EPFL

Laboratoire
d'Automatique

SYSTEM IDENTIFICATION

Prof. Alireza Karimi

Lausanne
Spring 2023

PREFACE

This course note is designed for graduate students enrolled in the System Identification course at École Polytechnique Fédérale de Lausanne (EPFL). It provides a comprehensive overview of the theoretical and practical aspects of dynamical system identification. This English version is based on a French course note, "Identification de Systèmes Dynamiques," written by Prof. Dominique Bonvin and Alireza Karimi. The new version includes updated information on stochastic aspects of system identification and parameter estimation in noisy environments.

Chapter 3 has been expanded to include an introduction to the state-space model and nonlinear identification, and the material in other chapters has been reorganized. The additional content is primarily sourced from two textbooks: "System Identification; Theory for the User" by Lennart Ljung (1998, Prentice Hall) and "System Identification" by Söderström and Stoica (2001, Prentice Hall).

The course note consists of three chapters. The first chapter outlines the fundamental concepts of models, systems, plants, and system identification, as well as input signal properties and least squares algorithms. Chapter 2 covers nonparametric methods, while Chapter 3 delves into parametric identification methods and their practical applications.

The course also features hands-on computer exercises in MATLAB to help students understand the theoretical concepts and apply identification algorithms to real systems using the MATLAB Identification Toolbox. The exercises use data collected from an experimental setup, offering students the opportunity to practice what they have learned.

Lausanne, February 2023

Contents

Chapter 1

Fitting a Model to the Data

1.1 Introduction

System identification is the field of constructing mathematical models for dynamical systems based on measured data. It draws its theoretical foundations from estimation theory, signal processing, and numerical optimization.

This chapter provides a review of the basic definitions for plants, systems, and models, as well as their classifications and representation. Since system identification relies on experimental data, the characteristics of input excitation signals are also studied. Finally, the least squares algorithm, which is widely used in system identification, is explained and its properties in the presence of noise are analyzed.

The concepts covered in this chapter form the basis for understanding the algorithms presented in Chapters 2 and 3. It defines system identification as the process of "fitting a model to the data" by introducing the least squares algorithm, classifying the mathematical models, and characterizing the input signals.

The chapter is organized as follows: Section 1.2 provides definitions for plants, systems, and models. The dynamic models are then classified and different types of representation models are discussed. Section 1.3 introduces various signals for excitation of dynamical systems and compares them based on their spectra. The least squares algorithm is explained and analyzed in a stochastic environment in the final section.

1.2 Plant, System and Model

Plant: The physical reality that we want to study is generally referred to the plant, for example, a stirred tank, an electrical motor or a car. These physical realities are usually very complex and their study needs some simplification.

System: A system is an abstraction of the plant that contains only the important elements for the study. The system is an object in which different sorts of variables

interact and produce observable signals. The system is usually driven by external *input* variables, while the useful information about the system are provided by the *output* variables. For example, consider a car as a plant whose trajectory should be studied. In this case, the position and the speed of the car that give the most useful information about the car trajectory are the output variables. While, the brake pedal, accelerator pedal and steering wheel positions are the driving signals or input variables of the system. The road slope and wind can be considered as some other input variables, called disturbance signals, that cannot be controlled. Figure 1.1 shows a schematic diagram of a system defined for studying the trajectory of a car. Note that in this study, we are not interested in the other variables in the plant

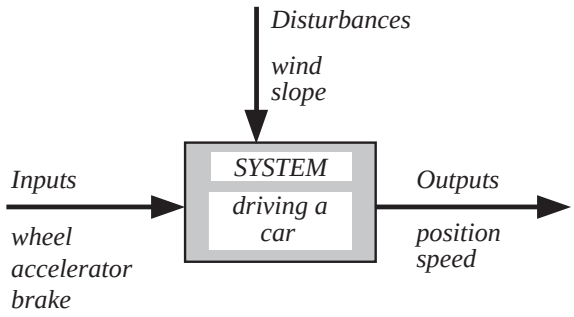


Figure 1.1: A system for studying the trajectory of a car with its inputs, outputs and disturbances

like the temperature of the cooling water, or the movements of the pistons that may be important for other studies.

Model: The relation between inputs and outputs of a system is given by a model. Many kinds of model may be defined for a system. For example a *verbal model* can be used to describe the system for driving a car ('turning the wheel causes the car to turn', 'pushing the brake decreases the speed', etc.). A typical example for a *graphical model* is the step response, i.e. the output of a system when excited by a step variation in its input. For the driving of a car, a step response can show how the speed of the car changes when the acceleration pedal is pushed. The frequency response of a linear system is another graphical model that is often used to describe linear systems. In a *mathematical model*, the relations between inputs and outputs are given by some algebraic and/or differential equations. Such models are very well suited for analysis, simulation, prediction and controller design.

1.2.1 Modeling versus identification

A model is very often constructed from the observed data. This is the case for verbal and graphical models. However, the mathematical models can be constructed by two different approaches:

Modeling: The system is divided to subsystems whose properties are fairly known and can be described by basic laws from physics (such as Newton's laws and balance equations). Then, by combining the mathematical equations of subsystems, a model for the whole system is obtained. This approach is usually called *first principle modeling*.

Identification: Some experiments are performed on the system and input and output signals are recorded. Then, a model structure is chosen and its parameters are computed such that the output of the model, when excited by the input signal, is close to the measured output signal (a model is fitted to the data).

Example 1.1. Consider that we are interested in studying the angular position of the axis of a DC motor with independent excitation when the motor voltage is controlled. Two approaches for building a mathematical model for this system are investigated.

In the first approach, we rely on our knowledge about the schematic diagram of a DC motor shown in Fig. 1.2. The system contains one input, the motor voltage, and

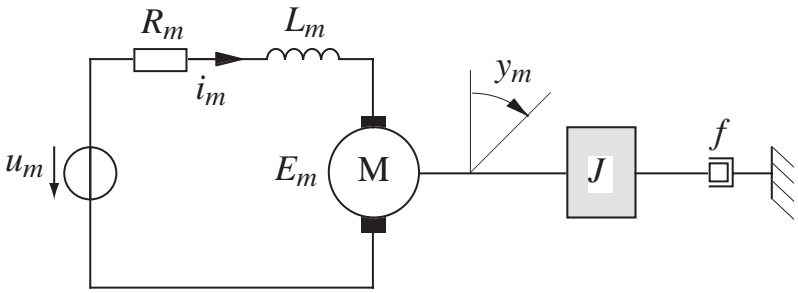


Figure 1.2: Schematic diagram of a DC motor

one output, the angular position of the motor shaft. The system can be divided into two subsystems: electrical and mechanical. The equation associated to the electrical circuit is given by:

$$u_m(t) = R_m i_m(t) + L_m \frac{di_m(t)}{dt} + E_m(t) \tag{1.1}$$

where $u_m(t)$ is the motor voltage, $i_m(t)$ the motor current, R_m and L_m are resistance and the inductance of the motor's coil, respectively. The back-electromotive-force, $E_m(t)$ is supposed to be proportional to the rotor speed, i.e., $E_m(t) = K_m \dot{y}_m(t)$, where $y_m(t)$ is the shaft angular position (system's output).

On the other hand, the electrical torque exerted by the motor, $T_m(t)$ acts on the mechanical structure following this equation:

$$T_m(t) = J\ddot{y}_m(t) + f\dot{y}_m(t) \tag{1.2}$$

where the inertia of the rotor and load is J and the viscous friction coefficient f. It is known that the torque is proportional to the motor current, i.e. $T_m(t) = K_m i_m(t)$.

The mechanical time constant of a motor is typically much larger than the electrical time constant, so the electrical dynamics can be ignored. Therefore, by ignoring the inductance of the motor, the following system of differential equations can be written for the system:

$$u_m(t) - R_m i_m(t) = K_m \dot{y}_m(t) \tag{1.3}$$
$$J\ddot{y}_m(t) + f\dot{y}_m(t) = K_m i_m(t) \tag{1.4}$$

By eliminating $i_m(t)$ from the above equations, the relation between the input, $u_m(t)$, and the output, $y_m(t)$ is given by:

$$JR_m\ddot{y}_m(t) + (fR_m + K_m^2)\dot{y}_m(t) = K_m u_m(t) \tag{1.5}$$

The Laplace transform can be used to obtain a transfer function associated to the differential equation in (1.5):

$$G_m(s) = \frac{Y_m(s)}{U_m(s)} = \frac{K_m}{JR_m s^2 + (fR_m + K_m^2)s} \tag{1.6}$$

It is clear that the parameters of the model, K_m, J, R_m and f should be perfectly known.

The second approach is to excite the motor voltage with some sort of input signal such as a step, a sinusoid or a random signal and measure the shaft angular position. Then, a model structure should be chosen. For this system, as we know that there is an integrator in the model, we choose the following structure:

$$G(s) = \frac{b}{s(as+1)} \tag{1.7}$$

By an identification procedure we can find the model parameters a and b, such that the output of $G(s)$ is as close as possible to the measured output. Note that, in this identification the model parameters are identified and not the physical parameters. In fact, by identifying a and b only two physical parameters can be computed (because we have only two equations). So, for this example, all the physical parameters are not *identifiable*. However, knowing K_m and R_m, the mechanical parameters J and f can be computed.

Comparison: System identification and physical modeling can be compared from different aspects. A physical model gives a direct relation between the physical parameters and model parameters. This type of models are very well suited for pedagogical purpose as well as system design and analysis. However, they have the following disadvantages:

- In many cases, the plant is too complex such that it is not possible to apply the first principles for obtaining a model.
- Even if a model is obtained based on the physical laws, its parameters are often unknown.

- The physical modeling leads usually to high-order models and it is not always clear that which dynamics can be neglected.

The models obtained by system identification are relatively simple to construct and use and give reasonably good results for prediction and controller design. However, in contrast to physical models:

- They have limited validity depending on the operating point and type of input. Therefore, the input design has a great importance in system identification.
- They need sensors to measure the outputs of the system, which is not always possible. Moreover, the measured data are noisy and this effect should be considered in identification.
- They fit a model to the data, so the identified model parameters, in most cases, have little physical meaning.
- They require a model structure which should be defined based on some prior knowledge on the physical system or iteratively using the measured data.

1.2.2 Model classification

In this course, we are interested in identification of *dynamical* systems. In these systems, the outputs are a function of the past inputs and outputs given by a set of differential equations, in contrast to *static* systems where the outputs are a function of the inputs at the same instant given by a set of algebraic equations (there is no memory). Mathematical models of dynamical systems can be classified as follows.

Monovariable-Multivariable models:

Models with a single input and a single output are called SISO models or mono variable models, while systems with more than one input and one output are called multivariable systems or MIMO systems (multi-input multi-output systems). The algorithms used for system identification in this course are usually illustrated for SISO systems, however, most of them can be applied straightforwardly to MIMO models.

Linear-Nonlinear models:

A model is linear if its output depends linearly to the inputs or in other words if it follows the superposition principle defined by two properties: additivity and homogeneity, otherwise it is nonlinear. Assume that the output of a system when excited by $u_1(t)$ and $u_2(t)$ is, respectively, $y_1(t)$ and $y_2(t)$. Then, this model is linear if and only if the output of the system for $\alpha_1 u_1(t)+\alpha_2 u_2(t)$ is equal to $\alpha_1 y_1(t)+\alpha_2 y_2(t)$ for any constant $\alpha_1, \alpha_2 \in \mathbb{R}$.

In this course, we primarily focus on linear models. However, some of the techniques can be applied to certain classes of nonlinear systems as well. It's important to note that while most physical systems exhibit some level of nonlinearity in their models, linear models provide highly effective predictions of their outputs. Linear models can also be utilized to design controllers that deliver good performance when implemented on real-world nonlinear systems. One way to obtain a linear model is by linearizing a nonlinear model around an operating point. In certain applications, a suitable selection of inputs and outputs, guided by the physics of the system, can result in a linear model.

Example 1.2. Consider the following nonlinear model, with the input $u(t)$ and the output $y(t)$, which operates around y_0:

$$\dot{y}(t) + 2y^2(t) = 3u(t-\tau) \qquad y(0) = y_0 \tag{1.8}$$

The nonlinearity of the model comes from the quadratic term $y^2(t)$ in the differential equation. Using the Taylor series, the nonlinear part can be linearized around the operating point y_0 as follows:

$$y^2(t) \approx y_0^2 + 2y_0(y(t) - y_0) \tag{1.9}$$

Then, let two new variables $\delta u(t) = u(t) - u_0$ and $\delta y(t) = y(t) - y_0$ be defined that are the variations of the input and output around the operating points u_0 and y_0, respectively. Therefore, the linearized model will be given by:

$$\dot{\delta y}(t) + 2y_0^2 + 4y_0\delta y(t) = 3\delta u(t-\tau) + 3u_0 \tag{1.10}$$

Note that at the steady state $u_0 = \frac{2}{3}y_0^2$, hence:

$$\dot{\delta y}(t) + 4y_0\delta y(t) = 3\delta u(t-\tau) \tag{1.11}$$

which gives the following transfer function:

$$G(s) = \frac{3e^{-\tau s}}{s + 4y_0} \tag{1.12}$$

that represents a linear model around the operating point y_0.

Example 1.3 (*Ljung 1998*)**.** Consider a plant consisting of a heater immersed in a cooling liquid. We measure $v(k)$, the voltage applied to the heater, $T(k)$, the temperature of the liquid and $y(k)$, the temperature of the heater coil surface. Suppose that a model is required to show the dependence of $y(k)$ to $T(k)$ and $v(k)$. Based on common sense and high school physics we know that:

- The change in temperature of the heater coil over one sample is proportional to the electrical power minus the heat loss to the liquid.
- The electrical power is proportional to $v^2(k)$.

- The heat loss is proportional to $y(k) - T(k)$.

This suggests the following model:

$$y(k) = y(k-1) + \alpha v^2(k-1) - \beta[y(k-1) - T(k-1)] \tag{1.13}$$

This model is nonlinear because of the quadratic term $v^2(k-1)$ in the difference equation. However, it can be easily linearized if we define a new variable $p(k) = v^2(k)$. So the linear model will have two inputs: $p(k)$ and $T(k)$ and one output: $y(k)$. The parameters of this model, α and β are unknown and should be identified using the measured data.

Time invariant - Time varying models:

In time-varying models, the model parameters change with time while in time-invariant models the parameters are constant. If the operating point, given by u_0 and y_0 in the linearized model (1.10) changes in time, the model becomes time varying, otherwise it is time invariant.

Deterministic-Stochastic models

The output of a deterministic model can be exactly computed based on the input signal, while the output of a stochastic model contains random terms that makes such an exact computation impossible. The random terms are usually described by a random disturbance at the output of the system. In system identification, since the measured data are always noisy, the stochastic models are preferred to describe the systems.

1.2.3 Representation models

The linear time-invariant models can be represented in different forms. These representations are discussed in the sequel.

Continuous time - Discrete time models:

The output of a continuous-time linear time-invariant (LTI) SISO system is usually defined by the convolution integral as follows:

$$y(t) = \int_0^\infty g(\tau)u(t-\tau)d\tau = g(t) * u(t) \tag{1.14}$$

where $g(t)$ is the response of the system to a Dirac impulse $\delta(t)$, i.e.:

$$y(t) = \int_0^\infty g(\tau)\delta(t-\tau)d\tau = g(t) * \delta(t) = g(t)$$

The impulse response, $g(t)$, is a nonparametric continuous-time model for the system. The Laplace transform of $g(t)$, called $G(s)$, is also a continuous-time model with a transfer function representation. By taking the Laplace transform of the both sides of (1.14), one obtains:

$$Y(s) = G(s)U(s) \Rightarrow G(s) = \frac{Y(s)}{U(s)} \tag{1.15}$$

A discrete-time model describes the relation between inputs and outputs at discrete time points with constant sampling time T such that $y(k) \equiv y(kT)$. The output of a discrete-time linear time-invariant SISO model can be represented by the following convolution sum:

$$y(k) = \sum_{j=0}^{\infty} g(j)u(k-j) = g(k) * u(k) \tag{1.16}$$

where $g(k)$ is the response of the system to a Kronecker delta defined by:

$$\delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \tag{1.17}$$

If the Z transform of $y(k), u(k)$ and $g(k)$ are called $Y(z), U(z)$ and $G(z)$, respectively, then a discrete-time transfer function can be obtained by

$$Y(z) = G(z)U(z) \Rightarrow G(z) = \frac{Y(z)}{U(z)} \tag{1.18}$$

It should be mentioned that there is no unique transformation of a continuous-time model to a discrete-time model and vice versa. Consider two different input signals $u_1(t)$ and $u_2(t)$ shown in Fig. 1.3 and their Laplace transform $U_1(s)$ and $U_2(s)$, respectively, applied to $G(s)$. Let the continuous output signals $y_1(t)$ and $y_2(t)$ be the response of the system $G(s)$ to $u_1(t)$ and $u_2(t)$. It is clear that $y_1(t) \neq y_2(t)$. On the other hand, the sampled input signals are equal, i.e. $u_1(k) = u_2(k)$ while the sampled outputs are not, i.e. $y_1(k) \neq y_2(k)$. As a result, $G_1(z) = Y_1(z)/U_1(z)$ and $G_2(z) = Y_2(z)/U_2(z)$ are two different discrete-time transfer functions associated to the same continuous-time model $G(s)$.

There are several methods to convert a continuous-time model to a discrete-time model and vice versa. In the following, the most used methods are briefly explained.

ZOH method: In this method, it is assumed that all input signals are applied to the continuous-time model after passing through a Zero-Order-Hold (ZOH) block. Therefore, the continuous-time system sees only the step signals at its input. As a result, the best way to convert the continuous-time model is to find an equivalent model for the input signals in step form. Thus, the sampled step response of $G(s)$ should be equal to the step response of $G(z)$:

$$y(t)|_{kT} = \mathcal{L}^{-1}\left[\frac{G(s)}{s}\right]_{kT} = \mathcal{Z}^{-1}\left[\frac{G(z)}{1-z^{-1}}\right]$$

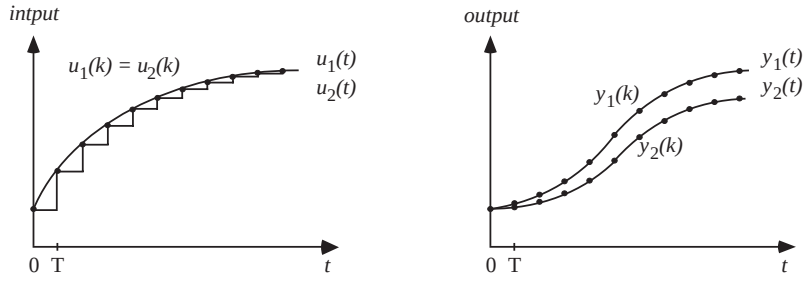


Figure 1.3: Inputs and outputs of a continuous-time model

From this equation $G(z)$ can be computed based on a given $G(s)$:

$$G(z) = \frac{z-1}{z}\mathcal{Z}\left\{\mathcal{L}^{-1}\left[\frac{G(s)}{s}\right]\right\}$$

In the same way $G(s)$ can be computed from $G(z)$:

$$G(s) = s\mathcal{L}\left\{\mathcal{Z}^{-1}\left[\frac{G(z)}{1-z^{-1}}\right]\right\}$$

Tustin's method: In this approach, the imaginary axis in the s-plane, i.e. $Re\{s\} = 0$, is mapped to the unit circle, i.e. $|z| = 1$ in the z-plane by a conformal mapping. This way, the left hand side of the s-plane, $Re\{s\} < 0$, is also mapped to the interior of the unit circle and so the stability will be preserved by this transformation. This mapping is given by:

$$s \to \frac{2}{T}\frac{z-1}{z+1} \qquad \text{or} \qquad z \to \frac{1+\frac{Ts}{2}}{1-\frac{Ts}{2}}$$

This method is generally used when a continuous-time controller $K(s)$ should be converted to $K(z)$ for implementation on a digital microcontroller.

Example 1.4. Find a discrete-time model for $G(s) = (s+1)^{-1}$ with a sampling period of $T = 0.2s$.

Using the ZOH method we obtain:

$$\begin{aligned}
G(z) &= (1-z^{-1})\mathcal{Z}\left\{\mathcal{L}^{-1}\left[\frac{1}{s(s+1)}\right]\Big|_{kT}\right\} \\
&= (1-z^{-1})\mathcal{Z}\left\{\mathcal{L}^{-1}\left[\frac{1}{s} - \frac{1}{(s+1)}\right]\Big|_{kT}\right\} \\
&= (1-z^{-1})\left[\frac{1}{1-z^{-1}} - \frac{1}{1-e^{-T}z^{-1}}\right] \\
&= 1 - \frac{1-z^{-1}}{1-e^{-T}z^{-1}} = \frac{(1-e^{-T})z^{-1}}{1-e^{-T}z^{-1}} = \frac{0.18z^{-1}}{1-0.82z^{-1}}
\end{aligned}$$

The Tustin's method gives:

$$G(z) = \left.\frac{1}{s+1}\right|_{s=\frac{2}{T}\frac{z-1}{z+1}} = \frac{1}{\frac{2}{T}\frac{z-1}{z+1}+1} = \frac{T(z+1)}{(2+T)z-(2-T)} = \frac{0.1(1+z^{-1})}{1.1-0.9z^{-1}}$$

In system identification, as sampled data are always used for fitting the models, discrete-time models are usually preferred. However, although unfrequent, it is always possible to fit a continuous-time model to the data.

Time domain - Frequency domain models:

A time-domain model is presented by a set of difference or differential equations. It can be parametric and represented by a state-space model or nonparametric and represented by a time-domain graph like the impulse response. The nonlinear systems are usually represented in time domain. A frequency-domain model is usually obtained from the transfer function of a linear model in a parametric representation or is given by a graph in Bode diagram in a nonparametric representation.

Example 1.5. A discrete-time linear SISO model can be represented by its impulse response

$$y(k) = \sum_{j=0}^{\infty} g(j)u(k-j) = g(k) * u(k)$$

So by knowing the impulse response $g(k)$ the output of the system for any known input $u(t)$ can be computed using the convolution sum. Here, $g(k)$ with $k \in 0, 1, \ldots$ is a nonparametric model. If we consider that $g(k) \approx 0$ for $k > K$, the above infinite sum can be approximated by a finite sum and $g(k)$ with $k \in 0, 1, \ldots, K$ will be called an FIR (Finite Impulse Response) model.

Example 1.6. A frequency-domain model for the transfer function in (1.12) is given by:

$$G(j\omega) = \frac{3e^{-j\tau\omega}}{j\omega + 4y_0}$$

in a parametric form. The magnitude and phase of $G(j\omega)$ in the Bode diagram represent a nonparametric frequency-domain model.

Input/output-State space models

A model represents the mathematical relation between the input and the output of a system. A transfer function in discrete- or continuous-time or the impulse response are the input/output models because they are the only variables in the model. Some times some other internal variables are involved in the mathematical model that are called the *states* of the system.

Linear state-space models (very suitable for MIMO linear systems) are time-domain models represented by a set of linear first order differential equations in matrix form

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

The above equations represent an n-th order MIMO linear model with p inputs and q outputs where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^p$ the vector of inputs, $y(t) \in \mathbb{R}^q$ the vector of outputs and $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times p}, C \in \mathbb{R}^{q\times n}$ and $D \in \mathbb{R}^{q\times p}$ are the matrices consisting of the model parameters.

Nonlinear models are usually represented by nonlinear state-space models like $\dot{x}(t) = f(x,u)$ and $y(t) = g(x,u)$. The state and input variables may have some constraints as well. The following equations represent a SISO nonlinear model with two states.

$$\begin{aligned}\dot{x}_1(t) &= 2x_1^2(t)x_2(t) + x_1(t)u(t)\\ \dot{x}_2(t) &= 2x_1(t) - 3x_1(t)x_2(t) + u(t)\\ y(t) &= 5x_1(t)x_2^2(t)u(t)\end{aligned}$$

Parametric-Nonparametric models:

A parametric model is described with a structure and a set of parameters like the models in (1.12) and (1.13). A nonparametric model is described by a graph or a function like the step response or the frequency response. In a computer, the nonparametric models are usually stored in a vector of finite length. So the nonparametric models can be viewed as a parametric model with a large number of parameters. For example, the impulse response of a system is considered as a nonparametric model, however, it can be also represented by a Finite Impulse Response (FIR) model which is parametric.

1.3 Input Signals

Identification methods are based on the analysis of inputs and outputs of a system to find a mathematical model. The quality of the models is directly related to the choice of the input signals in an identification experiment. The ability of a signal to sufficiently excite a physical system can be assessed by analyzing the spectrum of the signal.

1.3.1 Spectral density function

The spectrum or spectral density function of a signal is defined as the square of the magnitude of its Fourier transform. If a signal $u(k)$ has a Fourier transform $U(e^{j\omega})$,

then its spectrum, $\Phi_{uu}(\omega)$, is defined as:

$$\Phi_{uu}(\omega) = |U(e^{j\omega})|^2 \tag{1.19}$$

The spectrum of a signal can be also computed as the Fourier transform of the autocorrelation function of the signal:

$$\Phi_{uu}(\omega) = \sum_{h=-\infty}^{\infty} R_{uu}(h) e^{-j\omega hT} \tag{1.20}$$

where T is the sampling period and $R_{uu}(h)$ is the autocorrelation function. In this section, the spectrum is usually computed for normalized frequency, i.e. $T = 1$ and $\omega_s = 2\pi$. Note that the spectral density function for discrete signals is an even and periodic function with sampling frequency, ω_s, as period. Therefore, it is usually plotted for half a period $\omega \in [0\,,\,\pi]$ (the true frequency is between 0 and π/T rad/s, where T is the sampling time in second).

The autocorrelation function for deterministic discrete-time energy signals (with bounded energy) is defined as:

$$R_{uu}(h) = \lim_{N\to\infty} \sum_{k=-N}^{N} u(k)u(k-h) \tag{1.21}$$

and for power signals (with bounded power) by:

$$R_{uu}(h) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} u(k)u(k-h) \tag{1.22}$$

The equivalence of (1.19) and (1.20) for the energy signals can be shown as follows:

$$\begin{aligned}
\Phi_{uu}(\omega) &= \sum_{h=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} u(k)u(k-h)e^{-j\omega h}e^{j\omega k}e^{-j\omega k} \\
&= \sum_{k=-\infty}^{\infty} \sum_{h=-\infty}^{\infty} u(k)e^{-j\omega k}u(k-h)e^{j\omega(k-h)} \\
&= \sum_{k=-\infty}^{\infty} \sum_{s=-\infty}^{\infty} u(k)e^{-j\omega k}u(s)e^{j\omega s} \\
&= U(e^{j\omega})U(e^{-j\omega}) = |U(e^{j\omega})|^2
\end{aligned}$$

For a class of power signals, periodic signals, the equivalence of (1.19) and (1.20) can also be proved as follows:

Consider a periodic signal $u(k)$ with period M, i.e. $u(k+M) = u(k)$. Since the signal is a power signal, its autocorrelation can be computed from (1.22). However,

as the signal is periodic, the exact value of the autocorrelation function can be computed just for one period (we do not need to compute a sum over infinite values):

$$R_{uu}(h) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} u(k)u(k-h) = \frac{1}{M} \sum_{k=0}^{M-1} u(k)u(k-h) \tag{1.23}$$

It is clear that $R_{uu}(h)$ is also periodic with period M. The power spectral density function, therefore, will be a Fourier series with the following coefficients:

$$\Phi(\omega_n) = \frac{1}{M} \sum_{h=0}^{M-1} R_{uu}(h)e^{-j\omega_n h} \qquad n = 0, 1, \ldots, M-1 \tag{1.24}$$

where $\omega_n = 2\pi n/M$ and $n = 0, 1, \ldots, M-1$. We now replace $R_{uu}(h)$ in (1.24) with the expression in (1.23):

$$\Phi_{uu}(\omega_n) = \frac{1}{M} \sum_{h=0}^{M-1} \frac{1}{M} \sum_{k=0}^{M-1} u(k)u(k-h)e^{-j\omega_n h} \tag{1.25}$$

Let's define $s = k - h$ and replace h with $k - s$ in the above equation to obtain:

$$\Phi_{uu}(\omega_n) = \frac{1}{M} \sum_{s=k}^{k-M+1} \frac{1}{M} \sum_{k=0}^{M-1} u(k)u(s)e^{-j\omega_n k}e^{j\omega_n s} \tag{1.26}$$

$$= \frac{1}{M} \sum_{s=k}^{k-M+1} u(s)e^{j\omega_n s} \left[\frac{1}{M} \sum_{k=0}^{M-1} u(k)e^{-j\omega_n k}\right] \tag{1.27}$$

Since u is a periodic signal, $u(s) = u(s+M)$, and we have:

$$\frac{1}{M} \sum_{s=k}^{k-M+1} u(s)e^{j\omega_n s} = \frac{1}{M} \sum_{s=0}^{M-1} u(s)e^{j\omega_n s} \tag{1.28}$$

Therefore,

$$\Phi_{uu}(\omega_n) = U(e^{-j\omega_n})U(e^{j\omega_n}) = |U(j\omega_n)|^2 \tag{1.29}$$

Remark: For a non periodic power signal the same result is obtained asymptotically, i.e.:

$$\Phi_{uu}(\omega) = \lim_{N\to\infty} \frac{1}{N}|U_N(j\omega)|^2 \tag{1.30}$$

where

$$U_N(j\omega) = \sum_{k=1}^{N} u(k)e^{j\omega k}$$

Richness of a signal

The richness of a signal, in system identification, is closely related to the characteristics of its spectrum. Three important characteristics of a signal for system identification are given below:

Degree of excitation: The number of non-zero values of $\Phi_{uu}(\omega)$ between $[0\,, 2\pi[$, in a normalized frequency scale, is a measure of the richness of a signal and is related to the number of parameters that can be estimated by the signal.

Signal energy: The integral of the spectrum is equal to the energy of the signal:

$$\frac{1}{2\pi}\int_0^{2\pi}\Phi_{uu}(\omega)d\omega = \sum_{k=-\infty}^{\infty} u^2(k)$$

for energy signals and equal to the power of the signal:

$$\frac{1}{2\pi}\int_0^{2\pi}\Phi_{uu}(\omega)d\omega = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{k=-N}^{N} u^2(k)$$

for power signals. Having a good signal to noise ratio leads to more accurate estimation of the parameters. The above relation can be proved easily by evaluating the inverse Fourier transform of the spectrum at $h = 0$:

$$R_{uu}(h) = \frac{1}{2\pi}\int_0^{2\pi}\Phi_{uu}(\omega)e^{j\omega h}d\omega$$

Spectrum shape: The shape of the spectrum of the input signal shows in which frequencies better fit will be achieved. If the spectrum is large, or *rich*, in a frequency zone, we can expect to obtain a good model of the system at that region. In the contrary, a good model will not be identified in the frequency regions that are not well excited by the input signal.

In the next sections, some commonly used input signals are introduced and their spectral properties described.

1.3.2 Impulse signal

A Kronecker impulse function is defined in (1.17). The Fourier transform of $\delta(k)$ is given by:

$$\Delta(e^{j\omega}) = \sum_{k=-\infty}^{\infty}\delta(k)e^{-j\omega k} = 1$$

Therefore, its spectrum will be equal to 1 as well. The impulse signal, then, have a degree of excitation equal to infinity (because the spectrum is a continuous function), an energy of 1 and a flat shape spectrum exciting equally all frequencies. From

a theoretical point of view, the impulse signal is ideal for system identification. However, in terms of energy, it has a limitation. In fact, increasing the energy of the signal can be done by increasing the amplitude of the impulse that is not possible in many applications because of the nonlinear behaviour of the physical systems for the large inputs.

1.3.3 Step signal

A step signal is the most common used signal for identification of simple models of industrial plants. There is only one parameter, the amplitude of the step, that should be chosen. The steady-state gain, rise-time, settling time and dominant time constant can be roughly estimated from the step response (see Section 2.2.1).

In fact, the spectrum of a step signal is not well defined because a step signal does not have a Fourier transform. To have a better idea of the excitation frequencies of a step function, it is better to look at the spectrum of a pulse function with a period MT, chosen larger than the settling time of the system.

$$u(k) = \begin{cases} \alpha & 0 \leq k < M \\ 0 & k < 0 \text{ and } k \geq M \end{cases} \tag{1.31}$$

This pulse signal is an energy signal and its autocorrelation function will be a triangular function:

$$R_{uu}(h) = \lim_{N\to\infty} \sum_{k=-N}^{N} u(k)u(k-h) = \sum_{k=-M}^{M} u(k)u(k-h) = (M - |h|)\alpha^2 \tag{1.32}$$

The energy spectral density of a pulse signal, the Fourier transform of the autocorrelation function, is obtained by:

$$\Phi_{uu}(\omega) = \sum_{h=-\infty}^{\infty} R_{uu}(h)e^{-j\omega hT} = \sum_{h=-M}^{M} (M - |h|)\alpha^2 e^{-j\omega hT} \tag{1.33}$$

For $\alpha = 1$ and $M = 10$, the pulse function, its autocorrelation function and its spectral density function are given in Fig. 1.4. It can be observed that a step signal has good excitation in low frequencies (up to π/M). Therefore, it is not a good signal for exciting middle and high frequency regions of the system. For $M = 1$, the input signal will be a Kronecker delta function, $\delta(k)$, its autocorrelation function will be also a Kronecker delta, $\delta(h)$, and its spectrum will be equal to 1 for all $\omega \in [0\,,\,\pi]$ exciting all frequencies.

1.3.4 Sum of sinusoids

For exciting higher frequencies, a sum of sinusoids input signal can be defined as:

$$u(k) = \sum_{i=1}^{m} a_i \sin(\omega_i k + \phi_i) \tag{1.34}$$

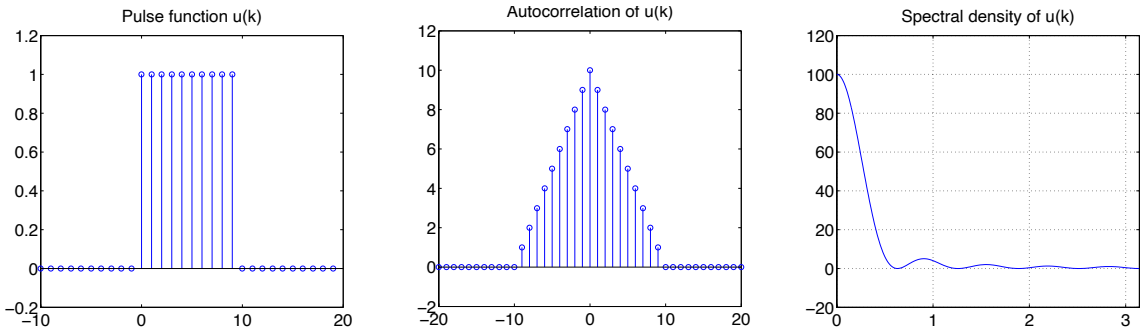


Figure 1.4: Pulse function $u(k)$, its autocorrelation function $R_{uu}(h)$ and its spectral density function $\Phi_{uu}(\omega)$

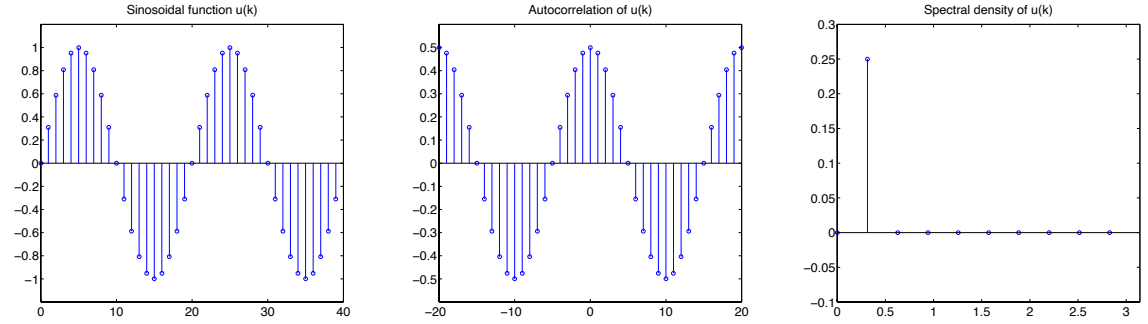


Figure 1.5: Sinusoidal function $u(k)$, its autocorrelation function $R_{uu}(h)$ and its spectral density function $\Phi_{uu}(\omega)$

where ω_i is the normalized frequency and belongs to $[0\,,\,\pi[$. For this type of signals, the amplitude a_i, the excitation frequencies ω_i and the phase angles ϕ_i can be chosen.

For one sinusoidal signal, $m = 1$, with $\omega_1 = \pi/10$ or $M = 20$, the input signal, its autocorrelation function and its spectrum are shown in Fig. 1.5. It can be observed that a sinusoid signal excites only one frequency.

The spectrum of the sinusoidal signal can be computed analytically. Knowing that the Fourier series coefficients of a sinusoidal signal is given by:

$$u(k) = a_1 \sin(\omega_1 k) = \frac{-a_1}{2j} e^{-j\omega_1 k} + \frac{a_1}{2j} e^{j\omega_1 k} \tag{1.35}$$

Therefore, $U(e^{j\omega_n})$ has one component at ω_1 and another at $-\omega_1$ with an amplitude of $a_1/2$. So its power spectral density has one component with amplitude of $a_1^2/4$ for positive frequencies in the interval $[0\,,\,\pi[$.

Figure 1.6 shows the autocorrelation function and the spectrum of the following multi-sinusoidal function:

$$u(k) = \sum_{i=1}^{10} \sin(\pi i k/10) \tag{1.36}$$

Note that the spectrum of the periodic functions is discrete, so the spectrum has

components only in the excited frequencies.

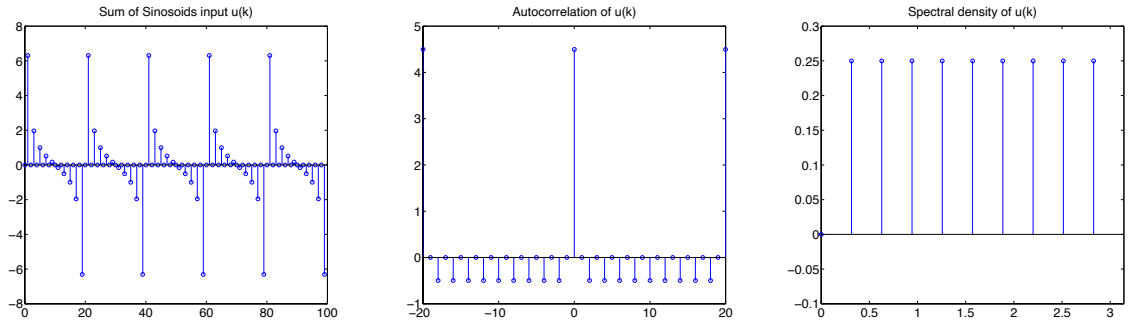


Figure 1.6: Multi-sinusoidal function $u(k)$, its autocorrelation function $R_{uu}(h)$ and its spectral density function $\Phi_{uu}(\omega)$

It is very common in industry to add a random phase to the sinusoidal signal. This will give a DC offset to the signal at zero frequency and may lead to a lower amplitude of the excitation signal. Figure 1.7 shows the autocorrelation function and the spectrum of the following multi-sinusoidal function:

$$u(k) = \sum_{i=1}^{10} \sin(\pi i k/10 + \phi_i) \tag{1.37}$$

where ϕ_i is a random phase with uniform distribution in the interval $[-\pi/2\,,\,\pi/2]$.

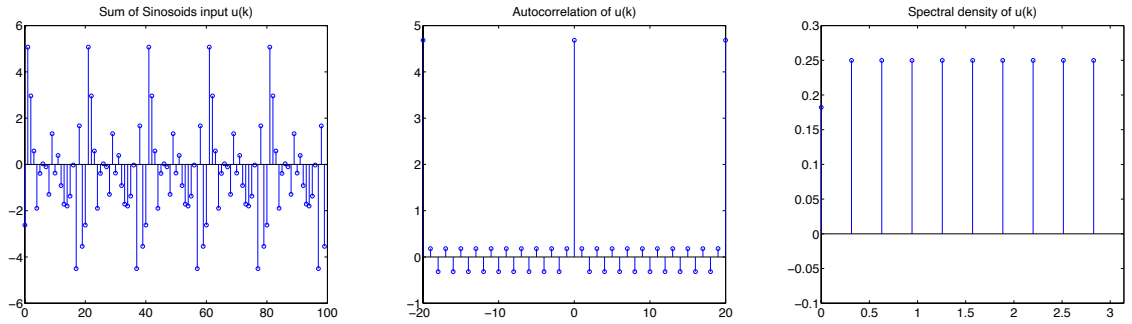


Figure 1.7: Multi-sinusoidal function $u(k)$ with random phase, its autocorrelation function $R_{uu}(h)$ and its spectral density function $\Phi_{uu}(\omega)$

1.3.5 Discrete random processes

By a multi-sinusoidal function, only a limited number of frequencies are excited. By increasing the number of sinusoidal signals the amplitude of the excitation signal increases and cannot be directly controlled. An alternative is to use a non periodic random signal that can excite all frequencies and its amplitude can be controlled.

A discrete random process, $u(k)$, is a sequence of random variables defined for every integer k. The mean value is defined as $\mathbb{E}\{u(k)\} = \mu(k)$ and can be computed as the limit of the average on infinite number of realizations for the same value of k:

$$\mathbb{E}\{u(k)\} = \lim_{M\to\infty} \frac{1}{M} \sum_{n=1}^{M} u_n(k) \tag{1.38}$$

where $u_n(k)$ is the n-th realization of $u(k)$. In a stationary process, the statistics of the random process do not depend on k, i.e. the mean value and the variance are not time-varying. For some stationary processes, the mean value can be computed by averaging over k for the same realization of the random process instead of over realizations for the same k. For this type of processes, called *ergodic*, the expectation of u is given as:

$$\mathbb{E}\{u\} = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} u(k) \tag{1.39}$$

For a stationary discrete random process the autocorrelation function is defined as:

$$R_{uu}(h) = \mathbb{E}\{u(k)u(k-h)\} = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} u(k)u(k-h) \tag{1.40}$$

and the power spectral density as:

$$\Phi_{uu}(\omega) = \sum_{h=-\infty}^{\infty} R_{uu}(h)e^{-j\omega h} \tag{1.41}$$

Discrete white noise

A discrete white noise is a sequence of random processes which has, like a white light, a uniform spectral density function in all frequencies between 0 and the Nyquist frequency (sampling frequency divided by 2). A white noise is not correlated with its delayed values. A zero-mean stationary white noise is defined as (see Fig. 1.8):

$$\mathbb{E}\{e(k)\} = 0 \qquad \forall k \tag{1.42}$$

$$R_{ee}(h) = \mathbb{E}\{e(k)e(k-h)\} = \sigma_e^2\delta(h) = \begin{cases} \sigma_e^2 & \text{for } h = 0 \\ 0 & \text{for } h \neq 0 \end{cases} \tag{1.43}$$

$$\Phi_{ee}(\omega) = \sum_{h=-\infty}^{\infty} R_{ee}(h)e^{-j\omega h} = \sum_{h=-\infty}^{\infty} \sigma_e^2\delta(h) = \sigma_e^2 \tag{1.44}$$

Note that a white noise can have any probability density function (e.g. uniform, exponential, Gaussian, etc.). However, in practice, usually a Gaussian distribution $\mathcal{N}(0, \sigma_e^2)$ or a uniform distribution for noise is assumed.

Remark: A discrete white noise has a physical realization because it has a finite energy (the integral of its spectrum is finite). However, a continuous white noise with the autocorrelation $R_{ee}(\tau) = C\delta(\tau)$ is a hypothetical signal and cannot be physically realized because it has infinite energy (its spectrum is uniform in all frequencies from 0 to infinity).

