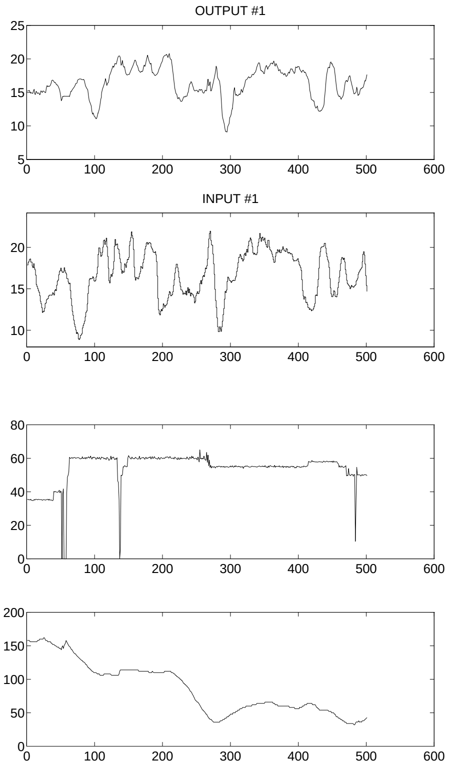

Figure 2: From the pulp factory at Skutskär, Sweden. The pulp flows continuously through the plant via several buffer tanks. From above: a) The κ-number of the pulp flowing into a buffer vessel. b) The κ-number of the pulp coming out from the buffer vessel. c) Flow out from the buffer vessel. d) Level in the buffer vessel.

on-line model estimation [Ljung and Söderström, 1983], non-parametric frequency domain methods [Brillinger, 1981], etc. To follow the development in the field, the IFAC series of Symposia on System Identification (Budapest, Hungary (1991), Copenhagen, Denmark (1994), Fukuoka, Japan (1997)) is also a good source.

1.3 Outline

The system identification procedure is characterized by four basic ingredients:

1. The observed data
2. A set of candidate models
3. A criterion of fit
4. Validation

The problem can be expressed as finding that model in the candidate set, that best describes the data, according to the criterion, and then evaluate and validate that model's properties. To do this we need to penetrate a number of things:

1. First, in Section 2 we give a preview of the whole process, as applied to the simplest set of candidate models.

2. Then, at some length, in Sections 3 and 4 we display and discuss the most common sets of candidate models used in system identification. In general terms, a model will be a predictor of the next output $y(t)$ from the process, given past observations Z^{t-1}, and parameterized in terms of a finite-dimensional parameter vector θ:

$$\hat{y}(t|\theta) = g(\theta, Z^{t-1}) \tag{1}$$

3. We then, in Section 5, discuss the criterion of fit for general model sets,. This will have the character

$$V_N(\theta) = \sum \|y(t) - \hat{y}(t|\theta)\|^2 \tag{2}$$

We also discuss how to find the best model (minimize the criterion), how to assess its properties.

4. In Section 6 we shall describe special methods for linear black-box models. This includes frequency analysis, spectral analysis and so called subspace methods for linear state-space models.

5. We then turn to the practical issues of system identification; to assure good quality of the data by proper experiment design (Section 7); how to decide upon a good model structure (Section 8); and how to deal with the data (Section 9).

2 Displaying the Basic Ideas: ARX Models and the Linear Least Squares Method

The Model

We shall generally denote the system's input and output at time t by $u(t)$ and $y(t)$, respectively. Perhaps the most basic relationship between the input and output is the *linear difference equation*

$$y(t) + a_1 y(t-1) + \ldots + a_n y(t-n) = b_1 u(t-1) + \ldots + b_m u(t-m) \quad (3)$$

We have chosen to represent the system in *discrete time*, primarily since observed data are always collected by sampling. It is thus more straightforward to relate observed data to discrete time models. Nothing prevents us however from working with continuous time models: we shall return to that in Section 3.4.

In (3) we assume the *sampling interval* to be one time unit. This is not essential, but makes notation easier.

A pragmatic and useful way to see (3) is to view it as a way of *determining the next output value* given previous observations:

$$y(t) = -a_1 y(t-1) - \ldots - a_n y(t-n) + b_1 u(t-1) + \ldots + b_m u(t-m) \quad (4)$$

For more compact notation we introduce the vectors

$$\theta = [a_1, \ldots, a_n \, b_1, \ldots, b_m]^T \tag{5}$$

$$\varphi(t) = [-y(t-1) \ldots - y(t-n) \; u(t-1) \ldots u(t-m)]^T \tag{6}$$

With these (4) can be rewritten as

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

To emphasize that the calculation of $y(t)$ from past data (4) indeed depends on the parameters in θ, we shall rather call this calculated value $\hat{y}(t|\theta)$ and write

$$\hat{y}(t|\theta) = \varphi^T(t)\theta \tag{7}$$

The Least Squares Method

Now suppose for a given system that we do not know the values of the parameters in θ, but that we have recorded inputs and outputs over a time interval $1 \leq t \leq N$:

$$Z^N = \{u(1), y(1), \ldots, u(N), y(N)\} \tag{8}$$

An obvious approach is then to select θ in (3) through (7) so as to fit the calculated values $\hat{y}(t|\theta)$ as well as possible to the measured outputs by the least squares method:

$$\min_\theta V_N(\theta, Z^N) \tag{9}$$

where

$$\begin{aligned} V_N(\theta, Z^N) &= \frac{1}{N} \sum_{t=1}^{N} (y(t) - \hat{y}(t|\theta))^2 = \\ &= \frac{1}{N} \sum_{t=1}^{N} (y(t) - \varphi^T(t)\theta)^2 \end{aligned} \tag{10}$$

We shall denote the value of θ that minimizes (9) by $\hat{\theta}_N$:

$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta, Z^N) \tag{11}$$

("arg min" means the minimizing argument, i.e., that value of θ which minimizes V_N.)

Since V_N is quadratic in θ, we can find the minimum value easily by setting the derivative to zero:

$$0 = \frac{d}{d\theta} V_N(\theta, Z^N) = \frac{2}{N} \sum_{t=1}^{N} \varphi(t)(y(t) - \varphi^T(t)\theta)$$

which gives

$$\sum_{t=1}^{N} \varphi(t) y(t) = \sum_{t=1}^{N} \varphi(t)\varphi^T(t)\theta \tag{12}$$

or

$$\hat{\theta}_N = \left[\sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right]^{-1} \sum_{t=1}^{N} \varphi(t) y(t) \tag{13}$$

Once the vectors $\varphi(t)$ are defined, the solution can easily be found by modern numerical software, such as MATLAB.

Example 3 *First order difference equation*

Consider the simple model

$$y(t) + ay(t-1) = bu(t-1).$$

This gives us the estimate according to (5), (6) and (13)

$$\begin{bmatrix} \hat{a}_N \\ \hat{b}_N \end{bmatrix} = \begin{bmatrix} \sum y^2(t-1) & -\sum y(t-1)u(t-1) \\ -\sum y(t-1)u(t-1) & \sum u^2(t-1) \end{bmatrix}^{-1} \begin{bmatrix} -\sum y(t)y(t-1) \\ \sum y(t)u(t-1) \end{bmatrix}$$

All sums are from $t = 1$ to $t = N$. A typical convention is to take values outside the measured range to be zero. In this case we would thus take $y(0) = 0$.

The simple model (3) and the well known least squares method (13) form the archetype of System Identification. Not only that – they also give the most commonly used parametric identification method and are much more versatile than perhaps perceived at first sight. In particular one should realize that (3) can directly be extended to several different inputs (this just calls for a redefinition of $\varphi(t)$ in (6)) and that the inputs and outputs do not have to be the raw measurements. On the contrary – it is often most important to think over the physics of the application and come up with suitable inputs and outputs for (3), formed from the actual measurements.

Example 4 An immersion heater

Consider a process consisting of an immersion heater immersed in a cooling liquid. We measure:

- $v(t)$*: The voltage applied to the heater*
- $r(t)$*: The temperature of the liquid*
- $y(t)$*: The temperature of the heater coil surface*

Suppose we need a model for how $y(t)$ depends on $r(t)$ and $v(t)$. Some simple considerations based on common sense and high school physics ("Semi-physical modeling") reveal the following:

- *The change in temperature of the heater coil over one sample is proportional to the electrical power in it (the inflow power) minus the heat loss to the liquid*
- *The electrical power is proportional to $v^2(t)$*
- *The heat loss is proportional to $y(t) - r(t)$*

This suggests the model

$$y(t) = y(t-1) + \alpha v^2(t-1) - \beta(y(t-1) - r(t-1))$$

which fits into the form

$$y(t) + \theta_1 y(t-1) = \theta_2 v^2(t-1) + \theta_3 r(t-1))$$

This is a two input (v^2 and r) and one output model, and corresponds to choosing

$$\varphi(t) = [-y(t-1) \quad v^2(t-1) \quad r(t-1)]^T$$

in (7).

Some Statistical Remarks

Model structures, such as (7) that are linear in θ are known in statistics as *linear regression* and the vector $\varphi(t)$ is called the *regression vector* (its components are the *regressors*). “Regress” here alludes to the fact that we try to calculate (or describe) $y(t)$ by “going back” to $\varphi(t)$. Models such as (3) where the regression vector – $\varphi(t)$ – contains old values of the variable to be explained – $y(t)$ – are then partly *auto-regressions*. For that reason the model structure (3) has the standard name ARX-model (Auto-regression with extra inputs).

There is a rich statistical literature on the properties of the estimate $\hat{\theta}_N$ under varying assumptions. See, e.g. [Draper and Smith, 1981]. So far we have just viewed (9) and (10) as “curve-fitting”. In Section 5.2 we shall deal with a more comprehensive statistical discussion, which includes the ARX model as a special case. Some direct calculations will be done in the following subsection.

Model Quality and Experiment Design

Let us consider the simplest special case, that of a Finite Impulse Response (FIR) model. That is obtained from (3) by taking $n = 0$:

$$y(t) = b_1 u(t-1) + \dots b_m u(t-m) \tag{14}$$

Suppose that the observed data really have been generated by a similar mechanism

$$y(t) = b_1^0 u(t-1) + \dots b_m^0 u(t-m) + e(t) \tag{15}$$

where $e(t)$ is a white noise sequence with variance λ, but otherwise unknown. (That is, $e(t)$ can be described as a sequence of independent random variables with zero mean values and variances λ.) Analogous to (7), we can write this as

$$y(t) = \varphi^T(t)\theta_0 + e(t) \tag{16}$$

We can now replace $y(t)$ in (13) by the above expression, and obtain

$$\begin{aligned}\hat{\theta}_N &= \left[\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\right]^{-1} \sum_{t=1}^{N} \varphi(t)y(t) \\ &= \left[\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\right]^{-1} \left[\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\theta_0 + \sum_{t=1}^{N} \varphi(t)e(t)\right]\end{aligned}$$

or

$$\tilde{\theta}_N = \hat{\theta}_N - \theta_0 = \left[\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\right]^{-1} \sum_{t=1}^{N} \varphi(t)e(t) \tag{17}$$

Suppose that the input u is independent of the noise e. Then φ and e are independent in this expression, so it is easy to see that $E\tilde{\theta}_N = 0$, since

e has zero mean. The estimate is consequently *unbiased.* Here E denotes *mathematical expectation.*

We can also form the expectation of $\tilde{\theta}_N \tilde{\theta}_N^T$, i.e., the covariance matrix of the parameter error. Denote the matrix within brackets by R_N. Take expectation with respect to the white noise e. Then R_N is a deterministic matrix and we have

$$P_N = E\tilde{\theta}_N \tilde{\theta}_N^T = R_N^{-1} \sum_{t,s=1}^{N} \varphi(t)\varphi^T(s) Ee(t)e(s) R_N^{-1} = \lambda R_N^{-1} \tag{18}$$

since the double sum collapses to λR_N.

