KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Politechnika Wrocławska

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

ROZWÓJ POTENCJAŁU I OFERTY DYDAKTYCZNEJ POLITECHNIKI WROCŁAWSKIEJ

Wrocław University of Technology

Advanced Informatics and Control

Tomasz Larkowski, Keith J. Burnham

SYSTEM IDENTIFICATION, PARAMETER ESTIMATION AND FILTERING

Wrocław 2011

Projekt współfinansowany ze środków Unii Europejskiej w ramach
Europejskiego Funduszu Społecznego

Wrocław University of Technology

Advanced Informatics and Control

Tomasz Larkowski, Keith J. Burnham

SYSTEM IDENTIFICATION, PARAMETER ESTIMATION AND FILTERING

Wrocław 2011

Reviewer: Olivier C. Haas

ISBN 978-83-62098-86-6

Published by PRINTPAP Łódź, www.printpap.pl

Contents

AR	Auto-regressive
ARARX	Auto-regressive with auxiliary input where process noise is an auto-regression
ARARMAX	Auto-regressive with auxiliary input where process noise is also auto-regressive moving average process
ARMA	Auto-regressive moving average
ARIMAX	Auto-regressive integrated moving average with auxiliary input
ARMAX	Auto-regressive moving average with auxiliary input
ARX	Auto-regressive with auxiliary input
BJ	Box-Jenkins
BLUE	Best linear unbiased estimator
cf.	Confer (compare)
dB	Decibel
EE	Equation error
e.g.	Exempli gratia (for example)
EIV	Errors-in-variables
EKF	Extended Kalman filter
FIR	Finite impulse response
SKF	Stationary (or steady-state) Kalman filter
i.e.	Id est (that is)
JEKF	Extended Kalman filter for joint state and parameter estimation
KF	Kalman filter
KFPE	Kalman filter tuned for parameter estimation
LS	Least squares
LTI	Linear time-invariant
LTV	Linear time-varying
NARX	Nonlinear auto-regressive with auxiliary input
OE	Output error
RLS	Recursive least squares algorithm
SISO	Single-input single-output
SKF	Stationary (or steady-state) Kalman filter
SNR	Signal-to-noise ratio

Preface

This is one of a series of Masters level monographs, which have been produced for taught modules within a common course designed for Advanced Informatics and Control. The new common course development forms a collaboration between Coventry University, United Kingdom and Wroclaw University of Technology, Poland. The new course recognises the complexity of new and emerging advanced technologies in informatics and control, and each text is matched to the topics covered in an individual taught module. The source of much of the material contained in each text is derived from lecture notes, which have evolved over the years, combined with illustrative examples which may well have been used by many other authors of similar texts that can be found. Whilst the sources of the material may be many any errors that may be found are the sole responsibility of the authors.

Chapter 1

Introduction

The 'art' of system identification is an experimental process, where by making use of the available input and output data and *a priori* knowledge, one aims to mathematically describe the causalities that govern the behaviour of a system (Söderström & Stoica 1989) and (Ljung 1999). The task of identifying a system or a process[1] , where these terms could refer to a standard mass-spring-damper example or, likewise, greenhouse effect, is a commonly encountered problem which is not necessarily restricted to only the control engineering domain but it is equally faced within various other fields of sciences. These range from economy, finance, social science through biology, medicine, chemistry to computer and earth science, to mention only few examples. A survey summarising some important developments in the subject of system identification has relatively recently been given in (Ljung 2008). Parameter estimation can be considered as a subtask of the system identification procedure, during which the parameters that describe a particular model structure are determined making use of a chosen algorithm, which can be referred as a parameter estimator. There is no single parameter estimation algorithm which is suitable for all model structures. Similarly, there is no single model structure which is capable of adequately modelling all systems. Consequently, the choice of a tandem - model structure and parameter estimator is not trivial. Whilst in the case of system identification one is interested in building a model of a process based on measured input/output data, filtering deals with a dual problem of recovering noise-free signals based on a known (or, in practice, estimated) system model. This script aims to provide

[1] Terms 'system' and 'process' are used interchangeably in the sequel.

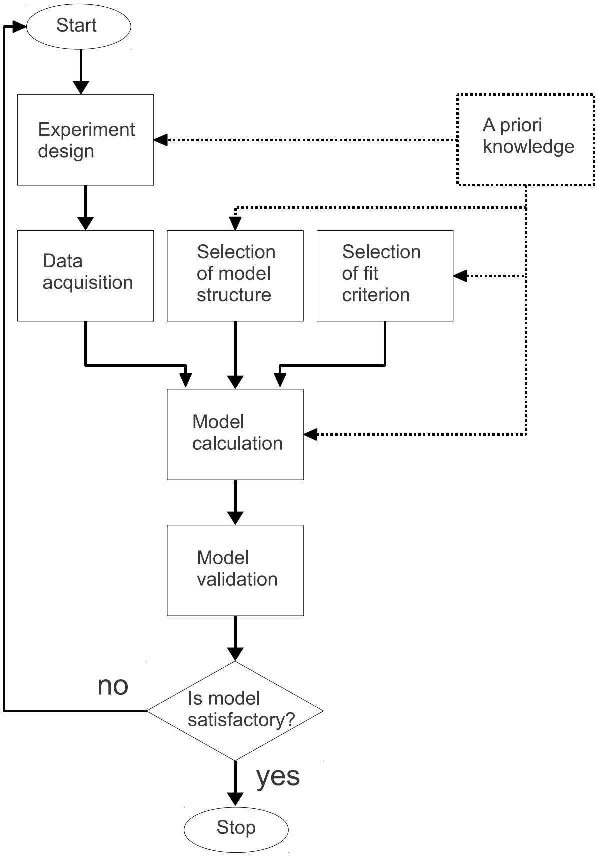

Figure 1.1: The methodology of system identification.

an introduction into basic concepts of system identification and filtering.

The process of system identification, depicted in Figure 1.1, cf. (Ljung 1999) and (Ikonen & Najim 2002), is of iterative nature and can be divided into the following steps:

1. Experiment design - This refers to the planning and preparation of experiments, which should be carried out such that they will provide suitable data, ideally, with maximum information content subject to constraints imposed by the process. For instance, the choice of an appropriate sam-

pling period and input signal are both crucial. The sampling time must be specified such that all system dynamics of interest can be captured, whilst the input should excite the system sufficiently in order to obtain data which is informative. These can be approximately found from *a priori* knowledge regarding the system bandwidth, the expected range of operation, typical system behaviour etc.

2. Data acquisition - This step refers to the acquisition of signals and their pre-treatment before utilising them for model calculation. Data recorded can contain undesired components such as measurement noise, quantisation noise, outliers etc. Some parts of data can even be missing or/and discrepancies in the time stamp between different signals measured can be present. All these factors can significantly reduce the information content of interest and prevent obtaining a good model. Therefore, the data recorded should be examined first and pre-treated appropriately. For instance, by possessing approximate *a priori* knowledge about the system bandwidth, all frequencies higher than twice the system bandwidth should be filtered via a low-pass filter.

3. Selection of model structure - Model structure refers to the choice of a particular candidate class of models from which the best model (for a given purpose) is sought. This also includes specification of inputs and outputs. Loosely speaking, it is expected the chosen model structure will be sufficiently flexible to capture the main and/or important dynamics of the system and at the same time to be of parsimonious complexity, so that it can be handled by practically available hardware equipment. Consequently, in practice there exists a trade-off between the model complexity and its modelling capabilities. More complex model structures yield more precise representation of the actual process, whilst more simple model structures are easier to be estimated, require less computational power and their behaviour and properties are easier to understand and analyse. In general, the choice of an appropriate model structure is very difficult and requires at least some prior engineering knowledge, intuition and/or insight into the actual process properties.

4. Selection of fit criterion - Prior to the calculation of a model one has to choose a criterion against which the model validity is to be assessed. The most commonly used and straightforward criteria are based, at least partially, on a goodness of fit of the data generated by the estimated model to the measured data.

5. Model calculation - This step deals with the choice of the algorithm for estimating the model parameters, and is often referred to as a parameter estimation phase. Such decisions depend on the model structure to which the model belongs and the data. The data quality can influence the ability of a chosen algorithm to estimate the model parameters. For instance, some algorithms are more susceptible to a high noise contamination than others.

6. Model validation - In this step the ability of the model to reproduce the behaviour of the actual system is quantified and assessed with respect to the intended purpose of its later usage. Usually, the stability and confidence in the parameters of the estimated model are checked. The most important validation step is to check the degree to which the estimated model reproduces the actual process behaviour using data sets not used in the parameter estimation phase (validation data sets). This procedure is called cross-validation. It verifies that the estimated model can be generalised to a wider sets of data than just those used for its estimation (training data sets).

It is observed from Figure 1.1 that the entire process of system identification is of iterative nature, since it is very unlikely to arrive at a satisfactory model at the first run. Usually, it takes several iterations during which some previously made choices are revisited. For instance, it may be found the the chosen model structure is not flexible enough or that it is too complex and needs to be changed.

Summarising, it should be emphasised that the final model should not be considered as a 'true' description of the actual process, since such notion does not exist in practice. At most the final model can be regarded as a good enough description for a given purpose. Following (Ljung 1999), 'Our acceptance of models should thus be guided by "usefulness" rather then "truth"'.

Questions

- Describe briefly the steps of the system identification process.
- Discuss the existence of a trade-off between model complexity and modelling capabilities.
- Explain why in practice the system identification procedure is of an iterative nature.
- Does the notion of a 'true' system model exist in practice?
- Explain what is meant by a 'model for purpose'.

Chapter 2

Basic concepts

2.1 White-box, grey-box and black-box modelling approaches

1. **White-box models** (or, alternatively, first principles/fundamental/physical models) - based on physical laws and relationships such as conservation of force, mass, energy, momentum etc. The parameters of such models have physical interpretation, e.g. mass, thermal conductivity or capacitance. Their development requires certain *a priori* knowledge about the process behaviour and relatively detailed engineering insight into the system. Hence, usually the modelling procedure involved is laborious. White-box models are in the form of ordinary/partial differential equations, integral equations or both, with appropriate initial/boundary conditions. The overall model complexity depends directly on the inherent complexity of the process being modelled and on the level of intended precision that is desired. In general, the resulting model can be quite complex and contain a significant number of parameters as well as non trivial nonlinear functional relationships between model components. In order to simplify such models it is often possible, for example, to substitute certain functions with less complex counterparts, neglect components which are not of a prime importance, consider using lumped parameters, introduce idealised assumptions etc. At least in principle, the white-box models can be obtained without the need of employing any identification tools.
   Advantages:

   + Model structure directly reflects the phenomena being modelled
   + Parameters possess physical meaning

   Disadvantages:

   – Usually complex models

   – Detailed engineering knowledge required

   – Laborious modelling procedure

2. **Black-box models** (or, alternatively empirical/experimental models) - constructed through the observation of the process input and output data, which is subsequently used to fit a model. Consequently, the model parameters do not (necessarily) possess any physical significance, i.e. they can be considered as auxiliary vehicles that help to explain the relationship between the system input and output, hence the name black-box models. Such models are obtained via identification procedure from experimental data, where the user can impose a particular model structure. Therefore, the model identified describes rather qualitatively the relationship between input and output, and not necessary the actual phenomena being modelled. Furthermore, their validity and reliability are both constrained by quality and information content of the data used for identification.

   Advantages:

   + Relatively simple models
   + Models created based solely on the input-output data, i.e. no prior engineering knowledge required
   + Relatively simple modelling procedure

   Disadvantages:

   – Lack of physical interpretation of the model hence parameters does not have physical meaning

   – Validity and reliability depend on data used for identification

3. **Grey-box models** (or, alternatively semi-physical models) - Grey-box models lie between the two extremes of white-box and black-box models. In brief, when constructing grey-box models some knowledge regarding the physics of the process is used, however not to the extent that a first principle model is built. For instance, if a given model comprises of two

sub-models, where the first sub-model is modelled as a white-box and the second is modelled as a black-box, the resulting overall model will be a grey-box model. The grey-box modelling overcomes (to some extent) limitations of white-box and black-box modelling approaches, whilst at the same time retains some of their corresponding advantages. As an example, one can consider a case where a process has a fairly linear dynamic behaviour but possesses a nonlinear steady-state characteristic. It is noted, that the knowledge regarding the steady-state characteristic, whose general shape can typically be inferred from the nature of a process under consideration, can be treated as an additional engineering insight. A grey-box modelling procedure could be to exploit this knowledge and approximate the system nonlinear steady-state characteristic via an appropriate static function (white-box modelling), whilst deriving a linear model to describe the process dynamics using only the measure data (black-box modelling).

Another illustrative example of a grey-box modelling methodology could be that presented in (Lindskog & Ljung 1993). There the additional engineering knowledge is used to transform the measured signals into new auxiliary signals that are more suitable for explaining the actual process behaviour, with standard black-box identification. With reference to (Lindskog & Ljung 1993), this idea is illustrated by a following example - the goal is to find a model relating the voltage applied to an electric heater to room temperature. On the one hand, the white-box modelling approach can be followed which requires writing down all physical equations for the conversion of the voltage to the power of the heater, the heat transfer via radiation and convection from the radiator to the room etc. The resulting set of equations can be relatively complex and include several coefficients, such as heat transfer coefficient, specific heat capacities, radiator exponent etc., which may not all be known. On the other hand, a black-box approach could be chosen and a simple linear dynamic model structure fitted to the data yielding a model, which will explain to some extent the data observed. However, it is very likely that it cannot be generalised to other data sets and different operating conditions. A grey-box modelling approach would use the fact that it is not the voltage but the heater power that actually changes the room temperature. Consequently, the black-box identification procedure can be used with same structure but with squared voltage as the input. This minor transformation is a direct result of additional engineering insight, hence the method represents the grey-box modelling approach.

Advantages:

- + Models of medium complexity
- + Allows exploitation of potential engineering knowledge
- + Knowledge of all parameters unnecessary

Disadvantages:

- – Only some parameters possess physical meaning
- – Validity and reliability depend to some extent on data used for identification

2.2 Treatment of disturbances on measurements

In black-box and grey-box modelling approaches, the model is constructed by observing the system input and output. In a general case, these signals have to be measured via sensors. Every measuring device introduces a measurement error into the signals being measured. Consequently, the input-output data used for the model calculation will not correspond exactly the actual input-output signals, see Figure 2.1, where such a situation is illustrated diagrammatically. This configuration is called errors-in-variables (EIV), since it assumes potential measurement errors in all variables (signals) measured, see (Söderström 2007) for further details. The EIV system setup is utilised mainly for the purpose of obtaining an accurate insight into the internal system behaviour, i.e. precise determination of system parameters, especially if these parameters possess meaningful physical interpretation.

A somewhat simplified setup is given in Figure 2.2, where only the output signal is measured, whilst the input is assumed to be known exactly. Although this configuration may appear less realistic, in fact, it is not. This is due to the property that in the case of most (if not all) control problems the input is produced by a controller, hence it is known and not required to be measured, which, in turn, automatically avoids any potential measurement errors. Configuration in Figure 2.2 can be considered as a classical (or non-EIV) setup. It is relevant primarily where the task of the model is to anticipate the future system behaviour, i.e. prediction of signals. Note that the classical system setup is a special case of the EIV setup.

As an illustration of the EIV situation consider a task of constructing a model of some natural phenomena where both the input and the output signals

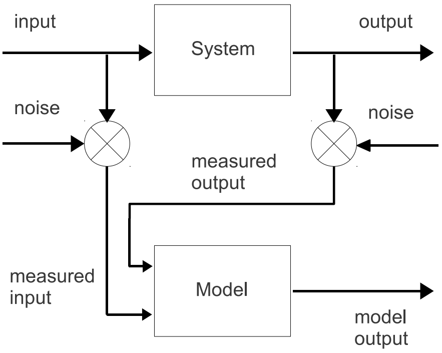

Figure 2.1: A general representation of the EIV estimation setup where both the input and output are measured.

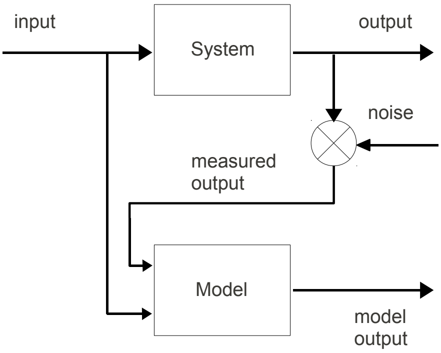

Figure 2.2: A general representation of the classical estimation setup where the input is known and only the output is measured.

are unknown and have to be measured. Such an example could be a rainfall model relating the amount of rain to the level of water in a river. Both of the quantities measured contain errors. Another example could be the greenhouse effect, where the input is the amount of emitted CO_2 into the atmosphere and the output the thickness of the ozone layer. Sometimes, although a given input signal could, at least theoretically, be calculated this is not feasible from a practical point of view. For instance, consider a radiator whose input is the mass-flow and the output being heat emitted. If one has access to the boiler, which supplies the water, then the mass-flow entering the radiator can be found (i.e. non-EIV setup), otherwise it has to be measured (i.e. EIV setup). The task of obtaining the actual mass-flow becomes increasingly less feasible when one considers a block of flats each with a number of rooms and radiators and where the water is supplied by a single boiler located in the basement. In this case, even if one had an access to the boiler the problem of determining a mass-flow to each individual radiator, taking into account inherent losses, leaks, pressure drops etc., would be very difficult.

A classical (non-EIV) configuration could be a control setup, where, for instance, one utilises a DC motor to change the azimuth position of a radar antenna. The voltage applied is set by a user (or by a controller) and is known, whilst the azimuth position of the antenna needs to be measured and thus is corrupted by measurement errors. Another example is the control of a greenhouse where the input is the speed of a ventilator and the output is the humidity inside the greenhouse. The ventilator speed depending directly on the voltage applied by a controller is known exactly whilst the humidity has to be measured via an appropriate sensor.

A pragmatic approach to cope with EIV situation, see (Ljung 1999), is to either filter the input or/and regard the measured noisy inputs as the noise-free inputs, while lumping their deviations from the actual inputs in the process noise or/and the noise on model output. Consequently, the influence of erroneous inputs is absorbed (to some extent) by other processes that model uncertainty. If such an approach is undertaken then identification algorithms suitable for a classical, i.e. non-EIV, setup can be used. In this script the main attention is given to the classical (control oriented) configuration.

Questions

- Explain what is meant by white-box, grey-box and black-box modelling approaches. Provide examples and discuss the corresponding advantages and disadvantages of the three modelling methodologies.
- Explain the differences between classical control setup and an EIV setup. Give examples of situations where these two frameworks can be advantageous.

Chapter 3

Linear and nonlinear systems

3.1 Introduction

In this chapter some basic notions regarding linear systems and their different representations are introduced. This is followed by a brief description of selected and frequently met nonlinear model structures.

3.2 Systems and their classification

3.2.1 Linearity

A linear system is defined as a system that fulfils the so-called principle of superposition. The principle of superposition states that the sum of responses of a system subjected to two separate inputs is the same as the response of that system when subjected to the input which is the sum of the two inputs. Expressed more formally it means that:

- if $f(u_1)$ is the output of a system to the input u_1, where $f(\cdot)$ is a continuous function, and
- if $f(u_2)$ is the output of a system to the input u_2
- then the system is said to be linear if $f(u_1 + u_2) = f(u_1) + f(u_2)$

Furthermore, if the system is linear, then the above condition also implies that

- if $f(u_1)$ is the output of a system to the input u_1
- then $\alpha f(u_1)$ is the output of a system to the input αu_1, i.e. $f(\alpha u_1) = \alpha f(u_1)$, where α is a scalar

The above proportionality property is called homogeneity (of degree one). If the principle of superposition does not hold a model or a system is said to be nonlinear. Most (if not all) systems, although nonlinear in nature, can behave approximately linearly within a certain range of operation.

It is also useful to distinguish the linearity in terms of the relationship between the input-output data and in terms of the system parameters. The first type of linearity, defined above via the principle of superposition, concerns the linear dependence of the output on the input. The second type of linearity, i.e. linearity in parameters, concerns the linear dependence of the output on the parameters. Consider the following examples, where y, u and α are the output, input and a scalar parameter, respectively, i.e.

- $y = \alpha u$ - system linear in both, i.e. input-output and parameter
- $y = \alpha^2 u$ - system linear in input-output, nonlinear in parameter
- $y = \alpha u^2$ - system nonlinear in input-output, linear in parameter
- $y = u^\alpha$ - system nonlinear in both, i.e. input-output and parameter

Dynamic and static systems

Systems (and models) can be divided into static and dynamic. The former class is a particular case of the latter class, where the output at a given time instance depends only on the value of the input at the same time instance, exclusively. In contrast, in the dynamic case, the system has memory and, as a consequence, the output at a given time instance can be expressed as a function of the past input and past output signals. This property extends their potential applicability into many areas, but can also lead to an increase in complexity in terms of modelling and identification. As an example consider the following systems, where the subscript k denotes a discrete-time instance, i.e.

- $y_k = \alpha u_k$ - static system
- $y_k = \alpha u_{k-1}$ - dynamic system

Note that in the case of static models there is no need to introduce the time index. A dynamic model is said to be casual if the output at a certain time instance is dependent on the input up to (and including) this time exclusively, i.e. it cannot depend on future values of the input. For instance consider the following dynamic systems, i.e.

- $y_k = \alpha u_{k-1}$ - casual dynamic system
- $y_k = \alpha u_{k+1}$ - not casual dynamic system

Time-invariant and time-varying systems

A model is defined as time-invariant if its response to a given input does not explicitly depend on time, e.g. it is assumed that laws that the system describes are identical regardless of time (hence the system parameters are constant). As an example consider the following systems, where α_k denotes that the parameter α varies over time, i.e.

- $y_k = \alpha u_k$ - linear time-invariant (LTI) system
- $y_k = \alpha_k u_k$ with $\alpha_k = \sin(k)$ - linear time-varying (LTV) system

Continuous-time and discrete-time systems

Systems can be continuous in time or discrete. The former class of systems appears to be the more natural and intuitive because models of such systems can be obtained directly by writing down, for instance, balance equations of a given process. This usually gives rise to a set of ordinary or partial differential equations. Discrete-time systems describe the relationship between the input and the output only at certain time instances. They are useful because the process data is usually available only at discrete-time instances at which it is sampled. Moreover, control and identification algorithms are inherently discrete-time, since they are implemented on digital platforms. If the sampling frequency is chosen appropriately then a continuous-time system can be approximated well by a corresponding discrete-time counterpart. As an example consider the following unforced first order systems, which both describe the process of exponential decay and where T_s is the sampling time and t is a continuous-time index, i.e

- $\frac{dy(t)}{dt} = -\alpha y(t)$ - continuous-time system
- $y_k = e^{-\alpha T_s} y_{k-1}$ - discrete-time system

Deterministic and stochastic systems

In systems which are deterministic the output can be completely determined from the knowledge of the input. That means there is no uncertainty embedded in the system. Such a case is however rather unrealistic in practice because there are always some signals present which are unmeasurable or/and dynamics which is not captured. These can be interpreted as unknown stochastic signals. Consequently, in a stochastic model, contrary to the deterministic case, at least one component is present within a system that is attributed to an unknown, hence unpredictable, portion of the output. This means that the output cannot be calculated completely knowing the input only. As an example consider the following systems:

- $y_k = \alpha u_k$ - deterministic system
- $y_k = \alpha u_k + e_k$ where e_k is an unknown noise sequence - stochastic system

In this text considerations are restricted mainly to dynamic discrete-time LTI stochastic models. However, some consideration is given to the LTV models and some selected nonlinear models too.

3.2.2 Different representations of linear systems

There are several ways to describe the dynamics of LTI systems such as differential/difference equations, transfer functions, impulse responses and state-space equations. If any of these representations is given the system is considered to be completely characterised (Hsia 1977).

Differential equation representation

Consider the general form of a continuous-time ordinary differential equation, (Nise 2008), i.e.

$$\begin{aligned} a_{n_a}\frac{d^{n_a}y(t)}{dt^{n_a}} + a_{n_a-1}\frac{d^{n_a-1}y(t)}{dt^{n_a-1}} + \ldots + a_0 y(t) = \\ b_{n_b}\frac{d^{n_b}u(t)}{dt^{n_b}} + b_{n_b-1}\frac{d^{n_b-1}u(t)}{dt^{n_b-1}} + \ldots + b_0 u(t), \quad (3.1) \end{aligned}$$

where $u(t)$ and $y(t)$ are the input and the output, respectively. The parameters a_i and b_i as well as the orders n_a and n_b (with $n_a > n_b$) define the particular process which is modelled. By applying the Laplace transform[1] to the both sides of equation (3.1) one obtains

$$\left(a_{n_a}s^{n_a} + a_{n_a-1}s^{n_a-1} + \ldots + a_0\right) Y(s) + Y_{\text{initial}} = \\ \left(b_{n_b}s^{n_b} + b_{n_b-1}s^{n_b-1} + \ldots + b_0\right) U(s) + U_{\text{initial}}, \quad (3.2)$$

where

$$Y_{\text{initial}} = -a_{n_a}\sum_{i=1}^{n_a} s^{n_a-i}\frac{d^{i-1}y(0)}{dt^{i-1}} - a_{n_a-1}\sum_{i=1}^{n_a-1} s^{n_a-1-i}\frac{d^{i-1}y(0)}{dt^{i-1}} - \ldots - a_1 y(0), \quad (3.3)$$

$$U_{\text{initial}} = -b_{n_b}\sum_{i=1}^{n_b} s^{n_b-i}\frac{d^{i-1}u(0)}{dt^{i-1}} - b_{n_b-1}\sum_{i=1}^{n_b-1} s^{n_b-1-i}\frac{d^{i-1}u(0)}{dt^{i-1}} - \ldots - b_1 u(0). \quad (3.4)$$

It is noted that in contrast to (3.1) equation (3.2) is algebraic. By postulating that all initial conditions are null, i.e. $Y_{\text{initial}} = U_{\text{initial}} = 0$, (3.2) simplifies to

$$\left(a_{n_a}s^{n_a} + a_{n_a-1}s^{n_a-1} + \ldots + a_0\right) Y(s) = \left(b_{n_b}s^{n_b} + b_{n_b-1}s^{n_b-1} + \ldots + b_0\right) U(s). \quad (3.5)$$

This can be written in a form of a ratio of the output over the input signal, i.e.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_{n_b}s^{n_b} + b_{n_b-1}s^{n_b-1} + \ldots + b_0}{a_{n_a}s^{n_a} + a_{n_a-1}s^{n_a-1} + \ldots + a_0} = \frac{\sum_{j=0}^{n_b} b_j s^j}{\sum_{l=0}^{n_a} a_l s^l}. \quad (3.6)$$

The ratio of the Laplace transformed input and output is called the transfer function and it is usually denoted by $G(s)$. The transfer function characterises a given system, and due to the property that the input and the output are separated, the response of a system to any excitation can be calculated via $Y(s) = G(s)U(s)$. Note that this useful property was not present in the original formulation of the differential equation (3.1). In general, LTI systems, such as electrical networks, mechanical/pneumatic/hydraulic/heat-transfer systems, can all be described by transfer functions.

[1] Laplace transform, denoted $\mathcal{L}(\cdot)$, of derivative of n-th order is given by $\mathcal{L}\left[\frac{d^n y(t)}{dt^n}\right] = s^n F(s) - \sum_{i=1}^{n} s^{n-i}\frac{d^{i-1}f(0)}{dt^{i-1}}$.

Difference equation representation

Nowadays, most control and identification algorithms are implemented on digital computers, hence it is of interest to consider a discrete-time counterpart of the continuous-time ordinary differential equation (3.1). Discrete-time techniques operate on samples of continuous-time signals, which are recorded at (usually equidistant) discrete-time instances. The time between sampling is utilised to perform necessary calculations (the more complex the algorithm the more computationally demanding and time consuming it is) as well as for sending the calculated signals to the system being under control (Dutton, Thompson & Barraclough 1997). Consequently, discrete-time models disregard information between the samples and provide only snapshots of the actual signal. As long as the sampling time is short enough, such that there is not much change in between the samples, a discrete-time approximation will constitute a good description of the continuous system. A discrete-time counterpart of equation (3.1) is called a difference equation (since it involves difference operations on successive samples) and it is given by

$$a_0 y_k + a_1 y_{k-1} + \ldots + a_{n_a} y_{k-n_a} = b_0 u_k + b_1 u_{k-1} + \ldots + b_{n_b} u_{k-n_b} \tag{3.7}$$

where $n_a \geq n_b$. It must be emphasised here that there is no direct correspondence of the a and b parameters between the discrete-time difference equation (3.7) and the continuous-time differential equation (3.1) (hence also the transfer function (3.6)). For example a_1 in (3.1) will not, in general, be the same value as a_1 in (3.7). Equation (3.7) can be more conveniently re-written using a polynomial formulation as

$$A(q^{-1})y_k = B(q^{-1})u_k, \tag{3.8}$$

where the polynomials $A(q^{-1})$ and $B(q^{-1})$ are, respectively, given by

$$A(q^{-1}) = a_0 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a}, \tag{3.9}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots + b_{n_b} q^{-n_b} \tag{3.10}$$

with q^{-1} denoting a discrete-time shift operator defined as $q^{-d} y_k = y_{k-d}$. For convenience, it is usually assumed that a_0 is unity, i.e. $a_0 = 1$ by definition, which implies that the polynomial $A(q^{-1})$ is monic. Because most discrete-time dynamic systems have an internal delay, denoted $d \geq 0$, it is sometimes useful to incorporate that property explicitly into the difference equation, which leads to

$$A(q^{-1})y_k = q^{-d} B(q^{-1})u_k. \tag{3.11}$$

It is noted that equation (3.11) can be interpreted as a special case of equation (3.7) with an additional constraint that $b_0 = b_1 = \ldots = b_{d-1} = 0$. Therefore, the form without an explicit delay (3.7) is used in the sequel.

