Università
degli Studi
di Padova

Sede Amministrativa:
UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Matematica "Tullio Levi-Civita"
Corso di Dottorato di Ricerca in: Matematica
Curriculo: Matematica Computazionale
Ciclo XXXIII

NUMERICAL METHODS FOR MODEL-BASED DESIGN OF PHYSICAL SYSTEMS

Tesi redatta con il contributo finanziario di Electrolux S.p.a.

Coordinatore:
Ch.mo Prof. Martino Bardi

Supervisore:
Prof. Fabio Marcuzzi

Dottoranda:
Marta Gatto

Abstract

This thesis treats the topic of Model-Based Design of Physical Systems.

Mathematical Modeling is a vast and multidisciplinary field which ranges different sciences and, depending on the application and the technique used, can be interpreted in different ways. The approach we consider in this thesis combines techniques from inverse and ill-posed problems, regularization and numerical linear algebra, and applies them also to signal processing.

This thesis will address the case of models driven by known Physical Equations, while the black-box (or data driven) case, which means models with structures that do not arise from physical relations on the system, will only be mentioned. Moreover, even if models are generally used for different purposes, the hidden aim of the modeling problems treated in this thesis is the industrial application of virtual prototypes and digital twins. This is the reason for using the term "*Model-Based design*", which refers to the process of describing an industrial *plant*, for example a machine or a process, that needs to be controlled. The aim is to use mathematical models to control and analyze the system.

The thesis will address different problems encountered during the three years of the doctoral degree, with particular attention to two main topics that led to publications [22, 23]. When dealing with models, in some cases the structure is known from physical hypothesis and only some parameters have to be deduced from data, in other cases the model may be well known and used to denoise measurements.

The thesis will be divided in two parts corresponding to these two cases. The first part will treat parameter estimation with different kinds of models (static and dynamic, continuous and discrete) in linear and nonlinear cases. An overview of different numerical methods will be proposed and then the problem of Unbiased Least Squares (ULS) will be introduced and analyzed [23], which consists in the problem of parameter estimation of a linear system in the case of unmodeled dynamics.

The second part will be based on numerical methods for the denoising of data from a physical system, through the knowledge of the true physical model that describes the system. The case of input-output denoising of a DLTI (Discrete Linear Time-Invariant) system with unknown noise covariance values is treated and the algorithm WMC-MBD is proposed [22]. In the end, the last Chapter deals with the extension to the nonlinear case at its current development status.

Note: The PhD scholarship was funded by the Company Electrolux spa, with title "High Performance Calculation for Model Based Design applied to home appliances design". Because of privacy agreement with Electrolux spa, we decided not to include in this thesis the projects and applications related to the research activity conducted within the Company, but only the theoretical aspects of this research.

Sommario

Questa tesi tratta il problema del Model-Based Design di Sistemi Fisici. La Modellistica Matematica è un campo molto vasto e multidisciplinare, che spazia diversi campi delle scienze, e può essere interpretata in modi diversi a seconda delle applicazioni e delle tecniche utilizzate. L'approccio che utilizzeremo in questa tesi combina tecniche di problemi inversi e mal-posti (ill-posed problems), regolarizzazione e algebra lineare numerica che vengono applicati anche all'elaborazione dei segnali (signal processing).

In questa tesi verranno trattati modelli retti da Equazioni Fisiche, mentre il caso black-box (o data driven), termine col quale si intendono modelli la cui struttura non emerge da relazioni fisiche del sistema, verranno solamente menzionati. Inoltre, nonostante gli utilizzi dei modelli siano diversi, lo scopo sottostante ai problemi di modellistica qui trattati è l'applicazione industriale di prototipi virtuali (virtual prototypes) e digital twins. Questo è il motivo per cui è stato utilizzato il termine "*Model-Based design*", che si riferisce al processo di descrizione di un sistema industriale, ad esempio un macchinario o un processo fisico, che deve essere controllato. Lo scopo è utilizzare modelli per controllare e analizzare il sistema.

In questa tesi verranno trattati diversi problemi incontrati durante i tre anni di dottorato, con particolare attenzione a due temi principali che hanno portato a pubblicazioni [22, 23]. Nel percorso di modellistica si possono verificare due situazioni: in alcuni casi la struttura del modello è nota da ipotesi fisiche e solamente alcuni parametri sono sconosciuti e da ricavare a partire dalle misure; in altri casi il modello può essere noto e utilizzato per togliere il rumore dalle misure e ricavare delle stime dei segnali. La tesi sarà suddivisa in due parti corrispondenti a questi due casi. La prima parte sarà focalizzata sulla stima di parametri con diversi tipi di modelli (statici e dinamici, a tempo continuo o discreto) nei casi lineare e nonlineare. Dopo una descrizione di diversi metodi numerici, introdurremo e analizzeremo il problema degli "Unbiased Least Squares" (ULS) [23], ossia il problema di stima dei parametri di un sistema lineare nel caso di dinamiche non modellate.

Il focus della seconda parte saranno i metodi numerici per il denoising dei dati provenienti da sistemi fisici, ossia i metodi di rimozione del rumore, tramite la conoscenza del vero modello fisico che descrive il sistema. Verrà esposto il problema del denoising di input e output di un sistema DLTI (Discreto Lineare Tempo-Invariante) con covarianze dei rumori ignote, assieme ad un algoritmo per la sua risoluzione [22]. Infine, sarà studiata l'estensione di questo problema al caso nonlineare.

Note: La borsa di dottorato è stata finanziata dall'industria Electrolux spa, con titolo "Calcolo ad alte prestazioni per il Model-Based Design applicato alla progettazione di elettrodomestici". A causa dell'accordo di confidenzialità e riservatezza con Electrolux spa, abbiamo deciso di non includere in questa tesi i progetti e le applicazioni sviluppati in collaborazione con essa, e solamente gli aspetti teorici della ricerca sono stati trattati.

Contents

Introduction and motivations

Mathematical physical models are often used for the description of physical phenomena and are essential in industrial applications for various aims, such as control and estimation of unmeasurable variables and physical parameters.

We start with some applications to give the idea of the importance of mathematical modeling in industry and justify the attention to the topic.

Examples of industrial applications The uses of models in industrial applications are very different and are referred to with various expressions: predictive and adaptive control, model-based design and shape optimization, indirect measures and virtual sensors, fault detection and predictive maintenance, virtual testing, parameter estimation, digital twins and prototypes and lots of others. We describe for example some of these cases:

1. Shape Optimization and Model-Based Design, are terms referred to the case in which an analytical and numerical study on the shape of a certain object is done through the optimization of mathematical equations to achieve certain desired performances before building a physical prototype;

2. Indirect measures and Virtual Sensors are terms that indicate the calculation of a certain quantity through physical equations, when it is not possible to measure it with true sensors, for example when the environment has a too high temperature or the sensor is too expensive;

3. The monitoring of the good functioning of a machine can be checked through models and this is referred to as Fault Detection or Predictive Maintenance;

4. In general the simulation of a model that describes a machine is called Virtual Testing, and it may be useful when the tests on the machine are expensive, dangerous, or when the machine is not available (suppose for example that workers are forced to smart-working at home for a certain time).

Experimental Physical Modelling As various the applications in which models are used, as various are the mathematical models that can be chosen. The activity of deriving a model from data is called with different terms, depending on the approach and the techniques used. Some examples are *System Identification*, *Signal Estimation or enhancement* [12, 14], and *Experimental Modelling*, all of which indicate the techniques used to extract the useful information of the system from the available measurement corrupted with noise. The structure of these procedures has some basic common concepts that can be pointed out. The process starts with the collection of some data of the system under study, that comes with a measurement error. Sometimes it is also possible to choose how to collect the data from the system (and this step is called Experiment Design). Then, some physical information about the system and relations among the measured (and also unmeasured) quantities can be collected. Among the relations and details on the system, only the ones useful to the aim of the problem must be considered.

The procedure can, in general, be divided in three steps.

1. In each case we choose a *model* structure to describe the system: the range goes from simpler to more complex. We can divide them in the following characterizations: linear or nonlinear, static or dynamic, with lumped or distributed parameters (Ordinary Differential Equations or Partial Differential Equations), dynamic with continuous or discrete time. Another distinction is the one between White Box models (i.e. governed by physical equations, in which parameters have physical meaning) and Black Box ones (or data-driven, in which parameters do not have physical meaning).

2. Secondly, we choose a *criterion function* to minimize, that represents our identification problem. For example deterministic (square error, absolute error, ...) or probabilistic (maximum likelihood error, ...). This criterion function is defined in such a way that its minima represent a "good" choice of the model parameters, i.e. the model simulation with those parameters gives results that are close to some experimental data collected from the real physical system.

3. In the end, we have to choose the numerical *algorithm* to solve the minimization of the criterion function (for example batch or recursive methods).

The choice of the model structure can be done following the main properties that a good model should possess:

- generality: is the model still able to describe the system for little modifications of the setting? For example the Hook's law that describes a spring is valid for springs with different stiffness factors;

- predictability: is the model able to describe the phenomena also in situations that were not used in the creation of the model?

- simplicity: this is a really important point and can be summarized with the Occam's Razor principle, and described by the quotation: "Everything should be made as simple as possible, but not simpler", Albert Einstein. This means that in the description of the model, only useful information must be taken into consideration, while non-useful details of the phenomena must be neglected.

The last one is a very important principle in modelling, and is the reason why simple model equations are the most widespread and we will focus on them. We will show different model structures, starting with Linear Least Squares for Linear Systems and the Nonlinear case, following the three-step structure described above. Consequently, we will introduce the State-Space form and the Discrete Linear Time Invariant systems (DLTI).

Thesis Outline We list in more details the contents of this thesis. In the first part the problem of model estimation is treated, following the previously mentioned subdivision of the procedure in models, criterion function and algorithms. First some model structures are introduced in Chapter 1 and some common formulation of the identification problems are described in Chapter 2. Different numerical methods to solve parameter estimation problems are explained in Chapter 3. After this overview of general methods from literature, we treat one of the main contributions of the thesis in Chapter 4 with the problem of unmodeled dynamics in the linear case, i.e. Unbiased Least Squares (ULS) problem.

The second part of the thesis deals with the case in which the model is known and is used to estimate the state and denoise the input and output signals of the system. In Chapter 5 the problem in the case of unknown covariance values is introduced, which is the second main contribution. In Section 5.2 the well known Kalman Algorithm is explained as a starting point and in 5.5 the WMC-MBD algorithm is proposed. The last Chapter 6 is the nonlinear extension of the previous one and concludes the thesis.

Part I

Model Estimation:
when the model is not known

Chapter 1

Model structures

1.1 Modeling noise

One of the starting concepts in signal processing is the difference between *deterministic signals*, and *random* ones. The first are repeatable, i.e. repeated measurements in the same conditions provide the same signal; while the second ones are not.

Since all the measures we obtain from real systems are corrupted with noise, the purpose of signal processing is the one of extracting the real deterministic information from the noisy ones. This is done thorough models and algorithms and hypothesis on the noise.

The term "model-based" processing was used in literature [14, 12] to represent the introduction of the description of noise inside the description of the system, to exploit as much information as possible from the system.

We will talk mainly about deterministic models, those in which all the variables are deterministic, and only briefly about stochastic models, in which some random terms will be added to describe noise.

1.1.1 Probability preliminaries

We introduce some basic probability concepts ([41], [77] and [12, 13, 14]) to add some noise terms in the models that will be introduced in the following Sections.

Consider a *probability space* $(\Omega, \mathcal{S}, P)$, i.e. the triplet with Ω the sample space, $\mathcal{S}$ a σ-algebra on it (the set of events), and P a probability measure, i.e. a measure with $P(\Omega) = 1$. Then a *random variable* with values in $\mathbb{R}^n$ is a Borel measurable function $X : \Omega \to \mathbb{R}^n$. If $n > 1$ we also refer to X as a random vector.

The random variable X defines a probability measure on $(\Omega, \mathcal{S})$ by $P_X(B) = P(X^{-1}(B)) = P(X \in B)$ with $B \in \mathcal{B}^n$ Borel sets of $\mathbb{R}^n$.

The random variable X is said to be absolutely continuous if P_X is absolutely continuous w.r.t. the Lebesgue measure on $\mathbb{R}^n$. Recall that, given two finite measures μ, ν on a measurable space $(M, \mathcal{M})$, we say μ is absolutely continuous with respect to ν if whenever $\nu(A) = 0$ for $A \in \mathcal{M}$, then $\mu(A) = 0$ and it is denoted $\mu \ll \nu$.

Definition 1 (Probability density function (Radon-Nikodym theorem))**.** *If X is an absolutely continuous random variable, then there exists a function called* probability density function $f_X : \mathbb{R}^n \to \mathbb{R}$ *such that*

$$P(X \in A) = P_X(A) = \int_A f_X(x)dx \quad \forall A \in \mathcal{B}^n.$$

We recall the following definitions:

$$\begin{aligned}
\text{Expected value (mean):} \quad & m_X \quad = \mathbb{E}[X] = \int_{\mathbb{R}^n} x f_X(x)dx, \\
\text{Cross-Correlation:} \quad & C_{XY} = \mathbb{E}[X\,Y], \\
\text{Covariance:} \quad & R_{XY} = \mathbb{E}[(X - m_X)(Y - m_Y)].
\end{aligned}$$

The definitions of mean and covariance for random vectors can be written more explicitly as:

$$m_X = \begin{bmatrix} E[X_1] \\ \vdots \\ E[X_n] \end{bmatrix} \quad \text{and covariance matrix } C_X \text{ equal to} \quad R_X = \begin{bmatrix} R_{X_1,X_1} & \dots & R_{X_1,X_n} \\ R_{X_2,X_1} & & R_{X_2,X_n} \\ \vdots & \ddots & \vdots \\ R_{X_n,X_1} & \dots & R_{X_n,X_n} \end{bmatrix}.$$

We will need the following particular case of random variable:

Definition 2. *A one dimensional Gaussian (or normal) random variable X is defined by its probability density function:*

$$f_X(\alpha) = \frac{1}{\sqrt{2\pi R_{XX}}} \exp\left\{-\frac{1}{2}\frac{(\alpha - m_X)^2}{R_{XX}}\right\} \quad \textit{i.e.} \quad X \sim \mathcal{N}(m_X, R_{XX})$$

with mean $m_X \in \mathbb{R}$ and covariance $R_{XX} \in \mathbb{R}^+$.