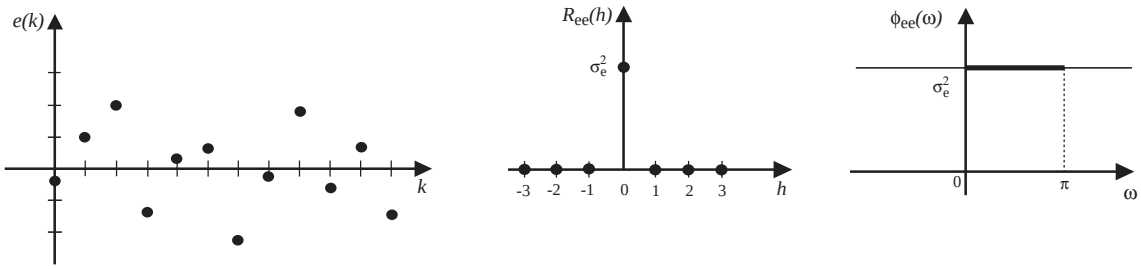


Figure 1.8: Discrete white noise $e(k)$, its autocorrelation and spectral density functions

Discrete colored noise

Most of random signals can be described by a filtered discrete white noise. A filtered white noise has a non uniform spectrum like a colored light, so is called a colored noise. For example, a zero-mean white noise, $e(k)$, with variance σ_e^2, filtered by a *Moving Average* filter represents a colored noise:

$$\begin{aligned} n(k) &= e(k) + c_1 e(k-1) + \cdots + c_{n_c} e(k-n_c) \\ &= (1 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c}) e(k) = C(q^{-1}) e(k) \end{aligned} \tag{1.45}$$

The mean value and the variance of this colored noise for $n_c = 1$ are:

$$\begin{aligned} \mathbb{E}\{n(k)\} &= \mathbb{E}\{(1 + c_1 q^{-1}) e(k)\} = \mathbb{E}\{e(k) + c_1 e(k-1)\} \\ &= \mathbb{E}\{e(k)\} + c_1 \mathbb{E}\{e(k-1)\} = 0 \\ \mathbb{E}\{n(k)n(k)\} &= \mathbb{E}\{[e(k) + c_1 e(k-1)]^2\} \\ &= \mathbb{E}\{e^2(k)\} + c_1^2 \mathbb{E}\{e^2(k-1)\} + 2c_1 \mathbb{E}\{e(k)e(k-1)\} \end{aligned}$$

Note that $\mathbb{E}\{e^2(k-1)\}$ in the second term is equal to σ_e^2 because the white noise is a stationary signal and the third term is zero because of the uncorrelation property of the white noise. Therefore:

$$R_{nn}(0) = \mathbb{E}\{n(k)n(k)\} = (1 + c_1^2)\sigma_e^2 \tag{1.46}$$

The autocorrelation of $n(k)$ for $h \neq 0$ can be computed as follows. For $h = 1$ we have:

$$n(k-1) = e(k-1) + c_1 e(k-2) \tag{1.47}$$

$$\begin{aligned} R_{nn}(1) &= \mathbb{E}\{n(k)n(k-1)\} = \mathbb{E}\{[e(k) + c_1 e(k-1)][e(k-1) + c_1 e(k-2)]\} \\ &= \mathbb{E}\{e(k)e(k-1)\} + c_1 \mathbb{E}\{e(k)e(k-2)\} + c_1 \mathbb{E}\{e(k-1)e(k-1)\} \\ &\quad + c_1^2 \mathbb{E}\{e(k-1)e(k-2)\} = c_1 \sigma_e^2 \end{aligned} \tag{1.48}$$

It is easy to show that for $|h| > 1$, we have $R_{nn}(h) = 0$, which leads to the graph of the autocorrelation function of $n(k)$ in Fig. 1.9. The power spectral density then is

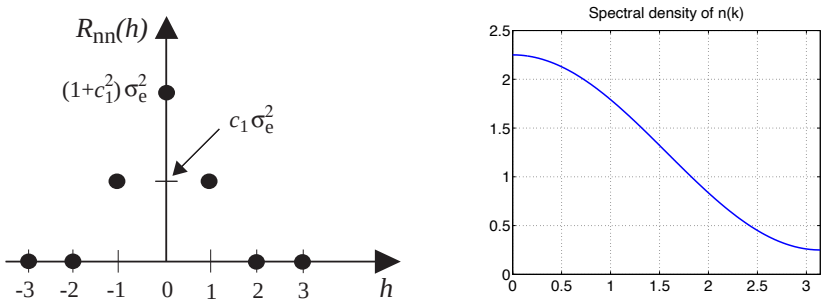


Figure 1.9: Autocorrelation and power spectral density functions of a first-order MA colored noise

computed as:

$$\Phi_{nn}(\omega) = \sum_{h=-\infty}^{\infty} R_{nn}(h)e^{-j\omega h} = c_1\sigma_e^2 e^{j\omega} + (1+c_1^2)\sigma_e^2 + c_1\sigma_e^2 e^{-j\omega}$$
$$= |1 + c_1 e^{-j\omega}|^2\sigma_e^2$$

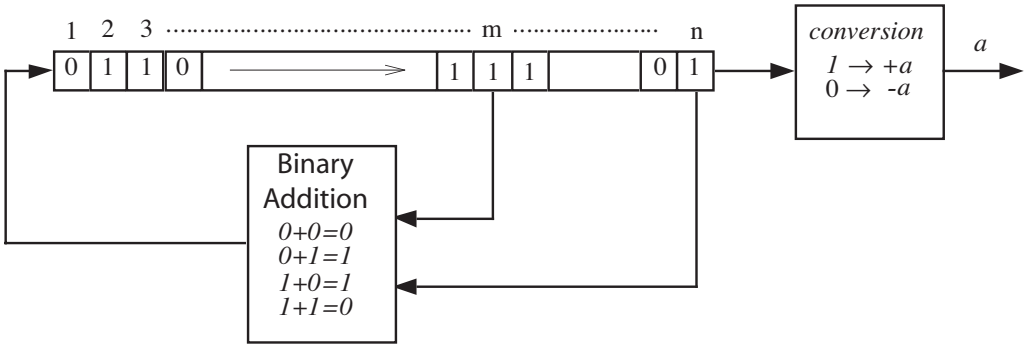
and shown in Fig. 1.9 for $c_1 = 0.5$ and $\sigma_e^2 = 1$. It can be seen that by a first-order moving average filter, the spectrum of the signal becomes higher in low frequencies and lower in high frequencies. With different choice of the noise filter, the spectrum of the excitation signal can be shaped appropriately. In general, the spectrum of a filtered white noise $n(k) = F(q^{-1})e(k)$, where $F(q^{-1})$ is a rational stable filter is given by:

$$\Phi_{nn}(\omega) = |F(e^{-j\omega})|^2\sigma_e^2 \tag{1.49}$$

Note also that the spectrum is continuous, meaning that all frequencies are excited. In practice, in order to increase the energy or the variance of a white noise, its amplitude will be limited to shift between a maximum and minimum value each with the same probability of 0.5.

1.3.6 Pseudo random binary sequence (PRBS)

Pseudo random binary sequence (PRBS) is a deterministic periodic signal that shifts between two levels with some specific rules. It is generated using a shift register of length n. The shift register is initialized with a binary value (not equal to zero). At each clock pulse, the contents of the shift register will shift one bit to the right side while a new value will be fed to the shift register from the left side. This new value is the binary addition of two (or more) bits of the shift register. The last bit of the shift register will generate a sequence with two different amplitudes, usually $+a$ and $-a$ and will be used as an input signal for system identification purpose. It is clear that this scheme will generate a periodic signal. The maximum period is obtained if all possible combinations of binary values in the shift register are produced. The total number of combinations is $2^n - 1$ (excluding zero value for all bits). So the



Shift register length n	3	4	5	6	7	8	9
Feedback value	2⊕3	1⊕4	2⊕5	1⊕6	1⊕7	1⊕2⊕7⊕8	4⊕9

Figure 1.10: Generation of a PRBS using a shift register

maximum period of a PRBS is $M = (2^n - 1)$ samples. Figure 1.10 shows the scheme for the generation of a PRBS and the bits that should be added to obtain a PRBS with maximum period.

In order to study the richness of a PRBS, its power spectral density function (the Fourier transform of its autocorrelation function), should be computed. The autocorrelation function of a PRBS signal can be shown to be:

$$R_{uu}(h) = \frac{1}{M}\sum_{k=0}^{M-1} u(k)u(k-h) = \begin{cases} a^2 & h = 0, \pm M, \pm 2M, \dots \\ -a^2/M & \text{elsewhere} \end{cases} \tag{1.50}$$

The autocorrelation function is periodic with period M and its value for $h \neq 0, \pm M, \pm 2M, \dots$ are very small and negligible, if M is large enough. In this case, the autocorrelation function will be very close to that of a white noise for $-M < h < M$.

The power spectral density function of a PRBS can be computed using the Fourier transform (Fourier series in this case) of its autocorrelation function as:

$$\Phi_{uu}(\omega_n) = \frac{1}{M}\sum_{h=0}^{M-1} R_{uu}(h)e^{-j\omega_n h} = \begin{cases} \dfrac{a^2}{M^2} & n = 0 \\ \dfrac{a^2}{M^2}(M+1) & n = 1, \dots, M-1 \end{cases} \tag{1.51}$$

where $\omega_n = \frac{2\pi n}{M}$. Figure 1.11 shows a PRBS of period $M = 15$ ($n = 4$) and $a = 1$, its autocorrelation and its spectral density function. The spectrum of PRBS is very similar to that of the sum of sinusoids (or sampled spectrum of a white noise). It has a too small excitation for zero frequency (a^2/M^2) and a uniform discrete excitation from 0 to π with a value of $(M+1)a^2/M^2$.

Degree of excitation: The degree of excitation of PRBS is equal to M and the number of non-zero values of its spectrum between $[0\,, \pi[$ is equal to $(M+1)/2$. Therefore, a PRBS generated by a shift register of length n has the same degree of

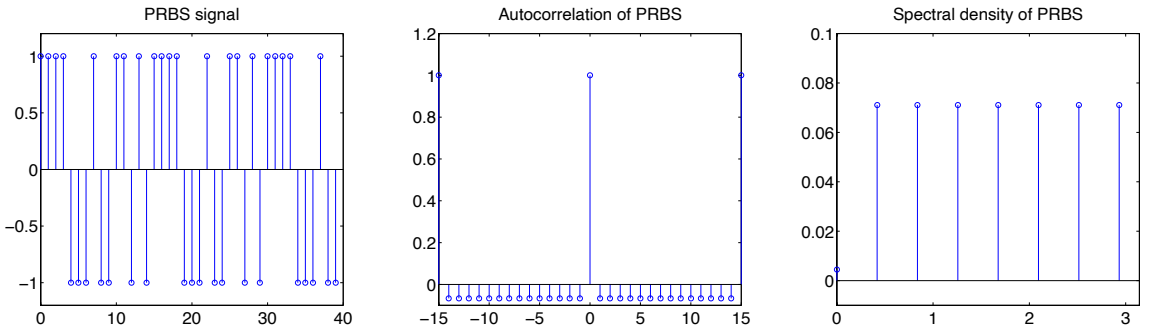


Figure 1.11: PRBS, its autocorrelation and spectral density function

excitation of sum of $2^{(n-1)}$ sinusoids. However, its amplitude can be fixed by the user to the maximum value tolerated by the system. Therefore, a PRBS has more energy than a sum of sinusoids with the same degree of excitation and same length. Note that a discrete white noise as well as a pulse function have an infinite degree of excitation because they have a continuous spectrum.

Frequency divider: The PRBS has a uniform spectrum in all frequencies (except at zero frequency). However, its spectrum can be shaped easily by using a frequency divider for the clock pulse of the shift register. The clock pulse has the same frequency as the sampling frequency, but if for example its frequency is divided by 2, after each two sampling periods, the bits in the register will be shifted. Therefore, the period of the PRBS will be multiplied by two (like stretching the signal in time) and as a result its low frequency components will become larger than the high frequency components. Figure 1.12 shows a PRBS with a divider 2 and 3, its autocorrelation function and its spectrum. It should be mentioned that the autocorrelation function and the spectrum of a PRBS with divider 2 are very similar to that of a filtered white noise with a first order moving average model (see Fig. 1.9). By increasing the divider value, the signal becomes richer in low frequencies and poorer in high frequencies. The frequency divider should be used with some cautions because for large dividers the richness of the signal at high frequency will be significantly reduced.

1.4 Least Squares Algorithm

System identification is an application field of the estimation theory. A parameter estimation problem is to determine an estimator $\hat{\theta}$ of an unknown parameter vector θ which depends on a given set of measured data. Therefore, the estimator is a function of the measured data. In a parametric identification problem, the measured data are the input and output signals and an estimator of the model parameters should be developed. Even nonparametric identification problems can be viewed as a standard estimation problem when number of to-be-estimated parameters are large. In general, a stochastic approach, in which the measured data are realization

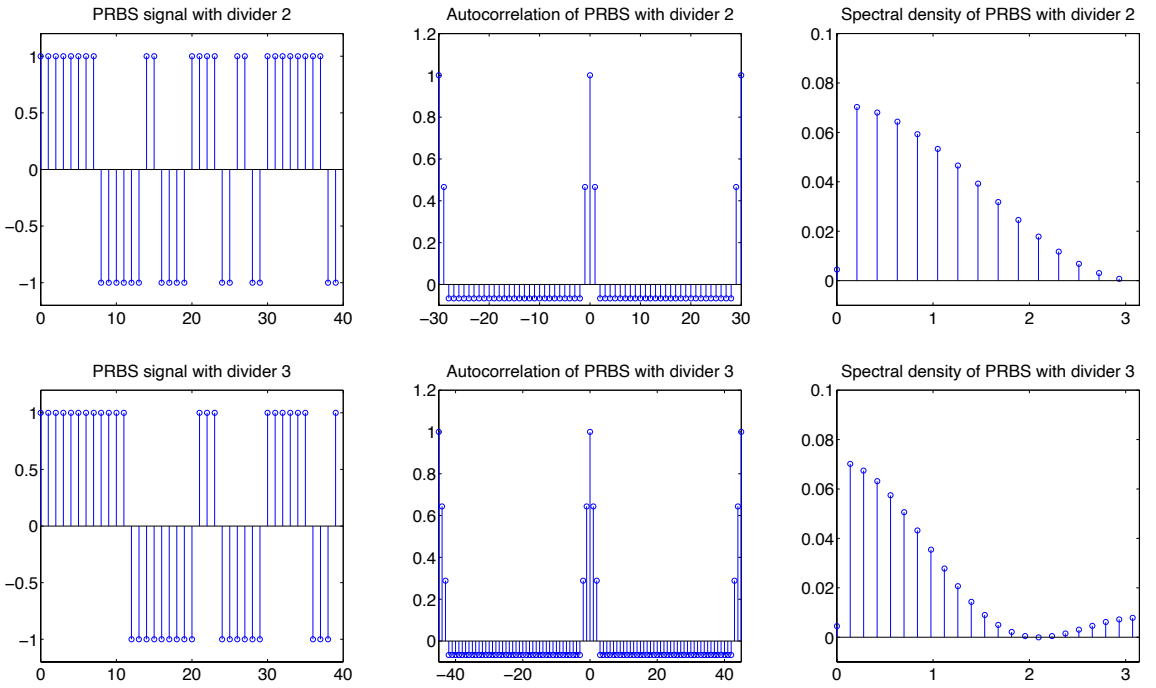


Figure 1.12: PRBS with a frequency divider of 2 and 3, its autocorrelation and spectral density function

of random variables, is employed for parameter estimation. In this section, the well-known least squares algorithm is discussed.

The method of least squares dates back to 1795, when it was used to study the planetary motions by Gauss. This method needs only a data generation model and is widely used in practice due to its simplicity. Furthermore, its statistical performance can be assessed by some specific assumptions about the probabilistic structure of the data.

Consider a measured signal $y(k)$ and assume a polynomial trend model for this signal as:

$$y(k) = a_1 + a_2 k + a_3 k^2 + \cdots + a_n k^{n-1} \tag{1.52}$$

The estimation problem is to find the parameters a_1 to a_n from N measurements $y(1)$ to $y(N)$. In a least squares problem, an equation error is first defined as:

$$\varepsilon(k) = y(k) - (a_1 + a_2 k + a_3 k^2 + \cdots + a_n k^{n-1}) \tag{1.53}$$

Then, the parameters are estimated such that the sum of squares of the equation errors is minimized.

If the data generation model is linear in parameters, like the polynomial trend problem, the signal model can be represented as a *linear regression* and consequently the LS problem becomes linear and can be solved analytically.

1.4.1 Linear regression problem

A linear regression problem is written as:

$$y(k) = \phi^T(k)\theta \tag{1.54}$$

where $y(k)$ is the measured signal, $\phi^T(k)$ is an n-dimensional vector of known quantities and θ is an n-dimensional vector of unknown parameters. For example, for the polynomial trend model, we have:

$$\begin{aligned} \phi^T(k) &= [1 \quad k \quad k^2 \quad \dots \quad k^{n-1}] \\ \theta^T &= [a_1 \quad a_2 \quad a_3 \quad \dots \quad a_n] \end{aligned}$$

Given N measurements, the following system of linear equations is obtained:

$$\begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} \phi^T(1) \\ \vdots \\ \phi^T(N) \end{bmatrix} \theta \tag{1.55}$$

that can be written as

$$Y = \Phi\theta \tag{1.56}$$

with the obvious definition of Y and Φ and the following dimensions:

$$\begin{array}{lcll} Y & : & (N \times 1) & \text{measurement vector} \\ \Phi & : & (N \times n) & \text{observation matrix} \\ \theta & : & (n \times 1) & \text{parameter vector} \end{array}$$

It is clear that in order to have a solution, the number of measurement N should be equal to n. Then, the observation matrix Φ becomes a square matrix and the linear system of equations has a solution if this matrix is nonsingular. In practice, the number of measurements are much higher than the number of parameters and so the system of equations becomes overdetermined and can be solved in the least squares error sense.

The equation error is given by:

$$\varepsilon(k) = y(k) - \phi^T(k)\theta \tag{1.57}$$

and the error vector by:

$$\mathcal{E}^T = [\varepsilon(1) \quad \varepsilon(2) \quad \dots \quad \varepsilon(N)] \tag{1.58}$$

The vector of parameters can be obtained by the minimization of the following criterion:

$$\begin{aligned} J(\theta) &= \sum_{k=1}^{N} \varepsilon^2(k) = \mathcal{E}^T\mathcal{E} \\ &= [Y - \Phi\theta]^T[Y - \Phi\theta] = Y^TY - 2Y^T\Phi\theta + \theta^T\Phi^T\Phi\theta \end{aligned} \tag{1.59}$$

The minimum can be obtained by setting the gradient of $J(\theta)$ to zero:

$$\frac{dJ(\theta)}{d\theta} = -2\Phi^T Y + 2\Phi^T \Phi\theta = 0 \tag{1.60}$$

which leads to:

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y \tag{1.61}$$

The solution $\hat{\theta}$ is the unique minimum point, if the Hessian of the criterion $J(\theta)$ is positive definite. The Hessian of $J(\theta)$ is equal to $2\Phi^T\Phi$ which is non negative by structure. Therefore, $\hat{\theta}$ is the unique minimum if $\Phi^T\Phi$, called the information matrix, is non singular or, in other words, Φ is full rank.

Note that (1.61) can be rewritten in the equivalent form:

$$\hat{\theta} = \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \left[\sum_{k=1}^{N} \phi(k)y(k)\right] \tag{1.62}$$

1.4.2 Weighted least squares method

With N measurements at instants $k = 1, 2, \ldots, N$, the error signal can be written in vector form as:

$$\mathcal{E} = Y - \Phi\theta \tag{1.63}$$

In practice, it is possible that all errors $\varepsilon(1), \ldots, \varepsilon(N)$ do not have the same importance in the criterion $J(\theta)$. It is thus possible to weight the errors as follows:

$$\mathcal{E}_W \equiv W[Y - \Phi\theta] \tag{1.64}$$

where W is a weighting matrix of dimension $(N \times N)$ and often diagonal. The quadratic criterion using the weighted errors becomes:

$$J(\theta) = \mathcal{E}_W^T \mathcal{E}_W = \mathcal{E}^T W^T W \mathcal{E} = [Y - \Phi\theta]^T W^T W [Y - \Phi\theta] \tag{1.65}$$

The vector of the parameter estimates can be found in a similar way:

$$\hat{\theta} = (\Phi^T W^T W \Phi)^{-1} \Phi^T W^T W Y \tag{1.66}$$

1.4.3 Analysis of LS estimates in stochastic case

The LS estimate in (1.61) is optimal in deterministic case, i.e. with no stochastic assumption in data. If we assume that the measurements are the realizations of random variables, the estimate $\hat{\theta}$ becomes also a random variable and its performance can only be described by its probability density function (PDF) or statistically.

Assume that the data satisfy

$$y(k) = \phi^T(k)\theta_0 + e(k) \tag{1.67}$$

where θ_0 is called the true parameter vector and $e(k)$ is a stationary zero-mean random variable, i.e. $\mathbb{E}\{e(k)\} = 0$. Note that, this is a strong assumption on the data generation model. In the deterministic case, we had no assumption on the *true* model and the equation error could contain the modeling and measurement errors. However, without this assumption, it is very difficult to analyze the quality of the estimates.

Equation (1.67) can be written in matrix form as:

$$Y = \Phi\theta_0 + E \tag{1.68}$$

where

$$E^T = [e(1) \quad e(2) \quad \dots \quad e(N)] \tag{1.69}$$

Furthermore, we assume that Φ is a deterministic matrix and the covariance of E is equal to R.

$$\text{cov}(E) = \mathbb{E}\{EE^T\} = R \tag{1.70}$$

The covariance matrix is a positive definite matrix and will be equal to $\sigma^2 I$ if $e(t)$ is a white noise with variance σ^2.

Bias of estimates: The first property of the LS estimator in (1.61) is its unbiasedness, i.e. $\mathbb{E}\{\hat{\theta}\} = \theta_0$. This can be proved by replacing Y from (1.68) into (1.61) as follows:

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T(\Phi\theta_0 + E) \tag{1.71}$$

and therefore,

$$\mathbb{E}\{\hat{\theta}\} = \theta_0 + (\Phi^T\Phi)^{-1}\Phi^T\mathbb{E}\{E\} = \theta_0 \tag{1.72}$$

Covariance of the parameters: The quality of an unbiased estimator depends on the covariance of the estimates, which can be computed for the LS estimator in (1.61):

$$\begin{aligned}
\text{cov}(\hat{\theta}) &= \mathbb{E}\{(\hat{\theta} - \theta_0)(\hat{\theta} - \theta_0)^T\} && (1.73)\\
&= \mathbb{E}\{[(\Phi^T\Phi)^{-1}\Phi^T E][(\Phi^T\Phi)^{-1}\Phi^T E]^T\} && (1.74)\\
&= (\Phi^T\Phi)^{-1}\Phi^T\mathbb{E}\{EE^T\}\Phi(\Phi^T\Phi)^{-1} = (\Phi^T\Phi)^{-1}\Phi^T R\Phi(\Phi^T\Phi)^{-1} && (1.75)
\end{aligned}$$

If $e(k)$ is a white noise with variance σ^2, the covariance of the estimates becomes:

$$\text{cov}(\hat{\theta}) = \sigma^2(\Phi^T\Phi)^{-1} \tag{1.76}$$

Variance of noise: In many applications the variance of noise is not a priori known and should be also estimated. In the following, we show that an unbiased estimate of σ^2 can be obtained by:

$$\hat{\sigma^2} = \frac{1}{N-n}J(\hat{\theta}) \tag{1.77}$$

where $J(\hat{\theta})$ is called *the residual* and given by:

$$\begin{aligned} J(\hat{\theta}) &= [Y - \Phi\hat{\theta}]^T[Y - \Phi\hat{\theta}] \\ &= [Y - \Phi(\Phi^T\Phi)^{-1}\Phi^T Y]^T[Y - \Phi(\Phi^T\Phi)^{-1}\Phi^T Y] \\ &= Y^T Y - Y^T\Phi(\Phi^T\Phi)^{-1}\Phi^T Y = Y^T[I - \Phi(\Phi^T\Phi)^{-1}\Phi^T]Y \\ &= (\Phi\theta_0 + E)^T[I - \Phi(\Phi^T\Phi)^{-1}\Phi^T](\Phi\theta_0 + E) \\ &= E^T[I - \Phi(\Phi^T\Phi)^{-1}\Phi^T]E \\ &= tr\{E^T[I - \Phi(\Phi^T\Phi)^{-1}\Phi^T]E\} \\ &= tr\{[I - \Phi(\Phi^T\Phi)^{-1}\Phi^T]EE^T\} \end{aligned}$$

In the calculation, we use the fact that a scalar value is equal to its trace and $tr\{AB\} = tr\{BA\}$. Now, we should show that the noise variance estimate in (1.77) is unbiased, i.e.:

$$\mathbb{E}\{\hat{\sigma^2}\} = \frac{1}{N-n}\mathbb{E}\{J(\hat{\theta})\} = \sigma^2 \tag{1.78}$$

We have:

$$\begin{aligned} \mathbb{E}\{J(\hat{\theta})\} &= tr\{[I - \Phi(\Phi^T\Phi)^{-1}\Phi^T]\mathbb{E}\{EE^T\}\} \\ &= [tr\{I_N\} - tr\{(\Phi^T\Phi)^{-1}\Phi^T\Phi\}]\sigma^2 = (N-n)\sigma^2 \end{aligned} \tag{1.79}$$

which completes the proof. Note that we assumed that Φ is deterministic so expectation is only with respect to noise.

Colored noise: If $e(k)$ is not white, it is reasonable to weight the quadratic errors by the inverse of the covariance of noise. In other words, we consider more weights for the errors that have less variance or more accurate. In this case, the weighted least squares algorithm can be used where:

$$W^T W = R^{-1} \tag{1.80}$$

This leads to the following parameter estimates:

$$\hat{\theta} = (\Phi^T R^{-1}\Phi)^{-1}\Phi^T R^{-1} Y \tag{1.81}$$

The covariance of the estimates is computed as:

$$\text{cov}(\hat{\theta}) = \mathbb{E}\{(\hat{\theta} - \theta_0)(\hat{\theta} - \theta_0)^T\} = (\Phi^T R^{-1}\Phi)^{-1} \tag{1.82}$$

Note that it can be shown this estimate has the minimum variance among all unbiased estimators for this problem.

1.4.4 Relation with Maximum Likelihood Estimator

The estimation theory deals with the problem of extracting information from observed data, where the observed data are realizations of stochastic variables. Consider that the observations are represented by a sequence of random variables $Y = [y(1), y(2), \ldots, y(N)]^T$ with the probability density function $p(Y, \theta)$, where θ is a n-dimensional vector of unknown parameters that describe the properties of the observed data. An estimator $\hat{\theta}$ is a function of the observed data.

The maximum likelihood estimator (MLE) chooses the parameter values that make the observed data, the most likely data to have been observed and introduced by Fisher in 1912. In fact the probability that the realization takes the value Y° is proportional to $p(Y^\circ, \theta)$. This is a deterministic function of θ for the known observed data Y° and is called the likelihood function. Therefore, a reasonable estimator of θ can obtained by maximizing the likelihood function:

$$\hat{\theta} = \arg\max \ln p(Y^\circ, \theta) \tag{1.83}$$

Remark: In practice, for the ease of calculation, usually the logarithm of the likelihood function is used.

Example 1.7. Consider a random variable $y = \theta + e$, where e is a zero-mean Gaussian white noise with variance σ^2 with the following probability density function:

$$p(y, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\theta)^2}{2\sigma^2}} \tag{1.84}$$

if we have one observation y°, what is the MLE of θ?

The likelihood function should be maximized to find an estimate of θ:

$$\begin{aligned}
\hat{\theta} &= \arg\max_{\theta} \ln \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y^\circ-\theta)^2/2\sigma^2} \right] \\
&= \arg\max_{\theta} \left[-\ln \sqrt{2\pi\sigma^2} - (y^\circ - \theta)^2/2\sigma^2 \right] \\
&= \arg\min_{\theta} (y^\circ - \theta)^2
\end{aligned}$$

It is clear that the minimum of $(y^\circ - \theta)^2$ happens for $\hat{\theta} = y^\circ$.

Example 1.8. For the previous example, what is the MLE of θ if we have N independent observation?

In this case, $Y = [y(1), y(2), \ldots, y(N)]$ and $y(i)$ has a Gaussian distribution $\mathcal{N}(\theta, \sigma^2)$. In order to compute the MLE, we should determine the joint *pdf* for the observations. Since the *pdf* of $y(i)$ is Gaussian (1.84), and $y(i)$ are independent, for the joint *pdf* we have:

$$p(Y, \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y(i)-\theta)^2/2\sigma^2} = \frac{1}{(2\pi\sigma^2)^{N/2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y(i)-\theta)^2} \tag{1.85}$$

Therefore, the MLE is:

$$\hat{\theta} = \arg\max_{\theta} \ln p(Y^\circ, \theta) = \arg\max_{\theta} \left[-\frac{N}{2} \ln 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y^\circ(i) - \theta)^2 \right]$$
$$= \arg\min_{\theta} \sum_{i=1}^{N} (y^\circ(i) - \theta)^2 \tag{1.86}$$

which coincides with the least squares estimator. The MLE is computed by putting the gradient equal to zero:

$$-2 \sum_{i=1}^{N} (y^\circ(i) - \hat{\theta}) = 0 \quad \Rightarrow \quad \hat{\theta} = \frac{1}{N} \sum_{i=1}^{N} y^\circ(i) \tag{1.87}$$

Now, we consider a more general case, when the data are generated by the following linear model:

$$y(k) = \phi^T(k)\theta_0 + e(k) \tag{1.88}$$

where θ_0 is the vector of true parameters and $e(k)$ a zero-mean stationary random Gaussian process. In matrix form we have $Y = \Phi\theta_0 + E$ and $R = \mathbb{E}\{EE^T\}$ is the noise covariance matrix. The joint Gaussian *pdf* of Y is given by (See Appendix A):

$$p(Y, \theta_0) = \frac{1}{\sqrt{(2\pi)^N \det(R)}} e^{-\frac{1}{2}(Y - \Phi\theta_0)^T R^{-1} (Y - \Phi\theta_0)} \tag{1.89}$$

Then, the MLE is:

$$\begin{aligned}
\hat{\theta} &= \arg\max_{\theta} \ln p(Y^\circ, \theta) \\
&= \arg\max_{\theta} \left[-\frac{N}{2} \ln 2\pi \det(R) - \frac{1}{2} (Y^\circ - \Phi\theta)^T R^{-1} (Y^\circ - \Phi\theta) \right] \\
&= \arg\min_{\theta} (Y^\circ - \Phi\theta)^T R^{-1} (Y^\circ - \Phi\theta) \\
&= (\Phi^T R^{-1} \Phi)^{-1} \Phi^T R^{-1} Y^\circ
\end{aligned}$$

which coincides with the weighted least squares estimator when the weight is the inverse of the covariance matrix of noise (minimum variance estimator).

One of the main characteristics of MLE is that the estimate $\hat{\theta}$ converges in distribution to a Gaussian distribution with $\mathcal{N}(\theta_0, (\Phi^T R^{-1} \Phi)^{-1})$. This is very useful to compute the confidence intervals for the estimated parameters.

We can make the following conclusions about the least squares estimator (LSE):

- LSE is an optimal estimator in the deterministic case.
- For linear models with zero-mean white noise, LSE is unbiased and has the optimal variance.

- For linear models with zero-mean colored noise of covariance R, weighted LSE with $W^TW = R^{-1}$ is unbiased and has the optimal variance.
- For linear models with zero-mean Gaussian white noise, LSE is the same as MLE.
- For linear models with zero-mean Gaussian colored noise of covariance R, weighted LSE with $W^TW = R^{-1}$ is the same as MLE.

1.4.5 Bias-Variance Tradeoff

In the previous section we supposed that the data are generated by a *true model* and contaminated with noise. We assumed that the dimension of the true parameter vector is *known* and *finite*. This assumption is very restrictive and unrealistic for physical systems that have theoretically an infinite order. On the other hand, we are usually interested in simple low-order model that describes the system approximatively. As a result, in addition to the stochastic errors, the data are usually contaminated by modelling error (or systematic error). Estimating a higher order model will give better fit for a given data set, but the variance of the parameters are higher (the variance of parameters is usually proportional to n/N). This leads to a property known as bias-variance tradeoff that concerns every estimator whatever the estimation method is.

In what follows, we show the bias-variance tradeoff for a linear model. Consider that the data are generated by the following linear regression model:

$$y(k) = \phi^T(k)\theta_0 + e(k) \tag{1.90}$$

where $e(k)$ is a zero-mean stationary random noise with variance σ^2. Let's consider an estimator $\hat{\theta}$ with a smaller dimension than that of θ_0 such that:

$$\hat{y}(t) = \varphi^T(k)\hat{\theta} \tag{1.91}$$

where $\varphi(k)$ is the regressor vector with the same dimension as θ^*. Since the estimator is a random variable, we define $\theta^* = \mathbb{E}\{\hat{\theta}\}$. In this representation the quality of the estimator cannot be assessed using the biasedness of the parameters (since θ_0 and θ^* have different dimensions). Therefore, we define the output estimation error as:

$$\varepsilon(k) = y(k) - \hat{y}(k) = y(k) - \varphi^T(k)\hat{\theta} \tag{1.92}$$

which can be used to assess the quality of an estimator by the mean square error (MSE):

$$\text{MSE} := \mathbb{E}\left\{\varepsilon^2(k)\right\} = \mathbb{E}\left\{[y(k) - \varphi^T(k)\hat{\theta}]^2\right\} \tag{1.93}$$

Let us define the bias and the variance error as follows:

$$\text{Bias error} := \phi^T(k)\theta_0 - \mathbb{E}\{\varphi^T(k)\hat{\theta}\} = \phi^T(k)\theta_0 - \varphi^T(k)\theta^* \tag{1.94}$$

$$\text{Variance error} := \mathbb{E}\left\{[\varphi^T(k)\hat{\theta} - \mathbb{E}\{\varphi^T(k)\hat{\theta}\}]^2\right\} = \mathbb{E}\left\{[\varphi^T(k)\hat{\theta} - \varphi^T(k)\theta^*]^2\right\} \tag{1.95}$$

The bias error shows the difference between the correct output value and the expected (or average) prediction by the model. The variance error shows the variability of the prediction in different realization of data. The variance decays as n/N, where n is the number of parameters and N number of data. For a finite number of data, we will have always some bias and variance errors. The bias usually decreases if the number of parameters increases, while the variance will augment.

We will show that for any estimator the mean square error is the sum of the variance of noise (irreducible), the variance error and the square of the bias error. The MSE can be computed as follows:

$$\mathbb{E}\left\{[y(k)-\varphi^T(k)\hat{\theta}]^2\right\}=\mathbb{E}\left\{y^2(k)+[\varphi^T(k)\hat{\theta}]^2-2y(k)\varphi^T(k)\hat{\theta}\right\} \tag{1.96}$$

Using (1.90), the first term becomes:

$$\begin{aligned}\mathbb{E}\left\{y^2(k)\right\}&=[\phi^T(k)\theta_0]^2+\mathbb{E}\left\{e^2(k)\right\}+2\phi^T(k)\theta_0\mathbb{E}\{e(k)\}\\&=[\phi^T(k)\theta_0]^2+\sigma^2\end{aligned} \tag{1.97}$$

On the other hand the variance error can be rewritten as:

$$\begin{aligned}\mathbb{E}\left\{[\varphi^T(k)\hat{\theta}-\varphi^T(k)\theta^*]^2\right\}&=\mathbb{E}\left\{[\varphi^T(k)\hat{\theta}]^2\right\}+[\varphi^T(k)\theta^*]^2-2\mathbb{E}\left\{\varphi^T(k)\hat{\theta}\right\}\varphi^T(k)\theta^*\\&=\mathbb{E}\left\{[\varphi^T(k)\hat{\theta}]^2\right\}-[\varphi^T(k)\theta^*]^2\end{aligned} \tag{1.98}$$

Therefore, the second term in (1.96) can be written as:

$$\mathbb{E}\left\{[\varphi^T(k)\hat{\theta}]^2\right\}=\mathbb{E}\left\{[\varphi^T(k)\hat{\theta}-\varphi^T(k)\theta^*]^2\right\}+[\varphi^T(k)\theta^*]^2 \tag{1.99}$$

Finally for the last term in (1.96), using (1.90), we have:

$$\mathbb{E}\left\{2y(k)\varphi^T(k)\hat{\theta}\right\}=\mathbb{E}\left\{2\phi^T(k)\theta_0\varphi^T(k)\hat{\theta}\right\}+\mathbb{E}\left\{e(k)\varphi^T(k)\hat{\theta}\right\} \tag{1.100}$$

The second term will be zero if we assume that $e(k)$ and $\phi^T(k)\hat{\theta}$ are independent. This leads to:

$$\mathbb{E}\left\{2y(k)\varphi^T(k)\hat{\theta}\right\}=2\phi^T(k)\theta_0\varphi^T(k)\theta^* \tag{1.101}$$

Therefore, the MSE can be written as:

$$\mathbb{E}\left\{[y(k)-\varphi^T(k)\hat{\theta}]^2\right\}=\underbrace{\sigma^2}_{\text{Noise error}}+\underbrace{\mathbb{E}\left\{[\varphi^T(k)\hat{\theta}-\varphi^T(k)\theta^*]^2\right\}}_{\text{Variance error}}+\underbrace{[\phi^T(k)\theta_0-\varphi^T(k)\theta^*]^2}_{\text{Bias error}} \tag{1.102}$$

It is clear that the MSE has three terms. The first term depends on the noise variance and cannot be reduced. The second term is the variance error which increases with the number of to-be-estimated parameters. The third term is the square of the bias error which decreases when number of parameters increases. As a result in every estimator, there is always a tradeoff between the bias error and the variance error which depends the number of estimated parameters. In other words, over parameterization will increase the variance error and under parameterization increases the bias error. So an optimal choice of the number of parameters can lead to the minimum of the mean square error.

Chapter 2

Nonparametric Identification Methods

2.1 Introduction

An LTI dynamical system can be described by a parametric (state-space or transfer function) model or a nonparametric (step response, frequency response) model. In this chapter some nonparametric approaches in system identification are discussed.

The nonparametric models are often the system's response to some particular excitation signals. They are usually represented by some curves in time or in frequency domain. For example, the response of a discrete-time system to a Kronecker impulse, called impulse response, can be recorded and used as a "model" for the system. Using the impulse response, the output of the system to any input signal can be computed using the convolution sum in (1.16). The frequency response of a system can be obtained by sinusoidal excitation signals and used as a "model" for controller design.

The nonparametric identification methods are simple and fast and need only a few a priori information about the system. Generally, no optimization is required and the measurement noise influences directly the model and it is not filtered out. They give some rough ideas about the order of the system, its bandwidth, time-delay and its dominant time-constant. These information can be used for parametric identification and model validation. The nonparametric models are also used for designing simple PID controllers using the Ziegler-Nichols algorithm or loop-shaping method. However, more advanced robust controller design methods based on nonparametric models have been also developed. In practice, a nonparametric model is recorded in a finite dimensional vector, so it can be considered as a parametric model with a large number of parameters.

In this chapter the following nonparametric identification methods are discussed:

Time-domain methods: The step or impulse response are identified by one of the following algorithms.

1. *Transient response:* The step or impulse response of a system are obtained by applying these signals on the real system and recording the output signal.
2. *Deconvolution algorithm:* The impulse response is computed for any excitation signal using the convolution relation. The step response can be computed by the integration of the impulse response.
3. *Correlation approach:* The impulse response is computed from an experiment in which the system is excited by a random signal, e.g. a white noise.

Frequency-domain methods: The frequency response (magnitude and phase) are identified by one of the following algorithms.

1. *Frequency analysis:* The system is excited by a sinusoidal signal, which leads, in steady state, to a sinusoidal signal at the output. The change in amplitude and phase give the frequency response of the system at the frequency of excitation signal.
2. *Fourier analysis:* The frequency response is obtained by dividing the Fourier transform of the output signal to that of the input signal, which is a sweep (chirp) signal or a pseudo-random periodic signal.
3. *Spectral analysis:* The system is excited by a random signal and the frequency response is obtained by dividing the cross-spectrum of output and input to the input spectrum.

The correlation approach and the spectral analysis use random signals for excitation and a stochastic approach to reduce the noise effects. The other approaches are deterministic and the noise effect can be reduced by averaging. In all methods linear time-invariant systems are identified so it is important that the initial conditions be zero and only the variations around an operating (or equilibrium) point be considered.

2.2 Time-Domain Methods

A dynamical system is completely characterized by its step or impulse response. Note that the impulse response is the derivative of the step response and includes all information about the system such that the output of the system to any input can be computed based on a known impulse response using the convolution relation.

2.2.1 Transient response

The easiest and most straightforward way to find the step and impulse response of a system is to apply a step signal or an impulse signal to the plant and record the output data. There are, however, some practical issues that should be taken into account for a correct data acquisition, which are discussed below.

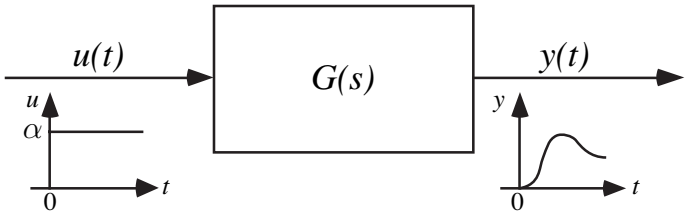


Figure 2.1: Step response experiment

Step response: The step response is recorded by applying a step signal to the system (see Fig. 2.1) under the following conditions:

- The system should be in a stationary state.
- In this state, the noise variance should be estimated.
- The amplitude of the step signal should be around 10 times the standard deviation of measurement noise.
- The amplitude of the step signal, α, should not be too high, in order to not excite the system's nonlinearities.
- The sampling period should be as small as possible.

By observing the step response, the following information can be extracted:

- The presence of a pure time delay. The delay can be estimated as the time between the application of the step signal and the first time that the step response leaves a confidence interval around zero. This confidence interval depends on the probability density function of noise. For example, for a Gaussian noise, the confidence interval for 95% of probability is equal to $\pm 2\sigma_e$, where σ_e is the standard deviation of noise.
- The steady state gain of the system. It is equal to the ratio between the final value of the step response, K, and the amplitude of the step signal α.
- The minimum order of the system. For example, a step response with overshoot represents a minimum order of two.

Impulse response: The impulse response of a continuous time model cannot be obtained experimentally, because a Dirac impulse signal cannot be realized physically. A solution to this problem is to derivate the step response if the measurement noise is not large.

The impulse response of a discrete-time model can be obtained by applying a Kronecker delta signal on the sampled system under the same condition for the step response. A Kronecker delta has an amplitude of 1 and a duration of one sampling period. Note that the Kronecker delta is not a sampled version of the Dirac impulse.

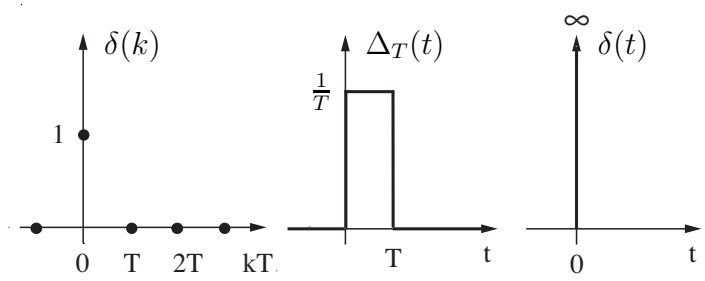


Figure 2.2: Relation between Kronecker and Dirac impulse

As a result, the discrete impulse response, $g(k)$, is not a sampled version of the continuous impulse response $g(t)$, either. We can consider the Dirac impulse as the limit of a pulse of duration T and amplitude $1/T$, say $\Delta_T(t)$, when T goes to zero (see Fig. 2.2). In this case, the Kronecker delta will be T times the sampled version of $\Delta_T(t)$. Therefore, the discrete impulse response is approximatively T times the sampled version of continuous impulse response, i.e. $g(k) \approx Tg(t)|_{kT}$.

In order to reduce the effect of the measurement noise, a Kronecker delta with a higher amplitude, say $\alpha\delta(k)$, should be applied to the real system. However, large amplitude signals may excite the nonlinear behavior of the system. The impulse response of a discrete-time linear system can be obtained from its step response, $y(k)$, by:

$$g(k) = \frac{1}{\alpha}[y(k) - y(k-1)] \tag{2.1}$$

where α is the amplitude of the step signal.

In a more general way, the impulse response can be computed from any input/output data using the numerical deconvolution technique which will be detailed in the next section.

2.2.2 Numerical deconvolution

The numerical deconvolution method consists of identifying the discrete impulse response of a linear system from the input output measurements using the convolution relation:

$$y(k) = \sum_{j=0}^{k} g(j)u(k-j) \qquad k = 0, 1, 2, \ldots \tag{2.2}$$

This equation can be rewritten for N measurements ($k = 0, 1, ...N-1$) in a matricial form as:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} u(0) & 0 & \cdots & 0 \\ u(1) & u(0) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ u(N-1) & u(N-2) & \cdots & u(0) \end{bmatrix} \begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(N-1) \end{bmatrix} \tag{2.3}$$

or

$$Y = U\Theta \tag{2.4}$$

with the evident definition of Y, U and Θ. From this equation, the impulse response can be computed by: $\Theta = U^{-1}Y$. However, U which is an asymmetric Toeplitz matrix is, in general, ill-conditioned and close to singular such that Θ cannot be computed. There are different solutions to this problem that are discussed here.