It is instructive to consider the relationship between the difference equation and the $\mathcal{Z}$-transform. Following (Pearson 1999), taking the $\mathcal{Z}$-transform of the difference equation (3.8) (which is defined in the discrete-time domain) one obtains the following equivalent representation (defined in the discrete frequency domain), i.e.

$$A(z^{-1})Y(z) = B(z^{-1})U(z), \tag{3.12}$$

where $Y(z)$ and $U(z)$ represent the $\mathcal{Z}$-transforms of the output and the input, respectively. These are, in general, given by

$$Y(z) = \sum_{k=-\infty}^{\infty} y_k z^{-k}, \tag{3.13}$$

$$U(z) = \sum_{k=-\infty}^{\infty} u_k z^{-k}, \tag{3.14}$$

where z is a complex number. By assuming zero initial conditions, i.e. $y_k = u_k = 0 \ \forall \ k < 0$, the summation can start from $k = 0$. The polynomials $A(z^{-1})$ and $B(z^{-1})$ have analogous structures to those given in (3.9)-(3.10). Consequently, a discrete-time transfer function is defined as follows

$$G(z) = \frac{Y(z)}{U(z)} = \frac{B(z^{-1})}{A(z^{-1})} = \frac{\sum_{j=0}^{n_b} b_j z^{-j}}{\sum_{l=0}^{n_a} a_l z^{-l}}. \tag{3.15}$$

It is important to emphasise that the difference equation representation given in (3.8) is equivalent to the transfer function representation (3.15). Also note the similarity of the continuous-time transfer function (3.6) to its discrete-time counterpart (3.15). Again a lack of direct correspondence between the a and b parameters of the two representations is stressed.

State-space representation

Alternatively to differential or difference equations one can make use of the corresponding continuous-time or discrete-time state-space equations. A continuous-time state-space representation is defined by

$$\dot{x}(t) = A_c(\theta)x(t) + B_c(\theta)u(t), \tag{3.16}$$

$$y(t) = C_c(\theta)x(t) + D_c(\theta)u(t), \tag{3.17}$$

where $\dot{x}(t) = \frac{dx(t)}{dt}$ and $A_c(\theta) \in \mathbb{R}^{n \times n}$, $B_c(\theta) \in \mathbb{R}^n$, $C_c(\theta) \in \mathbb{R}^{1 \times n}$, $D_c(\theta) \in \mathbb{R}$ are model matrices built from the parameters contained in the parameter vector θ. The continuous-time state vector $x(t) \in \mathbb{R}^n$ comprises of model states that usually have direct physical interpretation such as position, velocity, acceleration etc. It is observed that (3.16) comprises of n ordinary first-order differential equations and that the model output (3.17) is a linear combination of states. The continuous-time transfer function can be obtained from (3.16)-(3.17) as follows

$$G(s) = C_c(\theta) \left[sI - A_c(\theta) \right]^{-1} B_c(\theta) + D_c(\theta), \tag{3.18}$$

where I denotes an identity matrix of appropriate dimension.

A discrete-time state-space representation is defined, analogously to (3.16)-(3.17), as follows

$$x_{k+1} = A(\theta) x_k + B(\theta) u_k, \tag{3.19}$$

$$y_k = C(\theta) x_k + D(\theta) u_k, \tag{3.20}$$

where, similarly as in the continuous-time case, model matrices $A(\theta) \in \mathbb{R}^{n \times n}$, $B(\theta) \in \mathbb{R}^n$, $C(\theta) \in \mathbb{R}^{1 \times n}$, $D(\theta) \in \mathbb{R}$ are constructed from the model parameters θ and $x_k \in \mathbb{R}^n$ is a discrete-time state vector. It is to be emphasised that there is no direct correspondence between the model matrices of the continuous-time state-space model and its discrete-time counterpart. A discrete-time transfer function can be obtained from (3.19)-(3.20) by

$$G(z) = C(\theta) \left[zI - A(\theta) \right]^{-1} B(\theta). \tag{3.21}$$

Relationships between continuous-time and discrete-time systems

Until now, continuous-time and discrete-time systems have been treated separately, however it is important to examine their mutual relationships. The question of deciding between the continuous-time and discrete-time modelling depends on a particular application. Following (Ljung 1999), relationships between these two modelling approaches are interesting for two main reasons. Firstly, when a discrete-time model has been obtained from measured (sampled) input-output data, it is often desirable to compare that model against the continuous-time counterpart, whose parameters possess physical meaning. Secondly, when a continuous-time model has been constructed one may wish to determine how the output and states vary between successive sampling instances, with the input kept piece-wise constant. With the reference to (Ljung 1999), relationships between the continuous-time and discrete-time models can be divided into two categories, i.e. approximate and exact relations. Whilst the approximate relations are based on some approximation of the differential operator, the exact relations correspond to exact solutions of continuous-time system over a chosen sampling period, denoted $T_s = t_{k+1} - t_k$. The basis of an approximate realisation of continuous-time models is an approximation of the differential operator via a difference operator such as

$$\dot{x}(t) \approx \frac{x(t_{k+1}) - x(t_k)}{T_s}, \tag{3.22}$$

which corresponds to the so-called Euler approximation. Another, more precise, possibility is the so-called Tustin (or bilinear) transformation defined as

$$\dot{x}(t) \approx \frac{2}{T_s} \frac{x(t_{k+1}) - x(t_k)}{x(t_{k+1}) + x(t_k)}. \tag{3.23}$$

The goodness of approximation depends on the variability of the input $u(t)$ and the state vector $x(t)$ between the sampling instances. Therefore, if T_s is sufficiently small compared to the smallest time constant of the system then the discrete-time approximation will be accurate. The exact relationship between the continuous-time and the discrete-time representation can be obtained if it is assumed that the input signal is piece-wise constant between sampling instances, i.e.

$$u(t) = u(t_k) \qquad\qquad \text{for} \qquad\qquad t_k \leq t < t_{k+1}. \tag{3.24}$$

In fact this, at least to some extent, is not a too unrealistic assumption, especially when dealing with control systems where the control signal, being the output of a digital controller, is kept constant in between sampling instances. In such cases the differential equations can be solved analytically and provide exact solution from t_k to t_{k+1}, where

$$\begin{aligned} A(\theta) &= e^{A_c(\theta)T_s}, && (3.25) \\ B(\theta) &= \int_0^{T_s} e^{A_c(\theta)(T_s-\tau)} B_c(\theta) d\tau \\ &= A_c^{-1}(\theta) \left[e^{A_c(\theta)T_s} - I \right] B_c(\theta). && (3.26) \end{aligned}$$

If the condition stated in (3.24) is satisfied then no approximation is made and hence equations (3.25)-(3.26) form an exact discrete-time representation of the continuous-time system (Mańczak & Nahorski 1983). However, because $A_c(\theta)$ is usually quite a complex function of θ, the computation of the matrix exponential is difficult in practice. Therefore, use is frequently made of approximations such as those defined in (3.22) and (3.23). In fact, one can simplify the calculation of $e^{A_c(\theta)\tau}$ by making use of the Taylor series expansion, i.e.

$$e^{A_c(\theta)\tau} = I + A_c(\theta) + \frac{A_c^2(\theta)\tau^2}{2!} + \frac{A_c^3(\theta)\tau^3}{3!} + \ldots. \tag{3.27}$$

Note that the usage of (3.27) implies that the discrete-time system is an approximation of the continuous-time model even if (3.24) holds. Another possibility to convert a continuous-time system to a discrete form is to describe the input as a series of pulses. This is carried out via a sampler, which samples and then holds the input over a specified sampling interval T_s. Once a given sampling period expires, the sampler discards the stored value of the input and acquires a new value. The overall procedure is realised by a so-called zero-order hold (ZOH), whose transfer function is given by

$$G_{\text{ZOH}}(s) = \frac{1}{s} - \frac{1}{s}e^{-sT_s} = \frac{1 - e^{-sT_s}}{s}. \tag{3.28}$$

By using (3.28) the corresponding discrete-time model can be calculated via

$$G(z) = \mathcal{Z}\left\{G_{\text{ZOH}}(s)G(s)\right\} = (1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\}. \tag{3.29}$$

Note that equation (3.28) can be written more conveniently as

$$G_{\text{ZOH}}(s) = \frac{1 - z^{-1}}{s}, \tag{3.30}$$

where $z = e^{sT_s}$. Alternatively to the ZOH, which is regarded as an interpolator of a zero order, also higher order interpolators can be utilised. Whilst the ZOH provides a rectangular approximation of signals sampled, for instance a first order interpolation results in a triangular approximation. The transfer functions of a first order interpolator, referred to as the first order hold (FOH), is defined by

$$G_{\text{FOH}} = \frac{(1 - z^{-1})^2}{T_s z^{-1} s}. \tag{3.31}$$

In the case when the transfer function of a discrete-time model is available (for instance it has been estimated via some identification technique), the corresponding continuous-time model, for the ZOH method, can be obtained from

$$G(s) = \frac{\mathcal{L}\{G(z)\}}{G_{\text{ZOH}}(s)} \tag{3.32}$$

and, analogously, for the FOH method from

$$G(s) = \frac{\mathcal{L}\{G(z)\}}{G_{\text{FOH}}(s)}. \tag{3.33}$$

Impulse response representation

Another possibility to represent a linear system is the so-called impulse response. In fact an LTI casual system can be completely characterised by its impulse response only, see (Ljung 1999). Considering a continuous-time system, an impulse response is defined via a weighting sequence $\{g(\tau)\}_{\tau=0}^{\infty}$, which is the response of a relaxed[2] system to an excitation by the Dirac delta function. Loosely speaking, an impulse response is a reaction in time of a relaxed dynamic system to some very brief external excitation or disturbance. As an example consider a car moving forward in a centre of a road with a constant velocity which is suddenly stricken by some object from a lateral direction. If a momentum of the object is relatively small compared to that of the car, the result of this external excitation is that a driver will by manoeuvring the car left and right direction, until the car is in its initial position, i.e. in the centre of a road. By knowing the weighting sequence and the input $u(s)$ for $s \leq t$, the output $y(s)$ with $s \leq t$ to an arbitrary input signal (because any input can be considered as being a sum of impulses) can be calculated via the following convolution integral

$$\begin{aligned} y(t) = (u * g)(t) &= \int_{\tau=0}^{\infty} g(\tau)u(t-\tau)d\tau \\ &= \int_{\tau=-\infty}^{t} g(t-\tau)u(\tau)d\tau, \end{aligned} \tag{3.34}$$

where τ is a dummy variable and t corresponds to a time offset. It is assumed in (3.34) that initial conditions are null. Since the system is postulated to

[2] A casual system is said to be relaxed if no energy is stored in the system, i.e. all initial conditions are null.

be casual, the response is null before excitation, i.e. $u(t) = 0\ \forall t < 0$, hence $g(\tau) = 0\ \forall \tau < 0$. By assuming that the input is piece-wise constant between sampling instances, cf. (3.24), an exact discrete-time equivalent of equation (3.34) is the convolution summation given by

$$\begin{aligned} y_k = (u * g)_k &= \sum_{l=0}^{\infty} g_l u_{k-l} \\ &= \sum_{l=-\infty}^{k} u_l g_{k-l}, \end{aligned} \tag{3.35}$$

where $\{g_l\}_{l=0}^{\infty}$ and $k = 0, 1, \ldots$. The relationship between the system transfer function and its impulse response, for a discrete-time case, is obtained via the observation that the transfer function is, in fact, an infinite sum of the weighting sequence, see (Ljung 1999), i.e.

$$y_k = \sum_{l=0}^{\infty} g_l \left(q^{-l} u_k\right) = \left[\sum_{l=0}^{\infty} g_l q^{-l}\right] u_k = G(q) u_k, \tag{3.36}$$

where

$$G(q) = \sum_{k=0}^{\infty} g_k q^{-k}. \tag{3.37}$$

Consequently, the transfer function is related to the weighting sequence via

$$G(z) = \sum_{k=0}^{\infty} g_k z^{-k}. \tag{3.38}$$

It is remarked that although theoretically appealing, the description of the impulse response requires specification of an infinite number of parameters. From a pragmatic point of view it is considerably more convenient to parametrise a system in terms of a finite number of variables (Ljung 1999). Consequently, system description via a rational transfer function, cf. (3.15), and state-space equations, cf. (3.19)-(3.20), is preferable in most practical cases.

Simple process identification example

As a relatively simple example of system identification a process of an exponential radioactive decay is considered. A given substance undergoes an exponential decay if the mass of that substance changes, i.e. decreases, at a rate which is proportional to the mass. Such a process can be described by the following ordinary differential equation of first order:

$$\frac{dm(t)}{dt} = -\frac{1}{\tau}m(t), \tag{3.39}$$

where $m(t)$ denotes the mass and τ is a time constant of the process. A solution to (3.39) is given by

$$m(t) = m_0 e^{-\frac{t}{\tau}}, \tag{3.40}$$

where m_0 denotes the initial mass at time $t = 0$, i.e. $m(0) = m_0$. Note that (3.40) is characterised completely by two parameters only, i.e. m_0 and τ. Imagine the task consists of the following: having measured data describing an exponential decay of some unknown material, model such a process and potentially identify the actual radioactive substance using the associated time constant. The time constant is related to the so-called half-life, denoted $t_{1/2}$, which can be used to characterise a particular radioactive material. This is because the half-time corresponds to a time taken for half of the initial mass to decay. It is related to the time constant via $t_{1/2} = \tau \ln 2$. Figure 3.1 shows an exemplary plot of the exponential decay process, i.e. the grey line, and the data points which were actually measured, i.e. circles, using a relatively inaccurate measuring device. For the purpose of identification only noisy data are available, the actual noise-free data are unavailable. It is assumed that neither m_0 nor $t_{1/2}$ are known *a priori.* The white-box approach would use the initial mass of the material, m_0, and its half-life, $t_{1/2}$, to construct the model using equation (3.39). The measured data would only be used to verify the model is correct. However, the material type is unknown so this method cannot be used. The grey-box approach would be to obtain as much information as possible and estimate the remaining unknowns from the measured data. For instance, one could confirm that the process in question is indeed of an exponential nature and that it is due to only one material undergoing a radioactive decay. The value of the initial mass of the material with the half-time would have to be determined from the data. Additionally, it should be noted that it is not feasible for mass to be negative, therefore all such data points should be pretreated. For example, the data could be processed and all measurements where $m(t) < 0$ set to null. The black-box approach would attempt to infer all the information from the measured data only, including the nature of the decay, i.e. the exponential characteristic, as well as its order. (Note that in the ideal, i.e. noise-free, case

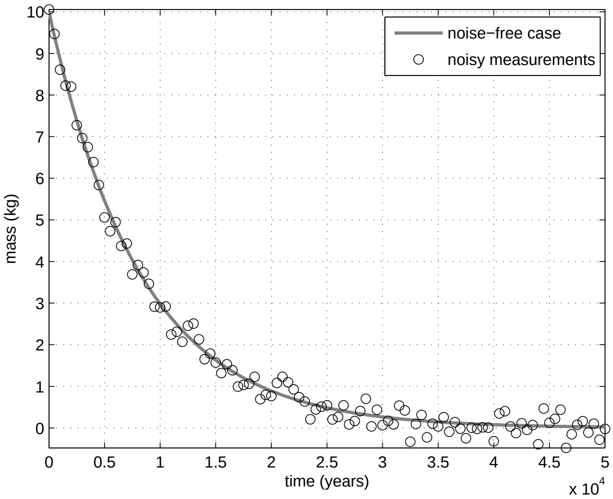

Figure 3.1: An exponential decay process of unknown radioactive substance.

the unknowns m_0 and $t_{1/2}$ can be found by knowing only a single arbitrary pair of measurements.)

Here, the grey-box approach is followed. First, the analysis of data suggests that the process is of first order, because the response does not overshoot the zero y-axis (which is rather obvious from a physical point of view as mass cannot be negative) and the gradient at zero of x-axis is nonzero. The initial mass is obtained from the plot at $t = 0$, i.e. $\hat{m}_0 = m(0) = 10.05$, and the time constant is found by determining a time for which the value of mass reaches approximately 37% of its initial value, i.e. $0.37\hat{m}_0 = 3.72$, hence $\hat{\tau} = 7500$.

An alternative approach would be to note that using a logarithmic transformation equation (3.40) can be expressed as

$$\ln m(t) = \ln m_0 - \frac{t}{\tau}. \tag{3.41}$$

The estimate of m_0 can again be obtained directly from the plot and the time constant can be found from

$$\tau = -\frac{t}{\ln \frac{m(t)}{m_0}}. \tag{3.42}$$

Because equation (3.42) provides a value of τ for every data point, τ can be calculated as their mean value yielding the estimate

$$\hat{\tau} = -\frac{1}{N}\sum_{t=1}^{N} \frac{t}{\ln \frac{m(t)}{\hat{m}_0}}, \tag{3.43}$$

where N is the total number of measurements taken. It can be expected that the accuracy yielded by (3.43) is better than that of the previous method because all measurements are used for the estimation of τ and not only a single data point. In this case the time constant is estimated to be $\hat{\tau} = 8109.5$. A visual comparison of the results obtained by these two approaches is shown in Figure 3.2, where, for completeness the noise-free and measured data are also given. It is observed that the latter approach is superior to the former, because the model obtained fits data more precisely. In fact, it almost completely coincides with the noise-free data. The corresponding half-time is 5621.1 years. Because it is known that the half-life of Carbon-14 is 5730 ± 40 years, it is quite likely that this set of data corresponds to the exponential decay of Carbon-14. The

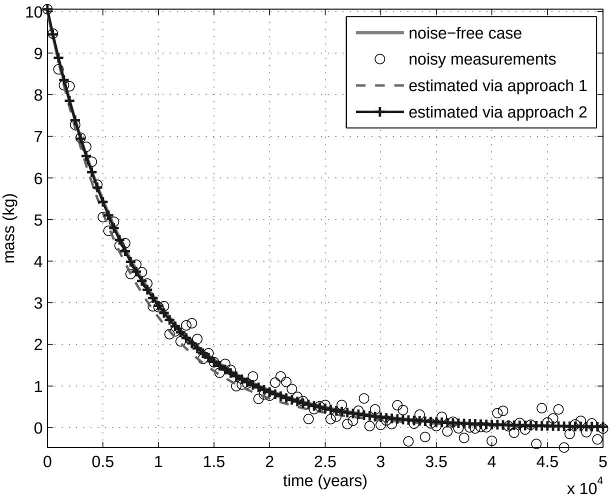

Figure 3.2: A visual comparison of the two estimated models against the measured data and the noise-free data.

relative error with respect to the half-life parameter is only approximately 1.9%. In the case of the first approach the corresponding half-life parameter is 5198.6 years, hence the relative error is of approximately 9.3%. Note that although the estimate of τ has improved, the estimate of m_0 remained the same with the relative error of approximately 0.5%. In fact, accuracies of both parameters can be improved further via computer aided estimation procedures described in later chapters by using, for instance, the least squares method.

3.2.3 Linear model structures

In this subsection some commonly used discrete-time dynamic LTI model structures are reviewed. In practice the notion of a ‘true’ system does not exist and means to account for this fact are necessary. A widely used method is to add signals (usually assumed to be random and unknown) to the input and output signals of the model. Their task is to absorb (or account for) the mismatch between the actual measured data and the data produced by the model.

In general, the model structures which will be considered can all be written in the following form:

$$y_k = G(q;\theta)u_k + H(q;\theta)e_k, \tag{3.44}$$

where e_k is a noise (or disturbance) sequence representing uncertainties within the model. The transfer functions[3] $G(q;\theta)$ and $H(q;\theta)$ relate the input u_k to the output y_k and the disturbances e_k to the output y_k, respectively. They are both assumed to be stable and of finite orders. Note that $G(q;\theta)$ and $H(q;\theta)$ are both explicit functions of the parameter vector $\theta \in \mathbb{R}^{n_\theta}$. Here $G(q;\theta)$ and $H(q;\theta)$ are rational functions of some polynomials (i.e. comprising of a ratio of two polynomials), therefore θ contains their coefficients. The vector θ parametrises the model, or more precisely the transfer functions $G(q;\theta)$ and $H(q;\theta)$. A particular model can be obtained from the general structure (3.44) by specifying $G(q;\theta)$, $H(q;\theta)$ and the probability density function of the sequence e_k, see (Ljung 1999). In practice, it is best to specify e_k in as simple terms as possible. Therefore e_k is usually assumed to be a random, zero-mean, white sequence that is uncorrelated with the input (thus also uncorrelated with the output). Additionally, it is postulated that e_k is Gaussian distributed and can

[3] Formally the term transfer function should be used with the $\mathcal{Z}$-transform only, but here, for simplicity, no differentiation is made and transfer function is used with both, i.e. $G(z)$ and $G(q)$.

be completely characterised by the two first moments only, i.e.

$$E[e_k] = 0 \qquad \text{(mean)}, \tag{3.45}$$

$$E[e_k^2] = \sigma_e^2 \qquad \text{(variance)}. \tag{3.46}$$

Summarising, the definition of a given model involves specification of: i) the parameter vector θ and ii) the noise variance σ_e^2.

Auto-regressive with exogenous input

Probably the simplest dynamic model structure is the auto-regressive with exogenous[4] input (ARX) structure defined by

$$A(q^{-1})y_k = B(q^{-1})u_k + e_k. \tag{3.47}$$

Note the lack of an exact relationship in (3.47) between y_k and u_k. This additional degree of freedom is accounted for by the introduction of sequence e_k. Because the signal e_k enters as a direct error into the difference equation (3.47), the ARX model structure belongs to the category of so-called equation error (EE) models and e_k is called the equation error. A block diagram of the ARX model structure is shown in Figure 3.3, where the arguments of polynomials are ignored for convenience. By considering the ARX model structure within the

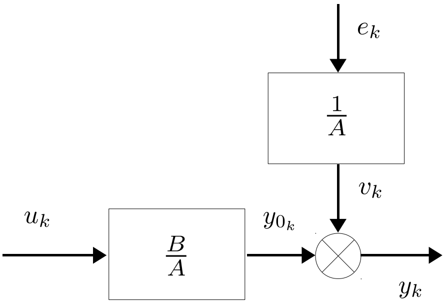

Figure 3.3: The ARX model structure.

[4] The term exogenous means that this signal enters into the system from outside and represents a manipulated time-varying process variable, i.e. a reference signal.

general model framework (3.44), the following relationships follow, i.e.

$$G(q;\theta) = \frac{B(q^{-1})}{A(q^{-1})}, \tag{3.48}$$

$$H(q;\theta) = \frac{1}{A(q^{-1})}, \tag{3.49}$$

which can be verified by re-expressing (3.47) as

$$y_k = \frac{B(q^{-1})}{A(q^{-1})}u_k + \frac{1}{A(q^{-1})}e_k. \tag{3.50}$$

The parameter vector is given by

$$\theta = \begin{bmatrix} a_1 & \ldots & a_{n_a} & b_0 & \ldots & b_{n_b} \end{bmatrix}^T \in \mathbb{R}^{n_\theta}, \tag{3.51}$$

where $n_\theta = n_a + n_b + 1$. Because by definition the coefficient a_0 is unity it is not included in θ. Note, that because e_k is a stochastic process hence, the overall ARX model is also stochastic, even though the input, u_k, is deterministic.

Auto-regressive moving average with exogenous input

The equation error e_k should not only include effects of measurement noise but also uncaptured dynamics, unmodelled nonlinearities, unmeasured inputs and etc., i.e. the combined effect of all uncertainties. In the ARX case, equation (3.47), all these effects are modelled jointly by the single term e_k, which, in practice, may lack sufficient flexibility. The degrees of freedom in the description of the input-output mismatch can be increased by allowing the overall equation error sequence to be coloured. One method to achieve this is to use the auto-regressive moving average with exogenous input (ARMAX) model. This models the disturbance by a moving average (MA) process, i.e.

$$A(q^{-1})y_k = B(q^{-1})u_k + C(q^{-1})e_k, \tag{3.52}$$

where

$$C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c}. \tag{3.53}$$

Note that in this case the input-output mismatch is accounted by a coloured sequence $C(q^{-1})e_k$. Because, as with the ARX case, the noise sequence enters the difference equation directly, the ARMAX model also belongs to the class of

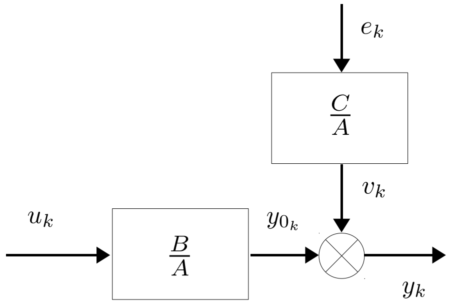

Figure 3.4: The ARMAX model structure.

EE models. A block diagram depicting the ARMAX model structure is given in Figure 3.4. In this case the parameter vector is defined as

$$\theta = \begin{bmatrix} a_1 & \dots & a_{n_a} & b_0 & \dots & b_{n_b} & c_1 & \dots & c_{n_c} \end{bmatrix}^T \in \mathbb{R}^{n_\theta} \tag{3.54}$$

with $n_\theta = n_a + n_b + n_c + 1$. In terms of the general model structure (3.44) the ARMAX model corresponds to

$$G(q;\theta) = \frac{B(q^{-1})}{A(q^{-1})}, \tag{3.55}$$

$$H(q;\theta) = \frac{C(q^{-1})}{A(q^{-1})}, \tag{3.56}$$

which can be verified by re-writing (3.52) as

$$y_k = \frac{B(q^{-1})}{A(q^{-1})}u_k + \frac{C(q^{-1})}{A(q^{-1})}e_k. \tag{3.57}$$

It is interesting to consider some specific models which all are special cases of the ARMAX structure, see (Söderström & Stoica 1989):

- Choosing $n_b = n_c = 0$ leads to auto-regression (AR), i.e.

  $$A(q^{-1})y_k = e_k. \tag{3.58}$$

  Note that the model (3.58) is not driven by a manipulated (or controlled) input and the only innovation is due to noise.

- Choosing $n_a = n_b = 0$ leads to moving average (MA) model, i.e.

$$y_k = C(q^{-1})e_k. \tag{3.59}$$

- Choosing $n_b = 0$ leads to auto-regressive moving average (ARMA) model, i.e.

$$A(q^{-1})y_k = C(q^{-1})e_k. \tag{3.60}$$

- Choosing $n_a = n_c = 0$ leads to finite impulse response (FIR) model, i.e.

$$y_k = B(q^{-1})u_k + e_k. \tag{3.61}$$

- Choosing $n_c = 0$ leads to ARX model defined by equation (3.47).

The ARMAX model structure is especially useful to model systems in which disturbances enter the system relatively close to the manipulated input.

Auto-regressive integrated moving average with exogenous input

In cases where a slow random drift is present a MA process of the ARMAX model may not be appropriate for representing the disturbances. In such circumstances an additional integral action is included, i.e.

$$v_k = v_{k-1} + \frac{C(q^{-1})}{A(q^{-1})}e_k = \frac{C(q^{-1})}{\Delta A(q^{-1})}e_k, \tag{3.62}$$

where $\Delta = 1 - q^{-1}$. This leads to an auto-regressive integrated moving average with exogenous input (ARIMAX) model given by

$$y_k = \frac{B(q^{-1})}{A(q^{-1})}u_k + \frac{C(q^{-1})}{\Delta A(q^{-1})}e_k. \tag{3.63}$$

Auto-regressive auto-regressive with exogenous input

Instead of modelling the equation error as a MA process, it can be modelled as an AR process, i.e.

$$A(q^{-1})y_k = B(q^{-1})u_k + \frac{1}{D(q^{-1})}e_k, \tag{3.64}$$

where

$$D(q^{-1}) = 1 + d_1 q^{-1} + \ldots + d_{n_d} q^{-n_d}. \tag{3.65}$$

In accordance with the terminology introduced, this is called an auto-regressive auto-regressive with exogenous input (ARARX) model, where the second AR term corresponds to the description of disturbances. As in the ARMAX case, the equation error sequence $\frac{1}{D(q^{-1})} e_k$ is coloured. The ARARX model structure is depicted in Figure 3.5. Alternatively, using block manipulations, it can be represented as shown in Figure 3.6. In this case the parameter vector is defined as

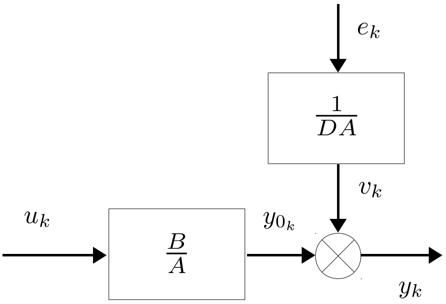

Figure 3.5: The ARARX model structure.

$$\theta = \begin{bmatrix} a_1 & \ldots & a_{n_a} & b_0 & \ldots & b_{n_b} & d_1 & \ldots & d_{n_d} \end{bmatrix}^T \in \mathbb{R}^{n_\theta} \tag{3.66}$$

with $n_\theta = n_a + n_b + n_d + 1$. With reference to the general model structure (3.44), the ARARX model corresponds to

$$G(q;\theta) = \frac{B(q^{-1})}{A(q^{-1})}, \tag{3.67}$$

$$H(q;\theta) = \frac{1}{A(q^{-1})D(q^{-1})}, \tag{3.68}$$

which can be verified by re-writing (3.64) as

$$y_k = \frac{B(q^{-1})}{A(q^{-1})} u_k + \frac{1}{A(q^{-1})D(q^{-1})} e_k. \tag{3.69}$$

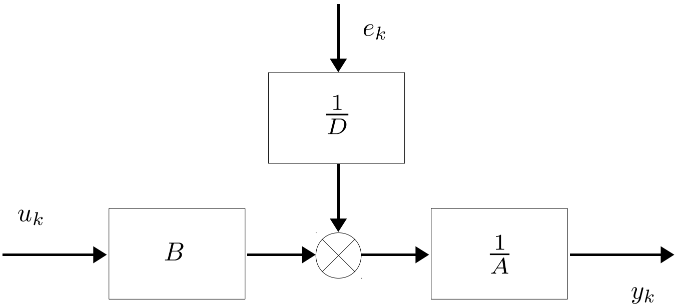

Figure 3.6: An alternative representation of the ARARX model structure.

Furthermore, since a MA process can be approximated arbitrarily closely to the AR process, i.e.

$$C(q^{-1}) \simeq \frac{1}{D(q^{-1})}, \tag{3.70}$$

the ARARX structure can be seen as an approximation of the ARMAX structure and vice-versa.