We have thus computed the covariance matrix of the estimate $\hat{\theta}_N$. It is determined entirely by the input properties and the noise level. Moreover define

$$\bar{R} = \lim_{N\to\infty} \frac{1}{N} R_N \tag{19}$$

This will be the *covariance matrix* of the input, i.e. the $i-j$-element of $\bar{R}$ is $R_{uu}(i-j) = \mathrm{E}u(t+i)u(t+j)$.

If the matrix $\bar{R}$ is non-singular, we find that the covariance matrix of the parameter estimate is approximately (and the approximation improves as $N \to \infty$)

$$P_N = \frac{\lambda}{N} \bar{R}^{-1} \tag{20}$$

A number of things follow from this. All of them are typical of the general properties to be described in Section 5.2:

- The covariance decays like $1/N$, so the parameters approach the limiting value at the rate $1/\sqrt{N}$.
- The covariance is proportional to the Noise-To-Signal ratio. That is, it is proportional to the noise variance and inversely proportional to the input power.

- The covariance does not depend on the input's or noise's signal shapes, only on their variance/covariance properties.
- Experiment design, i.e., the selection of the input u, aims at making the matrix $\bar{R}^{-1}$ "as small as possible". Note that the same $\bar{R}$ can be obtained for many different signals u.

3 Model structures I: Linear Models

3.1 Output error models

Starting from (3) there is actually another, quite different, way to approach the calculation of good values of a_i and b_i from observed data (8).

Equation (3) describes a linear, discrete-time system with transfer function

$$G(z) = \frac{b_1 z^{n-1} + b_2 z^{n-2} + \ldots + b_m z^{n-m}}{z^n + a_1 z^{n-1} + \ldots + a_n} \tag{21}$$

(assuming $n \geq m$)

Here z is the z-transform variable, and one may simply think of the transfer function G as a shorthand notation for the difference equation (3).

We shall here use the shift operator q as an alternative for the variable z in the (21). The shift operator q has the properties

$$qu(t) = u(t+1) \tag{22}$$

(just as multiplying a z-transform by z corresponds to a time shift).

Given only an input sequence

$$\{u(t),\, t = 1, \ldots, N\}$$

we could then calculate the output for system (21) by running u as input to this system:

$$\hat{y}(t|\theta) = G(q)u(t) \tag{23}$$

Example 5 *A first order system Consider the system*

$$y(t+1) + ay(t) = bu(t)$$

The output according to (23) is then obtained as

$$\hat{y}(t|\theta) = \frac{b}{q+a}u(t) = b\sum_{k=1}^{\infty}(-a)^{k-1}u(t-k)$$

or

$$\hat{y}(t+1|\theta) + a\hat{y}(t|\theta) = bu(t) \tag{24}$$

Notice the essential difference between (23) and (7)! In (7) we calculated $\hat{y}(t|\theta)$ using *both* past measured inputs and also *past measured outputs* $y(t-k)$. In (23) $\hat{y}(t|\theta)$ is calculated from *past inputs only.* As soon as we use data from a real system (that does not *exactly obey* (12)) there will always be a difference between these two ways of obtaining the computed output.

Now, we could of course still say that a reasonable estimate of θ is obtained by minimizing the quadratic fit:

$$\hat{\theta}_N = \arg\min_{\theta} \frac{1}{N}\sum_{t=1}^{N}[y(t) - \hat{y}(t|\theta)]^2 \tag{25}$$

even when $\hat{y}(t|\theta)$ is computed according to (23). Such an estimate is often called an *output-error* estimate, since we have formed the fit between a purely simulated output and the measured output. Note that $\hat{y}(t|\theta)$ according to (23) is *not* linear in θ, so the function to be minimized in (25) is not quadratic in θ. Hence some numerical search schemes have to be applied in order to find

$\hat{\theta}_N$ in (25). Most often in practice a Gauss-Newton iterative minimization procedure is used. See Section 5.4.

It follows from the discussion that the estimate obtained by (25) will in general differ from the one from (9). What is the essential difference? To answer that question we will have to discuss various ways of perceiving and describing the disturbances that act on the system.

3.2 Noise Models and Prediction Filters

(Readers who concentrate on the "bottom line" may skip directly to the end of this section.)

A linear, finite-dimensional dynamical system can be described by the equation

$$y(t) = \frac{B(q)}{A(q)}u(t) \tag{26}$$

See (21) – (22). Based on (26) we can predict the next output from *previous measurements* either as in (23)

$$\hat{y}(t|\theta) = \frac{B(q)}{A(q)}u(t) \tag{27}$$

or as in (4), (7):

$$\hat{y}(t|\theta) = (1 - A(q))y(t) + B(q)u(t) \tag{28}$$

Which one shall we choose? We can make the discussion more general by writing for (26)

$$y(t) = G(q, \theta)u(t) \tag{29}$$

to indicate that the transfer function depends on the (numerator and denominator) parameters θ (as in (5)). We can multiply both sides of (29) by an arbitrary stable filter $W(q,\, \theta)$ giving

$$W(q, \theta)y(t) = W(q, \theta)G(q, \theta)u(t) \tag{30}$$

then we can add $y(t)$ to both sides of the equation and rearrange to obtain

$$y(t) = (1 - W(q,\, \theta))y(t) + W(q,\, \theta)G(q,\, \theta)u(t) \tag{31}$$

We assume that the filter W starts with a 1:

$$W(q,\, \theta) = 1 + w_1 q^{-1} + w_2 q^{-1} + w_2 q^{-2} + \ldots$$

so that $1 - W(q,\, \theta)$ actually contains a delay. We thus obtain the predictor

$$\hat{y}(t|\theta) = (1 - W(q,\, \theta))y(t) + W(q,\, \theta)G(q,\, \theta)u(t) \tag{32}$$

Note that this formulation is now similar to that of (28).

We see that the method used in (27) corresponds to the choice $W(q,\, \theta) \equiv 1$, while the procedure in (28) is obtained for $W(q,\, \theta) = A(q)$.

Now, does the predictor (32) depend on the filter $W(q,\, \theta)$? Well, if the input – output data are exactly described by (29) and we know all relevant initial conditions the predictor (32) *produces identical predictions* $\hat{y}(t|\theta)$, regardless of the choice of stable filters $W(q, \theta)$.

To bring out the relevant differences, we must accept the fact that there will always be disturbances and noise that affect the system, so instead of (29) we have a true system that relates the inputs and outputs by

$$y(t) = G_0(q)u(t) + v(t) \tag{33}$$

for some disturbance sequence $\{v(t)\}$. So (32) becomes,

$$\hat{y}(t|\theta) = \{(1 - W(q, \theta))G_0(q) + W(q, \theta)G(q, \theta)\}u(t) + (1 - W(q, \theta))v(t)$$

Now, assume that there exists a value θ_0, such that $G(q, \theta_0) = G_0(q)$. Then the error of the above prediction becomes

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta_0) = W(q, \theta)v(t) \tag{34}$$

To make this error as small as possible we must thus match the choice of the filter $W(q, \theta_0)$ to the properties of the noise $v(t)$. Suppose $v(t)$ can be described as filtered white noise

$$v(t) = H_0(q)e(t) \tag{35}$$

where $e(t)$ is a sequence of independent random variables. Here we assume $H_0(q)$ to be normalized, so that to $H_0(q) = 1 + h_1 q^{-1} + \ldots$. Then, it is easy to see from (34) that no filter $W(q, \theta_0)$ can do better than $1/H_0(q)$, since this makes the prediction error $\varepsilon(t, \theta_0)$ equal to the white noise source $e(t)$.

All this leads to the following summarizing conclusion (which is the only thing one needs to understand from this section).

1. In order to distinguish between different predictors, one has to introduce descriptions of the disturbances that act on the process

2. If the input–output description is assumed to be

   $$y(t) = G(q)u(t) + H(q)e(t) \tag{36}$$

   where $\{e(t)\}$ is a white noise source, then the natural predictor of $y(t)$ given previous observations of inputs and outputs will be

   $$\hat{y}(t|\theta) = [1 - H^{-1}(q)]y(t) + H^{-1}(q)G(q)u(t) \tag{37}$$

   This predictor gives the smallest possible error – if $\{e(t)\}$ indeed is white noise.

3. Since the dynamics $G(q)$ and the noise model $H(q)$ are typically unknown, we will have to work with a parameterized description

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \tag{38}$$

The corresponding predictor in then obtained from (37):

$$\hat{y}(t|\theta) = [I - H^{-1}(q, \theta)]y(t) + H^{-1}(q, \theta)G(q, \theta)u(t) \tag{39}$$

We may now return to the question we posed at the end of Section 3.1. What is the practical difference between minimizing (10) and (25)? Comparing (23) with (29) we see that this predictor corresponds to the assumption that $H = 1$, i.e., that white measurement noise is added to the output. This also means that minimizing the corresponding prediction error – (25) – will give a clearly better estimate, if this assumptions more or less correct.

3.3 Linear Black-Box Model Parameterization

The model parameterization (38) contains a large number of much-used special cases. We have already seen that the *ARX-model* (3) corresponds to

$$G(q, \theta) = \frac{B(q)}{A(q)} \qquad H(q, \theta) = \frac{1}{A(q)} \tag{40}$$

That is, we assume the system (plant) dynamics and the noise model to have common poles, and no numerator dynamics for the noise. Its main feature is that the predictor $\hat{y}(t|\theta)$ will be linear in the parameters θ according to (11) or (7).

We can make (40) more general by allowing also numerator dynamics. We then obtain the parameterization

$$G(q, \theta) = \frac{B(q)}{A(q)} \qquad H(q, \theta) = \frac{C(q)}{A(q)} \tag{41}$$

The effect of the numerator C is that the current predicted value of y will depend upon previous predicted values, not just measured values. This is known as a *ARMAX model*, since the $C(q)$-term makes the noise model a *Moving Average* of a white noise source. Also, (41) assumes that the dynamics and the noise model have common poles, and is therefore particularly suited for the case where the disturbances enter together with the input, "early in the process" so to speak.

The *output error* (OE) model we considered in (23) corresponds to the case

$$G(q,\theta) = \frac{B(q)}{F(q)} \qquad H(q,\theta) = 1 \tag{42}$$

(We use F in the denominator to distinguish the case from (40).) Its unique feature is that the prediction is based on *past inputs only*. It also concentrates on the model dynamics and does not bother about describing the noise.

We can also generalize this model by allowing a general noise model

$$G(q,\theta) = \frac{B(q)}{F(q)}, \qquad H(q,\theta) = \frac{C(q)}{D(q)} \tag{43}$$

This particular model parameterization is known as the *Box-Jenkins* (BJ) model, since it as suggested in the well known book by [Box and Jenkins, 1970]. It differs from the ARMAX-model (42) is that it assigns different dynamics (poles) to the noise characteristics from the input - output properties. It is thus better suited for cases where the noise enters "late in the process", such as measurement noise. See figure 3!