Recursive formulation: The inversion of U can be avoided using a recursive formulation of the solution as follows (for $u(0) \neq 0$):

$$g(k) = \frac{1}{u(0)} \left[y(k) - \sum_{j=0}^{k-1} g(j)u(k-j) \right] \qquad k = 0, 1, \ldots N-1 \tag{2.5}$$

But, even this formulation is too sensitive to the numerical errors and often diverges. One of the non diverging solution concerns the unit step signal:

$$u(k) = 1 \qquad k = 0, 1, 2, \ldots \tag{2.6}$$

In this case, (2.5) leads to

$$g(k) = y(k) - \sum_{j=0}^{k-1} g(j) \tag{2.7}$$

and evaluating (2.5) for instant $(k-1)$ gives:

$$y(k-1) = \sum_{j=0}^{k-1} g(j) \qquad k = 0, 1, 2, \ldots \tag{2.8}$$

Combining (2.7) and (2.8) gives:

$$g(k) = y(k) - y(k-1) \tag{2.9}$$

Finite impulse response: The numerical problem of this approach can be fixed if we assume that $g(k) = 0$ for $k \geq K$ (i.e. the impulse response is finite). This

assumption is realistic for stable systems, because their impulse response converges asymptotically to zero. In this case, (2.3) can be rewritten as:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} u(0) & 0 & \cdots & 0 \\ u(1) & u(0) & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ u(N-1) & u(N-2) & \cdots & u(N-K) \end{bmatrix} \begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(K-1) \end{bmatrix}$$

or $Y = U_K \Theta_K$. In fact, U_K and Θ_K are the truncated version of U and Θ, where the last $N-K$ columns of U and the last $N-K$ raws of Θ are removed. Now, the number of equations, N, is more than the number of unknowns K (i.e. U_K is a $N \times K$ dimensional matrix). This problem can be solved in the least squares sense by computing the pseudo inverse of U_K:

$$\Theta_K = U_K^{\dagger} Y = (U_K^T U_K)^{-1} U_K^T Y \tag{2.10}$$

Alternative solution with linear regression: The same results can be obtained using the convolution relation:

$$y(k) = \sum_{j=0}^{K-1} g(j)u(k-j) + \varepsilon(k) \qquad k = 0, 1, \ldots, N-1 \tag{2.11}$$

where $\varepsilon(k)$ is an error signal. The above equation can be written in a linear regression form by:

$$y(k) = \phi^T(k)\Theta_K + \varepsilon(k) \tag{2.12}$$

where

$$\phi^T(k) = [u(k)\,,\, u(k-1)\,,\, \cdots\,,\, u(k-K+1)] \tag{2.13}$$

$$\Theta_K^T = [g(0)\,,\, g(1)\,,\, \cdots\,,\, g(K-1)] \tag{2.14}$$

Then the impulse response coefficients in Θ_K can be identified in the least squares sense by minimizing the sum of squares of the error signal:

$$J(\Theta_K) = \sum_{k=0}^{N-1} \varepsilon^2(k) \tag{2.15}$$

The solution to this problem is given by:

$$\Theta_K = (\Phi^T \Phi)^{-1} \Phi^T Y \tag{2.16}$$

where

$$\Phi = \begin{bmatrix} \phi^T(0) \\ \phi^T(1) \\ \vdots \\ \phi^T(K-1) \end{bmatrix} = \begin{bmatrix} u(0) & u(-1) & \cdots & u(1-K) \\ u(1) & u(0) & \cdots & u(2-K) \\ \vdots & \vdots & \cdots & \vdots \\ u(N-1) & u(N-2) & \cdots & u(N-K) \end{bmatrix}$$

Note that we assume $u(k) = 0$ for $k < 0$, so the matrix Φ is the same as U_K and $(\Phi^T\Phi)^{-1}\Phi^T$ is nothing else than the pseudo inverse of U_K.

Regularization: To apply the FIR solution, we require information about the length of the impulse response K. However, if this information is not available, we can use the regularization technique, which arises from the principle of bias-variance trade-off in optimization problems. In cases where U is close to being singular, the norm of $\Theta = U^{-1}Y$ may approach infinity, which can be avoided by minimizing the norm of Θ in the least squares solution. We can achieve this by modifying the least squares criterion as follows:

$$J(\Theta) = \mathcal{E}^T\mathcal{E} + \lambda\Theta^T\Theta \tag{2.17}$$

where $\mathcal{E} = Y - U\Theta$, and λ is a weighting factor for bias-variance trade-off. The partial derivative of J with respect to Θ is:

$$\frac{\partial J}{\partial \Theta} = -2U^TY + 2U^TU\Theta + 2\lambda\Theta = 0 \tag{2.18}$$

that leads to the following solution:

$$\Theta = (U^TU + \lambda I)^{-1}U^TY \tag{2.19}$$

By setting a non-zero value for λ, we make $U^TU + \lambda I$ a regular matrix, which is invertible and leads to a finite norm of Θ. To determine a suitable value for λ, we can use trial and error, but having an idea about the length of the impulse response can help in choosing an appropriate value for λ.

2.2.3 Correlation approach

The numerical deconvolution method is a deterministic approach and therefore nothing can be said on the quality of the estimates in (2.16) and the effects of the measurement noise. The correlation approach is a stochastic method for estimating the impulse response of a system in the presence of measurement noise. The correlation method is based on the autocorrelation and cross-correlation functions of random signals which will be defined in the sequel.

Cross-correlation function: The autocorrelation function for discrete random processes was defined in (1.40). In the same way, the cross-correlation function of two ergodic signals $y(k)$ and $u(k)$ is defined as:

$$R_{yu}(h) = \mathbb{E}\{y(k)u(k-h)\} = \lim_{N\to\infty}\frac{1}{N}\sum_{k=0}^{N-1} y(k)u(k-h) \tag{2.20}$$

If $u(k)$ and $y(k)$ are independent random processes, we have:

$$R_{yu}(h) = \mathbb{E}\{y(k)\}\mathbb{E}\{u(k-h)\} \tag{2.21}$$

and moreover if at least one of $y(k)$ and $u(k)$ has zero-mean value, we have:

$$R_{yu}(h) = \mathbb{E}\{y(k)u(k-h)\} = 0 \qquad \forall h \tag{2.22}$$

Some properties of the correlation functions are:

$$(1)R_{uu}(h) = R_{uu}(-h) \qquad (2)R_{yu}(h) = R_{uy}(-h) \qquad (3)|R_{uu}(h)| \leq R_{uu}(0)$$

The first two properties can be proved by defining $k' = k - h$:

$$\begin{aligned} R_{uu}(h) &= \mathbb{E}\{u(k)u(k-h)\} = \mathbb{E}\{u(k'+h)u(k')\} = \mathbb{E}\{u(k')u(k'-(-h))\} \\ &= R_{uu}(-h) \\ R_{uy}(h) &= \mathbb{E}\{u(k)y(k-h)\} = \mathbb{E}\{u(k'+h)y(k')\} = \mathbb{E}\{y(k')u(k'-(-h))\} \\ &= R_{yu}(-h) \end{aligned}$$

The last property can be proved using the fact that $\mathbb{E}\{[u(k) \pm u(k-h)]^2\} \geq 0$

$$\begin{aligned} &\Rightarrow \quad \mathbb{E}\{u^2(k)\} \pm 2\mathbb{E}\{u(k)u(k-h)\} + \mathbb{E}\{u^2(k-h)\} \geq 0 \\ &\Rightarrow \quad R_{uu}(0) \pm 2R_{uu}(h) + R_{uu}(0) \geq 0 \\ &\Rightarrow \quad -R_{uu}(0) \leq R_{uu}(h) \leq R_{uu}(0) \\ &\Rightarrow \quad |R_{uu}(h)| \leq R_{uu}(0) \end{aligned}$$

Principle: The output $y(k)$ of an LTI discrete-time system, in the presence of a zero-mean random disturbance $d(k)$, can be written as:

$$y(k) = \sum_{j=0}^{\infty} g(j)u(k-j) + d(k) = g(k) * u(k) + d(k) \tag{2.23}$$

Suppose that $u(k)$ is a stationary random process independent of $d(k)$. The expectation of the above equation after multiplication of both sides by $u(k-h)$ leads to:

$$\mathbb{E}\{y(k)u(k-h)\} = \sum_{j=0}^{\infty} g(j)\mathbb{E}\{u(k-j)u(k-h)\} + \mathbb{E}\{d(k)u(k-h)\} \tag{2.24}$$

If we replace the expected values with the corresponding correlation functions, we obtain:

$$R_{yu}(h) = \sum_{j=0}^{\infty} g(j)R_{uu}(h-j) + R_{du}(h) \tag{2.25}$$

Since $u(k)$ and $d(k)$ are independent and $d(k)$ is zero-mean, we have:

$$R_{du}(h) = 0 \qquad \forall h \tag{2.26}$$

Therefore, we obtain:

$$R_{yu}(h) = \sum_{j=0}^{\infty} g(j)R_{uu}(h-j) = g(h) * R_{uu}(h) \tag{2.27}$$

If we compare the above convolution relation with that in (2.23), we observe that the pair $[y(k)\,,\,u(k)]$ is replaced with the pair $[R_{yu}(h)\,,\,R_{uu}(h)]$ and that the effect of disturbance has been completely removed. However, the correlation functions are not measured and should be computed based on an infinite number of data (which are not available).

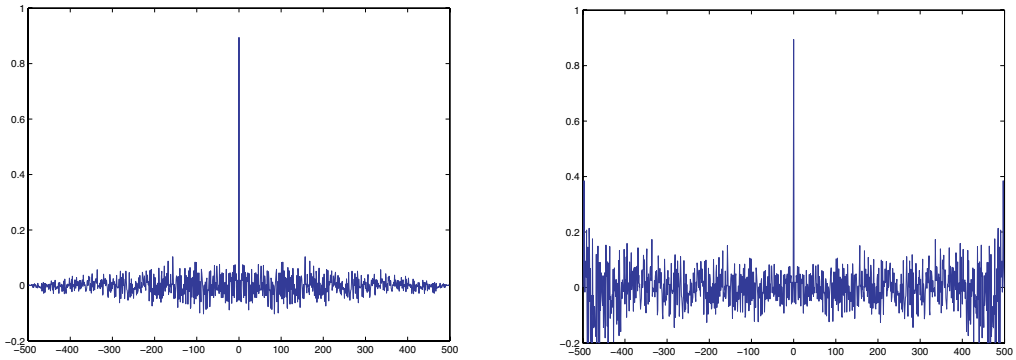


Figure 2.3: Biased (a) and unbiased (b) estimates of a white noise of length $N = 500$

Estimation of the autocorrelation function: The autocorrelation function can be estimated if the number of data N is sufficiently large. In this case, we have:

$$\hat{R}_{uu}(h) = \frac{1}{N}\sum_{k=h}^{N-1} u(k)u(k-h) \quad , \quad \hat{R}_{uu}(-h) = \hat{R}_{uu}(h) \quad , \quad h = 0, 1, 2, \dots \tag{2.28}$$

This estimation of the autocorrelation function is called *biased* because its expectation is not equal to the true value, i.e.:

$$\mathbb{E}\{\hat{R}_{uu}(h)\} = \frac{1}{N}\sum_{k=h}^{N-1} \mathbb{E}\{u(k)u(k-h)\} = \frac{N-h}{N} R_{uu}(h) \neq R_{uu}(h) \quad \forall h \tag{2.29}$$

The bias is higher for larger values of h. An *unbiased* estimate can be obtained by:

$$\hat{R}_{uu}(h) = \frac{1}{N-h}\sum_{k=h}^{N-1} u(k)u(k-h) \quad , \quad \hat{R}_{uu}(-h) = \hat{R}_{uu}(h) \quad , \quad h = 0, 1, 2, \dots \tag{2.30}$$

Although this estimate is unbiased, it has large variance for small $N - h$ (large h), because in this case the estimation of the autocorrelation function is based on averaging on a few number of data.

Example 2.1. For a finite number of data, the autocorrelation function of a zero-mean white noise can be estimated using (2.28) or (2.30) for a biased or unbiased estimation, respectively. Figure 2.3 shows these estimates for a realization of white noise with $N = 500$. It can be seen that the unbiased estimate has a larger variance for large value of $|h|$. The biased estimate always converges to zero for large $|h|$, which fortunately coincides with the true value for a zero-mean white noise.

Estimation of the cross-correlation function: In the same way, a biased estimate of the cross-correlation function for two ergodic signals $y(k)$ and $u(k)$ can be obtained

by:

$$\hat{R}_{yu}(h) = \frac{1}{N} \sum_{k=\max(h,0)}^{N-1+\min(h,0)} y(k)u(k-h) \quad , \quad h = 0, \pm 1, \pm 2, \ldots \tag{2.31}$$

and an unbiased estimate by

$$\hat{R}_{yu}(h) = \frac{1}{N-|h|} \sum_{k=\max(h,0)}^{N-1+\min(h,0)} y(k)u(k-h) \quad , \quad h = 0, \pm 1, \pm 2, \ldots \tag{2.32}$$

Computing the impulse response: The impulse response now can be computed by numerical deconvolution of (2.27), which involves an infinite number of linear equations. This problem is greatly simplified if we use a white noise as input. In this case, we have $R_{uu}(h) = R_{uu}(0)\delta(h)$, therefore:

$$R_{yu}(h) = [g(h) * \delta(h)]R_{uu}(0) \qquad \Rightarrow \qquad g(h) = \frac{R_{yu}(h)}{R_{uu}(0)} \tag{2.33}$$

The cross-correlation function $R_{yu}(h)$ can be estimated using a finite number of data by (2.31) or (2.32) for a biased or unbiased estimate, respectively.

If $u(k)$ is not white, then the numerical deconvolution method is applied to the correlation functions. For a finite number of data N, if we assume that $g(h) = 0$ for $h \geq K$, i.e. the impulse response has a finite length, we have:

$$R_{yu}(h) = \sum_{j=0}^{K-1} g(j)R_{uu}(h-j) \tag{2.34}$$

For $h = 0, 1, \ldots, K-1$, the following system of linear equation is obtained:

$$\begin{bmatrix} \hat{R}_{yu}(0) \\ \hat{R}_{yu}(1) \\ \vdots \\ \hat{R}_{yu}(K-1) \end{bmatrix} = \begin{bmatrix} \hat{R}_{uu}(0) & \hat{R}_{uu}(1) & \cdots & \hat{R}_{uu}(K-1) \\ \hat{R}_{uu}(1) & \hat{R}_{uu}(0) & \cdots & \hat{R}_{uu}(K-2) \\ \vdots & \vdots & & \vdots \\ \hat{R}_{uu}(K-1) & \hat{R}_{uu}(K-2) & \cdots & \hat{R}_{uu}(0) \end{bmatrix} \begin{bmatrix} \hat{g}(0) \\ \hat{g}(1) \\ \vdots \\ \hat{g}(K-1) \end{bmatrix} \tag{2.35}$$

The impulse response is computed by multiplying the vector of the cross-correlation functions and the inverse of the matrix of the input autocorrelation functions.

$$\begin{bmatrix} \hat{g}(0) \\ \hat{g}(1) \\ \vdots \\ \hat{g}(K-1) \end{bmatrix} = \begin{bmatrix} \hat{R}_{uu}(0) & \hat{R}_{uu}(1) & \cdots & \hat{R}_{uu}(K-1) \\ \hat{R}_{uu}(1) & \hat{R}_{uu}(0) & \cdots & \hat{R}_{uu}(K-2) \\ \vdots & \vdots & & \vdots \\ \hat{R}_{uu}(K-1) & \hat{R}_{uu}(K-2) & \cdots & \hat{R}_{uu}(0) \end{bmatrix}^{-1} \begin{bmatrix} \hat{R}_{yu}(0) \\ \hat{R}_{yu}(1) \\ \vdots \\ \hat{R}_{yu}(K-1) \end{bmatrix} \tag{2.36}$$

The quality of the impulse response estimates is directly related to the quality of the correlation function estimates and improves by increasing the number of data.

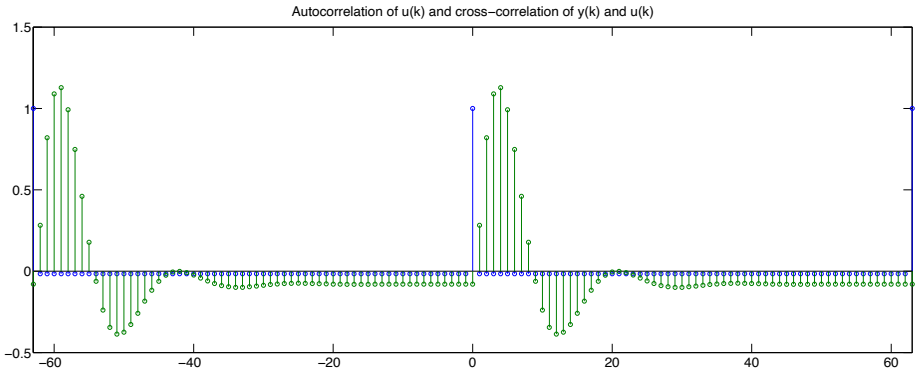


Figure 2.4: $R_{uu}(h)$ and $R_{yu}(h)$ for a noiseless simulation of a linear system with a PRBS as input

Correlation approach with PRBS excitation: In practice, a Pseudo Random Binary Sequence (PRBS) can be used as input instead of a random white noise. The main reason is that a PRBS has an autocorrelation function which is very close to that of a white noise except for the fact that it is periodic. However, the periodicity of the PRBS helps computing the exact value of its autocorrelation function, which can be used for impulse response computation. In fact, if a linear system is excited by a periodic signal, the deterministic part of the output will become periodic after a transient time. The exact value of the autocorrelation of a PRBS has already been computed in (1.50). The cross-correlation of $y(k)$ and $u(k)$ can be computed by:

$$R_{yu}(h) = \frac{1}{M} \sum_{k=1}^{M-1} y(k)u(k-h) \tag{2.37}$$

where $u(k-h) = u(k+M-h)$ because of the periodicity of $u(k)$. It should be mentioned that $R_{yu}(h)$ is not the exact value because $y(k)$ is affected by noise and is not completely periodic (even after the transient time). However, as the noise is independent from the PRBS, its effect is significantly reduced if M is sufficiently large. Figure 2.4 shows this estimate together with $R_{uu}(h)$ in simulation for a noiseless linear system. It can be seen that $R_{uu}(h)$ and $R_{yu}(h)$ are both periodic with period M. Since, the autocorrelation of a PRBS is very close to that of a white noise, R_{yu} is a good approximation of the impulse response (by a constant factor). From this figure, it is clear that the settling time of the impulse response should be less than MT to avoid the overlapping of the cross-correlation function in successive periods. It is also clear that the small non-zero values of the autocorrelation function of PRBS for $h \neq 0$ leads to an offset error in the estimate of the impulse response. This error can be removed if the numerical deconvolution in (2.36) is used for the impulse response estimation.

2.2.4 Identifying parametric models from step response

For simple systems, the parameters of first and second order continuous-time models can be identified from the step response.

First-order model: Consider a first order model

$$G(s) = \frac{\gamma}{\tau s + 1} \tag{2.38}$$

with the step response to a step of amplitude α given by

$$y(t) = K(1 - e^{-t/\tau}) \tag{2.39}$$

and shown in Fig. 2.5. From the step response, the steady-state gain $\gamma = K/\alpha$

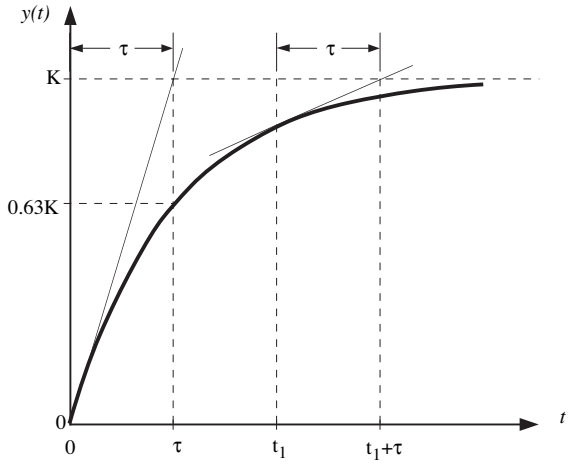


Figure 2.5: Step response of a first order system

and the time constant τ can be easily computed. In practice, in the presence of measurement noise, the exact value of τ is difficult to estimate. With this method, the time constants of higher order models cannot be estimated.

First-order model with time-delay: A first order model with time delay is given by:

$$G(s) = \frac{\gamma e^{-\theta s}}{\tau s + 1} \tag{2.40}$$

The parameters of this model can be identified from the step response of a high-order damped system using the Ziegler-Nichols method. The final value of the step response can be used to find the steady-state gain $\gamma = K/\alpha$. A tangent line with maximum slop of the step response can help identifying the two other parameters based on the scheme of Fig. 2.6. Although, it is a very rough estimate of the parameters, it is often used in industry for the tuning of PID controllers.

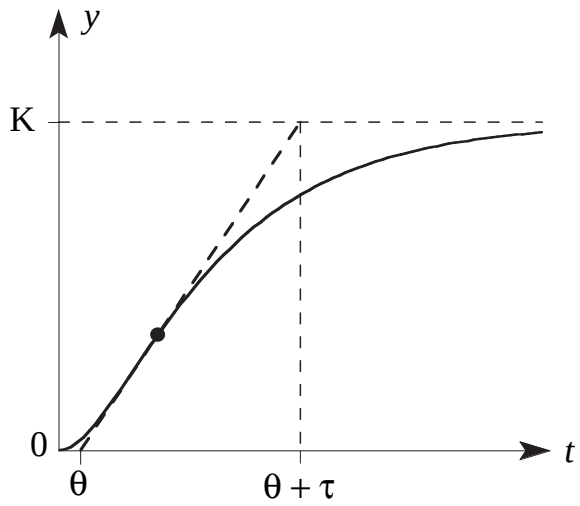


Figure 2.6: Approximation of high order models by a first order model with delay

Second-order model: A second-order model without zero is typically represented by:

$$G(s) = \frac{\gamma\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.41}$$

where γ is the steady-state gain, ζ is the damping factor and ω_n the natural frequency. If the damping factor is less than 1 ($0 \leq \zeta < 1$), the system is called under-damped. The step response to a step of amplitude α, shown in Figure 2.7, in this case will be oscillatory and given by:

$$y(t) = K - \frac{K}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega t + \theta) \qquad t \geq 0 \tag{2.42}$$

with $\omega = \omega_n\sqrt{1-\zeta^2}$ and $\theta = \arccos\zeta$. One particular point that can be easily measured on the step response $y(t)$, is its first maximum at time t_p. The theoretical values for t_p and $y(t_p)$ can be computed by exploiting the first derivative of $y(t)$ which is equal to the impulse response of $G(s)$:

$$\dot{y}(t) = \frac{K\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin\omega t \qquad t \geq 0 \tag{2.43}$$

The first maximum is obtained for $\omega t_p = \pi$. Therefore,

$$y(t_p) = K + Ke^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{2.44}$$

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{2.45}$$

The following steps can be performed to identify the parameters of a second-order model without zero from an oscillatory step response:

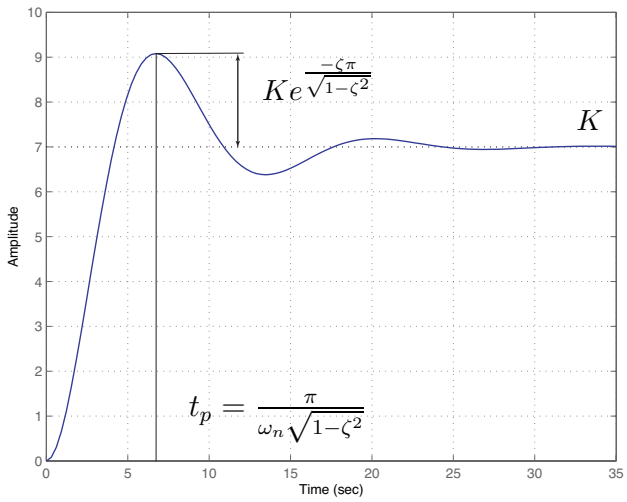


Figure 2.7: Step response of a second-order system without zero

1. Measure the final value of the step response to compute the steady-state gain $\gamma = K/\alpha$.

2. Measure the peak value of the step response, $y(t_p)$, and compute the damping factor ζ from (2.44).

3. Measure the peak time t_p and compute the natural frequency ω_n from (2.45).

2.3 Frequency-Domain Methods

A linear time invariant system can be represented in the frequency domain. For a continuous-time model, $G(s)$, the frequency response is obtained by replacing s with $j\omega$ in $G(s)$. The frequency response $G(j\omega)$ is an infinite dimensional vector of complex values. The magnitude and the phase can be plotted in two different diagrams which are called the Bode diagrams. The real and the imaginary part of frequency response $G(j\omega)$ can be drawn in the complex plane, which is known as the Nyquist diagram. For a discrete-time model, $G(z)$, the frequency response is obtained by replacing z with $e^{j\omega T}$, where T is the sampling period. If the sampling period is correctly chosen and small enough, then $G(j\omega)$ will be very close to the frequency response $G(e^{j\omega})$ of its discrete-time converted model. Note that the sampling period is dropped out in the argument of $G(e^{j\omega})$ to simplify the notation.

There are different methods to find the frequency-domain representation of a system depending on the type of inputs and the way that the measurements are treated.

1. Frequency analysis: The input signal is a sinusoidal signal, which leads in steady state to a sinusoidal output. The magnitude and the phase of the

output signal are measured and used to compute one point in the frequency response of the system.

2. Fourier analysis: A deterministic signal, preferably periodic, that excites a large number of frequencies (sum of sinusoids or PRBS) is applied as input. The frequency response is obtained by computing the Fourier transform of input and output data.

3. Spectral analysis: The input and output of the system are considered as stochastic signals (input is a random process and the output is affected by measurement noise). Power spectral density functions are used to identify the frequency response of the system.

For convenience the continuous-time models are sometimes considered in the discussions and analysis when appropriate. The results can be applied to discrete-time models as well.

2.3.1 Frequency analysis

If the input signal is a sinusoid:

$$u(t) = A\sin(\omega t) \tag{2.46}$$

the output for an LTI system, in the *steady state*, will be also a sinusoid with the same frequency but different amplitude and phase:

$$y(t) = A'\sin(\omega t + \phi) = A|G(j\omega)|\sin(\omega t + \arg G(j\omega)) \tag{2.47}$$

Therefore, by exciting the system with different sinusoids and each time measuring the amplitude and phase of the output at steady state, the frequency response $G(j\omega)$ can be obtained for a number of frequencies by (see Fig. 2.8):

$$R_A(\omega) = \frac{A'}{A} = |G(j\omega)| \tag{2.48}$$

$$\varphi(\omega) = \arg[G(j\omega)] \tag{2.49}$$

and plotted in the Bode diagram.

This approach is very simple and intuitive and requires only a few a priori knowledge about the process. In fact, only the interesting frequency zone for excitation should be known. Usually, three or more frequency decades are considered and at each decade at least 10 sinusoids are applied. The amplitude of the sinusoids should be large enough with respect to the noise level and small enough to not excite the system's nonlinearities.

If a continuous-time sinusoid is applied, $G(j\omega)$ will be identified and there will be no sampling error. If a sampled sinusoid signal with a zero-order-hold is applied, $G(e^{j\omega})$ will be identified. The sampling error, the difference between $G(j\omega)$ and $G(e^{j\omega})$, can be reduced by choosing the sampling time T as small as possible.

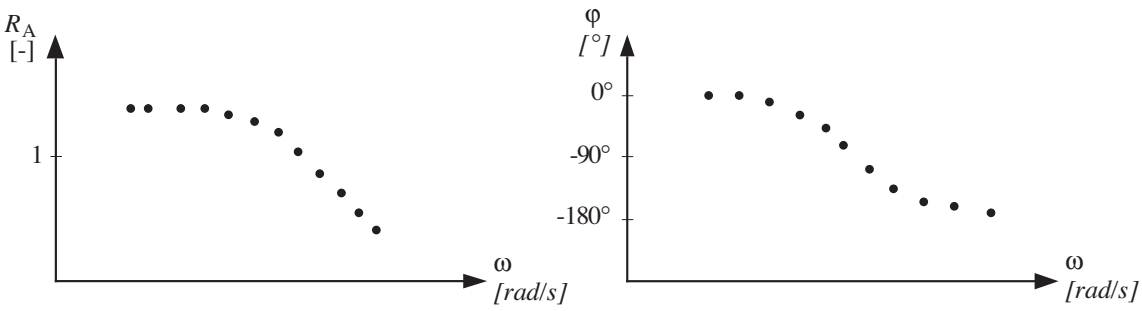


Figure 2.8: Experimental Bode diagram

If the noise level is high, the amplitude and phase measurements will be contaminated by measurement noise. The noise effect can be reduced by averaging over several periods.

The main drawback of the frequency analysis approach is the length of the experiments. For a good identification, a lot of frequency points are needed (40-50 experiments). Each experiment should be long enough such that the output attains the steady state (after about 5 times the dominant time-constant of the system). Moreover, several periods should be measured in order to reduce the noise effect. This type of experiments may be too long when exciting the system at low frequencies or in general for slow systems (chemical processes may have a time constant of about one hour).

2.3.2 Fourier analysis

The main idea of this approach is to apply a signal that contains several frequency components, so to excite several frequencies at one experiment (see Fig. 2.9). Then, using the Fourier transform, the frequency response is identified.

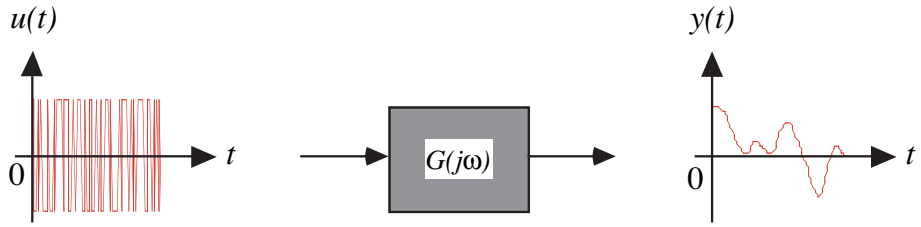


Figure 2.9: Frequency response identification using Fourier analysis

The theoretical ground of the approach is the input/output model of the system based on the convolution relation. For LTI continuous-time systems, we have:

$$y(t) = \int_0^\infty g(\tau)u(t-\tau)d\tau = g(t) * u(t) \tag{2.50}$$

Then by taking the Fourier transform from both sides, we obtain:

$$Y(j\omega) = G(j\omega)U(j\omega) \qquad \Rightarrow \qquad G(j\omega) = \frac{Y(j\omega)}{U(j\omega)} \tag{2.51}$$

where $U(j\omega)$ and $Y(j\omega)$ are respectively the Fourier transform of $u(t)$ and $y(t)$ are defined by:

$$U(j\omega) = \int_{-\infty}^{\infty} u(t)e^{-j\omega t}dt \tag{2.52}$$

$$Y(j\omega) = \int_{-\infty}^{\infty} y(t)e^{-j\omega t}dt \tag{2.53}$$

and $G(j\omega)$ is the Fourier transform of the impulse response $g(t)$. In practice, the sampled input and output data are used in the analysis, so the Fourier transform of discrete signals is used to identify $G(e^{j\omega})$. For the LTI discrete-time systems, we have:

$$y(k) = \sum_{j=0}^{\infty} g(j)u(k-j) = g(k) * u(k) \tag{2.54}$$

Taking the Fourier transform from the both sides gives:

$$Y(e^{j\omega}) = G(e^{j\omega})U(e^{j\omega}) \qquad \Rightarrow \qquad G(e^{j\omega}) = \frac{Y(e^{j\omega})}{U(e^{j\omega})} \tag{2.55}$$

where

$$U(e^{j\omega}) = \sum_{k=-\infty}^{\infty} u(k)e^{-jk\omega T} \tag{2.56}$$

$$Y(e^{j\omega}) = \sum_{k=-\infty}^{\infty} y(k)e^{-jk\omega T} \tag{2.57}$$

There are three major errors when computing the frequency response of a system using the sampled data:

Sampling error: Since we identify $G(e^{j\omega})$ instead of $G(j\omega)$, there will be a sampling error that can be reduced by decreasing the sampling time.

Measurement error: The measured output is usually affected by random noise and disturbances. This error affects the accuracy of the frequency response estimate.

Truncation error: The Fourier transforms are defined over a sum on infinite number of data, while in practice, always a finite number of data is available. Computing the Fourier transform over truncated data leads to the truncation error.

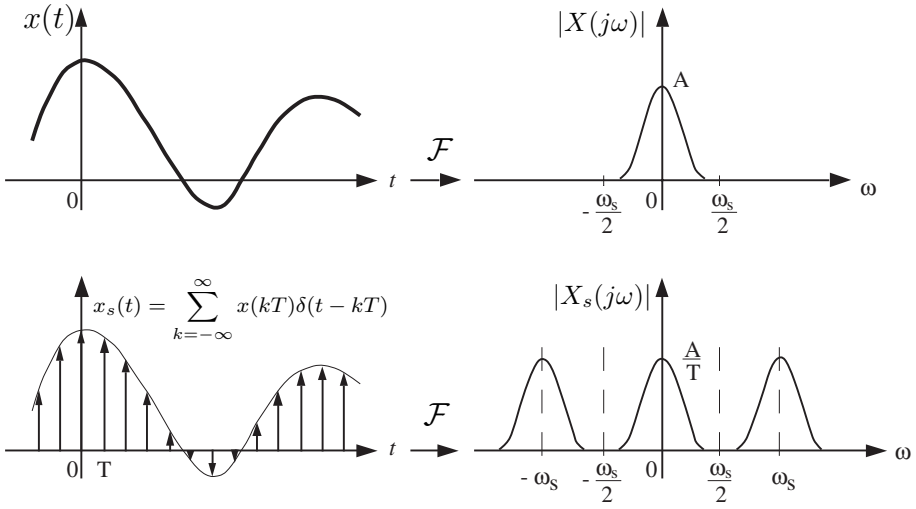


Figure 2.10: Fourier transform of continuous and sampled signals

Sampling error

Consider a continuous-time signal $x(t)$ and its sampled version as a function of t:

$$x_s(t) \equiv \sum_{k=-\infty}^{\infty} x(kT)\delta(t-kT) \tag{2.58}$$

where T is the sampling period and $\delta(t)$ is the Dirac function. The Fourier transform of the continuous-time signal $x_s(t)$ is given by:

$$X_s(j\omega) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(j\omega - \frac{2n\pi}{T}) \tag{2.59}$$

It can be seen that $X_s(j\omega)$ is a periodic function with the sampling frequency $\omega_s = 2\pi/T$ as period (see Fig. 2.10). On the other hand, if the condition of the Shannon's sampling theorem is met, i.e. $|X(j\omega)|$ is zero for $|\omega| > \omega_s/2$, then, inside the interval $[-\omega_s/2\,,\,\omega_s/2]$, $X_s(j\omega)$ has exactly the same shape as $X(j\omega)$ but multiplied by $1/T$. However, since in practice the conditions on the Shannon theorem is almost always violated by the output signal, there will be some aliasing effects that generate an error in the frequency response estimate. This error can be reduced by increasing the sampling frequency.

Note that $X_s(j\omega)$ is equal to $X(e^{j\omega})$, the discrete Fourier transform of the sampled signal $x(k)$.

Measurement error

The output of an LTI discrete-time system in the presence of disturbances and measurement noise is given by:

$$y(k) = \sum_{j=0}^{\infty} g(j)u(k-j) + d(k) \tag{2.60}$$

Taking the Fourier transform from the both sides gives:

$$Y(e^{j\omega}) = G(e^{j\omega})U(e^{j\omega}) + D(e^{j\omega}) \quad \Rightarrow \quad \frac{Y(e^{j\omega})}{U(e^{j\omega})} = G(e^{j\omega}) + \frac{D(e^{j\omega})}{U(e^{j\omega})} \tag{2.61}$$

where

$$D(e^{j\omega}) = \sum_{k=-\infty}^{\infty} d(k)e^{-jk\omega T} \tag{2.62}$$

and the frequency function estimate is

$$\hat{G}(e^{j\omega}) = \frac{Y(e^{j\omega})}{U(e^{j\omega})} \tag{2.63}$$

The error in the frequency function estimate is:

$$\hat{G}(e^{j\omega}) - G(e^{j\omega}) = \frac{D(e^{j\omega})}{U(e^{j\omega})} \tag{2.64}$$

It is clear that $D(e^{j\omega})$ is a random variable and its expectation is zero if $\mathbb{E}\{d(k)\} = 0$:

$$\mathbb{E}\{D(e^{j\omega})\} = \sum_{k=\infty}^{\infty} \mathbb{E}\{d(k)\}e^{-jk\omega T} = 0 \tag{2.65}$$

Then, the frequency response estimate in (2.63) is unbiased:

$$\mathbb{E}\{\hat{G}(e^{j\omega}) - G(e^{j\omega})\} = \frac{\mathbb{E}\{D(e^{j\omega})\}}{U(e^{j\omega})} = 0 \tag{2.66}$$

Quality of the estimates: The variance of the frequency function estimate is obtained by:

$$\text{var}[\hat{G}(e^{j\omega})] = \frac{\mathbb{E}\{D(e^{j\omega})D(e^{-j\omega})\}}{U(e^{j\omega})U(e^{-j\omega})} = \frac{\mathbb{E}\{|D(e^{j\omega})|^2\}}{|U(e^{j\omega})|^2} = \frac{\Phi_{dd}(\omega)}{\Phi_{uu}(\omega)} \tag{2.67}$$

The variance is proportional to the disturbance spectrum and inversely proportional to the spectrum of the input signal. Therefore, in order to reduce the effect of the measurement noise, the spectrum of the input signal should be large in the region that the noise spectrum is high. Note that by increasing the number of data the variance will not be affected.

Truncation error

The main source of this error is that the Fourier transform should be applied on the truncated data because the whole data from $-\infty$ to ∞ is not available. There are three methods for removing or reducing this error that are discussed here.

Excitation with energy signals: In system identification, the input signal is a user choice. So it can be chosen as an energy signal or time limited signal such that $u(k) = 0$ for $k < 0$ and $k \geq M$. Therefore its Fourier transform can be computed exactly by:

$$U(e^{j\omega}) = \sum_{k=0}^{M-1} u(k)e^{-jk\omega T} \tag{2.68}$$

On the other hand, if the system to be identified is stable, $y(k)$ will converge to zero when $k \to \infty$. However, in practice, after a finite time (5 times the dominant time constant of the system) $y(k) \approx 0$ and its variation will be at the noise level. Assuming that $y(k) \approx 0$ for $k \geq N$, its Fourier transform can be well approximated using a finite data length. As a result, the data length $N > M$, should be considered for the input and output signal to take into account the transient time of the plant to attain the steady state.

Excitation with periodic signals: The Fourier transform of the periodic signals is a Fourier series that can be computed from the information of just one period. If $u(k)$ is periodic with a period of M samples, then the coefficients of its Fourier series are computed by:

$$U(e^{j\omega_n}) = \frac{1}{M}\sum_{k=0}^{M-1} u(k)e^{-jk\omega_n T} \qquad \omega_n = \frac{2\pi n}{MT} \qquad n = 0, 1, \ldots, M-1 \tag{2.69}$$

On the other hand, the output of the system (ignoring the noise) will be periodic in the steady state. So it's Fourier series can be computed as:

$$Y(e^{j\omega_n}) = \frac{1}{M}\sum_{k=L}^{L+M-1} y(k)e^{-jk\omega_n T} \qquad \omega_n = \frac{2\pi n}{MT} \qquad n = 0, 1, \ldots, M-1 \tag{2.70}$$

where L is the settling time of the discrete-time system.

The measurement errors can be significantly reduced when using periodic signals by averaging the Fourier series coefficients in different periods. Suppose that p periods are available and $Y_i(e^{j\omega_n})$ corresponds to the Fourier series of the ith period, then:

$$G(e^{j\omega_n}) = \frac{\frac{1}{p}\sum_{i=1}^{p} Y_i(e^{j\omega_n})}{U(e^{j\omega_n})} \qquad \omega_n = \frac{2\pi n}{MT} \qquad n = 0, 1, \ldots, M-1 \tag{2.71}$$

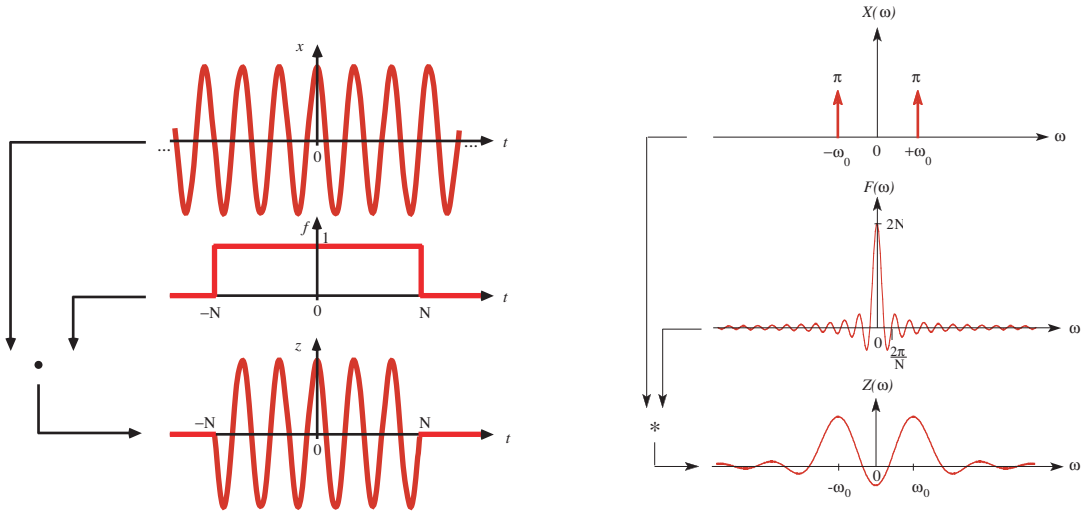


Figure 2.11: Relation between $x(t)$, $f(t)$ and $z(t) = x(t)f(t)$ and their Fourier transform

This way, the variance of the measurement error will be divided by the number of periods, p, used for averaging. Note that in open-loop identification $u(k)$, is not noisy and no averaging is needed. In a closed-loop operation, an averaging over $U(e^{j\omega_n})$ is also necessary.

Windowing: The truncation error can be reduced by the use of weighting functions in the time domain or windowing. In order to understand this method, consider for example the signal $x(t) = \cos(\omega_0 t)$ for $t \in (-\infty\,,\,\infty)$ and its Fourier transform given by:

$$X(\omega) = \pi[\delta(\omega + \omega_0) + \delta(\omega - \omega_0)] \tag{2.72}$$

The truncation of this signal in the interval $-N$ and N is equivalent to consider a new signal

$$z(t) = x(t)f(t) \tag{2.73}$$

where $f(t)$ represents a pulse signal shown in Fig. 2.11. It should be mentioned that the choice of a symmetric signal with respect to $t = 0$ is just for the simplicity of the presentation (the Fourier transform becomes a real valued function) and does not change the conclusions.

Since the Fourier transform of the product of two signals is equal to the convolution of their Fourier transform we have:

$$Z(\omega) = \mathcal{F}\{z(t)\} \equiv \int_{-\infty}^{\infty} z(t)e^{-j\omega t}dt = \int_{-N}^{N} x(t)e^{-j\omega t}dt \tag{2.74}$$

$$= \mathcal{F}\{x(t)f(t)\} = \frac{1}{2\pi}X(\omega) * F(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\nu)F(\omega - \nu)d\nu \tag{2.75}$$

The Fourier transform of the pulse signal is given by:

$$F(\omega) = \frac{2}{\omega}\sin(\omega N) = 2N\,\mathrm{sinc}(\omega N) \tag{2.76}$$

The Fourier transform of $x(t), f(t)$ and $z(t)$ are shown in Fig. 2.11. It is evident that by increasing the pulse width (the length of the window) to infinity, we obtain:

$$\lim_{N\to\infty} F(\omega) = \lim_{N\to\infty} \frac{2}{\omega}\sin(\omega N) = 2\pi\delta(\omega) \tag{2.77}$$

and (2.75) leads to $Z(\omega) = X(\omega)$, because the Dirac function is the neutral element for the convolution product. The Fourier transform of a rectangular window is far from the ideal case and generates two types of error. The first one is related to the width of the main lobe that leads to a smoothing effect and loosing the resolution of $Z(\omega)$ because of the convolution integral. The second error comes from the side lobes that bring some errors from other frequencies, which is called spectral leakage. The first error is commonly considered positively, because it reduces the measurement errors and smoothen the spectrum (with the price of loosing the resolution and accuracy). The second error, related to the amplitude of the side lobes, however should be reduced as much as possible. A rectangular window has a main lobe width of $2\pi/N$ and the amplitude of the second lobe is 21.7% of the amplitude of the main lobe. It is possible to modify the shape of the window such that the relative amplitude of the second lobe is reduced but the main lobe width is increased (there is always a trade-off). For example, consider the Hann window defined as:

$$f(t) = \begin{cases} 0.5\left(1+\cos\frac{\pi t}{N}\right) & \text{for} \quad t \in [-N\ ,\ N] \\ 0 & \text{elsewhere} \end{cases} \tag{2.78}$$

which is shown together with its Fourier transform in Fig. 2.12. This window has a main lobe width of $4\pi/N$ (twice that of rectangular window) but with a relative amplitude of the second lobe which is only 2.7%. Table 2.1 shows different windows of width $2N$, their Fourier transforms, the main lobe width (MLW) and the second lobe relative amplitude (SLA). It can be observed that the triangular, Hann and Hamming windows have all an MLW of twice that of the rectangular window but with significantly smaller SLA. Although the Hamming window has the smallest SLA, the amplitudes of side lobes in Hann window decrease faster.