Definition 3. *A Gaussian (or normal) random vector is a random vector $X = (X_1, \dots, X_n)$ with all the X_i Gaussian random variables. Its probability density function is given by*

$$f_X(\alpha) = \frac{1}{(2\pi)^{n/2} \det(R_X)^{1/2}} \exp\left\{-\frac{1}{2}(\alpha - m_X)^T R_X^{-1}(\alpha - m_X)\right\}$$

with mean $m_X \in \mathbb{R}^n$ and covariance matrix $R_X \in \mathbb{R}^{n\times n}$.

A Gaussian random vector where all the X_i are Gaussian random variables with zero mean and the same covariance R_{X_i} is called White noise vector *(or White Gaussian noise vector) and its covariance matrix is a scaled identity $R_X = R_{X_i} I_n$*

1.1.2 Time dependence

A *Random Signal* or *Stochastic Process* can be seen as a sequence of ordered in time random variables, more precisely:

Definition 4. *Let $I \subset \mathbb{R}$. A family of random variables $X = (X(t), t \in I)$ (on $(\Omega, \mathcal{S}, P)$) with values in $(\mathbb{R}^n, \mathcal{B}^n)$ is called a stochastic process with index set (or time set) I and range $\mathbb{R}^n$. When $I = [0, +\infty)$, it is called* continuous-time *stochastic process, while if $I = \mathbb{N}_0$ or $I = \mathbb{Z}$ it is called* discrete-time *stochastic process.*

Given a stochastic process X, fixed a time $t \in I$, the function $X(t, \cdot)$ is a random vector, and each of its components $X_i(t, \cdot)$ is a one dimensional random variable.

We recall some useful definitions we will use from now on: given X, Y two random signals, we define

$$\begin{aligned}
\text{Expected value (mean):} \quad & m_X(t) \quad = \mathbb{E}[X(t)] = \int_{\mathbb{R}^n} x f_{X(t,\cdot)}(x) dx, \\
\text{Auto-Correlation:} \quad & C_{XX}(t,k) = \mathbb{E}[X(t)X(k)], \\
\text{Cross-Correlation:} \quad & C_{XY}(t,k) = \mathbb{E}[X(t)Y(k)], \\
\text{Variance:} \quad & R_{XX}(t,k) = \mathbb{E}[(X(t) - m_X(t))(X(k) - m_X(k))], \\
\text{Covariance:} \quad & R_{XY}(t,k) = \mathbb{E}[(X(t) - m_X(t))(Y(k) - m_Y(k))].
\end{aligned}$$

A *White Gaussian noise signal* X with values in $(\mathbb{R}^n, \mathcal{B}^n)$ is a Random Signal such that $m_X(t) = 0 \quad$ for $t, k \in I$ and $C_{XX}(t,k) = \sigma\, I_n\, \delta_{t,k} \quad$ for $t, k \in I$ where

- σ is the correlation value of each random variable $X_i(t)$ component of the random vectors $X(t)$ for each $t \in I$,

- I_n is the identity matrix in $\mathbb{R}^{n \times n}$,

- $\delta_{t,k}$ is the Kronecker delta, defined such that $\delta_{t,k} = 1$ if $t = k$ and $\delta_{t,k} = 0$ otherwise.

1.2 Static models

We start with static models, which can be defined as models that represent a phenomenon at a given point in time. These models describe the relation among different quantities of a system at each fixed instant, hence, the variation of these quantities in time is not taken into account. In the following subsections we describe in more details linear and nonlinear static models.

1.2.1 Linear models

Given a matrix $A \in \mathbb{R}^{m\times n}$, vectors $y \in \mathbb{R}^m$, and $x \in \mathbb{R}^n$ we consider the linear system

$$y = Ax. \tag{1.1}$$

The "direct" and easier problem is the calculation of y, given A and x. We are interested in the famous linear *inverse problem* of the calculation of x, given A and y.

Calling a_i the columns of the matrix, and x_i the components of the vector x, the linear system above can be written also as

$$a_1x_1 + \cdots + a_nx_n = y$$

and we can see that y is a linear combination of the columns of the matrix A.

Usually, the linear system is built from data of a system as follows: we know that an output variable $y \in \mathbb{R}^m$ can be described by a known linear system $y = Ax$ with parameter vector $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m\times n}$. Moreover, the matrix A may be built from an input variable $u \in \mathbb{R}^l$, so that $A = A(u)$. The problem of finding the parameter vector from measurements of the variables of the system is called *linear parameter estimation*.

Observation 1. *Note that the assumption we are making here is that the model is linear in the parameters. The matrix A may depend in a linear or nonlinear way on a vector of input parameters $u \in \mathbb{R}^m$ as $A = A(u)$.*

This is one of the most widespread and studied systems and arise in applications from engineering, statistics, physics, economics, biology, medicine and others.

Although the linear model is very simple, it is sufficient to describe lots of different situations.

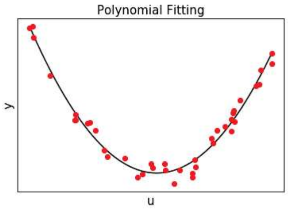


Figure 1.1: Polynomial fitting or regression

Example 1 (Example of a static model: Polynomial regression)**.** *One of the basic examples is the polynomial regression or fitting, in which the aim is to describe the quantity y as a polynomial in the variables u. In Figure 1.1 an example of a second order polynomial fitting is* *shown. Given measurements at samples of time $t_0, \dots, t_N$, we can build the linear system*

$$\begin{bmatrix} 1 & u(t_0) & u(t_0)^2 \\ 1 & u(t_1) & u(t_1)^2 \\ \vdots & \vdots & \vdots \\ 1 & u(t_N) & u(t_N)^2 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y(t_0) \\ y(t_1) \\ \vdots \\ y(t_N) \end{bmatrix}. \tag{1.2}$$

1.2.2 Nonlinear models

When a linear model is not sufficient to describe the system at hand, we must consider a nonlinear *model* of the kind

$$y = f(u, x) \tag{1.3}$$

where $y \in \mathbb{R}^m$ and $u \in \mathbb{R}^l$ are vectors of measured quantities of which we know the physical relation described by the nonlinear function f and we want to estimate a set of parameters $x \in \mathbb{R}^n$.

Example 2 (Example of a static model). *In the static case we can consider a simple example of Nonlinear exponential fitting, of which an example is shown in Figure 1.2.*

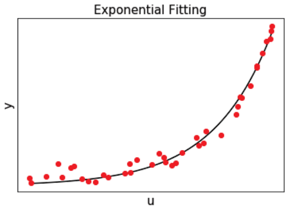


Figure 1.2: Exponential fitting

Given two quantities y and u, we want to describe the variable y with the exponential function

$$y = f(u) = x_1 e^{x_2 u}$$

estimating the value of the parameter vector $x = [x_1, x_2] \in \mathbb{R}^2$.

1.3 Dynamic models

1.3.1 Differential equations

Since most of the Physical Laws that describe Physical Systems assume the form of Ordinary and Partial differential equations (ODE and PDE) we will analyze the models given by these equations. First of all these equations relate variables and their rate of change, usually in time and space.

The common "direct" problem that arises from these equations is the one of simulating the system given the initial conditions (and the boundary conditions in the case of PDE). The inverse problem in this case consists in looking for the initial conditions and/or the parameters of the model, given some measurements of the variable y.

Note that the problem of estimating the initial condition of the differential equation, or the boundary conditions of the partial differential equations, can be seen as a particular case of parameter estimation.

In the case of ODE, we will consider models described by ordinary differential equations of the form

$$\begin{cases} \dot{y}(t) = f(y(t), u(t), p) & \text{for } t \in [0, T] \\ y(0) = y_0 \end{cases} \tag{1.4}$$

where $p \in \mathbb{R}^n$ is the parameter vector, $y_0 \in \mathbb{R}^m$ is the initial condition, f is a function which may not depend explicitly from the independent variable t, and u is an input variable ($u(t) \in \mathbb{R}^l$) that can be present or not.

Discretization methods Numerical methods for the discretization of ODEs of the kind (1.4) are a well studied topic, that we are not going to treat here and we only refer to the literature (for example [38, 11]).

In this work we will consider only Explicit Euler discretization, that we briefly recall here.

Given the Cauchy problem

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(0) = y_0 \end{cases} \tag{1.5}$$

for $t \in [t_0, T]$ and $y \in \mathbb{R}^n$.

We discretize the variable t with samples $t_n = t_0 + nh$ where h is the discretization step, for $t_n \in \{t_0, \dots, t_N = T\}$. The discretization methods give the approximated vales $y_n \approx y(t_n)$. The Explicit Euler scheme is

$$y_{n+1} = y_n + hf(t, y_n). \tag{1.6}$$

In the problems we will deal with in this thesis, we are going to neglect the discretization error $e_n = y_n - y(t_n)$, always with the hypothesis of a discretization step h sufficiently small.

Example 3 (Example of a dynamic model: the mechanical equation of a motor)**.** *Consider the mechanical equation of a motor*

$$J_M \frac{d\omega}{dt}(t) + B_M \omega(t) = T_M(t) - T_L(t),$$

with

- *ω the angular speed,*
- *J_M the inertia of the motor,*
- *B_M the coefficient of the friction term $B_M\omega(t)$,*
- *T_M and T_L the mechanical and load torque values.*

Suppose to know the measurements of $T_M, T_L, \omega, \frac{d\omega}{dt}$ and to need the estimate of the parameters J_M and B_M. We can write the problem as a linear system as follows

$$\begin{bmatrix} \frac{d\omega}{dt}(t_0) & \omega(t_0) \\ \frac{d\omega}{dt}(t_1) & \omega(t_1) \\ \vdots & \vdots \\ \frac{d\omega}{dt}(t_N) & \omega(t_N) \end{bmatrix} \begin{bmatrix} J_M \\ B_M \end{bmatrix} = \begin{bmatrix} T_M(t_0) - T_L(t_0) \\ T_M(t_1) - T_L(t_1) \\ \vdots \\ T_M(t_N) - T_L(t_N) \end{bmatrix}.$$

Example 4 (Example of a dynamic model:)**.** *The Lorentz model*

$$\begin{cases} \frac{dy_1}{dt}(t) & = -py_1(t) + py_2(t), \\ \frac{dy_2}{dt}(t) & = (r - y_3(t))y_1(t) - y_2(t), \\ \frac{dy_3}{dt}(t) & = y_1(t)y_2(t) - by_3(t), \end{cases} \tag{1.7}$$

is nonlinear with respect to the parameters p, r (Prandtl and Rayleigh numbers) while b is a known parameter. This system of equations has a peculiarity, indeed for high values of the Rayleigh number the system is near to chaotic, i.e., for little perturbations of the initial condition there are big variations in the dynamic.

We will see in Section 3.3.2 and 3.3.3 that this property of the system has important consequences on the choice of the numerical method in the case of initial condition estimation, problem that fall into the group of parameter estimation problems.

1.3.2 State-Space models

Until now, we considered models in which there were two sets of variables, an independent variable vector u and a dependent vector y. These kinds of models are called *Input-Output*.

A State-Space model is characterized by the presence of three main variables, not only the input vector u and the output y, but also the *state vector* x. Usually, the state is also called internal or hidden variable, because some of its components may be not measurable. More precisely, the *state* of a system at time t is the minimum set of variables that, with the input, is sufficient to uniquely specify the dynamic system behaviour for all t over the interval $[t_0, \infty)$.

The general time-variant formulations in the nonlinear and linear case are the following: **Continuous Nonlinear Time Variant State-Space Models**

$$\text{Nonlinear:} \quad \begin{cases} \dot{x}(t) = a(t, x(t)) + b(t, u(t)) \\ y(t) = c(t, x(t)) + d(t, u(t)) \end{cases} \tag{1.8}$$

Continuous Linear Time Variant State-Space Models

$$\text{Linear:}\quad \begin{cases} \dot{x}(t) = A_c(t)x(t) + B_c(t)u(t) \\ y(t) = C_c(t)x(t) + D_c(t)u(t) \end{cases} \tag{1.9}$$

with $x \in \mathbb{R}^{n_x}$, $y \in \mathbb{R}^{n_y}$, $u \in \mathbb{R}^{n_u}$, $A_c \in \mathbb{R}^{n_x \times n_x}$, $B_c \in \mathbb{R}^{n_x \times n_u}$, $C_c \in \mathbb{R}^{n_y \times n_x}$ and $D_c \in \mathbb{R}^{n_y \times n_u}$.

And a simpler case is the one with Time invariant parameters: **Linear Time Invariant (LTI) State-Space Models**

$$\text{Linear:}\quad \begin{cases} \dot{x}(t) = A_c x(t) + B_c u(t) \\ y(t) = C_c x(t) + D_c u(t). \end{cases} \tag{1.10}$$

Some systems may arise directly in a discrete form, or the continuous ones can be discretized to obtain the following discrete equations:
Discrete Linear State-Space Model

$$\begin{cases} x(k+1) & = A(k)x(k) + B(k)u(k) \\ y(k) & = C(k)x(k) + D(k)u(k). \end{cases} \tag{1.11}$$

The simpler case that we are interested in is when the matrices A, B, C, D do not depend on the time variable k, hence we have:
Discrete Linear Time-Invariant State-Space (DLTI) Models

$$\begin{cases} x(k+1) & = Ax(k) + Bu(k) \\ y(k) & = Cx(k) + Du(k). \end{cases} \tag{1.12}$$

We note that the state representation of a system is not unique, there are more states and matrices which give the same input-output relation. Given the state of system (1.12) we can obtain another state $x_T(k)$ for each nonsingular matrix T, and the relative system matrices (A_T, B_T, C_T, D_T) in the following way

$$x_T(k) = T^{-1}x(k) \quad \text{and} \quad A_T = T^{-1}AT, \quad B_T = T^-1B, \quad C_T = C^T, \quad D_T = D.$$

1.3.3 Discrete Linear Time-Invariant (DLTI) models

The importance and extensive use of DLTI models is due to the following aspects:

- DLTI models are a common structure for lots of physical phenomena, i.e., very different systems (from mechanical to electrical, thermal, . . .) can be described with the same mathematical structure of equations, with different meaning of the states and variables. This concept is called *system analogy*. The consequence is that, from the mathematical point of view, the theory and the dynamics of these systems can be studied independently of the application.

- The other characterization is that system theory is well developed, a lot of properties can be characterized and, in the deterministic case, the solution is known explicitly.

- In the linear case, linear algebra problems can be solved in real-time, with small computational effort.

- The most used controller systems are described by linear equations, hence nonlinear models of the system to be controlled would be worthless in such cases.

Properties: Controllability, Reachability, Observability

Controllability, Reachability and Observability are properties of a system which are important in Control Theory. When the model is used to control a certain system it is crucial to know how the input can influence the dynamic. We recall here the definitions [77].