One might wonder why we need all these different model parameterizations. As has been mentioned in the text each has its advantages, which can be summarized as follows

ARX: Gives a linear regression. Very simple to estimate θ

ARMAX: Gives reasonable flexibility to the noise description. Assumes that noise enters like the inputs

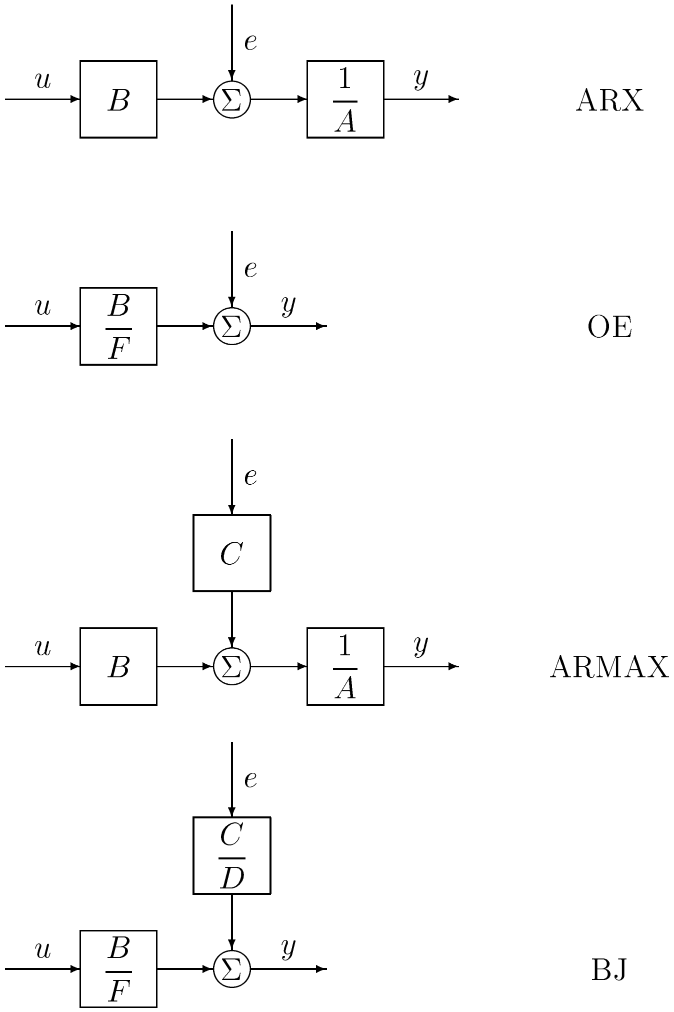

Figure 3: Linear Black-Box Model structures.

OE: Concentrates on the input–output dynamics

BJ: Very flexible. Assumes no common characteristics between noise and input–output behavior.

3.4 Physically parameterized linear models

So far we have treated the parameters θ only as vehicles to give reasonable flexibility to the transfer functions in the general linear model (38). This model can also be arrived at from other considerations.

Consider a continuous time state space model

$$\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) \tag{44a}$$

$$y(t) = C(\theta)x(t) + v(t) \tag{44b}$$

Here $x(t)$ is the state vector and typically consists of physical variables (such as positions and velocities etc). The state space matrices A, B and C are parameterized by the parameter vector θ, reflecting the physical insight we have into the process. The parameters could be physical constants (resistance, heat transfer coefficients, aerodynamical derivatives etc) whose values are not known. They could also reflect other types of insights into the system's properties.

Example 8.4 *An electric motor*

Consider an electric motor with the input u being the applied voltage and the output y being the angular position of the motor shaft.

A first, but reasonable approximation of the motor's dynamics is as a first order system from voltage to angular velocity, followed by an integrator:

$$G(s) = \frac{b}{s(s+a)}$$

If we select the state variables

$$x(t) = \begin{pmatrix} y(t) \\ \dot{y}(t) \end{pmatrix}$$

we obtain the state space form

$$\begin{aligned} \dot{x} &= \begin{pmatrix} 0 & 1 \\ 0 & -a \end{pmatrix} x + \begin{pmatrix} 0 \\ b \end{pmatrix} u \\ y &= \begin{pmatrix} 1 & 0 \end{pmatrix} x + v \end{aligned} \tag{45}$$

where v denotes disturbances and noise. In this case we thus have

$$\begin{aligned} \theta &= \begin{pmatrix} a \\ b \end{pmatrix} \\ A(\theta) &= \begin{pmatrix} 0 & 1 \\ 0 & -a \end{pmatrix} \qquad B(\theta) = \begin{pmatrix} 0 \\ b \end{pmatrix} \\ C &= \begin{pmatrix} 1 & 0 \end{pmatrix} \end{aligned} \tag{46}$$

The parameterization reflects our insight that the system contains an integration, but is in this case not directly derived from detailed physical modeling. Basic physical laws would in this case have given us how θ depends on physical constants, such as resistance of the wiring, amount of inertia, friction coefficients and magnetic field constants. □

Now, how do we fit a continuous-time model (44a) to sampled observed data? If the input $u(t)$ has been piecewise constant over the sampling interval

$$u(t) = u(kT) \qquad kT \leq t < (k+1)T$$

then the states, inputs and outputs at the sampling instants will be represented by the discrete time model

$$\begin{aligned} x((k+1)T) &= \bar{A}(\theta)x(kT) + \bar{B}(\theta)u(kT) \\ y(kT) &= C(\theta)x(kT) + v(kT) \end{aligned} \tag{47}$$

where

$$\bar{A}(\theta) = e^{A(\theta)T}, \quad \bar{B}(\theta) = \int_0^T e^{A(\theta)\tau} B(\theta) d\tau \tag{48}$$

This follows from solving (44) over one sampling period. We could also further model the added noise term $v(kT)$ and represent the system in the innovations form

$$\begin{aligned} \bar{x}((k+1)T) &= \bar{A}(\theta)\bar{x}(kT) + \bar{B}(\theta)u(kT) + \bar{K}(\theta)e(kT) \\ y(kT) &= C(\theta)\bar{x}(kT) + e(kT) \end{aligned} \tag{49}$$

where $\{e(kT)\}$ is white noise. The step from (47) to (49) is really a standard Kalman filter step: $\bar{x}$ will be the one-step ahead predicted Kalman states. A pragmatic way to think about it is as follows: In (47) the term $v(kT)$ may not be white noise. If it is colored we may separate out that part of $v(kT)$ that cannot be predicted from past values. Denote this part by $e(kT)$: it will be the *innovation*. The other part of $v(kT)$ – the one that can be predicted – can then be described as a combination of earlier innovations, $e(\ell T)$ $\ell < k$. Its effect on $y(kT)$ can then be described via the states, by changing them from x to $\bar{x}$, where $\bar{x}$ contains additional states associated with getting $v(kT)$ from $e(\ell T),\ k \leq \ell$.

Now (49) can be written in input – output from as (let $T = 1$)

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t) \tag{50}$$

with

$$\begin{aligned} G(q,\theta) &= C(\theta)(qI - \bar{A}(\theta))^{-1}\bar{B}(\theta) \\ H(q,\theta) &= I + C(\theta)(qI - \bar{A}(\theta))^{-1}\bar{K}(\theta) \end{aligned} \tag{51}$$

We are thus back at the basic linear model (38). The parameterization of G and H in terms of θ is however more complicated than the ones we discussed in Section 3.3.

The general estimation techniques, model properties (including the characterization (85)), algorithms, etc., apply exactly as described in Section 5.

From these examples it is also quite clear that non-linear models with unknown parameters can be approached in the same way. We would then typically arrive at a a structure

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), \theta) \\ y(t) &= h(x(t), u(t), \theta) + v(t) \end{aligned} \tag{52}$$

In this model, all noise effects are collected as additive output disturbances $v(t)$ which is a restriction, but also a very helpful simplification. If we define $\hat{y}(t|\theta)$ as the simulated output response to (52), for a given input, ignoring the noise $v(t)$, everything that was said in Section 5 about parameter estimation, model properties, etc. is still applicable.

4 Model Structures II: Non-linear Black Box Models

In this section we shall describe the basic ideas behind model structures that have the capability to cover any non-linear mapping from past data to the predicted value of $y(t)$. Recall that we defined a general model structure as a parameterized mapping in (1):

$$\hat{y}(t|\theta) = g(\theta, Z^{t-1}) \tag{53}$$

We shall consequently allow quite general non-linear mappings g. This section will deal with some general principles for how to construct such mappings, and will cover Artificial Neural Networks as a special case. See [Sjöberg et al., 1995] and [Juditsky et al., 1995] for recent and more comprehensive surveys.

4.1 Non-Linear Black-Box Structures

Now, the model structure family (53) is really too general, and it turns out to be useful to write g as a concatenation of two mappings: one that takes the increasing number of past observations Z^{t-1} and maps them into a finite dimensional vector $\varphi(t)$ of fixed dimension and one that takes this vector to the space of the outputs:

$$\hat{y}(t|\theta) = g(\theta, Z^{t-1}) = g(\varphi(t), \theta) \tag{54}$$

where

$$\varphi(t) = \varphi(Z^{t-1}) \tag{55}$$

Let the dimension of φ be d. As before, we shall call this vector the *regression vector* and its components will be referred to as the *regressors*. We also allow the more general case that the formation of the regressors is itself parameterized:

$$\varphi(t) = \varphi(Z^{t-1}, \eta) \tag{56}$$

which we for short write $\varphi(t, \eta)$. For simplicity, the extra argument η will however be used explicitly only when essential for the discussion.

The choice of the non-linear mapping in (53) has thus been reduced to two partial problems for dynamical systems:

1. How to choose the non-linear mapping $g(\varphi)$ from the regressor space to the output space (*i.e.*, from R^d to R^p).

2. How to choose the regressors $\varphi(t)$ from past inputs and outputs.

The second problem is the same for all dynamical systems, and it turns out that the most useful choices of regression vectors are to let them contain past inputs and outputs, and possibly also past predicted/simulated outputs. The regression vector will thus be of the character (6). We now turn to the first problem.

4.2 Non-Linear Mappings: Possibilities

Now let us turn to the nonlinear mapping

$$g(\varphi, \theta) \tag{57}$$

which for any given θ maps from R^d to R^p. For most of the discussion we will use $p = 1$, i.e., the output is scalar-valued. At this point it does not matter how the regression vector $\varphi = (\varphi_1, \ldots, \varphi_d)^T$ was constructed. It is just a vector that lives in R^d.

It is natural to think of the parameterized function family as function expansions:

$$g(\varphi, \theta) = \sum \alpha_k g_k(\varphi) \ . \tag{58}$$

We refer to g_k as *basis functions*, since the role they play in (58) is similar to that of a functional space basis. In some particular situations, they do constitute a functional basis. Typical examples are wavelet bases (see below).

We are going to show that expansion (58) with different basis functions, plays the role of a unified framework for investigating most known nonlinear black-box model structures.

Now, the key question is: How to choose the basis functions g_k? The following facts are essential to understand the connections between most known nonlinear black-box model structures:

- All the g_k are formed from one "mother basis function", that we generically denote by $\kappa(x)$.
- This function $\kappa(x)$ is a function of a scalar variable x.
- Typically g_k are dilated (scaled) and translated versions of κ. For the scalar case $d = 1$ we may write

  $$g_k(\varphi) = g_k(\varphi, \beta_k, \gamma_k) = \kappa(\beta_k(\varphi - \gamma_k)) \tag{59}$$

  We thus use β_k to denote the dilation parameters and γ_k to denote translation parameters.

A Scalar Example: Fourier Series Take $\kappa(x) = \cos(x)$. Then (58),(59) will be the Fourier series expansion, with β_k as the frequencies and γ_k as the phases.

Another Scalar Example: Piecewise Constant Functions Take κ as the unit interval indicator function:

$$\kappa(x) = \begin{cases} 1 \text{ for } 0 \leq x < 1 \\ 0 \text{ else} \end{cases} \tag{60}$$

and take, for example, $\gamma_k = k$, $\beta_k = 1/\Delta$ and $\alpha_k = f(k\Delta)$. Then (58), (59) gives a piecewise constant approximation of any function f. Clearly we would have obtained a quite similar result by a smooth version of the indicator function, e.g., the Gaussian bell:

$$\kappa(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{61}$$

A Variant of the Piece-wise constant case Take κ to be the unit step function

$$\kappa(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases} \tag{62}$$

We then just have a variant of (60), since the indicator function can be obtained as the difference of two steps. A smooth version of the step, like the *sigmoid* function

$$\kappa(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{63}$$

will of course give quite similar results.