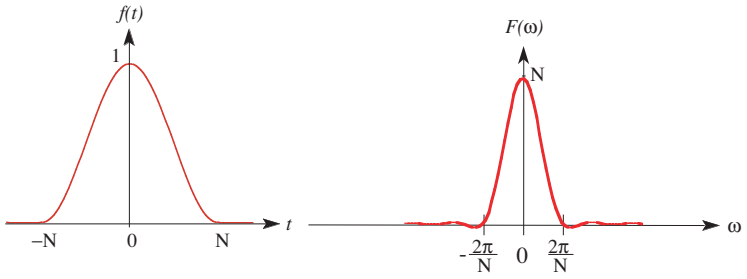


Figure 2.12: Hann window with its Fourier transform

Remark: The windowing method is used when there is no possibility to apply an energy signal or a periodic signal for which the truncation error is theoretically zero. In some applications, the identification data are taken from the normal operation of

Table 2.1: Time-domain windows of $2N$ width and their Fourier transforms (MLW: Main lobe width; SLA: Second lobe relative amplitude).

Window	Definition on $[-N\,,\,N]$	Fourier transform	MLW	SLA
Rectangular	1	$F(\omega) = 2N\text{sinc}(\omega N)$	$\frac{2\pi}{N}$	21.7%
Triangular	$1 - \frac{\lvert t\rvert}{N}$	$N\text{sinc}^2\left(\frac{\omega N}{2}\right)$	$\frac{4\pi}{N}$	4.7%
Hann	$0.5[1 + \cos\left(\frac{\pi t}{N}\right)]$	$\frac{1}{2}F(\omega) + \frac{1}{4}F\left(\omega + \frac{\pi}{N}\right)$ $+\frac{1}{4}F\left(\omega - \frac{\pi}{N}\right)$	$\frac{4\pi}{N}$	2.7%
Hamming	$0.54 + 0.46\cos\left(\frac{\pi t}{N}\right)$	$0.54F(\omega) + 0.23F\left(\omega + \frac{\pi}{N}\right)$ $+0.23F\left(\omega - \frac{\pi}{N}\right)$	$\frac{4\pi}{N}$	0.7%

the system and the input signal cannot be manipulated. In this type of problems, the truncation error can be reduced by increasing the number of data and using the appropriate windows.

Practical aspects of Fourier analysis

For computing the Fourier transform of sampled signals, the so called Discrete Fourier Transform (DFT) is used. The DFT of a sampled signal $x(k)$ for $k = 0, 1, \ldots, N-1$ is defined as:

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-jk\frac{2\pi n}{N}} \qquad \text{for} \quad n = 0, 1, \ldots, N-1 \tag{2.79}$$

Note that DFT is not equal to the Fourier transform of a discrete signal which is a continuous function for aperiodic signals. DFT samples the Fourier transform of an aperiodic discrete-time signal with a sampling interval ω_s/N, where ω_s is the sampling frequency of the discrete-time signal. Therefore, if we plot $X(n)$ with respect to a frequency vector

$$\left[0, \frac{\omega_s}{N}, \frac{2\omega_s}{N}, \cdots, \frac{(N-1)\omega_s}{N}\right]$$

$X(n)$ will be superimposed on the Fourier transform of $x(k)$ in N uniformly distributed frequency points. DFT is also N times the exact value of the Fourier transform of discrete-time periodic signals (the Fourier transform is discrete).

For an input signal, $u(k)$, and an output signal, $y(k)$, with $k = 0, 1, \ldots, N-1$, the frequency response of the system at N distinct points can be computed by dividing the DFT of y by the DFT of u:

$$G(e^{j\omega_n}) = \frac{Y(n)}{U(n)} \qquad \omega_n = \frac{n}{N}\omega_s \qquad n = 0, 1, \ldots, N-1 \tag{2.80}$$

then $G(e^{j\omega})$ can be estimated by linear interpolation of the consecutive points in $G(e^{j\omega_n})$. In practice, since $G(e^{j\omega})$ is symmetric with respect to the Nyquist frequency, $\omega_s/2$, it is sufficient to compute $G(e^{j\omega_n})$ only for frequencies less than the Nyquist frequency.

The quality of the identified model in (2.80) depends on several issues, namely:

- The level of the measurement noise and disturbances with respect to the level of the output signal. The noise effect can be reduced generally by increasing the number of data and averaging on several frequency response estimates.
- The frequency content of the excitation signal. In the frequencies where the excitation is poor ($|U(e^{j\omega})|$ is small), the quality of the frequency response estimate is poor. The use of rich signals (white noise or PRBS) can help to improve the quality of the identified model.
- The periodicity of the input signal has a major effect on the quality of the frequency response estimate. Using a periodic signal the truncation error is completely removed. Moreover, by increasing the number of periods and averaging the estimates, the measurement noise can be significantly reduced.
- For the aperiodic excitation signals, there exists always some truncation error that can be reduced by using a Hann or a Hamming window. Note that by increasing the number of data N, the resolution of the estimates will be increased (the number of the frequency response estimates is equal to the number of data N), but the accuracy remains unchanged so the effect of noise will not be reduced. If a large number of data is available, it is reasonable to divide the data in m equal length sets and take an average on the estimated frequency responses:

$$G(e^{j\omega_n}) = \frac{1}{m}\sum_{i=1}^{m} G_i(e^{j\omega_n}) \qquad \omega_n = \frac{n}{N}\omega_s \qquad n = 0, 1, \ldots, N-1 \quad (2.81)$$

  This way, the resolution will be decreased but the accuracy of the estimates will be improved.

Example 2.2. Consider the following plant model:

$$G(z) = \frac{0.5z + 1}{z^2 - 1.5z + 0.7} \quad (2.82)$$

excited by a zero-mean random signal of length $N = 40$. The input and output of the system are shown in Fig. 2.13 in a noiseless simulation. If we compute the DFT of the output and divide it by the DFT of the input, the frequency response of the system will be estimated. This result is compared with the frequency response of the true model in the same figure. The frequency response error is completely related to the truncation error. In fact, for $k > N$, $u(k) = 0$ but $y(k)$ is not equal to zero that leads to a truncation error. If we increase N to 60, $y(k)$ will converge

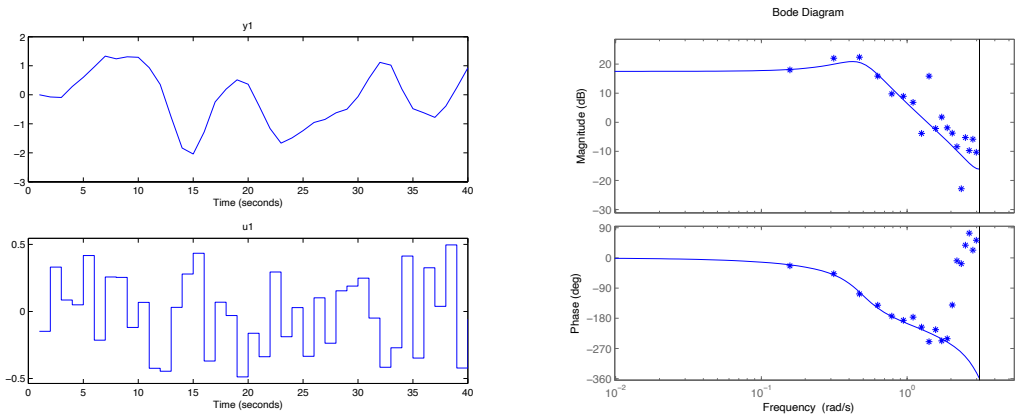


Figure 2.13: Illustration of the truncation error (the output is truncated before converging to zero)

to zero and the truncation error will be vanished. Fig. 2.14 shows the input and output of the same experiment but letting the output converging to zero. In this experiment, there will be no truncation error and therefore the frequency response estimates will be superimposed with the true one.

The truncation error can be suppressed if a periodic signal is applied to the system. In the next simulation, the input signal will be repeated four times and the data from the last period will be used for identification. Figure 2.15 shows the four periods of input and output data and the results of the Fourier analysis on the last period. In this case, there will be no truncation error.

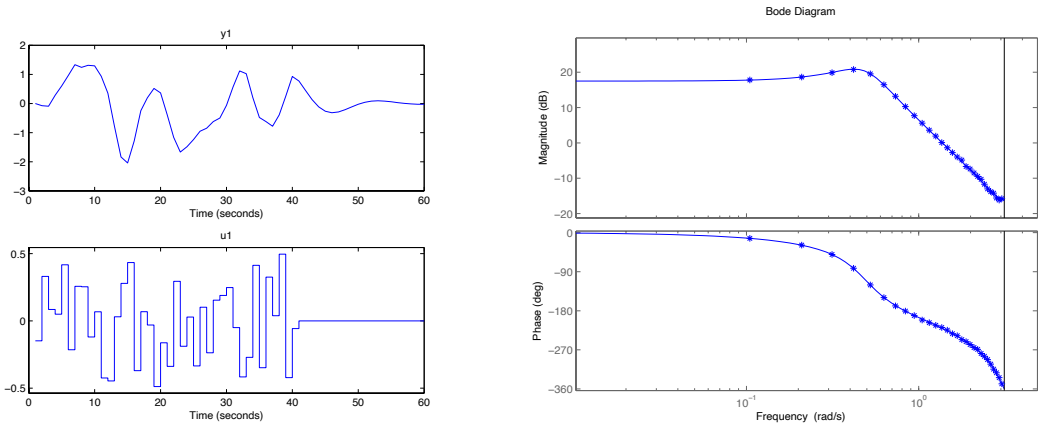


Figure 2.14: Excitation with an energy signal and waiting for transient vanishing (no truncation error)

The truncation error is completely removed with the cost of new experiments (with an energy or a periodic signal). If we are given the input/output data and there is no possibility to do a specific experiment, the windowing approach can be

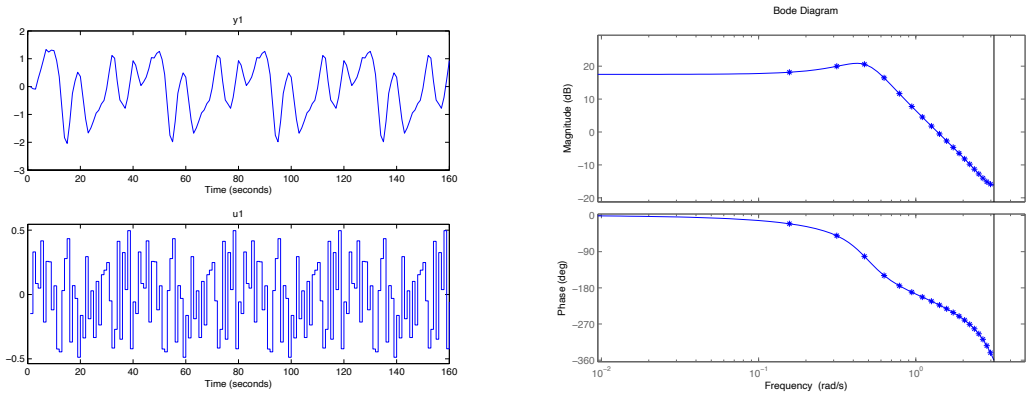


Figure 2.15: Excitation with a periodic signal and using the last period (no truncation error)

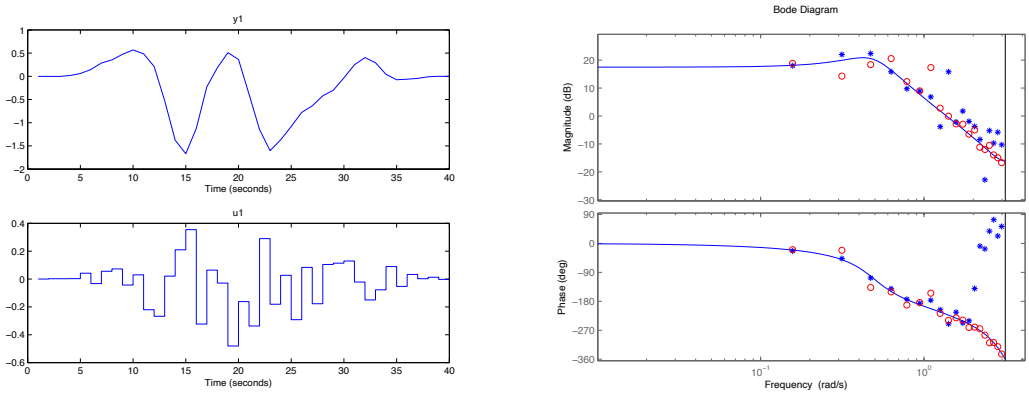


Figure 2.16: The effect of a Hann window on the data and on the frequency response estimate (red circles) compared with rectangular window (blue star)

used. Figure 2.16, shows the effect of a Hann window on the input/output data and the resulting frequency response. It can be observed that although the truncation error has not been removed it has been reduced significantly in high frequencies.

2.3.3 Spectral analysis

In this section, another method that will reduce the noise effect for non periodic excitation signals (random signals) based on the correlation approach is explained. It was shown in Section 2.2.3 that if an LTI system is excited by a random signal $u(k)$ which is uncorrelated with the disturbance signal $d(k)$, the following convolution relation is satisfied:

$$R_{yu}(h) = g(h) * R_{uu}(h) \qquad \text{for } h = 0, \pm 1, \pm 2, \ldots \tag{2.83}$$

Taking the Fourier transform from the both sides of the above equation we obtain:

$$\Phi_{yu}(\omega) = G(e^{j\omega})\Phi_{uu}(\omega) \tag{2.84}$$

where $\Phi_{yu}(\omega)$ is the cross-spectral density function of the input and the output signal and defined as:

$$\Phi_{yu}(\omega) = \sum_{h=-\infty}^{\infty} R_{yu}(h)e^{-j\omega hT} \tag{2.85}$$

Therefore, the frequency response can be obtained by:

$$G(e^{j\omega}) = \frac{\Phi_{yu}(\omega)}{\Phi_{uu}(\omega)} \tag{2.86}$$

It is observed that the above relation is not affected by noise so a better model should be identified. However, the exact value of the correlation functions $R_{uu}(h)$ and $R_{yu}(h)$ cannot be computed for the random signals (it requires an infinite number of data). On the other hand, the Fourier transform of the correlation functions is also contaminated by the truncation error and inaccurate estimates of the correlation functions.

In practice, a biased estimate of $R_{uu}(h)$ and $R_{yu}(h)$ can be computed using respectively (2.28) and (2.31). An unbiased estimate of $R_{uu}(h)$ and $R_{yu}(h)$ can also be obtained using (2.30) and (2.32), respectively. Then DFT can be used to compute the Fourier transform of the correlation functions which leads to the following estimate of the frequency response:

$$\hat{G}(e^{j\omega_n}) = \frac{\hat{\Phi}_{yu}(\omega_n)}{\hat{\Phi}_{uu}(\omega_n)} = \frac{\sum_{h=0}^{N-1} \hat{R}_{yu}(h)e^{-jh\frac{2\pi n}{N}}}{\sum_{h=0}^{N-1} \hat{R}_{uu}(h)e^{-jh\frac{2\pi n}{N}}}; \quad \omega_n = \frac{n\omega_s}{N}; \quad n = 0, 1, \ldots, N-1 \tag{2.87}$$

Estimating the noise spectrum: There is another useful relation between the spectra of a linear system. Consider that the output of a system, $y(k)$, excited by the input $u(k)$ independent of the disturbance $d(k)$, is given as:

$$y(k) = g(k) * u(k) + d(k) \tag{2.88}$$

then we have the following relation:

$$\Phi_{yy}(\omega) = |G(e^{j\omega})|^2\Phi_{uu}(\omega) + \Phi_{dd}(\omega) \tag{2.89}$$

This can be easily proved using $Y(e^{j\omega}) = G(e^{j\omega})U(e^{j\omega}) + D(e^{j\omega})$ and multiplying it with its conjugate and taking the expectation that leads to:

$$\mathbb{E}\{|Y(e^{j\omega})|^2\} = \mathbb{E}\{|G(e^{j\omega})|^2|U(e^{j\omega})|^2\} + \mathbb{E}\{|D(e^{j\omega})|^2\} \tag{2.90}$$

knowing that the expectation of the products of $U(e^{-j\omega})$ and $D(e^{j\omega})$ is zero because of their independence. The relation in (2.89) is concluded directly from the above equation. This relation can be used to estimate the noise spectrum from the data by:

$$\Phi_{dd}(\omega) = \Phi_{yy}(\omega) - \frac{|\Phi_{yu}(\omega)|^2}{\Phi_{uu}(\omega)} \tag{2.91}$$

Remark: Note that for periodic signals, or for the signals that are assumed periodic by ignoring the randomness of the measurement noise, the spectral analysis method will be equivalent to the Fourier analysis method. Because, it was shown that for a periodic signal $u(k)$, the power spectral density $\Phi_{uu}(\omega_n)$ can be obtained by:

$$\Phi_{uu}(\omega_n) = U(e^{j\omega_n})U(e^{-j\omega_n}) = |U(e^{j\omega_n})|^2$$

In the same way, it can be shown that

$$\Phi_{yu}(\omega_n) = Y(e^{j\omega_n})U(e^{-j\omega_n})$$

Therefore,

$$G(e^{j\omega_n}) = \frac{\Phi_{yu}(\omega_n)}{\Phi_{uu}(\omega_n)} = \frac{Y(e^{j\omega_n})}{U(e^{j\omega_n})} \tag{2.92}$$

In general, however, the results of two approaches in the presence of noise and finite number of data is different. The spectral analysis gives more flexibility to improve the results by using different estimators for the correlation functions and the choice of windows, that will be discussed in what follows, that influence considerably the final results.

Improving the spectral analysis method

The frequency response estimate by spectral analysis can be improved by two methods: averaging and windowing.

Averaging: It should be mentioned that averaging *unbiased* estimates will reduce the variance of the estimates and preserve the unbiasedness of the averaged estimates. However, it is not the case for *biased* estimates, where the averaging will not necessary improve the accuracy.

Suppose that m sets of input/output data are available. An unbiased estimate of $R_{uu}(h)$ and $R_{yu}(h)$ can be obtained using (2.30) and (2.32) for each set of data. Since, the spectral density functions are linear with respect to the correlation functions they will also be unbiased so their estimate can be improved by averaging as follows:

$$\overline{\Phi}_{uu}(\omega_n) = \frac{1}{m}\sum_{i=1}^{m} \Phi_{uu,i}(\omega_n) \tag{2.93}$$

$$\overline{\Phi}_{yu}(\omega_n) = \frac{1}{m}\sum_{i=1}^{m} \Phi_{yu,i}(\omega_n) \tag{2.94}$$

and finally:

$$G(e^{j\omega_n}) = \frac{\overline{\Phi}_{yu}(\omega_n)}{\overline{\Phi}_{uu}(\omega_n)} \tag{2.95}$$

Note that the averaging is in the spectral density level and not on different $G_i(e^{j\omega_n})$ as it was in (2.81). Because, in this case G_i, which is computed by the ratio of two unbiased estimates is not unbiased and its averaging will not improve the accuracy.

Windowing: The use of unbiased estimates of the correlation functions, however, has the drawback of large variance for large $|h|$ as it was discussed in Section 2.2.3. A windowing method can be used to reduce the effect of these errors for large $|h|$. In this case, the windowed cross-spectral density function can be defined as:

$$\hat{\Phi}_{yu,f}(\omega_n) = \sum_{h=0}^{N-1} \hat{R}_{yu}(h) f(h) e^{-j2\pi nh/N} \tag{2.96}$$

where $f(h)$ is a window function which is large for small h and small for large h (e.g. a Hann or Hamming window). This spectral estimation is known as the Blackman-Tucky method. In the frequency domain, this windowing is equivalent of the convolution product of the Fourier transform of $f(h)$ and the cross-spectral density function and has a smoothing effect on the estimates (note that the spectral leakage is negligible using Hann or Hamming windows). In practice, the window width is much smaller than the length of the correlation functions and is padded with zero to remove completely the imprecise values of the correlation function estimates.

Note that in Fourier analysis, the windows are used in time domain k to reduce the truncation errors, while in spectral analysis they are used in the lag domain h to improve the estimate of the correlation functions. In the spectral analysis methods, only half of the window is applied on the positive value of h and they are usually padded with zeros. The following windows are often used in the spectral analysis methods:

Rectangular:

$$f(h) = \begin{cases} 1 & h = 0, 1, \dots, M \\ 0 & h > M \end{cases} \tag{2.97}$$

Triangular (Bartlett):

$$f(h) = \begin{cases} 1 - h/M & h = 0, 1, \dots, M \\ 0 & h > M \end{cases} \tag{2.98}$$

Hann:

$$f(h) = \begin{cases} 0.5 + 0.5\cos(\pi h/M) & h = 0, 1, \dots, M \\ 0 & h > M \end{cases} \tag{2.99}$$

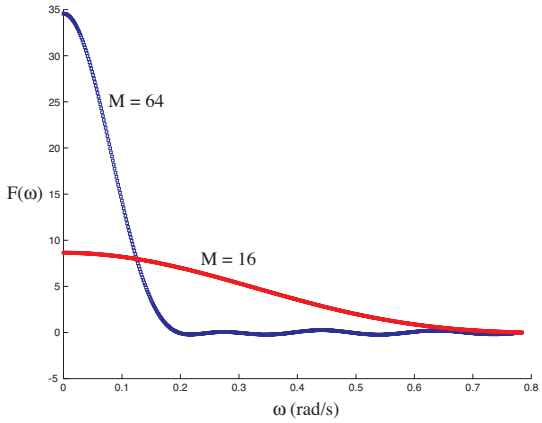


Figure 2.17: Fourier transform of a Hann window for two different MLW

Hamming:

$$f(h) = \begin{cases} 0.54 + 0.46\cos(\pi h/M) & h = 0, 1, \dots, M \\ 0 & h > M \end{cases} \tag{2.100}$$

The choice of window width M, has an important influence on the quality of the frequency response estimate. In general, larger M leads to a smaller MLW in the Fourier transform of the window and reduces the smoothing (increasing the resolution and the spectral variation). Contrarily, a smaller M leads to a larger MLW and smooth the frequency response (see Fig. 2.17). It is clear that if the frequency response contains low-damped resonance modes, a small M leads to chopping the resonance peaks and a larger M is more appropriate. As a general rule, the window should not cut the information on the estimate of the correlation functions. So, after observing $\hat{R}_{uu}(h)$ and $\hat{R}_{yu}(h)$ the window width should be chosen to keep the significant parts of the estimates and remove the imprecise parts. In fact, if a white noise signal is used as the excitation signal, the autocorrelation function $R_{uu}(h)$ will be theoretically equal to zero for $h \neq 0$. On the other hand, the cross-correlation function $R_{yu}(h)$ is an estimate of the impulse response of the system and converges to zero for large h. Therefore, the window width M is usually chosen as the settling time of the impulse response of the system (the settling time K can be estimated by observing $\hat{R}_{yu}(h) \approx 0$ for $h > K$).

Fig. 2.18 shows the effect of a Hamming window of $M = 25$ when applied to the cross-correlation function estimate of a system excited by a white noise. It can be seen that a large part of the main oscillation in $\hat{R}_{yu}(h)$ is removed by windowing. As a result, in the frequency response of the identified model the peak value of the resonance mode is cut and the estimate of the damping factor will not be correct. In Fig. 2.19 the window width is increased to $M = 75$ and as a consequence a better estimate of the resonance peak is obtained. However, the estimate is less smooth comparing to the case with $M = 25$.

In practice, it is reasonable to combine the averaging and windowing method

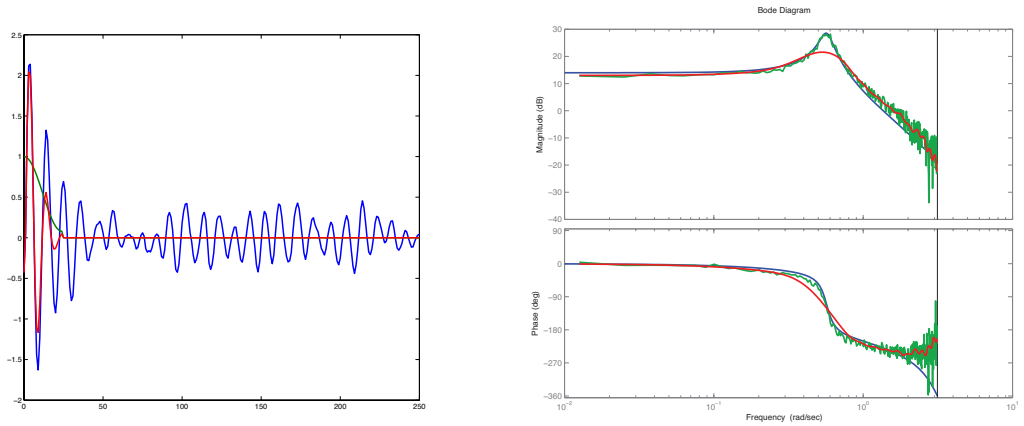


Figure 2.18: The effect of a Hamming window with $M = 25$. The green curve in the Bode diagram corresponds to no windowing and the red one to the estimates with the Hamming window.

to find the best results. The data will be divided into m sets and the correlation functions will be windowed with a Hann or Hamming window with an appropriate width. Then, the spectral density functions $\Phi_{uu}(\omega_n)$ and $\Phi_{yu}(\omega_n)$ are computed by averaging using (2.93) and (2.94) and finally $G(e^{j\omega_n})$ is computed by (2.95).

Quality of the estimates: The quality of the frequency response estimate in the spectral analysis method depends on the quality of the spectral estimates. When a Hamming (or Hann) window with width M is used, it can be shown that the variance of the frequency response is given by:

$$\text{var}[\hat{G}(j\omega)] \approx \frac{0.7M}{N} \frac{\Phi_{dd}(\omega)}{\Phi_{uu}(\omega)} \tag{2.101}$$

This variance can be computed using an estimate of the noise spectrum from (2.91). It can be observed that the variance is proportional to noise-to-signal ratio and typically increases in high frequencies where this ratio is large. It shows also the proportional effect of M on the variance of the estimates.

2.3.4 Closed-loop identification

It is sometimes necessary to use the data acquired in closed-loop operation for system identification. It is the case for systems that are unstable in open-loop operation (inverted pendulum, fighter aircrafts, some chemical systems etc.) and open-loop data acquisition is not possible. In some industrial applications, the production cannot be stopped, for the economic reasons, to allow an open-loop identification. On the other hand, some economic or biological systems have inherent feedback loops. Finally, it is argued that a better model for control design can be obtained

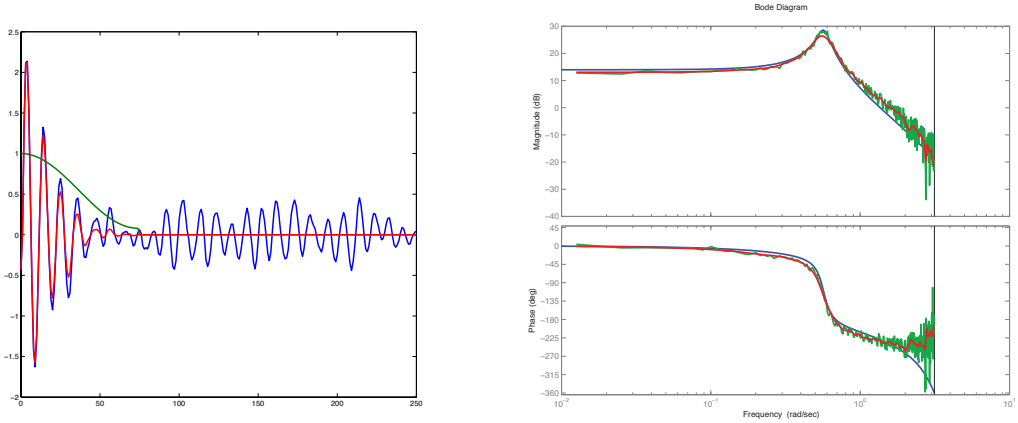


Figure 2.19: The effect of a Hamming window with $M = 75$. The green curve in the Bode diagram corresponds to no windowing and the red one to the estimates with the Hamming window.

if the system is identified when excited with a signal that is close to that of normal operation in closed-loop.

The main technical difference between open-loop identification and closed-loop identification is that the plant's input is not completely controlled by the user and is affected by noise through the feedback. As a result, the main assumption in the correlation approach and the spectral analysis method, i.e. the disturbance $d(k)$ is independent of the input $u(k)$, is not met. Therefore, $R_{du}(h) \neq 0$ for all h and $\Phi_{du}(\omega)$ cannot be neglected.

Consider the closed-loop system in Fig. 2.20 with $y_c(k) = 0$ for all k and the non zero excitation signal $s(k)$. Since $y(k) = y_p(k) + d(k)$, we have:

$$\hat{G}(e^{j\omega}) = \frac{\Phi_{yu}(\omega)}{\Phi_{uu}(\omega)} = \frac{\Phi_{y_p u}(\omega) + \Phi_{du}(\omega)}{\Phi_{uu}(\omega)} = G(e^{j\omega}) + \frac{\Phi_{du}(\omega)}{\Phi_{uu}(\omega)} \tag{2.102}$$

Therefore, the estimation of G will be correct only if $d(k) = 0$.

It is interesting to see that in the absence of the excitation signals ($y_c(k) = 0, s(k) = 0$ for all k), the spectral analysis method will lead to the identification of the inverse of the controller instead of the plant model. In this case, we have $e(k) = -y(k)$ and therefore:

$$\hat{G}(e^{j\omega}) = \frac{\Phi_{yu}(\omega)}{\Phi_{uu}(\omega)} = \frac{-\Phi_{eu}(\omega)}{\Phi_{uu}(\omega)} = \frac{-\Phi_{eu}(\omega)}{K(e^{j\omega})\Phi_{eu}(\omega)} = \frac{-1}{K(e^{j\omega})} \tag{2.103}$$

In the above demonstration, we used $\Phi_{uu}(\omega) = K(e^{j\omega})\Phi_{eu}(\omega)$. This relation is obtained by applying the spectral analysis relation considering $u(k)$ as the input and $e(k)$ as the output of the inverse of the controller.

A good model, however, can be identified using the spectral analysis method in closed-loop operation, if there exists an external excitation signal independent

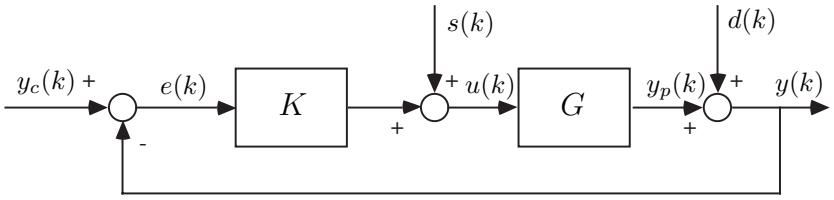


Figure 2.20: Closed-loop system with the external excitations $s(k)$ and $y_c(k)$ and output disturbance $d(k)$.

of $d(k)$. The main idea is to consider the closed-loop system (plant model with controller and feedback loop) as a black-box open-loop system with one input (the external excitation signal $s(k)$) and one independent disturbance signal $d(k)$. In the first step, the frequency response between the external excitation signal $s(k)$ and the output $y(k)$ can be identified:

$$\frac{\hat{G}(e^{j\omega})}{1 + K(e^{j\omega})\hat{G}(e^{j\omega})} = \frac{\Phi_{ys}(\omega)}{\Phi_{ss}(\omega)} \tag{2.104}$$

Similarly, the frequency response between $s(k)$ and $u(k)$ can be estimated by:

$$\frac{1}{1 + K(e^{j\omega})\hat{G}(e^{j\omega})} = \frac{\Phi_{us}(\omega)}{\Phi_{ss}(\omega)} \tag{2.105}$$

If we divide the equations (2.104) and (2.105), we obtain:

$$\hat{G}(e^{j\omega}) = \frac{\Phi_{ys}(\omega)}{\Phi_{us}(\omega)} \tag{2.106}$$

In the same way, if the closed-loop system is excited with $y_c(k)$ instead of $s(k)$ the frequency response of the plant model can be estimated by:

$$\hat{G}(e^{j\omega}) = \frac{\Phi_{yy_c}(\omega)}{\Phi_{uy_c}(\omega)} \tag{2.107}$$

2.3.5 Parametric identification from frequency response

The frequency response of the system identified by the Fourier or spectral analysis can be used to tune simple loop-shaping controllers or advanced data-driven robust control methods. The frequency response data can be also used to estimate the parameters of a parametric model given by:

$$G(s) = \frac{b_m s^m + \ldots + b_1 s + b_0}{a_n s^n + \ldots + a_1 s + 1} \qquad m \leq n \tag{2.108}$$

Two approaches can be used to obtain the parameters of such a model.

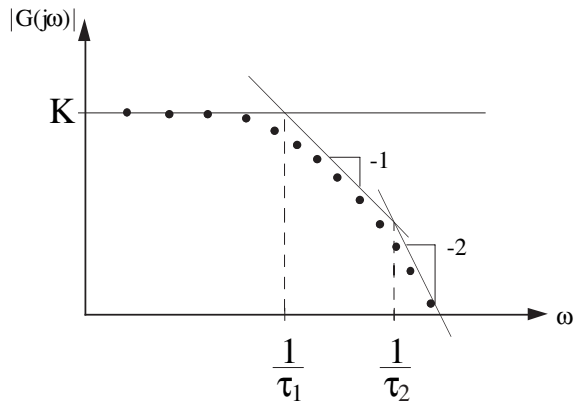


Figure 2.21: Graphical approach for parametric model identification from frequency response data

Graphical approach: The graphical approach is illustrated in Fig. 2.21. By this approach, the steady-state gain, the simple poles and zeros can be identified from the slop of the magnitude Bode diagram. As an example, for the system given in Fig. 2.21, the following transfer function can be identified.

$$G(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{2.109}$$

It is clear that this method is very approximative and can be used only to find some initial estimate about the order of the system and a rough estimate of the range of the parameters.

Numerical approach: It is supposed that a set of N complex value $G(j\omega_k)$ for $k = 1, \ldots, N$ is available from a nonparametric model identification. In the numerical approach, the real coefficients $(a_1, ...a_n, b_0, ...b_m)$ are computed such that the complex value computed at each frequency ω_k by the model:

$$\hat{G}(j\omega) = \frac{\hat{B}(j\omega)}{\hat{A}(j\omega)} = \frac{b_m(j\omega)^m + \ldots + b_1(j\omega) + b_0}{a_n(j\omega)^n + \ldots + a_1(j\omega) + 1} \tag{2.110}$$

is as close as possible to the identified value $G(j\omega_k)$. The following error, which is a complex value can be defined at each frequency:

$$\varepsilon(\omega_k) = G(j\omega_k) - \frac{\hat{B}(j\omega_k)}{\hat{A}(j\omega_k)} \tag{2.111}$$

Then, an optimization problem can be defined in which the vector of parameters $\theta = [a_1, ...a_n, b_0, ...b_m]^T$ are found such that the following criterion is minimized:

$$J(\theta) = \sum_{k=1}^{N} |\varepsilon(\omega_k)|^2 = \left| G(j\omega_k) - \frac{\hat{B}(j\omega_k)}{\hat{A}(j\omega_k)} \right|^2 \tag{2.112}$$

This optimization problem is nonlinear and may lead to a local optimal solution. To fix this problem, a linear error can be defined as:

$$\varepsilon_\ell(\omega_k) = \hat{A}(j\omega_k)G(j\omega_k) - \hat{B}(j\omega_k) \tag{2.113}$$

Consider the following system of N linear equations:

$$\begin{aligned}
\hat{A}(j\omega_1)G(j\omega_1) - \hat{B}(j\omega_1) &= 0 \\
\hat{A}(j\omega_2)G(j\omega_2) - \hat{B}(j\omega_2) &= 0 \\
&\vdots \\
\hat{A}(j\omega_N)G(j\omega_N) - \hat{B}(j\omega_N) &= 0
\end{aligned}$$

This system of linear equations can be solved in the least squares sense but it will lead to complex values for the vector of the parameters. In order to overcome this problem, each complex equation is replaced with two equations with real coefficients using the real and the imaginary parts of the complex values:

$$\begin{aligned}
R_e\{\hat{A}(j\omega_k)G(j\omega_k)\} - R_e\{\hat{B}(j\omega_k)\} = 0 \qquad \text{for} \quad k = 1, \ldots, N \\
I_m\{\hat{A}(j\omega_k)G(j\omega_k)\} - I_m\{\hat{B}(j\omega_k)\} = 0 \qquad \text{for} \quad k = 1, \ldots, N
\end{aligned}$$

The least squares solution to this problem is given by

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T\mathcal{G} \tag{2.114}$$

where

$$\Phi = \begin{bmatrix}
-R_e\{(j\omega_1)G(j\omega_1)\} & \ldots & -R_e\{(j\omega_1)^nG(j\omega_1)\} & 1 & R_e\{j\omega_1\} & \ldots & R_e\{j\omega_1\}^m \\
-R_e\{(j\omega_2)G(j\omega_2)\} & \ldots & -R_e\{(j\omega_2)^nG(j\omega_2)\} & 1 & R_e\{j\omega_2\} & \ldots & R_e\{j\omega_2\}^m \\
\vdots & & \vdots & \vdots & \vdots & & \vdots \\
-R_e\{(j\omega_N)G(j\omega_N)\} & \ldots & -R_e\{(j\omega_N)^nG(j\omega_N)\} & 1 & R_e\{j\omega_N\} & \ldots & R_e\{j\omega_N\}^m \\
-I_m\{(j\omega_1)G(j\omega_1)\} & \ldots & -I_m\{(j\omega_1)^nG(j\omega_1)\} & 0 & I_m\{j\omega_1\} & \ldots & I_m\{j\omega_1\}^m \\
-I_m\{(j\omega_2)G(j\omega_2)\} & \ldots & -I_m\{(j\omega_2)^nG(j\omega_2)\} & 0 & I_m\{j\omega_2\} & \ldots & I_m\{j\omega_2\}^m \\
\vdots & & \vdots & \vdots & \vdots & & \vdots \\
-I_m\{(j\omega_N)G(j\omega_N)\} & \ldots & -I_m\{(j\omega_N)^nG(j\omega_N)\} & 0 & I_m\{j\omega_N\} & \ldots & I_m\{j\omega_N\}^m
\end{bmatrix}$$

and

$$\mathcal{G} = \begin{bmatrix}
R_e\{G(j\omega_1)\} \\
R_e\{G(j\omega_2)\} \\
\vdots \\
R_e\{G(j\omega_N)\} \\
I_m\{G(j\omega_1)\} \\
I_m\{G(j\omega_2)\} \\
\vdots \\
I_m\{G(j\omega_N)\}
\end{bmatrix} \tag{2.115}$$

Remark: It should be mentioned that the absolute value of the linear error is:

$$|\varepsilon_\ell(j\omega_k)| = |[a_n(j\omega_k)^n + \ldots + a_1(j\omega_k) + 1]\varepsilon(j\omega_k)| \tag{2.116}$$

So the real error $\varepsilon(j\omega_k)$ is weighted by a frequency dependent weighting function, which is large at high frequencies (note that $1/\hat{A}$ is typically a low-pass filter). Therefore, the vector of parameters will be identified such that a much better fit in high frequencies will be obtained. In practice, it will not be a good model, since usually a better fit in low frequencies is required. A solution to this problem can be obtained by an iterative approach in which at each iteration the following criterion based on a filtered error is minimized:

$$J(\theta_i) = \sum_{k=1}^{N} \left| \frac{\varepsilon_\ell(\omega_k, \theta_i)}{\hat{A}(\theta_{i-1})} \right|^2 \tag{2.117}$$

where $\hat{A}(\theta_{i-1})$ is computed in the previous iteration. This algorithm can be initialized with the standard algorithm with no filtering.

Chapter 3

Parametric Identification Methods

3.1 Introduction

In this chapter, identification of parametric models is investigated. Parametric models describe the input/output behavior of a system with a few parameters, which should be estimated using an optimization algorithm. From this point of view, they have an important advantage with respect to the nonparametric identification methods in which the number of estimates is the same as the number of data. For example, a step or impulse response has the same size as the acquired data and the measurement noise enters directly on the estimates. In frequency-domain identification by Fourier analysis, the number of complex value estimates is equal to the number of data (in fact, if we consider the real and the imaginary parts of the estimates, they are twice the number of data but because of the symmetry in the frequency response only half of the estimates, up to the Nyquist frequency, are meaningful). Therefore, in nonparametric identification, the noise effect is not naturally filtered and it can be reduced by some averaging methods. In parametric identification, the number of data (number of equations) is much more than the number of parameters, which leads to a redundancy of the information that can be used to filter out the noise effect. For example, a first-order linear model with delay represented by $G(s) = Ke^{-T_d s}/(\tau s + 1)$ has only three parameters to describe the input/output relationship, which helps reducing the noise effect significantly. However, the model structure should be known a priori in a parametric identification method, which requires some a priori information about the system to be identified.

In the previous chapter, some algorithms to identify parametric models from nonparametric models in time-domain and frequency-domain were presented. In this chapter, the identification of the model parameters is performed based on the minimization of an error between the measured output and the output of a parameterized model for the same input signal. The main elements of a parametric identification are:

- A set of experimental data Z^N defined as:

$$Z^N = \{(y(k), u(k)) \,|\, k = 1, \ldots, N\}$$

In this part, the choice of input excitation signal is of great importance.

- A model structure. This is a parametrized mapping from past inputs and outputs Z^{k-1} to the space of the model outputs. The model structure is used to define a parameterized predictor:

$$\hat{y}(k,\theta) = \mathcal{F}(\theta, Z^{k-1})$$

where θ is the vector of the parameters of the model. The major problem in parametric model identification is the choice of a good model structure. Note that this mapping is very general and includes the continuous and discrete-time, SISO and MIMO as well as linear and nonlinear models in transfer function or state-space representation.

- A fit criterion that should be minimized. The criterion is often defined as:

$$J(\theta) = \sum_{k=1}^{N} \varepsilon^2(k,\theta) \tag{3.1}$$

where $\varepsilon(k,\theta) = y(k) - \hat{y}(k,\theta)$ is the prediction error. Instead of sum of squares, any other function of $\varepsilon(k,\theta)$ can also be considered in the criterion. Actually, the problem of fitting the data to the model is a numerical optimization method and is much simpler than the selection of the model structure. Depending on the model structure, the optimization problem may be linear or nonlinear with the possibility to find a local minimum in optimization.

- Model validation algorithm. Any identified model should be validated using another set of data. The nonparametric models can be used to validate the linear parametric models.

Example 3.1. In order to clarify the above elements, consider again the example of a heater immersed in a cooling liquid (Example 1.3). The following model was developed for the system:

$$y(k) = y(k-1) + \alpha v^2(k-1) - \beta[y(k-1) - T(k-1)]$$

So the following predictor can be proposed for this model structure:

$$\hat{y}(k,\theta) = \theta_1 y(k-1) + \theta_2 v^2(k-1) + \theta_3 T(k-1) = \phi^T(k)\theta$$

where $\theta^T = [\theta_1, \theta_2, \theta_3]$ and $\phi^T(k) = [y(k-1)\,,\, v^2(k-1)\,,\, T(k-1)]$. Since, the predictor is presented by a linear regression, the minimization of $\varepsilon(k,\theta) = y(k) - \hat{y}(k,\theta)$ can be performed by the standard least squares algorithm leading to:

$$\hat{\theta} = \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi(k)y(k)$$

Without a validation test the resulting model cannot be accepted. This requires another set of data, called validation data, and check if the identified model can fit

the new data as well. If it is not the case, the procedure of identification may be repeated by a new set of data with a more appropriate and richer excitation signal or by proposing another model structure that may be more complex.