Definition 5 (Controllability)**.** *The DLTI system* (1.12) *is controllable if, given any initial state $x(k_a)$, there exists an input signal $u(k)$ for $k_a \leq k \leq k_b$ such that $x(k_b) = 0$ for some k_b.*

Definition 6 (Reachability)**.** *The DLTI system* (1.12) *is* reachable *if for any two states x_a and x_b there exists an input signal $u(k)$ for $k_a \leq k \leq k_b$ that will transfer the system from the state $x(k_a) = x_a$ to $x(k_b) = x_b$.*

In few words, controllability means that the system state, through a certain input, can always be brought to the origin, and reachability that it can always be moved from one point to another.

Observability is the possibility to deduce univocally the state from the output measurement. For a precise definition of observability we need the concept of response of a system to a certain input, that can be calculated by recursion from the first equation of (1.12): the response from time instant k to time instant $k+j$ is given by

$$x(k+j) = A^k x(j) + \sum_{i=0}^{k-1} A^{k-i-1} B u(i+j).$$

The first part $A^k x(j)$ is called zero-input response, since is equivalent to the response of the system if the input is zero $u \equiv 0$, and the second part is called zero-state response. We can now introduce the definition of observability following [77]:

Definition 7 (Observability)**.** *The DLTI system* (1.12) *is* observable *if any initial state $x(k_a)$ is uniquely determined by the corresponding zero-input response $y(k)$ for $k_a \leq k \leq k_b$ with k_b finite.*

It is easy to check these properties on DLTI systems, since two theorems hold:

- reachability is equivalent to the matrix $C_n = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}$ to be full rank,

- observability is equivalent to the matrix $O_n = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$ to be full rank.

These properties are unified in the following one, that is a common assumption for a lot of theorems in system identification.

Definition 8 (Minimality). *The DLTI system* (1.12) *is minimal if it is both reachable and observable.*

Given a minimal DLTI system, the dimension of its state vector $x(k)$, is called the *order* of the DLTI system.

Explicit formulas of the Solutions

The solution equations for the State-Space problems, given the initial conditions on the state, have analytic expressions

- in the *continuous* deterministic LTI case (1.10) is:

$$\begin{cases} x(t) = \Phi_{t,t_0} x(t_0) + \int_{t_0}^{t} \Phi_{t,\alpha} B_c u_\alpha d\alpha \\ y(t) = C_c \Phi_{t,t_0} x(t_0) + \int_{t_0}^{t} C_c \Phi_{t,\alpha} B_c u_\alpha d\alpha + D_c u(t) \end{cases} \tag{1.13}$$

  with $\Phi_{t,t_0} = e^{A_c(t-t_0)}$.

- in the *discrete* deterministic linear time variant case (1.11) is:

$$\begin{cases} x(k) = \Phi(k,0)x(0) + \sum_{i=0}^{k-1} \Phi(k,i)B(i)u(i) \\ y(k) = C(k)\Phi(k,0)x(0) + \sum_{i=0}^{k-1} C(t)\Phi(k,i)B(i)u(i) + D(k)u(k) \end{cases} \tag{1.14}$$

  with $\Phi(k,i) = A(k-1)\,A(k-2)\,A(k-3)\cdots A(i)$ for $k > i$, for a general discrete linear time variant state-space; while in the time-invariant case, we have $\Phi(k,i) = A^{k-i}$ for $k > i$ for a DLTI state-space.

Chapter 2

Estimation Criteria: formulation of the identification problem

2.1 Linear Least Squares

Suppose we know that the relation between two set of variables called input and output of our system is given by a linear model $y = A(u)x$ as described in Section 1.2.1, with parameters x, and we have some measurements (u^{meas}, y^{meas}) of these quantities. We want to find the value of parameter vector $\hat{x}$ such that the model prediction $\hat{y} = A(u^{meas})\hat{x}$ is "close" to the measured data y^{meas}.

We will call for simplicity $A = A(u^{meas})$ and $b = y^{meas}$.

The two following facts holds:

- the problem has a solution if and only if $b \in Im(A)$ that is if the right-hand side lies in the image of A, i.e. its column space. In this case the system is said to be consistent;

- the solution is unique if and only if the kernel of the matrix is zero $Ker(A) = 0$, that is equivalent to asking that $A \in \mathbb{R}^{m \times n}$ have full column rank $rank(A) = n$.

Hence the problem has a unique solution only if both conditions are satisfied.

In the particular case in which A is a square invertible matrix (hence full rank), the solution can be written as $x = A^{-1}b$ and is unique.

What happens usually, is that the true data on the right hand side y, that belongs to the image of A, is not known, and only a noisy measurement vector $b = y^{meas} = y + e$ is available, where $e \in \mathbb{R}^n$ is the error vector, usually white noise.

In this case, an overdetermined linear system is built collecting a lot of measurements from the physical system, so that the model parameters are calculated exploiting the zero mean of the noise. Since the direct inversion of the system is not possible, another criterion for the determination of the parameter vector must be chosen.

The matrix A, can be assumed known with or without error, and this leads to different criteria for the calculation of the parameter vector. When it is supposed known without error, the most common problem formulation is the *least-squares problem*

$$\hat{x} = \underset{x}{\operatorname{argmin}} \, \|Ax - b\|_2^2. \tag{2.1}$$

We show an analytical formula for the solution of this problem (2.1). The cost function of the problem can be written as

$$f(x) = \|Ax - b\|_2^2 = (Ax - b)^T(Ax - b) = x^T A^T Ax - x^T A^T b - b^T Ax + b^T b$$

and its gradient

$$\frac{\partial f(x)}{\partial x} = 2A^T Ax - 2A^T b.$$

By imposing the gradient equal to zero we obtain the so called *normal equation*

$$A^T Ax = A^T b \tag{2.2}$$

which solutions are the same of the least-squares solutions of problem (2.1). Moreover, in the case in which A has full column rank, the unique solution is

$$\hat{x} = (A^T A)^{-1} A^T b.$$

If we rewrite the normal equation (2.2) as $A^T(Ax - b) = 0$ we can see that the residual $e = b - A\hat{x}$ is orthogonal to the range space of A.

2.1.1 Regularization

We recall that a linear problem is *well-posed* in the sense of Hadamard if it satisfies the following three conditions:

- Existence: The problem must have a solution.
- Uniqueness: There must be only one solution to the problem.
- Stability: The solution must depend continuously on the data

Problems which are not well-posed are called *ill-posed*. In particular when the stability property is not satisfied it means that a small error in the data determines a big error in the solution of the problem.

Regularization of ill-posed problems consists in all the methods to stabilize it and obtain new problems which are less sensitive to errors in the data.

One-Parameter regularization

Tikhonov regularization is the most common regularization method for linear ill-posed inverse problems. We do not treat other methods in this work, and we will focus on this since it will be used in more than one context later.

The Tikhonov method consists in the addition of a term to the cost function of the least-squares problem as follows

$$x_\alpha = \underset{x}{\operatorname{argmin}} \; \|Ax - b\|_2^2 + \alpha \, \|L\,x\|_2^2 \tag{2.3}$$

where α is called the *regularization parameter*, and L is a matrix that is usually equal to the identity or the discretization of a differentiation operator (commonly of the first or second order). The explicit formula for the solution is

$$x_\alpha = (\alpha L^T L + A^T A)^{-1} A^T b \tag{2.4}$$

where $A_\lambda^\# := (\alpha L^T L + A^T A)^{-1} A^T$ is called *Tikhonov regularized inverse*.

The choice of the regularization parameter is a crucial point in the solution of the new regularized problem (2.3). Various methods have been studied in literature and are in general divided in two opposing cathegories: *non-heuristic* methods, which assume the error magnitude (variance or norm) is known, and *heuristic* methods, or *noise level free rules*. An overview of various methods can be found in [31, 35, 64], and we give a brief summary in the following paragraphs.

Non-heuristic Non-heuristic rules are the ones that use information on the noise.One principle that is based on this knowledge is called *Discrepancy rule* and consists in choosing $\alpha = \alpha_{DP}$ such that

$$\|Ax_\alpha - b\|_2 = \nu_{dp}\|e\|_2,$$

where x_α is the solution of (2.3) with regularization parameter α, and $\nu_{dp} \geq 1$ is a "safety factor".

The discrepancy principle is often used because of its simplicity, but has the disadvantage that the information it requires, i.e. the norm of the error $\|e\|_2$, is not often available. An estimate can be used, but the solution is very sensitive to this quantity.

Heuristic Heuristic rules are the ones that do not use the norm of noise, also called *noise level-free rules*. The most common heuristic methods [31, 32, 64] are

- L-curve, which is a log-log plot of the solution norm $\|x_\alpha\|_2$ versus the residual norm $\|Ax_\alpha - b\|_2$ with α the parameter. The optimal parameter is chosen as the one on the corner of this curve, i.e. the point that maximizes the curvature.

- GCV (Generalized Cross Validation) which consists in choosing the parameter as the solution of

  $$\min_{\alpha} \frac{\|Ax_\alpha - b\|_2^2}{trace(I_m - AA^{\#})^2},$$

  where $A^{\#}$ is the *regularized inverse*, defined by $x_{reg} = A^{\#}b$ and depends on the chosen regularization method. In particular, in the case of Tikhonov method it is equal to the *Tikhonov regularized inverse* already defined $A^{\#} = A^{\#}_{\lambda} = (\alpha L^T L + A^T A)^{-1} A^T$.

  The idea at the origin of this method is to separate the measurements in two sets: from the first an estimate of the solution is computed, that is used to simulate the model on the data of the second set, on which the error is minimized to obtain the best regularization parameter α. In the simpler case, the second set consists of a single sample of the data: one row of the matrix A, and the corresponding component of the right-hand side vector b are removed from the original system to obtain a new system from which an estimate of the parameter vector is computed, call it

  $$x^{(i)} = \underset{x}{\operatorname{argmin}} \, \|A^{(i)}x - b^{(i)}\|_2^2$$

  where $A^{(i)}$ is the matrix A without the row i and $b^{(i)}$ is the vector b without component i. Hence, a good estimate for the regularization parameter can be obtain as

  $$\min_{\alpha} \frac{1}{m} \sum_{i=1}^{m} (A(i,:)x_\alpha^{(i)} - b_i)^2$$

  where $A(i,:)$ is the i-th row of the matrix A and b_i the i-th component of vector b, but in this way m different linear systems must be solved which determine a big computational cost. For this reason the minimization is substituted with

  $$\min_{\alpha} \frac{1}{m} \sum_{i=1}^{m} \left(\frac{A(i,:)x_\alpha - b_i}{1 - h_{ii}} \right)^2$$

  so that only one solution of the linear problem have to be computed, where h_{ii} are the diagonal elements of the matrix $AA^{\#}$. To get rid of the problem of ordering, the value h_{ii} is replaced with the average of the diagonal elements, to obtain the first given formula.

- NCP (Normalized Cumulative Periodogram). The idea behind this method is that the residual of the estimate of the least squares problem , i.e. $r = Ax_\alpha - b$ should be comparable to white noise. The NCP of a signal r is defined as the vector

  $$NCP(r)_i := \frac{(p_r)_2 + (p_r)_3 + \cdots + (p_r)_{i+1}}{(p_r)_2 + (p_r)_3 + \cdots + (p_r)_{q+1}} \qquad \text{for } i = 1, \ldots, q = \lfloor N/2 \rfloor$$

where

$$p_r = [|(f_r)_1|^2, |(f_r)_2|^2, \dots, |(f_r)_N|^2]^T$$

is the power spectrum density and $f_r = dft(r) = [(f_r)_1, (f_r)_2, \dots, (f_r)_N]^T \in \mathbb{C}^N$ is the discrete Fourier transform of r. The NPC gives a way to check if the residual is comparable to white noise, in fact if the vector r consists of white noise, by definition, the expected power spectrum is flat, i.e. $E((p_r)_2) = E((p_r)_3) = \dots = E((p_r)_{q+1})$, and the points on the expected NCP curve, $E(NPC(r)_i)$ lie on a straight line from the origin to the point $(q, 1)$. Hence, the method consists in looking for the regularizing parameter α that minimizes the distance of the NPC of the solution from this straight line.

Other rules are based on the residual $\|A\hat{x} - b\|_s$, and consist in maximizing a certain function. We list them below, following [63]. First call α the parameter to be found and $B_\alpha := \alpha^{-1}(\alpha I + AA^T)^{-1/2}$. Then we have:

Quasi-optimality rule	$\psi_Q(\alpha) = \alpha \left\| \frac{\partial x_\alpha}{\partial \alpha} \right\|_2 = \alpha^{-1} \left\| A^T B_\alpha^2 (Ax_\alpha - b) \right\|_2$
Hanke-Raus rule	$\psi_{HR}(\alpha) = \alpha^{-1/2} \left\| B_\alpha (Ax_\alpha - b) \right\|_2$
Reginska rule	$\psi_{RE}(\alpha) = \|Ax_\alpha - b\|_2 \ \|x_\alpha\|_2^\tau, \ \ \tau \geq 1$
Heuristic Monotone Error rule	$\psi_{HME}(\alpha) = \alpha^{-1/2} \frac{\|B_\alpha(Ax_\alpha - b)\|_2^2}{\|B_\alpha^2(Ax_\alpha - b)\|_2}$

Heuristic methods possess bad convergence properties, in fact they don't converge in the "worst case scenario", i.e. it is not true that the regularized solution converges to the least squares solution for every noise realization with noise level that tends to zero. Although this result, called *Bakushinskii veto* [3], convergence results have been demonstrated under appropriate conditions, that are usually satisfied in real situations (see [45] and references therein).

Multi-Parameter regularization

In some cases more than one regularization term is added to the cost function to obtain a solution with particular requested properties. We can consider the general case

$$x_\alpha = \underset{x}{\operatorname{argmin}} \ \|Ax - b\|_2^2 + \sum_{i=1}^{n_p} \alpha_i \, \|L_i \, x\|_2^2 \tag{2.5}$$

with n_p regularization terms and denote $\alpha = (\alpha_1, \dots, \alpha_{n_p})$ the vector of regularization parameters. This problem is called *multi-parameter regularization*.

Also in this case the methods for the choice of the parameters are divided in *heuristic* and *non-heuristic*.

Non-heuristic If the noise covariance is known (at least approximately), the parameters can be determined with the discrepancy principle, and in the multi-parameter case with its generalization, introducing the *discrepancy hypersurface* $\mathcal{D}$ ([24], [54]) defined as

$$\mathcal{D} = \{\alpha \in \mathbb{R}^{n_p} | \ \alpha \geq 0, \ \alpha \not\equiv 0, \ \|Ax_\alpha - b\|_2 = \nu_{dp} \|e\|_2\},$$

that is the generalization of the discrepancy principle.