General equation error type structure

All EE type model structures can be seen to be special cases of the following structure

$$A(q^{-1})y_k = B(q^{-1})u_k + \frac{C(q^{-1})}{D(q^{-1})}e_k, \tag{3.71}$$

which is called auto-regressive auto-regressive moving average with exogenous input (ARARMAX). It allows an increased flexibility in describing the properties of the disturbances, since the equation error is modelled by an ARMA process, see Figure 3.7. Recalling the general model structure (3.44), the ARARMAX model corresponds to

$$G(q;\theta) = \frac{B(q^{-1})}{A(q^{-1})}, \tag{3.72}$$

$$H(q;\theta) = \frac{C(q^{-1})}{A(q^{-1})D(q^{-1})}, \tag{3.73}$$

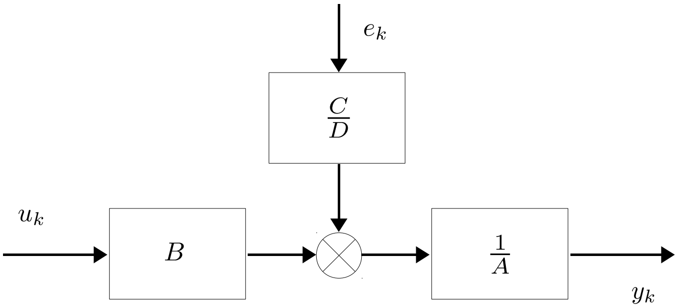

Figure 3.7: The EE model structure.

which can be verified by re-writing (3.64) as

$$y_k = \frac{B(q^{-1})}{A(q^{-1})}u_k + \frac{C(q^{-1}}{A(q^{-1})D(q^{-1})}e_k. \tag{3.74}$$

Output error

Note that in the case of the EE type structures both transfer functions $G(q;\theta)$ and $H(q;\theta)$ have a common polynomial in their corresponding denominators, i.e. the polynomial $A(q^{-1})$. However, the common polynomial in the denominator of both transfer functions may be difficult to justify from a physical point of view, see (Ljung 1999). Consequently, an alternative family of model structures, called the output error (OE) models, can be considered, where the transfer functions are parametrised independently, i.e. they do not share common polynomials.

In this case it is assumed that there is an exact relationship between the input u_k and the undisturbed (noise-free) output y_{0_k}, which is unmeasurable and available only via the noisy signal y_k. This leads to the following system setup, i.e.

$$A(q^{-1})y_{0_k} = B(q^{-1})u_k, \tag{3.75}$$

$$y_k = y_{0_k} + e_k. \tag{3.76}$$

By considering the OE model structure within the general model framework

(3.44), one obtains the following relationships

$$G(q;\theta) = \frac{B(q^{-1})}{A(q^{-1})}, \tag{3.77}$$

$$H(q;\theta) = e_k, \tag{3.78}$$

which can be verified by re-writing (3.75)-(3.76) as

$$y_k = \frac{B(q^{-1})}{A(q^{-1})} u_k + e_k. \tag{3.79}$$

Note that the error e_k is added to the noise-free output of the system y_{0_k}, which is exactly the reason why this model structure belongs to a class of OE models. A block diagram of the OE model structure (3.75)-(3.76) is given in Figure 3.8.

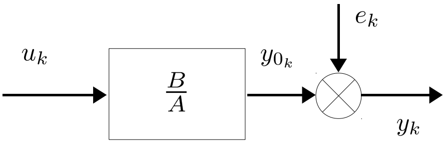

Figure 3.8: The OE model structure.

Box-Jenkins

A natural extension of the basic OE configuration is for the output error to be described as a separate transfer function, which is possibly different to that relating u_k to y_{0_k}. In the case when the output error is defined as an ARMA process, this leads to the so-called Box-Jenkins (BJ) model structure. With reference to (Ljung 1999), such description seems to be the most natural as it allows to separate parametrisations of the transfer functions $G(q;\theta)$ and $H(q;\theta)$. The BJ model structure is given by

$$A(q^{-1})y_{0_k} = B(q^{-1})u_k, \tag{3.80}$$

$$y_k = y_{0_k} + \frac{C(q^{-1})}{D(q^{-1})} e_k. \tag{3.81}$$

Consideration of the BJ model structure within the general model framework (3.44) leads to the following relationships

$$G(q;\theta) = \frac{B(q^{-1})}{A(q^{-1})}, \tag{3.82}$$

$$H(q;\theta) = \frac{C(q^{-1})}{D(q^{-1})}, \tag{3.83}$$

which can be verified by re-writing (3.80)-(3.81) as

$$y_k = \frac{B(q^{-1})}{A(q^{-1})} u_k + \frac{C(q^{-1})}{D(q^{-1})} e_k. \tag{3.84}$$

Note that in this case both transfer functions are described as separate ARMA processes. That means the description of disturbances is completely independent of the system dynamics. A diagrammatic description of the BJ model structure is given in Figure 3.9. It is instructive to note that some of the model structures introduced so far can be considered as being special cases of the BJ model. More precisely the OE is obtained by setting $C(q^{-1}) = D(q^{-1}) = 1$, whilst the EE type structures can be obtained by choosing $D(q^{-1}) = A(q^{-1})$, which yields the ARMAX model, and by setting $C(q^{-1}) = 1$ leading to the ARX model. Consequently, the EE type structures ARX and ARMAX can be interpreted as the BJ structures with an additional constraint imposed that $D(q^{-1}) = A(q^{-1})$. Furthermore, if the OE type model structures can be re-written in a form of the the EE type model structures and vice-versa, the same algorithms can be utilised for the estimation of their parameters.

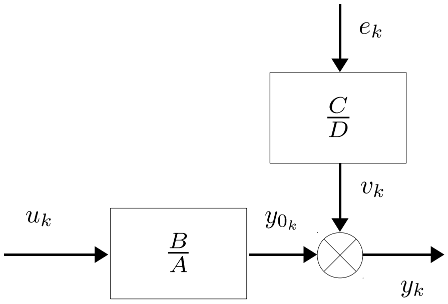

Figure 3.9: The BJ model structure.

Pragmatic approach for treatment of model uncertainty

The term 'measurement noise' refers to unavoidable uncertainties arising when signal is measured by a sensor, whilst the term 'process noise' corresponds to a model uncertainty arising from the fact that every model is just an approximation of the actual physical process. These concepts are illustrated in an upper block diagram in Figure 3.10, where d_k and $\tilde{y}_k$ denote the process noise and measurement noise, respectively. In practice, these two sources of uncertainties can be lumped together yielding an equivalent sequence of disturbances that accounts for the effects of both, i.e. the process and measurement noise. A setup illustrating such a pragmatic approach is given in a lower block diagram in Figure 3.10, where v_k is the equivalent noise sequence. Although this approach is dominant for the purpose of system identification, it is not the case when dealing with the task of filtering, where, in fact, it is preferable to distinguish between the process and measurement noise disturbances.

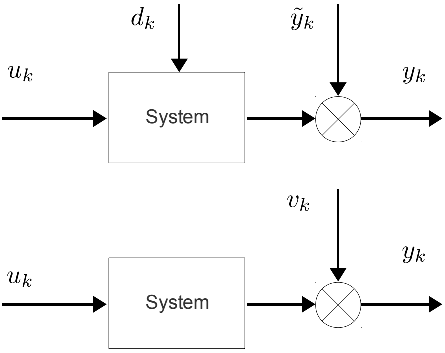

Figure 3.10: Two equivalent system configurations illustrating notions of process and measurement noise.

3.2.4 Prediction and simulation

A predictor for a given system is a model of that system which allows to predict the future system output based on past outputs and current and past inputs. Prediction can be made one-step ahead, n-steps ahead with $n < N$, or N-steps ahead. The N-steps ahead prediction is referred to as simulation. A diagrammatic representations of the one-step/n-steps ahead prediction and the simulation are given in Figures 3.11 and 3.12, respectively, where $\hat{y}_k$ is the predictor output and v_k denotes disturbances added to the system noise-free output. An optimal predictor yields the best prediction in some pre-defined sense. Since the predictor is defined based on a model of the system, its predicting capabilities depend on the goodness of that model.

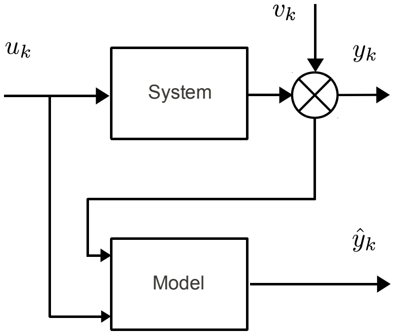

Figure 3.11: The setup for a one-step/n-steps ahead ahead prediction.

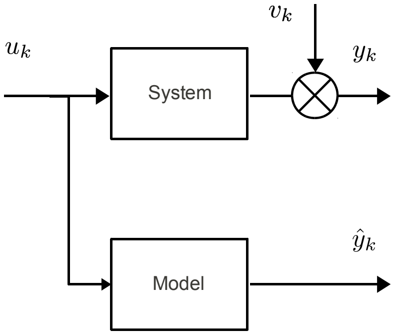

Figure 3.12: The setup for system simulation.

Consider a general structure of a discrete-time SISO LTI system given by (3.44), see Subsection 3.2.3, where the same assumptions are made with respect to the noise sequence e_k, i.e. that it is white, zero-mean Gaussian and uncorrelated with the input, see equations (3.45)-(3.46). With reference to (Ljung 1999), it can be demonstrated that the optimal one-step ahead predictor, denoted $\hat{y}_k$, in the following sense

$$\hat{y}_k = \arg\min_{\hat{y}_k^*} E\left[(y_k - \hat{y}_k^*)^2\right], \tag{3.85}$$

where $\hat{y}_k^*$ denotes some arbitrary predictor, is given by

$$\hat{y}_k = H^{-1}(q;\theta)G(q;\theta)u_k + \left[1 - H^{-1}(q;\theta)\right] y_k. \tag{3.86}$$

Note that this formulation requires for $H(q;\theta)$ to be invertible, i.e. non minimum-phase. In order to verify optimality of the predictor defined by (3.86),equation (3.44) is re-expressed to separate e_k from the regular part of the system, i.e.

$$\begin{aligned} y_k &= G(q;\theta)u_k + [H(q;\theta) - 1]\, e_k + e_k \\ &= G(q;\theta)u_k + [H(q;\theta) - 1]\, H^{-1}(q;\theta)\, [y_k - G(q;\theta)u_k] + e_k \\ &= H^{-1}(q;\theta)G(q;\theta)u_k + \left[1 - H^{-1}(q;\theta)\right] y_k + e_k \\ &= z_k + e_k, \end{aligned} \tag{3.87}$$

where

$$z_k = H^{-1}(q;\theta)G(q;\theta)u_k + \left[1 - H^{-1}(q;\theta)\right] y_k. \tag{3.88}$$

Substitution of expression (3.87) into (3.85) leads to

$$\begin{aligned} \hat{y}_k &= \arg\min_{\hat{y}_k^*} E\left[(z_k + e_k - \hat{y}_k^*)^2\right] \\ &= \arg\min_{\hat{y}_k^*} \left\{E\left[(z_k - \hat{y}_k^*)^2\right] + \sigma_e^2\right\}. \end{aligned} \tag{3.89}$$

The second equality follows from the fact that e_k is uncorrelated with z_k and it must also be uncorrelated with $\hat{y}_k^*$ (since $\hat{y}_k^*$ utilises only past values of output). The expression minimised is always greater than σ_e^2 and equal to σ_e^2 only in the case when $\hat{y}_k^* = z_k$. This shows in turn that the optimal predictor $\hat{y}_k = z_k$, hence it is given by expression (3.86).

Predictor for ARX model

By recalling formula (3.86), the one-step ahead predictor for ARX model structure is given by

$$\begin{aligned} \hat{y}_k &= B(q^{-1})u_k + \left[1 - A(q^{-1})\right] y_k \\ &= \varphi_k^T \theta, \end{aligned} \tag{3.90}$$

where

$$\varphi_k = \begin{bmatrix} -y_{k-1} & \dots & -y_{k-n_a}, & u_k & \dots & u_{k-n_b} \end{bmatrix}^T \in \mathbb{R}^{n_\theta}, \tag{3.91}$$

$$\theta = \begin{bmatrix} a_1 & \dots & a_{n_a} & b_0 & \dots & b_{n_b} \end{bmatrix}^T \in \mathbb{R}^{n_\theta} \tag{3.92}$$

with $n_\theta = n_a + n_b + 1$. Note that this is identical to the corresponding regression form of (3.50), see Subsection 3.2.3, but with e_k discarded, since a natural prediction of e_k is to assume that it is null (due to the property that $E[e_k] = 0$).

Predictor for ARMAX model

By recalling formula (3.86), the predictor for ARMAX model structure is given by

$$\hat{y}_k = \frac{B(q^{-1})}{C(q^{-1})} u_k + \left[1 - \frac{A(q^{-1})}{C(q^{-1})}\right] y_k. \tag{3.93}$$

Consider (3.93) in a more convenient form, i.e.

$$\begin{aligned} C(q^{-1})\hat{y}_k &= B(q^{-1})u_k + \left[C(q^{-1}) - A(q^{-1})\right] y_k, \\ C(q^{-1})\hat{y}_k + \left[1 - C(q^{-1})\right] \hat{y}_k &= B(q^{-1})u_k + \left[C(q^{-1}) - A(q^{-1})\right] y_k \\ &\quad + \left[1 - C(q^{-1})\right] \hat{y}_k. \end{aligned} \tag{3.94}$$

This leads to

$$\begin{aligned} \hat{y}_k &= B(q^{-1})u_k + C(q^{-1})y_k - A(q^{-1})y_k + \hat{y}_k - C(q^{-1})\hat{y}_k \\ &= B(q^{-1})u_k + C(q^{-1})y_k - A(q^{-1})y_k + \hat{y}_k - C(q^{-1})\hat{y}_k + y_k - y_k, \end{aligned} \tag{3.95}$$

which can be transformed into

$$\begin{aligned} \hat{y}_k &= B(q^{-1})u_k + \left[1 - A(q^{-1})\right] y_k + \left[C(q^{-1}) - 1\right] \varepsilon_k(\theta) \\ &= \varphi_k^T(\theta)\theta, \end{aligned} \tag{3.96}$$

where the (model dependent) residuals are defined by

$$\varepsilon_k(\theta) = y_k - \hat{y}_k. \tag{3.97}$$

The regressor and the parameter vector are given, respectively, by

$$\begin{aligned} \varphi_k(\theta) = \big[-y_{k-1} \quad \dots \quad -y_{k-n_a} \quad u_k \quad \dots \quad u_{k-n_b} \\ \varepsilon_{k-1}(\theta) \quad \dots \quad \varepsilon_{k-n_c}(\theta) \big]^T \in \mathbb{R}^{n_\theta}, \end{aligned} \tag{3.98}$$

$$\theta = \begin{bmatrix} a_1 & \dots & a_{n_a} & b_0 & \dots & b_{n_b} & c_1 & \dots & c_{n_c} \end{bmatrix}^T \in \mathbb{R}^{n_\theta} \tag{3.99}$$

with $n_\theta = n_a + n_b + n_c + 1$. It is noted that (3.96) is similar to the corresponding regression form of (3.57). The differences are that, first, e_k is discarded and, second, past values of signal e_k are substituted by residuals $\varepsilon_k(\theta)$. This is due to that e_k is unmeasurable in practice, thus unknown, and have to be approximated using residuals, which through $\hat{y}_k$ are dependent on θ. In the case when θ is known exactly, in fact, no approximation is made, i.e. $\varepsilon_k(\theta) = e_k$. However, in practice θ must be estimated hence $\varepsilon_k(\hat{\theta}) \simeq e_k$ and depends on the goodness of $\hat{\theta}$. Furthermore, since the regressor vector comprises the residuals generated based on θ it is nonlinear in the parameter vector and hence, strictly speaking, equation (3.97) is no longer a linear regression. Nevertheless, to stress a close relationship, it is termed a pseudo-linear regression.

Predictor for OE model

With reference to formula (3.86), the predictor for OE model structure is given by

$$\hat{y}_k = \frac{B(q^{-1})}{A(q^{-1})} u_k = y_{0_k}(\theta). \tag{3.100}$$

Note that this predictor is constructed from the current and past values of the input exclusively. A corresponding pseudo-linear regression is given by

$$\hat{y}_k = \varphi_k^T(\theta)\theta, \tag{3.101}$$

where

$$\varphi_k(\theta) = \begin{bmatrix} -y_{0_{k-1}}(\theta) & \dots & -y_{0_{k-n_a}}(\theta), & u_k & \dots & u_{k-n_b} \end{bmatrix}^T \in \mathbb{R}^{n_\theta}, \tag{3.102}$$

$$\theta = \begin{bmatrix} a_1 & \dots & a_{n_a} & b_0 & \dots & b_{n_b} \end{bmatrix}^T \in \mathbb{R}^{n_\theta} \tag{3.103}$$

with $n_\theta = n_a + n_b + 1$. The sequence $y_{0_k}(\theta)$ although unobserved in practice can be calculated via (3.100). Moreover, (3.101) is in a formal agreement with the predictor for the ARMAX model structure, see equation (3.96).

Predictor for BJ model

With reference to formula (3.86), the predictor for BJ model structure is given by

$$\hat{y}_k = \frac{D(q^{-1})}{C(q^{-1})} \frac{B(q^{-1})}{A(q^{-1})} u_k + \left[1 - \frac{D(q^{-1})}{C(q^{-1})}\right] y_k \tag{3.104}$$

and it is equivalent to the following recursion

$$C(q^{-1})A(q^{-1})\hat{y}_k = A(q^{-1})\left[C(q^{-1}) - D(q^{-1})\right] y_k + D(q^{-1})B(q^{-1})u_k. \quad (3.105)$$

The prediction error $\varepsilon_k(\theta)$ is given by

$$\begin{aligned} \varepsilon_k(\theta) &= \frac{1}{A(q^{-1})C(q^{-1})}\left[A(q^{-1})D(q^{-1})y_k - D(q^{-1})B(q^{-1})u_k\right] \\ &= \frac{D(q^{-1})}{C(q^{-1})}\left[y_k - \frac{B(q^{-1})}{A(q^{-1})}u_k\right]. \end{aligned} \quad (3.106)$$

Introduce $y_{0_k}(\theta)$ denoting the regular part of the system, i.e.

$$y_{0_k}(\theta) = \frac{B(q^{-1})}{A(q^{-1})}u_k. \quad (3.107)$$

Additionally, introduce $v_k(\theta)$ defined by

$$v_k(\theta) = y_k - y_{0_k}(\theta). \quad (3.108)$$

This allows to express equation (3.106) as

$$\varepsilon_k(\theta) = \frac{D(q^{-1})}{C(q^{-1})}v_k(\theta). \quad (3.109)$$

Equations (3.107) and (3.109) can, respectively, be rewritten as

$$y_{0_k}(\theta) = -\left[A(q^{-1}) - 1\right] y_{0_k}(\theta) + B(q^{-1})u_k \quad (3.110)$$

and

$$\begin{aligned} \varepsilon_k(\theta) &= -\left[C(q^{-1}) - 1\right]\varepsilon_k(\theta) + D(q^{-1})v_k(\theta) \\ &= -\left[C(q^{-1}) - 1\right]\varepsilon_k(\theta) + \left[D(q^{-1}) - 1\right] v_k(\theta) + v_k(\theta). \end{aligned} \quad (3.111)$$

Consequently, by inserting (3.110) into (3.108) and then substituting the resulting expression into (3.111) one obtains

$$\begin{aligned} \varepsilon_k(\theta) = &-\left[C(q^{-1}) - 1\right]\varepsilon_k(\theta) + \left[D(q^{-1}) - 1\right] v_k(\theta) + y_k \\ &+ \left[A(q^{-1}) - 1\right] y_{0_k}(\theta) - B(q^{-1})u_k. \end{aligned} \quad (3.112)$$

Note that expressions (3.106), (3.107) and (3.109) imply that

$$\hat{y}_k = y_k - \varepsilon_k(\theta). \quad (3.113)$$

By inserting (3.112) into (3.113) the following difference equation is obtained, i.e.

$$\begin{aligned}\hat{y}_k =& -\left[A(q^{-1})-1\right] y_{0_k}(\theta) + B(q^{-1})u_k + \left[C(q^{-1})-1\right] \varepsilon_k(\theta) \\ & - \left[D(q^{-1})-1\right] v_k(\theta).\end{aligned} \tag{3.114}$$

Equation (3.114) can be written in a pseudo-linear regression form as follows

$$\hat{y}_k = \varphi_k^T(\theta)\theta, \tag{3.115}$$

where

$$\begin{aligned}\varphi_k(\theta) = \big[& -y_{0_{k-1}}(\theta) \quad \dots \quad -y_{0_{k-n_a}}(\theta) \quad u_k \quad \dots \quad u_{k-n_b} \\ & \varepsilon_{k-1}(\theta) \quad \dots \quad \varepsilon_{k-n_c}(\theta) \quad -v_{k-1}(\theta) \quad \dots \quad -v_{k-n_d}(\theta)\big]^T \in \mathbb{R}^{n_\theta},\end{aligned} \tag{3.116}$$

$$\theta = \begin{bmatrix} a_1 & \dots & a_{n_a} & b_0 & \dots & b_{n_b}\end{bmatrix}^T \in \mathbb{R}^{n_\theta} \tag{3.117}$$

with $n_\theta = n_a + n_b + n_c + n_d + 1$. Note that for the generation of the optimal one-step ahead prediction two auxiliary signals have to be computed, i.e. y_{0_k} and v_k, via expressions (3.107) and (3.108), respectively.

3.3 Nonlinear systems

Although it has been through linear systems and approaches based on linear models that have provided a fundamental and solid ground for control systems engineering, with the increased demands for wider operating ranges hence improved flexibility of models and potential for more precise descriptions of various phenomena, the need for appropriate nonlinear models has become a prominent and indeed an important topics of research in the control community. In general, all models as the name itself suggests provide only approximations to the actual, i.e. real-word, systems and natural phenomena. The degree of fidelity to which a model matches a given system will depend upon the purpose, hence the notion of 'model for purpose' must be borne in mind. Whilst the precision of these approximations as well as the notion of their adequateness are both strongly dependent on the particular application, it is required that models developed are also of a reasonable complexity (Pearson 1999). Loosely speaking, it is therefore expected for the identified model to be sufficiently flexible to capture the main and/or important dynamics of the system, whilst at the same time to be of parsimonious complexity, such that it can be handled by practically available hardware equipment. Whilst the general term 'nonlinear system' has a considerably broad meaning (as it does not reflect the exact form of the manifested nonlinearity), in this section the attention is drawn to particular classes of nonlinear systems, namely Hammerstein systems, Wiener systems, Hammerstein-Wiener systems, bilinear systems and nonlinear ARX (NARX) systems.

In the following, since the emphasis is placed on the qualitative input-output behaviour, it is assumed, for simplicity, that that there is no uncertainty present on the output measurements.

3.3.1 Hammerstein and Wiener systems

Hammerstein and Wiener systems both belong to a class of block oriented models where the nonlinear function is static, i.e. it has no memory. These systems are especially useful in situations where the dynamic behaviour of the process can be well described by an LTI model, whilst there are nonlinear effects present that influence the system input, output or both.

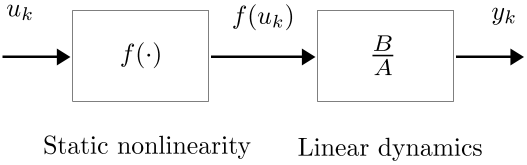

Figure 3.13: Structure of the Hammerstein model.

A Hammerstein model is given by

$$y_k = \frac{B(q^{-1})}{A(q^{-1})} f(u_k), \tag{3.118}$$

where $f(\cdot)$ denotes a general static nonlinear function. A diagrammatic representation of a Hammerstein system is given in Figure 3.13, where it is observed that it consists of a cascade connection of a static nonlinearity block followed by an LTI dynamic block. Note that the input undergoes a nonlinear transformation before entering the dynamic subsystem. A Hammerstein model structure is particularly useful to model nonlinear characteristics of system actuators.

A Wiener system is a dual of a Hammerstein system, which is obtained by reversing the order of the static nonlinearlity and the LTI block (Pearson & Pottmann 2000). A Wiener model structure is shown in Figure 3.14, where it is observed that it consists of a cascade of an LTI block followed by a static nonlinearity, and it can be described as follows

$$v_k = \frac{B(q^{-1})}{A(q^{-1})} u_k, \tag{3.119}$$

$$y_k = f(v_k). \tag{3.120}$$

Wiener systems are especially useful in cases when modelling systems with nonlinear characteristics of sensors.

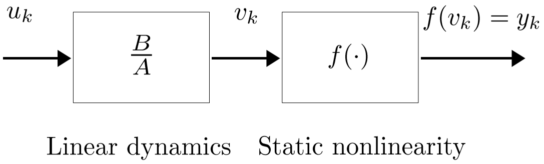

Figure 3.14: Structure of the Wiener model.

Note that both, i.e. Wiener and Hammerstein, models combine a dynamic LTI model with a nonlinear steady-state curve defined by the function $f(\cdot)$. However, although both can exhibit the same steady-state behaviour, their dynamic responses can be profoundly different, see (Pearson & Pottmann 2000).

By combining together the Hammerstein and Wiener model structures a so-called Wiener-Hammerstein model arises. It is given by a cascade of three blocks, namely, static nonlinearity on the system input, an LTI part and a static nonlinearity on the system output, where the two static nonlinearities are typically different. A Wiener-Hammerstein model is defined by

$$v_k = \frac{B(q^{-1})}{A(q^{-1})} f(u_k), \tag{3.121}$$

$$y_k = g(v_k), \tag{3.122}$$

where $g(\cdot)$ is a general static nonlinear function. It is remarked the in all three cases the transient behaviour, including stability properties, is governed by the LTI block exclusively, whilst the steady-state characteristic is given by the static nonlinear function(s).

3.3.2 Bilinear systems

Bilinear model structures, while retaining to large extent the well understood properties of linear models, such as time constants, damping/natural frequency and steady-state gain, are characterised by improved capabilities of replicating certain nonlinear phenomena. Due to aforementioned advantages as well as the parsimony of description bilinear system models can be considered as one possibility for permitting a satisfactory approximation to many nonlinear systems. Although being nonlinear in terms of input-output characteristics, they are still relatively closely related to linear models (via the possibility of their interpretation as LTV systems), and, therefore, are often considered as a stepping stone when modelling nonlinear systems. Especially when dealing with applications where there is heat exchange and/or transfer of heat is involved. Bilinear models can also arise when a nonlinear system is approximated by including linear and bilinear terms in a Taylor approximation series. Moreover, a bilinear model structure can often appear naturally, especially in the context of chemical processes. In such cases it is quite common for the exogenous inputs to be flow-rates. By choosing system states to correspond to concentrations of substances of interest and by considering a balance of energy, a model obtained comprises a (bilinear) product between the input and the state variables.

A discrete-time bilinear models can be defined using at least two forms, namely a state-space and input-output representation, see (Pearson 1999). However, it must be emphasised that these two representations are, in fact, not equivalent. In general, a given state-space bilinear system may not possess a corresponding input-output representation. In the state-space form, in which the bilinear systems have been originally proposed, the bilinearity is defined by a product between system state and control input. A discrete-time time-invariant SISO bilinear system can be described by:

$$x_{k+1} = A(\theta)x_k + B(\theta)u_k + u_k G(\theta)x_k, \qquad x_0 = \bar{x}_0, \tag{3.123}$$

$$y_k = C(\theta)x_k + D(\theta)u_k, \tag{3.124}$$

where $x_k \in \mathbb{R}^n$ denotes the state vector and $\bar{x}_0$ its initial value. The time-invariant matrices $A(\theta)$, $B(\theta)$, $C(\theta)$, $D(\theta)$ and $G(\theta)$ are of appropriate dimensions and characterise the dynamical behaviour of the system. It is to be noted that an input dependent (hence time-varying) system matrix can be expressed as

$$A_k(\theta) = A(\theta) + u_k G(\theta) \tag{3.125}$$

yielding input dependent steady-state and dynamic characteristics of the system.

A discrete-time time-invariant SISO bilinear system can also be represented by the following input-output difference equation, i.e.

$$A(q^{-1})y_k = B(q^{-1})u_k + \sum_{i=1}^{n_b}\sum_{j=1}^{n_a}\eta_{ij}u_{k-i}y_{k-j}, \tag{3.126}$$

where the polynomials $A(q^{-1})$ and $B(q^{-1})$ are defined identically as in the case of LTI systems.

In general, input-output bilinear system models can be partitioned into sub-, super- and diagonal categories, see (Pearson 1999) for details, i.e.

- Subdiagonal $\eta_{ij} = 0 \ \forall j > i$,
- Superdiagonal $\eta_{ij} = 0 \ \forall j < i$,
- Diagonal $\eta_{ij} = 0 \ \forall j \neq i$.

It is noted that, similarly to the state-space description (3.123)-(3.124), the input-output representation (3.126) can be re-expressed such that the resulting system is LTV with the input or, alternatively, the output dependency of the parameters. The corresponding LTV system with input dependent parameters is given by

$$y_k = -\sum_{j=1}^{n_a} a_j^k y_{k-j} + \sum_{i=0}^{n_b} b_i u_{k-i} = \left[1 - A_k(q^{-1})\right] y_k + B(q^{-1})u_k, \tag{3.127}$$

where the time-varying polynomial $A_k(q^{-1})$ comprises of the time-varying coefficients

$$a_j^k = a_j - \sum_{i=1}^{n_b} \eta_{ij}u_{k-i}. \tag{3.128}$$

Similarly, the corresponding LTV system with output dependent parameters is defined by

$$y_k = -\sum_{j=1}^{n_a} a_j y_{k-j} + \sum_{i=0}^{n_b} b_i^k u_{k-i} = \left[1 - A(q^{-1})\right] y_k + B_k(q^{-1})u_k, \tag{3.129}$$

where the time-varying polynomial $B_k(q^{-1})$ comprises of the time-varying coefficients

$$b_i^k = b_i + \sum_{j=1}^{n_a} \eta_{ij}y_{k-j} \tag{3.130}$$

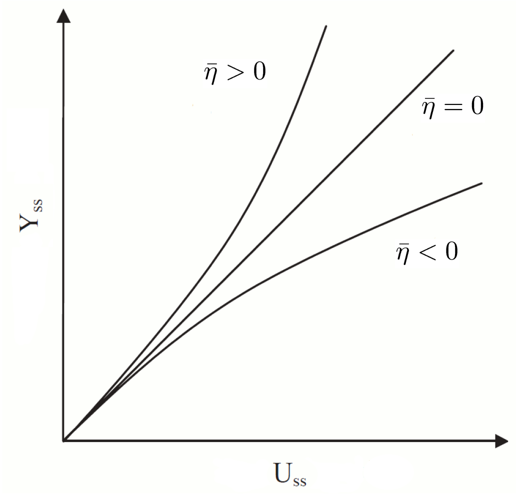

Figure 3.15: Steady-state input-output characteristic of bilinear system.

with $\eta_{0j} = 0 \; \forall j$.