Classification of single-variable basis functions

Two classes of single-variable basis functions can be distinguished depending on their nature:

- *Local Basis Functions* are functions having their gradient with bounded support, or at least vanishing rapidly at infinity. Loosely speaking, their variations are concentrated to some interval.
- *Global Basis Functions* are functions having infinitely spreading (bounded or not) gradient.

Clearly the Fourier series is an example of a global basis function, while (60), (61), (62) and (63) are all local functions.

Construction of multi-variable basis functions

In multi-dimensional case ($d > 1$), g_k are multi-variable functions. In practice they are often constructed from the single-variable function κ in some simple manner. Let us recall the three most often used methods for constructing multi-variable basis functions from single-variable basis functions.

1. **Tensor product.** Given d single-variable functions of the different components φ_j of a d-dimensional vector φ, $h_1(\varphi_1), \ldots, h_d(\varphi_d)$ (identical or not). The tensor product construction of the corresponding function from R^d is then given by their product. In the present case this means that the basis functions are constructed from the scalar function κ as

$$g_k(\varphi) = \prod_{j=1}^{d} \kappa(\beta_k^j(\varphi_j - \gamma_k^j)) \tag{64}$$

2. **Radial construction.** For any single-variable function κ the radial construction of multi-variable basis function of $\varphi \in R^d$, has the form

$$g_k(\varphi) = g_k(\varphi, \beta_k, \gamma_k) = \kappa(||\varphi - \gamma_k||_{\beta_k}) \tag{65}$$

where $\| \cdot \|_{\beta_k}$ denotes any chosen norm on the space of the regression vector φ. The norm could typically be a quadratic norm

$$\|\varphi\|_{\beta_k}^2 = \varphi^T \beta_k \varphi \tag{66}$$

with β_k as a possibly k-dependent positive definite matrix of dilation (scale) parameters. In simple cases β_k may be just scaled versions of the identity matrix.

3. **Ridge construction.** Let κ be any single-variable function. Then for all $\beta_k \in R^d$, $\gamma_k \in R$, a *ridge* function is given by

$$g_k(\varphi) = g_k(\varphi, \beta_k, \gamma_k) = \kappa(\beta_k^T \varphi + \gamma_k), \ \varphi \in R^d \tag{67}$$

The ridge function is thus constant for all φ in the sub-space $\{\varphi \in R^d \ : \ \beta_k^T \varphi = \text{ constant}\}$. As a consequence, even if the mother basis function κ has local support, the basis functions g_k will have unbounded support in this subspace. The resulting basis could be said to be *semi-global*, but the term *ridge function* is more precise.

Approximation Issues

For any of the described choices the resulting model becomes

$$g(\varphi, \theta) = \sum_{k=1}^{n} \alpha_k \kappa(\beta_k(\varphi - \gamma_k)) \tag{68}$$

with the different exact interpretations of the argument $\beta_k(\varphi - \gamma_k)$ just discussed. The expansion is entirely determined by

- the scalar valued function $\kappa(x)$ of a scalar variable x
- the way the basis functions are expanded to depend on a vector φ.

The parameterization in terms of θ can be characterized by three types of parameters:

- The *coordinates* α
- The *scale* or *dilation* parameters β
- The *location* parameters γ

A key issue is how well the function expansion is capable of approximating any possible "true system" $g_0(\varphi)$. There is a rather extensive literature on this subject. For an identification oriented survey, see, e.g., [Juditsky et al., 1995].

The bottom line is easy: *For almost any choice of $\kappa(x)$ – except being a polynomial – the expansion (68) can approximate any "reasonable" function $g_0(\varphi)$ arbitrarily well for sufficiently large n.*

It is not difficult to understand this. It is sufficient to check that the delta function – or the indicator function for arbitrarily small areas – can be arbitrarily well approximated within the expansion. Then clearly all reasonable functions can also be approximated. For local κ with radial construction this is immediate: Indeed by scaling and location an arbitrarily small indicator function can be places anywhere. For the ridge construction one needs to show that a number of hyperplanes defined by β and γ can be placed and intersect so that any small area in R^d is cut out.

The question of how *efficient* the expansion is, i.e., how large n is required to achieve a certain degree of approximation is more difficult, and has no general answer. We may point to the following aspects:

- If the scale and location parameters β and γ are allowed to depend on the function g_0 to be approximated, then the number of terms n required for a certain degree of approximation is much less than if $\beta_k, \gamma_k,\ k = 1, \ldots$ is an *a priori* fixed sequence.
- For the local, radial approach the number of terms required to achieve a certain degree of approximation δ of an p times differentiable function is proportional to

  $$n \sim \frac{1}{\delta^{(d/p)}} \tag{69}$$

  It thus increases exponentially with the number of regressors. This is often referred to as *the curse of dimensionality.*

Connection to “Named Structures”

Here we briefly review some popular structures, other structures related to interpolation techniques are discussed in [Sjöberg et al., 1995, Juditsky et al., 1995].

Wavelets The local approach corresponding to (58,65) has direct connections to wavelet networks and wavelet transforms. The exact relationships are discussed in [Sjöberg et al., 1995]. Loosely, we note that via the dilation parameters in ρ_k we can work with different scales simultaneously to pick up both local and not-so-local variations. With appropriate translations and dilations of a single suitably chosen function κ (the “mother wavelet”), we can make the expansion (58) orthonormal. This is discussed extensively in [Juditsky et al., 1995].

Wavelet and Radial Basis Networks. The choice (61) without any orthogonalization is found in both wavelet networks, [Zhang and Benveniste, 1992] and radial basis neural networks [Poggio and Girosi, 1990].

Neural Networks The ridge choice (67) with κ given by (63) gives a much-used neural network structure, viz. the *one hidden layer feedforward sigmoidal net.*

Hinging Hyperplanes If instead of using the sigmoid σ function we choose “V-shaped” functions (in the form of a higher-dimensional “open book”) Breiman’s *hinging hyperplane* structure is obtained, [Breiman, 1993].

Nearest Neighbors or Interpolation By selecting κ as in (60) and the location and scale vector $\_k, \gamma_k$ in the structure (65), such that exactly one observation falls into each “cube”, the nearest neighbor model is obtained: just load the input-output record into a table, and, for a given φ, pick the pair $(\widehat{y}, \widehat{\varphi})$ for $\widehat{\varphi}$ closest to the given φ, $\widehat{y}$ is the desired output estimate. If one replaces (60) by a smoother function and allow some overlapping of the basis functions, we get interpolation type techniques such as kernel estimators.

Fuzzy Models Also so called *fuzzy models* based on fuzzy set membership belong to the model structures of the class (58). The basis functions g_k then are constructed from the fuzzy set membership functions and the inference rules using the tensor approach (64). The exact relationship is described in [Sjöberg et al., 1995].

4.3 Estimating Non-linear Black Box Models

The model structure is determined by the following choices

- The regression vector (typically built up from past inputs and outputs)
- The basic function κ (local) or σ (ridge)
- The number of elements (nodes) in the expansion (58).

Once these choices have been made $\hat{y}(t|\theta) = g(\varphi(t), \theta)$ is a well defined function of past data and the parameters θ. The parameters are made up of coordinates in the expansion (58), and from location and scale parameters in the different basis functions.

All the algorithms and analytical results of Section 5 can thus be applied. For Neural Network applications these are also the typical estimation algorithms used, often complemented with *regularization*, which means that a term is added to the criterion (74), that penalizes the norm of θ. This will reduce the variance of the model, in that "spurious" parameters are not allowed to take on large, and mostly random values. See e.g. [Sjöberg et al., 1995].

For wavelet applications it is common to distinguish between those parameters that enter linearly in $\hat{y}(t|\theta)$ (i.e. the coordinates in the function expansion) and those that enter non-linearly (i.e. the location and scale parameters). Often the latter are seeded to fixed values and the coordinates are estimated by the linear least squares method. Basis functions that give a small contribution to the fit (corresponding to non-useful values of the scale and location parameters) can them be trimmed away ("pruning" or "shrinking").

5 General Parameter Estimation Techniques

In this section we shall deal with issues that are independent of model structure. Principles and algorithms for fitting models to data, as well as the general properties of the estimated models are all model-structure independent and equally well applicable to, say, ARMAX models and Neural Network models.

The section is organized as follows. In Section 5.1 the general principles for parameter estimation are outlined. Sections 5.2 and 5.3 deal with the asymptotic (in the number of observed data) properties of the models, while algorithms are described in Section 5.4.

5.1 Fitting Models to Data

In Section 2 we showed one way to parameterize descriptions of dynamical systems. There are many other possibilities and we shall spend a fair amount of this contribution to discuss the different choices and approaches. *This is actually the key problem in system identification.* No matter how the problem is approached, the bottom line is that such a model parameterization leads to a predictor

$$\hat{y}(t|\theta) = g(\theta, Z^{t-1}) \tag{70}$$

that depends on the unknown parameter vector and past data Z^{t-1} (see (8). This predictor can be linear in y and u. This in turn contains several special cases both in terms of black-box models and physically parameterized ones, as was discussed in Sections 3 and 3.4, respectively. The predictor could also be of general, non-linear nature, as was discussed in Section 4.

In any case *we now need a method to determine a good value of θ,* based on the information in an observed, sampled data set (8). It suggests itself that the basic least-squares like approach (9) through (11) still is a natural approach, even when the predictor $\hat{y}(t|\theta)$ is a more general function of θ.

A procedure with some more degrees of freedom is the following one

1. From observed data and the predictor $\hat{y}(t|\theta)$ form the sequence of prediction errors,

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta), \quad t = 1, 2, \ldots N \tag{71}$$

2. Possibly filter the prediction errors through a linear filter $L(q)$,

$$\varepsilon_F(t,\theta) = L(q)\varepsilon(t,\theta) \tag{72}$$

   so as to enhance or depress interesting or unimportant frequency bands in the signals.

3. Choose a scalar valued, positive function $\ell(\cdot)$ so as to measure the "size" or "norm" of the prediction error:

$$\ell(\varepsilon_F(t,\theta)) \tag{73}$$

4. Minimize the sum of these norms:

$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta, Z^N) \tag{74}$$

   where

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{t=1}^{N} \ell(\varepsilon_F(t,\theta)) \tag{75}$$

This procedure is natural and pragmatic – we can still think of it as "curve-fitting" between $y(t)$ and $\hat{y}(t|\theta)$. It also has several statistical and information theoretic interpretations. Most importantly, if the noise source in the system (like in (38)) is supposed to be a sequence of independent random variables $\{e(t)\}$ each having a probability density function $f_e(x)$, then (74) becomes the Maximum Likelihood estimate (MLE) if we choose

$$L(q) = 1 \quad \text{and} \quad \ell(\varepsilon) = -\log f_e(\varepsilon) \tag{76}$$

The MLE has several nice statistical features and thus gives a strong "moral support" for using the outlined method. Another pleasing aspect is that the method is independent of the particular model parameterization used (although this will affect the actual minimization procedure). For example, the method of "back propagation" often used in connection with neural network parameterizations amounts to computing $\hat{\theta}_N$ in (74) by a recursive gradient method. We shall deal with these aspects in Section 5.4.

5.2 Model Quality

An essential question is, of course, what properties will the estimate resulting from (74) have. These will naturally depend on the properties of the data record Z^N defined by (8). It is in general a difficult problem to characterize the quality of $\hat{\theta}_N$ exactly. One normally has to be content with the asymptotic properties of $\hat{\theta}_N$ as the number of data, N, tends to infinity.