The optimal parameters are found on that hypersurface through the optimization of a function of the parameters, for example the norm of the solution can be maximized [24] or the quasi-optimality criterion (introduced in [73]) can be considered [21].

Heuristic Among heuristic methods for multi-parameter regularization, we can find the generalizations of the L-curve, i.e. the L-hypersurface [7], of the Generalized Cross Validation (GCV) [10], a balancing principle [39] and parameter learning for denoising [47],[37].

2.2 Total Least Squares

Total Least Squares (TLS), first introduced in [25, 26], is a generalization of the Least Squares problem for the case in which both data A and b of the linear system $Ax = b$ are perturbed with noise. The *Total Least Squares (TLS)* problem seeks to

$$\min_{[\hat{A},\hat{b}]\in\mathbb{R}^{m\times(n+1)}} \|[A,b]-[\hat{A},\hat{b}]\|_F \quad \text{subject to } \hat{b}\in Im(\hat{A}) \tag{2.6}$$

Once a minimizing $[\hat{A},\hat{b}]$ is found, then any x satisfying $\hat{A}x = \hat{b}$ is called TLS *solution*. Note that the two errors on A and b are supposed of the same magnitude when the TLS criteria is used.

The similarity with the least-squares problem is highlighted if we rewrite the LS problem (2.1) in the following equivalent way

$$\min_{\hat{b}\in\mathbb{R}^m} \|\hat{b}-b\|_2 \quad \text{subject to } \hat{b}\in R(A).$$

Once a minimizing $\hat{b}$ is found, then any x satisfying $Ax = \hat{b}$ is called LS *solution*.

It has been studied in various fields (such as statistics, system identification, signal processing, numerical analysis) and is referred to with different names, for example "orthogonal regression" or "error-in-variables regression" in statistics. As for the ordinary linear least-squares problem, the fields of application of TLS are the most various and range over all sciences.

A fundamental tool has been the singular value decomposition (SVD), which allows to compute the solution in an easy way.

Example 5 (One dimensional case example)**.** *We describe the geometric interpretation of TLS on the easy example of linear fitting: given the true model $ax = b$ with $a, b \in \mathbb{R}$, we suppose to know some couples of noisy measures collected on the two vectors $a^{meas}, b^{meas} \in \mathbb{R}^m$. We are looking for the value of the parameter x of the model. In Figure 2.1 the two solutions with LS and TLS are shown.*

The LS method minimizes the sum of the squared vertical distances from the data points to the fitting line, i.e. finds the x that minimizes $\|a^{meas}x - b^{meas}\|_2$.

The TLS method minimizes the sum of the squared orthogonal distances from the data points to the fitting line, i.e. finds $[a'; b']$ that minimizes

$$\|[a'; b'] - [a^{meas}; b^{meas}]\|_2$$

and such that $a'x = b'$.

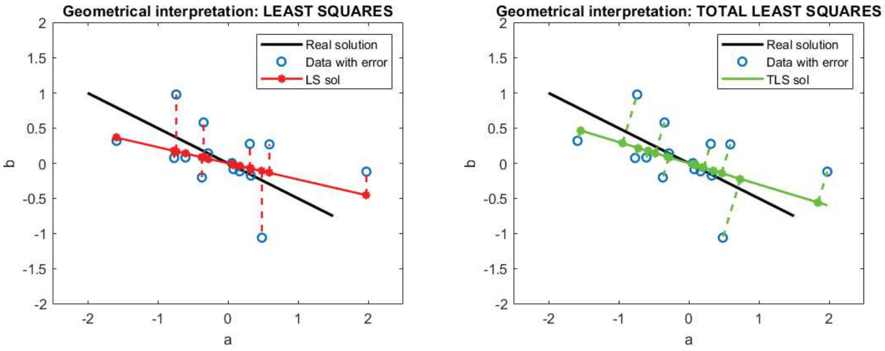


Figure 2.1: Comparison between the solution of the LS and TLS problem for a one dimensional example

The closed-form expression of the basic TLS solution can be calculated in the following way. Call $\sigma_1 \geq \cdots \geq \sigma_n \geq \sigma_{n+1}$ the singular values of the augmented matrix $\begin{bmatrix} A & b \end{bmatrix}$, and $\sigma_1' \geq \cdots \geq \sigma_n'$ the singular values of A. If $\sigma_n' > \sigma_{n+1}$ (condition for the existence of the TLS solution), then

$$x_{TLS} = (A^T A - \sigma_{n+1}^2 I)^{-1} A^T b.$$

Observation 2. *This formulation is similar to the Tikhonov regularized solution* (2.4) *for the LS problem, but with negative α and for this reason the new matrix $(A^T A - \sigma_{n+1}^2 I)$ has a larger condition number than $A^T A$. So for Ill-Posed Problems this method does not lead to a better formulation but to a worse one, hence regularization of the problem is needed.*

Regularization for Ill-Posed Problems: The same regularization methods introduced in the linear least-squares problem can be used in this case.

2.3 Nonlinear Least Squares

A *criterion* to find the solution of nonlinear problems of the kind (1.3) is the generalization of the linear case, called *nonlinear least squares*. It consists in minimizing the norm of the error between the measured vector y^{meas} and the vector y estimated through the model. For this reason it is also called the *prediction error method* (PEM), and consists in the following problem

$$\begin{aligned} \underset{x}{min}\ F(x) = \frac{1}{2}\|f(u,x) - y^{meas}\|_2^2 \quad &= \quad \underset{x}{min}\ \frac{1}{2}\|y - y^{meas}\|_2^2 = \\ &\quad\ \text{s.t.} \quad y = f(u,x) \end{aligned}$$

$$\begin{aligned} =&\underset{x}{min}\ \frac{1}{2}\sum_{i=0}^{N}(y_i - y_i^{meas})^2. \\ \text{s.t.}\quad & y_i = f(u,x_i) \quad \forall i \end{aligned}$$

The residual vector is defined as

$$r(x) = \begin{bmatrix} f_0(u,x) - y_0^{meas} \\ \vdots \\ f_i(u,x) - y_i^{meas} \\ \vdots \\ f_N(u,x) - y_N^{meas} \end{bmatrix} \tag{2.7}$$

so that the function $F(x) = \frac{1}{2}r^T(x)r(x)$ and the matrix

$$J(x) = \frac{\partial r}{\partial x}(x) \tag{2.8}$$

is called the *sensitivity matrix*. The gradient of the cost function is then

$$\nabla F(x) = \frac{\partial F}{\partial x}(x) = J^T(x)r(x) \tag{2.9}$$

and the Hessian matrix

$$H(x) = \frac{\partial^2 F}{\partial x^2}(x) = \frac{\partial J^T r}{\partial x}(x) = J^T(x)\frac{\partial r}{\partial x}(x) + \frac{\partial J^T}{\partial x}(x)r(x) = J^T(x)J(x) + \frac{\partial J^T}{\partial x}(x)r(x) \tag{2.10}$$

is the sum of two terms.

More in general we can write it in the form

$$\begin{aligned} &\underset{x}{min}\ \ F(x) = \frac{1}{2}\|y - y^{meas}\|_2^2 \\ &\text{s.t.} \qquad \mathcal{M}(y,u,x) = 0 \end{aligned} \tag{2.11}$$

where $\mathcal{M}(y,u,x) = 0$ is the nonlinear model that gives the relationship between the variables y and u and the parameter vector x.

Chapter 3

Numerical methods and algorithms

3.1 Linear Least Squares

The solution of the linear system is given analytically by the normal equations (2.2), however, this formula can not be used for the numerical computation of the solution because of its instability [8, 50, 32]. The most common methods for the batch solution of the Linear Least-Squares problem 2.1 are the QR factorization $A = QR$ and the SVD decomposition $A = U\Sigma V^T$.

We just introduce the two factorization here and refer to literature for a more detailed description.

QR factorization We recall that the QR factorization of an $m - by - n$ matrix A is given by $A = QR$ where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R \in \mathbb{R}^{m\times n}$ is upper triangular. While the decomposition $A = Q(:, 1 : n)R(1 : n, 1 : n)$ is referred to as the thin QR factorization.

The computation of this factorization can be done with different methods like Householder, Givens transformations and Gram-Schmidt orthogonalization process.

Assume that $rank(A) = n$, and hence the solution to the least-squares problem is unique and given the QR factorization, the normal equations (2.2) simplifies to become

$$Rx = Q^T b$$

that is an upper-triangular system that can be efficiently solved by back substitution. The solution of this linear system is either the solution of $Ax = b$ or the least-squares solution depending on wether or not $Ax = b$ is consistent [60].

For a recursive estimation of the solution, the QR factorization can be efficiently updated ([27] Sec. 12.5.3) every time a new measurement is available, moreover a forgetting factor that weights exponentially the rows of the linear system can be used to give more importance to the last ones.

SVD decomposition Let us now recall the definition of the SVD decomposition.

Theorem 1 (Singular Value Decomposition (Theorem 2.5.2 of [27]). *If A is a real $m \times n$ matrix, then there exist orthogonal matrices*

$$U = [u_1, \dots, u_m] \in \mathbb{R}^{m \times m} \quad \text{and } V = [v_1, \dots, v_n] \in \mathbb{R}^{n \times n}$$

such that

$$U^T A V = \operatorname{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m \times n} \quad p = \min(m, n)$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$.

The σ_i are called singular values, and the columns of the matrices U and V are the left and right singular vectors respectively.

The SVD decomposition is fundamental for the calculation of the least-squares solution of problem (2.1). Given the SVD decomposition $A = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^T$, the Moore-Penrose inverse is

$$A^\dagger = V \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^T = \sum_{i=1}^{r} \frac{v_i u_i^T}{\sigma_i}$$

where r is the rank of A and $\Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_r)$. Moreover,

$$\hat{x} = A^\dagger b = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i$$

is the least-squares solution of (2.1) of minimum 2-norm.

3.2 Total Least Squares

The solution of the TLS problem 2.6 can be calculated rewriting the problem from $A\,x = b$ to

$$\begin{bmatrix} A & b \end{bmatrix} \begin{bmatrix} x \\ -1 \end{bmatrix} = 0$$

and through the SVD decomposition of the augmented matrix $\begin{bmatrix} A & b \end{bmatrix}$. The solution is given by Algorithm 1.

Algorithm 1 TLS: Basic solution

```
1: Calculate the SVD
       $$\begin{bmatrix} A & b \end{bmatrix} = U\Sigma V^T = U\,\mathrm{diag}(\sigma_1, \dots, \sigma_n, \sigma_{n+1}) \begin{bmatrix} v_1, \dots, v_{n+1} \end{bmatrix}^T$$
2: If $\sigma_{n+1} \neq 0$ there is no exact solution and hence we must consider the **rank $n$ approximation** of the matrix $\begin{bmatrix} A & b \end{bmatrix}$, i.e. the one obtained imposing
       $$\sigma_{n+1} = 0 \quad \text{(Eckart-Young-Mirsky Theorem)}$$
       $$\begin{bmatrix} \hat{A} & \hat{b} \end{bmatrix} = U\hat{\Sigma} V^T = U\,\mathrm{diag}(\sigma_1, \dots, \sigma_n, 0) \begin{bmatrix} v_1, \dots, v_{n+1} \end{bmatrix}^T$$
3: The solution of the new problem $\begin{bmatrix} \hat{A} & \hat{b} \end{bmatrix} \hat{x} = 0$ is given by the vector $\hat{x} = v_{n+1}$. The TLS solution is obtained by scaling this vector to obtain $-1$ as last entry
       $$x_{TLS} = \frac{-1}{v_{n+1,n+1}} v_{n+1}$$
   where $v_{n+1,n+1}$ is the last entry of vector $v_{n+1}$.
4: If $\sigma_{n+1} = 0$ and $\sigma_n > 0$ then the solution above is the exact solution.
```

3.3 Nonlinear Least Squares: Local and Global optimization methods

The optimization methods that can be used to solve the nonlinear least-squares problem are various, and can be divided in local and global optimization methods. Among the local methods, which finds a local minima of the cost function, the most common are Gauss-Newton and Levenberg-Marquardt methods.

For completeness we write here the formulation of the iterative Gauss-Newton method for the solution of (2.11): with the notation already introduced the scheme is the following

$$\begin{cases} J^T(x^i)J(x^i)\delta x^i = -J^T(x^i)r \\ x^{i+1} = x^i + \delta x^i \end{cases} \tag{3.1}$$

and is obtained by neglecting the second term of the Hessian formula (2.10), i.e with the approximation

$$H(x) \approx J^T(x)J(x). \tag{3.2}$$

Note that the resolution of a nonlinear optimization problem is iterative and computationally more difficult than the solution of the linear case. Moreover, the problem of local minima may require the use of global optimization algorithms.

We will use the brute-force optimization in Chapter 5, that consists in the computation of the cost function $F = \frac{1}{2}\|r\|_2^2$ of the minimization (2.11) on a predefined grid of points.

3.3.1 Computation of the gradient

In this section we show different ways to compute numerically the gradient $\nabla F(x) = \frac{\partial F}{\partial x_i}(x) = J(x)^T r(x)$, following mainly [42]. This is a bottleneck in the resolution of the nonlinear inverse problem (2.11), especially when the number of parameters to be calculated is high.

Finite Differences

The simpler method for the computation of the gradient is to approximate it by finite differences. This method is not recommended and not used in real applications but gives a way to validate the code implementation of other methods. The reasons are that its computational cost is high, proportional to the number of parameters of the problem, and it gives approximated results with low precision.

The gradient calculated with finite differences has the following form

$$\frac{\partial F}{\partial x_i} \approx \frac{F(x+h_i) - F(x)}{h_i}, \quad i = 1, \dots, n. \tag{3.3}$$

where x is the parameter vector with components x_i and h_i is the discrete step for the $i-$th component.