Since the bilinear systems can be interpreted as LTV systems, the system dynamics (via the corresponding poles of the equivalent LTV system) is dependent on the input signal. Therefore, loosely speaking, the input must be specified such that the equivalent poles of an equivalent LTV system will remain within a unit disk. It is hence common to postulate that the input is confined within some specified upper and lower limits, say $\pm M$. With reference to (Pearson 1999) it can be shown that the discrete-time SISO bilinear system defined by equation (3.126) is stable if the following two conditions are satisfied, i.e.

$$|\lambda_j| < 1 \quad \forall \; j, \tag{3.131}$$

$$\sum_{i=1}^{n_b} \sum_{j=1}^{n_a} |\eta_{ij}| < \frac{\Pi_{j=1}^{n_a}(1 - |\lambda_j|)}{M}, \tag{3.132}$$

where λ_j are the roots of the polynomial $A(q^{-1})$.

The steady-state characteristic of the bilinear systems is given by

$$Y_{\text{ss}} = \frac{U_{\text{ss}}\bar{b}}{\bar{a} - U_{\text{ss}}\bar{\eta}}, \tag{3.133}$$

where Y_{ss} and U_{ss} are the steady-state output and steady-state input, respectively, and

$$\bar{a} = \sum_{j=0}^{n_a} a_j, \qquad \bar{b} = \sum_{i=0}^{n_b} b_i, \qquad \bar{\eta} = \sum_{i=1}^{n_b} \sum_{j=1}^{n_a} \eta_{ij}. \tag{3.134}$$

The steady-state input-output characteristics for the three different cases of the bilinear term $\bar{\eta}$ are illustrated in Figure 3.15. Clearly, if $\bar{\eta}$ is zero, equation (3.133) represents steady-state characteristics of a linear system, hence linear systems may be considered as a special subclass[5]. Positive values of $\bar{\eta}$ result in a gain which increases as U_{ss} increases, typical of exothermic chemical processes. Conversely, negative $\bar{\eta}$ produces a gain, which decreases as U_{ss} increases, leading to eventual saturation, and is typical of many industrial systems. Should a system exhibit bilinear characteristics of the form illustrated in Figure 3.15, then it is pertinent to consider adopting a bilinear systems modelling and control approach.

3.3.3 Class of NARX models

The general class of nonlinear ARX (NARX) models is defined, see (Pearson 1999), as follows

$$y_k = \mathcal{F}(y_{k-1}, \ldots, y_{k-n_a}, u_k, \ldots, u_{k-n_b}), \tag{3.135}$$

where $\mathcal{F}(\cdot)$ is a nonlinear function in $n_a + n_b + 1$ arguments. It is noted that the discrete-time index k is not present explicitly in the equation (3.135). A particular case of the NARX models is a class of so-called (structurally) additive NARX models where the nonlinear mapping $\mathcal{F}(\cdot)$ is constrained to be additive in its arguments. The additive NARX (NAARX) models are defined by

$$y_k = \sum_{j=1}^{n_a} g_j(y_{k-j}) + \sum_{i=0}^{n_b} f_i(u_{k-i}), \tag{3.136}$$

where $g_j(\cdot)$ and $f_i(\cdot)$ are static nonlinear functions.

The Hammerstein model structure belongs to the family of the NAARX (hence also NARX) models, where $g_j(x) = -a_j x$ and $f_i(x) = b_i f(x)$ with $f(\cdot)$

[5] However, note that $\bar{\eta} = 0$ does not imply that all η_{ij} are null and hence the system is linear.

being a static input nonlinearity of the Hammerstein model. Consequently, the NAARX representation of the Hammerstein model is given by

$$y_k = -\sum_{j=1}^{n_a} a_j y_{k-j} + \sum_{i=0}^{n_b} b_i f(u_{k-i}). \tag{3.137}$$

In contrast, Wiener models do not belong to the class of NARX (nor NAARX) models, since, in general, they cannot be directly expressed in a form of a difference equation relating the output at time instance k to the previous outputs and current previous inputs. The Wiener system can be written as follows

$$y_k = g\left(-\sum_{j=1}^{n_a} a_j v_{k-j} + \sum_{i=0}^{n_b} b_i u_{k-i}\right). \tag{3.138}$$

Note that (3.138) includes the auxiliary signals v_k, hence does not conform to the NARX representation. A Wiener model can possess a NARX representation only if the nonlinear static function $g(\cdot)$ is invertible. By assuming that $g^{-1}(\cdot)$ exists a Wiener model is re-written with the intermediate signal v_k eliminated by substituting $v_k = g^{-1}(y_k)$, i.e.

$$y_k = g\left(-\sum_{j=1}^{n_a} a_j g^{-1}(y_{k-j}) + \sum_{i=0}^{n_b} b_i u_{k-i}\right). \tag{3.139}$$

Although in this case the system (3.139) belongs to the class of NARX models, it is still not a member of NAARX models.

Analogous observation is also valid with respect to the Hammerstein-Wiener models. Namely, if the static nonlinear function on the output is invertible then the Hammerstein-Wiener model has the following NARX (but not NAARX) representation, i.e.

$$y_k = g\left(-\sum_{j=1}^{n_a} a_j g^{-1}(y_{k-j}) + \sum_{i=0}^{n_b} b_i f(u_{k-i})\right). \tag{3.140}$$

Bilinear system models although belong to the family of NARX models, cannot be considered as a part of the class of NAARX models defined by equation (3.136).

Questions

- Explain the difference between linearity in terms of input-output signals and linearity in terms of model parameters. Provide examples.
- Discuss the difference between static and dynamic models. Provide examples.
- Discuss the difference between LTI and LTV models. Provide examples.
- Discuss the difference between discrete-time and continuous-time models. Provide examples.
- Discuss the difference between deterministic and stochastic models. Provide examples.
- Explain possible representations of continuous-time/discrete-time systems. Give examples of each.
- Explain the motivation for using continuous-time/discrete-time models.
- Explain how discrete-time models can be obtained from corresponding continuous-time models.
- Discuss differences between transfer function and state-space representations.
- Write down the transfer function representation for the general stochastic LTI model.
- Explain the main motivation for postulating that disturbances follow a Gaussian distribution. Is this assumption realistic?
- Explain the abbreviations AR, MA, FIR, ARX, ARMA, ARMAX, ARIMAX, ARARX, ARARMAX, OE, BJ. Draw the corresponding block diagrams and write down the corresponding transfer function representations.
- Explain the differences between EE and OE type model structures.
- What is the pragmatic approach for coping with uncertainties?
- Explain differences between prediction and simulation. Which output, i.e. predicted or simulated, can be expected to be closer to the actual system output?
- Explain the concept of an optimal one-step ahead predictor.
- Explain why there is often a need to use nonlinear models in practice.
- Explain Hammerstein and Wiener models, draw the corresponding block diagrams and write down the corresponding input-output relations. Comment on the steady-state behaviour and stability properties.

- Discuss bilinear systems and their interpretation as LTV systems. Write down the corresponding state-space and input-output representations. Comment on the steady-state behaviour and stability properties.
- Discuss the class of NARX models.

Chapter 4

Identification of low order continuous-time linear time-invariant systems from step response

4.1 Introduction

In this chapter relatively simple identification procedures for first and second order continuous-time LTI systems, when subject to step excitation, are explained. The material is a summary of Subsection 3.9 of (Wellstead & Zarrop 1991). Reasoning analogous to that described here can also be applied in the case when the input is an impulse, by using a fact that a step response is an integrated impulse response.

4.2 First order system

A first order continuous-time LTI system is completely characterised by a steady-state value, called system/process/d.c. gain, denoted g_{ss}, and the time constant, denoted τ. The corresponding transfer function is given by

$$G(s) = \frac{b}{s+a} = \frac{g_{ss}}{\tau s + 1}, \tag{4.1}$$

where $\tau = \frac{1}{a}$ and $g_{ss} = \frac{b}{a}$. By assuming zero initial conditions, the response of system (4.1) to a unit step of magnitude K, i.e. $U(s) = \frac{K}{s}$, is given by

$$Y(s) = G(s)U(s) = K\frac{g_{ss}}{s(\tau s + 1)} = Kg_{ss}\left(\frac{1}{s} - \frac{\tau}{\tau s + 1}\right), \tag{4.2}$$

which, by using the inverse Laplace transform, converted to a time domain yields

$$y(t) = Kg_{ss}\left(1 - e^{-\frac{t}{\tau}}\right). \tag{4.3}$$

Figure 4.1 shows and exemplary response of a first order system subject to a unity step input. By analysing Figure 4.1 it is observed that the steady-state gain is equal 2, i.e. $g_{ss} = 2$ and that the time constant is 0.5, hence the system

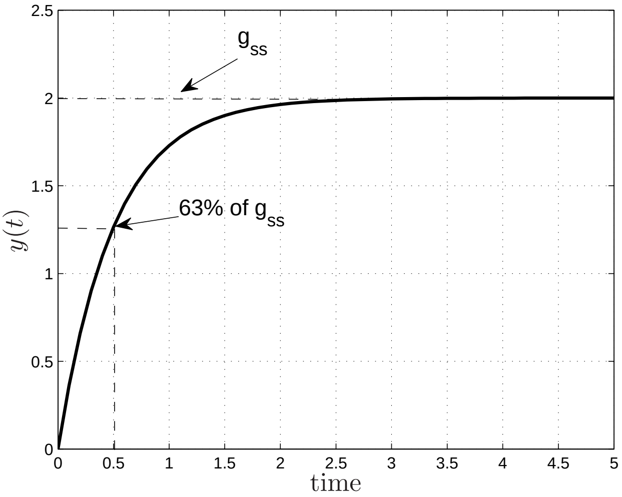

Figure 4.1: A typical response of a first order LTI system.

is given by

$$G(s) = \frac{2}{0.5s+1} = \frac{4}{s+2} \tag{4.4}$$

or, equivalently, by

$$y(t) = 2\left(1 - e^{-\frac{t}{2}}\right). \tag{4.5}$$

Moreover, it is observed that the response reaches the steady-state value already at approximately $t = 2.5$, which is five times the time constant. It can be verified that the value equal $0.99g_{ss}$ is reached after $t = 5\tau$. Consider an analogous response, given in Figure 4.2, where the only difference to Figure 4.1 is that it is delayed in time by $T_d \geq 0$, i.e. a transportation lag is present. Since $\mathcal{L}\left\{f(t-T)\right\} = e^{-sT_d}F(s)$, a corresponding response of such a system is defined by

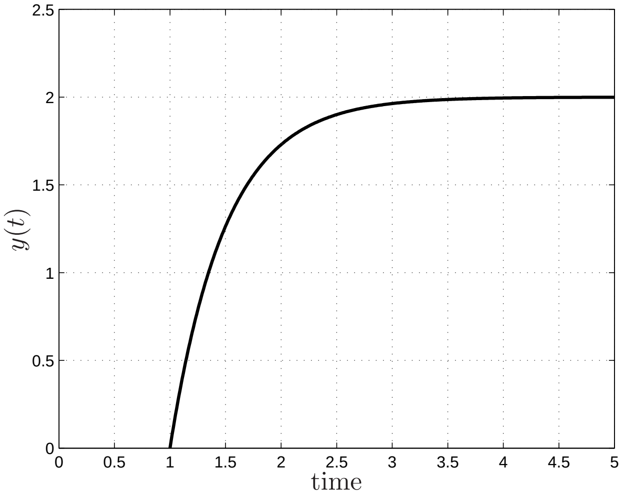

Figure 4.2: A typical response of a first order LTI system with a transportation lag of unity.

$$Y(s) = e^{-sT_d}K\frac{g_{ss}}{s(\tau s+1)} = Kg_{ss}\left(\frac{e^{-sT_d}}{s} - \frac{e^{-sT_d}\tau}{\tau s+1}\right), \tag{4.6}$$

which, in a time domain, is given by

$$y(t - T_d) = Kg_{ss}\left(1 - e^{-\frac{t}{\tau}}\right). \tag{4.7}$$

In the case presented in Figure 4.2 the transportation lag is equal unity, i.e. $T_d = 1$, hence the corresponding transfer function is given by

$$G(s) = e^{-s}\frac{4}{s+2} \tag{4.8}$$

or, equivalently in a time domain, by

$$y(t - 1) = 2\left(1 - e^{-\frac{t}{2}}\right). \tag{4.9}$$

4.3 Second order system

A transfer function of a second order continuous-time LTI system with no zeros is, in general, given by

$$G(s) = \frac{g_{ss}\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \tag{4.10}$$

where ω_n and ζ are the natural frequency of oscillations and damping factor, respectively. A damped natural frequency ω_d is related to ω_n by the relation $\omega_d = \omega_n\sqrt{1-\zeta^2}$. Consequently, a second order system is completely characterised by three parameters, i.e. g_{ss}, ω_n and ζ. The response to a step input of magnitude K is given by

$$G(s) = K\frac{g_{ss}\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} = Kg_{ss}\left[\frac{1}{s} + \frac{-\frac{1}{2}\left(\zeta\frac{\omega_n}{\omega_d} + j\right)}{s + \zeta\omega_n - j\omega_d} + \frac{-\frac{1}{2}\left(\zeta\frac{\omega_n}{\omega_d} - j\right)}{s + \zeta\omega_n + j\omega_d}\right], \tag{4.11}$$

where j denotes the imaginary part. The inverse Laplace transform of equation (4.11) yields

$$y(t) = Kg_{ss}\left[1 - e^{-\zeta\omega_n t}\left(\zeta\frac{\omega_n}{\omega_d}\sin\omega_d t + \cos\omega_d t\right)\right]. \tag{4.12}$$

Depending on the general character of the step response two cases are distinguished, i.e. overdamped and underdamped response.

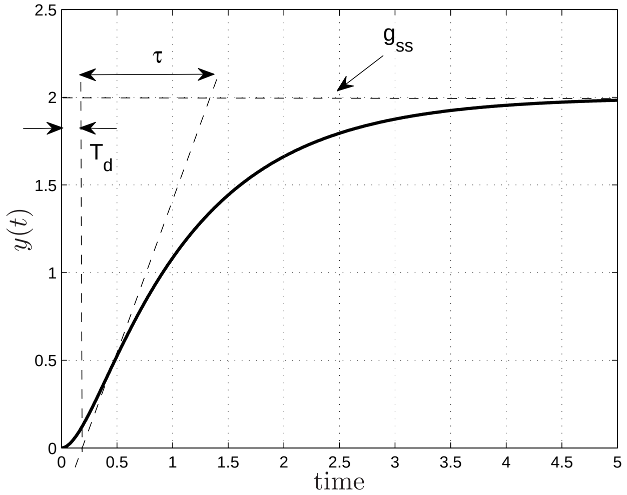

Figure 4.3: A typical response of an overdamped second order LTI system.

4.3.1 Overdamped response

Although the actual system is of second order a simple identification procedure, by using a Ziegler-Nichols method, involves an approximation by a first order model with a transportation lag, i.e.

$$G(s) = e^{-sT_d}\frac{g_{ss}}{\tau s + 1}. \tag{4.13}$$

The main reason for utilising this approximation is that, since no oscillations are present in an overdamped response, it is not feasible to infer information regarding their natural frequency. The system gain is found as described previously in the case of the first order system. The transportation lag and the time constant are both determined by drawing a straight line tangent at the steepest ascent of the system response. The time value at which this line intersect the x-axis corresponds to the system lag, whilst the time value at which an intersection with the system gain occurs corresponds to the system time constant. In this case it is found that $T_d = 0.19$ and $\tau = 1.1$, hence a first order system with a transportation lag, which approximates the second order underdamped system is given by

$$G(s) = \frac{2e^{-0.19s}}{1.1s + 1}. \tag{4.14}$$

The response of the estimated model (4.14) juxtaposed with that of the actual system is depicted in Figure 4.14. It is observed that the approximation obtained shows a relatively good resemblance of the actual system response.

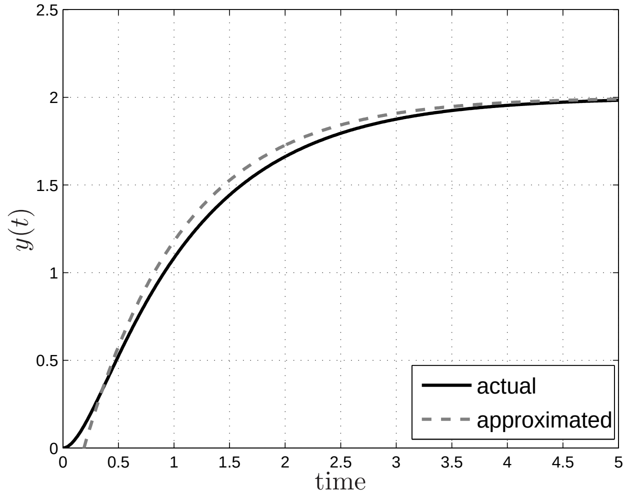

Figure 4.4: A figure showing a result of the approximation of a second order system via a first order system with a transportation delay.

4.3.2 Underdamped response

The underdamped response, see Figure 4.4, provides much more information than the overdamped response allowing to infer information of both, i.e. the natural frequency and the damping ratio. Note that, since a first order model exhibits no oscillations, it is not feasible to use it for the purpose of approximating the underdamped second order system. By considering Figure 4.4, the frequency of damped oscillations, denoted ω_d, can be determined by estimating

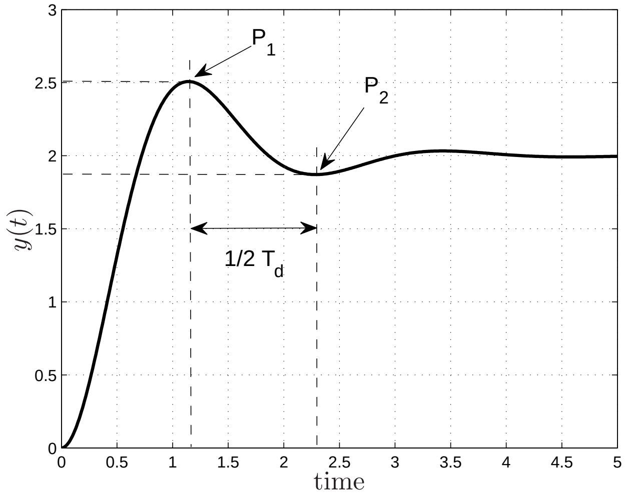

Figure 4.5: A typical response of a second order.

the distance between two neighbouring peaks, denoted here P_1 and P_2, which is equal to $\frac{1}{2}T_d$. Therefore, the damped frequency of oscillations is given by

$$\omega_d = \frac{2\pi}{T_d}. \tag{4.15}$$

By recalling equation (4.12), it is noticed that the decay of oscillations is due to the exponential term $e^{-\zeta\omega_n t}$. Therefore, the ratio of decrease in heights of two neighbouring peaks P_1 and P_2, with respect to the steady-state value, can be related to the distance between them via

$$\frac{|P_2 - g_{ss}|}{|P_1 - g_{ss}|} = e^{-\frac{1}{2}\zeta\omega_n T_d} \tag{4.16}$$

from which the term $\zeta\omega_n$ is calculated as

$$\zeta\omega_n = \frac{2}{T_d}\left(\ln|P_1 - g_{ss}| - \ln|P_2 - g_{ss}|\right). \tag{4.17}$$

The damped frequency of oscillations is related to the natural frequency via

$$\omega_d = \omega_n\sqrt{1-\zeta^2}, \tag{4.18}$$

which leads to

$$\omega_n^2 = \omega_d^2 + (\zeta\omega_n)^2 . \tag{4.19}$$

Consequently, the natural frequency of oscillations can be determined from (4.19) by substituting equations (4.17) and (4.15), whilst the steady-state gain is determined directly from the plot of the system step response. By applying the reasoning introduced to the response shown in Figure 4.4 it is found that $T_d = 2.28$, hence $\omega_d = 2.76$. The steady-state gain g_{ss} is equal 2, thus the relative heights at points P_1 and P_2 are 0.51 and 0.13, respectively, which means that the term $\zeta\omega_n$ is equal to 1.2. Subsequently, by using (4.19) the value of ω_n^2 is estimated to be 9.06. Consequently, the overall estimated transfer function is described by

$$G(s) = \frac{18.12}{s^2 + 2.4s + 9.06}. \tag{4.20}$$

Since the actual system was given by

$$G(s) = \frac{18}{s^2 + 2.4s + 9} \tag{4.21}$$

the close values of parameters obtained support the validity and appropriateness of the reasoning conducted (note also the agreement of the steady-state gains).

4.4 Noisy measurements

It is noted that the identification procedures carried out in this chapter postulated an idealised setup, where no measurement noise was present. In practice, one can expect that the system response is corrupted by disturbances, whose strength will depend on the quality of sensors used and the environment in which the experiments are conducted. Figure 4.6 shows an exemplary step response of first order system considered previously in Section 4.2 with Gaussian white noise sequences of progressively increasing strength imposed, i.e. no noise, low noise, medium noise and high noise contamination. It is observed from Figure 4.6 that with an increasing strength of noise it becomes also increasingly more difficult to precisely determine the time constant and the steady-state gain of the system. In particular, even if in all cases it is assumed that the steady-state vales are guessed correctly to be 2, the time constants corresponding to the three

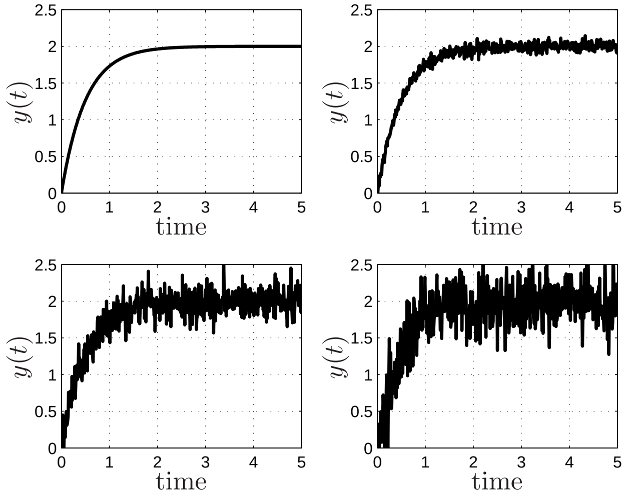

Figure 4.6: A typical response of a first order LTI system with Gaussian white measurement noise sequences of different strengths imposed. Top-left - no noise, top-right - low noise, bottom-left - medium noise, bottom-right - high noise contamination.

noisy cases are determined to be: 0.49, 0.46 and 0.58. Consequently, by a comparison with the actual time constant, the corresponding relative errors are 2%, 8% and 16%, respectively. Additionally, it is remarked that in the case of noisy measurements a single value on the y-axis can correspond to multiple values on the x-axis, hence the determination of the time constant is not straightforward. In fact, even in the case of relatively noisy measurements, it is still possible to obtain relatively accurate estimates of system parameters by using a computer assistance, which will be described in details in the sequel. Furthermore, when using a computer aid the input is not restricted to any particular signal.

Questions

- Explain the steps required for identification of a first order continuous-time model from a step response.

- Explain the steps required for identification of a second order continuous-time model from a step response. Consider two cases, i.e. underdamped and overdamped system.
- Discuss advantages and disadvantages of system identification from a step input.

Chapter 5

Least squares

In this chapter an introduction into the method of least squares (LS) is presented. The LS technique was developed by C. F. Gauss around year 1795 and its development was motivated by a desire to accurately calculate orbits of planets, motion of which was characterised by Kepler laws. This task required six parameters to be inferred from raw measurements taken by a telescope. Due to various reasons the work of Gauss was not published until 1809. It is worth mentioning that within this period, the method of LS was actually re-discovered independently by A. M. Legendre, see (Sorenson 1970) for more historical details.

5.1 Least squares for static systems

In general, the method of LS can be used for estimating the parameters of differential/difference equations, weighting sequences as well as static systems, cf. Subsection 3.2.2. The crucial requirement is that the model has to be linear with respect to the estimated parameters.

Table 5.1: An exemplary table consisting of three measurements.

no.	x_k	y_k
1	1	2.00
2	2	5.25
3	3	5.75

5.1.1 Introductory example

The introduction into the method of LS is given based on an archetypical problem of fitting a curve (in the case considered being a straight line), into a set of inconsistent, i.e. noisy, measurements. Consider a problem of describing the relationship between two arbitrary signals, denoted x_k and y_k, based on the following three measurements given in Table 5.1. This example is based on that given in (Mańczak & Nahorski 1983). The three data points, marked with crosses, are shown visually in Figure 5.1. It is observed that the relationship between the x_k and y_k values seems to be approximately linear. However, it is not possible to draw a single straight line coinciding with all points. It can be stipulated that although, in fact, all points should be laying on a single straight line, due to measuring inaccuracies on y_k, this is not the case. Consequently, a following question arises - is it possible to draw a straight line in some optimal manner such that it will lay as close to data points as possible? A general equation for a straight line is defined by

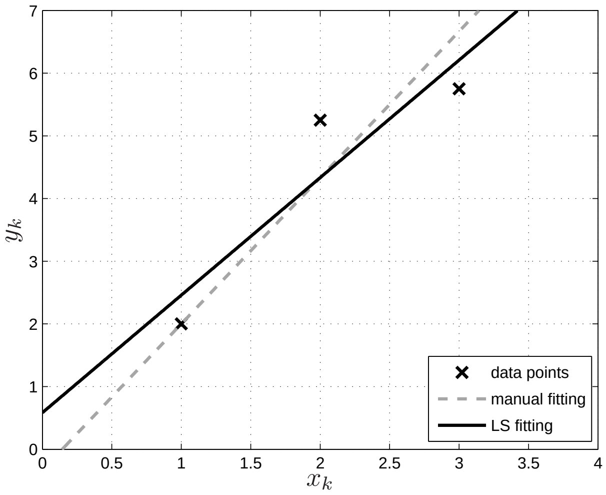

Figure 5.1: Two straight lines fitted to inconsistent set of three measurements.

$$y_{0_k} = \alpha x_k + \beta, \tag{5.1}$$

where α and β are unknown parameters that are required to be determined. Since the set of equations is inconsistent, i.e.

$$y_k \approx \alpha x_k + \beta, \tag{5.2}$$

the actual measurement had to be generated according to

$$y_k = \alpha x_k + \beta + e_k, \tag{5.3}$$

where $e_k = y_k - y_{0_k}$ is the measurement noise (or a model uncertainty, in general). Note that in generic terms equation (5.1) can be considered to be a (static) system, which is parametrised by only two parameters. A rather straightforward engineering solution would be to draw a straight line through the first value of y_k and approximately in between of the two remaining values. The result of such approach, referred to as a manual fitting, is shown in Figure 5.1 using a grey dashed straight line. The corresponding parameters are

$$\hat{\alpha} = 2.333 \qquad \text{and} \qquad \hat{\beta} = -0.333 \tag{5.4}$$

where a hat over a parameter indicates that the corresponding value is an estimate.

A more scientific approach would be, first, to define the notion of optimality, second, to introduce a performance index, which would quantify a given result, and, third, obtain the minimum of such performance index yielding optimal estimates. In the framework of LS the optimality is defined in terms of minimising vertical distances of data points to the curve fitted. The performance index (called also a cost function) is a sum of squared vertical distances, which, in the case considered, can be written as

$$V(\alpha, \beta) = \frac{1}{2}\sum_{k=1}^{3} \varepsilon_k^2 = \frac{1}{2}\sum_{k=1}^{3} \left[y_k - y_{0_k}(\alpha, \beta)\right]^2 = \frac{1}{2}\sum_{k=1}^{3} \left(y_k - \alpha x_k - \beta\right)^2, \tag{5.5}$$

where terms ε_k and $y_{0_k}(\alpha, \beta)$ are the residual and the value of y_{0_k} obtained from a particular model (dependent on the choice of α and β), respectively. In order to find a minimum of the cost function (5.5) partial derivatives, with respect to α and β, are calculated, i.e.

$$\frac{\partial V(\alpha, \beta)}{\partial \alpha} = -\sum_{k=1}^{3} x_k \left(y_k - \alpha x_k - \beta\right), \tag{5.6}$$

$$\frac{\partial V(\alpha, \beta)}{\partial \beta} = -\sum_{k=1}^{3} y_k - \alpha x_k - \beta. \tag{5.7}$$

The optimal values of the parameters are found by comparing the partial derivatives to zero, i.e.

$$0 = -2 \sum_{k=1}^{3} x_k \left(y_k - \alpha x_k - \beta \right), \tag{5.8}$$

$$0 = -2 \sum_{k=1}^{3} y_k - \alpha x_k - \beta. \tag{5.9}$$

By re-arranging (5.8) one obtains a set of so-called normal equations

$$\sum_{k=1}^{3} y_k = 3\beta + \alpha \sum_{k=1}^{3} x_k, \tag{5.10}$$

$$\sum_{k=1}^{3} x_k y_k = \beta \sum_{k=1}^{3} x_k + \alpha \sum_{k=1}^{3} x_k^2. \tag{5.11}$$

After some algebraic manipulations the optimal (in the LS sense) estimates of the parameters are obtained, i.e.

$$\hat{\beta} = \frac{1}{3} \left(\sum_{k=1}^{3} y_k - \alpha \sum_{k=1}^{3} x_k \right),$$

$$\hat{\alpha} = \frac{3 \sum_{k=1}^{3} x_k y_k - \sum_{k=1}^{3} x_k \sum_{k=1}^{3} y_k}{3 \sum_{k=1}^{3} x_k^2 - \sum_{k=1}^{3} x_k \sum_{k=1}^{3} x_k}. \tag{5.12}$$

In order to confirm that the extremum found is a minimum and not a maximum, a second derivative of the cost function (5.5), called the Hessian matrix, is calculated, which is given by

$$\frac{\partial^2 V(\alpha, \beta)}{\partial [\alpha \ \beta]^T} = \begin{bmatrix} \frac{\partial^2 V(\alpha,\beta)}{\partial \alpha^2} & \frac{\partial^2 V(\alpha,\beta)}{\partial \alpha \partial \beta} \\ \frac{\partial^2 V(\alpha,\beta)}{\partial \beta \partial \alpha} & \frac{\partial^2 V(\alpha,\beta)}{\partial \beta^2} \end{bmatrix} \tag{5.13}$$

$$= \begin{bmatrix} \sum_{k=1}^{3} x_k^2 & \sum_{k=1}^{3} x_k \\ \sum_{k=1}^{3} x_k & 3 \end{bmatrix} = \begin{bmatrix} 14 & 6 \\ 6 & 3 \end{bmatrix}. \tag{5.14}$$

Since the Hessian matrix is positive definite the extremum corresponds to a minimum of the cost function (5.5).