It is an important aspect of the general identification method (74) that the asymptotic properties of the resulting estimate can be expressed in general terms for arbitrary model parameterizations.

The first basic result is the following one:

$$\hat{\theta}_N \to \theta^* \quad \text{as} \quad N \to \infty \quad \text{where} \tag{77}$$

$$\theta^* = \arg\min_{\theta} E\ell(\varepsilon_F(t,\theta)) \tag{78}$$

That is, as more and more data become available, the estimate converges to that value θ^*, that would minimize the expected value of the "norm" of the filtered prediction errors. This is in a sense *the best possible approximation* of the true system that is available within the model structure. The expectation E in (78) is taken with respect to all random disturbances that affect the data and it also includes averaging over the input properties. This means in particular that θ^* will make $\hat{y}(t|\theta^*)$ a good approximation of $y(t)$ with respect to those aspects of the system that are enhanced by the input signal used.

The second basic result is the following one: If $\{\varepsilon(t,\theta^*)\}$ is approximately white noise, then the covariance matrix of $\hat{\theta}_N$ is approximately given by

$$E(\hat{\theta}_N - \theta^*)(\hat{\theta}_N - \theta^*)^T \sim \frac{\lambda}{N}[E\psi(t)\psi^T(t)]^{-1} \tag{79}$$

where

$$\lambda = E\varepsilon^2(t, \theta^*) \tag{80}$$

$$\psi(t) = \frac{d}{d\theta}\hat{y}(t|\theta)|_{\theta=\theta^*} \tag{81}$$

Think of ψ as the sensitivity derivative of the predictor with respect to the parameters. Then (79) says that the covariance matrix for $\hat{\theta}_N$ is proportional to the inverse of the covariance matrix of this sensitivity derivative. This is a quite natural result.

Note: For all these results, the expectation operator E can, under most general conditions, be replaced by the limit of the sample mean, that is

$$E\psi(t)\psi^T(t) \leftrightarrow \lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N}\psi(t)\psi^T(t) \tag{82}$$

□

The results (77) through (81) are general and hold for all model structures, both linear and non-linear ones, subject only to some regularity and smoothness conditions. They are also fairly natural, and will give the guidelines for all user choices involved in the process of identification. See [Ljung, 1987] for more details around this.

A Characterization of the Limiting Model in a General Class of Linear Models

Let us apply the general limit result (77)-(78) to the linear model structure (38). If we choose a quadratic criterion $\ell(\varepsilon) = \varepsilon^2$ (in the scalar output case) then this result tells us, in the time domain, that the limiting parameter estimate is the one that minimizes the filtered prediction error variance (for the input used during the experiment.) Suppose that the data actually have been generated by

$$y(t) = G_0(q)u(t) + v(t) \tag{83}$$

Let $\Phi_u(\omega)$ be the input spectrum and $\Phi_v(\omega)$ be the spectrum for the additive disturbance v. Then the filtered prediction error can be written

$$\begin{aligned}\varepsilon_F(t,\theta) = \frac{L(q)}{H(q,\theta)}[y(t) - G(q,\theta)u(t)] = \\ \frac{L(q)}{H(q,\theta)}[(G_0(q) - G(q,\theta))u(t) + v(t)]\end{aligned} \tag{84}$$

By Parseval's relation, the prediction error variance can also be written as an integral over the spectrum of the prediction error. This spectrum, in turn, is directly obtained from (84), so the limit estimate θ^* in (78) can also be defined as

$$\begin{aligned}\theta^* = \arg\min_\theta \Bigg[\int_{-\pi}^{\pi} |G_0(e^{i\omega}) - G(e^{i\omega},\theta)|^2 \frac{\Phi_u(\omega)|L(e^{i\omega})|^2}{|H(e^{i\omega},\theta)|^2} d\omega \\ + \int_{-\pi}^{\pi} \Phi_v(\omega)|L(e^{i\omega})|^2/|H(e^{i\omega},\theta)|^2 d\omega\Bigg]\end{aligned} \tag{85}$$

If the noise model $H(q,\theta) = H_*(q)$ does not depend on θ (as in the output error model (42)) the expression (85) thus shows that the resulting model $G(e^{i\omega},\theta^*)$ will give that frequency function in the model set that is closest to the true one, in a quadratic frequency norm with weighting function

$$Q(\omega) = \Phi_u(\omega)|L(e^{i\omega})|^2/|H_*(e^{i\omega})|^2 \tag{86}$$

This shows clearly that the fit can be affected by the choice of prefilter L, the input spectrum Φ_u and the noise model H_*.

5.3 Measures of Model Fit

Some quite general expressions for the expected model fit, that are independent of the model structure, can also be developed.

Let us measure the (average) fit between any model (70) and the true system as

$$\bar{V}(\theta) = E|y(t) - \hat{y}(t|\theta)|^2 \tag{87}$$

Here expectation E is over the data properties (i.e. expectation over "Z^∞" with the notation (8)). Recall that expectation also can be interpreted as sample means as in (82).

Before we continue, let us note the very important aspect that the fit $\bar{V}$ will depend, not only on the model and the true system, *but also on data properties*, like input spectra, possible feedback, etc. We shall say that the fit depends on the *experimental conditions*.

The estimated model parameter $\hat{\theta}_N$ is a random variable, because it is constructed from observed data, that can be described as random variables. To evaluate the model fit, we then take the expectation of $\bar{V}(\hat{\theta}_N)$ with respect to the estimation data. That gives our measure

$$F_N = E\bar{V}(\hat{\theta}_N) \tag{88}$$

In general, the measure F_N depends on a number of things:

- The model structure used.
- The number of data points N.
- The data properties for which the fit $\bar{V}$ is defined.
- The properties of the data used to estimate $\hat{\theta}_N$.

The rather remarkable fact is that if the two last data properties coincide, then, asymptotically in N, (see, e.g., [Ljung, 1987], Chapter 16)

$$F_N \approx \bar{V}_N(\theta^*)(1 + \frac{dim\theta}{N}) \tag{89}$$

Here θ^* is the value that minimizes the expected criterion (78). The notation $dim\theta$ means the number of estimated parameters. The result also assumes that the criterion function $\ell(\varepsilon) = \|\varepsilon\|^2$, and that the model structure is successful in the sense that $\varepsilon_F(t)$ is approximately white noise.

Despite the reservations about the formal validity of (89), it carries a most important conceptual message: If a model is evaluated on a data set with the same properties as the estimation data, then *the fit will not depend on the data properties*, and it will depend on the model structure *only in terms of the number of parameters used and of the best fit offered within the structure.*

The expression can be rewritten as follows. Let $\hat{y}_0(t|t-1)$ denote the "true" one step ahead prediction of $y(t)$, and let

$$W(\theta) = E|\hat{y}_0(t|t-1) - \hat{y}(t|\theta)|^2 \tag{90}$$

and let

$$\lambda = E|y(t) - \hat{y}_0(t|t-1)|^2 \tag{91}$$

Then λ is the *innovations* variance, i.e., that part of $y(t)$ that cannot be predicted from the past. Moreover $W(\theta^*)$ is the *bias error*, i.e. the discrepancy between the true predictor and the best one available in the model structure. Under the same assumptions as above, (89) can be rewritten as

$$F_N \approx \lambda + W(\theta^*) + \lambda\frac{dim\theta}{N} \tag{92}$$

The three terms constituting the model error then have the following interpretations

- λ is the unavoidable error, stemming from the fact that the output cannot be exactly predicted, even with perfect system knowledge.
- $W(\theta^*)$ is the bias error. It depends on the model structure, and on the experimental conditions. It will typically decrease as $dim\theta$ increases.
- The last term is the *variance error*. It is proportional to the number of estimated parameters and inversely proportional to the number of data points. It does not depend on the particular model structure or the experimental conditions.

5.4 Algorithmic Aspects

In this section we shall discuss how to achieve the best fit between observed data and the model, i.e. how to carry out the minimization of (74). For simplicity we here assume a quadratic criterion and set the prefilter L to unity:

$$V_N(\theta) = \frac{1}{2N}\sum_{t=1}^{N} |y(t) - \hat{y}(t|\theta)|^2 \tag{93}$$

No analytic solution to this problem is possible unless the model $\hat{y}(t|\theta)$ is linear in θ, so the minimization has to be done by some numerical search procedure. A classical treatment of the problem of how to minimize the sum of squares is given in [Dennis and Schnabel, 1983].

Most efficient search routines are based on iterative local search in a "downhill" direction from the current point. We then have an iterative scheme of the following kind

$$\hat{\theta}^{(i+1)} = \hat{\theta}^{(i)} - \mu_i R_i^{-1}\hat{g}_i \tag{94}$$

Here $\hat{\theta}^{(i)}$ is the parameter estimate after iteration number i. The search scheme is thus made up of the three entities

- μ_i step size

- $\hat{g}_i$ an estimate of the gradient $V'_N(\hat{\theta}^{(i)})$
- R_i a matrix that modifies the search direction

Search directions

The basis for the local search is the gradient

$$V'_N(\theta) = \frac{dV_N(\theta)}{d\theta} = -\frac{1}{N}\sum_{t=1}^{N}(y(t) - \hat{y}(t|\theta))\psi(t,\theta) \tag{95}$$

where

$$\psi(t,\theta) = \frac{\partial}{\partial\theta}\hat{y}(t|\theta) \tag{96}$$

The gradient ψ is in the general case a matrix with *dim* θ rows and *dim* y columns. It is well known that gradient search for the minimum is inefficient, especially close to the minimum. Then it is optimal to use the *Newton search direction*

$$R^{-1}(\theta)V'_N(\theta) \tag{97}$$

where

$$\begin{aligned} R(\theta) = V''_N(\theta) = \frac{d^2V_N(\theta)}{d\theta^2} &= \frac{1}{N}\sum_{t=1}^{N}\psi(t,\theta)\psi^T(t,\theta) \\ &+ \frac{1}{N}\sum_{t=1}^{N}(y(t) - \hat{y}(t|\theta))\frac{\partial^2}{\partial\theta^2}\hat{y}(t|\theta) \end{aligned} \tag{98}$$

The true Newton direction will thus require that the second derivative

$$\frac{\partial^2}{\partial\theta^2}\hat{y}(t|\theta)$$

be computed. Also, far from the minimum, $R(\theta)$ need not be positive semidefinite. Therefore alternative search directions are more common in practice:

- *Gradient direction.* Simply take

$$R_i = I \tag{99}$$

- *Gauss-Newton direction.* Use

$$R_i = H_i = \frac{1}{N} \sum_{t=1}^{N} \psi(t, \hat{\theta}^{(i)}) \psi^T(t, \hat{\theta}^{(i)}) \tag{100}$$

- *Levenberg-Marquard direction.* Use

$$R_i = H_i + \delta I \tag{101}$$

  where H_i is defined by (100).

- *Conjugate gradient direction.* Construct the Newton direction from a sequence of gradient estimates. Loosely, think of V_N'' as constructed by difference approximation of d gradients. The direction (97) is however constructed directly, without explicitly forming and inverting V''.

It is generally considered, [Dennis and Schnabel, 1983], that the Gauss-Newton search direction is to be preferred. For ill-conditioned problems the Levenberg-Marquard modification is recommended.

Local Minima

A fundamental problem with minimization tasks like (9) is that $V_N(\theta)$ may have several or many local (non-global) minima, where local search algorithms may get caught. There is no easy solution to this problem. It is usually well worth the effort to find a good initial value $\theta^{(0)}$ where to start the iterations. Other than that, only various global search strategies are left, such as random search, random restarts, simulated annealing, and the genetic algorithm.

6 Special Estimation Techniques for Linear Black Box Models

An important feature of a linear, time invariant system is that it is entirely characterized by its *impulse response.* So if we know the system's response to an impulse, we will also know its response to any input. Equivalently, we could study the *frequency response*, which is the Fourier transform of the impulse response.