Sensitivity/Variational equations

Given the continuous ODE

$$\dot{y}(t) = f(t, p, y(p)) \tag{3.4}$$

with p parameters, assumed constants, and f the continuous nonlinear function (or "field"), if y is twice differentiable and applying the Schwarz theorem we have

$$\frac{d}{dt}\frac{\partial y}{\partial p}(t) = \frac{\partial \dot{y}}{\partial p}(t) = \frac{\partial f}{\partial p}(t, p, y(p))$$

and the following *sensitivity or variational differential equation* holds

$$\frac{\partial \dot{y}}{\partial p}(t) = \frac{\partial f}{\partial y}(t, p, y(p))\frac{\partial y}{\partial p}(t) + \frac{\partial f}{\partial p}(t, p, y(p)) \quad \text{with} \quad \frac{dy}{dp}\Big|_{t=0} = 0.$$

In the case in which the initial condition is not known we can impose $p = y(0)$, and the previous equation becomes

$$\frac{\partial \dot{y}}{\partial y(0)}(t) = \frac{\partial f}{\partial y}(t, p, y(p))\frac{\partial y}{\partial y(0)}(t) \qquad \text{with} \quad \frac{\partial y}{\partial y(0)}\Big|_{t=0} = I.$$

More compactly the two previous equations can be written as

$$dt[D_{c,p}y(c,p;t)] = D_{c,p}f \cdot D_{y,p}\begin{bmatrix} y \\ p \end{bmatrix} \tag{3.5}$$

where $c = y(0)$ is the initial condition, and p are the parameters to be estimated. The previous equation can be rewritten in a more explicit form as

$$\left[\frac{\partial \dot{y}}{\partial y_0}, \frac{\partial \dot{y}}{\partial p_0}, \dots, \frac{\partial \dot{y}}{\partial p_N}\right] = \left[\frac{\partial f}{\partial y_0}, \frac{\partial f}{\partial p_0}, \dots, \frac{\partial f}{\partial p_N}\right] \cdot \begin{bmatrix} \frac{\partial y}{\partial y_0} & \frac{\partial y}{\partial p_0} & \cdots & \frac{\partial y}{\partial p_N} \\ 0 & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Solving this differential equation through a discretization, we obtain $\frac{\partial y}{\partial y(0)}(k)$, $\frac{\partial y}{\partial \hat{p}(i)}(k)$ for all k, for all i. Then, $\hat{y} \approx y$ neglecting the approximation error, as already said at the beginning.

At each iteration of the Gauss-Newton minimization (3.1) on

$$p_i = [y(0), p(0), \dots, p(N-1)]$$

we need to calculate the Jacobian $J = \frac{\partial F}{\partial p}(p_i)$. To do so we must solve the variational equations calculated in p_i.

The calculation of the gradient of the cost function $F = \frac{1}{2}\|r\|_2^2$ with the sensitivity equations has a cost proportional to the number of parameters to be estimated.

Adjoint methods

For a theoretical and accurate description of adjoint methods we refer to [16], and to [42] for a more applied approach. We will follow here the formulation of [48] for the discrete case since it will be applied later.

We describe the discrete adjoint method for the computation of the solution of the discretized problem

$$\min_p \; F(x,y,p) = \frac{1}{2}\sum_{i=0}^{N-1} \nu_i (y(x_i) - \bar{y}(t_i))^2 \tag{3.6}$$

$$s.t. \;\; f_{i+1} = f(x_{i+1}, x_i, t_{i+1}, t_i, p) = 0 \quad \text{for } i = 0, \dots, N-1 \tag{3.7}$$

$$x_0 \text{ given} \tag{3.8}$$

where

- $p \in \mathbb{R}^m$ is the vector of parameters to identify,
- $t_0, \dots, t_N \in \mathbb{R}$ the given times of the discretization,

- $x \in \mathbb{R}^n$ the state vector that satisfy the differential equation

$$\dot{x}(t) = f(t, x, p)$$

  and x_i the state calculated at the discretization times that satisfy the discretized equations

$$f(x_{i+1}, x_i, t_{i+1}, t_i, p) = 0 \quad \text{for } i = 0, \ldots, N-1$$

- ν_i weighting factors,
- $y(x_i)$ the system output vector calculated in the state state x_i and $\bar{y}(t_i)$ the measurement vector at the discretization times t_i.

We extend the cost function with some zero terms to obtain the Lagrangian function $\mathcal{L}$ that has the following form

$$\mathcal{L} = \sum_{i=0}^{N-1} \left\{ \frac{1}{2} \nu_i (y(x_i) - \bar{y}_i)^2 + \lambda_{i+1}^T f(x_{i+1}, x_i, t_{i+1}, t_i, p) \right\} \tag{3.9}$$

where λ_i are the *adjoint variables*. Since we are imposing the state equation to hold as constraints of problem (3.9), the additive terms are zero for any choice of the variables λ_i, i.e. it holds

$$F(x, y, p) = \mathcal{L}(x, y, p, \lambda_i) \quad \forall \lambda_i$$

and so we can differentiate the relationship obtaining

$$\frac{\partial F}{\partial p} \delta p = \frac{\partial \mathcal{L}}{\partial p} \delta p + \frac{\partial \mathcal{L}}{\partial y} \frac{\partial y}{\partial p} \delta p. \tag{3.10}$$

The aim is the computation of the gradient of the cost function w.r.t the parameters to be identified. The bottleneck of this formula is the computation of $\frac{\partial y}{\partial p}$, since it requires the solution of the sensitivity equations, which are proportional to the number of parameters, as seen above. Since the equation (3.10) holds for any choice of the adjoint variables, it is possible to choose them such that the second term of (3.10) is zero and it becomes

$$\frac{\partial F}{\partial p} \delta p = \frac{\partial \mathcal{L}}{\partial p} \delta p \tag{3.11}$$

that gives a simple expression for the gradient of the cost function F.

The variation of the cost function, equal to the variation of the Lagrangian, is the following

$$\delta F = \delta \mathcal{L} = \sum_{i=0}^{N-1} \nu_i (y_i - \bar{y}(t_i)) \frac{\partial y_i}{\partial x_i} \delta x_i + \lambda_{i+1}^T \left(\frac{\partial f_{i+1}}{\partial x_{i+1}} \delta x_{i+1} + \frac{\partial f_{i+1}}{\partial x_i} \delta x_i + \frac{\partial f_{i+1}}{\partial p} \delta p \right) \tag{3.12}$$

and reformulating it in terms of δx_i and δp we obtain

$$
\begin{aligned}
\delta\mathcal{L} = &\left(\lambda_1^T \frac{\partial f_1}{\partial x_0} + \nu_0 (y_0 - \bar{y}(t_0))^T \frac{\partial y_0}{\partial x_0}\right)\delta x_0 + \lambda_1^T \frac{\partial f_1}{\partial p}\delta p \\
&+ \sum_{i=1}^{N-1}\left[\left(\nu_i (y_i - \bar{y}(t_i))^T \frac{\partial y_i}{\partial x_i} + \lambda_i^T \frac{\partial f_i}{\partial x_i} + \lambda_{i+1}^T \frac{f_{i+1}}{\partial x_i}\right)\delta x_i + \lambda_{i+1}^T \frac{\partial f_{i+1}}{\partial p}\delta p\right] \\
&+ \lambda_N^T \frac{\partial f_N}{\partial x_N}\delta x_N
\end{aligned}
\tag{3.13}
$$

Adjoint variables and the variables to estimate satisfy the following optimality conditions at the optimal point:

$$
\begin{cases}
\frac{\partial \mathcal{L}}{\partial \lambda_i} = 0 & \text{for } i = 0, \dots, N-1 \\
\frac{\partial \mathcal{L}}{\partial p} = 0 & \text{for } i = 0, \dots, N-1 \\
\frac{\partial \mathcal{L}}{\partial x_i} = 0 & \text{for } i = 0, \dots, N-1
\end{cases}
\tag{3.14}
$$

where the first equation is the system equation, given by the constraints of (3.9). The second term is the sensitivity of the Lagrangian function w.r.t the parameters and equals to the sensitivity of the cost function w.r.t. the parameters that correspond to the minimization of the cost function.

Finally, the third equation of (3.14) is the adjoint equation, from which the adjoint variables are computed.

From this condition, we can impose equal to zero the terms in the brackets relative to the δx_i. Note that, since the value of x_0 is given, the term δx_0 is already equal to zero. We obtain the so called *adjoint equations*:

$$
\begin{cases}
\left(\frac{\partial f_N}{\partial x_N}\right)^T \lambda_N = 0, \\
\left(\frac{\partial f_i}{\partial x_i}\right)^T \lambda_i = -\nu_i (\frac{\partial y_i}{\partial x_i})^T (y_i - \bar{y}(t_i)) - \left(\frac{\partial f_{i+1}}{\partial x_i}\right)^T \lambda_{i+1}.
\end{cases}
\tag{3.15}
$$

In the matrix notation we see this is a bidiagonal linear system

$$
\begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & 0 & \dots & 0 \\
0 & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_3}{\partial x_2} & \dots & 0 \\
\vdots & & \ddots & \ddots & \\
0 & & & \frac{\partial f_{N-1}}{\partial x_{N-1}} & \frac{\partial f_N}{\partial x_{N-1}} \\
0 & & & 0 & \frac{\partial f_N}{\partial x_N}
\end{bmatrix}
\begin{bmatrix}
\lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{N-1} \\ \lambda_N
\end{bmatrix}
=
\begin{bmatrix}
-\nu_1 (\frac{\partial y_1}{\partial x_1})^T (y_1 - \bar{y}(t_1)) \\
-\nu_2 (\frac{\partial y_2}{\partial x_2})^T (y_2 - \bar{y}(t_2)) \\
\vdots \\
-\nu_{N-1} (\frac{\partial y_{N-1}}{\partial x_{N-1}})^T (y_{N-1} - \bar{y}(t_{N-1})) \\
0
\end{bmatrix}.
\tag{3.16}
$$

With this condition the equation of the variation of the Lagrangian becomes simpler

$$
\delta\mathcal{L} = \sum_{i=0}^{N-1}\left(\lambda_{i+1}^T \frac{\partial f_{i+1}}{\partial p}\right)\delta p = \left(\frac{\delta F}{\delta p}\right)^T \delta p
\tag{3.17}
$$

and the sensitivity of the cost function w.r.t the parameters is

$$\frac{\delta F}{\delta p} = \sum_{i=0}^{N-1} \left(\frac{\partial f_{i+1}}{\partial p} \right)^T \lambda_{i+1}. \tag{3.18}$$

The adjoint-state algorithm is summarized in Algorithm 2.

Algorithm 2 ”Solution of the nonlinear problem with the adjoint-state method”

```
Initialization: initial value of the parameters $p$
for $k = 1, \dots, K_{maxiter}$ do
    0) given the parameters $p^k$ of the k-th iteration of the optimization
    1) solve the discretized dynamical system to obtain the values of $\hat{y}$
    2) solve the system (3.15) arising from the third equation of (3.14), and obtain the adjoint variables $\lambda_i$
    3) use the values of $\hat{y}$ and the adjoint variables $\lambda_i$ to compute the gradient with (3.18)
    4) calculate the new parameters for the next iteration $p^{k+1}$ of the optimization
end for
```

Comparison between the calculation of the gradient with the sensitivity equation and the adjoint method Following [42] we summarize some considerations on the two methods just seen.

1. The additional cost for the computation of the gradient with the sensitivity approach is proportional to the number of the parameters, while in the adjoint method it does not depend on it. Hence, for problems with a high number of parameters the adjoint method is recommended.

2. The sensitivity approach results in the calculation of the Jacobian matrix of the residual r, i.e. $J = \frac{\partial r}{\partial p}$, from which it is possible to calculate the gradient of the cost function as $\frac{\partial F}{\partial p} = J^T r$; while the adjoint method yield only the gradient $\frac{\partial F}{\partial p}$.

3.3.2 Multiple Shooting method for chaotic dynamical systems

First we introduce the classical Multiple shooting method [2, 1], which is a method for the resolution of Boundary Value Problems (BVP) in the linear and nonlinear cases. Then, we will show how this method can be used as a parameter and initial condition identification method for chaotic dynamical systems [30].

Multiple Shooting for the solution of BVP linear problems

Consider the general linear Boundary Value Problem (BVP)

$$y'(t) = A(t)\, y + q(t), \quad a < t < b \tag{3.19a}$$

$$B_a\, y(a) + B_b\, y(b) = \beta \tag{3.19b}$$

with $y \in \mathbb{R}^n$, $t \in \mathbb{R}$ the independent variable, $A(t) \in \mathbb{R}^{n\times n}$, a known term $q(t) \in \mathbb{R}^n$, $a, b \in \mathbb{R}$ the endpoints of the interval on which the differential equation is defined, and the boundary condition with $B_a, B_b \in \mathbb{R}^{n\times n}$, $\beta \in \mathbb{R}$.

We divide the interval $[a, b]$ in subintervals $[t_i, t_{i+1}]$, with $1 \leq i \leq N$ on which the general solution of (3.19a) can be written as

$$y(t) = Y_i(t)\, s_i + p_i(t), \quad t_i \leq t \leq t_{i+1} \tag{3.20}$$

where $Y_i \in \mathbb{R}^{n\times n}$ is the fundamental solution, s_i is a parameter vector in $\mathbb{R}^n$ and p_i is a particular solution in $\mathbb{R}^n$. For each subinterval the fundamental and particular solutions are described by the following IVPs

$$\begin{cases} Y_i' = A(t)\, Y_i, & t_i \leq t \leq t_{i+1} \\ Y_i(t_i) = F_i \end{cases} \qquad \begin{cases} p_i' = A(t)\, p_i + q(t), & t_i \leq t \leq t_{i+1} \\ p_i(t_i) = \alpha_i \end{cases} \tag{3.21}$$

and in this case we will consider $F_i = I$, $\alpha_i = 0$ for all i, and q is the term independent from y of equation (3.19a). More details can be found in [2].

The vectors s_i in (3.20), called *shooting points*, are chosen to satisfy the *continuity conditions* that guarantee the continuity of the approximate solution at the mesh points:

$$Y_i(t_{i+1})\, s_i + p_i(t_{i+1}) = Y_{i+1}(t_{i+1})\, s_{i+1} + p_{i+1}(t_{i+1}), \quad i = 1, 2, \ldots, N-1. \tag{3.22}$$

Substituting the initial conditions

$$-Y_i(t_{i+1}) s_i + F_{i+1}\, s_{i+1} = p_i(t_{i+1}) - \alpha_{i+1}, \quad 1 \leq i \leq N-1 \tag{3.23}$$

and combining these equations (3.23) with the boundary conditions (3.19b) we obtain the square linear system

$$\begin{pmatrix} -Y_1(t_2) & F_2 & & & \\ & -Y_2(t_3) & F_2 & & \\ & & \ddots & \ddots & \\ & & & -Y_{N-1}(t_N) & F_N \\ B_a\, F_1 & & & & B_b\, Y_N(b) \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{pmatrix} = \begin{pmatrix} p_1(t_2) - \alpha_2 \\ p_2(t_3) - \alpha_3 \\ \vdots \\ p_{N-1}(t_N) - \alpha_N \\ \beta - B_a\, \alpha_1 - B_b\, p_N(b) \end{pmatrix} \tag{3.24}$$

and with the assumptions on F_i, α_i

$$\begin{pmatrix} -Y_1(t_2) & I & & & \\ & -Y_2(t_3) & I & & \\ & & \ddots & \ddots & \\ & & & -Y_{N-1}(t_N) & I \\ B_a & & & & B_b\, Y_N(b) \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{pmatrix} = \begin{pmatrix} p_1(t_2) \\ p_2(t_3) \\ \vdots \\ p_{N-1}(t_N) \\ \beta - B_b\, p_N(b) \end{pmatrix}. \tag{3.25}$$

Note that this matrix is in $\mathbb{R}^{n\,N \times n\,N}$.