In the LS case the values of the estimated parameters are

$$\hat{\alpha} = 1.875 \qquad \text{and} \qquad \hat{\beta} = 0.583, \tag{5.15}$$

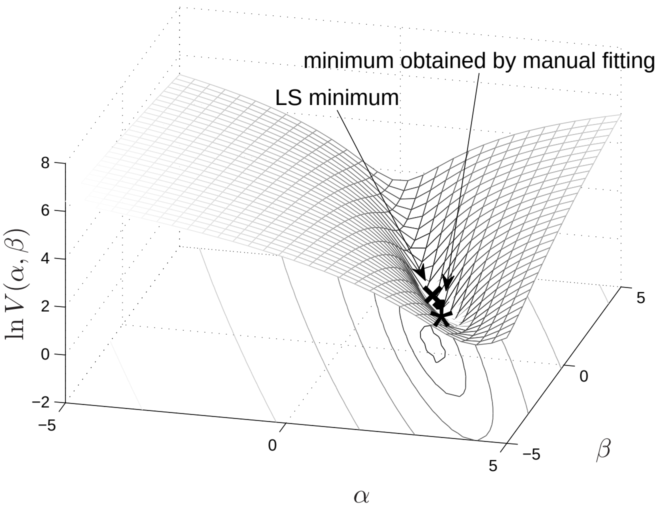

Figure 5.2: The LS cost function with marked values corresponding to the LS estimate (cross) and the estimate obtained manually (star).

and the corresponding curve is shown in Figure 5.1 as a solid black straight line. In order to assess which of the two sets of parameters provides a curve which is closer to the data points, the corresponding values of the cost function (5.5) can be considered. In the case of the curve fitted manually $V(2.333, -0.333) = 0.841$, whilst in the case of the LS estimates $V(1.875, 0.583) = 0.630$. Therefore, it is clearly observed that the line corresponding to the estimates calculated using the LS method yields a superior fitting. In fact, it can be shown that under some further assumptions regarding the properties of the measurement noise, the LS method is the best linear unbiased estimator (BLUE). Moreover, it is interesting to note that a sum of residuals is in both cases, in fact, identical and equal zero.

It is also instructive to plot the LS cost function, which, in the simple case considered here, is feasible, since it is parametrised by two parameters only. The LS cost function is shown in Figure 5.2, where, in order to improve readability, it is transformed via natural logarithm. Additionally, the corresponding values of the two solutions obtained via manual fitting and the LS method are presented. It is observed that, first, the cost function is convex, and, second, that the LS

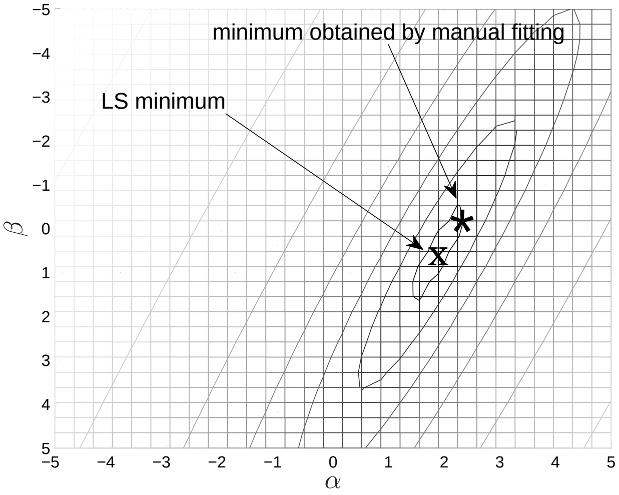

Figure 5.3: The LS cost function with marked values corresponding to the LS estimate (cross) and the estimate obtained manually (star), projected onto the plane $\ln V(\alpha, \beta) = 0$.

solution lies in the middle of the valley, exactly at the minimum. In contrast, the solution obtained manually, although close, is not located at the minimum (note the estimated negative value of β). The LS cost function projected onto the plane $\ln V(\alpha, \beta) = 0$ is given in Figure 5.3.

5.1.2 General case

The methodology for calculating the LS estimate described in the previous subsection for a particular simple example can be extended to a general case of static system defined by

$$y_{0_k} = \beta_1 x_{k1} + \beta_2 x_{k2} + \ldots + \beta_n x_{kn}, \tag{5.16}$$

where, for the ease of notation, the n parameters are denoted as β with subscripts from 1 to n. Note that the simple system considered in Subsection 5.1.1 and given by equation (5.1) is a special case of (5.16) with $n = 2$, $\beta_1 = \beta$, $x_{k1} = 1\ \forall k$, $\beta_2 = \alpha$ and $x_{k2} = x_k$. Equation (5.16), due to the linearity in parameters, can be re-expressed as

$$y_{0_k} = \begin{bmatrix} x_{k1} & x_{k2} & \ldots & x_{kn} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} = \varphi_k^T \theta, \tag{5.17}$$

where the vector consisting of available signals $\varphi_k \in \mathbb{R}^n$ is termed a regressor vector and the vector comprising of unknown parameters $\theta \in \mathbb{R}^n$ is termed a parameter vector. By assuming that the output of (5.16) is observed N times, the following set of equations can be formulated

$$\begin{aligned} y_{0_1} &= \beta_1 x_{11} + \beta_2 x_{12} + \ldots + \beta_n x_{1n} \\ y_{0_2} &= \beta_1 x_{21} + \beta_2 x_{22} + \ldots + \beta_n x_{2n} \\ &\vdots \\ y_{0_N} &= \beta_1 x_{N1} + \beta_2 x_{N2} + \ldots + \beta_n x_{Nn}. \end{aligned} \tag{5.18}$$

It is noted that in the case of an example considered in the previous subsection $N = 3$. The set of equations (5.18) can be re-expressed more conveniently using matrix notation as

$$Y_0 = \Phi\theta, \tag{5.19}$$

where $Y \in \mathbb{R}^N$ is a vector of stacked output signals, i.e. left-hand side of the set of equations (5.18), and the data (or observation) matrix $\Phi \in \mathbb{R}^{N \times n}$ is given by

$$\Phi = \begin{bmatrix} \varphi_1^T \\ \varphi_2^T \\ \vdots \\ \varphi_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \ldots & x_{Nn} \end{bmatrix}. \tag{5.20}$$

The task, similarly as in Subsection 5.1.1, comprises of determining the parameter vector θ in the case when the set of equations (5.19) is inconsistent, i.e.

$$Y \approx \Phi\theta. \tag{5.21}$$

Consequently, it is assumed that the measured values in Y are generated by

$$Y = \Phi\theta + e, \tag{5.22}$$

where $e = Y - Y_0$ denotes a vector of stacked values of the noise sequence e_k. In the case when $N < n$ there exist more unknowns than equations, thus no single solution exists. The minimum number of measurements required to obtain a solution to the problem is $N = n$. In this case equation (5.22) is a set N linear equations in N unknown parameters. These can be directly and easily solved for the parameters by, for example, the Gaussian elimination method. However, in practice one aims to record as many measurements as possible in order to reduce the effects of measurement uncertainty etc. and to improve the accuracy of estimates (Söderström & Stoica 1989). Consequently, if $N > n$ the set of equations is overdetermined (in practice $N >> n$). Analogously to Subsection 5.1.1, a following cost function is introduced, i.e.

$$V(\theta) = \frac{1}{2}\sum_{k=1}^{N} \varepsilon_k^2 = \frac{1}{2}\varepsilon^T\varepsilon = \frac{1}{2}\|\varepsilon\|^2, \tag{5.23}$$

where $\|\cdot\|$ denotes a square, i.e. Euclidean, vector norm. The LS solution is obtained by solving the following optimisation, i.e. minimisation, problem

$$\begin{aligned}\hat{\theta} = \arg\min_{\theta} V(\theta) &= \frac{1}{2}\left(Y - Y_0\right)^T\left(Y - Y_0\right) = \frac{1}{2}\left(Y - \Phi\theta\right)^T\left(Y - \Phi\theta\right) \\ &= \frac{1}{2}\left(Y^TY - Y^T\Phi\theta - \theta^T\Phi^TY + \theta^T\Phi^T\Phi\theta\right). \end{aligned} \tag{5.24}$$

The minimum value of (5.24) is obtained by calculating a partial derivative with respect to the parameter vector θ[1] and setting it to zero, i.e.

$$\frac{\partial V(\theta)}{\partial \theta} = -\Phi^TY + \Phi^T\Phi\theta = 0. \tag{5.25}$$

This leads to the following set of normal equations

$$\Phi^TY = \Phi^T\Phi\theta. \tag{5.26}$$

Hence, the optimal (in the LS sense) solution is given by[2]

$$\hat{\theta} = \left(\Phi^T\Phi\right)^{-1}\Phi^TY. \tag{5.27}$$

[1] The rules for calculating partial derivatives for vectors and matrices are: $\frac{\partial(Mx)}{\partial x} = M^T$, $\frac{\partial(x^TM)}{\partial x} = M$, $\frac{\partial(x^THx)}{\partial x} = 2Hx$, where x is a vector, M is a matrix and H is a symmetric matrix.

[2] Equation (5.27) can also be expressed as $\hat{\theta} = \Phi^{\dagger}Y$, where $\Phi^{\dagger} = \left(\Phi^T\Phi\right)^{-1}\Phi^T$ denotes the Moore-Penrose pseudoinverse.

The LS solution (5.27) can be, alternatively, expressed as

$$\hat{\theta} = \left(\sum_{k=1}^{N} \varphi_k \varphi_k^T \right)^{-1} \left(\sum_{k=1}^{N} \varphi_k y_k \right), \tag{5.28}$$

which is the basis of a recursive version of the LS estimator described in the next chapter. Note that a single solution exists only if $\left(\Phi^T \Phi\right)^{-1}$, called a covariance matrix, exists, i.e. if $\Phi^T \Phi$ is invertible. This requires the rows of the matrix Φ to be linearly independent, or equivalently, for the matrix product $\Phi^T \Phi$ to be of full rank. In order to verify that the extremum obtained corresponds to a minimum of the cost function (5.23) the Hessian matrix is calculated, i.e.

$$\frac{\partial^2 V(\theta)}{\partial \theta^2} = \Phi^T \Phi. \tag{5.29}$$

Since the quadratic matrix product $\Phi^T \Phi$ is non negative definite by definition, the solution found is a minimum.

5.1.3 Properties of LS

In this section a few properties of interest in the LS estimator are described in detail; see (Hsia 1977) or (Söderström & Stoica 1989) for a more thorough discussion. First some, rather mild, assumptions regarding the properties of the noise sequence e_k are made, namely:

i) e_k is a zero mean random sequence, i.e. $E[e_k] = 0$ (where $E[\cdot]$ is the expectation operator)

ii) e_k is serially (mutually) uncorrelated and possesses a constant variance (i.e. it is a stationary process) σ_e^2, i.e.

$$E[e_k e_j] = \sigma_e^2 \delta_{kj}, \tag{5.30}$$

where δ_{kj} is a Kronecker delta function defined by

$$\delta_{kj} = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases} \tag{5.31}$$

iii) e_k is uncorrelated with signals contained in the regressor vector φ_k, i.e.

$$E[e_k \varphi_j] = 0 \quad \forall\ k,\ j \tag{5.32}$$

Having assumed the properties of the stochastic part of the system, it is now possible to quantify the accuracy of estimates yielded by the LS method.

First, a notion of bias is discussed. In the case where the estimate is unbiased its expectation is equal to the true value, i.e. $E[\hat{\theta}] = \theta$. This property can be determined by substituting equation (5.22) into (5.27) yielding

$$\hat{\theta} = \left(\Phi^T\Phi\right)^{-1}\Phi^T\left(\Phi\theta + e\right) = \theta + \left(\Phi^T\Phi\right)^{-1}\Phi^T e. \tag{5.33}$$

By taking the expectation of both side of equation (5.33) and recalling assumptions i) and iii), one obtains

$$E[\hat{\theta}] = E[\theta] + E[\left(\Phi^T\Phi\right)^{-1}\Phi^T]E[e] = E[\theta], \tag{5.34}$$

which shows that the LS estimate is unbiased, i.e. the estimation error can be expected to be zero.

Now consider a covariance matrix, denoted $P^* = \mathrm{cov}(\tilde{\theta})$ of the estimation error $\tilde{\theta} = \theta - \hat{\theta}$, which is defined as

$$P^* = E[(\theta - \hat{\theta})(\theta - \hat{\theta})^T] = E[(\Phi^T\Phi)^{-1}\Phi^T e\left((\Phi^T\Phi)^{-1}\Phi^T e\right)^T]. \tag{5.35}$$

By taking into account assumption iii) it follows that

$$\begin{aligned} P^* &= E[(\Phi^T\Phi)^{-1}\Phi^T]E[ee^T]E[\left((\Phi^T\Phi)^{-1}\Phi^T\right)^T] \\ &= (\Phi^T\Phi)^{-1}\Phi^T E[ee^T]\Phi(\Phi^T\Phi)^{-1}. \end{aligned} \tag{5.36}$$

It is noted that due to assumption ii) the matrix $E[ee^T]$ is given by

$$E[ee^T] = \sigma_e^2 I, \tag{5.37}$$

where I is the identity matrix of an appropriate dimension. Consequently, the error covariance matrix simplifies to

$$P^* = \sigma_e^2(\Phi^T\Phi)^{-1}\Phi^T\Phi(\Phi^T\Phi)^{-1} = \sigma_e^2(\Phi^T\Phi)^{-1}. \tag{5.38}$$

Note further that by comparing this with equation (5.27) the error covariance matrix is the estimation covariance matrix multiplied by the variance of noise. This property allows to the inference of information regarding the accuracy of individual parameter estimates directly from the LS covariance matrix. This is because the variance of the i-th estimated parameter is given by i-th column and i-th row of $(\Phi^T\Phi)^{-1}$ multiplied by σ_e^2. Note also that P^* is proportional to the noise variance σ_e^2 and inversely proportional to the power of the input signal (P^* is also inversely proportional to the so-called signal-to-noise (SNR) ratio). Therefore, during the phase of the experiment design, see Figure 1.1 in Chapter 1, one aims to choose the input so that the matrix product $\Phi^T\Phi$ is maximised (Ljung 1999).

In practice one does not have access to the true value of the noise variance, it also has to be estimated from data. It will be demonstrated that it can be estimated based on the LS residuals. Consider the following expression for the LS residuals

$$\varepsilon = Y - \Phi\hat{\theta} = \Phi\theta + e - \Phi\left[\left(\Phi^T\Phi\right)^{-1}\Phi^T\left(\Phi\theta + e\right)\right], \tag{5.39}$$

which is found by using equations (5.22) and (5.27). This can be simplified to

$$\varepsilon = e - \Phi\left(\Phi^T\Phi\right)^{-1}\Phi^T e = Be, \tag{5.40}$$

where the symmetric idempotent matrix[3] B is given by

$$B = I - \Phi\left(\Phi^T\Phi\right)^{-1}\Phi^T. \tag{5.41}$$

The sum of squared LS residuals can now be expressed as

$$\sum_{k=1}^{N}\varepsilon_k^2 = \varepsilon^T\varepsilon = e^T B^T B e = e^T B e. \tag{5.42}$$

Consequently, the estimate of the variance of the LS residuals can be calculated as follows

$$\begin{aligned} E[\varepsilon^T\varepsilon] &= E[e^T B e] = E[e^T\left(I - \Phi(\Phi^T\Phi)^{-1}\Phi^T\right)e] \\ &= E[e^T e] - E[e^T\Phi(\Phi^T\Phi)^{-1}\Phi^T e]. \end{aligned} \tag{5.43}$$

By noting that

$$\begin{aligned} E[e^T e] &= \sigma_e^2 N, \\ E[e^T\Phi(\Phi^T\Phi)^{-1}\Phi^T e] &= \sigma_e^2 \mathrm{tr}\left[\Phi(\Phi^T\Phi)^{-1}\Phi^T\right], \end{aligned} \tag{5.44}$$

[3] The matrix B is said to be idempotent if $B^2 = B$.

where tr(·) denotes the trace of a matrix[4]. Consequently, equation (5.43) can be simplified to

$$\begin{aligned} E[\varepsilon^T \varepsilon] &= \sigma_e^2 \left\{ N - \operatorname{tr} \left[(\Phi^T \Phi)^{-1} \Phi^T \Phi \right] \right\} \\ &= \sigma_e^2 \left(N - \operatorname{tr} I \right) = \sigma_e^2 (N - n), \end{aligned} \tag{5.45}$$

since the matrix product $\Phi^T\Phi$ is of order n. Finally, the estimate of the noise variance is given by

$$\hat{\sigma}_e^2 = \frac{1}{N-n} \varepsilon^T \varepsilon = \frac{1}{N-n} \sum_{k=1}^{N} \varepsilon_k^2. \tag{5.46}$$

Note that equation (5.46) means that the variance of noise can be inferred from the variance of the LS residuals. However, in order for the estimate to be unbiased, i.e. to account for degrees of freedom, the sum of squared LS residuals has to be normalised by one over $N - n$ and not N only.

It can be shown that LS provides consistent estimates, i.e. the estimates converge asymptotically (in probability) to their true values. This requires that the associated distribution of the estimates becomes more concentrated in vicinity of the true values as $N \to \infty$ and the corresponding error covariance matrix approaches a zero matrix, see (Hsia 1977). Consider the following expression

$$P^* = \sigma_e^2 (\Phi^T \Phi)^{-1} = \frac{\sigma_e^2}{N} \left(\frac{1}{N} \Phi^T \Phi \right)^{-1}. \tag{5.47}$$

By assuming that the term $\left(\frac{1}{N}\Phi^T\Phi\right)^{-1}$ converges to some nonsingular matrix, say Γ, as N increases[5], i.e.

$$\lim_{N \to \infty} \left(\frac{1}{N} \Phi^T \Phi \right)^{-1} = \Gamma, \tag{5.48}$$

it follows that

$$\lim_{N \to \infty} P^* = \Gamma \lim_{N \to \infty} \frac{\sigma_e^2}{N} = 0. \tag{5.49}$$

[4] A trace of a square matrix $A \in \mathbb{R}^{n \times n}$ is a sum of its diagonal elements, i.e. $\operatorname{tr}(A) = \sum_{i=1}^{n} a_{ii}$.

[5] In the case of dynamic systems, this assumption is directly related to the notion of so-called sufficient excitation of the input signal. Loosely speaking, the input is said to be sufficiently exciting if it is of sufficiently informative content, e.g. changes relatively quickly and over a relatively wide range, such that it allows the maximum information to be obtained from the system response, see (Hsia 1977) or (Söderström & Stoica 1989) for details.

This demonstrates that $\lim\limits_{N\to\infty} \hat{\theta} = \theta$ and therefore the LS estimator is a consistent estimator. It is said that a system is identifiable (within a given subset of models) if the parameter estimates are consistent, see (Söderström & Stoica 1989). It can be shown that in the case when the equation error is white, cf. assumption ii), the LS estimate is BLUE. If additionally the equation error is also Gaussian, then the LS estimate is the best of all both linear and nonlinear estimators and it is identical to the maximum likelihood estimator, see (Söderström & Stoica 1989).

5.1.4 Geometrical interpretation of LS

Referring to (Söderström & Stoica 1989), the LS estimate can be interpreted geometrically as the orthogonal projection of the residual vector ε onto the n-dimensional sub-space spanned by columns of the data matrix Φ. Such situation is depicted visually in Figure 5.4 for a simple two dimensional case, i.e. $n = 2$, with three measurements considered, i.e. $N = 3$, cf. the example from Subsection (5.1.1). Denote the i-th column of the data matrix Φ by $\mathrm{vec}(\Phi_i)$. The LS forces the system of equations (5.21), see Subsection 5.1.2, to be consistent by minimising the distance, corresponding to the residual vector ε, between the output vector Y and the sub-space spanned by $\mathrm{vec}(\Phi_i)$ for $i = 1, \ldots, n$. The minimal distance is obtained in the case when ε is orthogonal (i.e. perpendicular) to the sub-space spanned by $\mathrm{vec}(\Phi_i)$ for $i = 1, \ldots, n$, which can be expressed as

$$\mathrm{vec}(\Phi_i) \perp \varepsilon \quad \forall\, i. \tag{5.50}$$

This condition is equivalent to requiring that the following dot products are all null, i.e.

$$\Phi_i^T \varepsilon = \Phi_i^T \left(Y - \hat{Y} \right) = 0 \quad \forall\, i, \tag{5.51}$$

where $\hat{Y}$ denotes the projected output vector Y. Since $\hat{Y}$ belongs to the sub-space spanned by $\mathrm{vec}(\Phi_i)$ for $i = 1, \ldots, n$, it can be expressed as a weighted sum of the vectors $\mathrm{vec}(\Phi_i)$, which leads to

$$\Phi_i^T \left(Y - \sum_{j=1}^{n} \Phi_j \theta_j \right) = 0 \quad \forall\, i. \tag{5.52}$$

By recalling that the overall task is to determine θ_j the following set of equations

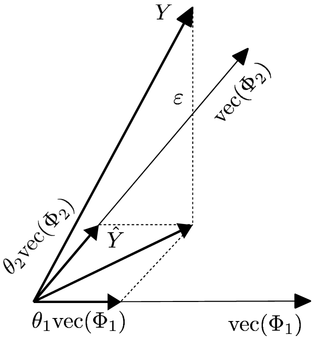

Figure 5.4: A geometrical interpretation of the LS method for a two dimensional case, i.e. $n = 2$, with three measurements, i.e. $N = 3$.

is obtained

$$\Phi_i^T Y - \sum_{j=1}^{n} \Phi_i^T \Phi_j \theta_j = 0 \quad \forall\ i, \tag{5.53}$$

which, written for all i and by using a matrix notation, yields

$$\underbrace{\begin{bmatrix} \mathrm{vec}^T(\Phi_1)Y \\ \vdots \\ \mathrm{vec}^T(\Phi_n)Y \end{bmatrix}}_{\Phi^T Y} = \underbrace{\begin{bmatrix} \mathrm{vec}^T(\Phi_1)\mathrm{vec}(\Phi_1) & \dots & \mathrm{vec}^T(\Phi_1)\mathrm{vec}(\Phi_n) \\ \vdots & \ddots & \vdots \\ \mathrm{vec}^T(\Phi_n)\mathrm{vec}(\Phi_1) & \dots & \mathrm{vec}^T(\Phi_n)\mathrm{vec}(\Phi_n) \end{bmatrix}}_{\Phi^T\Phi} \underbrace{\begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}}_{\theta}. \tag{5.54}$$

It is noted that (5.54) is identical to the normal equations obtained when calculating the minimum of the LS cost function, cf. (5.26).

5.2 Least squares for dynamic systems

In this section the LS method is used for the purpose of estimating the parameters of the ARX model structure, cf. Subsection 3.2.3 and equation (3.47). Because (3.47) is linear in the $n_\theta = n_a + n_b + 1$ parameters, it can be expressed in terms of the following regression equation

$$y_k = \varphi_k^T \theta + e_k, \tag{5.55}$$

where the regressor vector $\varphi_k \in \mathbb{R}^{n_\theta}$ is given by

$$\varphi_k = \begin{bmatrix} -y_{k-1} & \dots & -y_{k-n_a} & u_k & \dots & u_{k-n_b} \end{bmatrix}^T \tag{5.56}$$

and the parameter vector $\theta \in \mathbb{R}^{n_\theta}$ comprises the model parameters, i.e.

$$\theta = \begin{bmatrix} a_1 & \dots & a_{n_a} & b_0 & \dots & b_{n_b} \end{bmatrix}^T . \tag{5.57}$$

With reference to the reasoning conducted in Subsection 5.1.2, the LS cost function is given by (5.23), where in the case of the dynamic system (5.55) considered, the residuals are

$$\varepsilon_k = y_k - \varphi_k^T \theta. \tag{5.58}$$

The procedure of determining a minimum of the LS cost function is analogous to that carried out in Subsection 5.1.2 and the solution is given by expression (5.27) or equivalently (5.28). Note, however, that the statistical analysis presented in Subsection 5.1.3 is not entirely valid in the case of dynamical system because the regressor vector is not deterministic. The regressor vector, thus data matrix Φ, contains uncertain output signals, and therefore should be treated as a realisation of stochastic process, see (Söderström & Stoica 1989).

With reference to the analysis presented in (Söderström & Stoica 1989), consider the parameter estimation error $\hat{\theta} - \theta$. This is done by making use of equation (5.28) and the following transformations

$$\begin{aligned} \frac{1}{N}\sum_{k=1}^{N} \varphi_k \varphi_k^T \hat{\theta} &= \frac{1}{N}\sum_{k=1}^{N} \varphi_k y_k \\ \frac{1}{N}\sum_{k=1}^{N} \varphi_k \varphi_k^T \hat{\theta} &= \frac{1}{N}\sum_{k=1}^{N} \varphi_k \left(\varphi_k^T \theta + e_k\right) \\ \hat{\theta} - \theta &= \left(\frac{1}{N}\sum_{k=1}^{N} \varphi_k \varphi_k^T\right)^{-1} \frac{1}{N}\sum_{k=1}^{N} \varphi_k e_k. \end{aligned} \tag{5.59}$$

It can be shown that under some rather mild assumptions (e.g. that φ_k and e_k are stationary stochastic realisations of white noise, see (Söderström & Stoica 1989) for details) the sums in (5.59) converge asymptotically to their corresponding expected values, i.e.

$$\frac{1}{N}\sum_{k=1}^{N}\varphi_k\varphi_k^T \rightarrow E\left[\varphi_k\varphi_k^T\right] \qquad \text{as} \qquad N\rightarrow\infty, \tag{5.60}$$

$$\frac{1}{N}\sum_{k=1}^{N}\varphi_k e_k \rightarrow E\left[\varphi_k e_k\right] \qquad \text{as} \qquad N\rightarrow\infty. \tag{5.61}$$

Consequently, a reasoning similar to that conducted for a static case can be applied. In particular, equation (5.59) shows that the LS estimator is consistent if the two following conditions are satisfied, i.e.

i) The covariance matrix $E\left[\varphi_k\varphi_k^T\right]$ is nonsingular, i.e. $\det\left(E\left[\varphi_k\varphi_k^T\right]\right)\neq 0$.

ii) The covariance vector $E\left[\varphi_k e_k\right]$ is null, i.e. $E\left[\varphi_k e_k\right]=0$.

The first assumption means that the input signal is persistently exciting of sufficiently high order (in this case of order at least n_b+1), cf. Subsection 5.1.3. This may not hold if, for instance, linear feedback is present in the system loop, see (Wellstead & Zarrop 1991). The second assumption is fulfilled if the noise sequence e_k is white as only in this case is e_k uncorrelated with all previous signals, in particular, those contained in the regressor vector φ_k. Note that this assumption is quite restrictive and rarely met in practice. Another situation where the second condition holds is with $n_a=0$, corresponding to a FIR model, as in this case φ_k contains only input signals.

If condition ii) is not satisfied, for example e_k is coloured, the LS estimate will not be consistent (nor unbiased). However, there are, at least, two particular situations in which the LS estimate can still retain the consistency properties (Ljung 1999). These cases are as follows:

a) The properties of a stable (and inversely stable) filter, say $F(q)$, colouring the equation error are known *a priori*, i.e.

$$A(q^{-1})y_k = B(q^{-1})u_k + F(q)e_k. \tag{5.62}$$

In this case the input and output signals can both be filtered via $F^{-1}(q)$ leading to the expression in which the equation error is white, hence the LS method can be readily applied, i.e.

$$A(q^{-1})y_k^F = B(q^{-1})u_k^F + e_k, \tag{5.63}$$

where

$$y_k^F = F^{-1}(q) y_k \qquad \text{and} \qquad u_k^F = F^{-1}(q) u_k. \tag{5.64}$$

In particular, if the filter $F(q)$ is rational, i.e. $F(q) = \frac{C(q^{-1})}{D(q^{-1})}$ then this corresponds to the case of the general EE type model structure, cf. Subsection 3.2.3 and equations (3.71). In such a case the input and the output are to be filtered via $F^{-1}(q) = \frac{D(q^{-1})}{C(q^{-1})}$. Note that this situation, i.e. the knowledge regarding the filter $F(q)$ prior the identification, is rather impractical. Therefore, identification algorithms typically estimate the parameter vector θ and the parameters of the filter $F(q)$ simultaneously in an alternating fashion, see (Young 1984).

b) The coloured equation noise can be approximated by an autoregression process. This situation corresponds to the ARARX model structure described in Subsection 3.2.3, cf. equations (3.64). In this case by multiplying both sides of the difference equation (3.64) by the polynomial $D(q^{-1})$ one obtains

$$A_D(q^{-1}) y_k = B_D(q^{-1}) u_k + e_k, \tag{5.65}$$

where the auxiliary polynomials $A_D(q^{-1})$ and $B_D(q^{-1})$ of orders $n_a + n_d$ and $n_b + n_d$, respectively, are defined by

$$A_D(q^{-1}) = D(q^{-1}) A(q^{-1}) \quad \text{and} \quad B_D(q^{-1}) = D(q^{-1}) B(q^{-1}). \tag{5.66}$$

Because the equation error in (5.65) is white, the LS method will yield a consistent estimate. Note that in this case the computed parameter vector does not directly comprise the coefficients of $A(q^{-1})$ and $B(q^{-1})$ polynomials but the corresponding coefficients of polynomials $A_D(q^{-1})$ and $B_D(q^{-1})$, from which subsequently the a and b parameters can be obtained.

Questions

- Explain the main motivation for using the LS method.
- Explain type of problems/model structures the method of LS is applicable. Provide examples.
- Explain which error criterion the method of LS is based on.