This chapter is organized as follows. Section 3.2 presents some basic model structures like FIR and ARX and state-space models and simple algorithms to identify the parameters of such models. The general linear black-box models are presented in Section 3.3 and the bias and the variance of the parameters are studied. An introduction to nonlinear system identification is given in Section 3.7. The numerical optimization methods to minimize the fit criterion are discussed in Section 3.6. This includes the Gauss-Newton algorithm, the pseudo-linear regression and on-line recursive algorithms. Finally, the practical aspects of identification, including the choice of the sampling period, pretreatment of data, model structure selection and cross validation are discussed in Section 3.5.

3.2 Basic Model Structures

In this section, some simple classical model structures for linear time-invariant systems are discussed and their statistical properties will be studied in the presence of measurement noise. These structures are chosen because the minimization of the fit criterion leads to a least squares algorithm and can be solved by standard methods.

3.2.1 Finite impulse response (FIR) model

FIR models are studied in the nonparametric identification algorithms. It was shown that the problems in identifying the impulse response by deconvolution technique and the correlation approach can be fixed if the impulse response is supposed to be finite. In this section, an FIR model is identified using the time-domain data from the view point of parametric identification methods. Not surprisingly, the results will be the same, although the approaches are different.

Suppose that the output of a system depends only on the present and past inputs, i.e.:

$$y(k) = b_0 u(k) + b_1 u(k-1) + \cdots + b_m u(k-m) \tag{3.2}$$

Taking the z-transform from the above equation leads to

$$Y(z) = (b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}) U(z) = G(z) U(z) \tag{3.3}$$

where

$$G(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m} \tag{3.4}$$

is an FIR model. Note that in the discrete-time systems, there is always some delay between the inputs and outputs such that the first leading coefficients of the FIR model becomes equal to zero. The number of these coefficients is called delay and

denoted by d. Because of sampling time delay the output $y(k)$ will not depend on $u(k)$ (b_0 is always zero and so the delay d is always greater than or equal to 1).

In the identification literature instead of the z transform, a time-domain delay operator q^{-1} is used, which is defined as:

$$q^{-1}x(k) \equiv x(k-1) \tag{3.5}$$

Therefore, an FIR model can be equivalently defined as:

$$G(q^{-1}) = b_d q^{-d} + b_{d+1} q^{-d-1} + \cdots + b_m q^{-m} \tag{3.6}$$

and the output of the model is given by:

$$y(k) = G(q^{-1})u(k) \tag{3.7}$$

The interest of the above representation is that the time-domain signals and a parametric model appear in one equation, while in the standard discrete-time model (3.3), the z-transform of the signals are involved.

If this model is compared with the convolution relation given in (2.11), it becomes clear that the parameters of this model are the values of the impulse response of the system:

$$b_d = g(d), \quad b_{d+1} = g(d+1), \quad \cdots \quad b_m = g(m)$$

and $g(k) = 0$ for all $k < d$. The FIR model can be written in the regression form as:

$$y(k) = \phi^T(k)\theta \tag{3.8}$$

where

$$\phi^T(k) = [u(k-d)\,,\, u(k-d-1)\,,\, \ldots\,,\, u(k-m)] \tag{3.9}$$

$$\theta^T = [b_d\,,\, b_{d+1}\,,\, \ldots\,,\, b_m] \tag{3.10}$$

A predictor for the output of the FIR model is given by:

$$\hat{y}(k,\theta) = \phi^T(k)\theta \tag{3.11}$$

Note that the predictor should depend on the unknown parameters and the past data, which is the case for the proposed predictor. If, for the fit criterion, we consider the sum of the squares of the prediction error, we obtain:

$$J(\theta) = \sum_{k=1}^{N} [y(k) - \hat{y}(k,\theta)]^2 \tag{3.12}$$

This is a least squares problem and the solution is given by:

$$\hat{\theta} = \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi(k)y(k) \tag{3.13}$$

The solution exists if the matrix in the bracket is nonsingular. This matrix is called *the information matrix* and has an important effect in the quality of the estimates.

Quality of the estimates

Suppose that the data are really generated by an FIR model given as:

$$y(k) = b_d^\circ u(k-d)+b_{d+1}^\circ u(k-d-1)+\cdots+b_m^\circ u(k-m)+e(k) = \phi^T(k)\theta_0+e(k) \quad (3.14)$$

where $e(k)$ is a zero-mean noise independent of $u(k)$ and

$$\theta_0^T = [b_d^\circ\,,\, b_{d+1}^\circ\,,\,\ldots\,,\, b_m^\circ]$$

is the vector of the true parameters of the model. Note that the assumption of the existence of the *true parameters* is required, if we want to assess the quality of the parameter estimates (bias and variance). However, it should be mentioned that in practice this assumption is rarely satisfied because of the nonlinearity in the system and unmodelled dynamics.

We can now replace $y(k)$ from the above equation in (3.13) to obtain:

$$\hat{\theta} = \left[\sum_{k=1}^{N}\phi(k)\phi^T(k)\right]^{-1}\sum_{k=1}^{N}\phi(k)[\phi^T(k)\theta_0 + e(k)] \quad (3.15)$$

$$= \theta_0 + \left[\sum_{k=1}^{N}\phi(k)\phi^T(k)\right]^{-1}\sum_{k=1}^{N}\phi(k)e(k) \quad (3.16)$$

Since $e(k)$ is zero-mean and independent of all elements of the ϕ vector, the expectation of the second term in the above equation is zero and we have:

$$\mathbb{E}\{\hat{\theta}\} = \theta_0 \quad (3.17)$$

Therefore, the parameter estimates are *unbiased* if the matrix in the bracket is not singular.

If the noise is white with variance σ^2, the covariance matrix of the estimates can be obtained as:

$$\mathrm{cov}[\hat{\theta}] = \mathbb{E}\left\{(\hat{\theta}-\theta_0)(\hat{\theta}-\theta_0)^T\right\} = \sigma^2\left[\sum_{k=1}^{N}\phi(k)\phi^T(k)\right]^{-1} \quad (3.18)$$

The covariance of the estimate can be rewritten as

$$\mathrm{cov}[\hat{\theta}] = \frac{\sigma^2}{N}\left[\frac{1}{N}\sum_{k=1}^{N}\phi(k)\phi^T(k)\right]^{-1} \quad (3.19)$$

Note that the term in the bracket is an estimate of the autocorrelation of the vector $\phi(k)$ at $h = 0$ or an estimate of the covariance of the vector of the input signal, i.e. the ij-th element of the matrix is an estimate of $R_{uu}(i-j)$.

The general properties of the FIR estimates can be summarized as:

- The covariance of the parameter error decays like $1/N$.
- The covariance is proportional to noise-to-signal-ratio.
- The covariance does not depend on the specific shape of input or noise signal.
- A good experiment is the one in which the covariance matrix of the input signal, or the information matrix, is large.

As it is shown in (1.77), an unbiased estimate of the noise variance can be obtained by:

$$\widehat{\sigma^2} = \frac{1}{N-m} J(\hat{\theta}) \tag{3.20}$$

Identification by orthogonal basis functions

The main problem of the FIR models is that the number of parameters should be large (theoretically infinite) in order to represent the input/output behavior of a linear system. For example, a first order discrete-time model with two parameters in a rational transfer function representation may have more than 30 parameters in an FIR representation. The number of parameters may be much larger for low-damped oscillatory systems.

Let us define a model as:

$$G(q^{-1}) = \sum_{i=d}^{m} b_i F_i(q^{-1}) \tag{3.21}$$

where

$$F_i(q^{-1}) = \frac{q^{-i}}{A(q^{-1})}, \quad i = d, \ldots, m \tag{3.22}$$

are called basis functions. Note that $F_i(q^{-1})$ has infinite impulse response and so there will be no problem to describe a system with a finite number of basis functions. If the poles of the basis functions coincides (or are very close) to those of the real system, with a few number of basis functions the system can be very well described. In this case, the input signal is filtered by $\frac{1}{A(q^{-1})}$ to obtain $u_f(k)$ and then the output is expressed by a linear regression:

$$y(k) = \sum_{i=d}^{m} b_i F_i(q^{-1}) u(k) = \sum_{i=d}^{m} b_i q^{-i} u_f(k) \tag{3.23}$$

$$= b_d u_f(k-d) + b_{d+1} u_f(k-d-1) + \cdots + b_m u_f(k-m) \tag{3.24}$$

and the parameter can be obtained by the classical least squares algorithm.

The key problem is, however, how to choose the poles of $F_i(q^{-1})$. From an engineering point of view, it is reasonable to choose the poles as close as possible to the true poles of the system. If we want that by increasing the number of the

basis functions to infinity the approximation error of any stable transfer function by (3.21) converges to zero, the basis functions should be complete. It has been shown that the necessary and sufficient condition for the completeness of rational basis functions is that:

$$\sum_{i=1}^{\infty}(1-|p_i|)=\infty \tag{3.25}$$

where p_i are the roots of $A(q^{-1})$. The Laguerre basis functions are found by choosing $p_i = a$ for all i, i.e. $A(q^{-1}) = (1 - aq^{-1})^m$. If we choose $a = 0$. i.e. $A(q^{-1}) = 1$, we retrieve the FIR model. Note that FIR and Laguerre basis functions are orthogonal basis functions and have better numerical properties than the non-orthogonal basis functions.

3.2.2 Auto-regressive with external input (ARX) model

The most basic relationship between the input and output of a linear system is given by a linear differential equation as:

$$y(k)+a_1y(k-1)+\cdots+a_ny(k-n)=b_du(k-d)+b_{d+1}u(k-d-1)+\cdots+b_mu(k-m) \tag{3.26}$$

The z transform of this equation gives:

$$[1+a_1z^{-1}+\ldots+a_nz^{-n}]Y(z)=[b_dz^{-d}+b_{d+1}z^{-d-1}+\ldots+b_mz^{-m}]U(z) \tag{3.27}$$

and the following transfer function can be defined:

$$G(z)=\frac{Y(z)}{U(z)}=\frac{b_dz^{-d}+b_{d+1}z^{-d-1}+\ldots+b_mz^{-m}}{1+a_1z^{-1}+\ldots+a_nz^{-n}}=\frac{B(z)}{A(z)} \tag{3.28}$$

where

$$A(z)=1+a_1z^{-1}+\ldots+a_nz^{-n} \tag{3.29}$$

$$B(z)=b_dz^{-d}+b_{d+1}z^{-d-1}+\ldots+b_mz^{-m} \tag{3.30}$$

Since the model is represented by the negative powers of z and $A(z)$ is monic, the model is causal for any positive integer value of m and n. The order of $G(z)$ is $\max(n, m)$. The model can be presented using the delay operator as:

$$A(q^{-1})y(k)=B(q^{-1})u(k) \tag{3.31}$$

with

$$A(q^{-1})=1+a_1q^{-1}+\ldots+a_nq^{-n} \tag{3.32}$$

$$B(q^{-1})=b_dq^{-d}+b_{d+1}q^{-d-1}+\ldots+b_mq^{-m} \tag{3.33}$$

or in a rational form:

$$y(k)=G(q^{-1})u(k)=\frac{B(q^{-1})}{A(q^{-1})}u(k) \tag{3.34}$$

Note that the above equation is used for simplicity of notation (division of two time-domain operator is not mathematically defined) and is equivalent to the meaningful equation in (3.31).

In order to identify the parameters of such a model, let the following output predictor be considered:

$$\begin{aligned}\hat{y}(k,\theta) &= -a_1 y(k-1) - \cdots - a_n y(k-n) \\ &\quad + b_d u(k-d) + \cdots + b_m u(k-m) = \phi^T(k)\theta \end{aligned} \tag{3.35}$$

where:

$$\begin{aligned}\phi^T(k) &= [-y(k-1), \ldots, -y(k-n), u(k-d), \ldots, u(k-m)] \\ \theta^T &= [a_1, \ldots, a_n, b_d, \ldots, b_m]\end{aligned}$$

The prediction error is given by:

$$\varepsilon(k,\theta) = y(k) - \phi^T(k)\theta \tag{3.36}$$

which is a linear regression so the parameters can be identified using the least squares algorithm.

$$\hat{\theta} = \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \left[\sum_{k=1}^{N} \phi(k)y(k)\right] \tag{3.37}$$

The main interest of this model is that the parameters of the numerator and denominator are identified by the least squares algorithm and with a few parameters, rational transfer function models can be identified.

Quality of the estimates

Suppose that the data are generated by the following true model:

$$y(k) + a_1^\circ y(k-1) + \cdots + a_n^\circ y(k-n) = b_d^\circ u(k-d) + \cdots + b_m^\circ u(k-m) + e(k) \tag{3.38}$$

where $e(k)$ is a zero-mean noise independent of $u(k)$ and

$$\theta_0^T = [a_1^\circ, \ldots, a_n^\circ, b_d^\circ, \ldots, b_m^\circ]$$

is the vector of the true parameters. Applying the delay operator, q^{-1}, this model can be represented by:

$$y(k) = \frac{B_0(q^{-1})}{A_0(q^{-1})}u(k) + \frac{1}{A_0(q^{-1})}e(k) \tag{3.39}$$

It can be seen that the noise at the output of the system is filtered by the inverse of the denominator of the plant model. This model is called Auto-Regressive with eXternal input or ARX.

The true output of the system can be written in regression form as:

$$y(k) = \phi^T(k)\theta_0 + e(k) \tag{3.40}$$

If we replace $y(k)$ from the above equation in (3.37), we obtain:

$$\hat{\theta} = \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi(k)[\phi^T(k)\theta_0 + e(k)] \tag{3.41}$$

$$= \theta_0 + \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi(k)e(k) \tag{3.42}$$

which is the same relation that we had for the analysis of the FIR estimates with the difference that now, the regressor vector contains the past values of the output and the input. The parametric error can be rewritten as:

$$\tilde{\theta} \equiv \hat{\theta} - \theta_0 = \left[\frac{1}{N}\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \left[\frac{1}{N}\sum_{k=1}^{N} \phi(k)e(k)\right] \tag{3.43}$$

If the number of data N goes to infinity, the terms in the brackets will be the auto-correlation function of ϕ and the cross-correlation function of ϕ and e, respectively, i.e.:

$$\lim_{N\to\infty} (\hat{\theta} - \theta_0) = R_{\phi\phi}^{-1}(0)R_{\phi e}(0) \tag{3.44}$$

Therefore, the parameter estimates will be asymptotically unbiased if the following conditions are met:

1. $R_{\phi\phi}(0)$ is not singular,
2. $R_{\phi e}(0) = 0$.

The first condition is satisfied if the excitation signal is sufficiently rich (if the degree of the persistent excitation of the input is equal or greater than the number of parameters to estimate $(n + m)$). The second condition is met, if the noise signal $e(k)$ is not correlated with the vector ϕ, which contains the past inputs and outputs. Since the noise is independent of the input signal, they are not correlated. On the other hand, the first element of the vector $\phi(k)$ is $y(k-1)$ which is contaminated by $e(k-1)$. Therefore, $e(k)$ is not correlated with $y(k-1)$ and $y(k-2)$ etc., if $e(k)$ is white. In this case, if the variance of the white noise is σ^2, the asymptotic covariance of the parameter estimates will be:

$$\text{cov}(\hat{\theta}) = E\{(\hat{\theta} - \theta_0)(\hat{\theta} - \theta_0)^T\} = \frac{\sigma^2}{N} R_{\phi\phi}^{-1}(0) \tag{3.45}$$

It should be mentioned that the noise at the output of the plant is $e(k)/A_0(q^{-1})$, see (3.39). So the whiteness of $e(k)$ implies that the noise at the output is a filtered white noise with the inverse of the denominator of the plant. In practice, this assumption is almost never satisfied so this method usually leads to biased estimate of the parameters. If $e(k)$ is a colored noise, it will be correlated with $y(k-1)$ and the parameter estimates will be biased as well.

Instrumental Variables method

It was shown that we need two conditions for the unbiasedness of the parameter estimates of an ARX model. These two conditions may be satisfied if we replace $\phi(k)$ in (3.37) with $\phi_{iv}(k)$, which is called the vector of instrumental variables. The new estimates are given by:

$$\hat{\theta}_{iv} = \left[\sum_{k=1}^{N} \phi_{iv}(k)\phi^T(k)\right]^{-1} \left[\sum_{k=1}^{N} \phi_{iv}(k)y(k)\right] \tag{3.46}$$

Note that $\phi^T(k)$ remains unchanged. The parametric error for the new estimates is given by:

$$\begin{aligned}\tilde{\theta}_{iv} = \hat{\theta}_{iv} - \theta_0 &= \left[\sum_{k=1}^{N} \phi_{iv}(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi_{iv}(k)[\phi^T(k)\theta_0 + e(k)] - \theta_0 \\ &= \left[\sum_{k=1}^{N} \phi_{iv}(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi_{iv}(k)e(k)\end{aligned} \tag{3.47}$$

Therefore, in a similar way, the parameter estimates are asymptotically unbiased if:

1. $R_{\phi_{iv}\phi}(0)$ is not singular,
2. $R_{\phi_{iv}e}(0) = 0$.

As a result, if we select the vector of instrumental variables such that it is correlated with the regressor vector and not correlated with the noise, the parameter estimates will be asymptotically unbiased.

There are several choices of the vector of instrumental variables proposed in the literature. A very simple construction is to keep the past inputs in the vector of instrumental variables, which are not noisy, and change the past noisy outputs with another signal which is correlated with the output and not correlated with noise. A good way to generate a non noisy signal correlated with the output is to simulate the output with an initial model of the plant. Suppose that a model $M(q^{-1})$, which is close to $G(q^{-1})$ is available. This auxiliary model may be a high order FIR model or an ARX model with biased parameters obtained by (3.37). Then, a noiseless output of this model can be computed by:

$$y_M(k) = M(q^{-1})u(k) \tag{3.48}$$

and the vector of the instrumental variable can be defined as:

$$\phi_{iv}^T(k) = [-y_M(k-1), \ldots, -y_M(k-n), u(k-d), \ldots, u(k-m)] \tag{3.49}$$

This vector is very well correlated with ϕ and not correlated with noise, so it will lead to an unbiased estimate of the parameters. It should be mentioned that the

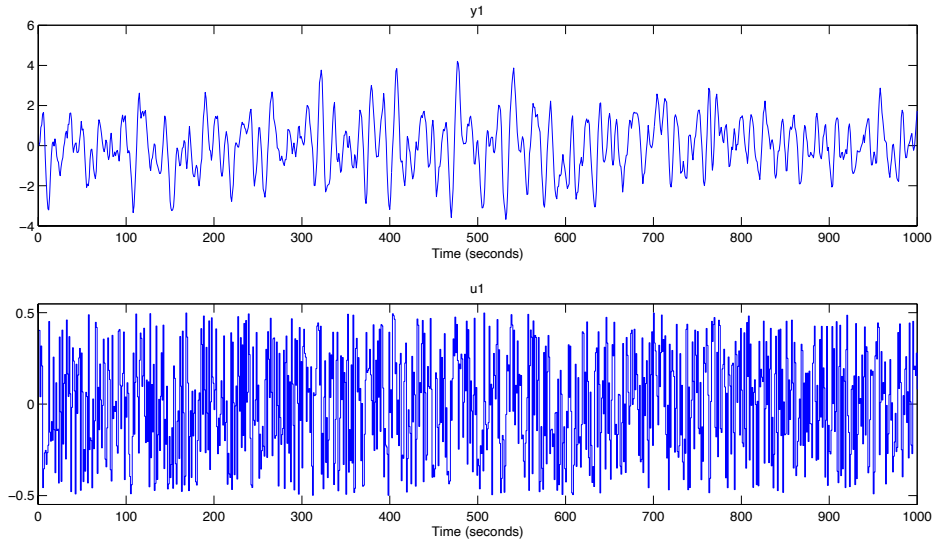


Figure 3.1: Input and output data for identification of an ARX model

variance of the estimates depends on the inverse of the cross-correlation function between $\phi_{iv}(k)$ and $\phi(k)$. Therefore, better initial model will lead to smaller variance of the estimates. An iterative method in which the IV estimates at iteration i are used to construct a new M_i, which is used to generate a new $\phi_{iv}(k)$ for the next iteration, can reduce the variance of the estimates. It should be mentioned that in the IV method no fit criterion is minimized, so if the assumption on the true model is not satisfied a very bad fit may be obtained.

Example 3.2. Consider the following second order discrete-time model:

$$G(q^{-1}) = \frac{0.5q^{-1} + q^{-2}}{1 - 1.5q^{-1} + 0.7}$$

The system is excited by a white noise with uniform distribution and variance $\sigma_u^2 = 0.08$. A zero-mean Gaussian white noise with var $\sigma_e^2 = 0.01$ is added to the output. Note that the noise is not filtered by the inverse of the denominator of the plant model, so the true system does not have an ARX structure. The input and output data are shown in Fig. 3.1. A second-order ARX model is identified using (3.37), which gives the following vector of parameters:

$$\hat{\theta} = [-1.440\,,\ 0.642\,,\ 0.510\,,\ 1.022]$$

The estimated parameters are clearly biased if compared to the true parameters:

$$\theta_0 = [-1.50\,,\ 0.70\,,\ 0.50\,,\ 1.00]$$

If the identified ARX model is used as an auxiliary model M to generate a vector of instrumental variables as proposed in (3.48) and (3.49), the following parameters are estimated based on (3.46):

$$\hat{\theta}_{iv} = [-1.504\,,\ 0.703\,,\ 0.503\,,\ 0.984]$$

which are theoretically unbiased and practically much closer to the true values.

3.2.3 State-space model

State-space models are regularly used for optimal and advanced control design methods. Any discrete-time linear system can be represented in the state-space form as:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= Cx(k) + Du(k) + e(k) \end{aligned} \tag{3.50}$$

where $x(k) \in R^{n\times 1}$ is the state vector, $A \in R^{n\times n}$ the state matrix, $B \in R^{n\times n_u}$ the input matrix, $C \in R^{n_y\times n}$ the output matrix and $D \in R^{n_y\times n_u}$ the feedthrough matrix. The system order is n, the number of inputs n_u and the number of outputs n_y. The random signals $w(k)$ and $e(k)$ are the state and output noise, respectively, with the following covariance matrix:

$$\mathbb{E}\left\{ \begin{bmatrix} w(k) \\ e(k) \end{bmatrix} \begin{bmatrix} w(k) & e(k) \end{bmatrix} \right\} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \tag{3.51}$$

A state-space model identification is to estimate A, B, C, D, Q, R, S and the order n of the system based on a finite set of input and output data. The representation in (3.50) is not unique, and there are infinite number of different structures that describe the system using the similarity transforms. A unique solution may be obtained if the state-space realization is fixed.

Assume that the sequence of the state vector is measured together with the system output. In this case, as the states are known, the state-space realization will be fixed and (3.50) becomes a linear regression:

$$Y(k) = \Theta\Phi(k) + E(k) \tag{3.52}$$

where

$$Y(k) = \begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix}, \quad \Theta = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \quad \Phi(k) = \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}, \quad E(k) = \begin{bmatrix} w(k) \\ e(k) \end{bmatrix}$$

This problem can be solved using the least squares algorithm which leads to:

$$\hat{\Theta}_i^T = \left[\sum_{k=1}^{N} \Phi(k)\Phi^T(k) \right]^{-1} \left[\sum_{k=1}^{N} \Phi(k) Y_i(k) \right] \tag{3.53}$$

where $\hat{\Theta}_i$ is the LS estimate of the i-th row of Θ and $Y_i(k)$ is the i-th row of $Y(k)$. Moreover, the covariance of noise can be computed using the model residuals:

$$\begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = \frac{1}{N} \sum_{k=1}^{N} \hat{E}(k)\hat{E}^T(k) \tag{3.54}$$

where

$$\hat{E}(k) = Y(k) - \hat{\Theta}\Phi(k) \tag{3.55}$$

Note that the covariance matrices can be used to compute the Kalman filter of a state estimator in a state feedback controller.

Subspace identification methods

The main problem, in the state-space model identification, is how to find the state vector, if they are not measured. The problem of computing the states only from the input and output data, using linear algebra tools is investigated in the subspace identification methods. It has been shown that the state vector of a system can be reconstructed by linear combinations of the components of an i-step ahead output predictor for $i = 1, 2, \ldots, n$, i.e.:

$$x(k) = L \begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \vdots \\ \hat{y}(k+n) \end{bmatrix} \tag{3.56}$$

The choice of L determine the basis for the state-space realization of the system. The predictor $\hat{y}(k+i)$ is a linear function of the past input and output data and can be efficiently computed.

There are several other variant algorithms for subspace identification methods in which the states are not explicitly computed. These variants are based on estimating the observability matrix instead. The main idea is as follows: first estimate the observability matrix from the data (by a rank test of this matrix the order of the system can be estimated); then, estimate the matrices A and C from the observability matrix; and finally, estimate B and D by the least squares method. Different steps of this algorithm are detailed in a reverse direction.

Estimation of B and D: It is assumed that $\hat{A}$ and $\hat{C}$ are available. Therefore, the following model structure for the output predictor is proposed:

$$\hat{y}(k) = \hat{C}(qI - \hat{A})^{-1}Bu(k) + Du(k) \tag{3.57}$$

which is linear in B and D. As a result the standard least squares algorithm can be used to compute $\hat{B}$ and $\hat{D}$. Since the predictor is a function of only past inputs, the regressor vector will not be noisy (in open-loop operation), and the estimates will not be biased if $\hat{A}$ and $\hat{C}$ are correct. Moreover, it can be shown that if $\hat{A}$ and $\hat{C}$ are asymptotically unbiased, or consistent, then $\hat{B}$ and $\hat{D}$ will be asymptotically unbiased as well.

For a SISO model with n states we have:

$$\hat{C}(qI - \hat{A})^{-1} = \left[F_1(q^{-1}), \cdots, F_n(q^{-1})\right] \tag{3.58}$$

therefore

$$\hat{y}(k) = b_1F_1(q^{-1})u(k) + \cdots + b_nF_n(q^{-1})u(k) + Du(k) = \phi^T(k)\theta \tag{3.59}$$

where

$$\phi^T(k) = [u_{f_1}(k), u_{f_2}(k), \ldots, u_{f_n}(k), u(k)]$$
$$\theta^T = [b_1, b_2, \ldots, b_n, D] = [B^T\,,\, D]$$

with $u_{f_i}(k) = F_i(q^{-1})u(k)$. So the vector B and the scalar D can be estimated with the classical LS algorithm. A similar procedure can be used for MIMO systems.

Estimation of A and C: It is assumed that the extended observability matrix O_r is available.

$$O_r = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix}_{rn_y \times n} \tag{3.60}$$

Then, $\hat{C}$ can be estimated immediately as the first n_y rows of O_r. Similarly, $\hat{A}$ can be computed from the following equation:

$$[\text{the last } (r-1)n_y \text{ rows of } O_r] = [\text{the first } (r-1)n_y \text{ rows of } O_r] \times \hat{A} \tag{3.61}$$

Since the rank of O_r for observability is n, there will be a unique solution for $\hat{A}$.

Estimation of the order n: Suppose that the extended observability matrix with an overestimated order $n^* > n$ is given (the number of column is n^*). Note also that if O_r is an observability matrix then O_rT is also an observability matrix for another representation of the same system, where T is a nonsingular similarity transform matrix. So the given matrix can be written as: $Q = O_r\tilde{T}$, where $\tilde{T}$ is an unknown but full rank $n \times n^*$ matrix. The easiest way to find n, is to find the rank of Q by computing its singular values using the singular value decomposition (SVD) algorithm. Q will have n^* singular values from which $n^* - n$ are zero. In practice, because of noisy data, some singular values will be very small and none of them will be zero. In this case, some threshold should be defined as a function of the noise variance. It is very usual to use some weighting matrices in order to reduce the noise effect and improve the quality of the estimates of $\hat{A}$ and $\hat{C}$. The matrix Q is pre and post-multiplied by weighting filter before performing the SVD, i.e. $\hat{Q} = W_1QW_2$.

Estimation of the observability matrix: The output of the state-space model in (3.50) can be written as:

$$\begin{aligned} y(k+i) &= Cx(k+i) + Du(k+i) + e(k+i) \\ &= CAx(k+i-1) + CBu(k+i-1) + Du(k+i) \\ &\quad + Cw(k+i-1) + e(k+i) \\ &= \dots \\ &= CA^i x(k) + CA^{i-1}Bu(k) + CA^{i-2}Bu(k+1) + \cdots \\ &\quad + CBu(k+i-1) + Du(k+i) \\ &\quad + CA^{i-1}w(k) + \cdots + Cw(k+i-1) + e(k+i) \end{aligned} \tag{3.62}$$

Now, define the following vectors constructed by known input and output data:

$$Y_r(k) = \begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+r-1) \end{bmatrix} \quad , \quad U_r(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+r-1) \end{bmatrix} \tag{3.63}$$

and the unknown noise vector :

$$V_r(k) = \begin{bmatrix} V(k) \\ V(k+1) \\ \vdots \\ V(k+r-1) \end{bmatrix} = \begin{bmatrix} e(k) \\ Cw(k) + e(k+1) \\ \vdots \\ CA^{r-2}w(k) + \cdots + Cw(k+r-2) + e(k+r-1) \end{bmatrix}$$

Then collect all equations in (3.62) as:

$$Y_r(k) = O_r x(k) + S_r U_r(k) + V_r(k) \tag{3.64}$$

where S_r is:

$$S_r = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{r-2}B & CA^{r-3}B & \cdots & D \end{bmatrix} \tag{3.65}$$

Let us define the following matrices:

$$\boldsymbol{Y} = [Y_r(1)\,,\, Y_r(2)\,,\, \ldots\,,\, Y_r(N)] \tag{3.66}$$
$$\boldsymbol{U} = [U_r(1)\,,\, U_r(2)\,,\, \ldots\,,\, U_r(N)] \tag{3.67}$$
$$\boldsymbol{V} = [V_r(1)\,,\, V_r(2)\,,\, \ldots\,,\, V_r(N)] \tag{3.68}$$
$$\boldsymbol{X} = [x(1)\,,\, x(2)\,,\, \ldots\,,\, x(N)] \tag{3.69}$$

and rewrite (3.64) as:

$$\boldsymbol{Y} = O_r \boldsymbol{X} + S_r \boldsymbol{U} + \boldsymbol{V} \tag{3.70}$$

Note that in the above equation $\boldsymbol{Y}$ is constructed from output measurements and $\boldsymbol{U}$ from the inputs and both are available, while the other matrices are unknown.

Noiseless case: In this case we assume that $\boldsymbol{V}$ is negligible. In order to remove from the equation the unknown matrix S_r, we form the $N \times N$ matrix:

$$\boldsymbol{U}^{\perp} = \boldsymbol{I} - \boldsymbol{U}^{\boldsymbol{T}}(\boldsymbol{U}\boldsymbol{U}^{\boldsymbol{T}})^{-\mathbf{1}}\boldsymbol{U} \tag{3.71}$$

which is orthogonal to $\boldsymbol{U}$, i.e. $\boldsymbol{U}\boldsymbol{U}^{\perp} = 0$. Multiplying (3.70) from the right by $\boldsymbol{U}^{\perp}$ leads to:

$$\boldsymbol{Y}\boldsymbol{U}^{\perp} = O_r \boldsymbol{X}\boldsymbol{U}^{\perp} \tag{3.72}$$

Therefore, $Q = \boldsymbol{Y}\boldsymbol{U}^{\perp}$ is available which is equal to the observability matrix multiplied by another matrix. So, by computing the rank of Q, the order of the system

n will be estimated, then the first n columns of Q will be considered as an estimate of the observability matrix, which will be used to estimate $\hat{C}$ and $\hat{A}$ and finally $\hat{B}$ and $\hat{D}$ are estimated using the least squares technique.

Noisy case: In the presence of noise, the problem becomes more complicated because the noise effect should be removed as well. This is usually performed using the instrumental variables techniques. Let's define:

$$\phi_r(k) = [\phi(k-1)\,,\,\ldots\,,\,\phi(k-r)]^T$$

not correlated with $V_r(k)$ and

$$\Phi = [\phi_r(1)\,,\,\ldots\,,\,\phi_r(N)]$$

Then multiply (3.70) by $\frac{1}{N}\boldsymbol{U}^{\perp}\Phi^T$ to obtain:

$$\tilde{Q} \equiv \frac{1}{N}\boldsymbol{Y}\boldsymbol{U}^{\perp}\Phi^T = O_r\frac{1}{N}\boldsymbol{X}\boldsymbol{U}^{\perp}\Phi^T + \frac{1}{N}\boldsymbol{V}\boldsymbol{U}^{\perp}\Phi^T = O_r\tilde{T}_N + V_N$$

Here $\tilde{T}_N$ is an $n \times r$ matrix. Suppose we can find $\phi_r(k)$ such that:

$$\lim_{N\to\infty} V_N = \lim_{N\to\infty}\frac{1}{N}\boldsymbol{V}\boldsymbol{U}^{\perp}\Phi^T = 0$$
$$\lim_{N\to\infty} \tilde{T}_N = \lim_{N\to\infty}\frac{1}{N}\boldsymbol{U}^{\perp}\Phi^T = \tilde{T} \quad \text{has full rank } n$$

Then the effect of noise will be asymptotically canceled. This can be achieved if the instrumental variable $\phi_r(k)$ is chosen as a function of past inputs and outputs which are not correlated with $V_r(k)$ that includes only the present and future values of noise signals $e(k)$ and $w(k)$. The choice of the instrumental variables lead to different algorithms with different asymptotic properties that are not discussed here.

3.3 Linear Black-Box Models

The three main steps of parametric identifications are: (1) selection of a model structure, (2) defining a parameterized output predictor and (3) minimizing the prediction error. In this section, linear discrete-time rational black-box model structures are considered and appropriate output predictors are defined. The main difference between the different structures is the assumption on the noise and the noise model. The main assumptions on the noise are:

1. The noise at the plant output is not correlated with the plant input. This assumption leads to the so called *output error* model structures.

2. The noise at the plant output is a zero-mean white noise filtered by a finite order stable filter with unknown parameters. This assumption lead to the structures with noise model.

3.3.1 Output error structure

Assume that the measured output of the plant can be expressed by the following *true* model:

$$y(k) = G_0(q^{-1})u(k) + n(k) \tag{3.73}$$

where $n(k)$ is a zero-mean stationary noise independent from the plant input $u(k)$, and $G_0(q^{-1})$ is the true plant model represented by the following finite order discrete-time model:

$$G_0(q^{-1}) = \frac{B_0(q^{-1})}{A_0(q^{-1})} \tag{3.74}$$

with

$$B_0(q^{-1}) = b^\circ_{d_0} q^{-d_0} + \cdots + b^\circ_{m_0} q^{-m_0} \tag{3.75}$$

$$A_0(q^{-1}) = 1 + a^\circ_1 q^{-1} + \cdots + a^\circ_{n_0} q^{-n_0} \tag{3.76}$$

$$\theta_0^T = [a^\circ_1, \ldots, a^\circ_{n_0}, b^\circ_{d_0}, \ldots, b^\circ_{m_0}] \tag{3.77}$$

The model structure in (3.73) and in Fig. 3.2 is known as *output error* or OE structure. In order to find an output predictor for this model structure, let us suppose,

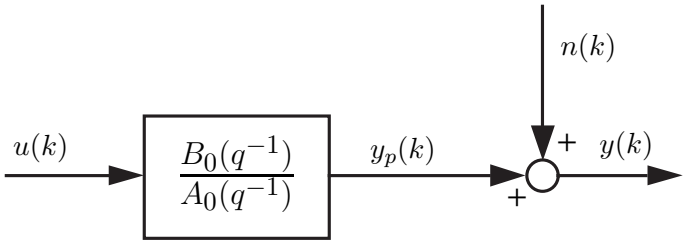


Figure 3.2: OE structure

for instant, that the true parameters are known. Therefore, a good predictor for the output of the system can be given by:

$$\hat{y}(k, \theta_0) = G_0(q^{-1})u(k) \tag{3.78}$$

where $\hat{y}(k, \theta_0)$ is the predicted output at instant k based on the true model parameters. It should be mentioned that the prediction error for this ideal predictor is

$$\varepsilon(k, \theta_0) = y(k) - \hat{y}(k, \theta_0) = n(k) \tag{3.79}$$

which, by assumption, is independent of the plant input. Now, we define a parameterized predictor as:

$$\hat{y}(k, \theta) = G(q^{-1}, \theta)u(k) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})}u(k) \tag{3.80}$$

where $G(q^{-1}, \theta)$ is a parameterized set of all candidate models with

$$B(q^{-1}) = b_d q^{-d} + \cdots + b_m q^{-m} \tag{3.81}$$

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_n q^{-n} \tag{3.82}$$

$$\theta^T = [a_1, \ldots a_n, b_d, \ldots, b_m] \tag{3.83}$$

If $d_0 = d, m_0 = m$ and $n_0 = n$, then the true model $G_0(q^{-1})$ belongs to the candidate model set, i.e. $G(q^{-1}, \theta_0) = G_0(q^{-1})$. The prediction error is a nonlinear function of the unknown parameter vector θ and is given by:

$$\begin{aligned}\varepsilon(k,\theta) = y(k) - \hat{y}(k,\theta) &= G_0(q^{-1})u(k) + n(k) - G(q^{-1},\theta)u(k) \\ &= [G_0(q^{-1}) - G(q^{-1},\theta)]u(k) + n(k)\end{aligned} \tag{3.84}$$

The minimization of the prediction error leads to a nonlinear optimization problem, which can be solved using the iterative numerical optimization methods. However, the convergence of the solution to the global optimal solution is not guaranteed.

Remark: It is clear that if $\theta = \theta_0$, the prediction error becomes equal to $n(k)$ and independent from the input. This property can be used later on for model validation.

Relation with ARX and FIR structures

The output error predictor in (3.80) can be rewritten as:

$$A(q^{-1})\hat{y}(k,\theta) = B(q^{-1})u(k-d) \tag{3.85}$$

and rearranged to obtain:

$$\begin{aligned}\hat{y}(k,\theta) = &-a_1\hat{y}(k-1,\theta) - \cdots - a_n\hat{y}(k-n,\theta) \\ &+ b_d u(k-d) + \cdots + b_m u(k-m)\end{aligned} \tag{3.86}$$

It is interesting to compare this output error predictor with the ARX predictor defined in (3.35) as:

$$\hat{y}(k,\theta) = -a_1 y(k-1) - \cdots - a_n y(k-n) + b_d u(k-d) + \cdots + b_m u(k-m)$$

The main point is that the past values of the output in the ARX predictor are replaced with the past values of the predicted output in the OE predictor. So the output error predictor depends only on the past inputs and not correlated with the measurement noise.

It is also interesting to note that the FIR structure discussed before in (3.11) as

$$\hat{y}(k,\theta) = b_d u(k-d) + \cdots + b_m u(k-m)$$

is a special case of the output error structure, where $n = 0$ or $A(q^{-1}) = 1$. In this case, the output error predictor becomes linear in parameters and the least squares algorithm can be employed to find the parameters.

3.3.2 Modeling the noise

It can be shown that if the power spectral density of a zero-mean noise signal is strictly positive in all normalized frequencies, i.e. $\phi_{nn}(\omega) > 0$, $\forall\omega \in [0\ \pi]$, then the

noise signal $n(k)$ can always be modeled by a zero-mean white noise $e(k)$ filtered by a rational finite-order stable and inversely stable transfer function:

$$n(k) = H_0(q^{-1})e(k) = \frac{C_0(q^{-1})}{D_0(q^{-1})}e(k) \tag{3.87}$$

where $C_0(q^{-1})$ and $D_0(q^{-1})$ are monic polynomials (the first coefficient is equal to 1) with all their roots strictly inside the unit circle. Considering this noise model in (3.73), we obtain:

$$y(k) = G_0(q^{-1})u(k) + H_0(q^{-1})e(k) \tag{3.88}$$

Since the noise at instant k depends on the past values of $e(k)$, they may be used in an output predictor in order to improve the prediction. To find the *predictable* part of the noise, let us write it as:

$$\begin{aligned} n(k) &= H_0(q^{-1})e(k) = [H_0(q^{-1}) - 1]e(k) + e(k) \\ &= \left[\frac{C_0(q^{-1}) - D_0(q^{-1})}{D_0(q^{-1})}\right] e(k) + e(k) \end{aligned} \tag{3.89}$$

We observe that $n(k)$ is composed of two terms. The second term, $e(k)$, is not predictable at instant $k-1$, while the first term is predictable if $H_0(q^{-1})$ is known. This can be seen by developing $C_0(q^{-1}) - D_0(q^{-1})$, which are both monic polynomials:

$$C_0(q^{-1}) - D_0(q^{-1}) = (c_1^\circ - d_1^\circ)q^{-1} + (c_2^\circ - d_2^\circ)q^{-2} + \cdots \tag{3.90}$$

We observe that the first term in (3.89) depends only on $e(k-1), e(k-2), \ldots$ and so is predictable. Therefore, the best output predictor for this model with known parameters is given as:

$$\hat{y}(k, \theta_0) = G_0(q^{-1})u(k) + [H_0(q^{-1}) - 1]e(k) \tag{3.91}$$

where θ_0 includes the parameters of the plant model $G_0(q^{-1})$ as well as the parameters of the noise model $H_0(q^{-1})$. With this ideal predictor, the prediction error will be equal to $e(k)$ and will be white too.

$$\varepsilon(k, \theta_0) = y(k) - \hat{y}(k, \theta_0) = e(k) \tag{3.92}$$

In practice, the vector of parameters θ_0 is unknown and the past values of noise, $e(k-1), e(k-2), \ldots$ are not directly measurable. Therefore, a parameterized predictor is defined, where θ_0 is replaced with the vector of unknown parameters θ and the past values of the white noise, $e(k-1), e(k-2), \ldots$, are replaced with the past values of the prediction error $\varepsilon(k-1, \theta), \varepsilon(k-2, \theta), \ldots$, which are available:

$$\hat{y}(k, \theta) = G(q^{-1})u(k) + [H(q^{-1}) - 1]\varepsilon(k, \theta) \tag{3.93}$$

The prediction error can be written as:

$$\varepsilon(k, \theta) = y(k) - \hat{y}(k, \theta) = y(k) - G(q^{-1})u(k) - [H(q^{-1}) - 1]\varepsilon(k, \theta) \tag{3.94}$$

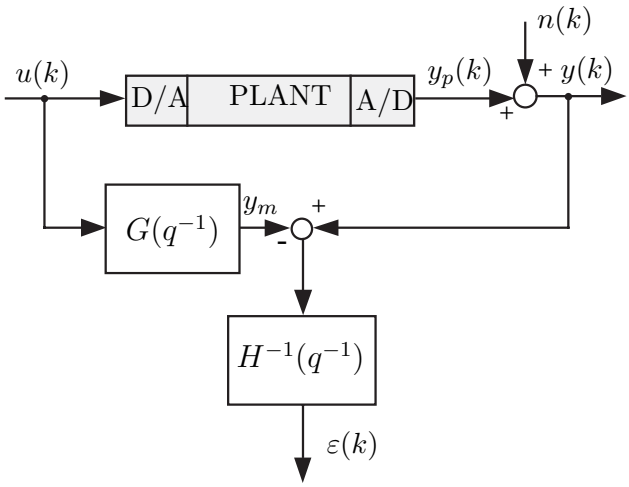


Figure 3.3: Schematic diagram of the prediction error as a function of the plant and noise model.

which leads to:

$$\varepsilon(k,\theta) = H^{-1}(q^{-1})[y(k) - G(q^{-1})u(k)] \tag{3.95}$$

Figure 3.3 shows the schematic diagram of the prediction error as a function of the plant and noise model.

By replacing $y(k)$ from (3.88) in (3.95), we obtain:

$$\begin{aligned}\varepsilon(k,\theta) &= H^{-1}(q^{-1})[G_0(q^{-1})u(k) + H_0(q^{-1})e(k) - G(q^{-1})u(k)] \\ &= H^{-1}(q^{-1})\left\{\left[G_0(q^{-1}) - G(q^{-1})\right]u(k) + \left[H_0(q^{-1}) - H(q^{-1})\right]e(k)\right\} + e(k)\end{aligned} \tag{3.96}$$

Therefore, if the true model of the plant G_0 and the true model of noise H_0 belongs the set of the candidate models G and H (i.e. the value of the time delay and the order of all polynomials are correctly chosen), for $\hat{\theta} = \theta_0$ or $G = G_0$ and $H = H_0$, we obtain $\varepsilon(k,\hat{\theta}) = e(k)$. The prediction error evaluated for the estimated parameter $\hat{\theta}$ is called *the residual* and plays an important role for model validation. In fact, the whiteness of the residual is an indicator of the good quality of the estimated parameters.