In this section we shall consider estimation methods for linear systems, that do not use particular model parameterizations. First, in Section 6.1, we shall consider direct methods to determine the impulse response and the frequency response, by simply applying the definitions of these concepts.

In Section 6.2 spectral analysis for frequency function estimation will be discussed. Finally, in Section 6.3 a recent method to estimate general linear systems (of given order, by unspecified structure) will be described.

6.1 Transient and Frequency Analysis

Transient Analysis

The first step in modeling is to decide which quantities and variables are important to describe what happens in the system. A simple and common kind of experiment that shows how and in what time span various variables affect each other is called *step-response analysis* or *transient analysis.* In such experiments the inputs are varied (typically one at a time) as a step: $u(t) = u_0$, $t < t_0$; $u(t) = u_1$, $t \geq t_0$. The other measurable variables in the system are recorded during this time. We thus study the *step response* of the system. An alternative would be to study the impulse response of the system by letting the input be a pulse of short duration. From such measurements, information of the following nature can be found:

1. The variables affected by the input in question. This makes it easier to draw block diagrams for the system and to decide which influences

can be neglected.

2. The time constants of the system. This also allows us to decide which relationships in the model can be described as static (that is, they have significantly faster time constants than the time scale we are working with.

3. The characteristic (oscillatory, poorly damped, monotone, and the like) of the step responses, as well as the levels of static gains. Such information is useful when studying the behavior of the final model in simulation. Good agreement with the measured step responses should give a certain confidence in the model.

Frequency Analysis

If a linear system has the transfer function $G(q)$ and the input is

$$u(t) = u_0 \cos \omega kT, \quad (k-1)T \leq t \leq kT \tag{102}$$

then the output after possible transients have faded away will be

$$y(t) = y_0 \cos(\omega t + \varphi), \quad \text{for} \quad t = T, 2T, 3T, \ldots \tag{103}$$

where

$$y_0 = |G(e^{i\omega T})| \cdot u_0 \tag{104}$$

$$\varphi = \arg G(e^{i\omega T}) \tag{105}$$

If the system is driven by the input (102) for a certain u_0 and ω_1 and we measure y_0 and φ from the output signal, it is possible to determine the complex number $G(e^{i\omega_1 T})$ using (104)–(105). By repeating this procedure for a number of different ω, we can get a good estimate of the frequency function $G(e^{i\omega T})$. This method is called *frequency analysis.* Sometimes it is possible to see or measure u_0, y_0, and φ directly from graphs of the input and output signals. Most of the time, however, there will be noise and irregularities that make it difficult to determine φ directly. A suitable procedure is then to correlate the output with $\cos \omega t$ and $\sin \omega t$.

6.2 Estimating the Frequency Response by Spectral Analysis

Definitions

The *cross spectrum* between two (stationary) signals $u(t)$ and $y(t)$ is defined as the Fourier transform of their cross covariance function, provided this exists:

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) e^{-i\omega\tau} \tag{106}$$

where $R_{yu}(\tau)$ is defined by

$$R_{yu}(\tau) = \mathrm{E} y(t) u(t-\tau) \tag{107}$$

The (auto) *spectrum* $\Phi_u(\omega)$ of a signal u is defined as $\Phi_{uu}(\omega)$, i.e. as its cross spectrum with itself.

The spectrum describes the frequency contents of the signal. The connection to more explicit Fourier techniques is evident by the following relationship

$$\Phi_u(\omega) = \lim_{N\to\infty} \frac{1}{N} |U_N(\omega)|^2 \tag{108}$$

where U_N is the discrete time Fourier transform

$$U_N(\omega) = \sum_{t=1}^{N} u(t) e^{i\omega t} \tag{109}$$

The relationship (108) is shown, e.g. in [Ljung and Glad, 1994].

Consider now the general linear model:

$$y(t) = G(q)u(t) + v(t) \tag{110}$$

It is straightforward to show that the relationships between the spectra and cross spectra of y and u (provided u and v are uncorrelated) is given by

$$\begin{aligned} \Phi_{yu}(\omega) &= G(e^{i\omega})\Phi_u(\omega) &(111)\\ \Phi_y(\omega) &= |G(e^{i\omega})|^2\Phi_u(\omega) + \Phi_v(\omega) &(112) \end{aligned}$$

It is easy to see how the transfer function $G(e^{i\omega})$ and the noise spectrum $\phi_v(\omega)$ can be estimated using these expressions, if only we have a method to estimate cross spectra.

Estimation of Spectra

The spectrum is defined as the Fourier transform of the correlation function. A natural idea would then be to take the transform of the estimate

$$\hat{R}^N_{yu}(\tau) = \frac{1}{N}\sum_{t=1}^{N} y(t)u(t-\tau) \tag{113}$$

That will not work in most cases, though. The reason could be described as follows: The estimate $\hat{R}^N_{yu}(\tau)$ is not reliable for large τ, since it is based on only a few observations. These "bad" estimates are mixed with good ones in the Fourier transform, thus creating an overall bad estimate. It is better to introduce a weighting, so that correlation estimates for large lags τ carry a smaller weight:

$$\hat{\Phi}^N_{yu}(\omega) = \sum_{\ell=-\gamma}^{\gamma} \hat{R}^N_{yu}(\ell) \cdot w_\gamma(\ell) e^{-i\ell\omega} \tag{114}$$

This spectral estimation method is known as the The *Blackman-Tukey approach.* Here $w_\gamma(\ell)$ is a window function that decreases with $|\tau|$. This function controls the trade-off between *frequency resolution* and *variance of the estimate.* A function that gives significant weights to the correlation at large lags will be able to provide finer frequency details (a longer time span is covered). At the same time it will have to use "bad" estimates, so the statistical quality (the variance) is poorer. We shall return to this trade-off in a moment. How should we choose the shape of the window function $w_\gamma(\ell)$? There is no optimal solution to this problem, but the most common window used in spectral analysis is the *Hamming window*:

$$\begin{array}{ll} w_\gamma(k) = \frac{1}{2}(1+\cos\frac{\pi k}{\gamma}) & |k| < \gamma \\ w_\gamma(k) = 0 & |k| \geq \gamma \end{array} \tag{115}$$

From the spectral estimates Φ_u, Φ_y and Φ_{yu} obtained in this way, we can now use (111) to obtain a natural estimate of the frequency function $G(e^{i\omega})$:

$$\hat{G}_N(e^{i\omega}) = \frac{\hat{\Phi}_{yu}^N(\omega)}{\hat{\Phi}_u^N(\omega)} \tag{116}$$

Furthermore, the disturbance spectrum can be estimated from (112) as

$$\hat{\Phi}_v^N(\omega) = \hat{\Phi}_y^N(\omega) - \frac{|\hat{\Phi}_{yu}^N(\omega)|^2}{\hat{\Phi}_u^N(\omega)} \tag{117}$$

To compute these estimates, the following steps are performed:

1. Collect data $y(k)$, $u(k)$ $k = 1, \ldots, N$.

2. Subtract the corresponding sample means form the data. This will avoid bad estimates at very low frequencies.

3. Choose the width of the lag window $w_\gamma(k)$.

4. Compute $\hat{R}_y^N(k)$, $\hat{R}_u^N(k)$, and $\hat{R}_{yu}^N(k)$ for $|k| \leq \gamma$ according to (113).

5. Form the spectral estimates $\hat{\Phi}_y^N(\omega)$, $\hat{\Phi}_u^N(\omega)$, and $\hat{\Phi}_{yu}^N(\omega)$ according to (114) and analogous expressions.

6. Form (116) and possibly also (117).

Quality of the Estimates

The estimates $\hat{G}_N$ and $\hat{\Phi}_w^N$ are formed entirely from estimates of spectra and cross spectra. Their properties will therefore be inherited from the properties of the spectral estimates. For the Hamming window with width γ, it can be shown that the frequency resolution will be about

$$\frac{\pi}{\gamma\sqrt{2}} \qquad \text{radians/time unit} \tag{118}$$

This means that details in the true frequency function that are finer than this expression will be smeared out in the estimate. It is also possible to show that the estimate's variances satisfy

$$\text{Var } \hat{G}_N(i\omega) \approx 0.7 \cdot \frac{\gamma}{N} \cdot \frac{\Phi_v(\omega)}{\Phi_u(\omega)} \tag{119}$$

and

$$\text{Var } \hat{\Phi}_v^N(\omega) \approx 0.7 \cdot \frac{\gamma}{N} \cdot \Phi_v^2(\omega) \tag{120}$$

[Variance" here refers to taking expectation over the noise sequence $v(t)$.] Note that the relative variance in (119) typically increases dramatically as ω tends to the Nyquist frequency. The reason is that $|G(i\omega)|$ typically decays rapidly, while the noise-to-signal ratio $\Phi_v(\omega)/\Phi_u(\omega)$ has a tendency to increase as ω increases. In a Bode diagram the estimates will thus show considerable fluctuations at high frequencies. Moreover, the constant frequency resolution (118) will look thinner and thinner at higher frequencies in a Bode diagram due to the logarithmic frequency scale.

See [Ljung and Glad, 1994] for a more detailed discussion.

Choice of Window Size

The choice of γ is a pure trade-off between frequency resolution and variance (variability). For a spectrum with narrow resonance peaks it is thus necessary to choose a large value of γ and accept a higher variance. For a more flat spectrum, smaller values of γ will do well. In practice a number of different values of γ are tried out. Often we start with a small value of γ and increase it successively until an estimate is found that balances the trade-off between frequency resolution (true details) and variance (random fluctuations). A typical value for spectra without narrow resonances is γ= 20–30.

6.3 Subspace Estimation Techniques for State Space Models

A linear system can always be represented in state space form:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + w(t) \\ y(t) &= Cx(t) + Du(t) + e(t) \end{aligned} \tag{121}$$

We assume that we have no insight into the particular structure, and we would just estimate any matrices A, B, C, and D, that give a good description of the input-output behavior of the system. This is not without problems, among other things because there are an infinite number of such matrices that describe the same system (the similarity transforms). The coordinate basis of the state-space realization thus needs to be fixed.

Let us for a moment assume that not only are u and y measured, but also the sequence of state vectors x. This would, by the way, fix the state-space realization coordinate basis. Now, with known u, y and x, the model (121) becomes a linear regression: the unknown parameters, all of the matrix entries in all the matrices, mix with measured signals in linear combinations. To see this clearly, let

$$\begin{aligned} Y(t) &= \begin{pmatrix} x(t+1) \\ y(t) \end{pmatrix} \\ \Theta &= \begin{pmatrix} A & B \\ C & D \end{pmatrix} \\ \Phi(t) &= \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \end{aligned}$$

$$E(t) = \begin{pmatrix} w(t) \\ e(t) \end{pmatrix}$$

Then, (121) can be rewritten as

$$Y(t) = \Theta\Phi(t) + E(t) \tag{122}$$

From this all the matrix elements in Θ can be estimated by the simple least squares method, as described in Section 2. The covariance matrix for $E(t)$ can also be estimated easily as the sample sum of the model residuals. That will give the covariance matrices for w and e, as well as the cross covariance matrix between w and e. These matrices will, among other things, allow us to compute the Kalman filter for (121). Note that all of the above holds without changes for multivariable systems, i.e., when the output and input signals are vectors.