- Discuss steps involved in the derivation of the LS method.
- Specify conditions that are required for the LS estimate to be unbiased and consistent.
- Explain how to assess the confidence in the values of LS estimates.
- Explain how to estimate the variance of the equation error based on residuals.
- Explain the abbreviation BLUE in the context of LS.
- Explain the geometrical interpretation of the method of LS.
- Explain the differences arising in the LS method when used for static and dynamic models.
- Explain what is meant by the property of persistent excitation of an input signal.
- Are there any cases in which the LS method can be used to yield unbiased estimates although the equation error is coloured?
- Write down the formula for LS estimator in a matrix-vector form and then in a covariance matrix/vector form. Comment on both representations.

Chapter 6

Recursive least squares

This section provides an introduction to the recursive identification methods, in particular the recursive least squares (RLS) algorithm is discussed in detail. Recursive methods allow the update of the model parameter estimates in a continuous manner while the process of interest is operating in real time. The main motivations for the usage of recursive techniques are as follows:

i) Decrease in the computational burden required for the continuous computation of new parameter estimates in an off-line, i.e. batch, fashion.

ii) Decrease in the memory storage requirements, i.e. information is stored in a 'compressed' form, independent of the continuously increasing number of measurements.

iii) Ability to track parameter variations on-line. This is especially useful in applications related to fault detection and isolation, where an instantaneous detection and accommodation of incipient faults (e.g. via adaptive reconfigurable control schemes) is critical for safety reasons. Changes of system parameters can be used as indicators of such faulty conditions, e.g. detected alteration of resistance in an electric circuit can allow for a replacement before the entire circuit stops functioning leading to potentially major unrecoverable damage.

iv) Adaptive control strategies, i.e. self-tuning control, where the control action is continuously adjusted based on a changing parametrisation of the model. Such control is useful when dealing with nonlinear systems, which can frequently be approximated (to some extent) by LTV models. In such a case the parameters of LTV models, varying in time, are required to be estimated on-line.

v) Trajectories of estimated parameters can provide additional insight into the functionality of a given algorithm. For instance, a continuous trend in estimates leading to a lack of convergence can indicate that the model is overparametrised or/and non-identifiable.

vi) Trajectories of estimates can also lead to a better understanding of the physics of a given process. A variation of the parameter trajectory may be a result of changing operational conditions, i.e. changing state of the system, in which case such parameter can be interpreted as a so-called state-dependent, see (Young 1984). Subsequently, such state dependence can be incorporated into the model structure improving its modelling capabilities. For instance, a resistor can exhibit a significant dependence on the temperature.

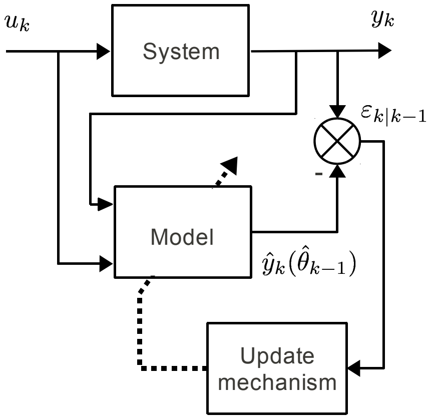

Figure 6.1: A diagrammatic representation of a recursive algorithm. For simplicity, disturbances are not included.

6.1 General structure of recursive algorithms

In general, a recursive estimation algorithm at time instance k can be described by the following recursion, see (Ikonen & Najim 2002), i.e.

$$\hat{\theta}_k = \hat{\theta}_{k-1} + L_k \left[y_k - \hat{y}_k(\hat{\theta}_{k-1}) \right], \tag{6.1}$$

where $\hat{\theta}_k$ is the new (i.e. present) estimate, $\hat{\theta}_{k-1}$ is the old (i.e. previous) estimate, L_k is a (possibly variable) correction gain and $y_k - \hat{y}_k(\hat{\theta}_{k-1})$ denotes the correction factor. The correction factor is defined as a difference between the actual system output y_k and that of the model $\hat{y}_k(\hat{\theta}_{k-1})$ obtained (or predicted) using a previous estimate of the parameter vector, i.e. $\hat{\theta}_{k-1}$. A diagrammatic visualisation of such a recursive procedure is given in Figure 6.1, cf. (Wellstead & Zarrop 1991), where the system is defined by a true parameter vector θ, the model by an estimate $\hat{\theta}_k$ and the term $\varepsilon_{k|k-1}$, being the correction factor, is called a one-step ahead prediction error[1]. The update mechanism is a method/algorithm chose to update the estimate based on $\varepsilon_{k|k-1}$, which, with the reference to equation (6.1), corresponds to the specification of the vector L_k. The diagonal dashed line with an arrow symbolises the property that the model, because it is continuously updated, changes over time.

6.2 Derivation of RLS

In order to develop a recursive formula for the LS method, consider the off-line LS estimate at time instance k given by

$$\hat{\theta}_k = \left(\sum_{i=1}^{k} \varphi_i \varphi_i^T \right)^{-1} \left(\sum_{i=1}^{k} \varphi_i y_i \right). \tag{6.2}$$

For ease of notation, denote the inverse of the covariance matrix at time instance k by P_k, i.e.

$$P_k = \left(\sum_{i=1}^{k} \varphi_i \varphi_i^T \right)^{-1}. \tag{6.3}$$

[1] The subscript notation $\varepsilon_{k|k-1}$ means that the signal ε_k has been obtained at time instance k by utilising (for prediction purpose) information up to and including time instance $k-1$ only, i.e. knowledge/data available at time instance k is not used.

It is noted that the inverse of P_k can be updated recursively as follows

$$P_k^{-1} = P_{k-1}^{-1} + \varphi_k \varphi_k^T \tag{6.4}$$

and, similarly, the expression $\sum_{i=1}^{k} \varphi_i y_i$ can also be expressed in a recursive fashion as

$$\sum_{i=1}^{k} \varphi_i y_i = \sum_{i=1}^{k-1} \varphi_i y_i + \varphi_k y_k. \tag{6.5}$$

This allows (6.2) to be re-write as follows

$$\hat{\theta}_k = P_k \left(\sum_{i=1}^{k-1} \varphi_i y_i + \varphi_k y_k \right). \tag{6.6}$$

Because the estimate at the time instance $k-1$ is given by

$$\hat{\theta}_{k-1} = P_{k-1} \sum_{i=1}^{k-1} \varphi_i y_i \tag{6.7}$$

the expression $\sum_{i=1}^{k-1} \varphi_i y_i$ can be written as

$$\sum_{i=1}^{k-1} \varphi_i y_i = P_{k-1}^{-1} \hat{\theta}_{k-1} \tag{6.8}$$

and substituting into equation (6.6) yields

$$\hat{\theta}_k = P_k \left(P_{k-1}^{-1} \hat{\theta}_{k-1} + \varphi_k y_k \right). \tag{6.9}$$

By using (6.4) it follows that

$$\begin{aligned} \hat{\theta}_k &= P_k \left[\left(P_k^{-1} - \varphi_k \varphi_k^T \right) \hat{\theta}_{k-1} + \varphi_k y_k \right] \\ &= \hat{\theta}_{k-1} + P_k \varphi_k \left(y_k - \varphi_k^T \hat{\theta}_{k-1} \right). \end{aligned} \tag{6.10}$$

Note the agreement with the general formula of a recursive algorithm given by (6.1), where, in this case, the predicted (one-step ahead) output is $\hat{y}_k(\hat{\theta}_{k-1}) = \varphi_k^T \hat{\theta}_{k-1}$ and $L_k = P_k \varphi_k$. The remaining task is to find a tractable method for updating the inverse of the covariance matrix, i.e. P_k, because the required inversion of P_k^{-1} at each time step can lead to a considerable computational burden (especially for large n_θ). This problem can be solved by employing the so-called matrix inversion lemma, defined as follows, see (Söderström & Stoica 1989):

Lemma 1 (Matrix inversion lemma) *If there exist inverses of matrices A, C and $C^{-1} + DA^{-1}B$ then it follows that*

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B\left(C^{-1} + DA^{-1}B\right)^{-1} DA^{-1}. \tag{6.11}$$

Proof can be found by multiplying the right-hand side of (6.11) *by expression $A + BCD$ and showing that it will yield a unity matrix.*

Applying the matrix inversion lemma to the update of P_k, i.e. equation (6.4), with

$$\begin{aligned} A &= P_{k-1}^{-1}, \\ B &= \varphi_k, \\ C &= 1, \\ D &= \varphi_k^T, \end{aligned} \tag{6.12}$$

leads to

$$\begin{aligned} P_k &= \left(P_{k-1}^{-1} + \varphi_k\varphi_k^T\right)^{-1} \\ &= P_{k-1} - P_{k-1}\varphi_k\left(1 + \varphi_k^T P_{k-1}\varphi_k\right)^{-1}\varphi_k^T P_{k-1} \\ &= P_{k-1} - L_k\varphi_k^T P_{k-1}, \end{aligned} \tag{6.13}$$

where the gain vector L_k defined by

$$L_k = P_{k-1}\varphi_k\left(1 + \varphi_k^T P_{k-1}\varphi_k\right)^{-1}. \tag{6.14}$$

Finally, by gathering expressions (6.10), (6.13) and (6.14) the overall RLS algorithm is summarised in the three steps as follows

$$L_k = P_{k-1}\varphi_k\left(1 + \varphi_k^T P_{k-1}\varphi_k\right)^{-1}, \tag{6.15}$$

$$P_k = P_{k-1} - L_k\varphi_k^T P_{k-1}, \tag{6.16}$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + P_k\varphi_k\left(y_k - \varphi_k^T\hat{\theta}_{k-1}\right), \tag{6.17}$$

or, alternatively, as

$$L_k = P_{k-1}\varphi_k \left(1 + \varphi_k^T P_{k-1}\varphi_k\right)^{-1}, \tag{6.18}$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + L_k \left(y_k - \varphi_k^T \hat{\theta}_{k-1}\right), \tag{6.19}$$

$$P_k = P_{k-1} - L_k \varphi_k^T P_{k-1}, \tag{6.20}$$

which is obtained by exploiting the property that $L_k = P_k \varphi_k$ (i.e. multiply L_k by $P_k P_k^{-1}$ and substitute equation (6.4) for P_k^{-1}).

Note that the inversion in the first step, i.e. (6.18), is, in fact, simply a scalar division, which decreases considerably the computational burden. Therefore, the first equation could also be written as $L_k = \frac{P_{k-1}\varphi_k}{1+\varphi_k^T P_{k-1}\varphi_k}$. Alternatively, the RLS algorithm can be expressed in two steps as

$$P_k = P_{k-1} - P_{k-1}\varphi_k \left(1 + \varphi_k^T P_{k-1}\varphi_k\right)^{-1} \varphi_k^T P_{k-1}, \tag{6.21}$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + P_k \varphi_k \left(y_k - \varphi_k^T \hat{\theta}_{k-1}\right). \tag{6.22}$$

By including the property given by equation (5.38) obtained in Subsection 5.1.3 for the off-line LS method, i.e. that $P_k^* = \sigma_e^2 P_k$, another alternative version of the RLS algorithm follows

$$L_k = P_{k-1}^* \varphi_k \left(\sigma_e^2 + \varphi_k^T P_{k-1}^* \varphi_k\right)^{-1}, \tag{6.23}$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + L_k \left(y_k - \varphi_k^T \hat{\theta}_{k-1}\right), \tag{6.24}$$

$$P_k^* = P_{k-1}^* - L_k \varphi_k^T P_{k-1}^*. \tag{6.25}$$

Note that the above algorithm[2] not only uses all the potentially available information regarding the noise variance but also provides a direct indication of the accuracy of the estimated parameters at each recursion via the error covariance matrix P_k^*. This feature combined with the fact that no matrix inverse is required forms a particularly useful method for practical, i.e. industrial, applications (Young 1984).

6.3 Estimator initialisation

In order to use the RLS algorithm initialisation of P_k and $\hat{\theta}_k$ at $k = 0$, i.e. P_0 and $\hat{\theta}_0$, is required. Sometimes *a priori* knowledge regarding the system parameters

[2] This version of the RLS algorithm is sometimes referred to as a stochastic RLS algorithm, whilst the standard RLS method is called a deterministic RLS algorithm.

exists. For instance, when the model is constructed using the white/grey-box approach, it is often possible to choose relatively good initial estimates (at least for some of the parameters). However, if no prior knowledge is available one possibility is to use the off-line LS for first $k_0 \geq n_\theta$ samples in order to obtain an estimate of the parameter vector. Note that only if $kgen_\theta$ is the covariance matrix P_k^{-1} invertible. Another possibility is to set the parameter vector to zero, i.e. $\hat{\theta}_0 = 0$, and to initialise the covariance matrix P_0 with some relatively large positive scalar value, i.e.

$$P_0 = \mu I, \tag{6.26}$$

where $\mu >> 0$. The choice of μ reflects the confidence in the *a priori* knowledge of $\hat{\theta}_0$ as follows: a large value, say $\mu = 10^3$, corresponds to a lack of confidence in $\hat{\theta}_0$, whilst a small value, say $\mu = 10$, implies the opposite, i.e. confidence that $\hat{\theta}_0$ is close to the true parameter vector. Consequently, if μ is large the parameter estimates adapt rapidly, when μ is small the adaptation is slower. In order to better understand this property consider the following expression, which follows from (6.2) when using initial values at $k = 0$, i.e.

$$\hat{\theta}_k = \left(P_0^{-1} + \sum_{i=1}^{k} \varphi_i \varphi_i^T \right)^{-1} \left(P_0^{-1} \hat{\theta}_0 + \sum_{i=1}^{k} \varphi_i y_i \right). \tag{6.27}$$

It can be observed that the relative importance of the initial values chosen decreases over time as the summation terms come to dominate and the corresponding estimates approach the off-line LS solution (Ljung & Söderström 1983). Another method to start up the estimator (see (Wellstead & Zarrop 1991)) is to assume initially that the system is an integrator of unity gain in the continuous-time domain, i.e. $\frac{1}{s}$, which corresponds to $\frac{T_s}{q-1} = \frac{T_s q^{-1}}{1-q^{-1}}$ in the discrete-time domain (using the ZOH), hence

$$\begin{aligned} a_1 &= -1, & a_i &= 0 \ \forall i \neq 1, \\ b_1 &= T_s, & b_i &= 0 \ \forall i \neq 1. \end{aligned} \tag{6.28}$$

Moreover, when utilising recursive methods to analyse data off-line, it is a common practice to re-run algorithm several times on the same data and use the final estimates obtained in previous runs as initial values for subsequent runs. In general, a particular choice of initial parameters should not influence the convergence behaviour of estimates, especially if N is large.

6.4 Residual vs. one-step ahead prediction error

It should be emphasised that the expression $\varepsilon_{k|k-1}$ present in the RLS algorithm is not equivalent to the residual ε_k, see (Wellstead & Zarrop 1991). While the one-step ahead error $\varepsilon_{k|k-1}$, sometimes called the *a priori* prediction error, is defined by

$$\varepsilon_{k|k-1} = y_k - \varphi_k^T \hat{\theta}_{k-1}, \tag{6.29}$$

the residual $\varepsilon_k = \varepsilon_{k|k}$, sometimes called the *posteriori* prediction error, is given by

$$\varepsilon_{k|k} = y_k - \varphi_k^T \hat{\theta}_k. \tag{6.30}$$

Note that the difference between expressions (6.29) and (6.30) is that the former is based on the estimate of the parameter vector at recursion $k-1$, while the latter utilises the estimate at recursion k. As the recursion progresses this difference diminishes, however it can be relatively significant in the initial period of the estimation procedure. By comparing (6.29) with (6.30) one obtains

$$\varepsilon_{k|k} = \varepsilon_{k|k-1} + \varphi_k^T \left(\hat{\theta}_{k-1} - \hat{\theta}_k \right), \tag{6.31}$$

which, by making use of equation (6.19), can be re-expressed as

$$\begin{aligned} \varepsilon_{k|k} &= \varepsilon_{k|k-1} \left(1 - \varphi_k^T L_k \right) \\ &= \varepsilon_{k|k-1} \left(1 + \varphi_k^T P_{k-1} \varphi_k \right)^{-1}. \end{aligned} \tag{6.32}$$

Note that the expression to be inverted in (6.31), as in the case of the RLS algorithm, is a scalar.

6.5 RLS with forgetting factor

The RLS algorithm can be modified in order to allow for tracking of time-varying parameters. This idea to reduce the relative importance of past measurements that are stored in compressed form inside the covariance expressions. Heuristically, such an approach can be interpreted as affecting directly the memory of the estimator by discounting the importance of old measurements, i.e. forgetting past data. Consider the following weighted LS cost function at time instance k, i.e.

$$\hat{\theta} = \arg\min_{\theta} \bar{V}_k(\theta), \tag{6.33}$$

where

$$\bar{V}_k(\theta) = \frac{1}{2}\varepsilon^T \Lambda \varepsilon = \frac{1}{2}\sum_{i=1}^{k} \lambda^{k-i}\varepsilon_i^2.$$

The scalar $\lambda \in (0,\ 1]$ is called the forgetting factor (typically $\lambda \in (0.95,\ 0.995)$) and $\Lambda \in \mathbb{R}^{N\times N}$ is a diagonal weighting matrix given by

$$\Lambda = \begin{bmatrix} \lambda^{k-1} & & & \\ & \ddots & & \\ & & \lambda^1 & \\ & & & \lambda^0 \end{bmatrix} = \text{diag}\begin{bmatrix} \lambda^{k-1} & \dots & \lambda^1 & \lambda^0 \end{bmatrix}, \tag{6.34}$$

where diag $\left[\cdot\right]$ denotes a diagonal matrix. The idea of discounting old measurements can be seen better by re-expressing (6.33) as follows

$$\begin{aligned} \bar{V}_k(\theta) &= \frac{1}{2}\sum_{i=1}^{k}\lambda^{k-i}\left(y_i - \varphi_i^T\theta\right)^2 = \frac{1}{2}\lambda\sum_{i=1}^{k-1}\lambda^{k-1-i}\left(y_i - \varphi_i^T\theta\right)^2 + \frac{1}{2}\left(y_k - \varphi_k^T\theta\right)^2 \\ &= \lambda\bar{V}_{k-1}(\theta) + \frac{1}{2}\left(y_k - \varphi_k^T\theta\right)^2. \end{aligned} \tag{6.35}$$

By considering expression (6.35) it is observed that the importance of past data is progressively reduced because $\lambda \leq 1$.

Carrying out reasoning analogous to that presented in Subsection 5.1.2, it can be shown that the minimum of the weighted LS cost function (6.33) at time instant k is given by

$$\hat{\theta}_k = \left(\Phi^T\Lambda\Phi\right)^{-1}\Phi^T\Lambda Y, \tag{6.36}$$

or, alternatively, by

$$\hat{\theta}_k = \left(\sum_{i=1}^{k}\lambda^{k-i}\varphi_i\varphi_i^T\right)^{-1}\left(\sum_{i=1}^{k}\lambda^{k-i}\varphi_i y_i\right). \tag{6.37}$$

By following a methodology similar to that used for the derivation of the RLS in Section 6.2, the weighted covariance matrix, denoted with a bar, can be written as

$$\bar{P}_k^{-1} = \sum_{i=1}^{k} \lambda^{k-i} \varphi_i \varphi_i^T = \lambda \sum_{i=1}^{k-1} \lambda^{k-1-i} \varphi_i \varphi_i^T + \varphi_k \varphi_k^T = \lambda \bar{P}_{k-1}^{-1} + \varphi_k \varphi_k^T, \tag{6.38}$$

which leads to

$$\hat{\theta}_k = \bar{P}_k \left(\lambda \sum_{i=1}^{k-1} \lambda^{k-1-i} \varphi_i y_i + \varphi_k y_k \right). \tag{6.39}$$

Since the estimate at time instant $k-1$ is given by

$$\hat{\theta}_{k-1} = \bar{P}_{k-1} \sum_{i=1}^{k-1} \lambda^{k-1-i} \varphi_i y_i, \tag{6.40}$$

the term $\sum_{i=1}^{k-1} \lambda^{k-1-i} \varphi_i y_i$ can be expressed as

$$\bar{P}_{k-1}^{-1} \hat{\theta}_{k-1} = \sum_{i=1}^{k-1} \lambda^{k-1-i} \varphi_i y_i. \tag{6.41}$$

By substituting equation (6.41) into (6.39) one obtains

$$\hat{\theta}_k = \bar{P}_k \left(\lambda \bar{P}_{k-1}^{-1} \hat{\theta}_{k-1} + \varphi_k y_k \right), \tag{6.42}$$

which, by using (6.38), leads to

$$\begin{aligned} \hat{\theta}_k &= \bar{P}_k \left[\left(\bar{P}_k^{-1} - \varphi_k \varphi_k^T \right) \hat{\theta}_{k-1} + \varphi_k y_k \right] \\ &= \hat{\theta}_{k-1} + \bar{P}_k \varphi_k \left(y_k - \varphi_k^T \hat{\theta}_{k-1} \right). \end{aligned} \tag{6.43}$$

In order to update $\bar{P}_k$ recursively the matrix inversion lemma is used, cf. Section 6.2 and equation (6.11), with the following settings

$$\begin{aligned} A &= \lambda \bar{P}_{k-1}^{-1}, \\ B &= \varphi_k, \\ C &= 1, \\ D &= \varphi_k^T. \end{aligned} \tag{6.44}$$

This leads to

$$\left(\lambda\bar{P}_{k-1}^{-1}+\varphi_k\varphi_k^T\right)^{-1}=\frac{1}{\lambda}\left[\bar{P}_{k-1}-\bar{P}_{k-1}\varphi_k\left(\lambda+\varphi_k^T\bar{P}_{k-1}\varphi_k\right)^{-1}\varphi_k^T\bar{P}_{k-1}\right], \tag{6.45}$$

which when combined with equation (6.43) gives the RLS algorithm with (fixed) forgetting factor, i.e.

$$L_k=\bar{P}_{k-1}\varphi_k\left(\lambda+\varphi_k^T\bar{P}_{k-1}\varphi_k\right)^{-1}, \tag{6.46}$$

$$\hat{\theta}_k=\hat{\theta}_{k-1}+L_k\left(y_k-\varphi_k^T\hat{\theta}_{k-1}\right), \tag{6.47}$$

$$\bar{P}_k=\frac{1}{\lambda}\left(\bar{P}_{k-1}-L_k\varphi_k^T\bar{P}_{k-1}\right). \tag{6.48}$$

Note that since the forgetting factor $\lambda\in(0,\ 1]$, it has the effect of inflating the covariance matrix $\bar{P}_k$. Therefore, the gain vector L_k is kept larger, which results in a larger correction of the parameter vector. Consequently, the covariance matrix will not tend to zero as k increases and the RLS algorithm will always be able to track potential variations in the system parameters. Observe that it implies that, even if the true parameters are constant, the corresponding estimates will not be consistent as $k\to\infty$. The adaptive RLS algorithm (6.46)-(6.48) reduces to that given by equations (6.18)-(6.20), i.e. the standard non-adaptive RLS algorithm, if no forgetting of past information occurs, i.e. when $\lambda=1$.

An alternative modification of the RLS algorithm used to cope with tracking of time-varying parameters is the implementation of a moving (or sliding) rectangular window technique, see (Young 1984) for details. In this case the estimates are calculated based on an interval of past data with a pre-specified length, say h. The required modification of the standard RLS algorithm comprises the incorporation of an additional step in which the data received at the recursion $k-h$ is removed.

6.6 Memory of RLS with forgetting factor

Because the forgetting factor λ influences the memory of the estimator, it is of interest to estimate approximately its length. This can be obtained by considering the weighting at time instance k, i.e.

$$\lambda^{k-i}=e^{\ln\lambda^{k-i}}\approx e^{-(k-i)(1-\lambda)}, \tag{6.49}$$

which holds because $\ln \lambda \approx 1 - \lambda$ at $\lambda \approx 1$. Consequently, expression (6.49) can be interpreted as an impulse response of a first order system with a time constant given by

$$M = \frac{1}{1-\lambda}. \tag{6.50}$$

Therefore, the measurements older than M are collectively assigned a weight of only $e^{-1} \approx 36\%$, see (Ljung 1999). This also explains the reason the forgetting factor of the form (6.49) is often called an exponential forgetting factor. Figure 6.2 shows weightings corresponding to different forgetting factors, where k is at $i = 0$. The time constant M is usually referred to as the memory of the estimator. If the system remains relatively constant over M samples, the appropriate value of the forgetting factor can be obtained from (6.50), i.e.

$$\lambda = \frac{M-1}{M}. \tag{6.51}$$

The choice of the forgetting factor results in a trade-off between the ability to track time-varying parameters and sensitivity of estimates. Because all elements of the covariance matrix are scaled by the same factor, one disadvantage of such approach is that all estimates are considered to exhibit equal variations, which may not be the case in practice. In the case if some parameters are known to vary with a different speed to others it may be better to use individually adjusted forgetting factors. For instance one parameter may vary, whilst the other can be constant. To realise this idea, it is possible to scale selected diagonal elements of the covariance matrix by a chosen forgetting factor (Wellstead & Zarrop 1991)[3]. Another problem occurs in situations when the input signal is not sufficiently exciting, i.e. it does not provide new information to the estimator. In this case $\bar{P}_k \approx \bar{P}_{k-1}/\lambda$, cf. (6.48), which means that the covariance matrix will be continuously inflated leading to a phenomena called a covariance blow-up and, consequently, a potential numerical overflow. A useful measure which can be used to detect this behaviour is to monitor the trace of the covariance matrix $\bar{P}_k$, i.e. $\mathrm{tr}(\bar{P}_k)$, see Subsection 6.8 later.

6.7 Variable forgetting factor

The problem of the appropriate choice of the forgetting factor can be helped to some extent by allowing the forgetting factor λ to be time-varying or adaptive,

[3] In fact, this approach is conceptually similar to the KF tuned for parameter estimation introduced later in Section 6.9, cf. also Section 7.4.

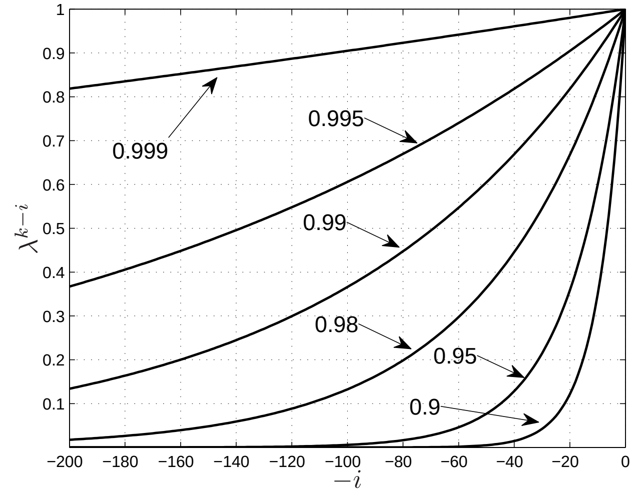

Figure 6.2: Illustration of weighting corresponding to different forgetting factors λ.

i.e. λ_k. In such a case the memory of the estimator can be adjusted based on, for instance, a value of the instantaneous one-step ahead prediction error. The rationale is that if $\varepsilon_{k|k-1}$ increases it means that the model is not capable of predicting accurately the system output, therefore the parameter estimates should be allowed to adapt. If $\varepsilon_{k|k-1}$ decreases the present estimates are good, therefore they should be kept and there is no need for adaptation. Some approaches selected from the literature for the implementation of this strategy are presented below:

- An adaptive variable forgetting factor proposed in (Fortescue, Kershenbaum & Ydstie 1981) is given by

$$\lambda_k = 1 - \frac{(1 - \varphi_k^T \bar{P}_{k-1} \varphi_k)\varepsilon_{k|k-1}^2}{\sigma_e^2 M_0}, \tag{6.52}$$

  where M_0 denotes the initial memory of the estimator. Additionally, care must be taken to ensure that $\lambda_k \in (0,\ 1]\ \forall k$, hence it is advisable to include additional constraints, i.e.

$$\lambda_k = \begin{cases} 1 & \text{if } \lambda_k > 1 \\ \lambda_0 & \text{if } \lambda_k < 0 \end{cases}, \tag{6.53}$$

where λ_0 denotes a default value of the forgetting factor.

- An adaptive variable forgetting factor proposed in (Wellstead & Zarrop 1991) is given by

  $$\lambda_k = \frac{s_{k-1}}{s_k}, \tag{6.54}$$

  where

  $$s_k = \frac{\tau - 1}{\tau} s_{k-1} + \frac{\varepsilon^2_{k|k-1}}{\tau}. \tag{6.55}$$

  The scalar τ determines the rate of adaptation and s_k is a weighted average of the past values of $\varepsilon^2_{k|k-1}$. As in the previous case, it is advisable to append constraints given by (6.53).

- Start-up forgetting factor is useful during the initial period of the estimation. Initially, λ_k should be small, thus allowing for a quick adaptation and subsequently converge to unity as time progresses. This means that any undesired effects present in the initial period of the estimation are quickly discounted, i.e. forgotten. Such a variable forgetting factor, see (Young 1984) and (Wellstead & Zarrop 1991), is given by

  $$\lambda_k = \lambda_0 + (1 - \lambda_0)\left(1 - e^{-\frac{k}{\tau}}\right), \tag{6.56}$$

  where λ_0 denotes the initial value of the forgetting factor at $k = 0$, τ is the time constant of the forgetting factor determining the speed of change of λ_k. Expression (6.56) can be conveniently re-written as the following recursion

  $$\lambda_k = e^{-\frac{1}{\tau}} \lambda_{k-1} + (1 - e^{-\frac{1}{\tau}}). \tag{6.57}$$

Note that as $k \to N$ the forgetting factor tends to unity. Further, it is possible to combine the start-up variable forgetting factor with any other adaptive or constant forgetting factors.

6.8 Other modifications of RLS

This subsection describes some other possible modifications of the standard RLS algorithm, which can be found in the literature.