Intuitively, the residual shows the error between the measurements $y(k)$ and the best prediction $\hat{y}(k,\hat{\theta})$ based on the plant and noise model. A good model will give a good prediction of the deterministic part of the output $y_p(k)$ as well as the predictable part of the noise. If the plant model is correctly identified, i.e. $G = G_0$, the residual will be $H^{-1}(q^{-1})n(k)$ (see Fig. 3.3), which is not correlated with the input signal. Moreover, if the noise model is correctly identified, i.e. $H = H_0$, the residual will be white that shows both the plant and noise model are validated.

ARX structure

Different structures for the noise model can be considered. The simplest and the most restrictive one is to consider

$$H_0(q^{-1}) = \frac{1}{A_0(q^{-1})} \tag{3.97}$$

which leads to the ARX structure given by (see Fig. 3.4):

$$y(k) = \frac{B_0(q^{-1})}{A_0(q^{-1})}u(k) + \frac{1}{A_0(q^{-1})}e(k) \tag{3.98}$$

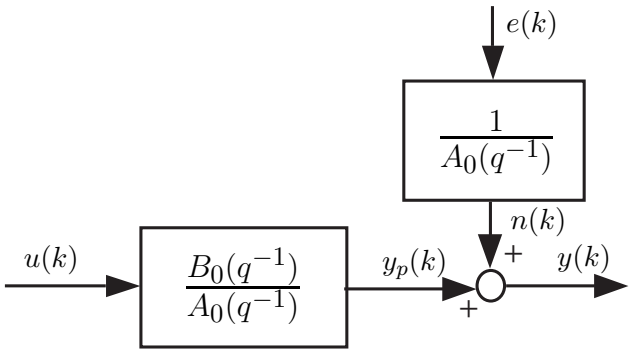


Figure 3.4: ARX structure

The ARX predictor can be obtained from the generalized predictor in (3.93) by replacing $H(q^{-1})$ with $1/A(q^{-1})$ to obtain:

$$\hat{y}(k,\theta) = G(q^{-1})u(k) + \left[\frac{1}{A(q^{-1})} - 1\right]\varepsilon(k,\theta) \tag{3.99}$$

and the following prediction error:

$$\begin{aligned}\varepsilon(k,\theta) &= A(q^{-1})[y(k) - G(q^{-1})u(k)] = A(q^{-1})y(k) - B(q^{-1})u(k)\\ &= y(k) + a_1 y(k-1) - \cdots + a_n y(k-n)\\ &\quad - b_d u(k-d) - \cdots - b_m u(k-m)\end{aligned} \tag{3.100}$$

which is linear in parameters and can be minimized, as explained before, by the least squares algorithm. If the real system is represented by an ARX model, and the true model belongs to the candidate model set, the minimization of the prediction error leads to asymptotically unbiased estimation of the parameters and the residual will be white according to (3.96).

ARMAX structure

For this structure, the noise model is an Auto-Regressive with Moving Average (ARMA) process given by:

$$H_0(q^{-1}) = \frac{C_0(q^{-1})}{A_0(q^{-1})} \tag{3.101}$$

The output of the true system is (see Fig. 3.5):

$$y(k) = \frac{B_0(q^{-1})}{A_0(q^{-1})}u(k) + \frac{C_0(q^{-1})}{A_0(q^{-1})}e(k) \tag{3.102}$$

which is called ARMAX for ARMA with eXternal input.

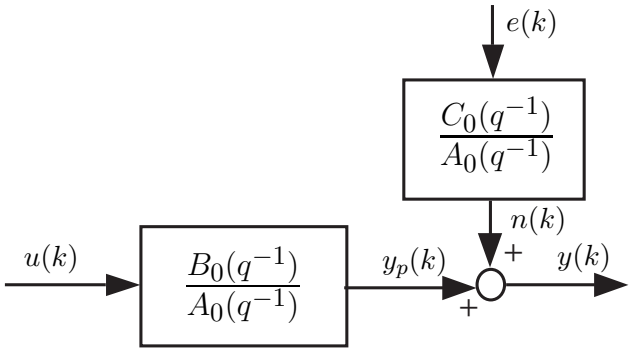


Figure 3.5: ARMAX structure

This structure is more flexible than the ARX structure and, because of the polynomial $C_0(q^{-1})$ in the noise model, has more degree of freedom to model the measurement noise at the output of the system. However, as it will be shown, the prediction error will be nonlinear in model parameters and like the OE structure the least squares algorithm cannot be used to find the parameters. In practice, nonlinear optimization methods are used with no convergence guarantee to the global optimal solution.

Since the plant and noise model share the same denominator, the ARMAX model is well suited for the systems where noise and disturbances are applied at the input of the plant.

By replacing $H(q^{-1})$ by $\dfrac{C(q^{-1})}{A(q^{-1})}$ in (3.93), the ARMAX predictor is obtained:

$$\hat{y}(k,\theta) = \frac{B(q^{-1})}{A(q^{-1})}u(k) + \left[\frac{C(q^{-1})}{A(q^{-1})} - 1\right]\varepsilon(k) \tag{3.103}$$

The prediction error is then computed by:

$$\begin{aligned}\varepsilon(k,\theta) &= \frac{A(q^{-1})}{C(q^{-1})}\left[y(k) - \frac{B(q^{-1})}{A(q^{-1})}u(k)\right] \\ &= \frac{1}{C(q^{-1})}[A(q^{-1})y(k) - B(q^{-1})u(k)]\end{aligned} \tag{3.104}$$

Assume that the real system is represented by an ARMAX model, and the true model belongs to the candidate model set, i.e. $d = d_0$ and the orders of A, B and C are respectively equal to the orders of A_0, B_0 et C_0. Then, the minimization of the prediction error leads to asymptotically unbiased estimation of the parameters and the residual will be white according to (3.96).

3.3.2.1 Box-Jenkins structure

The structure proposed by Box and Jenkins and shown in (3.6) is more general than the other structures, since the plant model and noise model are independently parameterized (they do not share any common parameter). Therefore, there is more

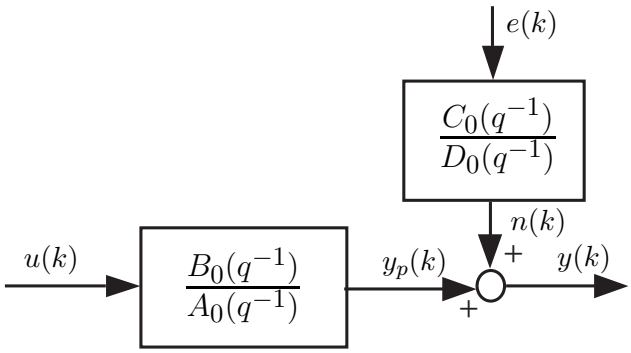


Figure 3.6: Structure Box-Jenkins

degree of freedom to model any plant and noise model with the cost of increasing the number of the parameters to estimate. The output of the BJ model (true system) is given by:

$$y(k) = \frac{B_0(q^{-1})}{A_0(q^{-1})}u(k) + \frac{C_0(q^{-1})}{D_0(q^{-1})}e(k) \tag{3.105}$$

The BJ predictor can be obtained by replacing

$$H(q^{-1}) = \frac{C(q^{-1})}{D(q^{-1})}$$

in (3.93) as

$$\hat{y}(k,\theta) = \frac{B(q^{-1})}{A(q^{-1})}u(k) + \left[\frac{C(q^{-1})}{D(q^{-1})} - 1\right]\varepsilon(k,\theta) \tag{3.106}$$

The prediction error is given by

$$\varepsilon(k,\theta) = \frac{D(q^{-1})}{C(q^{-1})}\left[y(k) - \frac{B(q^{-1})}{A(q^{-1})}u(k)\right] \tag{3.107}$$

which is nonlinear in parameters. Minimizing the prediction error should be performed by nonlinear optimization algorithms with the risk of converging to a local minimum.

3.3.3 Bias analysis in the frequency domain

In the previous section, we always assumed that the true models G_0 and H_0 belong to the set of candidate models. This assumption is violated in practice, when using data from real physical systems that have very high orders (theoretically of infinite

order). In this case, the parametric convergence or parametric error do not make sense any more and the error should be characterized in the frequency domain. Since a perfect matching between the identified and true model is not possible, the modeling error will be distributed in the frequency domain, i.e. in some frequency zones we have better approximation while in some other frequencies the error may be large. In this section, we will show that the distribution of the modeling error in the frequency domain, which is called *the bias distribution*, is affected by the noise model and the data filter.

The bias distribution is obtained by the frequency-domain interpretation of the sum of squares of the prediction error in (3.1), which is minimized in the parametric identification procedure. This will be performed using the Parseval's relation, which shows the equality of the signal energy in the time domain and in the frequency domain. According to the Parseval's relation:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} x^2(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{xx}(\omega) d\omega \tag{3.108}$$

This relation can be easily proved using the relation between the autocorrelation function and the power spectral density function (Fourier and inverse Fourier transform pair):

$$\Phi_{xx}(\omega) = \sum_{h=-\infty}^{\infty} R_{xx}(h) e^{-j\omega h} \quad , \quad R_{xx}(h) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{xx}(\omega) e^{j\omega h} d\omega$$

If we evaluate the second relation for $h = 0$, the Parseval's relation in (3.108) will be obtained. Note that this relation is valid only asymptotically, when the number of data goes to infinity.

Now, let us apply the Parseval's relation to the prediction error:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \varepsilon^2(k,\theta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{\varepsilon\varepsilon}(\omega) d\omega \tag{3.109}$$

For a large enough finite number of data N, minimizing the left hand side and the right hand side will give the same results, i.e.:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{k=1}^{N} \varepsilon^2(k,\theta) = \arg\min_{\theta} \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{\varepsilon\varepsilon}(\omega) d\omega \tag{3.110}$$

Therefore, the bias distribution can be obtained by studying the minimum of the integral of the spectrum of the prediction error. Note that multiplication by $1/N$ will not change the minimum of the criterion.

Bias distribution for OE structure: The prediction error for the output error structure is given by

$$\varepsilon(k,\theta) = [G_0(q^{-1}) - G(q^{-1})]u(k) + n(k) \tag{3.111}$$

Using the relation in (2.89), we can write:

$$\Phi_{\varepsilon\varepsilon}(\omega) = \left|G_0(e^{j\omega}) - G(e^{j\omega})\right|^2 \Phi_{uu}(\omega) + \Phi_{nn}(\omega) \tag{3.112}$$

Therefore, the vector of the parameter estimates can be written as:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[|G_0(e^{j\omega}) - G(e^{j\omega})|^2 \Phi_{uu}(\omega) + \Phi_{nn}(\omega)\right] d\omega \tag{3.113}$$

This expression clearly shows that:

- The minimum of the criterion is obtained for $\hat{\theta} = \theta_0$, if G_0 and G have the same structure (the estimates are asymptotically unbiased). Note that $\Phi_{uu}(\omega)$ and $\Phi_{nn}(\omega)$ are both positive for all ω and the only way to minimize the integral is to have $G_0 = G$. Note also that $\Phi_{nn}(\omega)$ is not a function of θ and has asymptotically no influence on the estimates.

- If G and G_0 have different structure and the possibility of $G = G_0$ is discarded, the modeling error $|G_0 - G|^2$ is better minimized in the zone where the spectrum of the input signal, $\Phi_{uu}(\omega)$, is larger. If the input signal is a white noise with flat spectrum, we will have a uniform weighting in all frequencies.

- In order to reduce the modeling error in a specific frequency zone (e.g. low frequencies or around the crossover frequency, etc.), it is sufficient to filter the input and the output data by an appropriate filter $F(q^{-1})$ before applying the optimization procedure. In this case, the vector of the parameter estimates becomes:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2\pi} \int_{-\pi}^{\pi} |F(e^{j\omega})|^2 \left[|G_0(e^{j\omega}) - G(e^{j\omega})|^2 \Phi_{uu}(\omega) + \Phi_{nn}(\omega)\right] d\omega$$

Note that the main advantage of the OE structure is that the bias distribution is completely controlled by the excitation signal and if necessary an additional filter.

Bias distribution for structures with noise model: Now, consider the prediction error for the structures with noise model:

$$\varepsilon(k,\theta) = H^{-1}(q^{-1}) \left\{ \left[G_0(q^{-1}) - G(q^{-1})\right] u(k) + \left[H_0(q^{-1}) - H(q^{-1})\right] e(k) \right\} + e(k)$$

In a similar way, by applying the Parseval's relation, we obtain:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2\pi} \int_{-\pi}^{\pi} |H^{-1}(e^{j\omega})|^2 [|G_0(e^{j\omega}) - G(e^{j\omega})|^2 \Phi_{uu}(\omega) + |H_0(e^{j\omega}) - H(e^{j\omega})|^2 \Phi_{ee}(\omega)] d\omega \tag{3.114}$$

From this expression we can conclude:

- If G_0 and H_0 have the same structure as G and H, respectively, an asymptotically unbiased estimate of the parameters will be obtained ($\hat{\theta} = \theta_0$).

- If G_0 and H_0 do not belong to the candidate model sets for the plant and noise model, the plant modeling error $|G_0 - G|^2$ will be small in the frequency region where $\Phi_{uu}(\omega)$ and $|H^{-1}(e^{j\omega})|$ are both large. The inverse of the noise model, thus, plays the role of a data filter for weighting the modeling error. Since the frequency response of this filter is not known a priori, it may have an undesired effect on the bias distribution. For example, in the ARX structure the bias distribution is filtered by $|H^{-1}|^2 = |A|^2$ which is typically a high pass filter. As a result, the ARX structure using a reduced order model leads to an identified model with smaller modeling error in high frequencies. This undesired effect can be overcome by a second identification using the data filtered by $H(q^{-1})$. This way, the inherent effect of filtering by the inverse of the noise model is cancelled and the biased distribution will become close to that of the OE structure. This technique is called identification with *focus on simulation* in MATLAB.

- If G_0 and G have the same structure but not H_0 and H, an unbiased estimation of the plant model parameters can be obtained if the plant and noise models are independently parameterized (e.g. Box-Jenkins structure). For ARX and ARMAX structure, a structural error in the noise model introduces a bias in the plant model because of common parameters, polynomial $A(q^{-1})$, in the noise and the plant model.

- If G_0 and H_0 do not have the same structure as G and H, respectively, then the plant model G is usually better identified than the noise model H because the signal to noise ratio is often large in a good identification setting.

3.3.4 Closed-loop identification

The assumptions on noise characteristics in identification algorithms are usually based on open-loop experiments. It was discussed in Section 2.3.4 that in some applications an open-loop experiment may not be possible or too costly and the data from a closed-loop experiment should be used for system identification. For nonparametric frequency-domain identification algorithm, two cases were studied. It was shown that in the absence of the external excitation signal, the spectral analysis leads to the identification of the inverse of the controller instead of the plant model. In the presence of an external excitation signal a specific method based on spectral analysis was proposed. In this section, the parametric identification of the plant model in closed-loop operation in the same two cases are studied.

Identification with no external excitation signal

Consider the closed-loop system in Fig. 3.7, where $(y_c(k) = 0, n(k) \neq 0, \forall k)$. Since

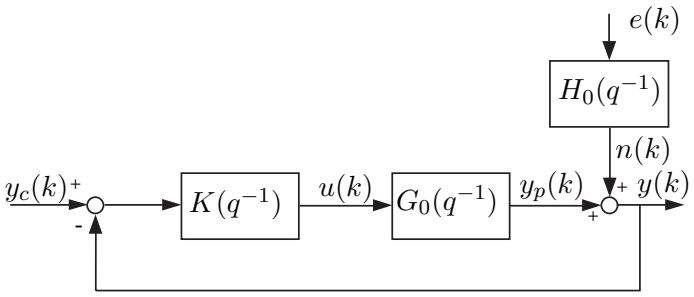


Figure 3.7: Schematic diagram of a system in closed-loop operation

there is no external excitation signal, the noise will be the only signal that excites the system. Apart from the problem of the correlation between the input of the plant and the measurement noise, which will be discussed later, we will have the problem of singularity of the information matrix if the controller is not sufficiently complex. Consider for example a proportional controller $K(q^{-1}) = K_p$ and a first-order ARX structure for the true model as follows:

$$(1 + a_1^\circ q^{-1})y(k) = b_1^\circ u(k-1) + e(k) \tag{3.115}$$

The output predictor is given as:

$$\begin{aligned}\hat{y}(k) &= -a_1 y(k-1) + b_1 u(k-1) \\ &= [-y(k-1) \quad u(k-1)] \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \phi^T(k)\theta\end{aligned} \tag{3.116}$$

The information matrix is:

$$\sum_{k=1}^{N} \phi(k)\phi^T(k) = \sum_{k=1}^{N} \begin{bmatrix} y^2(k-1) & -y(k-1)u(k-1) \\ -u(k-1)y(k-1) & u^2(k-1) \end{bmatrix} \tag{3.117}$$

The control signal $u(k)$ is

$$u(k) = -K_p y(k) \tag{3.118}$$

which leads to the following information matrix:

$$\sum_{k=1}^{N} \begin{bmatrix} y^2(k-1) & K_p y^2(k-1) \\ K_p y^2(k-1) & K_p^2 y^2(k-1) \end{bmatrix} \tag{3.119}$$

It is evident that the information matrix is singular as the second row of the matrix is a multiple of the first row. As a result, the parameters of the plant model cannot be identified. It can be shown that a sufficient condition for the identifiability of the plant model parameters is that the controller order is greater than or equal to that of the plant model. This condition is rarely met in industrial applications, where simple PID controllers are used for complex high order systems. A practical solution is change the controller parameters slowly around their nominal values during the data acquisition in order to remove the singularity of the information matrix.

Identification with external excitation

Consider again the closed-loop system in Fig. 3.7, which is excited with the reference signal $y_c(k)$. The same effect can be obtained if the excitation is added to the input signal. However, in practice, the first variant (excitation of $y_c(k)$) is preferred because its effect on the output is predictable (the output will follow the reference signal), while the effect of a signal added to the control input may have undesirable effect on the system (saturation, large input, too small excitation, etc.).

There exist two methods for identification of the plant model in closed-loop operation: direct and indirect methods.

Directe method: In this method, the system is excited with an external signal and the input of the plant $u(t)$ and its output $y(t)$ are measured. Then, an open-loop identification method is applied directly on the closed-loop data.

OE structure: Because of the correlation between the input signal $u(t)$ and noise, the main assumption for the OE structure is not satisfied and this structure can lead to biased estimates and we cannot expect that the residual be independent from the input even if the correct parameters are identified.

Modeling the noise: Even in closed-loop operation, the assumption that, the output noise can be represented as a filtered white noise, remains a valid assumption. The only difference is that, the noise will be also filtered by the closed-loop sensitivity function. Therefore, if the open-loop noise model is H_0, then the closed-loop noise model will be $H_0(1 + KG_0)^{-1}$. As a result, the assumption that the true noise mode belongs to the candidate noise model H will be unlikely satisfied with a low-order noise model. On the other hand, it can be shown that in closed-loop operation a bias in the noise model can lead to a bias in the plant model even if they are independently parameterized (like BJ structure). In practice, if the noise variance is not large, the open-loop method can be used for direct identification and may lead to a valid model if the order of the noise model is sufficiently large.

Remark: In open-loop identification the order of the noise model is supposed to be equal the order of the plant model. However, in closed-loop operation, it is reasonable to take the order of the noise model twice that of the plant model.

IV method: The instrumental variables method, discussed in Section 3.2.2, is very well suited for direct identification of the plant model in closed-loop operation. A good choice for the vector of instrumental variables is:

$$\phi_{IV}^T(k) = [-y_M(k-1) \ldots - y_M(k-n)\ u_M(k-d) \ldots u_M(k-m)] \quad (3.120)$$

where $y_M(k)$ and $u_M(k)$ are noise-free simulated data generated by two auxiliary models as follows:

$$y_M(k) = \mathcal{T}(q^{-1})y_c(k) \quad , \quad u_M(k) = \mathcal{U}(q^{-1})y_c(k)$$

In the above equations, $\mathcal{T}$ is the model between the reference signal y_c and the output y identified by an open-loop type identification. In the same way, $\mathcal{U}$ is the model between the reference signal y_c and the plant input u identified by an open-loop type identification. It is clear that y_M and u_M are not correlated with noise and strongly correlated with y and u, respectively. Therefore, the instrumental variable vector in (3.120) is not correlated with noise but correlated with the regressor vector of the ARX structure.

Indirecte method: In this approach, the transfer function $\mathcal{T}(q^{-1})$ is first identified by an open-loop black-box model structure. With the assumption that the controller, $K(q^{-1})$, is perfectly known, the plant model can be computed as follows:

$$\mathcal{T}(q^{-1}) = \frac{y(k)}{y_c(k)} = \frac{K(q^{-1})G_0(q^{-1})}{1 + K(q^{-1})G_0(q^{-1})} \tag{3.121}$$

and

$$G(q^{-1}) = \frac{\mathcal{T}(q^{-1})}{K(q^{-1})[1 - \mathcal{T}(q^{-1})]} \tag{3.122}$$

It should be mentioned that the order of $\mathcal{T}(q^{-1})$ is generally higher than that of the plant model $G_0(q^{-1})$. Therefore, the plant model computed by the relation in (3.121) has a very high order and in particular, contains the zero/pole cancellation.

3.4 Model validation

The identification procedure will lead to the best model taking into account the available data, the selected model structure and the convergence of the optimization algorithm to the global optimum. Now the question is if this *best* model is sufficiently good or more specifically:

- Is the identified model in accordance with a new set of experimental data?
- Is the identified model appropriate for the final objective?
- Does the identified model describe the *true* system?

Although the third question is very interesting, the second one is the most important in practice. In fact, if a simple model is sufficient for our aim, there is no need to look for a more complicated one, which describes better the true system. As a result, the first validation test is to see if the objectives can be achieved with the identified model. For example, if the objective of model identification is to design a controller to achieve a certain closed-loop performance, and the identified model meets this condition, we can validate the model. In some applications, however, it may be very costly or dangerous to validate a model in this way.

The experimental data are often used to validate a model. It is suggested that a new set of data, which has not been used for identification, is employed for validation.

If the identified model is coherent with the new set of data, the model is validated. The coherence can be assessed in different domains, namely, time-domain, frequency-domain and statistic domain. The main reason to use a new set of data is to avoid to fit a model to a specific noise realization. A good model, should fit the deterministic part of the data for any noise realization.

The measured data are usually divided in two parts, while one part is used for model identification and the other for model validation. If the number of data is limited, we may prefer to use all data for identification and reduce the covariance of the estimates. In this case, the same data can be also used for validation. If the model is identified with minimizing a time-domain criterion, frequency-domain validation and statistical validation can be done with the same set of data.

3.4.1 Time-domain validation

In this method, the output of the model is compared with the measured output when the same input is applied to the system and the identified model. This error is called *the output error* and is defined as:

$$\varepsilon_s(k) = y(k) - y_m(k, \hat{\theta}) \tag{3.123}$$

where

$$y_m(k, \hat{\theta}) = \frac{\hat{B}(q^{-1})}{\hat{A}(q^{-1})} \tag{3.124}$$

Note that $y_m(k, \hat{\theta})$ is equal to the predicted output $\hat{y}(k, \hat{\theta})$ only for the output error structure. For the structures with noise model (ARX, ARMAX, BJ), the predicted output depends also on the noise model, while $y_m(k, \hat{\theta})$ depends only on the plant model.

In order to compare several models in terms of their output error, a percentage fit criterion can be defined as:

$$\text{fit} = 100 \left(1 - \frac{\|y - y_m\|}{\|y - \bar{y}\|} \right) \tag{3.125}$$

where $\bar{y}$ is the mean value of y and the two-norm is defined as:

$$\|y - y_m\| = \sqrt{\frac{1}{N} \sum_{k=1}^{N} [y(k) - y_m(k, \hat{\theta})]^2} \tag{3.126}$$

A model can be also validated in terms of its ability in prediction. In this case, the prediction error (instead of the output error) will be compared with the measured output. This validation test makes sense if the model is used for prediction, e.g. in model predictive control. Otherwise, for model-based controller design, the output error is more important than the prediction error for validation.

3.4.2 Frequency-domain Validation

The validation data can be used to identify the frequency response of the system with no assumption on the model order. Therefore, this model can be used to validate the selected order of a parametric model and its frequency response.

Suppose that $G(e^{j\omega})$ is the frequency response of the system identified by Fourier or spectral analysis of the validation data. The frequency response of the parametric model to be validated is $\hat{G}(e^{j\omega}, \hat{\theta})$. Therefore, the parametric model is validated if the following norm is small:

$$\|G - \hat{G}\| = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left|G(e^{j\omega_k}) - \hat{G}(e^{j\omega_k}, \hat{\theta})\right|^2} \tag{3.127}$$

A percentage fit criterion can be also used for comparison of several models.

Note that the norm is computed in the complex plan and not in magnitude and phase as usually observed in the Bode diagram. Therefore, the Bode diagram of $G(e^{j\omega})$ and $\hat{G}(e^{j\omega}, \hat{\theta})$ can also be compared visually to see in which frequency zones the parametric model is close to the frequency-domain data. So it is possible that a model which leads to a better fit is discarded because in the frequency zone of interest (e.g. around the crossover frequency) it leads to a worse fit.

3.4.3 Statistical validation

It has been shown in Section 3.3 that the residual, $\varepsilon(k, \hat{\theta})$, which is the prediction error evaluated at the estimated parameter $\hat{\theta}$, plays an important role in the model validation. In fact, for the output error (OE) structure, the residual will be independent of the past inputs if the estimated parameters are equal to the true ones. For the structures with noise model (ARX, ARMAX, BJ), the residual will be uncorrelated with the past inputs if the plant model is correctly identified and will be white if the parameters of the plant and noise model are correctly identified. As a result, the uncorrelation test between the residual and the past inputs can be used as a measure for validating the plant model. On the other hand, a whiteness test of the residual can help validating the noise model as well.

Uncorrelation test: The residual is defined as:

$$\varepsilon(k, \hat{\theta}) = y(k) - \hat{y}(k, \hat{\theta}) \tag{3.128}$$

If the plant model is correctly identified, the residual will contain no information from the past inputs and should be uncorrelated with them. If two signals are uncorrelated, their cross-correlation function should be zero for all h. An uncorrelation test is to compute the cross-correlation function $R_{\varepsilon u}(h)$ and check if it is zero everywhere. Since, the exact value of the cross-correlation function cannot be computed

based on a finite number of data, an unbiased estimate can be computed as:

$$\hat{R}_{\varepsilon u}(h) = \frac{1}{N-|h|} \sum_{k=\max(h,0)}^{N-1+\min(h,0)} \varepsilon(k,\hat{\theta})u(k-h) \quad , \quad h = 0, \pm 1, \pm 2, \ldots, \pm h_{\max}$$

where $h_{\max}$ is the maximum value of h, which is chosen typically less than 25. In practice, the cross-correlation function is a random variable and will not be equal to zero. However, if the residual is uncorrelated with the past inputs, it will have a mean value equal to zero, i.e $\mathbb{E}\{\varepsilon(k,\hat{\theta})u(k-h)\} = 0$. As a result, using the Central Limit Theorem, $\sqrt{N-|h|}\hat{R}_{\varepsilon u}(h)$ will have a Gaussian distribution for sufficiently large values of $N-|h|$. Based on the same Theorem, the variance of this distribution is $\sigma^2 = R_{uu}(0)R_{\varepsilon\varepsilon}(0)$. Therefore, a confidence interval of $\pm 2\sigma$ around zero can be computed and $\sqrt{N-|h|}\hat{R}_{\varepsilon u}(h)$ lays inside the interval with a probability of 95%.

Usually, a new normalized random variable (with Gaussian distribution and unity variance) is defined and verified if the new random variable is in the interval $[-2\,,\,2]$:

$$-2 \leq \frac{\sqrt{N-|h|}\hat{R}_{\varepsilon u}(h)}{\sqrt{\hat{R}_{\varepsilon\varepsilon}(0)\hat{R}_{uu}(0)}} \leq 2 \qquad \text{for } -25 \leq h \leq 25, N > 100 \tag{3.129}$$

Note that if there is a correlation between the residual and the past inputs ($h > 0$), the order of the model should be increased to reduce this correlation. However, if there is a correlation between the residual and the future inputs ($h < 0$), it may show the existence of a feedback in the system (a data collection from a closed-loop operation).

Remark: This test can be used for all structures (OE, ARX, ARMAX, BJ) to validate the plant model. It gives no information about the validation of the noise model.

Whiteness test: If the parameters of the plant model and those of the noise model are correctly identified, then the residual will be white. The whiteness of the residual can be tested by computing its autocorrelation function $R_{\varepsilon\varepsilon}(h)$. If a signal is white, its auto correlation function will be zero for all $h \neq 0$. In practice, an unbiased estimate of the autocorrelation function can be computed as:

$$\hat{R}_{\varepsilon\varepsilon}(h) = \frac{1}{N-h} \sum_{k=h}^{N-1} \varepsilon(k)\varepsilon(k-h) \quad , \quad \hat{R}_{\varepsilon\varepsilon}(-h) = \hat{R}_{\varepsilon\varepsilon}(h) \quad , \quad h = 0, 1, \ldots, h_{\max}$$

This estimate will be a zero-mean random variable for all $h \neq 0$, if the residual is white (i.e. $\mathbb{E}\{\varepsilon(k)\varepsilon(k-h)\} = 0$). Therefore, based on the Central Limit Theorem, $\sqrt{N-h}\hat{R}_{\varepsilon\varepsilon}(h)$ will have a Gaussian distribution with variance $\sigma^2 = R^2_{\varepsilon\varepsilon}(0)$. In a similar way, the 95% confidence interval for a new normalized random variable can be defined as:

$$-2 \leq \frac{\sqrt{N-h}\hat{R}_{\varepsilon\varepsilon}(h)}{\hat{R}_{\varepsilon\varepsilon}(0)} \leq 2 \qquad \text{for } 1 \leq h < 25, N > 100 \tag{3.130}$$

If the residual is white, the probability that it lays in the interval is 95%. Note that this validation test cannot be interpreted as "if the residual is inside the interval, it is white with 95% probability".

Remark: This test makes sense only for the structures with a noise model (ARX, ARMAX, BJ) and validate the correctness of the noise model if the plant model has already been validated. For the structures without a noise model (FIR, OE), this test is irrelevant.

Variance of the estimates: When a parametric model is identified, the covariance matrix of the estimates can also be computed. Under some mild assumptions, it can be shown that the estimates have a Gaussian distribution so the confidence intervals for the parameters can be computed. When comparing different models for validation purpose, it is of interest to look at the variance of the parameters. Among several validated models, the best one is the one with the smallest variance.

This is interesting to check the variance of a_n and b_m and compute $a_n \pm 2\sigma$ and $b_n \pm 2\sigma$. If zero belongs to these intervals, it shows an overestimation of n and m, respectively.

3.5 Practical Aspects of Identification

A successful system identification depends on certain a priori choices that play an important role on the quality of the identified models. These choices are summarized here:

- Choice of sampling period.
- Choice of input signal in terms of its frequency contents, its amplitude, its energy etc.
- Choice of the model structure (linear, nonlinear, time-varying, etc.), its representation (state-space, continuous-time, discrete-time, etc.) and its order.

The pre-treatment of the data before an optimization and model validation are also some other important issues for a successful identification. These aspects are discussed from a practical point of view in this section.

3.5.1 Choice of sampling period

The sampling period T_s should be carefully chosen before performing an experiment on the system, because it cannot be reduced after data acquisition. However, it is always possible to double or triple the sampling period by dropping some samples. This technique is called undersampling.

An upper bound for T_s is given by the Shannon theorem as $T_s < \pi/\omega_{\max}$ (or $\omega_s > 2\omega_{\max}$), where $\omega_{\max}$ is the highest frequency at which the spectrum of the

output signal is greater than zero. In practice, it is recommended to use an anti-aliasing filter (a low-pass continuous-time filter with a cut-off frequency less than the Nyquist frequency $\omega_N = \omega_s/2$), before sampling. Note that in the case of undersampling, and only in this case, the anti-aliasing filter may be a digital filter. The value of T_s depends essentially to the dominant time-constant of the system to be identified and the numerical problems associated to too small sampling periods. These points are elaborated in what follows.

Dominant time-constant: The maximum frequency, $\omega_{\max}$, of a system is considered usually equal to 10 times the bandwidth frequency, ω_b, of the system. The reason is that at this frequency, we have an attenuation of -3dB and one decade farther at least -20dB will be added to this attenuation (if the system is of first order). Therefore, the gain of the system can be considered sufficiently small around $\omega_{\max} = 10\omega_b$. The sampling frequency is usually chosen 2 to 3 times $\omega_{\max}$ or 20 to 30 times ω_b:

$$20\omega_b < \omega_s < 30\omega_b \tag{3.131}$$

Since $\omega_b = 1/\tau$, where τ is the dominant time-constant of the system, the interval of the sampling time is computed roughly as follows:

$$\frac{2\pi\tau}{30} < T_s < \frac{2\pi\tau}{20} \quad \Rightarrow \quad \frac{\tau}{5} < T_s < \frac{\tau}{3} \tag{3.132}$$

Another useful rule to choose the sampling time is to select it such that we have 5 to 10 samples during the rise time, T_r, of the step response of the system:

$$\frac{T_r}{10} < T_s < \frac{T_r}{5} \tag{3.133}$$

In system identification for control purpose, the bandwidth frequency ω_b is usually the desired closed-loop bandwidth. If this value is not known, it is usually taken equal to or slightly greater than the bandwidth of the plant model. For simulation purpose smaller sampling time is preferred.

Numerical problems: Choosing a too small sampling period can lead to different types of numerical problems. The poles of a discrete-time transfer function will converge to 1 when T_s converges to zero (because $z = e^{T_s s}$). As a result, all poles and zeros of the system will be too close to 1 and there will be some approximate zero pole cancellation in the model. This makes the control design for such systems very sensitive to the numerical problems. Moreover, by increasing the sampling frequency the high frequency output noise will also be sampled (a small sampling frequency has a filtering effect on noise).

Example 3.3. Consider a first order model with $\tau_1 = 50s$ sampled with a sampling period of $T_s = 10s$. The discrete-time pole of the system will be $e^{-1/5} = 0.8187$. The same system with a sampling time of $T_s = 0.1s$ will lead to a pole at $e^{-0.1/50} = 0.9980$. Now, consider a second pole with $\tau_2 = 75s$ (50% larger), then the new pole in

discrete-time becomes $e^{-0.1/75} = 0.9987$. Therefore, two completely distinct poles in the continuous-time model, will give a difference of less than 1% in the poles of the discrete-time system. This shows that the discrete-time model will loose its sensitivity to the parameters of the continuous time model.

3.5.2 Pretreatment of data

The quality of data acquisition can be adversely affected by several artifacts: fault or drift in the sensors, random parasites, fault in coding (one bit of the digital converters jumps randomly during one or more sampling periods), data loss during data recording, fault in actuators, external disturbances during the experiment, etc. Therefore, the data should be carefully inspected and all artifacts should be removed from the data before starting the identification procedure. It should be mentioned that none of the identification algorithms create correct *information* from the artifacts (garbage in → garbage out).

After removing all artifacts and aberrant points, the preprocessing of data as it will be explained, will improve significantly the quality of the identified models.

Data filtering

The experimental data have often the low frequency disturbances like drift, trends or periodic variations as well as high frequency measurement noise. Although, these disturbances can be modeled by choosing appropriate noise models in parametric identification methods, it is of interest to remove them before applying an identification algorithm. For example, a low-frequency sinusoidal disturbance can be modeled by a second order noise model, however, this will increase the number of parameters in the identified model and reduces the accuracy. If this periodic disturbance is removed before identification a simpler noise model will be sufficient.

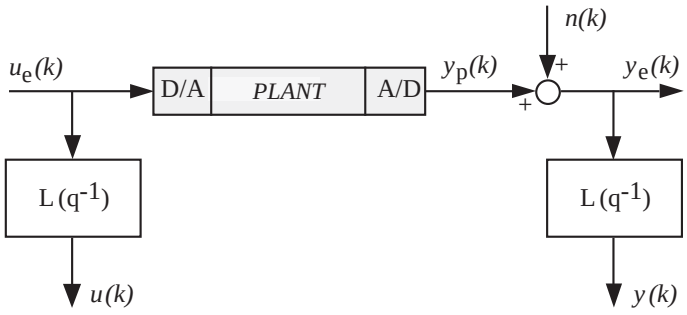


Figure 3.8: Filtering the experimental data before identification

In general, it is convenient to filter the data by a band-pass filter to remove the low-frequency disturbances and high-frequency measurement noise. The bandwidth of this filter depends on the frequency zone of interest for modeling.

Removing the mean value: The linear model of a system, considers only the variations around an operating point. Therefore, it is absolutely recommended to remove the contribution of the operating point from the data, which is often a constant or a DC offset.

From a numerical point of view, the main interest of removing the DC offset, is a better initialization of the algorithms. The regressor vector, $\phi(k)$, includes the past values of the measurements and the initial values like $u(-1), u(-2), y(-1), y(-2)$, etc. are usually initialized to zero. This means, it is inherently assumed that the operating point of the system is around zero. If the DC offset is not removed, the truncation error will be increased for frequency-domain identification and a large initial error will be introduced in the parametric models. This error, however, is reduced for large number of data N.

If the operating point is known as $(\bar{u}, \bar{y})$, these values can be subtracted from the measurements to find only the variations of the input and output signals:

$$u(k) = u_e(k) - \overline{u} \qquad y(k) = y_e(k) - \overline{y} \tag{3.134}$$

where, $u_e(k)$ and $y_e(k)$ are the experimental measured values. If the operating point $(\bar{u}, \bar{y})$ is not a priori known, it can be estimated using the mean value of the measurements:

$$\overline{u} = \frac{1}{N}\sum_{k=1}^{N} u_e(k) \quad , \quad \overline{y} = \frac{1}{N}\sum_{k=1}^{N} y_e(k) \tag{3.135}$$

For online identification using the recursive algorithms, the mean value can be estimated recursively as:

$$\overline{u} = \overline{u}(k-1) + \frac{1}{k}[u_e(k) - \overline{u}(k-1)] \tag{3.136}$$

$$\overline{y} = \overline{y}(k-1) + \frac{1}{k}[y_e(k) - \overline{y}(k-1)] \tag{3.137}$$

Removing the trends: The data may have a linear or polynomial trend. A linear trend can be observed in many systems and is shown in Fig. 3.9. In this case, the operating point is considered as time varying and the parameters of the linear trend, i.e. β_0 and β_1, can be computed by a linear regression:

$$\overline{y}(k) = \beta_0 + \beta_1 k \tag{3.138}$$

In a recursive algorithm, the time varying operating point can be estimated online and a forgetting factor, $\lambda < 1$, can be used to reduce the effect of old data:

$$\overline{y}(k) = \lambda \overline{y}(k-1) + (1-\lambda) y_e(k) \tag{3.139}$$

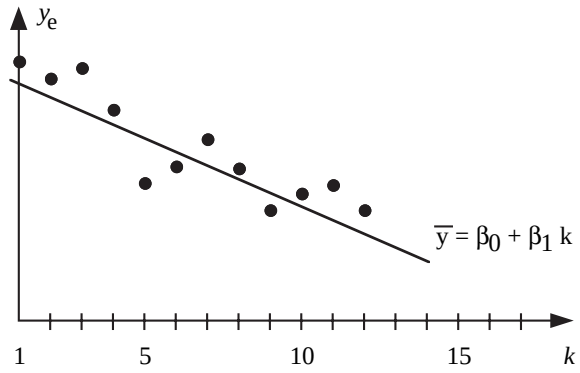


Figure 3.9: Linear trend in measured data

Removing low-frequency disturbances: In general, a high-pass filter can be used to remove all low-frequency disturbances in the data. The filter is applied to the input and the output data as it is shown in Fig. 3.8. A simple first-order discrete-time high-pass filter is given by:

$$L(q^{-1}) = \frac{1 - q^{-1}}{1 - \alpha q^{-1}}$$

where $\alpha = e^{-\omega_f T}$. Using this filter, the disturbances with a lower frequency than ω_f will be attenuated. The value of ω_f should be carefully chosen in order not to attenuate the low-frequency components of the excitation input signal. For example, if a PRBS is used as the input signal with a period $M = (2^n - 1)T_s$, then ω_f is chosen smaller than the lowest frequency component of the PRBS signal, which is $2\pi/M$.

Removing high-frequency disturbances: It was mentioned that a continuous-time anti-aliasing filter should be used to remove all frequency components of the measured signals above the Nyquist frequency (half of the sampling frequency). However, if the sampling period is too small, the spectrum of the measured signals may be much longer than what is needed for model identification. For example, by observing the Bode diagram of the model obtained from a frequency-domain identification, we may decide that we are more interested in modeling the low-frequency behavior of our system. In this case, we need a discrete-time low-pass filter to remove the high-frequency components of the model and disturbances. A simple first-order discrete-time low-pass filter is given by:

$$L(q^{-1}) = \frac{1 - \alpha}{1 - \alpha q^{-1}}$$

where $\alpha = e^{-\omega_f T}$ is the cutoff frequency of the filter.

Data scaling

The input signal and the measured output may have different orders of magnitude. The regressor vector contains the past values of $u(k)$ and $y(k)$ and the information

matrix includes the covariance matrix of inputs and outputs. Large difference between the values in this matrix makes some numerical problems when its inverse is computed in the least squares algorithm or as the Hessian in the Gauss-Newton optimization method. As a result, the convergence speed for the parameters of the numerator and denominator becomes different. Therefore, if the input and output data do not have the same order of magnitude, they should be scaled in order to fix this type of numerical problems.

3.5.3 Structure selection

One of the objectives of system identification is to find the simplest model that meets the validation criteria. For the LTI models, the choice of the degrees of the polynomials $B(q^{-1}), A(q^{-1})$ and the time delay, respectively, m, n and d, is of great importance. A rough estimate of these values can be obtained by inspecting the nonparametric identified models of the system. It was discussed in Section 2.2 that the delay can be estimated from the step or impulse response of the system. The minimum degree of the denominator can be also estimated from the impulse or step response (a damped response corresponds to a first-degree denominator and an oscillatory response to a second-degree denominator). The number of zeros and poles of a system can be estimated by studying the variations of the slope of the magnitude Bode diagram (see Section 2.3). These methods are approximative and can give only an estimate of the interval to which the degrees may belong.

Another method to find an estimate of the order is to investigate the physical model obtained by the first principles approach and select the order of the to-be-identified model accordingly. In this section, some parametric identification methods to estimate the model order are proposed.

Note that the order of the model estimated by the methods presented in this section does not necessarily guarantee that an identified model with that order will be validated. The procedure of system identification is an iterative procedure and if at the end the model is not validated, one solution is to reinvestigate the order selection and eventually increase the model order.

Loss function method

The loss function, $J(\hat{\theta})/N$, is the mean value of the optimization criterion evaluated at the estimated value. For an ARX structure, the loss function is a monotonically non-increasing function with respect to the number of the parameter estimates $n_\theta = m + n - d + 1$ (dimension of θ). The reason is that the parameter estimates are the global optimum of the criterion and increasing the number of parameters gives more degree of freedom to the optimization problem and can only improve the value of the criterion. In the worst case, adding more parameters will not change the value of the criterion but never increase it. Based on this fact, the model order $\delta = \max(m, n)$, can be estimated by inspecting the evolution of the loss function of an ARX structure.

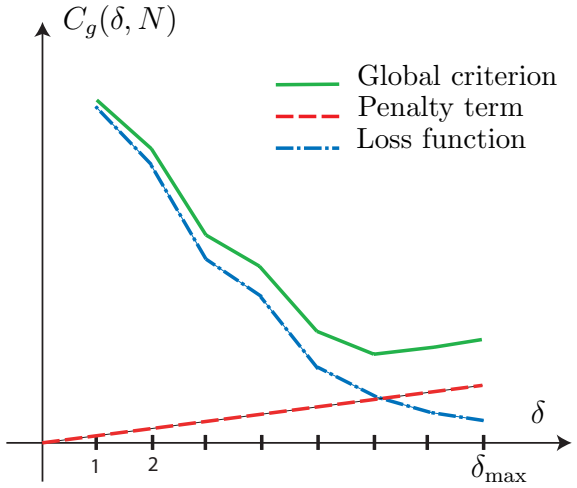


Figure 3.10: The evolution of the loss function $(- \cdot -)$, penalty term $(--)$ and global criterion $(-)$ as a function of δ

Let's fix the delay to $d = 1$ and take $\delta = n = m$ and plot the evolution of the loss function for $\delta = 1, \ldots, \delta_{\max}$, where $\delta_{\max}$ is the maximum order of the model. In other words, for each value of δ an ARX model is identified and the loss function is computed based on the value of the residuals $\varepsilon(k, \hat{\theta})$ as follows:

$$L_f(\delta, N) = \frac{1}{N} \sum_{k=1}^{N} \varepsilon^2(k, \hat{\theta}) \tag{3.140}$$

In a noiseless simulation with a finite order model, the value of the loss function will become equal to zero for δ greater than or equal to the model order. However, in the presence of noise, the loss function will converge to a non-zero value (the noise variance), and the exact order of the model cannot be easily estimated. By inspecting the evolution of the loss function, we can find a rough estimate of the model order at which the decrease of the loss function is not significant. This means that it may not be worth increasing the order if the decrease in the loss function is negligible. In order to have a quantitative criterion for model order selection, a penalty term can be added to the loss function, as shown in Fig. 3.10:

$$C_g(\delta, N) = L_f(\delta, N) + S(\delta, N) \tag{3.141}$$

where $C_g(\delta, N)$ is a global criterion and $S(\delta, N)$ is the penalty term which grows linearly with δ and decays when N increases. The minimum of the global criterion gives the order of the model. There are several penalty terms that are proposed in literature that lead to an asymptotically unbiased estimate of the model order.