The shape of this system is called Bordered Almost Block Diagonal (BABD). It reduces to an ABD system if the initial and final conditions are separated [1].

	Multiple Shooting for BVP problems in short:
1)	solve the ODEs (3.21) for the general and particular solutions in each subinterval to obtain the diagonal values of the matrix in (3.25)
2)	solve the linear system (3.25) to obtain the shooting nodes s_i

Table 3.1: Multiple Shooting for BVP problems in short

Multiple Shooting for the solution of BVP Nonlinear problems

We generalize the previous section to the case of a nonlinear BVP problem

$$y'(t) = f(t, y), \quad a < t < b \tag{3.26a}$$
$$g(y(a), y(b)) = 0. \tag{3.26b}$$

Consider the subdivision of the interval $[a, b]$ as before and the IVP problems in each of the subintervals given by

$$\begin{cases} y_i' = f(t, y), & t_i < t < t_{i+1} \\ y_i(t_i) = s_i. \end{cases} \tag{3.27}$$

Now the unknowns are the shooting nodes s_i, that must be chosen to satisfy the continuity conditions

$$y_i(t_{i+1}) = s_{i+1} \tag{3.28}$$

and the boundary conditions

$$g(s_1, y_N(b, s_N)) = 0. \tag{3.29}$$

We define the residual function

$$r_{ms}(s) = \begin{pmatrix} s_2 - y_1(t_2, s_1) \\ \vdots \\ s_N - y_{N-1}(t_N, s_{N-1}) \\ g(s_1, y_N(b, s_N)) \end{pmatrix} \tag{3.30}$$

The system we obtain from imposing this residual function equal to zero is a set of $n\,N$ equations in the same number of unknowns. If we use the Newton method to solve this system of nonlinear algebraic equations we obtain a Jacobian matrix exactly of the form of (3.23), where the diagonal values are obtained as the solution of the linearized ODE problems

$$\begin{cases} Y_i' = A(t)\,Y_i, & t_i \leq t \leq t_{i+1} \\ Y_i(t_i) = F_i \end{cases} \tag{3.31}$$

where $A(t) := \frac{\partial}{\partial y} f(t, y(t,s))$.

Multiple Shooting for Parameter Estimation

Consider now the problem of the estimation of the initial condition and/or parameters of a linear ODE of the form (3.19a). The aim is to estimate the unknown values minimizing the distance of the obtained trajectory of the ODE system with some noisy measured data points, i.e. the aim is to approximate some noisy data points with a trajectory of the ODE, obtaining the PEM formulation

$$\begin{aligned} \min_{c,p} \quad & \|r(t,c,p)\|_2^2 \\ \text{s.t.} \quad & y'(t) = f(t,y,p), \quad a < t < b \\ & y(a) = c \end{aligned}$$

where c is the initial condition, p is the parameter vector and r is the residual function given by the difference of the measured data samples d_i for $i = 1, \ldots, M$ at times t_{mi}. The residual function has the following form

$$r = \begin{pmatrix} y(t_0, c, p; t_{m1}) - d_1 \\ y(t_0, c, p; t_{m2}) - d_2 \\ \vdots \\ y(t_0, c, p; t_{mM}) - d_M \end{pmatrix} \tag{3.32}$$

Remark: the mesh subdivision of the measures is in general different from the mesh of the multiple-shooting nodes; the two meshes can coincide, share some points or neither of these two options.

We apply now the multiple-shooting method as done in the previous section. In this case continuity equations (3.28) remain the same, but the boundary conditions are not defined, hence the residual function is not the same as in (3.30). The continuity conditions are not sufficient to determine the multiple shooting nodes in a unique way, hence we add the minimization of the residual function (3.32) given from the observations and we obtain an extended residual vector that is a system of $M + (N -$

1) equality conditions:

$$r_{ms}(c,p,s) = \begin{pmatrix} Y_{1*}(t_0, c, p; t_{m1}) - d_1 \\ Y_{2*}(t_*, s_* c, p; t_{m2}) - d_2 \\ \vdots \\ Y_{M*}(t_*, s_*, p; t_{mM}) - d_M \\ s_2 - y_1(t_2, s_1) \\ \vdots \\ s_N - y_{N-1}(t_N, s_{N-1}) \end{pmatrix} \tag{3.33}$$

Note that the trajectory values in the observation errors are chosen in the respective interval of the Multiple-shooting mesh.

We can solve this Least Squares problem by the Gauss-Newton method and obtain the linear system:

$$J_r \, \delta_{[c,p,s]} = -r_{ms}(c,p,s) \tag{3.34}$$

where J_r is the Jacobian of the extended residual function with respect to the initial conditions, parameters and shooting nodes.

The matrix J_r is in $\mathbb{R}^{n(M+N) \times nN + n_p}$. Moreover it has a block structure of the following form:

$$J = \begin{bmatrix} J_s^0 & 0 & \cdots & 0 & J_p^0 \\ 0 & J_s^1 & \cdots & 0 & J_p^1 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & J_s^K & J_p^k \\ I_n & 0 & \cdots & 0 & 0_{n \times n_p} \\ 0 & I_n & \cdots & 0 & 0_{n \times n_p} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & I_n & 0_{n \times n_p} \end{bmatrix} \tag{3.35}$$

where

- $J_s^i = \frac{\partial y_i}{\partial s_i} \in \mathbb{R}^{nM_i \times n}$, where we called M_i the number of measurement relative to the interval i. Note that if we choose the shooting grid equal to the measurements grid, we obtain $M_i = 1$ for all i and hence square blocks in $\mathbb{R}^{n \times n}$,
- $J_p^i = \frac{\partial y_i}{\partial p} \in \mathbb{R}^{n \times n_p}$,
- I_n are the identity matrices in $\mathbb{R}^{n \times n}$ and
- $0_{n \times n_p}$ are the zero matrices in $\mathbb{R}^{n \times n_p}$

Remark: The multiple shooting method for parameter estimation is important in the case in which the trajectory is sensitive to initial conditions or parameters: if small variations of parameters or initial conditions cause a big difference in the trajectories this can cause the estimated trajectory to diverge if a single-shooting method is used (for example chaotic systems).

3.3.3 The Lorentz model example

Recall the example of the Lorentz model of Example 4:

$$\frac{dy_1}{dt}(t) = -py_1(t) + py_2(t) \tag{3.36}$$
$$\frac{dy_2}{dt}(t) = (r - y_3(t))y_1(t) - y_2(t) \tag{3.37}$$
$$\frac{dy_3}{dt}(t) = y_1(t)y_2(t) - by_3(t) \tag{3.38}$$

in the original formulation, these equations described a hydrodynamic dissipative flux with forcing term, and the parameters p and r the Prandtl and Rayleigh numbers respectively.

We will show the solution of parameter and initial condition estimation in two cases, with small values of p and r and for bigger values of them. In this second case we will notice that the classical PEM formulation is not sufficient to retrieve good estimates when the initial guess of the unknown values is not sufficiently good. Instead, the Multiple-shooting method allow us to regularize the problem.

As opposed to Multiple-shooting, the PEM formulation of the estimation problem is also called in this context Single-shooting.

In Figure 3.1 we can see the real trajectory of the system with parameters $p = 8$, $r = 10$, $b = 8/3$, and initial condition $y0 = [-8.0969, -6.9108, 28.0485]$. The system is solved by the Explicit Euler method with time step $dt = 0.01$ in the interval $[0, 2.7]$.

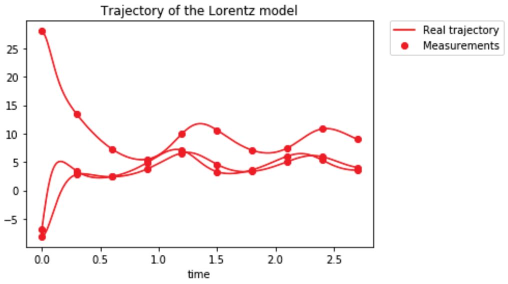


Figure 3.1: Real trajectory

First we try to retrieve the parameters p, r and the initial condition, starting from a guess of $[p, r] * 0.9$ and $y0 * 1.1$, the result with the Single shooting (PEM) formulation is in Figure 3.2.

Then we try with an initial guess more distant from the true solution, i.e. with $y0 * 1.2$, and solving it with the Single shooting method, the optimization of the library

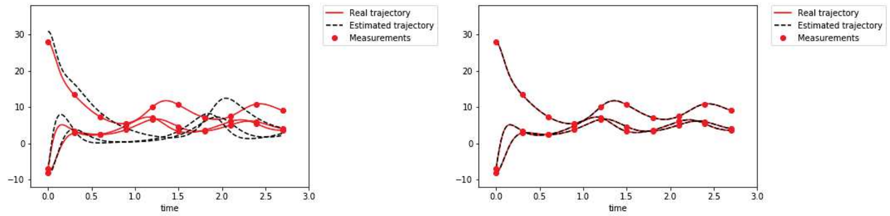


Figure 3.2: Single Shooting, convergent case

scipy is not able to retrieve the solution, as we can see in Figure 3.3. This is due to the fact that the system is near to chaotic and the solution explode during the iterative steps of the optimization. Multiple-shooting is able to handle this situation, in fact dividing the interval of time in more little subintervals, the cumulative error in the integration are smaller since the time of its propagation is smaller: the conditioning of the problem is less severe. The solution corresponding to the Multiple-shooting formulation is shown in Figure 3.4.

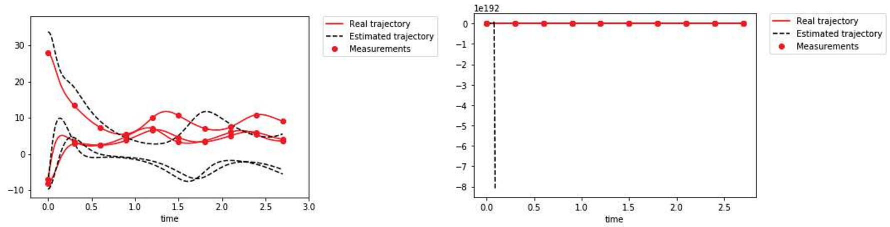


Figure 3.3: Single Shooting, not convergent case, chaotic example

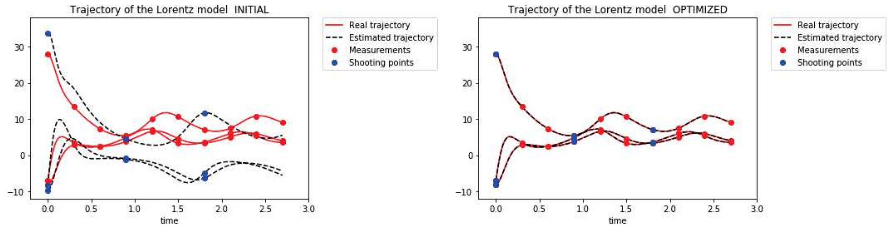


Figure 3.4: Multiple Shooting, chaotic case

3.4 Subspace methods for DLTI models

Subspace methods ([77], [76], and various extensions such as [51]) are a class of methods to solve the problem of identifying an LTI state-space model from input-output data through some structured block Hankel matrices. These methods are based on linear algebra techniques such as SVD, QR factorization and least-squares solutions.

These methods are divided in the case with deterministic DLTI system and the one for the stochastic case, in which the DLTI system with both measurement and model white noises is used [76]

$$\begin{cases} x(k+1) &= A(k)x(k) + B(k)u(k) + w(k) \\ y(k) &= C(k)x(k) + D(k)u(k) + v(k) \end{cases} \tag{3.39}$$

with the process noise $w(k)$ and measurement noise $v(k)$ assumed to be zero mean white-noise sequences with joint covariance matrix

$$E\left[\begin{bmatrix} v(k) \\ w(k) \end{bmatrix} \begin{bmatrix} v(j)^T & w(j)^T \end{bmatrix}^T\right] = \begin{bmatrix} R_{vv}(k) & R_{wv}(k)^T \\ R_{wv}(k) & R_{ww}(k) \end{bmatrix} \Delta(k-j) \geq 0$$

with $R_{vv}(k) > 0$ and where $\Delta(k)$ is the unit pulse.

In general the problem we are interested in is to find the parameters of an LTI system, i.e. to identify state-space models, on the basis of measured data. This problem can be solved with the Subspace methods, that we will introduce in this Section, but also with a PEM formulation.

The PEM method has a simpler formulation and consist in the constrained optimization problem where the variables are the unknown parameters of the model and the initial state, and the objective function is the difference between the measured data and the predictions obtained from the model:

$$\min_{(A,B,C,D),x(0)} \quad \frac{1}{2}\|y - y_{meas}\|_2^2 \tag{3.40}$$

$$\text{s.t.} \quad \begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{cases} \tag{3.41}$$

This approach does not exploit the linear structure of the model and requires a high computational cost to be solved.

The Subspace method instead, uses linear algebra methods such as the QR and SVD factorizations to solve the above problem. This method, however, finds only a suboptimal solution which is usually used as an initial condition for the PEM method.

Subspace methods for deterministic DLTI systems

We treat here only the subspace identification problem of DLTI systems in the deterministic case, to give an idea of the structure at the basis of these methods.

Problem 1. *Given a minimal (reachable and observable) DLTI system*

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{cases} \tag{3.42}$$

with $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $y(k) \in \mathbb{R}^l$, and given a finite number of measured samples of input and output signals u and y, the problem is to determine the system matrices (A, B, C, D) and the initial state vector $x(0)$ up to a similarity transformation.