- Constant trace method described in (Wellstead & Zarrop 1991) - the idea of this method is to keep the trace of the covariance matrix P_k constant, hence preventing it from blowing up. The approach is to add to P_k at each recursion a matrix $R_k \in \mathbb{R}^{n_\theta \times n_\theta}$, chosen to ensure that $\mathrm{tr}(P_k) = \mathrm{tr}(P_{k-1})$, i.e.

  $$P_k = P_{k-1} - P_{k-1}\varphi_k \left(1 + \varphi_k^T P_{k-1}\varphi_k\right)^{-1} \varphi_k^T P_{k-1} + R_k. \tag{6.58}$$

  The appropriate value of R_k is found as follows, i.e.

  $$\mathrm{tr}(P_k) = \mathrm{tr}(P_{k-1}) - \mathrm{tr}\left(P_{k-1}\varphi_k \left(1 + \varphi_k^T P_{k-1}\varphi_k\right)^{-1} \varphi_k^T P_{k-1}\right) + \mathrm{tr}(R_k), \tag{6.59}$$

  which by including the constraint $\mathrm{tr}(P_k) = \mathrm{tr}(P_{k-1})$ leads to

  $$\mathrm{tr}(R_k) = \mathrm{tr}\left(P_{k-1}\varphi_k \left(1 + \varphi_k^T P_{k-1}\varphi_k\right)^{-1} \varphi_k^T P_{k-1}\right). \tag{6.60}$$

  This can be further simplified yielding

  $$\mathrm{tr}(R_k) = \mathrm{tr}\left(\frac{\varphi_k^T P_{k-1}^2 \varphi_k}{1 + \varphi_k^T P_{k-1}\varphi_k}\right) = \frac{\varphi_k^T P_{k-1}^2 \varphi_k}{1 + \varphi_k^T P_{k-1}\varphi_k}. \tag{6.61}$$

  A matrix which satisfies (6.61), hence ensures a constant trace condition is, for instance, given by

  $$R_k = \frac{\mathrm{tr}(R_k)}{n_\theta} I, \tag{6.62}$$

  where the identity matrix is of dimension n_θ. Note that although the constant trace method prevents blow-up of the covariance matrix, the lack of new information can lead the it to become almost singular. To cope with this problem an excitation should be input to the system or/and the matrix R_k should be bounded from below (Wellstead & Zarrop 1991).

- Directional forgetting described in (Wellstead & Zarrop 1991) - in this case covariance blow-up is avoided by using the forgetting factor only in directions of the parameter space for which new information enters the system. Using the directional forgetting the update of P_k^{-1} is defined as

$$P_k^{-1} = P_{k-1}^{-1} + r_{k-1}\varphi_k\varphi_k^T, \tag{6.63}$$

by using the matrix inversion lemma with $C = r_{k-1}$, cf. Section 6.2 and equation (6.11). This yields the following update of the covariance matrix P_k

$$P_k = P_{k-1} - P_{k-1}\varphi_k \left(\frac{1}{r_{k-1}} + \varphi_k^T P_{k-1}\varphi_k\right)^{-1} \varphi_k^T P_{k-1}. \tag{6.64}$$

While the standard forgetting factor controls the outflow of old information (i.e. forgetting) in all directions, the directional forgetting factor controls the inflow of the new information (i.e. remembering) associated with the rank one update of P_k via $\varphi_k\varphi_k^T$. A choice for r_k proposed in (Wellstead & Zarrop 1991) is as follows

$$r_{k-1} = \lambda - (1-\lambda)\left(\varphi_k^T P_{k-1}\varphi_k\right)^{-1}, \tag{6.65}$$

where λ is the standard constant forgetting factor. Note that when $\lambda = 1$ the RLS algorithm with no forgetting is obtained.

- Covariance reset techniques proposed in (Balmer 1986) and (Hagglund 1993) - the main motivation for using a covariance reset is to improve the speed of parameters tracking. Note that in the case of variable forgetting factors even if λ is relatively small, some memory of the past data is still retained in the covariance matrix. Consequently, in order to track rapid changes in the model parametrisation all past information is required to be discarded, i.e. forgotten. This is realised via a reset of the covariance matrix. Such a reset can be obtained by setting the covariance matrix to an identity matrix with relatively large values on its diagonal, i.e. $P_k = \mu I$. This is similar to the initialisation of the estimator, however in this case the value of μ is typically chosen to be smaller, e.g. $\mu = 10^2$. The reset can be realised when operation conditions change or if some prior knowledge regarding the occurrence of nonlinearities exists (James 1998). The covariance reset can be implemented as follows

$$P_k = \begin{cases} P_{k-1} - L_k\varphi_k^T P_{k-1} & \text{if } k \neq mn_\theta \\ \mu I & \text{if } k = mn_\theta \end{cases}, \tag{6.66}$$

where $m >> n_\theta$ corresponds to the interval between successive reset actions. The choice of the resetting interval m as well as the value of μ is crucial since bad tuning can lead to undesirable transient effects in the estimates. Another approach is to use a logical reset based on the detection of the set point change, i.e.

$$P_k = \begin{cases} P_{k-1} - L_k \varphi_k^T P_{k-1} & \text{if } u_k = u_{k-1} \\ \mu I & \text{if } u_k \neq u_{k-1} \end{cases}, \tag{6.67}$$

where u_k is the reference signal. Additionally, care must be taken in the case where the set point changes frequently or in a continuous fashion. Note, if it is known that only a certain parameter varies rapidly the covariance reset method can be applied to a diagonal element of the covariance matrix corresponding to this particular parameter exclusively.

6.9 RLS with an inherent mechanism for tracking time-varying parameters (KF tuned for parameter estimation)

In this section the RLS algorithm with an inherent mechanism for tracking time-varying parameters is described. The main difference between this approach and techniques introduced in previous sections is that, in this case, the time-varying nature of parameters will be incorporated directly into the model. This means that the underlying model of the system is considered as time-varying. Note that this is in contrast to the RLS algorithm with forgetting factor, where the model was postulated to be time-invariant, i.e. $\theta_k = \theta_{k-1}$. The adaptivity properties of the RLS were obtained in a rather heuristic or engineering fashion by not allowing the elements of the covariance matrix to converge to zero (via inflation of the covariance matrix). In the case of the algorithm described in this section adaptivity is achieved by describing the evolution of the model parameters in a stochastic framework. The reasoning and derivation presented have both been taken from (Young 1974).

Consider the following model describing a random walk evolution of the parameters

$$\theta_k = \theta_{k-1} + v_k, \tag{6.68}$$

where $v_k \in \mathbb{R}^{n_\theta}$ is a vector consisting of serially uncorrelated random variables of zero mean, i.e. $E[v_k] = 0$ and $E[v_k v_j] = \Sigma_v \delta_{kj}$. The expression (6.68) corresponds to a so-called (multivariable) Markov model[4]. If additionally v_k is assumed to be Gaussian distributed, then equation (6.68) corresponds to a so-called Gauss-Markov model. The system model is considered to be given by the following regression

$$y_k = \theta_k^T \varphi_k + e_k, \tag{6.69}$$

where e_k is a scalar sequence of zero mean purely random equation errors having variance $E[e_k^2] = \sigma_e^2$ and the parameter vector θ_k is allowed to vary (between sampling instances) with accordance to (6.68). Assuming that the prior knowledge regarding the values of Σ_v and σ_e^2 exists it is possible to utilise this information to make *a priori* predictions of both the parameter vector and the error covariance matrix. These predictions are denoted by $\hat{\theta}_{k|k-1}$ and $P^*_{k|k-1}$, respectively, where the subscript notation $k|k-1$ means that a given expression depends on the measurements up to (and including) time instance $k-1$. The prediction $\hat{\theta}_{k|k-1}$ is found by considering the expected value of the evolution of the parameter vector (6.68), which, since $E[v_k] = 0$, is given by

$$E[\theta_k] = \theta_{k-1}. \tag{6.70}$$

Consequently, the *a priori* prediction of θ_k based on the past information at $k-1$ and the knowledge of the parameter evolution law (6.68) is given by

$$\hat{\theta}_{k|k-1} = \hat{\theta}_{k-1|k-1}, \tag{6.71}$$

which is simply the estimate of the parameter vector at the previous time instance. In order to find an analogous expression for the *a priori* prediction of the error covariance matrix P_k^*, consider the *a priori* prediction error of the parameter estimate, denoted by $\tilde{\theta}_{k|k-1}$, i.e.

$$\tilde{\theta}_{k|k-1} = \hat{\theta}_{k|k-1} - \theta_k. \tag{6.72}$$

With reference to (6.68) and (6.71), this can be transformed into

$$\tilde{\theta}_{k|k-1} = \hat{\theta}_{k-1|k-1} - \theta_{k-1} - v_k = \tilde{\theta}_{k-1|k-1} - v_k, \tag{6.73}$$

where $\tilde{\theta}_{k-1|k-1}$ denotes the *posteriori* parameter estimation error at the time instance $k-1$. Equation (6.73) is now used to determine the *a priori* prediction

[4] The main property of the Markov process is that it is memoryless, i.e. loosely speaking its future state depends upon the present state exclusively and not on past states.

of the covariance matrix P_k^* as follows

$$\begin{aligned}P_{k|k-1}^* &= E[\tilde{\theta}_{k|k-1}\tilde{\theta}_{k|k-1}^T] = E[(\tilde{\theta}_{k-1|k-1} - v_k)(\tilde{\theta}_{k-1|k-1} - v_k)^T] \\ &= E[\tilde{\theta}_{k-1|k-1}\tilde{\theta}_{k-1|k-1}^T] + E[v_k v_k^T] - E[\tilde{\theta}_{k-1|k-1} v_k^T] - E[v_k \tilde{\theta}_{k-1|k-1}^T] \\ &= P_{k-1|k-1}^* + \Sigma_v. \qquad (6.74)\end{aligned}$$

The last equality follows from the fact that $\tilde{\theta}_{k-1|k-1}$ comprises the signal v_{k-1} and because v_k is white, $E[\tilde{\theta}_{k-1|k-1} v_k^T] = E[v_k \tilde{\theta}_{k-1|k-1}^T] = 0$.

Consequently, equations (6.71) and (6.74) provide the sought expressions for the prior predictions of the parameter vector and the error covariance matrix. These can be combined with the recursive equations of the RLS algorithm, see (6.23)-(6.25), to yield the following two phase prediction-correction approach for parameter estimation, which is summarised below:

Prediction step:

$$\hat{\theta}_{k|k-1} = \hat{\theta}_{k-1|k-1}, \qquad (6.75)$$

$$P_{k|k-1}^* = P_{k-1|k-1}^* + \Sigma_v. \qquad (6.76)$$

Correction step:

$$L_k = P_{k|k-1}^* \varphi_k (\varphi_k^T P_{k|k-1}^* \varphi_k + \sigma_e^2)^{-1}, \qquad (6.77)$$

$$\hat{\theta}_{k|k} = \hat{\theta}_{k|k-1} + L_k (y_k - \varphi_k^T \hat{\theta}_{k|k-1}), \qquad (6.78)$$

$$P_{k|k}^* = (I - L_k \varphi_k^T) P_{k|k-1}^*. \qquad (6.79)$$

In fact, the above algorithm, denoted KFPE, is equivalent to the KF tuned for parameter estimation, see Section 7.4 later, where this relationship is demonstrated. Provided that σ_e^2 and Σ_v are both known, it means the KFPE is optimal in the sense of minimising the estimation error $\tilde{\theta}_k$.

The KFPE requires the specification of σ_e^2 and Σ_v. While σ_e^2 reflects level of noise contamination in the measurements, the diagonal elements of Σ_v are chosen to reflect the expected rates of variation of individual parameters and correspond to their expected variances, denoted var$(\cdot)$, i.e.

$$\Sigma_v = \text{diag}\begin{bmatrix}\text{var}(a_1) & \ldots & \text{var}(b_{n_b}) & \text{var}(b_0) & \ldots & \text{var}(b_{n_b})\end{bmatrix}. \qquad (6.80)$$

Note that this setting is valid if the parameters evolve according to a random walk process (6.68) and if additionally their corresponding variances are known. Otherwise, appropriate values to construct Σ_v must be found experimentally.

Analogously to the RLS with a forgetting factor, the KFPE allows the trajectories of time-varying parameters to be tracked by inflating the covariance matrix. Because $L_k = P^*_{k|k}\varphi_k$ the gain L_k will not tend to zero as $k \to \infty$. Therefore, the parameter estimate will be continuously subject to corrections, i.e. it will be continuously updated thus inconsistent. The difference between the KFPE and the RLS with a forgetting factor is that while in the case of the RLS this property is achieved by a post-division of $P^*_{k|k}$ by a forgetting factor λ, in the case of the KFPE a diagonal matrix is pre-added to $P^*_{k|k-1}$ in the prediction step of the algorithm. However, the final effect resulting from both approaches is similar. The other difference between the two techniques is that the KFPE is inherently more flexible than the RLS with a forgetting factor, because different expected rates of variation can be specified for each parameter. For instance, if a particular parameter is known to be constant the corresponding diagonal value of Σ_v can be simply set to null. A similar effect can be achieved heuristically in the case of the RLS by dividing the diagonal elements of $P^*_{k|k}$ by different forgetting factors, see Section 6.6. Additionally, the KPFE provides a statistical interpretation of the initial values provided to the RLS algorithm. Namely, the initial value of the parameter vector, i.e. $\hat{\theta}_0$, is the prior mean and the initial value of P^*_0 is the prior error covariance matrix (Ikonen & Najim 2002). Furthermore, the KPFE substantiates the rationale for including the variance of the noise into the RLS algorithm, cf. Section 6.2 and equations (6.23)-(6.25).

6.10 Pseudo-regression and RLS

The RLS described in this chapter has been developed for the case of an ARX model structure. When the RLS algorithm is directly used for the purpose of estimating parameters of other linear model structures, then the estimates obtained will be biased in general (excluding the specific cases described in Section 5.2). This is because the necessary assumption of the whiteness of equation errors is violated, see Section 5.1.3. However, it is possible to modify the RLS method and to obtain unbiased estimates by making use of the idea of the optimal one-step ahead predictor described in Section 3.2.4. Considering the RLS alagorithm given by equations (6.18)-(6.20), this modification requires the prediction in (6.19) to be optimal. This means that the product $\varphi_k^T\hat{\theta}_{k-1}$ is to be replaced by the corresponding (to a particular model structure) optimal one-step ahead predictor. As an example consider the ARMAX model structure, where the corresponding predictor is given by equation (3.96), see Subsection

3.2.4. Note, that in this case the regressor vector contains residuals, which can be generated based on $\hat{\theta}_{k-1}$ as *a posteriori* prediction errors. Moreover, the parameter vector is appended with the coefficients of polynomial $C(q^{-1})$, which also have to be estimated together with the a and b parameters. Because the residual at time instance k is generated based on the estimated parameter vector at time instance $k-1$, the overall procedure corresponds to a pseudo-regression. Modification of the RLS method for other model structures is analogous. In general, it involves, first, approximations of unmeasurable signals (present in the corresponding regressor vectors) that are made based on a recursively estimated parameter vector as the recursion progresses and estimation of the parameter vector extended with parameters of polynomials that model the equation error.

Questions

- Explain the motivation for recursive estimation.
- Draw a diagram of a general recursive algorithm.
- Explain the idea of the matrix inversion lemma and its application in the derivation of the RLS algorithm.
- Comment on differences between standard and stochastic RLS algorithms.

- Discuss initialisation of the RLS algorithm and its significance.
- Comment on differences between the one-step ahead prediction error and the residual.
- Explain the motivation for the introduction of a forgetting factor. Comment on the notion of estimator memory.
- Explain motivations for using variable forgetting factors and provide examples of different approaches to this problem.
- Discuss the constant trace method, directional forgetting and covariance reset technique. Explain the motivations for their usage.
- Explain how to incorporate an inherent mechanism for tracking time-varying parameters into the RLS method, which leads to the KFPE. Discuss estimator initialisation and analogies with the RLS algorithm with fixed forgetting factor.
- Describe modifications that are required for the RLS algorithm to yield unbiased estimates for model structures other than ARX.

Chapter 7

Kalman filtering

7.1 Introduction

This section provides a brief introduction to the Kalman filtering. First, the notion of an observer is introduced, which, in a non-deterministic setup leads to the KF. Subsequently, some modifications of the standard KF are described including the steady-state KF, KF tuned for parameter estimation and the extended KF (EKF).

7.2 Notion of an observer

In contrast to the parameter estimation task, where the goal is to estimate the unknown vector of the parameters describing the system, state estimation is concerned with the estimation of system states under the assumption that the system model is available. The main motivations for using the state estimation are as follows, see (Nise 2008) and (Dutton et al. 1997), i.e.

- Some controllers rely on access to system states, e.g. state variable feedback controllers.
- System states usually possess a physical meaning, hence can provide valuable information regarding the system behaviour and its current state. This is useful, for instance, for the purpose of condition monitoring and fault detection.

- Typically direct access to system states is infeasible or impractical due to the costs and installation complexity involved with the additional hardware required. As an example of such a situation consider a state corresponding to a temperature inside a hot furnace or a diesel engine.

- A plant model may result from an identification procedure that is expressed in terms of an input-output transfer function, which can be converted to a state-space description. Typically, in such a case the only states which can be interpreted as physically meaningful, and hence measured directly with hardware, are those related to the system output. Because other states do not possess a clear physical meaning, it is not possible to measure them.

Consequently, it is desirable to be able to reconstruct the system states using an algorithm, which is called an observer, because it observers (or mimics) the system states. The concept of an observer is similar to that of a parameter estimator with the crucial difference that, while the latter estimates parameters, the former estimates states, i.e. an observer is a state estimator.

7.2.1 Identity observer

Because an observer it is required to provide estimates of the system states, the simplest observer is just a copy (model) of the true system. Recalling the state-space representation, see Subsection 3.2.2 and equations (3.19)-(3.20), means that the observer is given by

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k, \tag{7.1}$$

$$\hat{y}_k = C\hat{x}_k, \tag{7.2}$$

where a hat indicates that the state vector x_k and the system output y_k are reconstructed (or estimated). For simplicity the feedforward term D and the dependency of the matrices on the parameter vector are both omitted. By inspecting the difference between the true state vector and that estimated via observer, denoted $\tilde{x}_k = x_k - \hat{x}_k$, one obtains

$$\tilde{x}_{k+1} = x_{k+1} - \hat{x}_{k+1} = A\tilde{x}_k, \tag{7.3}$$

$$\tilde{y}_k = y_k - \hat{y}_k = C\tilde{x}_k. \tag{7.4}$$

Assuming that the actual process is stable the state estimation difference decreases as time progresses, hence the observer output will eventually approach the true output. However, this property holds only asymptotically and the speed of convergence can be slow. In fact, in such a case the speed of convergence is the same as the transient response of the actual system. The other major disadvantage of such an approach is that it requires the system model, i.e. the matrices A, B and C, to be extremely accurate.

7.2.2 Luenberger observer

To increase the speed of convergence and to simultaneously allow for the existence of potential discrepancies between the actual plan and its model, the idea of a feedback is incorporated into the observer. This allows the state estimates to be corrected based on the mismatch between the output of the observer and that of the actual process. The design procedure comprises of the selection of a vector of gains

$$L = \begin{bmatrix} l_1 & l_2 & \dots & l_n \end{bmatrix}^T \in \mathbb{R}^n \tag{7.5}$$

that will yield the desired transient performance of the observer, i.e.

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - \hat{y}_k), \tag{7.6}$$

$$\hat{y}_k = C\hat{x}_k. \tag{7.7}$$

Note the agreement with the general formula for a recursive algorithm given by equation (6.1) in Section 6.1. A comparison of the difference between the true and the estimated state vector gives

$$\tilde{x}_{k+1} = A\tilde{x}_k - L(y_k - \hat{y}_k), \tag{7.8}$$

$$\tilde{y}_k = y_k - \hat{y}_k = C\tilde{x}_k. \tag{7.9}$$

This leads to the following expression

$$\tilde{x}_{k+1} = (A - LC)\tilde{x}_k, \tag{7.10}$$

where the speed of convergence of $\tilde{x}_k$ to zero can be influenced by a choice of the gain vector L. The first step in the procedure of choosing L is to form the characteristic equation of the state estimation error, i.e.

$$\det\left[qI - (A - LC)\right] = 0 \tag{7.11}$$

and, the second step is to choose the eigenvalues to yield the desired transient response. The eigenvalues are specified so that the observer is much faster than the actual system (typically 5 to 10 times). The design of the observer can be conveniently carried out using a system model described in an observer canonical or in a phase variable form. Other state-space representations can lead to quite complex calculations. Note, the incorporation of the feedback allows for some degree of imprecision to be present in the matrices A and B but the matrix C, because it appears in the feedback action, still has to be known accurately. Fortunately, this condition is not difficult to satisfy as typically the elements of the matrix C are chosen so that one of the states is the system output (Dutton et al. 1997). The feedback observer described is called the Luenberger observer.

A possible refinement of the observer given by (7.6)-(7.7) is to remove the inherent delay, which is present because the state vector depends on the measurements up to (and including) time instance $k-1$ (Åström & Wittenmark 1997). This undesired property is apparent if the observer equations are expressed more precisely as follows

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + Bu_k + L(y_k - \hat{y}_k), \tag{7.12}$$

$$\hat{y}_k = C\hat{x}_{k|k-1}. \tag{7.13}$$

The delay is removed by considering the following expression for the estimate of the state vector, i.e.

$$\hat{x}_{k|k} = A\hat{x}_{k-1|k-1} + Bu_{k-1} + L\left[y_k - C(A\hat{x}_{k-1|k-1} + Bu_{k-1})\right] \tag{7.14}$$

$$= (I - LC)(A\hat{x}_{k-1|k-1} + Bu_{k-1}) + Ly_k. \tag{7.15}$$

The error corresponding to the reconstructed state vector is given by

$$\tilde{x}_{k|k} = x_k - \hat{x}_{k|k} = (A - LCA)\tilde{x}_{k-1|k-1}, \tag{7.16}$$

where, as previously, the transient of $\tilde{x}_{k-1|k-1}$ is controlled by the choice of the gain vector L. The output equation is given by

$$\hat{y}_k = C\hat{x}_{k|k}. \tag{7.17}$$

Notice that

$$y_k - C\hat{x}_{k|k} = C\tilde{x}_{k|k} = (1 - CL)CA\tilde{x}_{k-1|k-1}. \tag{7.18}$$

If L is chosen such that $CL = 1$ then $y_k = C\hat{x}_{k|k}$ and the system output is estimated without any error. This observation also makes possible the elimination of one state from (7.14) and the design of a so-called reduced order observer, see (Dutton et al. 1997) for details.

7.2.3 Observability

In order to be able to design an observer the reconstructed states all need to be observable. Loosely, this means that it must be feasible to infer the information regarding the system states based on the input-output data. If any of the states does not influence the system output then its value cannot be deduced from observing the output. A formal definition of observability is as follows (Dutton et al. 1997): a system is (completely) state observable if the initial value of the state vector, i.e. x_0, can be determined from the knowledge of u_k and y_k in a finite time interval. The observability of the system representation can be determined by considering the rank of the so-called observability matrix O_M defined as

$$O_M = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}. \tag{7.19}$$

If $\operatorname{rank} O_M = n$ (or, equivalently, if $\det O_M \neq 0$) the system model is said to be observable. Otherwise, the system model is unobservable, which implies that it is not possible to reconstruct the system states. It is emphasised that the observability property is related to a particular state-space representation and not to the system itself. Therefore, whilst one state-space representation of a given system may be observable, another (different) state-space representation of the same system may not.

7.3 Kalman filter

7.3.1 Introduction

A drawback of the feedback observer described in the previous section is that any potential noise on output measurements is ignored, i.e. the problem of the system states reconstructing is posed in a completely deterministic manner. Theoretically, as long as the system model is chosen to be sufficiently accurate, the state vector and the output can be precisely predicted. However, in practice this is not the case due to uncertainty in measurements (i.e. measurement noise) as well as a presence of unknown inputs that enter the system, errors due to discretisation, etc. (i.e. process noise). Consequently, in order to cope with practical problems a stochastic framework has to be employed and the state estimation needs to be considered by making use of statistical tools. These considerations lead to the KF which is an optimal filter that minimises the covariance matrix of the state estimation errors.

7.3.2 Derivation of KF

Consider the following modified state-space description of the actual process

$$x_{k+1} = Ax_k + Bu_k + \Gamma v_k, \quad x_0 = \bar{x}_0, \tag{7.20}$$

$$y_k = Cx_k + e_k, \tag{7.21}$$

where $\bar{x}_0$ is the expected value of the initial state vector. Vector $v_k \in \mathbb{R}^n$ is a set zero-mean stationary noise sequences with covariance matrix $\Sigma_v \in \mathbb{R}^{n\times n}$, e_k is a zero-mean stationary noise signal of variance σ_e^2 and $\xi_{ve} \in \mathbb{R}^n$ is the covariance vector between v_k and e_k.

$$E\left[\begin{bmatrix} v_k \\ e_k \end{bmatrix}\begin{bmatrix} v_j & e_j \end{bmatrix}\right] = \begin{bmatrix} \Sigma_v & \xi_{ve} \\ \xi_{ve}^T & \sigma_e^2 \end{bmatrix}, \tag{7.22}$$

$$E[v_k] = 0 \qquad \text{and} \qquad E[e_k] = 0. \tag{7.23}$$

Usually, it is also assumed that both v_k and e_k are white and mutually uncorrelated, which means that expression (7.22) simplifies to

$$E\left[\begin{bmatrix} v_k \\ e_k \end{bmatrix}\begin{bmatrix} v_j & e_j \end{bmatrix}\right] = \begin{bmatrix} \Sigma_v\delta_{kj} & 0 \\ 0 & \sigma_e^2 \end{bmatrix}. \tag{7.24}$$

Additionally, it is postulated that both v_k and e_k are uncorrelated with the input u_k.

The inclusion of noise sequences in the system state description means that the internal system behaviour can be considered to be uncertain with all the non-deterministic components collectively represented by a random vector v_k. This allows the introduction of some degree of flexibility into the model. Note that the vector v_k enters states directly and it is distributed via the matrix $\Gamma \in \mathbb{R}^{n\times n}$. The output is considered to be also uncertain, which is accounted by the addition of a random scalar e_k. Because v_k is added to states of a process/system it is typically referred to as a process noise. Similarly, because e_k is added to the measured system output it is usually referred to as a measurement noise.

Discarding the input, the evolution of the state vector is given by

$$x_{k+1} = Ax_k + \Gamma v_k, \tag{7.25}$$

which in the case of $A = \Gamma = I$ is called a multivariable random walk; when $\Gamma = I$ it corresponds to a Markov process. If $\Gamma = I$ and additionally v_k is Gaussian this leads to a multivariable Gauss-Markov process.

Assuming that one has u_k, y_k as well as the model of the process, the task consists of reconstructing optimally the system state vector x_k, such that the undesired effects of process and measurement noise are both minimised. Consider the estimated state vector without a delay that was developed in the case of the Luenberger observer, i.e. equation (7.14). The estimation error $\tilde{x}_k$ in the case of the system (7.20)-(7.21) is given by

$$\tilde{x}_{k|k} = Ax_{k-1} + Bu_{k-1} + \Gamma v_{k-1} - (I - LC)(A\hat{x}_{k-1|k-1} + Bu_{k-1}) - Ly_k. \tag{7.26}$$

The optimally condition considered is expressed in terms of the minimisation of the estimation error $\tilde{x}_k = \tilde{x}_{k|k}$. This is achieved by forming the following cost function

$$V_k = E[\tilde{x}_k \tilde{x}_k^T] = P_k, \tag{7.27}$$

and by choosing L appropriately so that the resulting observer is a minimum variance estimator of the actual state vector. The following derivation is based on that presented in (Dutton et al. 1997)

Equation (7.26) can be transformed by the inclusion of (7.21) to

$$\begin{aligned}\tilde{x}_k &= Ax_{k-1} + Bu_{k-1} + \Gamma v_{k-1} - (I - LC)(A\hat{x}_{k-1|k-1} + Bu_{k-1}) \\ &- L(Cx_k + e_k) \\ &= Ax_{k-1} + Bu_{k-1} + \Gamma v_{k-1} - (I - LC)(A\hat{x}_{k-1|k-1} + Bu_{k-1}) \\ &- L\left[C(Ax_{k-1} + Bu_{k-1} + \Gamma v_{k-1}) + e_k\right],\end{aligned} \tag{7.28}$$

which can be simplified yielding

$$\begin{aligned}\tilde{x}_k &= A\tilde{x}_{k-1} - LCA\tilde{x}_{k-1} + \Gamma v_{k-1} - LC\Gamma v_{k-1} - Le_k \\ &= FA\tilde{x}_{k-1} + F\Gamma v_{k-1} - Le_k\end{aligned} \tag{7.29}$$

with $F = I - LC$. Recall that the task consists of selecting the gain vector L such that (7.27) is minimised. The product $\tilde{x}_k \tilde{x}_k^T$ can now be written as

$$\begin{aligned}\tilde{x}_k \tilde{x}_k^T &= \left(FA\tilde{x}_{k-1} + F\Gamma v_{k-1} - Le_k\right)\left(FA\tilde{x}_{k-1} + F\Gamma v_{k-1} - Le_k\right)^T \\ &= FA\tilde{x}_{k-1}\tilde{x}_{k-1}^T A^T F^T + FA\tilde{x}_{k-1} v_{k-1}^T \Gamma^T F^T - FA\tilde{x}_{k-1} e_k L^T \\ &+ F\Gamma v_{k-1}\tilde{x}_{k-1}^T A^T F^T + F\Gamma v_{k-1} v_{k-1}^T \Gamma^T F^T - F\Gamma v_{k-1} e_k L^T \\ &- Le_k \tilde{x}_{k-1}^T A^T F^T - Le_k v_{k-1}^T \Gamma^T F^T + Le_k e_k L^T.\end{aligned} \tag{7.30}$$

By taking the expectation of (7.30) one obtains the KF cost function (7.27), i.e.

$$\begin{aligned} P_k &= E[FA\tilde{x}_{k-1}\tilde{x}_{k-1}^T A^T F^T] + E[FA\tilde{x}_{k-1}v_{k-1}^T \Gamma^T F^T] - E[FA\tilde{x}_{k-1}e_k L^T] \\ &+ E[F\Gamma v_{k-1}\tilde{x}_{k-1}^T A^T F^T] + E[F\Gamma v_{k-1}v_{k-1}^T \Gamma^T F^T] - E[F\Gamma v_{k-1}e_k L^T] \\ &- E[Le_k\tilde{x}_{k-1}^T A^T F^T] - E[Le_k v_{k-1}^T \Gamma^T F^T] + E[Le_k e_k L^T]. \end{aligned} \tag{7.31}$$

Subsequently, by utilising the assumptions regarding the uncorrelated nature of the noise sequences v_k and e_k, cf. (7.24), equation (7.31) is greatly simplified to

$$\begin{aligned} P_k &= FAE[\tilde{x}_{k-1}\tilde{x}_{k-1}^T]A^T F^T + F\Gamma E[v_{k-1}v_{k-1}^T]\Gamma^T F^T + LE[e_k e_k]L^T \\ &= FAP_{k-1}A^T F^T + F\Gamma\Sigma_v\Gamma^T F^T + \sigma_e^2 LL^T \\ &= F(AP_{k-1}A^T + \Gamma\Sigma_v\Gamma^T)F^T + \sigma_e^2 LL^T. \end{aligned} \tag{7.32}$$

By introducing an auxiliary variable $\breve{P}_{k-1}$, i.e.

$$\breve{P}_{k-1} = AP_{k-1}A^T + \Gamma\Sigma_v\Gamma^T \tag{7.33}$$

equation (7.32) is expressed as follows

$$\begin{aligned} P_k &= LC\breve{P}_{k-1}C^T L^T + \sigma_e^2 LL^T - \breve{P}_{k-1}C^T L^T - LC\breve{P}_{k-1} + \breve{P}_{k-1} \\ &= L(C\breve{P}_{k-1}C^T + \sigma_e^2)L^T - \breve{P}_{k-1}C^T L^T - LC\breve{P}_{k-1} + \breve{P}_{k-1}. \end{aligned} \tag{7.34}$$

Equation (7.34) is the so-called Riccati equation and it is analogous to a quadratic equation in a scalar case. As a consequence, the value of L which minimises (7.34) is found by completing squares of terms containing L. It is desired to re-express (7.34) so that it will have the following structure

$$\begin{aligned} P_k &= (L-M)G(L-M)^T - MGM^T + \breve{P}_{k-1} \\ &= LGL^T - MGL^T - LGM^T + \breve{P}_{k-1}, \end{aligned} \tag{7.35}$$

where M and G are auxiliary variables of appropriate dimension yet to be determined. Note that the minimum of expression (7.35) is found simply by setting $L = M$. By comparing equations (7.35) and (7.34) the following equalities arise, i.e.

$$G = C\breve{P}_{k-1}C^T + \sigma_e^2, \tag{7.36}$$

$$\breve{P}_{k-1}C^T = MG, \tag{7.37}$$

$$C\breve{P}_{k-1} = GM^T. \tag{7.38}$$

It is noted that because G and $\breve{P}_{k-1}$ are both symmetric, the two last equations are transposes of each other and the condition that $\breve{P}_{k-1}C^T = MG$ can be used to find M by substituting (7.36), i.e.

$$M = \breve{P}_{k-1}C^T G^{-1} = \breve{P}_{k-1}C^T(C\breve{P}_{k-1}C^T + \sigma_e^2)^{-1}. \tag{7.39}$$

Consequently, the minimum of the covariance matrix P_k is obtained with a gain vector $L = M$, i.e. L is the optimal gain, which is called the Kalman gain. Because L varies with time, it is denoted with a discrete-time subscript k, i.e. L_k. Substituting (7.39) into (7.35) gives

$$P_k = -L_k G L_k^T + \breve{P}_{k-1} = -L_k(C\breve{P}_{k-1}C^T + \sigma_e^2)L_k^T + \breve{P}_{k-1}, \tag{7.40}$$

which by noting from (7.39) that

$$L_k(C\breve{P}_{k-1}C^T + \sigma_e^2) = \breve{P}_{k-1}C^T \tag{7.41}$$

is simplified to

$$P_k = -\breve{P}_{k-1}C^T L_k^T + \breve{P}_{k-1} = \breve{P}_{k-1}(I - C^T L_k^T) = (I - L_k C)\breve{P}_{k-1}. \tag{7.42}$$

The last equality in (7.42) follows from the fact that P_k is symmetric.