Some of the well-known global criteria based on these penalty terms are given below:

$$\text{Akaike Information Criterion (AIC)} = L_f(\delta, N) + \frac{2\delta}{N} L_f(\delta, N) \tag{3.142}$$

$$\text{Bayesian Information Criterion (BIC)} = L_f(\delta, N) + \frac{\delta \log N}{N} \tag{3.143}$$

$$\text{Correlation Information Criterion (CIC)} = L_f(\delta, N) + \frac{\delta \log^2 N}{N} \tag{3.144}$$

$$\text{Final Prediction Error (FPE)} = L_f(\delta, N) \frac{1 + \delta/N}{1 - \delta/N} \tag{3.145}$$

Although these criteria have solid theoretical foundations for $N \to \infty$, in practice, for a finite number of data, they overestimate very often the model order. A visual inspection of the evolution of the loss function together with other information about the order can generally lead to more reasonable conclusions.

Remark: Model order selection is always performed with an ARX structure because the least square algorithm gives the global minimum of the identification criterion and the monotonically non-increasing property of the loss function is preserved. The other structures (e.g. OE, ARMAX, BJ) that are based on nonlinear optimization methods are not appropriate for order selection. If the noise level is too large the instrumental variables method can be used instead.

This method can be generalized to find n, m and d. Note that the number of parameters $n_\theta = n+m-d+1 = n_A+n_B$ is a good indicator of the model complexity, where $n_A = n$ is the number of parameters in the denominator and $n_B = m - d + 1$ is the number of parameters in the numerator. First, some intervals for n_A, n_B and d are defined as:

$$n_A \in [n_{A_{\min}}\,,\, n_{A_{\max}}] \quad , \quad n_B \in [n_{B_{\min}}\,,\, n_{B_{\max}}] \quad , \quad d \in [d_{\min}\,,\, d_{\max}]$$

Then, a set of models concerning all combinations of orders in the intervals are identified using the ARX structure. The number of parameters will increase from $n_{A\min} + n_{B_{\min}}$ to $n_{A_{\max}} + n_{B_{\max}}$. Next, the lowest loss function for each number of parameters is plotted versus the number of parameters in the model. From the evolution of the loss function, the number of parameters of numerator, denominator and d will be selected.

The selected orders of n_A, n_B and the delay d (denoted by n_k in Matlab) can be used to identify a parametric model, which has a noise model as well (e.g. ARMAX, BJ). In this case, in the absence of a priori information the order of the noise model is chosen less than or equal to the order of the plant model.

Zero/pole cancellation method

If the model order is overestimated, it can be reduced without affecting the input/output behavior of the plant model. In this case, some poles and zeros of the

plant model will be very close to each other and there is a possibility of zero/pole cancellation. This can be easily seen in an ARMAX structure as follows:

$$A_0(q^{-1})y(k) = B_0(q^{-1})u(k) + C_0(q^{-1})e(k) \tag{3.146}$$

If $A(q^{-1}) = A_0(q^{-1}), B(q^{-1}) = B_0(q^{-1})$ and $C(q^{-1}) = C_0(q^{-1})$ is a solution to the above polynomial equation, then $A(q^{-1})M(q^{-1}), B(q^{-1})M(q^{-1})$ and $C(q^{-1})M(q^{-1})$ is also a solution and can represent the behavior of the system.

This fact can be used to estimate the order of a model as follows. Let's take $d = 1, \delta = n = m$ and change δ from 1 to $\delta_{\max}$ and identify a model with ARMAX structure for each order (the degree of $C(q^{-1})$ is chosen equal to δ). The zero/pole map of each model can be inspected for possible existence of zero/pole cancellation. One zero/pole cancellation indicates that the order of the model is overestimated by 1. In the presence of noise, an exact zero/pole cancellation is impossible even if the order is overestimated. It is more reasonable to compute the variance of the poles and zeros and plot the 95% confidence intervals around the poles and zeros (the variance of the poles and zeros can be computed from the variance of the estimated model parameters). An intersection between the confidence interval of a pole and that of a zero indicates an overestimation of the model order. Therefore, the model order is the maximum value of δ for which there is no zero/pole cancellation.

Estimation of the delay

The delay can be estimated from the impulse response of the system. The time of the first reaction of the output to the impulse signal corresponds to the time delay: $g(k) = 0$ for all $k < d$. The impulse response can be identified by the FIR structure with a large value of m (greater than the settling time of the impulse response). An FIR model with $d = 0$ is given by:

$$G(q^{-1}) = b_0 + b_1 q^{-1} + b_2 q^{-2} + \cdots + b_m q^{-m} = B(q^{-1})$$

Therefore, $g(k) = b_k$ in an FIR model. As a result, by inspecting the parameters of a high-order FIR model, an estimate of the time delay can be obtained as *the number of the first coefficients of $B(q^{-1})$ which are equal to zero.*

In practice, in the presence of noise, none of the coefficients of $B(q^{-1})$ will be equal to zero. Therefore, a coefficient can be considered as zero if

$$0 \in [b_k - 2\sigma_k \, , \, b_k + 2\sigma_k]$$

where σ_k is the standard deviation of the k-th coefficient of $B(q^{-1})$ and the interval $\pm 2\sigma_k$ corresponds to the 95% confidence interval associated to a random variable with Gaussian distribution.

Although, FIR model is preferred for estimation of the delay, since it can be identified with the least squares algorithm, the other structures can be also used for the estimation of the delay by inspecting the $B(q^{-1})$ polynomial when d is fixed to zero.

3.5.4 Identification procedure

A successful identification depends on several a priori choices as the sampling period, the excitation signal, the model structure and order of different polynomials, noise model and the optimization algorithm. These choices are based on some a priori information about the system and can be modified during the identification procedure in an *iterative* way.

We present an iterative procedure to identify the parameters of an LTI model. The general rule is to start with a simple structure and try to validate the identified model. If the model cannot be validated, the more complex structures with more parameters can be considered. The identification procedure has Five important points as follows:

1. Based on a step response or impulse response (e.g. using the correlation approach), find a rough estimate of the time-delay, the dominant time-constant and the variance of the measurement noise. These information can help to choose the sampling period, the amplitude and the frequency content and the length of the excitation signal.

2. Using a Fourier or spectral analysis, identify the frequency response of the system. Inspecting the frequency response can give some information to modify the sampling period and choose an input signal with good excitation around the interesting zones (e.g. resonance modes, crossover frequency).

3. Estimate the order of the model using the parametric methods (e.g. loss function, zero/pole cancellation).

4. Use different model structures (ARX, ARMAX, BJ, OE) to identify different models. A state-space structure can also be considered.

5. Use different validation techniques (time-domain, frequency-domain, statistical) using a new set of validation data to compare the available models. Choose the best model based on the validation criteria. If none of the model is validated, the choice of structure and order should be reinvestigated.

The flowchart of this procedure is presented in Fig. 3.11.

3.6 Optimization Algorithms

The model parameters are usually identified by minimizing a quadratic criterion based on the prediction error defined as:

$$J(\theta) = \sum_{k=1}^{N} \varepsilon^2(k,\theta) = \sum_{k=1}^{N} [y(k) - \hat{y}(k,\theta)]^2 \tag{3.147}$$

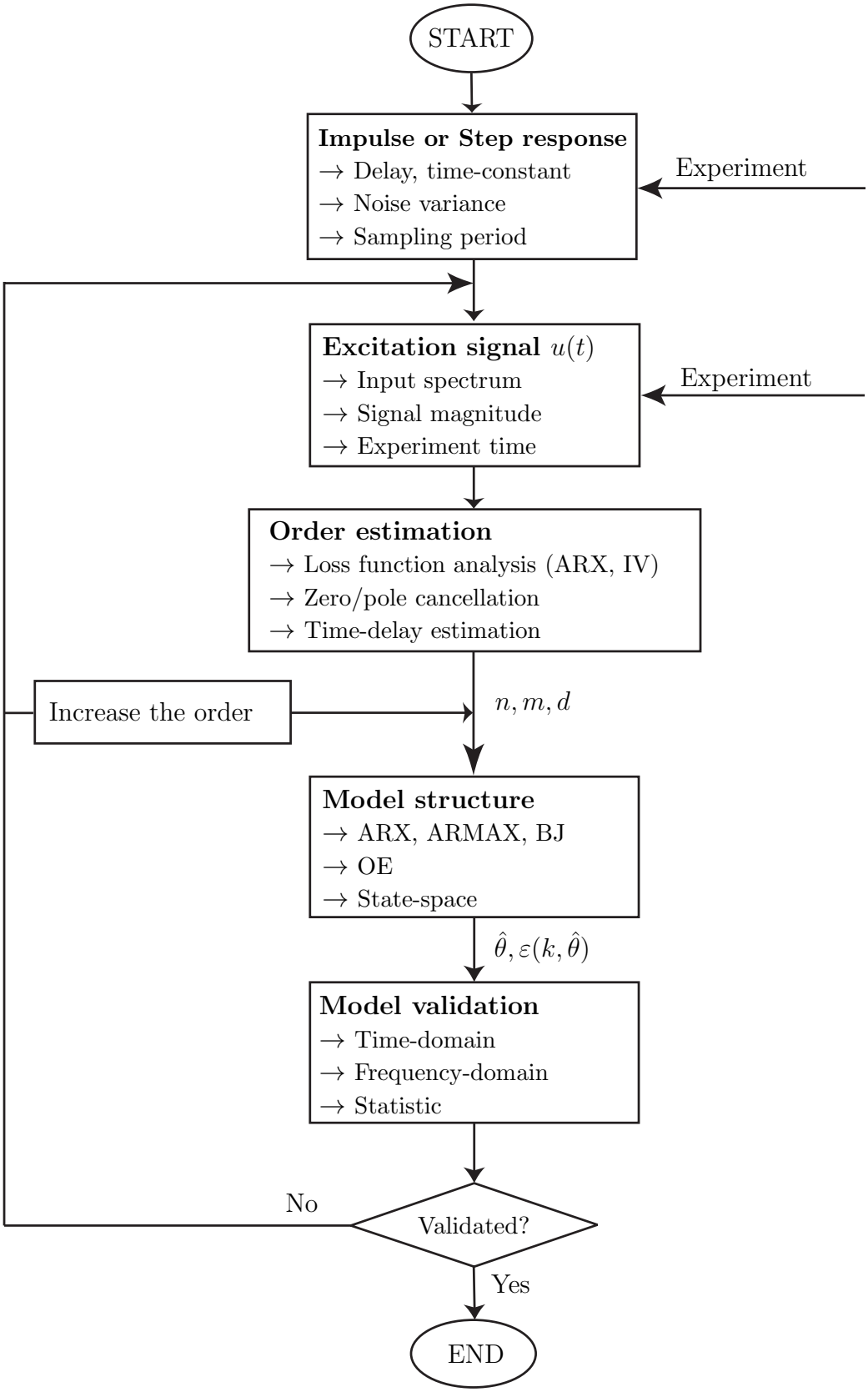


Figure 3.11: A procedure for parametric identification of LTI systems

If the prediction error is linear in θ, then minimization of $J(\theta)$ is a least squares problem and can be solved by the least squares algorithm. The model structures ARX and FIR lead to such criteria. In general, however, the prediction error may be nonlinear in parameters and a numerical iterative method should be used to find a local minimum of the criterion. For this type of algorithms the initialization of the parameter vector becomes very crucial.

In this section, first a simple method based on the pseudo linear regression is introduced. Then, the well-known Gauss-Newton algorithm for the minimization of the prediction error is discussed. Finally, a recursive algorithm for on-line identification of model parameters based on the least squares algorithm is developed.

3.6.1 Pseudo-linear regression algorithm

It was discussed that only FIR and ARX structures lead to a linear regression problem. Here, we show that the prediction error for the other model structures can be represented by a pseudo linear regression and solved with a simple optimization algorithm.

Output error structure: The output error predictor in (3.86) can be rewritten as:

$$\begin{aligned}\hat{y}(k,\theta) = &-a_1\hat{y}(k-1,\theta) - \cdots - a_n\hat{y}(k-n,\theta) \\ &+ b_d u(k-d) + \cdots + b_m u(k-m) = \phi^T(k,\theta)\theta\end{aligned} \tag{3.148}$$

where

$$\phi^T(k,\theta) = [\hat{y}(k-1,\theta)\,,\,\ldots\,,\,\hat{y}(k-n,\theta)\,,\,u(k-d)\,,\,\ldots\,,\,u(k-m)] \tag{3.149}$$

is the regressor vector. Note that this type of regression is not linear because of dependence of the regressor to θ. However, this type of regression is called *pseudo linear regression* and can be solved by the following iterative algorithm:

$$\hat{\theta}_{i+1} = \left[\sum_{k=1}^{N}\phi(k,\hat{\theta}_i)\phi^T(k,\hat{\theta}_i)\right]^{-1}\left[\sum_{k=1}^{N}\phi(k,\hat{\theta}_i)y(k)\right] \tag{3.150}$$

This approach known as the substitution algorithm is the simplest algorithm for solving some class of nonlinear optimization problems. The algorithm should be initialized with the least squares estimates and converges usually after a few iteration. However, there is no proof of converging to a local minimum of the criterion.

ARMAX structure: The prediction error for the ARMAX structure is nonlinear in parameters and a nonlinear optimization method is required to find the parameters. However, the prediction error can be rewritten as a pseudo linear regression. Let us define the monic polynomial $C(q^{-1})$ as

$$C(q^{-1}) = 1 + c_1 q^{-1} + \cdots + c_{n_c} q^{-n_c} \tag{3.151}$$

and multiply (3.104) by this polynomial to obtain:

$$\begin{aligned}\varepsilon(k,\theta) &= y(k) - [-a_1 y(k-1) - \cdots - a_n y(k-n) \\ &\quad + b_d u(k-d) + \cdots + b_m u(k-m) \\ &\quad + c_1 \varepsilon(k-1,\theta) + \cdots + c_{n_c}\varepsilon(k-n_c,\theta)] \\ &= y(k) - \phi_x^T(k,\theta)\theta \end{aligned} \tag{3.152}$$

where

$$\begin{aligned}\phi_x^T(k,\theta) = [&-y(k-1),\ldots,-y(k-n),u(k-d),\ldots,u(k-m), \\ &\varepsilon(k-1,\theta),\ldots,\varepsilon(k-n_c,\theta)]\end{aligned} \tag{3.153}$$

$$\theta^T = [a_1,\ldots,a_n,b_d,\ldots,b_m,c_1,\ldots,c_{n_c}] \tag{3.154}$$

Therefore an estimate of θ can be obtained by the pseudo linear regression algorithm as:

$$\hat{\theta}_{i+1} = \left[\sum_{k=1}^{N} \phi_x(k,\hat{\theta}_i)\phi_x^T(k,\hat{\theta}_i)\right]^{-1}\left[\sum_{k=1}^{N}\phi_x(k,\hat{\theta}_i)y(k)\right] \tag{3.155}$$

Reformulating the prediction error of the Box-Jenkins structure to a pseudo linear regression remains as an exercise for the interested students.

3.6.2 Gauss-Newton algorithm

This algorithm is iterative and converges to a local minimum of the criterion. At iteration $i+1$, the parameter vector is computed by the following relation:

$$\hat{\theta}_{i+1} = \hat{\theta}_i - [J''(\hat{\theta}_i)]^{-1}J'(\hat{\theta}_i) \tag{3.156}$$

where $J'(\hat{\theta}_i)$ is the gradient and $J''(\hat{\theta}_i)$ the Hessian of the criterion (3.147) evaluated at $\hat{\theta}_i$. The algorithm is initialized with the classical least squares algorithm for the plant model. The parameters of the noise model are often initialized to zero. The gradient and the Hessian can be computed straightforwardly for each structure.

The gradient of $J(\theta)$ with respect to θ is written as:

$$J'(\theta) = \frac{\partial J}{\partial \theta} = -2\sum_{k=1}^{N}\frac{\partial \hat{y}}{\partial\theta}\varepsilon(k,\theta) = -2\sum_{k=1}^{N}\psi(k,\theta)\varepsilon(k,\theta) \tag{3.157}$$

where $\psi(k,\theta)$ is the gradient of the output predictor:

$$\psi(k,\theta) \equiv \frac{\partial\hat{y}}{\partial\theta} \tag{3.158}$$

The Hessian is computed as follows:

$$\begin{aligned}J''(\theta) = \frac{\partial^2 J}{\partial\theta\partial\theta^T} &= 2\sum_{k=1}^{N}\left[\psi(k,\theta)\psi^T(k,\theta) - \frac{\partial\psi}{\partial\theta}\varepsilon(k,\theta)\right] \\ &\approx 2\sum_{k=1}^{N}\psi(k,\theta)\psi^T(k,\theta)\end{aligned} \tag{3.159}$$

The second term in the Hessian is ignored because it is too small with respect to the first term in the neighbourhood of the solution. In fact, $\varepsilon(k,\theta)$ will be close to a white noise for the structures with noise model and will not be correlated with the derivative of $\psi(k,\theta)$, so its expectation will be zero. To implement the Gauss-Newton algorithm, the gradient of the output predictor $\psi(k,\theta)$ is the main element for computing the gradient and the Hessian of the criterion.

Example 3.4. Compute the gradient of the output predictor, $\psi(k,\theta)$, for the output error structure. We have the following output error predictor (3.80):

$$\hat{y}(k,\theta) = \frac{B(q^{-1})}{A(q^{-1})}u(k) = \frac{b_d q^{-d} + \cdots + b_m q^{-m}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n}}u(k) \tag{3.160}$$

The derivative of $\hat{y}(k,\theta)$ with respect to the parameters of the numerator and the denominator are computed separately as:

$$\frac{\partial \hat{y}}{\partial b_i} = \frac{q^{-i}}{A(q^{-1})}u(k) = \frac{1}{A(q^{-1})}u(k-i) \qquad i = d,\ldots,m \tag{3.161}$$

$$\frac{\partial \hat{y}}{\partial a_i} = \frac{-q^{-i}B(q^{-1})}{A^2(q^{-1})}u(k) = \frac{-1}{A(q^{-1})}\hat{y}(k-i) \qquad i = 1,\ldots,n \tag{3.162}$$

Thus, we obtain:

$$\begin{aligned}\psi^T(k,\theta) &= \frac{1}{A(q^{-1})}[-\hat{y}(k-1,\theta),\ldots,-\hat{y}(k-n,\theta),u(k-d),\ldots,u(k-m)] \\ &= \frac{1}{A(q^{-1})}\phi^T(k,\theta)\end{aligned} \tag{3.163}$$

Remark: In a similar way, it can be shown that for the ARMAX structure, the derivative of the output predictor is given by:

$$\frac{\partial \hat{y}}{\partial \theta} = \frac{1}{C(q^{-1})}\phi_x(k,\theta)$$

where $\phi_x(k,\theta)$ is defined in (3.153).

Algorithm: The step-by-step Gauss-Newton algorithm for identification of the parameters of the output error structure are summarized as follows:

1. Choose the order of the plant model (d,n,m) and a stop criterion $\epsilon > 0$.
2. Initialize the vector of the parameters by the least squares algorithm $(i=1)$: $\hat{\theta}_1 = (\Phi^T\Phi)^{-1}\Phi Y$.
3. Compute $\hat{y}(k,\hat{\theta}_i) = G(q^{-1},\hat{\theta}_i)u(k)$ for $k = 1,\ldots,N$
4. Compute the prediction error $\varepsilon(k,\hat{\theta}_i) = y(k) - \hat{y}(k,\hat{\theta}_i)$ for $k = 1,\ldots,N$.

5. Compute the derivative of the output predictor $\psi^T(k,\hat{\theta}_i) = \dfrac{1}{A(q^{-1},\hat{\theta}_i)}\phi^T(k,\hat{\theta}_i)$ for $k = 1,\ldots,N$ with

$$\phi^T(k,\hat{\theta}_i) = [-\hat{y}(k-1,\hat{\theta}_i),\ldots,-\hat{y}(k-n,\hat{\theta}_i),u(k-d),\ldots.u(k-m)]$$

6. Compute the gradient of the criterion $J'(\hat{\theta}_i) = -2\sum_{k=1}^{N}\psi(k,\hat{\theta}_i)\varepsilon(k,\hat{\theta}_i)$

7. Compute the Hessian of the criterion $J''(\hat{\theta}_i) = 2\sum_{k=1}^{N}\psi(k,\hat{\theta}_i)\psi^T(k,\hat{\theta}_i)$

8. Compute $\hat{\theta}_{i+1} = \hat{\theta}_i - [J''(\hat{\theta}_i)]^{-1}J'(\hat{\theta}_i)$

9. If $(\hat{\theta}_{i+1}-\hat{\theta}_i)^T(\hat{\theta}_{i+1}-\hat{\theta}_i) < \epsilon$, stop the algorithm with $\hat{\theta} = \hat{\theta}_{i+1}$ unless $i = i+1$ and go to 3.

A similar algorithm is used in MATLAB and converges usually after a few iterations.

Asymptotic covariance of the parameter estimates

Note that the measured output of the system is a random process because of the effect of the measurement noise. With the same reason, the prediction error $\varepsilon(k,\theta)$ and the vector of the estimated parameters $\hat{\theta}$ are also random processes. It means that the parameter estimates for two different data sets will be different because of different noise realization. Therefore, the quality of the estimates can be assessed only statistically by analyzing its bias and variance.

Assume that the number of data N is large and the parameter estimates converge to θ^*, where $\theta^* = \mathbb{E}\{\hat{\theta}\}$. Then, knowing that $J'(\hat{\theta}) = 0$, its first-order Taylor expansion around θ^* can be computed as:

$$J'(\hat{\theta}) \approx J'(\theta^*) + J''(\theta^*)(\hat{\theta}-\theta^*) = 0 \tag{3.164}$$

This leads to:

$$\hat{\theta}-\theta^* = -[J''(\theta^*)]^{-1}J'(\theta^*) \tag{3.165}$$

Therefore, the covariance of the parameters is given by:

$$\begin{aligned}\mathrm{cov}(\hat{\theta}) &= \mathbb{E}\{(\hat{\theta}-\theta^*)(\hat{\theta}-\theta^*)^T\}\\ &= \mathbb{E}\left\{[J''(\theta^*)]^{-1}J'(\theta^*)J'^T(\theta^*)[J''(\theta^*)]^{-1}\right\}\end{aligned} \tag{3.166}$$

Then using (3.157), we obtain:

$$\mathbb{E}\{J'(\theta^*)J'^T(\theta^*)\} = \mathbb{E}\left\{\left[-2\sum_{k=1}^{N}\psi(k,\theta^*)\varepsilon(k,\theta^*)\right]\left[-2\sum_{k=\ell}^{N}\psi(\ell,\theta)\varepsilon(\ell,\theta^*)\right]^T\right\}$$

For the prediction error methods $\varepsilon(k,\theta_0)$ is a sequence of zero-mean white noise with variance σ_e^2. If we assume that $\varepsilon(k,\theta^*)$ is also a zero-mean white noise, the expected values of all terms including $\varepsilon(k,\theta^*)\varepsilon(\ell,\theta^*)$ with $k \neq \ell$ will vanish and we get:

$$\mathbb{E}\{J'(\theta^*)J'^T(\theta^*)\} = 4\sigma_e^2 \sum_{k=1}^{N} \psi(k,\theta^*)\psi^T(k,\theta^*) = 2\sigma_e^2 J''(\theta^*) \tag{3.167}$$

which, using (3.166) and (3.159), leads to:

$$\mathrm{cov}(\hat{\theta}) \approx \sigma_e^2 \left[\sum_{k=1}^{N} \psi(k,\hat{\theta})\psi^T(k,\hat{\theta})\right]^{-1} = \frac{\sigma_e^2}{N}\left[\frac{1}{N}\sum_{k=1}^{N} \psi(k,\hat{\theta})\psi^T(k,\hat{\theta})\right]^{-1} \tag{3.168}$$

The noise variance can be estimated by:

$$\widehat{\sigma_e^2} = \frac{1}{N-p}\sum_{k=1}^{N} \varepsilon^2(k,\hat{\theta}) \tag{3.169}$$

where p is the number of the parameters (the dimension of θ). Note that, in practice as $N >> p$ the term p in the denominator is often ignored. The above expression is very general and can be used for any model structure (continuous-time, nonlinear, etc). The main condition is the whiteness of $\varepsilon(k,\theta_0)$.

It turns out that $(\hat{\theta}-\theta^*)$ typically decays as $1/\sqrt{N}$. Therefore, under some weak assumptions based on the central limit theorem, $\sqrt{N}(\hat{\theta}-\theta^*)$ converges in distribution to a Gaussian distribution with $\mathcal{N}(0, N\mathrm{cov}(\hat{\theta}))$. This can be used to compute the confidence intervals with a probability level around $\hat{\theta}$. The confidence intervals can be computed for any other function of $\hat{\theta}$ like the zeros and poles, magnitude and phase in Bode diagram or real and imaginary parts in Nyquist diagram.

3.6.3 Recursive algorithms

The least squares solution given by

$$\hat{\theta} = \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1}\left[\sum_{k=1}^{N} \phi(k)y(k)\right] \tag{3.170}$$

can be transformed to a recursive equation for real-time implementation. The main objective is to get rid of the inversion of a $p \times p$ matrix during one sampling period and a long summation over N data when N grows.

Consider the measurements up to k and the vector of the parameter estimates at instant k as:

$$\hat{\theta}_k = \left[\sum_{i=1}^{k} \phi(i)\phi^T(i)\right]^{-1}\sum_{i=1}^{k} \phi(i)y(i) \tag{3.171}$$

When the measurements arrive successively in time, it is interesting to compute recursively $\hat{\theta}_k, \hat{\theta}_{k+1}, \ldots$. It means that computing $\hat{\theta}_{k+1}$ from $\hat{\theta}_k$ without taking the whole summation. For this purpose, we can rewrite (3.171) as follows:

$$\hat{\theta}_k = P_k \sum_{i=1}^{k} \phi(i) y(i) \tag{3.172}$$

where the p dimensional square matrix P_k is the inverse of the information matrix.

$$P_k = \left[\sum_{i=1}^{k} \phi(i) \phi^T(i) \right]^{-1} \tag{3.173}$$

On the other hand, P_k is proportional to the covariance of the parameter estimates (see (3.168) and consider that in a linear regression algorithm $\psi(k) = \phi(k)$). Note that the matrix P_{k+1}^{-1} can be computed recursively as:

$$\begin{aligned} P_{k+1}^{-1} &= \sum_{i=1}^{k+1} \phi(i) \phi^T(i) = \sum_{i=1}^{k} \phi(i) \phi^T(i) + \phi(k+1) \phi^T(k+1) \\ &= P_k^{-1} + \phi(k+1) \phi^T(k+1) \end{aligned} \tag{3.174}$$

The vector of the parameter estimates at instant $k+1$ can be written as:

$$\hat{\theta}_{k+1} = P_{k+1} \sum_{i=1}^{k+1} \phi(i) y(i) = P_{k+1} \left[\sum_{i=1}^{k} \phi(i) y(i) + \phi(k+1) y(k+1) \right] \tag{3.175}$$

$$= P_{k+1} [P_k^{-1} \hat{\theta}_k + \phi(k+1) y(k+1)] \tag{3.176}$$

$$= P_{k+1} [P_{k+1}^{-1} - \phi(k+1) \phi^T(k+1)] \hat{\theta}_k + P_{k+1} \phi(k+1) y(k+1) \tag{3.177}$$

$$= \hat{\theta}_k + P_{k+1} \phi(k+1) [y(k+1) - \phi^T(k+1) \hat{\theta}_k] \tag{3.178}$$

The last term in the bracket is in fact the prediction error at instant $k+1$ computed based on the latest estimated parameters $\hat{\theta}_k$. This error is weighted by $P_{k+1}\phi(k+1)$ and updates the value of the parameter estimates. This recursive formula makes sense because the update of the parameters is proportional to the prediction error. On the other hand, if the covariance of the parameter estimates is small, i.e. P_{k+1} is small, a large prediction error because of an external disturbance in the output at $y(k+1)$ will not affect considerably the update of the parameters.

The least squares algorithm, thus, can be represented by the following recursive formulas:

$$P_{k+1}^{-1} = P_k^{-1} + \phi(k+1) \phi^T(k+1) \tag{3.179}$$

$$\varepsilon(k+1, \hat{\theta}_k) = y(k+1) - \phi^T(k+1) \hat{\theta}_k \tag{3.180}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + P_{k+1} \phi(k+1) \varepsilon(k+1, \hat{\theta}_k) \tag{3.181}$$

At instant k, we measure $y(k)$ and build the vector $\phi(k+1)$ and compute P_{k+1}^{-1} for the next sampling period based on (3.179). At instant $k+1$, we measure $y(k+1)$ and compute the prediction error $\varepsilon(k+1, \hat{\theta}_i)$ from (3.180) and $\hat{\theta}_{k+1}$ from (3.181). Finally, we compute $\phi(k+2)$ and P_{k+2}^{-1} for the next sampling period.

Matrix inversion lemma: To avoid the inversion of P_{k+1} at each iteration, the matrix inversion lemma can be used. According to this lemma, if A, C and $C^{-1} + DA^{-1}B$ are the invertible matrices, then:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1} \tag{3.182}$$

For proof, it is easy to show that

$$(A + BCD)(A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1}) = I$$

Let's take $A = P_k^{-1}$, $B = \phi(k+1)$, $C = 1$, $D = \phi^T(k+1)$ in (3.179), we obtain:

$$P_{k+1} = P_k - \frac{P_k\phi(k+1)\phi^T(k+1)P_k}{1 + \phi^T(k+1)P_k\phi(k+1)} \tag{3.183}$$

The advantage of the above equation is that it requires no matrix inversion but only a division by a scalar.

Initialization: There are two ways for initializing the recursive algorithm that are discussed here.

1. The initial values are fixed a priori. In general, $\hat{\theta}_0 = 0$ and $P_0 = \alpha I$, where α is a large value multiplied by the number of parameters p, say $\alpha = 1000p$, and I is the unity matrix. The reason for this choice comes from the fact that P_0 is an initial estimate of the covariance matrix of the parameters. Since the initial value is far from the true one, a large covariance matrix is chosen.

2. The recursion starts after p sampling period. At iteration p, $\hat{\theta}_p$ is estimated by solving a system of linear equation as:

   $$\hat{\theta}_p = \Phi_p^{-1}Y_p \quad ; \quad P_p = [\Phi_p^T\Phi_p]^{-1}$$

   where

   $$\Phi_p = \begin{bmatrix} \phi^T(p) \\ \vdots \\ \phi^T(1) \end{bmatrix}$$

   and $Y_p = [y(p), \ldots, y(1)]$.

Comparison with non recursive algorithm: The recursive algorithms comparing the classical batch algorithms have the following advantages:

- The model parameters can be estimated in real time.
- It compresses significantly the measured data since the algorithm uses at each instant one pair of input/output data instead of the whole acquired one.
- It requires much less memory and computation power.
- It can be easily implemented in the microcontrollers.

Forgetting factor

Note that the matrix $\phi(k+1)\phi^T(k+1)$ in (3.179) is a positive semidefinite matrix, i.e. all its eigenvalues are non-negative and so its trace is non-negative. As a result, the trace of P_{k+1}^{-1} will be monotonically non decreasing in time. In the presence of some excitation such that $\phi\phi^T$ is positive definite in average, the trace of P_{k+1} will converge to zero. Therefore, after a while, the vector of the parameter estimates will converge to a constant vector. So if the system parameters vary in time, the algorithm will not follow this variation as the adaptation gain $P_{k+1}\phi(k+1)$ will be almost zero.

In order to use this recursive algorithm to identify the parameters of time-varying systems a *forgetting factor* should be included in the algorithm. The main idea is to weight the older data with a smaller weight with respect to the fresh data. For this purpose, a recursive weighted least squares algorithm should be developed with the following weighting matrix:

$$W^TW = \begin{bmatrix} \lambda^{N-1} & & & 0 \\ 0 & \ddots & & \\ & & \lambda^1 & \\ 0 & & & \lambda^0 \end{bmatrix} \qquad \text{with } \lambda \leq 1 \,(\text{usually } \; 0.9 \leq \lambda \leq 0.99) \tag{3.184}$$

where λ is called the forgetting factor. It is clear that the most recent error will be weighted with $\lambda^0 = 1$ and the past errors will have smaller weights.

With this weighting matrix and the weighted least squares estimates in (1.66), we obtain:

$$\hat{\theta}_k = \left[\sum_{i=1}^{k} \phi(i)\lambda^{k-i}\phi^T(i)\right]^{-1} \sum_{i=1}^{k} \phi(i)\lambda^{k-i}y(i) = P_k \sum_{i=1}^{k} \phi(i)\lambda^{k-i}y(i) \tag{3.185}$$

The matrix P_{k+1} is computed recursively as follows:

$$P_{k+1}^{-1} = \lambda P_k^{-1} + \phi(k+1)\phi^T(k+1) \tag{3.186}$$

The use of a forgetting factor does not allow the fast decay of $P_{k+1}\phi(k+1)$ to zero and the algorithm stays *alive* and can detect a parameter variation and correct the estimates.

In a similar way, a recursive algorithm with a forgetting factor can be derived as follows:

$$P_{k+1} = \frac{1}{\lambda}\left[P_k - \frac{P_k\phi(k+1)\phi^T(k+1)P_k}{\lambda + \phi^T(k+1)P_k\phi(k+1)}\right] \tag{3.187}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + P_{k+1}\phi(k+1)[y(k+1) - \phi^T(k+1)\hat{\theta}_k] \tag{3.188}$$

This algorithm can be used for parametric identification of slowly time-varying systems. A smaller λ leads to better tracking of parameter variation, while a larger λ, closer to 1, reduces the effect of external disturbances on the parameter estimates.

Example 3.5. Consider a first-order discrete-time LTI model as:

$$y(k) + a_1^\circ y(k-1) = b_1^\circ u(k-1) + e(k)$$

where, $\theta_0 = [a_1^\circ\,,\, b_1^\circ]^T$ is the true unknown parameters of the system, $u(k)$ the input signal, $y(k)$ the measured output and $e(k)$ a zero-mean white noise. Note that the white noise is not an output noise but an equation noise (the above model is in fact a first order ARX model).

The objective is to identify the parameters of the model based on the input/output data in three different cases. In the first case, the parameters are fixed and a recursive algorithm is used to estimates the parameters in real-time. In the second case, the model parameters will change during the experiment but no forgetting factor is used in the algorithm. In the third case, a forgetting factor is used and its effect on the parameter estimates will be observed.

First case: The parameter of the model are fixed to $\theta_0 = [-0.5\,,\, 0.5]^T$. The input signal, a square wave, and the noisy output of the system are shown respectively in Fig. 3.12a and Fig. 3.12b. The recursive algorithm is initialized with $\hat{\theta}_0 = [0\,,\, 0]^T$ and $P_0 = I$. The recursive equations in (3.179)-(3.181) are used to estimate the model parameters (shown in Fig. 3.12c). The symmetric matrix P_k, which shows the covariance of the parameter estimates, has only three parameters which are plotted in Fig. 3.12d. It can be observed that the parameter estimates converge to the vicinity of the true parameters after a few iteration. The elements of the matrix P_k will also converge asymptotically to zero.

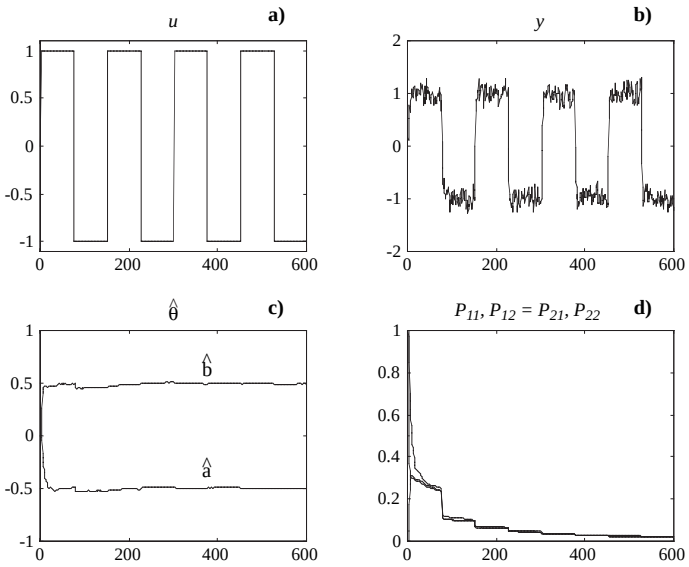


Figure 3.12: Identification of a time-invariant system: (a) input signal ; (b) noisy output; (c) parameter estimates; (d) elements of matrix P_k.

Second case: The parameters of the model are varying as follows:

$$\theta_0 = \begin{cases} [-0.5 \quad 0.5]^T & \text{for} \quad k < 200 \\ [0.5 \quad -0.5]^T & \text{for} \quad k \geq 200 \end{cases}$$

The algorithm for the first case is used again here and the results are shown in Fig. 3.13. It can be observed that the parameter estimates do not converge to the true parameters because the trace of the matrix P_k is too small for $k = 200$ and is not sufficient to adapt the parameters.

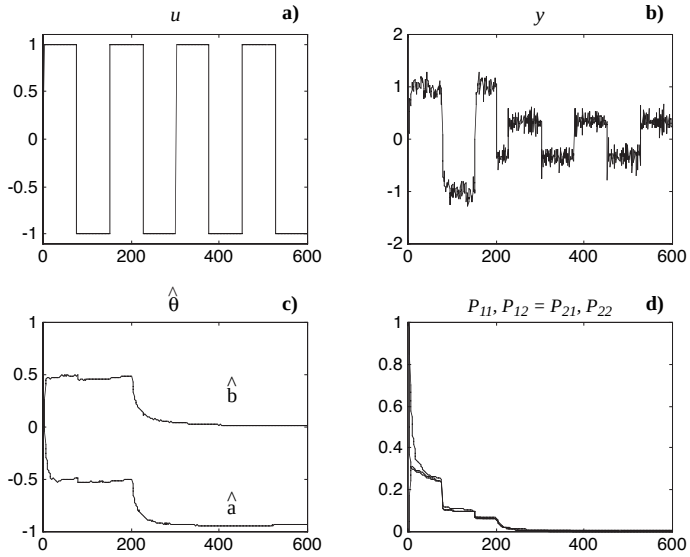


Figure 3.13: Identification of a time-varying system without forgetting factor (a) input signal ; (b) noisy output; (c) parameter estimates; (d) elements of matrix P_k.

Third case: For the time-varying model a recursive algorithm with forgetting factor $\lambda = 0.97$ is used. The forgetting factor prevents the fast convergence of the trace of P_k to zero. It can be seen that the trace of P_k increases when the input signal has low excitation and decays when the input signal has more excitation. As a result, at $k = 200$, when the system parameters change, the trace of P_k is not small and there is enough gain to adapt the parameters. The comparison between the elements of P_k in Fig. 3.13d and Fig. 3.14d, shows clearly the effect of the forgetting factor on the adaptation gain. The side effect of the high value of P_k is that the covariance of the parameter estimates will be much larger than the first case. This can be seen from the variation of the parameters around the true one about the end of the simulation (see Fig. 3.14c).

3.7 Identification of Nonlinear Systems

Identification of nonlinear systems, as for identification of linear systems, can be divided into two main categories: grey-box model identification and black-box model

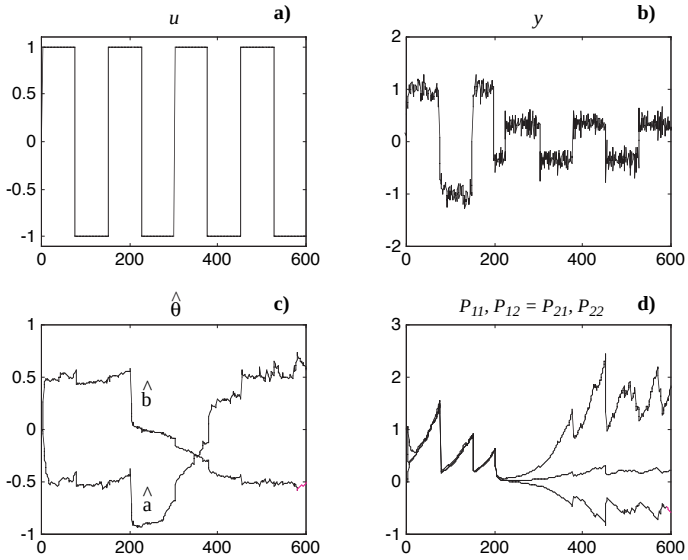


Figure 3.14: Identification of a time-varying system using a forgetting factor ($\lambda = 0.97$), (a) input signal ; (b) noisy output; (c) parameter estimates; (d) elements of matrix P_k.

identification. The grey-box modelling concerns the case that some physical insight is available and the objective is to estimate some parameters from the observed data. While in black-box modelling no physical insight is used but the model structure has good flexibility to cover approximately a large class of nonlinear behaviour.

When the model structure is chosen, grey-box or black-box, the estimation of the unknown parameters can be converted to an optimization problem and solved by standard algorithms. So the main difference between identification of linear systems and that of nonlinear systems is the choice of the model structure. As a result, the principle of prediction error methods and model validation for linear case can be used for the nonlinear case as well.

In this section we review briefly some grey-box nonlinear identification example and discuss the principles of black-box models.

3.7.1 Grey-Box Modelling

Let's consider a nonlinear model given by:

$$y(k) = \alpha_0 u(k-1)y(k-1) + \beta_0 y^2(k-2) + e(k) \tag{3.189}$$

where $\theta_0 = [\alpha_0 \quad \beta_0]^T$ is the vector of parameter to estimate. In this example we have a linear regression problem that can be solved using the least squares method. The only difference with the linear model identification is that the regressor vector is nonlinear with respect to the excitation signal and measurements. The predicted output is given by:

$$\hat{y}(k) = \phi^T(k)\theta \tag{3.190}$$

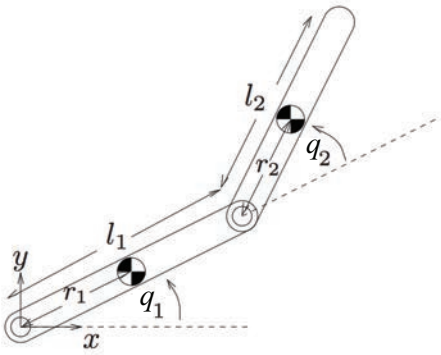


Figure 3.15: Two-link planar robot manipulator

where

$$\phi^T(k) = [u(k-1)y(k-1) \quad y^2(k-2)] \tag{3.191}$$

$$\theta^T = [\alpha \quad \beta] \tag{3.192}$$

Then, the parameters can be estimated using the least squares technique.