The only remaining problem is where to get the state vector sequence x from. It has long been known, e.g., [Rissanen, 1974], [Akaike, 1974b], that all state vectors $x(t)$ that can be reconstructed from input-output data in fact are linear combinations of the components of the n k-step ahead output predictors

$$\hat{y}(t+k|t), \quad k = \{1, 2, \ldots, n\} \tag{123}$$

where n is the model order (the dimension of x). See also Appendix 4.A in [Ljung, 1987]. We could then form these predictors, and select a basis among their components:

$$x(t) = L \begin{pmatrix} \hat{y}(t+1|t) \\ \vdots \\ \hat{y}(t+n|t) \end{pmatrix} \tag{124}$$

The choice of L will determine the basis for the state-space realization, and is done in such a way that it is well conditioned. The predictor $\hat{y}(t+k|t)$ is a linear function of $u(s), y(s), \;\; 1 \leq s \leq t$ and can efficiently be determined by linear projections directly on the input output data. (There is one complication in that $u(t+1), \ldots, u(t+k)$ should not be predicted, even if they affect $y(t+k)$.)

What we have described now is the *subspace projection* approach to estimating the matrices of the state-space model (121), including the basis for the representation and the noise covariance matrices. There are a number of variants of this approach. See among several references, e.g. [Overschee and DeMoor, 1994], [Larimore, 1983]

The approach gives very useful algorithms for model estimation, and is particularly well suited for multivariable systems. The algorithms also allow numerically very reliable implementations. At present, the asymptotic properties of the methods are not fully investigated, and the general results quoted in Section 5.2 are not directly applicable. Experience has shown, however, that confidence intervals computed according to the general asymptotic theory, are good approximations. One may also use the estimates obtained by a subspace method as initial conditions for minimizing the prediction error criterion (74).

7 Data Quality

It is desirable to affect the conditions under which the data are collected. The objective with such *experiment design* is to make the collected data set Z^N as informative as possible with respect to the models to be built using the data. A considerable amount of theory around this topic can be developed and we shall here just review some basic points.

The first and most important point is the following one

1. *The input signal u must be such that it exposes all the relevant properties of the system.* It must thus not be too "simple". For example, a pure sinusoid

$$u(t) = A\cos\omega t$$

will only give information about the system's frequency response at frequency ω. This can also be seen from (85). The rule is that

- the input must contain at least as many different frequencies as the order of the linear model to be built.

  To be on the safe side, a good choice is to let the input be random (such as filtered white noise). It then contains all frequencies.

Another case where the input is too simple is when it is generated by feedback such as

$$u(t) = -Ky(t) \tag{125}$$

If we would like to build a first order ARX model

$$y(t) + ay(t-1) = bu(t-1) + e(t)$$

we find that for any given α all models such that

$$a + bK = \alpha$$

will give identical input-output data. We can thus not distinguish between these models using an experiment with (125). That is, we can not distinguish between any combinations of "a" and "b" if they satisfy the above condition for a given "α". The rule is

- If closed-loop experiments have to be performed, the feedback law must not be too simple. It is to be preferred that a set-point in the regulator is being changed in a random fashion.

The second main point in experimental design is

2. *Allocate the input power to those frequency bands where a good model in particularly important.*

   This is also seen from the expression (85).

   If we let the input be filtered white noise, this gives information how to choose the filter. In the time domain it is often useful to think like this:

- Use binary (two-level) inputs if linear models are to be built: This gives maximal variance for amplitude-constrained inputs.
- Check that the changes between the levels are such that the input occasionally stays on one level so long that a step response from the system has time, more or less, to settle. There is no need to let the input signal switch so quickly back and forth that no response in the output is clearly visible.

Note that the second point is really just a reformulation in the time domain of the basic frequency domain advice: let the input energy be concentrated in the important frequency bands.

A third basic piece of advice about experiment design concerns the choice of sampling interval.

3. *A typical good sampling frequency is 10 times the bandwidth of the system.* That corresponds roughly to 5 – 7 samples along the rise time of a step response.

8 Model Validation and Model Selection

8.1 A Pragmatic Viewpoint

The system identification process has, as we have seen, these basic ingredients

- The set of models
- The data
- The selection criterion

Once these have been decided upon, we have, at least implicitly, defined a model: The one in the set that best describes the data according to the criterion. It is thus, in a sense, the best available model in the chosen set. But is it good enough? It is the objective of *model validation* to answer that question. Often the answer turns out to be "no", and we then have to go

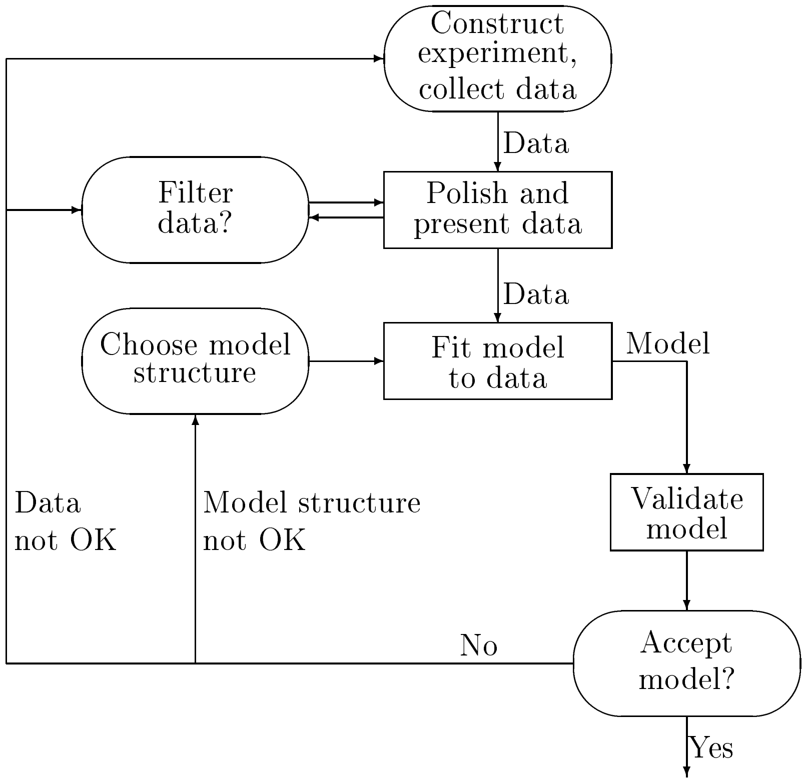

Figure 4: The identification loop

back and review the choice of model set, or perhaps modify the data set. See Figure 4!

How do we check the quality of a model? The prime method is to investigate how well it is capable of reproducing the behavior of a new set of data *(the validation data)* that was not used to fit the model. That is, we simulate the obtained model with a new input and compare this simulated output. One may then use one's eyes or numerical measurements of fit to decide if the fit in question is good enough. Suppose we have obtained several different models in different model structures (say a 4th order ARX model, a 2nd order BJ model, a physically parameterized one and so on) and would like to know which one is best. The simplest and most pragmatic approach to this problem is then to simulate each one of them on validation data, evaluate their performance, and pick the one that shows the most favorable fit to measured data. (This could indeed be a subjective criterion!)

8.2 The Bias-Variance Trade-off

At the heart of the model structure selection process is to handle the trade-off between bias and variance, as formalized by (92). The "best" model structure is the one that minimizes F_N, the fit between the model and the data for a *fresh* data set – one that was not used for estimating the model. Most procedures for choosing the model structures are also aiming at finding this best choice.

Cross Validation

A very natural and pragmatic approach is *Cross Validation*. This means that the available data set is split into two parts, *estimation data*, $Z_{\text{est}}^{N_1}$ that is used to estimate the models:

$$\hat{\theta}_{N_1} = \arg\min V_{N_1}(\theta, Z_{\text{est}}^{N_1}) \tag{126}$$

and *validation data*, $Z_{\text{val}}^{N_2}$ for which the criterion is evaluated:

$$\hat{F}_{N_1} = V_{N_2}(\hat{\theta}_{N_1}, Z_{\text{val}}^{N_2}) \tag{127}$$

Here V_N is the criterion (75). Then $\hat{F}_N$ will be an unbiased estimate of the measure F_N, defined by (88), which was discussed at length in the previous section. The procedure would the be to try out a number of model structures, and choose the one that minimizes $\hat{F}_{N_1}$.

Such cross validation techniques to find a good model structure has an immediate intuitive appeal. We simply check if the candidate model is capable of "reproducing" data it hasn't yet seen. If that works well, we have some confidence in the model, regardless of any probabilistic framework that might be imposed. Such techniques are also the most commonly used ones.

A few comments could be added. In the first place, one could use different splits of the original data into estimation and validation data. For example, in statistics, there is a common cross validation technique called "leave one out". This means that the validation data set consists of one data point "at a time", but successively applied to the whole original set. In the second place, the test of the model on the validation data does not have to be in terms of the particular criterion (127). In system identification it is common practice to simulate (or predict several steps ahead) the model using the validation data, and then visually inspect the agreement between measured and simulated (predicted) output.

Estimating the Variance Contribution – Penalizing the Model Complexity

It is clear that the criterion (127) has to be evaluated on the validation data to be of any use – it would be strictly decreasing as a function of model flexibility if evaluated on the estimation data. In other words, the adverse effect of the dimension of θ shown in (92) would be missed. There are a number of criteria – often derived from entirely different viewpoints – that try to capture the influence of this variance error term. The two best known ones are *Akaike's Information Theoretic Criterion, AIC*, which has the form (for Gaussian disturbances)

$$\tilde{V}_N(\theta, Z^N) = \left(1 + \frac{2 dim\theta}{N}\right) \frac{1}{N} \sum_{t=1}^{N} \varepsilon^2(t,\theta) \tag{128}$$

and *Rissanen's Minimum Description Length Criterion, MDL* in which $dim\theta$ in the expression above is replaced by $\log N dim\theta$. See [Akaike, 1974a] and [Rissanen, 1978].

The criterion $\tilde{V}_N$ is then to be minimized both with respect to θ and to a family of model structures. The relation to the expression (89) for F_N is obvious.

8.3 Residual Analysis

The second basic method for model validation is to examine the residuals ("the leftovers") from the identification process. These are the prediction errors

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

i.e. what the model could not "explain". Ideally these should be independent of information that was at hand at time $t-1$. For example if $\varepsilon(t)$ and $u(t-\tau)$ turn out to be correlated, then there are things in $y(t)$ that originate from $u(t-\tau)$ but have not been properly accounted for by $\hat{y}(t|\hat{\theta}_N)$ The model has then not squeezed out all relevant information about the system from the data.

It is good practice to always check the residuals for such (and other) dependencies. This is known as *residual analysis.* A basic reference for how to perform this is [Draper and Smith, 1981].

9 Back to Data: The Practical Side of Identification

9.1 Software for System Identification

In practice System Identification is characterized by some quite heavy numerical calculations to determine the best model in each given class of models. This is mixed with several user choices, trying different model structures, filtering data and so on. In practical applications we will thus need good software support. There are now many different commercial packages for identification available, such as Mathwork's System Identification Toolbox [Ljung, 1995], Matrix$'_{\mathrm{x}}$s System Identification Module [$MATRIX_x$, 1991] and PIM [Landau, 1990]. They all have in common that they offer the following routines:

A *Handling of data, plotting, etc.*

Filtering of data, removal of drift, choice of data segments, etc.

B *Non-parametric identification methods*

Estimation of covariances, Fourier transforms, correlation- and spectral-analysis, etc.

C *Parametric estimation methods*

Calculation of parametric estimates in different model structures.

D *Presentation of models*

Simulation of models, estimation and plotting of poles and zeros, computation of frequency functions, and plotting Bode diagrams, etc.

E *Model validation*

Computation and analysis of residuals ($\varepsilon(t, \hat{\theta}_N)$). Comparison between different models' properties, etc.

The existing program packages differ mainly in various user interfaces and by different options regarding the choice of model structure according to C above. For example, MATLAB's Identification Toolbox [Ljung, 1995] covers all linear model structures discussed here, including arbitrarily parameterized linear models in continuous time.