The first step for the resolution of this problem is to build the *data equation* that relates the measured input and output data and the matrices in one unique equation. We recall that the state of the system with initial state $x(0)$ is given by

$$x(k) = A^k x(0) + \sum_{i=0}^{k-1} A^{k-i-1} B u(i) \qquad \forall k > 0 \tag{3.43}$$

and from the second equation of the system (3.42) we have

$$\begin{aligned} y(k+1) &= Cx(k+1) + Du(k+1) = \\ &= CAx(k) + CBu(k) + Du(k+1) = \\ &= C\left(A^{k+1}x(0) + \sum_{i=0}^{k} A^{k-i-1}Bu(i)\right) + Du(k+1) = \\ &= CA^{k+1}x(0) + \sum_{i=0}^{k} CA^{k-i-1}Bu(i) + Du(k+1). \end{aligned}$$

Considering a batch of data with samples $k = 0, \dots, s-1$, with s a finite positive integer, we have the system

$$\begin{bmatrix} y(0) \\ \vdots \\ y(s-1) \end{bmatrix} = \mathcal{O}_s x(0) + \mathcal{T}_s \begin{bmatrix} u(0) \\ \vdots \\ u(s-1) \end{bmatrix}, \tag{3.44}$$

where

$$\mathcal{O}_s = \begin{bmatrix} CA \\ \vdots \\ CA^{s-1} \end{bmatrix} \quad \text{and} \quad \mathcal{T}_s = \begin{bmatrix} D & 0 & \dots & 0 & 0 \\ CB & D & \dots & 0 & 0 \\ \vdots & \vdots & \ddots \vdots & & \vdots \\ CA^{s-2}B & CA^{s-3}B & \dots & CB & D \end{bmatrix}.$$

The matrix $\mathcal{O}_s$ is called *extended observability matrix* and the lower block triangular Toeplitz matrix $\mathcal{T}_s$ contains the Markov parameters of the system, i.e. $(D, CB, CAB, CA^2B \dots)$. Since the system is time-invariant, we can consider time shifts of the previous equality, and extend the equation. For this, we need Hankel matrices, that are matrices characterized by constant block anti-diagonals. We define the $sl \times N$ matrix

$$Y_{i,s,N} = \begin{bmatrix} y(i) & y(i+1) & \dots & y(i+N-1) \\ y(i+1) & y(i+2) & \dots & y(i+N) \\ \vdots & \vdots & \ddots & \vdots \\ y(i+s-1) & y(i+s) & \dots & y(i+N+s-2) \end{bmatrix} \tag{3.45}$$

and analogously the $sm \times N$ matrix $U_{i,s,N}$ built from the input signal $u(k)$. With these matrices we can write

$$Y_{0,s,N} = \mathcal{O}_s X_{0,N} + \mathcal{T}_s U_{0,s,N} \tag{3.46}$$

where

$$X_{0,N} = [x(0), \ldots, x(i+N-1)]$$

and $n < s \ll N$ with N the number of samples.

We want to use this equation to compute matrices (A, B, C, D) up to a similarity transform, from measurement input-output data. The first step will be to compute the column space of $\mathcal{O}_s$ from which the matrices A, C are derived and then from these the remaining values $(x(0), B, D)$ will be determined.

Column space of $\mathcal{O}_s$ The data equation is multiplied on the right by the projection matrix $\Pi^{\perp}_{U_{0,s,N}}$ which is the orthogonal projection on the column space of $U_{0,s,N}$

$$\Pi^{\perp}_{U_{0,s,N}} = I_N - U^T_{0,s,N}(U_{0,s,N}U^T_{0,s,N})^{-1}U_{0,s,N}$$

and satisfies $U_{0,s,N}\Pi^{\perp}_{U_{0,s,N}} = 0$. After this multiplication the data equation becomes

$$Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}} = \mathcal{O}_s X_{0,N}\Pi^{\perp}_{U_{0,s,N}}.$$

This step requires a rank condition on $U_{0,s,N}$: the matrix must be full row rank, i.e. $U_{0,s,N}U^T_{0,s,N}$ must be full rank. This means that not all input signals are admitted, the condition of *persistency of excitation* is used to characterize the valid input signals that can be used to identify the system.

Definition 9 (Persistency of excitation, [77] pag. 358)**.** *The input sequence $u(k) \in \mathbb{R}^m$ is persistently exciting of order n if and only if there exists an integer N such that the matrix*

$$U_{0,n,N} = \begin{bmatrix} u(0) & u(1) & \ldots & u(N-1) \\ u(1) & u(2) & \ldots & u(N) \\ \vdots & \vdots & \ddots & \vdots \\ u(n-1) & u(n) & \ldots & u(N+n-2) \end{bmatrix} \tag{3.47}$$

has full rank $n * m$.

Moreover, it holds that, under certain hypothesis on the input $u(k)$, $Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}$ has the same column space of $\mathcal{O}_s$, more precisely (Lemma 9.1 of [77]) if

$$\text{rank}\left(\begin{bmatrix} X_{0,N} \\ U_{0,s,N} \end{bmatrix}\right) = n + sm \qquad \text{then} \qquad \text{rank}\left(Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}\right) = n$$

and

$$range(Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}) = range(\mathcal{O}_s).$$

For efficiency purposes, usually it is considered the QR factorization

$$\begin{bmatrix} U_{0,s,N} \\ Y_{0,s,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

with $R_{11} \in \mathbb{R}^{sm \times sm}$, $R_{22} \in \mathbb{R}^{sl \times sl}$ and $R_{21} \in \mathbb{R}^{sl \times sm}$. This gives another estimate for the range of $\mathcal{O}_s$: from Theorem 9.1 of [77] it holds $range(\mathcal{O}_s) = range(R_{22})$. In this way the computation of the product $Y_{0,s,N} \Pi^{\perp}_{U_{0,s,N}}$, which implies big matrices due to the big number of samples N, is avoided.

System Realization The next step is the computation of the matrices (A, B, C, D) up to a similarity transformation, hence the system realization problem. First the matrices A, C are computed from the estimate of the range of $\mathcal{O}_s$, that we know to be R_{22} from the QR decomposition Given the SVD decomposition

$$R_{22} = U_n \sigma_n V_n^T$$

we have that $U_n = \mathcal{O}_s T$ for some similarity matrix T . Hence the matrix C equals, up to a similarity transformation, the first l rows of U_n and the matrix A can be computed by solving the overdetermined linear system

$$U_n(1 : (s-1)l, :)A = U_n(l+1 : sl, :).$$

Then the estimates B, D and $x(0)$ are the solution of the linear least-squares problem

$$(B, D, x(0)) = argmin \sum_{k=0}^{N+s-2} \|CA^k x(0) + \sum_{i=0}^{k-1} (CA^{k-i-1} Bu(i)) + Du(k) - y(k)\|_2^2.$$

Chapter 4

Parameter Estimation with unmodeled dynamics

4.1 Model Error Modeling

The application of System Identification for control purposes has been one of the most important applications of the subject since its birth [71]. In the recent years, the interest has been on Robust Identification that consists in the estimation not only of a nominal model for the system but also of the uncertainty associated with it. The uncertainty is usually divided in two components, that are unmodeled dynamics and noise on the available data, and when some structure for its modelization is considered it is called *Model Error Model*. Various approaches have been studied depending on the situation and the aim, which are summarized for example in [52, 65]. In this context, the focus is on the validation of the nominal model, which is intended to be used as it is, if validated, and not improved with the estimated model error model to obtain a more complex model. Our concern however is different, and consists in the physical meaning of the parameter estimation when unmodeled dynamics are present.

In fact, if some dynamics of a system are neglected, the fitting of a nominal physical model (white box model) on the real data, will not give the true parameters because of the error in the model. In the approach we propose (to deal with this problem) we are not interested in creating a model for the uncertainty, but instead to retrieve an estimate for the true physical parameters of the nominal model.

4.2 Unmodeled dynamics and the Unbiased Least Squares (ULS)

4.2.1 Introduction

The well known least-squares problem [8], very often used to estimate the parameters of a mathematical model, assumes an equivalence between a matrix-vector product Ax on the left, and a vector b on the right hand side: the matrix A is produced by the true model equations, evaluated at some operating conditions, the vector x contains the unknown parameters and the vector b are measurements, corrupted by white, Gaussian noise. This equivalence cannot be satisfied exactly, but the least-squares solution yields a minimum variance, maximum likelihood estimate of the parameters x, with a nice geometric interpretation: the resulting predictions Ax are at the minimum Euclidean distance from the true measurements b and the vector of residuals is orthogonal w.r.t. the subspace of all possible predictions.

Unfortunately, each violation of these assumptions produces in general a bias in the estimates. Various modifications have been introduced in the literature to cope with some of them: mainly, colored noise on b and/or A due to model error and/or colored measurement noise. The model error is often assumed as an additive stochastic term in the model, e.g., error-in-variables [74, 69], with consequent solution methods like Total Least-Squares [75] and Extended Least-Squares [62], to cite a few. All these techniques let the model to be modified to describe, in some sense, the model error.

Here, instead, we assume that the model error depends from deterministic variables in a way that has not been included in the model, i.e., we suppose to use a reduced model of the real system, as it is often the case in applications. We propose a method to cope with the bias in the parameter estimates of the approximate model by exploiting the geometric properties of least-squares and using small additional a-priori information about the norm of the modelled and un-modelled components of the system response, available with some approximation in most applications. To eliminate the bias on the parameter estimates we perturb the right-hand-side without modifying the reduced model, since we assume it describes accurately one part of the true model.

4.2.2 Model Problem

In applied mathematics, physical models are often available, usually rather precise at describing quantitatively the main phenomena, but not satisfactory at the level of detail required by the application at hand. Here we refer to models described by differential equations, with ordinary and/or partial derivatives, commonly used in engineering and applied sciences. We assume, therefore, that there are two models at hand: a true, unknown model $\mathcal{M}$ and an approximate, known model $\mathcal{M}_a$. These models are usually parametric and they must be tuned to describe a specific physical system, using a-priori knowledge about the application and experimental measurements. Model tuning, and in particular parameter estimation, is usually done with a prediction error minimization criterion that makes the model response to be a good approximation of the dynamics shown by the measured variables used in the estimation process. Assuming that the true model $\mathcal{M}$ is linear in the parameters that must be estimated, the application of this criterion brings to a linear least-squares problem:

$$\bar{x} = \operatorname*{argmin}_{x' \in \mathbb{R}^n} \|Ax' - \bar{f}\|^2, \tag{4.1}$$

where, from here on, $\|\cdot\|$ is the Euclidean norm, $A \in \mathbb{R}^{m \times n}$ is supposed full rank, i.e. $\operatorname{rank}(A) = n$, $m \geq n$, $\bar{x} \in \mathbb{R}^{n \times 1}$, Ax' are the model response values and $\bar{f}$ is the vector of experimental measurements. Usually the measured data contain noise, i.e., we measure $f = \bar{f} + \epsilon$, with ϵ a certain kind of additive noise (e.g., white Gaussian). Since we are interested here in algebraic and geometric aspects of the problem, we suppose $\epsilon = 0$ and set $f = \bar{f}$. Moreover, we assume ideally that $\bar{f} = A\bar{x}$ holds exactly. Let us consider also the estimation problem for the approximate model $\mathcal{M}_a$:

$$x^{\|} = \operatorname*{argmin}_{x' \in \mathbb{R}^{n_a}} \|A_a x' - \bar{f}\|^2, \tag{4.2}$$

where $A_a \in \mathbb{R}^{m \times n_a}$, $x^{\|} \in \mathbb{R}^{n_a \times 1}$, with $n_a < n$. The choice of the notation for $x^{\|}$ is to remind that the least-squares solution satisfies $A_a x^{\|} = P_{A_a}(f) =: f^{\|}$, where $f^{\|}$ is the orthogonal projection of $\bar{f}$ on the subspace generated by A_a, and the residual $A_a x^{\|} - \bar{f}$ is orthogonal to this subspace. Let us suppose that A_a corresponds to the first n_a columns of A, which means that the approximate model $\mathcal{M}_a$ is exactly one part of the true model $\mathcal{M}$, i.e., $A = [A_a, A_u]$ and so the solution $\bar{x}$ of (4.1) can be decomposed in two parts such that

$$A\bar{x} = [A_a, A_u] \begin{bmatrix} \bar{x}_a \\ \bar{x}_u \end{bmatrix} = A_a \bar{x}_a + A_u \bar{x}_u = \bar{f}. \tag{4.3}$$

This means that the model error corresponds to an additive term $A_u \bar{x}_u$ in the estimation problem.

Note that the columns of A_a are linearly independent since A is supposed to be of full rank. We do not consider the case in which A_a is rank-deficient, because it would mean that the model is not well parametrized. Moreover, some noise in the data is sufficient to determine a full rank matrix.

For brevity, we will call $\mathcal{A}$ the subspace generated by the columns of A and $\mathcal{A}_a$, $\mathcal{A}_u$ the subspaces generated by the columns of A_a, A_u respectively. Note that if $\mathcal{A}_a$ and $\mathcal{A}_u$ were orthogonal, decomposition (4.3) would be orthogonal. However, in the following we will consider the case in which the two subspaces are not orthogonal, as it commonly happens in practice. Oblique projections, even if not as common as orthogonal ones, have a large literature, e.g. [60, 33].

Before introducing the definitions of orthogonal projections and projectors, we recall some basic definitions: linear transformations and operators, and range and null space of a linear function.

Definition 10 (Linear Transformations and operators , from [60] pag 238)**.** *Let $\mathcal{U}$ and $\mathcal{V}$ be vector spaces over a field $\mathcal{F}$ ($\mathbb{R}$ or $\mathbb{C}$ for us).*

- *A linear transformation from $\mathcal{U}$ and $\mathcal{V}$ is defined to be a linear function T mapping $\mathcal{U}$ into $\mathcal{V}$. That is, $T(\alpha x + y) = \alpha T(x) + T(y) \quad \text{for all} \quad x, y \in \mathcal{U}, \alpha \in \mathcal{F}$.*

- *A linear operator on $\mathcal{U}$ is defined to be a linear transformation from $\mathcal{U}$ into itself—i.e., a linear function mapping $\mathcal{U}$ back into $\mathcal{U}$.*

For a linear function f mapping $\mathbb{R}^n$ into $\mathbb{R}^m$, let $R(f)$ denote the **range** of f. That is, $R(f) = \{f(x)|x \in \mathbb{R}^n\} \subseteq \mathbb{R}^m$ is the set of all "images" as x varies freely over $\mathbb{R}^n$, moreover it is a subspace of $\mathbb{R}^m$.

While the set of vectors that are mapped to 0, i.e. $N(f) = \{x|f(x) = 0\}$, is called the **nullspace** of f (or kernel of f).