A good practical example of grey-box modelling is the model of the robotic arms. The mathematical model obtained by the Euler-Lagrange method can be written as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{3.193}$$

where $q \in \mathbb{R}^{n\times 1}$ denotes the vector of joint angles (n the number of joints), $M(q) \in \mathbb{R}^{n\times n}$ is the robot inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ contains the Coriolis terms, $G(q)$ are the gravity terms and $\tau \in \mathbb{R}^{n\times 1}$ is the torque vector applied to the joints. In order to identify the parameters of the model, the system should be excited by applying some torques to the joints (by choosing different trajectories) and the joint angles q, their speeds $\dot{q}$ and their accelerations $\ddot{q}$ are measured. Then, the parameters of the matrices $M(q)$ and $C(q,\dot{q})$ as well as the vector $G(q)$ can be identified from the measured data by the least squares algorithm. This can be shown by identification of a simple two-link planar robot manipulator shown in Fig. 3.15. The dynamic model of the system can be obtained as:

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{3.194}$$

where

$$M_{11} = I_1 + I_2 + m_1 r_1^2 + m_2(l_1^2 + r_2^2) + 2m_2 l_1 r_2 \cos(q_2) \tag{3.195}$$

$$M_{12} = M_{21} = I_2 + m_2 r_2^2 + m_2 l_1 r_2 \cos(q_2) \tag{3.196}$$

$$M_{22} = I_2 + m_2 r_2^2 \tag{3.197}$$

are the elements of the inertia matrix $M(q)$ with m_1 and m_2 the mass of each link, I_1 and I_2 the moment inertia in z direction of each link. The other parameters are

defined in Fig. 3.15. The elements of the Coriolis matrix are defined as:

$$C_{11} = -\dot{q}_2 m_2 l_1 r_2 \sin(q_2) \tag{3.198}$$
$$C_{12} = -(\dot{q}_1 + \dot{q}_2) m_2 l_1 r_2 \sin(q_2) \tag{3.199}$$
$$C_{21} = \dot{q}_1 m_2 l_1 r_2 \sin(q_2) \tag{3.200}$$
$$C_{22} = 0 \tag{3.201}$$

The equation of the system can be represented by linear regression as $\phi^T(k)\theta = \tau(k)$, where $\theta^T = [\alpha \quad \beta \quad \gamma]$ and $\phi(k) \in \mathbb{R}^{3\times 2}$. The parameters in θ are related to the physical parameters of the model as follows:

$$\alpha = I_1 + I_2 + m_1 r_1^2 + m_2(l_1^2 + r_2^2) \tag{3.202}$$
$$\beta = m_2 l_1 r_2 \tag{3.203}$$
$$\gamma = I_2 + m_2 r_2^2 \tag{3.204}$$

Therefore, the model in linear regression form will be:

$$\begin{bmatrix} \phi_{11}(k) & \phi_{12}(k) & \phi_{13}(k) \\ 0 & \phi_{22}(k) & \phi_{23}(k) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \tau_1(k) \\ \tau_2(k) \end{bmatrix} \tag{3.205}$$

where for each instant k:

$$\phi_{11} = \ddot{q}_1 \tag{3.206}$$
$$\phi_{12} = [2\ddot{q}_1 + \ddot{q}_2]\cos(q_2) - [\dot{q}_2\dot{q}_1 + \dot{q}_1^2 + \dot{q}_2^2]\sin(q_2) \tag{3.207}$$
$$\phi_{13} = \ddot{q}_1 \tag{3.208}$$
$$\phi_{22} = \ddot{q}_1 \cos(q_2) + \dot{q}_1^2 \sin(q_2) \tag{3.209}$$
$$\phi_{23} = \ddot{q}_1 + \ddot{q}_2 \tag{3.210}$$

Then, using the least squares method the parameters α, β and γ can be estimated. It is clear that the physical parameters are not all identifiable, but knowing some of them (e.g. l_1, l_2, r_1, r_2) some of the others may be estimated. However, knowing only θ, the matrix M and C are completely known and can be used for controller design.

3.7.2 Black-Box Modelling

In general for any nonlinear system, we can work with a structure for the parameterized output predictor:

$$\hat{y}(k, \theta) = F(\phi(k), \theta) \tag{3.211}$$

where F is some nonlinear function of θ and $\phi(k)$. The regressor vector is a (linear or nonlinear) function of past measured data. In some structures, $\phi(k)$ may be a function of past predicted output as well, so we should write it as $\phi(k, \theta)$.

Similar to the black-box models for linear systems, the following structures can be defined for the nonlinear models:

NFIR: The regressor vector will use only past inputs $u(k-l), l > 0$.

NARX: The regressor will use past inputs $u(k-l)$ and past outputs $y(k-l)$.

NOE: The regressor will use past inputs $u(k-l)$ and past predicted output $\hat{y}(k-l, \theta)$.

NARMAX: The regressor will use past inputs $u(k-l)$, past outputs $y(k-l)$ and past prediction errors $\varepsilon(k-l, \theta)$.

In the nonlinear state-space models, the regressor will contain also past measured states of the system.

It should be noted that when using the past predicted output in the predictor (in the structures like NOE), the optimization problem becomes harder and some stability condition for the predictor should be checked.

Basis functions

The nonlinear function $F(\phi(k), \theta)$ can be well approximated using some basis functions:

$$F(\phi(k), \theta) = \sum_{i=1}^{n} \theta_i F_i(\phi(k)) \tag{3.212}$$

where $\phi(k)$ can be any function of past inputs and outputs, F_i is the basis function and θ_i refers to the model parameters. Usually by increasing n, the number of model parameters, better approximation of the nonlinear function is obtained. Theoretically, when n goes to infinity any nonlinear function can be approximated by a well chosen basis function. The trade-off between bias and variance is also an issue in this type of modelling (increasing n will reduce the bias but increase the variance of the parameter estimates).

Now, the key point is to select an appropriate basis function. Most well-known black-box model structures use a single parameterized mother basis function that we denote it by $P(x)$. For the scalar case (where $\phi(k)$ is scalar), a typical example of a mother basis function is a piecewise-constant pulse function:

$$P(x) = \begin{cases} 1 & \text{for } 0 \leq x < \Delta \\ 0 & \text{otherwise} \end{cases} \tag{3.213}$$

then any nonlinear function $F(x)$ can be approximated by:

$$F(x) = \sum_{i=0}^{\infty} \theta_i P(x - i\Delta) \tag{3.214}$$

where $\theta_i = F(i\Delta)$ and $F_i(x) = P(x - i\Delta)$. A similar approximation can be obtained using a smooth version of the pulse function, e.g. by the Gaussian function:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.215}$$

Then $F(x)$ can be approximated by:

$$F(x) = \sum_{i=1}^{n} \theta_i P(x - \mu_i) \tag{3.216}$$

Usually a mother basis function has some parameter to be tuned as well. For example, in the Gaussian function the mean value and variance are the parameters of the basis function $\beta_i = [\mu_i \quad \sigma_i]$. An appropriate choice of these parameters can reduce significantly the number of parameters n.

For the multidimensional case (where $\phi(k)$ is a vector), the basis functions are constructed from the scalar case in some simple manner. For example a multivariate function can be approximated by the multiplication of single-variable basis functions. Suppose that d is the dimension of $\phi(k)$ then $P(\phi_1)P(\phi_2)\cdots P(\phi_d)$ can be considered as a basis function for F. An alternative, called Radial basis function, is constructed by a norm (or weighted norm) of ϕ, i.e. $P(\|\phi\|)$ is considered as a basis function.

When basis function is selected, computing the parameters θ becomes a least squares problem. However, as the results depend heavily on the choice of the parameters of the basis function β, these parameters are also optimized in an optimization algorithm. Therefore, the optimization problem becomes nonlinear and numerical methods are used to solve them.

In summary, in a nonlinear black-box modelling the parameters are obtained as the minimum of the following fit criterion:

$$\theta = \arg\min_{\theta,\beta} \sum_{k=1}^{N} \|y(k) - \sum_{i=1}^{n} \theta_i F_i(\phi(k), \beta)\| \tag{3.217}$$

Therefore, the first step is to choose $\phi(k)$ based on some a priori knowledge about the system. If this knowledge is not available, a high dimension $\phi(k)$ may be considered that complicates the optimization problem. The second step is to choose a basis function like Gaussian function, wavelet, sigmoid, etc. If the parameters of the basis function are fixed (defined by user), the optimization becomes a least squares problem, otherwise a nonlinear, gradient-based, numerical optimization should be solved. In order to consider the bias-variance trade-off a new term $\lambda\|\theta\|$ can be added to the fit criterion, such that bias and variance are minimized together. A good trade-off can be obtained by minimizing the one norm of θ to make it as sparse as possible and obtain finally a low-order model for the system.

3.7.3 Neural Network Models

A neural network model is a type of model inspired by the structure and function of the human brain. It consists of multiple interconnected nodes (also known as neurons) organized in layers that process information and generate output based on the input data.

Neural network models are particularly useful for modelling complex nonlinear systems because they can learn to extract high-level features from the input data and represent them in a way that allows the model to make accurate predictions. For example, neural networks have been used to model complex systems in fields such as computer vision, natural language processing, and financial forecasting.

However, it's important to note that neural network models can be computationally intensive and require a significant amount of data and computational resources to train and optimize. Additionally, they can be prone to overfitting, where the model becomes too specialized to the training data and performs poorly on new data. Therefore, it's important to carefully design and train neural network models to ensure that they are effective and robust.

A neural network model typically consists of multiple layers of interconnected neurons, with each layer processing and transforming the input data in a different way. The three main types of layers in a neural network are (see Fig. 3.16):

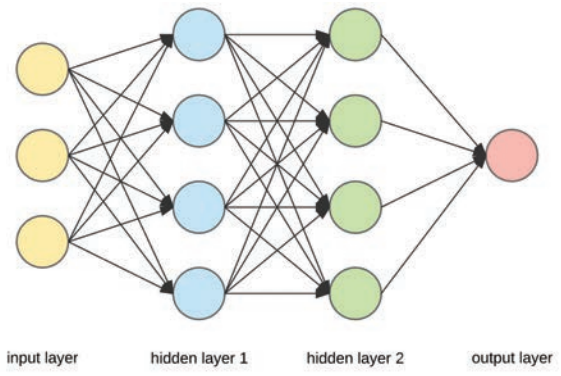


Figure 3.16: Structure of a Neural Network Model

Input layer: This is the first layer of the neural network and is responsible for receiving the input data. The size of the input layer is determined by the number of features in the input data.

Hidden layers: These are intermediate layers in the neural network that transform the input data and extract features from it. The number and size of the hidden layers can vary depending on the complexity of the problem being solved. Deeper neural networks with more hidden layers are often able to learn more complex and abstract representations of the input data.

Output layer: This is the final layer of the neural network and produces the output predictions. The size of the output layer is typically determined by the number of classes or values being predicted.

Each neuron in the neural network receives inputs from the neurons in the previous layer, and produces an output based on its weighted sum of inputs, followed by the application of an activation function. The activation function is a non-linear

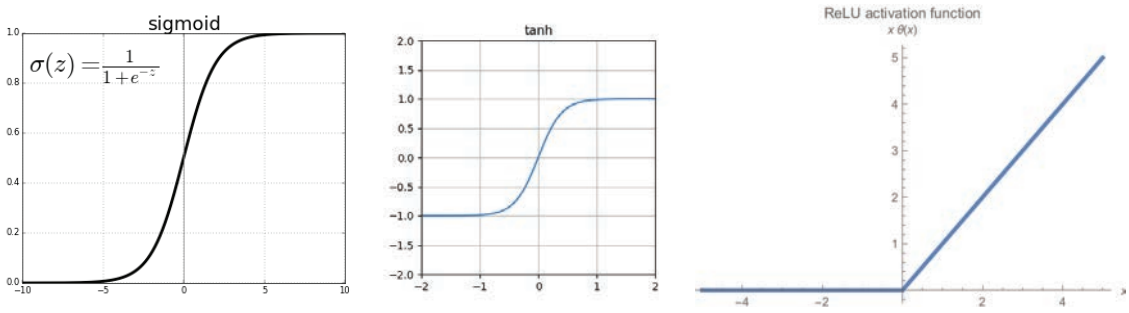


Figure 3.17: Some commonly used activation functions

function that introduces non-linearity into the output of the neuron, allowing the neural network to learn complex non-linear relationships between the input and output data. Some commonly used activation functions in neural networks include:

Sigmoid function: This function produces an output between 0 and 1, and is commonly used in the output layer of binary classification problems.

$$h(x) = \frac{1}{1 + e^{-x}}$$

tanh function: The hyperbolic tangent is a smooth function, that maps its input to the range [-1, 1]. It is zero-centred, which can help with mitigating the vanishing gradient problem, and is commonly used in recurrent neural networks (RNNs) and some variants of feedforward neural networks.

$$h(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

(ReLU) function: Rectified Linear Unit function produces an output of 0 for negative inputs and the input value for positive inputs, and is commonly used in the hidden layers of neural networks.

There are many other activation functions available, and the choice of activation function can have a significant impact on the performance of the neural network. The choice of activation function, as well as other hyper parameters such as the learning rate and number of neurons in each layer, must be carefully tuned to ensure optimal performance of the neural network.

Relation with basis function expansion: Neural network models and convolving basis function expansions are related in that they both involve the use of non-linear functions to transform input data into a higher-dimensional space in order to learn and represent complex relationships between the input and output data. In neural network models, the choice of activation function is similar to the choice of basis functions in convolving basis function expansions.

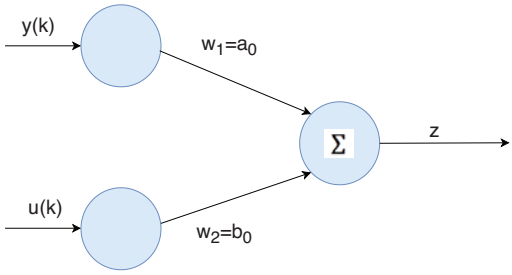


Figure 3.18: A simple ARX model as an NN model

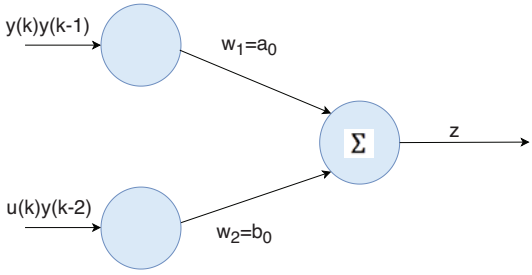


Figure 3.19: A simple NARX model as an NN model

Example 3.6. ARX model as an NN model: While these two models may seem quite different, it's possible to show that an ARX model can be considered as a specific case of a neural network model with a single output neuron and a linear activation function. The output of this neural network can be expressed as:

$$z = w_1 y(k) + w_2 u(k)$$

which corresponds to a simple ARX model with

$$y(k+1) = a_0 y(k) + b_0 u(k)$$

if we take $z = y(k+1), w_1 = a_0$ and $w_2 = b_0$.

Example 3.7. NARX model as an NN model: If we choose the specific architecture and activation function such that the output of the neural network depends only on the past values of the output and external input variables, and not on the current output or any future inputs, then the resulting model is equivalent to a NARX model. Consider the following simple neural network model with no hidden layer: The output of the NN model is:

$$z = w_1 y(k)y(k-1) + w_2 u(k)y(k-2)$$

which corresponds to a simple NARX model with

$$y(k+1) = a_0 y(k)y(k-1) + b_0 u(k)y(k-2)$$

if we take $z = y(k+1), w_1 = a_0$ and $w_2 = b_0$.

Feedforward neural network

A feedforward neural network (FFNN) is a type of artificial neural network in which the information flows only in one direction, from the input layer to the output layer, without any feedback loops. In other words, the output of one layer is used as input for the next layer, and this process continues until the final output is produced. FFNNs are widely used in a variety of applications and are relatively easy to train and can approximate complex non-linear functions, making them a powerful tool for modeling complex systems.

FFNNs can have different architectures, depending on the number of hidden layers, the number of neurons in each layer, and the type of activation function used. The choice of architecture and activation function can affect the performance of the network, and it's often determined through a trial-and-error process.

Consider the following FFNN which is a generalisation of a NARX model (see Fig. 3.20). The inputs x_i of this NN includes all second degree combinations of past inputs $u(k)$ and past outputs $y(k)$ of a nonlinear system.

$$\begin{aligned} x_i \in \{&y(k), \cdots, y(k-n), u(k), \cdots, u(k-m), y(k)^2, \cdots, y(k-n)^2, \\ &u(k)^2, \cdots, u(k-m)^2, y(k)u(k), \cdots, y(k-n)u(k-m)\} \end{aligned}$$

Assume that the input of the hidden layer s_i is the weighted sum of the inputs x_i:

$$s_i = \sum_{j=1}^{q} x_j v_{ji}$$

where the weights v_{ji} are known. Therefore the output of the NN can be represented as a linear regression:

$$z = \sum_{i=1}^{p} w_i h_i(s_i) + b = W^T H$$

where h_i is the activation function (basis function) and

$$W = [w_1, \cdots, w_p, b] \qquad ; \qquad H = [h_1, \cdots, h_p, 1]$$

Recurrent neural network

A recurrent neural network (RNN) is a type of artificial neural network that can operate on sequential data, such as time-series data, text data, and audio data, by maintaining an internal state or memory of previous inputs. In an RNN, each neuron has an additional input, which is the output of the neuron from the previous time step. This allows the network to capture the temporal dependencies between the inputs, and to maintain a state that summarizes the information from previous inputs.

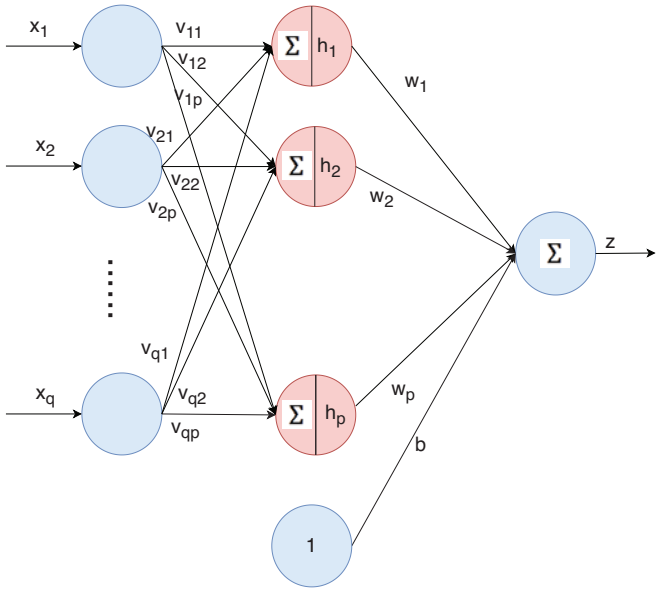


Figure 3.20: A feedforward neural network (FFNN)

Nonlinear Output Error (NOE) model can be seen as a type of Recurrent neural networks (RNNs), in which the internal state of the system is the past predicted outputs. Consider the following RNN (see Fig. 3.21). The inputs x_i of this NN includes all second degree combinations of past inputs $u(k)$ and past predicted outputs $\hat{y}(k)$ of a nonlinear system.

$$\begin{aligned} x_i \in \{&\hat{y}(k), \cdots, \hat{y}(k-n), u(k), \cdots, u(k-m), \hat{y}(k)^2, \cdots, \hat{y}(k-n)^2, \\ &u(k)^2, \cdots, u(k-m)^2, \hat{y}(k)u(k), \cdots, \hat{y}(k-n)u(k-m)\} \end{aligned}$$

The output of the NN model will be $\hat{y}(k+1) = W^T H_k(W)$, where H_k is a function of the weights. Since we have a nonlinear regression, the weights can be computed by the iterative Gauss-Newton algorithm:

$$\hat{W}_{i+1} = \hat{W}_i + \gamma \left(\sum_{k=1}^{N} H_k(\hat{W}_i) H_k^T(\hat{W}_i) \right)^{-1} \sum_{k=1}^{N} H_k(\hat{W}_i) \varepsilon_k$$

An alternative is to use the back propagation (or gradient descent) algorithm. One issue with RNNs is that the gradients used to update the weights and biases during training can become very large or very small, which can lead to problems with vanishing or exploding gradients. This can be mitigated by using techniques such as gradient clipping, or by using more advanced architectures such as Long Short-Term Memory (LSTM) networks or Gated Recurrent Units (GRUs), which are designed to better handle long-term dependencies.

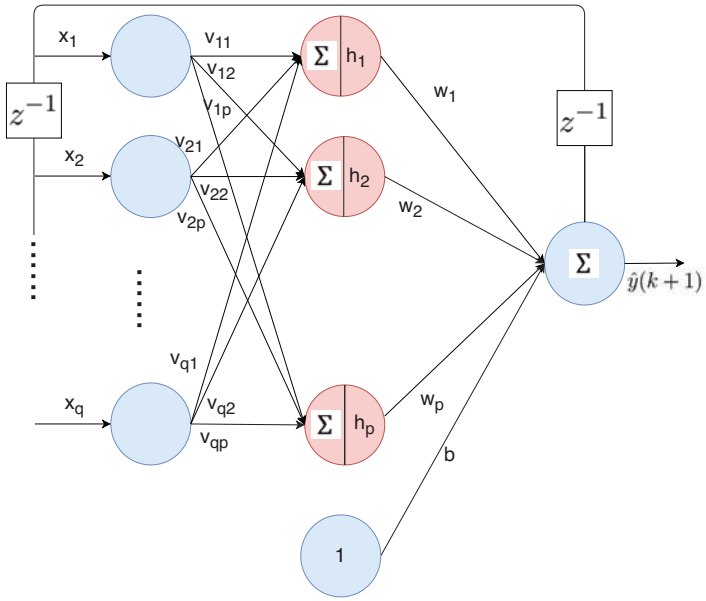


Figure 3.21: A recurrent neural network(RNN)

Back propagation method

Back propagation is the process of updating the weights and biases of the network to minimize an error function, which measures the difference between the network's predicted output and the true output for a given input. The general steps involved in training a neural network using back-propagation are as follows:

1. **Initialization:** The weights and biases of the network are initialized to random values.

2. **Forward propagation:** An input is fed into the network, and the outputs of each layer are computed using the current weights and biases.

3. **Calculation of error:** The difference between the predicted output and the true output is computed and used in a cost function, such as mean squared error.

4. **Backward propagation:** The error is propagated backwards through the network, starting from the output layer and working backwards towards the input layer. The weights and biases of each layer are adjusted to minimize the error using the chain rule of calculus.

5. **Update of weights and biases:** The weights and biases of the network are updated based on the gradients computed in the previous step, using a learning rate that determines the step size of the update. The learning rate is typically a small positive number that is chosen empirically.

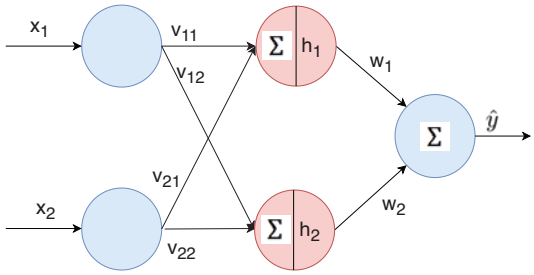


Figure 3.22: A simple NN for the back propagation algorithm

6. **Repeat:** Steps 2-5 are repeated for each training example in the dataset, and for multiple epochs or iterations until the error on the training set converges or reaches a minimum.

The back-propagation algorithm works by computing the gradients of the cost function with respect to the weights and biases of each neuron in the network. This is done using the chain rule of calculus, which allows the gradients to be propagated backwards through the network. The gradients are then used to update the weights and biases in a direction that minimizes the cost function.

Example 3.8. NN Training by back-propagation Consider the simple NN in Fig. 3.22, while the objective is to compute the weights iteratively. The inputs of the hidden layer are the weighted sum of the main inputs x_1, x_2:

$$s_1 = \hat{v}_{11}x_1 + \hat{v}_{21}x_2$$
$$s_2 = \hat{v}_{12}x_1 + \hat{v}_{22}x_2$$

and the output of the NN (predicted output) is given by:

$$\hat{y} = \hat{w}_1 h_1(s_1) + \hat{w}_2 h_2(s_2) = \hat{W}^T H$$

where the activation function is a sigmoid function: $h(s) = (1 + e^{-s})^{-1}$ and its derivative is:

$$\frac{\partial h}{\partial s} = h(s)(1 - h(s))$$

The cost function is defined as the square of the prediction error divided by 2:

$$E = \frac{1}{2}\varepsilon^2 = \frac{1}{2}(y - \hat{y})^2$$

The derivative of the cost function with respect to W is computed first using the chain rule as follows:

$$\frac{\partial E}{\partial \hat{w}_1} = \frac{\partial E}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial \hat{w}_1} = [-\varepsilon][(h_1(s_1)] = -h_1(s_1)\varepsilon \tag{3.218}$$

$$\frac{\partial E}{\partial \hat{w}_2} = -h_2(s_2) \tag{3.219}$$

$$\frac{\partial E}{\partial \hat{W}} = -H\varepsilon \tag{3.220}$$

Using the derivative, the weights W can be updated by the gradient descent algorithm:

$$\hat{W}_{i+1} = \hat{W}_i - \gamma \frac{\partial E}{\partial \hat{W}} = \hat{W}_i + \gamma H \varepsilon$$

where γ is the learning rate. In the next step the weights of the previous layer V are computed. Let's define the matrix of the weights V and the derivative of the activation function D_H as follows:

$$V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}, \qquad D_H = \begin{bmatrix} \frac{\partial h_1}{\partial s_1} & \frac{\partial h_2}{\partial s_2} \end{bmatrix}^T$$

Then again using the chain rule we compute the derivative of the cost function with respect to V. We start with the derivative with respect to v_{11} as:

$$\frac{\partial E}{\partial \hat{v}_{11}} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h_1} \frac{\partial h_1}{\partial s_1} \frac{\partial s_1}{\partial \hat{v}_{11}} = -x_1 \hat{w}_1 (\partial h_1 / \partial s_1) \varepsilon$$

Then, in the same way we compute:

$$\frac{\partial E}{\partial \hat{v}_{21}} = -x_2 \hat{w}_1 \frac{\partial h_1}{\partial s_1} \varepsilon \qquad \frac{\partial E}{\partial \hat{v}_{12}} = -x_1 \hat{w}_2 \frac{\partial h_2}{\partial s_2} \varepsilon \qquad \frac{\partial E}{\partial \hat{v}_{22}} = -x_2 \hat{w}_2 \frac{\partial h_2}{\partial s_2} \varepsilon$$

Therefore, the derivative with respect to V becomes:

$$\frac{\partial E}{\partial V} = -X(\hat{W} \circ D_H)^T \varepsilon$$

which leads to the following update law:

$$V_{i+1} = V_i - \gamma \frac{\partial E}{\partial V} = V_i + \gamma X(\hat{W} \circ D_H)^T \varepsilon$$

One issue with gradient descent is that it can get stuck in local minima or plateaus, where the gradient becomes very small or zero, preventing further progress towards the global minimum. To overcome this, various optimization techniques such as momentum, adaptive learning rates, and batch normalization can be used to improve the convergence and stability of the training process.

Batch gradient descent, stochastic gradient descent, and mini-batch gradient descent are three variations of gradient descent optimization algorithms used for training neural networks. Here are the differences between these three approaches:

Batch Gradient Descent: In batch gradient descent, the entire training dataset is used to compute the gradient of the cost function with respect to the weights and biases of the network. The weights and biases are then updated based on the average gradient over the entire dataset. Batch gradient descent is computationally expensive, but can lead to more stable updates since the gradient is computed over the entire dataset.

Stochastic Gradient Descent: In stochastic gradient descent, the weights and biases are updated after processing each training example. The gradient is computed on a single example at a time, resulting in a noisy estimate of the true gradient. Stochastic gradient descent is computationally efficient and can converge faster than batch gradient descent, but can be less stable and lead to oscillations in the optimization process.

Mini-Batch Gradient Descent: In mini-batch gradient descent, the training dataset is divided into small batches of fixed size, and the weights and biases are updated after processing each batch. The gradient is computed on a mini-batch of examples, resulting in a more accurate estimate of the true gradient than stochastic gradient descent. Mini-batch gradient descent is computationally efficient and can converge faster than batch gradient descent, while still maintaining stability in the optimization process.

In summary, batch gradient descent computes the gradient over the entire dataset, while stochastic gradient descent computes the gradient on a single example at a time. Mini-batch gradient descent is a compromise between the two, computing the gradient on a small batch of examples at a time. The choice of which method to use depends on the specific problem and the size of the dataset, with mini-batch gradient descent being the most common approach in practice.

Some basic terminology

Understanding these terms is important for designing and training neural network models, as they allow you to control the training process and optimize the performance of the network.

Epoch: An epoch is a complete pass through the entire training dataset. During each epoch, the network processes every training example once.

Batch: A batch is a subset of the training dataset that is used to update the weights and biases of the network. Instead of processing all the training examples at once, the dataset is split into smaller batches and each batch is processed one at a time.

Iteration: An iteration is a single update of the weights and biases of the network. In other words, each iteration involves processing one batch of training examples and updating the network parameters based on the errors computed for that batch.

Batch size: The batch size refers to the number of training examples in each batch. A larger batch size will result in more stable updates to the network parameters, but will also require more memory and processing time. A smaller batch size may result in less stable updates, but can allow for more frequent updates and faster training times.

Training set: The training set is a subset of the dataset used to train the network and is used to update the network parameters during the training process.

Validation set: The validation set is a subset of the dataset used to monitor the performance of the network during training. It is used to evaluate the network on examples that it has not seen during training, and can be used to detect overfitting and determine when to stop the training process.

Test set: The test set is a subset of the dataset used to evaluate the final performance of the network after training and to estimate the generalization error of the network, or its ability to perform well on new, unseen examples.

Appendix A

Random Processes and Parameter Estimation

A.1 Random Processes

Random Variable: A rule $X(\cdot)$ that assigns to every element of a sample space Ω a real value is called a random variable. So X is not really a variable that varies randomly but a function whose domain is Ω and whose range is some subset of the real values.

Example A.1. Consider the experiment of flipping a coin twice. The sample space (the possible outcomes) is :

$$\Omega = \{HH, HT, TH, TT\}$$

We can define a random variable X such that

$$X(HH) = 1, X(HT) = 1.1, X(TH) = 1.6, X(TT) = 1.8$$

Random variable X assigns to each event (e.g. $E = \{HT, TH\} \subset \Omega$) a subset of the real line (in this case $B = \{1.1, 1.6\}$).

Probability Distribution Function: For any element ζ in Ω, the event

$$\{\zeta | X(\zeta) \leq x\}$$

is an important event. The probability of this event

$$Pr[\{\zeta | X(\zeta) \leq x\}] = P_X(x)$$

is called the probability distribution function of X.

Example A.2. For the random variable defined earlier, we have:

$$P_X(1.5) = Pr[\{\zeta | X(\zeta) \leq 1.5\}] = Pr[\{HH, HT\}] = 0.5$$

$P_X(x)$ can be computed for all $x \in R$. It is clear that $0 \leq P_X(x) \leq 1$.

Note that for the same experiment (flipping a coin twice), we could define another random variable that would lead to a different $P_X(x)$. In most of the engineering problems, the sample space is a subset of the real line. Therefore, $X(\zeta) = \zeta$ and $P_X(x)$ is a continuous function of x. When we deal with a single random variable the subscript is removed.

Probability Density Function (PDF): The Probability Density Function, if it exists, is given by:

$$p(x) = \frac{dP(x)}{dx}$$

A PDF has the following properties:

$$(i) \qquad \int_{-\infty}^{\infty} p(x)dx = P(\infty) - P(-\infty) = 1$$

$$(ii) \qquad Pr[\{\zeta | X(\zeta) \leq x\}] = Pr[X \leq x] = P(x) = \int_{-\infty}^{x} p(\alpha)d\alpha$$

$$(iii) \qquad Pr[x_1 < X \leq x_2] = \int_{x_1}^{x_2} p(x)dx$$

Gaussian PDF: A random variable is distributed according to a Gaussian or normal distribution if the PDF is given by:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The PDF has two parameters : μ, the mean and σ^2 the variance. We note $X \sim \mathcal{N}(\mu, \sigma^2)$ when the random variable X has a normal (Gaussian) distribution with the mean μ and the standard deviation σ.

The Gaussian distribution is important because according to the Central Limit Theorem the sum of N independent random variables has a PDF that converges to a Gaussian distribution when N goes to infinity.

Theorem A.1. *Central Limit Theorem: Consider N independent RV* $x_1, x_2, \ldots, x_N$ *with mean* μ *and finite variance* σ^2. *Let* $\bar{x}_N = \frac{1}{N}\sum_{n=1}^{N} x_n$, *then* $z_N = \sqrt{N}\frac{\bar{x}_N - \mu}{\sigma}$ *converges in distribution to* $z \sim \mathcal{N}(0, 1)$.

Some properties of Gaussian random variables are:

- If $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ then $Z = (X - \mu_x)/\sigma_x \sim \mathcal{N}(0, 1)$.
- If $Z \sim \mathcal{N}(0, 1)$ then $X = \sigma_x Z + \mu_x \sim \mathcal{N}(\mu_x, \sigma_x^2)$.
- If $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ then $Z = aX + b \sim \mathcal{N}(a\mu_x + b, a^2\sigma_x^2)$.

- If $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$ are two independent random variables, then
$$aX + bY \sim \mathcal{N}(a\mu_x + b\mu_y, a\sigma_x^2 + b\sigma_y^2)$$
- The sum of square of n independent random variables with standard normal distribution $\mathcal{N}(0, 1)$ has a χ_n^2 distribution with n degree of freedom. For large value of n, χ_n^2 converges to $\mathcal{N}(n, 2n)$.
- The Euclidian norm $\sqrt{X^2 + Y^2}$ of two independent random variables with standard normal distribution has the Rayleigh distribution.

Independence, Expectation and Covariance

Independence: Two random variables X and Y are independent if and only if:

$$p(x, y) = p(x)p(y)$$

Expected Value: The expected value, if it exists, of a random variable X with PDF $p(x)$ is defined by:

$$\mathbb{E}\{X\} = \int_{-\infty}^{\infty} xp(x)dx$$

The expected value is a linear operator and so has the following properties:

- $\mathbb{E}\{X + Y\} = \mathbb{E}\{X\} + \mathbb{E}\{Y\}$
- $\mathbb{E}\{aX\} = a\mathbb{E}\{X\}$
- The expected value of $Y = g(X)$ can be computed by:
$$\mathbb{E}\{Y\} = \int_{-\infty}^{\infty} g(x)p(x)dx$$

Covariance: For two random variables X and Y, the covariance is defined as

$$\sigma_{xy} = \mathbb{E}\{(X - \mu_x)(Y - \mu_y)\}$$

The differences between these notations are clarified as:

- If $\sigma_{xy} = 0$, then X and Y are uncorrelated and
$$\mathbb{E}\{XY\} = \mathbb{E}\{X\}\mathbb{E}\{Y\}$$
- X and Y called orthogonal if $\mathbb{E}\{XY\} = 0$.
- If X and Y are independent then they are uncorrelated.
$$p(x, y) = p(x)p(y) \Rightarrow \mathbb{E}\{XY\} = \mathbb{E}\{X\}\mathbb{E}\{Y\}$$

- Uncorrelatedness does not necessarily imply the independence. For example, if X is a normal random variable with zero mean and $Y = X^2$ then they are not independent but

$$\sigma_{xy} = \mathbb{E}\{XY\} - \mathbb{E}\{X\}\mathbb{E}\{Y\} = \mathbb{E}\{X^3\} - 0 = 0$$

  The correlation only shows the linear dependence between two random variables so is weaker than independence.

- For jointly Gaussian random variables, independence is equivalent to being uncorrelated.

Random Vector: is a vector of random variables [1] :

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$$

Expectation Vector: $\boldsymbol{\mu}_x = \mathbb{E}\{\boldsymbol{x}\} = [\mathbb{E}\{x_1\}, \mathbb{E}\{x_2\}, \ldots, \mathbb{E}\{x_n\}]^T$

Covariance Matrix: $\boldsymbol{C}_{xx} = \mathbb{E}\{(\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^T\}$

- $\boldsymbol{C}_{xx}$ is an $n \times n$ symmetric matrix which is assumed to be positive definite and so invertible.
- The elements of this matrix are : $[\boldsymbol{C}_{xx}]_{ij} = \mathbb{E}\{[x_i - \mathbb{E}\{x_i\}][x_j - \mathbb{E}\{x_j\}]\}$.
- If the random variables are uncorrelated then $\boldsymbol{C}_{xx}$ is a diagonal matrix.

Multivariate Gaussian PDF:

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n \det(\boldsymbol{C}_{xx})}} e^{\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_x)^T \boldsymbol{C}_{xx}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_x)\right]}$$

Discrete Random Process: $x(k)$ is a sequence of random variables defined for every integer k.

Mean value: is defined as $\mathbb{E}\{x(k)\} = \mu_x(k)$.

Autocorrelation Function (ACF): is defined as

$$R_{xx}(h, k) = \mathbb{E}\{x(k)x(k-h)\}$$

Wide Sense Stationary (WSS): $x(k)$ is WSS if its mean and its autocorrelation function (ACF) do not depend on k.

Autocovariance function: is defined as

$$C_{xx}(h) = \mathbb{E}\{[(x(k) - \mu_x][x(k-h) - \mu_x)] = R_{xx}(h) - \mu_x^2$$

[1] In many engineering books, there is no distinction between random variable X and its specific value x. From now on we follow the same rule.

Cross-correlation Function (CCF): is defined as

$$R_{xy}(h) = \mathbb{E}\{x(k)y(k-h)\}$$

Cross-covariance function: is defined as

$$C_{xy}(h) = \mathbb{E}\{[(x(k)-\mu_x][y(k-h)-\mu_y)] = R_{xy}(h) - \mu_x\mu_y$$

Discrete White Noise: is a discrete random process with zero mean and $R_{xx}(h) = \sigma^2\delta(k)$ where $\delta(k)$ is the Kronecker impulse function and σ^2 is the noise variance. The power spectral density of white noise becomes $\Phi_{xx}(\omega) = \sigma^2$ and is completely flat with frequency.

A.2 Parameter Estimation Problem

Given a set of measured data

$$\boldsymbol{x} = \{x(0), x(1), \ldots, x(N-1)\}$$

which depends on an unknown parameter vector $\boldsymbol{\theta}$, the parameter estimation problem is to determine an estimator

$$\hat{\boldsymbol{\theta}} = \mathcal{F}(x(0), x(1), \ldots, x(N-1))$$

where $\mathcal{F}$ is some function.

Example A.3. Consider a data sequence that can be modeled with a linear trend in white Gaussian noise

$$x(k) = A + Bk + w(k) \qquad k = 1, \ldots, N-1$$

Suppose that $w(k) \sim \mathcal{N}(0, \sigma^2)$ and is uncorrelated with all the other samples. Letting $\boldsymbol{\theta} = [A\ B]$ and $\boldsymbol{x} = [x(1), x(2), \ldots, x(N)]$ the PDF is:

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \prod_{k=1}^{N} p(x(k); \boldsymbol{\theta}) = \frac{1}{(\sqrt{2\pi\sigma^2})^N} e^{\left[-\frac{1}{2\sigma^2}\sum_{k=1}^{N}(x(k)-A-Bk)^2\right]}$$

Unbiased estimator: An estimator that *on the average* yield the true value is unbiased. Mathematically:

$$\mathbb{E}(\theta - \hat{\theta}) = 0 \qquad \text{for } a < \theta < b$$

Remark: When several unbiased estimators of the same parameters from independent set of data are available, i.e., $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_n$, a better estimator can be obtained by averaging:

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} \hat{\theta}_i \quad \Rightarrow \quad \mathbb{E}(\hat{\theta}) = \theta$$

Assuming that the estimators have the same variance, we have:

$$\text{var}(\hat{\theta}) = \frac{1}{n^2}\sum_{i=1}^{n}\text{var}(\hat{\theta}_i) = \frac{1}{n^2}n\,\text{var}(\hat{\theta}_i) = \frac{\text{var}(\hat{\theta}_i)}{n}$$

By increasing n, the variance will decrease (if $n \to \infty, \hat{\theta} \to \theta$). This is not the case for biased estimators, no matter how many estimators are averaged.

Maximum Likelihood Estimator (MLE):

Choose the parameter value that makes the *observed data*, the most likely data to have been observed. The maximum likelihood estimate is the value of θ that maximizes the likelihood function. The likelihood function is the PDF $p(x;\theta)$, when θ is regarded as a variable (not a parameter). Instead of computing the maximum of the likelihood function the maximum of its logarithm is usually computed. For the signals with Gaussian distribution, this technique reduces significantly the computational complexity.

Example A.4. Consider the general linear model $\boldsymbol{y} = \boldsymbol{\Phi\theta} + \boldsymbol{w}$ where $\boldsymbol{w}$ is a noise vector with PDF $\mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$:

$$p(\boldsymbol{y};\boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^N \det(\boldsymbol{C})}} e^{\left[-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\Phi\theta})^T\boldsymbol{C}^{-1}(\boldsymbol{y}-\boldsymbol{\Phi\theta})\right]}$$

Taking the derivative of $\ln p(\boldsymbol{y};\boldsymbol{\theta})$ leads to:

$$\frac{\partial \ln p(\boldsymbol{y};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial(\boldsymbol{\Phi\theta})^T}{\partial \boldsymbol{\theta}}\boldsymbol{C}^{-1}(\boldsymbol{y}-\boldsymbol{\Phi\theta})$$

Then

$$\boldsymbol{\Phi}^T\boldsymbol{C}^{-1}(\boldsymbol{y}-\boldsymbol{\Phi\theta}) = 0 \quad \Rightarrow \quad \hat{\boldsymbol{\theta}} = (\boldsymbol{\Phi}^T\boldsymbol{C}^{-1}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{C}^{-1}\boldsymbol{y}$$

The PDF of $\hat{\boldsymbol{\theta}}$ is:

$$\hat{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\theta}, (\boldsymbol{\Phi}^T\boldsymbol{C}^{-1}\boldsymbol{\Phi})^{-1})$$

References

- K. J. Aström and B. Wittenmark, *Adaptive Control*, Addison-Wesley, 1989.
- G. E. P. Box , G. M. Jenkins and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, Prentice-Hall, 1994.
- C. T. Chen, *Linear System Theory and Design*, Holt, Rinehart and Winston, 1984.
- F. de Coulon, *Théorie et traitement des signaux*, Traité d'Electricité Vol. VI, Presses polytechniques romandes, 1984.
- J. M. Flaus, *La régulation industrielle*, Hermès, 1994.
- G. F. Franklin and J.D. Powell, *Digital Control of Dynamic Systems*, Addison-Wesley, 1980.
- D. Graupe, *Identification of Systems*, Krieger, 1976.
- R. Isermann, K.H. Lachmann and D. Matko, *Adaptive Control Systems*, Prentice-Hall, 1992.
- R. G. Jacquot, *Modern Digital Control Systems*, Marcel Dekker, 1981.
- R. Johansson, *System Modeling and Identification*, Prentice-Hall, 1993.
- J. N. Juang, *Applied System Identification*, Prentice-Hall, 1994.
- H. Kwakernaak and R. Sivan, *Modern Signal and Systems*, Prentice-Hall, 1991.
- I. D. Landau, *Identification et commande des systèmes*, Hermès, 1993.
- I. D. Landau, R. Lozano, M. M'Saad and A. Karimi, *Adaptive Control,Algorithms, Analysis and Applications* , Springer-Verlag, 2011.
- P. de Larminat and Y. Thomas, *Automatique des systèmes linéaires. 2. Identification*, Flammarion Sciences, 1977.
- J. R. Leigh, *Modeling and Simulation*, IEE Topics in Control Series 1, Peter Peregrinus Ltd, 1983.

- L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, 1998.
- L. Ljung and T. Glad , *Modeling of Dynamic Systems*, Prentice-Hall, 1994.
- L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, MIT Press, 1987.
- M. O'Flynn and E. Moriarty, *Linear Systems. Time Domain and Transform Analysis*, John Wiley, 1987.
- K. Ogata, *Discrete-Time Control Systems*, Prentice-Hall, 1987.
- R. Petit, *L'outil mathématique*, Masson, 1991.
- J. Richalet, *Pratique de l'identification*, Hermès, 1998.
- J. Richalet, A. Rault and R. Pouliquen, *Identification des processus par la méthode du modèle*, Gordon & Breach, 1971.
- J. Schoukens and R. Pintelon, *Identification of Linear Systems*, Pergamon Press, 1991.
- B. Shahian and M. Haussul, *Control System Design Using MATLAB*, Prentice-Hall, 1993.
- N. K. Sinha,*Linear Systems*, John Wiley, 1991.
- N. K. Sinha and B. Kuszta, *Modeling and Identification of Dynamic Systems*, Van Nostrand Reinhold, 1983.
- J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Y. Glorennec, H. Hjalmarsson and A. Juditsky, *Nonlinear Black-box Modeling in System Identification: a Unified Overview*, Automatica, 31(12), 1691–1724, 1995.
- T. Söderström and P. Stoica, *System Identification*, Prentice-Hall, 2001ö.
- P. Stoica and R. Moses, *Introduction to Spectral Analysis*, Prentice-Hall, 1997.
- H. Unbehauen, *Regelungstechnik I-III*, Vieweg, 1985.
- J. Van De Vegte, *Feedback Control Systems*, Prentice-Hall, 1994.
- P. P. J. Van Den Bosch and A. C. Van Der Klauw, *Modeling, Identification and Simulation of Dynamical Systems*, CRC Press, 1994.
- E. Walter L. Pronzato, *Identification de modèles paramétriques à partir de données expérimentales* , Masson, 1994.
- Y. Zhu and T. Backx, *Identification of Multivariable Industrial Processes*, Springer-Verlag, 1993.