Regarding the user interface, there is now a clear trend to make it graphically oriented. This avoids syntax problems and relies more on "click and move", at the same time as tedious menu-labyrinths are avoided. More aspects of CAD tools for system identification are treated in [Ljung, 1993].

9.2 How to Get to a Good Model?

It follows from our discussion that the most essential element in the process of identification – once the data have been recorded – is to try out various model structures, compute the best model in the structures; using (38), and then validate this model. Typically this has to be repeated with quite a few different structures before a satisfactory model can be found.

While one should not underestimate the difficulties of this process, the following simple procedure to get started and gain insight into the models could be suggested:

1. Find out a good value for the delay between input and output, e.g. by using correlation analysis.
2. Estimate a fourth order linear model with this delay using part of the data, and simulate this model with the input and compare the model's simulated output with the measured output over the whole data record. In MATLAB language this is simple,

   ```
   z = [y  u];
   compare(z,arx(z(1:200,:),[4 4 1]));
   ```

If the model/system is unstable or has integrators, use prediction over a reasonable large time horizon instead of simulation.

Now, either of two things happen:

- *The comparison "looks good".* Then we can be confident that with some extra work – trying out different orders, and various noise models – we can fine tune the model and have an acceptable model quite soon.
- *The comparison "does not look good".* Then we must do further work. There are three basic reasons for the failure.
  1. *A good description needs higher order linear dynamics.* This is actually in practice the least likely reason, except for systems with mechanical resonances. One then obviously has to try higher order models or focus on certain frequency bands by band pass filtering.
  2. *There are more signals that significantly affect the output.* We must then look for what these signals might be, check if they can be measured and if so include them among the inputs. Signal sources that cannot be traced or measured are called "disturbances" and we simply have to live with the fact that they will have an adverse effect on the comparisons.

3. *Some important non-linearities have been overlooked.* We must then resort to semi-physical modeling to find out if some of the measured signals should be subjected to non-linear transformations. If no such transformations suggest themselves, one might have to try some non-linear black-box model, like a neural network.

Clearly, this advice does not cover all the art of identification, but it is a reasonable first approximation.

Example 6 *Aircraft dynamics*

Let us try the recipe on the aircraft data in figure 1! Picking the canard angle only as the input, estimating a fourth order model based on the data points 90 to 180, gives figure 5. (We use 10-step ahead prediction in this example since the models are unstable – as they should be, JAS has unstable dynamics in this flight case). It does not "look good". Let us try alternative 2: More inputs. We repeat the procedure using all three inputs in figure 1. That is, the model is computed as

```
arx([y u1 u2 u3], [4 4 4 4 1 1 1])
```

on the same data set. The comparison is shown in figure 4. It "looks good". By further fine-tuning, as well as using model structures from physical modeling, only slight improvements are obtained.

Example 7 *Buffer vessel dynamics*

Let us now consider the pulp process of figure 2. We use the κ-number before the vessel as input and the κ-number after the vessel as output. The delay is preliminarily estimated to 12 samples. Our recipe, where a fourth order linear model is estimated using the first 200 samples and then simulated over the whole record gives figure 7. It does not look good.

Some reflection shows that this process indeed must be non-linear (or time-varying): the flow and the vessel level definitely affect the dynamics. For example, if the flow was a plug flow (no mixing) the vessel would have a dynamics of a pure delay equal to vessel volume divided by flow.

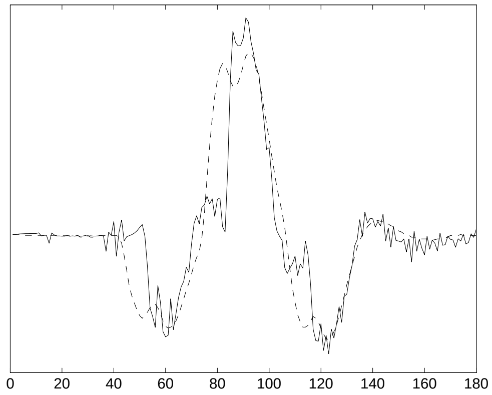

Figure 5: Dashed line: Actual Pitch rate. Solid line: 10 step ahead predicted pitch rate, based on the fourth order model from canard angle only.

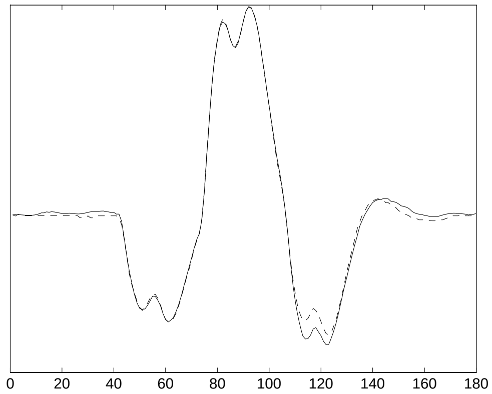

Figure 6: As figure 5 but using all three inputs.

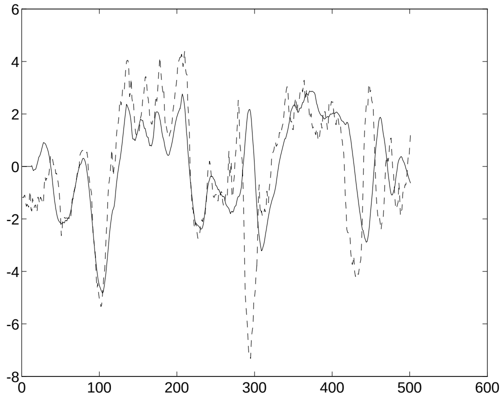

Figure 7: Dashed line: κ-number after the vessel, actual measurements. Solid line: Simulated κ-number using the input only and a fourth order linear model with delay 12, estimated using the first 200 data points.

Let us thus resample the date accordingly, i.e. so that a new sample is taken (by interpolation from the original measurement) equidistantly in terms of integrated flows divided by volume. In MATLAB terms this will be

```
z = [y,u]; pf = flow./level;
t =1:length(z)
newt =
table1([cumsum(pf),t],[pf(1)sum:(pf)]’ );
newz = table1([t,z], newt);
```

We now apply the same procedure to the resampled data. This gives figure 8. This “looks good”. Somewhat better numbers can then be obtained by fine-tuning the orders.

References

[Akaike, 1974a] Akaike, H. (1974a). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19:716–723.

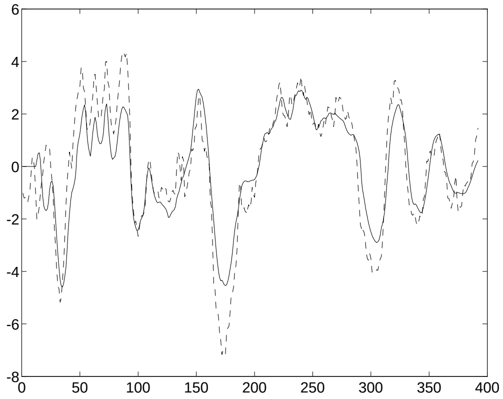

Figure 8: Same as figure 7 but applied to resampled data

[Akaike, 1974b] Akaike, H. (1974b). Stochastic theory of minimal realization. *IEEE Transactions on Automatic Control*, AC-19:667–674.

[Åström and Bohlin, 1965] Åström, K. J. and Bohlin, T. (1965). Numerical identification of linear dynamic systems from normal operating records. In *IFAC Symposium on Self-Adaptive Systems*, Teddington, England.

[Box and Jenkins, 1970] Box, G. E. P. and Jenkins, D. R. (1970). *Time Series Analysis, Forcasting and Control.* Holden-Day, San Francisco.

[Breiman, 1993] Breiman, L. (1993). Hinging hyperplanes for regression, classification and function approximation. *IEEE Trans. Info. Theory*, 39:999–1013.

[Brillinger, 1981] Brillinger, D. (1981). *Time Series: Data Analysis and Theory.* Holden-Day, San Francisco.

[Dennis and Schnabel, 1983] Dennis, J. E. and Schnabel, R. B. (1983). *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall.

[Draper and Smith, 1981] Draper, N. and Smith, H. (1981). *Applied Regression Analysis, 2nd ed.* Wiley, New York.

[Juditsky et al., 1995] Juditsky, A., Hjalmarsson, H., Benveniste, A., Delyon, B., Ljung, L., Sjöberg, J., and Zhang, Q. (1995). Nonlinear black-box modeling in system identification: Mathematical foundations. *Automatica*, 31(12):1724–1750.

[Landau, 1990] Landau, I. D. (1990). *System Identificaiton and Control Design Using P.I.M. + Software.* Prentice Hall, Engelwood Cliffs.

[Larimore, 1983] Larimore, W. E. (1983). System identification, reduced order filtering and modelling via canonical variate analysis. In *Proc 1983 American Control Conference*, San Francisco.

[Ljung, 1987] Ljung, L. (1987). *System Identification - Theory for the User.* Prentice-Hall, Englewood Cliffs, N.J.

[Ljung, 1993] Ljung, L. (1993). Identification of linear systems. In Linkens, D. A., editor, *CAD for Control Systems*, chapter 6, pages 147–165. Marcel Dekker, New York.

[Ljung, 1995] Ljung, L. (1995). *The System Identification Toolbox: The Manual.* The MathWorks Inc. 1st edition 1986, 4th edition 1995, Natick, MA.

[Ljung and Glad, 1994] Ljung, L. and Glad, T. (1994). *Modeling of Dynamic Systems.* Prentice Hall, Englewood Cliffs.

[Ljung and Söderström, 1983] Ljung, L. and Söderström, T. (1983). *Theory and Practice of Recursive Identification.* MIT press, Cambridge, Mass.

[$MATRIX_x$, 1991] $MATRIX_x$ (1991). $MATRIX_x$ users guide. *Integrated Systems Inc.*, Santa Clara, CA.

[Overschee and DeMoor, 1994] Overschee, P. V. and DeMoor, B. (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30:75–93.

[Poggio and Girosi, 1990] Poggio, T. and Girosi, F. (1990). Networks for approximation and learning. *Proc. of the IEEE*, 78:1481–1497.

[Rissanen, 1974] Rissanen, J. (1974). Basis of invariants and canonical forms for linear dynamic systems. *Automatica*, 10:175–182.

[Rissanen, 1978] Rissanen, J. (1978). Modelling by shortest data description. *Automatica*, 14:465–471.

[Schoukens and Pintelon, 1991] Schoukens, J. and Pintelon, R. (1991). *Identification of Linear Systems: A Practical Guideline to Accurate Modeling.* Pergamon Press, London (U.K.).

[Sjöberg et al., 1995] Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P., Hjalmarsson, H., and Juditsky, A. (1995). Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, 31(12):1691–1724.

[Söderström and Stoica, 1989] Söderström, T. and Stoica, P. (1989). *System Identification.* Prentice-Hall Int., London.

[Zhang and Benveniste, 1992] Zhang, Q. and Benveniste, A. (1992). Wavelet networks. *IEEE Trans Neural Networks*, 3:889–898.

/home/rt/ljung/papers/eeenc/enc.tex

Avdelning, Institution
Division, Department

Division of Automatic Control
Department of Electrical Engineering

Datum
Date

2007-06-29

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ________________

Rapporttyp
Report category

☐ Licentiatavhandling
☐ Examensarbete
☐ C-uppsats
☐ D-uppsats
☒ Övrig rapport
☐ ________________

ISBN
—

ISRN
—

Serietitel och serienummer
Title of series, numbering

LiTH-ISY-R-2809

ISSN
1400-3902

URL för elektronisk version

http://www.control.isy.liu.se

Titel
Title

System Identification

Författare
Author

Lennart Ljung

Sammanfattning
Abstract

This is a survey of System Identification.

Nyckelord
Keywords

identification