Collecting all equations the overall KF algorithm is given by the following recursion:

$$\begin{aligned} \breve{P}_{k-1} &= AP_{k-1}A^T + \Gamma\Sigma_v\Gamma^T, & (7.43)\\ L_k &= \breve{P}_{k-1}C^T(C\breve{P}_{k-1}C^T + \sigma_e^2)^{-1}, & (7.44)\\ \hat{x}_{k|k} &= (I - L_k C)(A\hat{x}_{k-1|k-1} + Bu_{k-1}) + L_k y_k, & (7.45)\\ P_k &= (I - L_k C)\breve{P}_{k-1}. & (7.46) \end{aligned}$$

The KF given by equations (7.43)-(7.46) can be expressed more conveniently by noting that the matrix $\breve{P}_{k-1}$ is the *a priori* error covariance matrix, i.e. it is the covariance matrix of $\tilde{x}_{k|k-1} = x_k - \hat{x}_{k|k-1}$, where $\hat{x}_{k|k-1}$ is the state estimate predicted based upon data up to $k-1$. Because $E[v_k] = 0$, the best prediction of x_k given data up to $k-1$ is obtained from the state equation by simply ignoring the contribution of the process noise, i.e.

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}. \tag{7.47}$$

Consequently

$$\begin{aligned} \tilde{x}_{k|k-1} &= x_k - \hat{x}_{k|k-1} = Ax_{k-1} + Bu_{k-1} + \Gamma v_{k-1} - A\hat{x}_{k-1|k-1} - Bu_{k-1}\\ &= A\tilde{x}_{k-1} + \Gamma v_{k-1}, & (7.48) \end{aligned}$$

which, due to the postulated lack of correlation between v_k and e_k, leads to

$$\begin{aligned} P_{k|k-1} = E[\tilde{x}_{k|k-1}\tilde{x}_{k|k-1}^T] &= AE[\tilde{x}_{k-1}\tilde{x}_{k-1}^T]A^T + AE[\tilde{x}_{k-1}v_{k-1}^T]\Gamma^T \\ &\quad + \Gamma E[v_{k-1}\tilde{x}_{k-1}^T]A^T + \Gamma E[v_{k-1}v_{k-1}^T]\Gamma^T \\ &= AP_{k-1|k-1}A + \Gamma\Sigma_v\Gamma^T \\ &= \breve{P}_{k-1}. \end{aligned} \tag{7.49}$$

Substituting $P_{k|k-1}$ for $\breve{P}_{k-1}$ and separating equations into a prediction step (using data only up to $k-1$) and a subsequent correction step (inclusion of data at k via y_k) the KF algorithm can be alternatively expressed in the following two step procedure:

Prediction step:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}, \tag{7.50}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + \Gamma\Sigma_v\Gamma^T. \tag{7.51}$$

Correction step:

$$L_k = P_{k|k-1}C^T(CP_{k|k-1}C^T + \sigma_e^2)^{-1}, \tag{7.52}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k(y_k - C\hat{x}_{k|k-1}), \tag{7.53}$$

$$P_{k|k} = (I - L_kC)P_{k|k-1}. \tag{7.54}$$

Note the agreement with the formula of a general recursive method defined by equation (6.1) in Section 6.1. The *a priori* one-step ahead predicted output yielded by the KF (in fact, present in equation (7.53)) is given by

$$\hat{y}_{k|k-1} = C\hat{x}_{k|k-1}. \tag{7.55}$$

The *a posteriori* estimate of the system output is obtained from

$$\hat{y}_{k|k} = C\hat{x}_{k|k}, \tag{7.56}$$

i.e. after $\hat{x}_{k|k}$ in equation (7.53) has been calculated.

7.4 KF tuned for parameter estimation

The structure of the KF is similar to that of the RLS algorithm given by equations (6.23)-(6.25) in Section 6.2. In fact, the RLS can be interpreted as a special case of the KF, where the parameters to be estimated are treated as system states, i.e. $x_k = \theta_k$. The KF tuned for parameter estimation, denoted KFPE, is obtained by considering the regression equation written in a state-space form, i.e.

$$\theta_k = \theta_{k-1}, \tag{7.57}$$

$$y_k = \varphi_k^T \theta_k, \tag{7.58}$$

where

$$A = I, \tag{7.59}$$

$$B = 0, \tag{7.60}$$

$$C = \varphi_k^T, \tag{7.61}$$

$$\Gamma = 0. \tag{7.62}$$

The KF tuned for parameter estimation is obtained by modifying equations (7.50)-(7.54) as follows:

Prediction step:

$$\hat{\theta}_{k|k-1} = \hat{\theta}_{k-1|k-1}, \tag{7.63}$$

$$P_{k|k-1} = P_{k-1|k-1}. \tag{7.64}$$

Correction step:

$$L_k = P_{k|k-1}\varphi_k(\varphi_k^T P_{k|k-1}\varphi_k + \sigma_e^2)^{-1}, \tag{7.65}$$

$$\hat{\theta}_{k|k} = \hat{\theta}_{k|k-1} + L_k(y_k - \varphi_k^T \hat{\theta}_{k|k-1}), \tag{7.66}$$

$$P_{k|k} = (I - L_k\varphi_k^T)P_{k|k-1}. \tag{7.67}$$

By comparing the above KFPE algorithm with the RLS, cf. (6.23)-(6.25) in Section 6.2, it is observed that the two approach are, in fact, identical. Furthermore, if the estimated parameters, as for states, are treated as potentially time-varying according to a random walk:

$$\theta_k = \theta_{k-1} + v_k \tag{7.68}$$

with $\Gamma = I$, the second equation in the prediction step, i.e. (7.64), will change to

$$P_{k|k-1} = P_{k-1|k-1} + \Sigma_v. \tag{7.69}$$

7.5 Stationary KF

If the system is time-invariant, i.e. matrices A, B and C are constant, the noise sequences v_k and e_k are stationary and the pair (C, A) is detectable[1] then, it can be shown, that the covariance matrix $P_{k-1|k-1}$ tends to a constant value as $k \to \infty$, see (Walter & Pronzato 1997). Introduce

$$P = \lim_{k\to\infty} P_{k-1|k-1}, \tag{7.70}$$

$$P^+ = \lim_{k\to\infty} P_{k|k-1}. \tag{7.71}$$

Consequently, L_k also tends to a constant value L, see (7.52), i.e.

$$L = \lim_{k\to\infty} L_k = P^+C^T(CP^+C^T + \sigma_e^2)^{-1}. \tag{7.72}$$

The matrix P^+ can be obtained by recalling equation (7.51)

$$P^+ = APA^T + \Gamma\Sigma_v\Gamma^T, \tag{7.73}$$

which combined with equations (7.52) and (7.54) yields

$$P = P^+ - P^+C^T(CP^+C^T + \sigma_e^2)^{-1}CP^+. \tag{7.74}$$

Then, by, first, pre-multiplying (7.74) by A and, second, by post-multiplying by A^T one obtains the Riccati equation, i.e.

$$P^+ = AP^+A^T - AP^+C^T(CP^+C^T + \sigma_e^2)^{-1}CP^+A^T + \Gamma\Sigma_v\Gamma^T, \tag{7.75}$$

where equation (7.73) was also used. Finally, the matrix P^+ can be calculated either by i) determining the positive definite solution of the Riccati equation (7.75) or by ii) iterating the evolution of the covariance matrix P^+ until it becomes constant (Walter & Pronzato 1997).

To summarise, the stationary (or steady-state) KF, denoted SKF, is given by

$$\hat{x}_{k+1|k+1} = A\hat{x}_{k|k} + Bu_k + L(y_k - \hat{y}_k), \tag{7.76}$$

$$\hat{y}_k = C\hat{x}_{k|k}, \tag{7.77}$$

where L is obtained from expression (7.72).

[1] The system is detectable if all of its unobservable modes are stable. This is a slightly weaker assumption than the observability.

7.6 Innovations form and directly parametrisable representation

An alternative form for the state-space equations of the SKF is the so-called innovations representation, see (Ljung 1999). Consider equations of the SKF, i.e. (7.76)-(7.77), where the prediction error is given by

$$y_k - C\hat{x}_{k|k} = C(x_k - \hat{x}_{k|k}) + e_k. \tag{7.78}$$

The prediction error can be interpreted as the part of the system output, that cannot be predicted from past data (i.e. past outputs and present and past inputs), hence the name - the innovation sequence. By denoting the innovation sequence by ϵ_k, i.e.

$$\epsilon_k = y_k - C\hat{x}_{k|k}, \tag{7.79}$$

the equations of the SKF can be expressed as

$$\hat{x}_{k+1|k+1} = A\hat{x}_{k|k} + Bu_k + L\epsilon_k, \tag{7.80}$$

$$y_k = C\hat{x}_{k|k} + \epsilon_k. \tag{7.81}$$

Note that the innovations sequence appears explicitly in both equations. Moreover, by utilising the properties of the forward shift operator q, expressions (7.80)-(7.81) can be re-written in transfer function form, see (Ljung 1999), as follows

$$y_k = G(q)u_k + H(q)\epsilon_k, \tag{7.82}$$

where

$$G(q) = C\left(qI - A\right)^{-1} B, \tag{7.83}$$

$$H(q) = C\left(qI - A\right)^{-1} L + 1. \tag{7.84}$$

In Section 7.5, it has been shown that the Kalman gain L can be calculated in two ways, both, however, involving rather complicated expressions. Alternatively, use of the innovations representation can be made by parametrising the Kalman gain in terms of the model parameter vector θ. This simplifies the task of determining L greatly and leads to a so-called directly parametrised innovations form. As an example, see (Ljung 1999), consider the following system in an observer canonical form, where for simplicity it is assumed that $n_a = n_b = n_c = n$, i.e.

$$A = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_n & 0 & 0 & \dots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}. \tag{7.85}$$

The Kalman gain L is defined by

$$L = \begin{bmatrix} l_1 & l_2 & \dots & l_n \end{bmatrix}^T. \tag{7.86}$$

By utilising expressions (7.83)-(7.84) it can be deduced that the parameters contained in L are given by

$$l_i = c_i - a_i, \tag{7.87}$$

where c_i are the coefficients of the $C(q^{-1})$ polynomial of an ARMAX system. Consequently, the vector

$$L = \begin{bmatrix} c_1 - a_1 & c_2 - a_2 & \dots & c_n - a_n \end{bmatrix}^T \tag{7.88}$$

is the gain of the SKF for the ARMAX model structure in the case of a directly parametrised innovations form.

7.7 Extended KF

The KF relies on the assumption that the underlying set of equations describing the system is linear. However, in most practical applications the system (hence its model) is nonlinear, which makes the direct application of the KF infeasible. To cope with this problem an extension of the KF for nonlinear systems, called the EKF, has been developed. The fundamental idea of the EKF is to linearise the nonlinear set of equations at instantaneous operating point providing a linearised model of the nonlinear system. Therefore, the EKF can be considered a tool for solving a nonlinear set of inconsistent stochastic equations.

The following state-space description of a nonlinear system is considered

$$z_{k+1} = \mathcal{F}(z_k, u_k) + \bar{\Gamma}\bar{v}_k, \quad z_0 = \bar{z}_0, \tag{7.89}$$

$$y_k = \mathcal{H}(z_k) + e_k, \tag{7.90}$$

where z_k denotes the extended state vector and $\bar{z}_0$ its expected initial value. Assumptions regarding the noise sequences $\bar{v}_k$ and e_k are the same as those for the KF case, see equations (7.22)-(7.23) in Subsection 7.3.2. The nonlinear functions describing transition of states and the output are denoted by $\mathcal{F}(\cdot)$ and $\mathcal{H}(\cdot)$, respectively, and it is postulated that both are differentiable. Note, that since $\mathcal{F}(\cdot)$ and $\mathcal{H}(\cdot)$ are nonlinear, in general, the KF can not be used directly. However, both functions can be replaced by their corresponding first order Taylor series expansions around current working point. This yields a linearised version of the original system for which the KF can be applied. A first order approximation of $\mathcal{F}(\cdot)$ around $\hat{z}_{k-1|k-1}$ is given by

$$\mathcal{F}(z_{k-1}, u_{k-1}) \approx \mathcal{F}(\hat{z}_{k-1|k-1}, u_{k-1}) + \mathcal{F}^*_{k-1}(z_{k-1} - \hat{z}_{k-1|k-1}), \tag{7.91}$$

where the Jacobian matrix is defined by

$$\mathcal{F}^*_{k-1} = \frac{\partial \mathcal{F}(z_{k-1}, u_{k-1})}{\partial z_{k-1}}\Big|_{\hat{z}_{k-1|k-1}}. \tag{7.92}$$

Similarly, a first order approximation of $\mathcal{H}(\cdot)$ around $\hat{z}_{k|k-1}$ is given by

$$\mathcal{H}(z_k) \approx \mathcal{H}(\hat{z}_{k|k-1}) + \mathcal{H}^*_k(z_k - \hat{z}_{k|k-1}), \tag{7.93}$$

where the Jacobian matrix is defined by

$$\mathcal{H}^*_k = \frac{\partial \mathcal{H}(z_k)}{\partial z_k}\Big|_{\hat{z}_{k|k-1}}. \tag{7.94}$$

Consequently, the approximate linearised state-space system is given by

$$z_{k+1} = \mathcal{F}^*_k z_k + \mathcal{F}(\hat{z}_{k|k}, u_k) - \mathcal{F}^*_k \hat{z}_{k|k} + \bar{\Gamma}\bar{v}_k, \quad z_0 = \bar{z}_0, \tag{7.95}$$

$$y_k = \mathcal{H}^*_k z_k + \mathcal{H}(\hat{z}_{k|k-1}) - \mathcal{H}^*_k \hat{z}_{k|k-1} + e_k. \tag{7.96}$$

Since equations (7.95) and (7.96) are now linear in z_k the KF can be used. The overall EKF algorithm is summarised as follows:

Prediction step:

$$\hat{z}_{k|k-1} = \mathcal{F}(\hat{z}_{k-1|k-1}, u_{k-1}), \tag{7.97}$$

$$P_{k|k-1} = \mathcal{F}^*_{k-1} P_{k-1|k-1} \mathcal{F}^{*T}_{k-1} + \bar{\Gamma}\Sigma_v \bar{\Gamma}^T. \tag{7.98}$$

Correction step:

$$L_k = P_{k|k-1}\mathcal{H}_k^{*T}(\mathcal{H}_k^{*}P_{k|k-1}\mathcal{H}_k^{*T} + \sigma_e^2)^{-1}, \tag{7.99}$$

$$\hat{z}_{k|k} = \hat{z}_{k|k-1} + L_k[y_k - \mathcal{H}(\hat{z}_{k|k-1})], \tag{7.100}$$

$$P_{k|k} = (I - L_k\mathcal{H}_k^{*})P_{k|k-1}, \tag{7.101}$$

where the Jacobians of $\mathcal{F}(z_{k-1}, u_{k-1})$ and $\mathcal{H}(z_k)$ are given by equations (7.92) and (7.94), respectively.

Note that the structure of the EKF is very similar to that of the KF with the crucial differences that nonlinear functions describing state transition and system output are replaced by their corresponding Jacobian matrices calculated at a current operating point in equations (7.98), (7.99) and (7.101). In equation (7.97) the original nonlinear state equation $\mathcal{F}(\cdot)$ is utilised to calculate the *a priori* prediction of the extended state vector and in equation (7.100) the nonlinear output equation $\mathcal{H}(\cdot)$ is used to compute the one-step ahead prediction of the system output.

Unlike the KF, the EKF is no longer an optimal filter and in general it is considerably more difficult to tune. The EKF can also diverge if the initial values are not selected with care. Moreover, since the EKF relies on the linearisation of the nonlinear system, it only represents the actual system reliably within a small locality around the linearisation point. Therefore, if the sampling is too slow or/and the process dynamics is fast, the validity of the linearisation is restricted and the EKF can exhibit problems with tracking.

7.8 Extended KF for joint parameter and state estimation

The EKF introduced in the previous section can be used to estimate the state and parameter vectors of a linear system in a joint fashion. The necessity for using the EKF and not simply the KF is that if both, i.e. x_k and θ_k, are treated as unknown, the overall estimation problem becomes nonlinear with respect to the new extended parameter vector regardless of the fact that the underlying system is, in fact, linear. This is because of the products of the state transition matrix $A(\theta_k)$ and the output vector $C(\theta_k)$, which both are constructed from the elements of θ_k, with the state vector x_k, i.e.

$$x_{k+1} = A(\theta_k)x_k + B(\theta_k)u_k + \Gamma v_k, \quad x_0 = \bar{x}_0, \tag{7.102}$$

$$y_k = C(\theta_k)x_k + e_k. \tag{7.103}$$

The extended parameter vector z_k is in this case given by

$$z_k = \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} \in \mathbb{R}^{n+n_\theta}. \tag{7.104}$$

The state-space description (7.102)-(7.103) can be added to with an additional equation describing the (random walk) evolution of the parameter vector, i.e.

$$\theta_{k+1} = \theta_k + q_k \tag{7.105}$$

where $q_k \in \mathbb{R}^{n_\theta}$ is a vector consisting of serially uncorrelated random variables of zero mean, i.e. $E[q_k] = 0$ and $E[q_k q_j] = \Sigma_q \delta_{kj}$. This leads to the extended state-space form, i.e.

$$z_{k+1} = \mathcal{F}(z_k, u_k) + \bar{\Gamma}\bar{v}_k, \tag{7.106}$$

$$y_k = \mathcal{H}(z_k) + e_k, \tag{7.107}$$

where

$$\mathcal{F}(z_k, u_k) = \begin{bmatrix} A(\theta_k)x_k + B(\theta_k)u_k \\ \theta_k \end{bmatrix}, \qquad \bar{\Gamma} = \begin{bmatrix} \Gamma & 0 \\ 0 & I \end{bmatrix}, \tag{7.108}$$

$$\mathcal{H}(z_k) = C(\theta_k)x_k, \qquad \bar{v}_k = \begin{bmatrix} v_k \\ q_k \end{bmatrix}. \tag{7.109}$$

This satisfies the general nonlinear state-space description defined by expressions (7.89)-(7.90). Consequently, the EKF tuned for joint state and parameter estimation uses the model where $\hat{z}_{k|k}$ is substituted for z_k in the parametrisation of functions $\mathcal{F}(z_k, u_k)$ and $\mathcal{H}(z_k)$. The corresponding Jacobian matrices are calculated as follows

$$\mathcal{F}^*_{k-1} = \begin{bmatrix} A(\hat{\theta}_{k-1|k-1}) & R_1 \\ 0 & I \end{bmatrix}, \tag{7.110}$$

$$\mathcal{H}^*_k = \begin{bmatrix} C(\hat{\theta}_{k|k-1}) & R_2 \end{bmatrix} \tag{7.111}$$

and

$$R_1 = \frac{\partial \left[A(\theta_k)x_k + B(\theta_k)u_k\right]}{\partial \theta_k}\Big|_{\hat{z}_{k-1|k-1}}, \tag{7.112}$$

$$R_2 = \frac{\partial \left[C(\theta_k)x_k\right]}{\partial \theta_k}\Big|_{\hat{z}_{k|k-1}}. \tag{7.113}$$

The initial values for the estimation can be chosen as

$$\hat{z}_{0|0} = \begin{bmatrix} 0 \\ \hat{\theta}_0 \end{bmatrix}, \qquad P_{0|0} = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}, \tag{7.114}$$

where P_1 and P_2 reflects the prior confidence in the initial values set for the state and parameter vector, respectively.

As alternative to the EKF for joint state and parameter estimation, denoted JEKF, it is possible to use a parameter estimation method cross-coupled with a state estimation algorithm. For example, the following combinations can be considered:

- RLS & SKF
- RLS & KF
- KFPE & SKF
- KFPE & KF

The cross-coupled or tandem methods of parameter and state estimation are typically easier to tune compared to the JEKF. However, the JEKF provides a more elegant solution, because both state and parameter vector are estimated together by a single algorithm.

Questions

- Explain the notion of an observer.
- Discuss the practical motivations for using observers.
- Explain the concept of an identity observer. Discuss disadvantages of such an approach.
- Explain the idea of incorporating feedback into an observer. Discuss advantages and disadvantages of such an approach.
- Explain what is meant by the observability property.
- Discuss the optimality properties of the KF.
- Explain how to obtain the KFPE from the KF algorithm.
- Discuss the idea of the SKF.
- Comment on the link between the Riccati equation and the KF, SKF and a directly parametrisable innovation representation.

- Explain the advantages and disadvantages of using the KF, SKF and a directly parametrisable innovation representation.
- Explain the motivation for using the EKF. Discuss the drawbacks and advantages.
- Explain how the EKF can be used for joint state and parameter estimation. Why is there a need to use the EKF and not the KF?
- Are there any other alternatives to the EKF for joint state and parameter estimation?

Chapter 8

Summary and concluding remarks

This text aimed at providing a sound introductory basis to the subject of system identification and filtering. In Chapter 1 the concept of system identification was divided into several stages. The various stages have been discussed and diagrammatically depicted in terms of an iterative procedure.

Subsequently, in Chapter 2, three modelling methodologies, i.e. white-box, grey-box and black-box have been introduced and their corresponding advantages and disadvantages have been discussed. A treatment of disturbances on measured signals has been addressed and a distinction between the classical, i.e. control, framework and the EIV framework has been made.

In Chapter 3 several important properties of system models supported by examples have been introduced and discussed. Distinctions have been made between: linear and nonlinear, dynamic and static, LTI and LTV, continuous-time and discrete-time models. Different representations of linear models for both discrete-time and continuous-time have been introduced and the relationships between them have been highlighted. Different linear model structures have been analysed and the concept of a one-step ahead optimal predictor has been introduced. The chapter ends with a description of frequently adopted nonlinear system structures such as Wiener and Hammerstein models, bilinear system models and the NARX class of models.

Chapter 4 has addressed the estimation problem of low order, i.e. first and second order, continuous-time models directly from step response tests. In the case of second order models the notion of underdamped and overdamped responses have been considered.

In Chapter 5 the method of LS has been introduced with an aid of a simple example. Subsequently, it has been demonstrated for the general case of linear static models. Properties of LS estimates such as bias and consistency have been analysed thoroughly and a geometrical interpretation of the LS method has been explained. The chapter ends with a discussion on the application of the LS estimation method to linear dynamic models.

Chapter 6 provides an introduction to recursive estimation algorithms where the emphasis has been placed on the RLS technique. First, a general diagrammatic structure of a recursive method has been described, which is followed by a derivation of the RLS algorithm. Issues involved with initialisation of RLS have been discussed and a distinction between one-step ahead prediction and system simulation has been explained. Subsequently, modifications of the standard RLS algorithm for the purpose of coping with LTV models as well as for the purpose of covariance matrix management have been introduced. Also the notion of estimator memory has been considered. Furthermore, a modification of RLS incorporating an inherent mechanism for tracking time-varying parameters leading to the KFPE has been described.

Chapter 7 has dealt with the problem of state estimation and filtering. The concept of a state observer has been introduced. The Luenberger observer and the identity observer have then been described together with the notion of system observability. This has formed the basis for a derivation of the KF algorithm. Subsequently, the KF has been configured and tuned for parameter estimation. SKF and a directly parametrisable innovation representation have been discussed. Finally the chapter ends with an introduction to the EKF, which has been demonstrated to be applicable for joint state and parameter estimation.

Bibliography

Åström, K. J. & Wittenmark, B. (1997), *Computer-Controlled Systems: Theory and Design*, 3 edn, Prentice-Hall Inc., USA.

Balmer, L. (1986), Short term spectral estimation and applications, PhD thesis, Warwick University, Coventry, UK.

Dutton, K., Thompson, S. & Barraclough, B. (1997), *The Art of Control Engineering*, Addison Wesley Longman, Harlow, England.

Fortescue, T. R., Kershenbaum, L. S. & Ydstie, B. E. (1981), 'Implementation of self-tuning regulators with variable forgetting factors', *Automatica* **17**(6), 831–835.

Hagglund, T. (1993), New estimation techniques for adaptive control, PhD thesis, Lund University, Lund, Sweden.

Hsia, T. (1977), *System Identification: Least Squares Methods*, Lexington Books, Toronto, Canada.

Ikonen, E. & Najim, K. (2002), *Advanced Process Identification and Control*, Marcel Dekker, Inc., USA.

James, G. (1998), Parameter estimation and control algorithms for self-tuning control, Lecture notes, Coventry University, Coventry, UK.

Lindskog, P. & Ljung, L. (1993), Tools for semi-physical modeling, *in* 'Proc. 10th IFAC Symp. on System Identification', Copenhagen, Denmark, pp. 237–242.

Ljung, L. (1999), *System Identification - Theory for the User*, 2nd edn, Prentice Hall PTR, New Jersey, USA.

Ljung, L. (2008), Perspectives on system identification, *in* ‘Proc. of 17th IFAC World Congress’, Seoul, Korea, pp. 7172–7184.

Ljung, L. & Söderström, T. (1983), *Theory and practice of recursive identification*, MIT Press, Cambridge, UK.

Mańczak, K. & Nahorski, Z. (1983), *Komputerowa identyfikacja obiektów dynamicznych (Computer based identification of dynamical models)*, Państwowe Wydawnictwo Naukowe, Warsaw, Poland.

Nise, N. (2008), *Control Systems Engineering*, 5 edn, John Wiley & Sons.

Pearson, R. K. (1999), *Discrete-time dynamic models*, Oxford University Press, New York, USA.

Pearson, R. K. & Pottmann, M. (2000), ‘Gray-box identification of block-oriented nonlinear models’, *J. of Process Control* **10**, 301–315.

Söderström, T. (2007), ‘Errors-in-variables methods in system identification’, *Automatica* **43**(6), 939–958.

Söderström, T. & Stoica, P. (1989), *System Identification*, Prentice Hall International, Hemel Hempstead, UK.

Sorenson, H. W. (1970), ‘Least squares estimation: from Gauss to Kalman’, *IEEE Spectrum* **7**, 63–68.

Walter, E. & Pronzato, L. (1997), *Identification of parametric models from experimental data*, Communications and Control Engineering Series, Springer, Londres.

Wellstead, P. E. & Zarrop, M. B. (1991), *Self-Tuning Systems: Control and Signal Processing*, John Wiley & Sons, England.

Young, P. (1984), *Recursive Estimation and Time-Series Analysis*, Springer-Verlag, Berlin, Germany.

Young, P. C. (1974), ‘Recursive approaches to time series analysis’, *Bull. of Inst. Maths and its Applications* **10**, 209–224.