We can introduce here the definitions of orthogonal and oblique projections as in [60]:

Definition 11 (Orthogonal Projection [60], pag 429)**.** *For $v \in \mathcal{V}$, let $v = m + n$, where $m \in \mathcal{M}$ and $n \in \mathcal{M}^\perp$.*

- *m is called the orthogonal projection of v onto $\mathcal{M}$.*

- *The projector $P_\mathcal{M}$ onto $\mathcal{M}$ along $\mathcal{M}^\perp$ is called the orthogonal projector onto $\mathcal{M}$.*

- *$P_\mathcal{M}$ is the unique linear operator such that $P_\mathcal{M}(v) = m$.*

Definition 12 (Projectors [60], pag 386)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be complementary subspaces of a vector space $\mathcal{V}$ so that each $v \in \mathcal{V}$ can be uniquely resolved as $v = x + y$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The unique linear operator P, defined by $Pv = x$ is called the* projector onto $\mathcal{X}$ along $\mathcal{Y}$*, and P has the following properties.*

- *$P^2 = P$ (P is idempotent).*

- *$I - P$ is the complementary projector onto $\mathcal{Y}$ along $\mathcal{X}$.*

- *$R(P) = \{x|Px = x\}$ (the set of "fixed points" for P), with $R(P)$ the range space of P.*

- *$R(P) = N(I - P) = \mathcal{X}$ and $R(I - P) = N(P) = \mathcal{Y}$, where $N(f)$ is the nullspace of the linear operator f.*

- *If $\mathcal{V} = \mathbb{R}^n$ or $\mathbb{C}^n$, then P is given by $P = [X|0]\,[X|Y]^{-1} = [X|Y] \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} [X|Y]^{-1}$, where the columns of X and Y are respective bases for $\mathcal{X}$ and $\mathcal{Y}$.*

Now, it is well known and easy to demonstrate that, when we solve problem (4.2) and A_u is not orthogonal to A_a, we get a biased solution, i.e., $x^{\|} \neq \bar{x}_a$:

Lemma 1. *Given $A \in \mathbb{R}^{m\times n}$ with $n \geq 2$ and $A = [A_a, A_u]$, and given $b \in \mathbb{R}^{m\times 1} \notin \mathcal{I}_m(A_a)$, call x the least-squares solution of* (4.2) *and $\bar{x} = [\bar{x}_a, \bar{x}_u]$ the solution of* (4.1) *decomposed as in* (4.3)*. Then*

(i) *if $A_u \perp A_a$ then $x^{\|} = \bar{x}_a$,*

(ii) *if $A_u \not\perp A_a$ then $x^{\|} \neq \bar{x}_a$.*

Proof. The least-squares problem $Ax = f$ boils down to finding x such that $Ax = P_{\mathcal{A}_a}(f)$. Let us consider the unique decomposition of f on $\mathcal{A}_a$ and $\mathcal{A}_a^{\perp}$ as $f = f^{\|} + f^{\perp}$ with $f^{\|} = P_{\mathcal{A}_a}(f)$ and $f^{\perp} = P_{\mathcal{A}_a^{\perp}}(f)$. Call $f = f_a + f_u$ the decomposition on $\mathcal{A}_a$ and $\mathcal{A}_u$, hence there exist two vectors $x_a \in \mathbb{R}^{n_a}, x_u \in \mathbb{R}^{n-n_a}$ such that $f_a = A_a x_a$ and $f_u = A_u x_u$. If $\mathcal{A}_u \perp \mathcal{A}_a$ then the two decompositions are the same, hence $f^{\|} = f_a$ and so $x^{\|} = \bar{x}_a$. Otherwise, for the definition of orthogonal projection ([60], third point of Def at page 429), it must hold $x^{\|} \neq \bar{x}_a$. □

This and the following three Lemmas are preliminary results to a more general one shown in Theorem 2.

4.2.3 Analysis of the Parameter Estimation Error

The aim here is to propose a method to decrease substantially the bias of the solution of the approximated problem (4.2), with the smallest additional information about the norms of the model error and of the modelled part responses.

In this subsection we will introduce sufficient conditions to remove the bias and retrieve the true solution in a unique way, as summarized in Lemma 4. Let us start with a definition.

Definition 13 (Intensity Ratio)**.** *The intensity ratio I_f between modelled and un-modelled dynamics is defined as*

$$I_f = \frac{\|A_a x_a\|}{\|A_u x_u\|}.$$

In the following we assume that a good approximation of this intensity ratio is available and that its magnitude is sufficiently big, i.e., we have an approximate model that is quite accurate. This information about the model error will be used to reduce the bias, as shown in the following subsections. Moreover we will consider also the norm $N_f = \|A_a x_a\|$ (or, equivalently, the norm $\|A_u x_u\|$).

The Case of Exact Knowledge about I_f and N_f

Here we assume, initially, to know the exact values of I_f and N_f, i.e.,

$$\begin{cases} N_f = \bar{N}_f = \|A_a \bar{x}_a\|, \\ I_f = \bar{I}_f = \frac{\|A_a \bar{x}_a\|}{\|A_u \bar{x}_u\|}. \end{cases} \tag{4.4}$$

This ideal setting is important to figure out the problem also with more practical assumptions. First of all, let us show a nice geometric property that relates x_a and f_a under a condition like (4.4).

Lemma 2. *The problem of finding the set of $x_a \in \mathbb{R}^n$ that gives a constant, prescribed value for I_f and N_f is equivalent to that of finding the set of $f_a = A_a x_a \in \mathcal{A}_a$ of the decomposition $f = f_a + f_u$ (introduced in the proof of Lemma 1) lying on the intersubsection of $\mathcal{A}_a$ and the boundaries of two n-dimensional balls in $\mathbb{R}^n$. In fact, it holds:*

$$\begin{cases} N_f = \|A_a x_a\| \\ I_f = \frac{\|A_a x_a\|}{\|A_u x_u\|} \end{cases} \iff \begin{cases} f_a \in \partial B_n(0, N_f) \\ f_a \in \partial B_n(f^{\|}, T_f) \end{cases} \quad \textit{with} \quad T_f := \sqrt{\left(\frac{N_f}{I_f}\right)^2 - \|f^{\perp}\|^2}. \tag{4.5}$$

Proof. For every $x_a \in \mathbb{R}^{n_a}$ it holds,

$$\begin{cases} N_f = \|f_a\| = \|A_a x_a\| \\ I_f = \frac{\|f_a\|}{\|f_u\|} = \frac{N_f}{\|f_u^{\perp} + f_u^{\|}\|} = \frac{N_f}{\sqrt{\|f^{\perp}\|^2 + \|f^{\|} - A_a x_a\|^2}} = \frac{N_f}{\sqrt{\|f^{\perp}\|^2 + \|f^{\|} - f_a\|^2}} \end{cases} \iff \tag{4.6}$$

$$\iff \begin{cases} \|f_a\| = N_f \\ \|f^{\|} - f_a\| = \sqrt{\left(\frac{N_f}{I_f}\right)^2 - \|f^{\perp}\|^2} =: T_f, \end{cases} \tag{4.7}$$

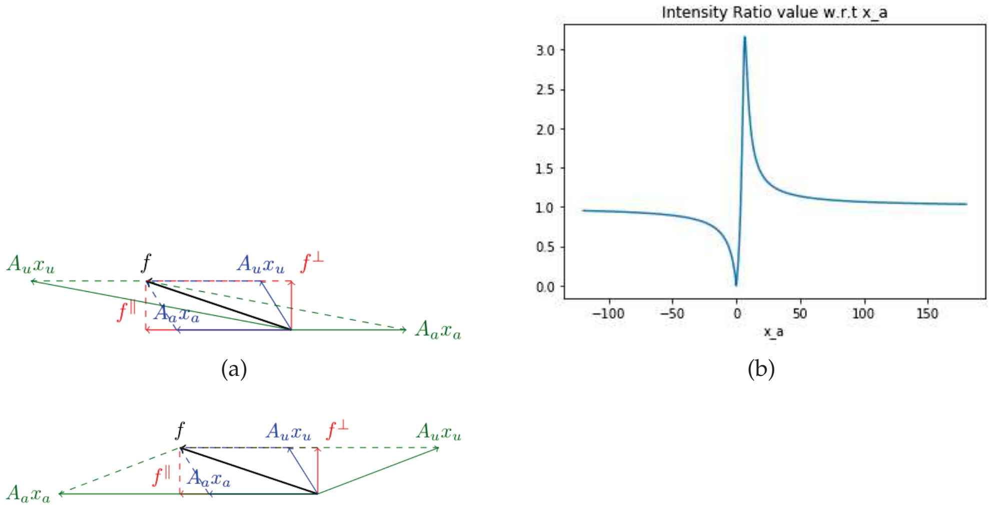
where we used the fact that $f_u = f_u^{\|} + f_u^{\perp}$ with $f_u^{\perp} := P_{A_a^{\perp}}(f_u) = f^{\perp}$, $f_u^{\|} := P_{A_a}(f_u) = A_a \delta x_a = f^{\|} - A_a x_a$, and $\delta x_a = (x^{\|} - x_a)$. Hence the equivalence (4.5) is proved. □

Given I_f and N_f, we call the feasible set of accurate model responses all the f_a that satisfy the relations (4.5). Now we will see that Lemma 2 allows us to reformulate problem (4.2) in the problem of finding a feasible f_a that, replaced to $\bar{f}$ in (4.2), gives as solution an unbiased estimate of $\bar{x}_a$. Indeed, it is easy to note that $A_a \bar{x}_a$ belongs to this feasible set. Moreover, since $f_a \in \mathcal{A}_a$, we can reduce the dimensionality of the problem and work on the subspace $\mathcal{A}_a$ which has dimension n_a, instead of the global space $\mathcal{A}$ of dimension n. To this aim, let us consider U_a the matrix of the SVD decomposition of A_a, $A_a = U_a S_a V_a^T$, and complete its columns to an orthonormal basis of $\mathbb{R}^n$ to obtain a matrix U. Since the vectors $f_a, f^{\|} \in \mathbb{R}^n$ belong to the subspace $\mathcal{A}_a$, the vectors $\tilde{f}_a, \tilde{f}^{\|} \in \mathbb{R}^n$ defined such that $f_a = U\tilde{f}_a$ and $f^{\|} = U\tilde{f}^{\|}$ must have zeros on the last $n - n_a$ components. Since U has orthonormal columns, it preserves the norms and so $\|f^{\|}\| = \|\tilde{f}^{\|}\|$ and $\|f_a\| = \|\tilde{f}_a\|$. If we call $\hat{f}_a, \hat{f}^{\|} \in \mathbb{R}^{n_a}$ the first n_a components of the vectors $\tilde{f}_a, \tilde{f}^{\|}$ (which have again the same norms of the full vectors in $\mathbb{R}^n$) respectively, we have

$$\begin{cases} \hat{f}_a \in \partial B_{n_a}(0, N_f), \\ \hat{f}_a \in \partial B_{n_a}(f^{\|}, T_f). \end{cases} \tag{4.8}$$

In this way the problem depends only on the dimension of the known subspace, i.e., the value of n_a, and does not depend on the dimensions $m \gg n_a$ and $n > n_a$. From (4.8) we can deduce the equation of the $(n_a - 2)$-dimensional boundary of an $(n_a - 1)$-ball to which the vector $f_a = A_a x_a$ must belong. In the following we discuss the various cases.

Case $n_a = 1$. In this case, we have one unique solution when both conditions on I_f and N_f are imposed. When only one of these two is imposed, two solutions are found, shown in Figure 4.1a and Figure 4.1c. Figure 4.1b shows the intensity ratio I_f.



(a)

(b)

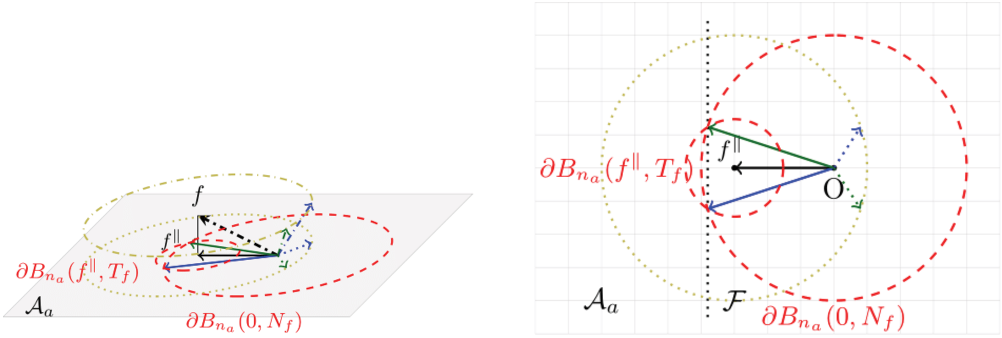
(c)

Figure 4.1: Case $n_a = 1$. **(a)**: Case $n_a = 1$, $m = n = 2$. Solutions with the condition on N_f. In the figure: the true decomposition obtained imposing both the conditions (blue), the orthogonal decomposition (red), another possible decomposition (green) that satisfy the same norm condition N_f, but different I_f; **(b)**: Case $n_a = 1$. Intensity Ratio value w.r.t the norm of the vector $A_a x_a$: given a fixed value of Intensity Ratio there can be two solution, i.e. two possible decomposition of f as sum of two vectors with the same Intensity Ratio; **(c)**: Case $n_a = 1$, $m = n = 2$. Solutions with the condition on I_f. In the figure: the true decomposition obtained imposing both the conditions (blue), the orthogonal decomposition (red), another possible decomposition (green) with the same intensity ratio I_f, but different N_f.

Case $n_a = 2$**.** Consider the vectors $\hat{f}_a, f^{\|} \in \mathbb{R}^{n_a=2}$ as defined previously, in particular we are looking for $\hat{f}_a = [\xi_1, \xi_2] \in \mathbb{R}^2$. Hence, conditions (4.8) can be written as

$$\begin{cases} \xi_1^2 + \xi_2^2 = N_f^2 \\ (\xi_1 - \hat{f}^{\|}_{\xi_1})^2 + (\xi_2 - \hat{f}^{\|}_{\xi_2})^2 = T_f^2 \end{cases} \longrightarrow \quad \mathcal{F}: \ (\hat{f}^{\|}_{\xi_1})^2 - 2\hat{f}^{\|}_{\xi_1}\xi_1 + (\hat{f}^{\|}_{\xi_2})^2 - 2\hat{f}^{\|}_{\xi_2}\xi_2 = N_f^2 - T_f^2, \tag{4.9}$$

where the right equation is the $(n_a - 1) = 1$-dimensional subspace (line) $\mathcal{F}$ obtained subtracting the first equation to the second. This subspace has to be intersected with one of the beginning circumferences to obtain the feasible vectors $\hat{f}_a$, as can be seen in Figure 4.2a and its projection on $\mathcal{A}_a$ in Figure 4.2b. The intersubsection of the two circumferences (4.5) can have different solutions depending on the value of $(N_f - \| f^{\|} \|) - T_f$. When this value is strictly positive there are zero solutions, this means that the estimates of I_f and N_f are not correct: we are not interested in this case because we suppose the two values to be sufficiently well estimated. When the value is strictly negative there are two solutions, that coincide when the value is zero.


f
$f^{\|}$
$\partial B_{n_a}(f^{\|}, T_f)$
$\mathcal{A}_a$
$\partial B_{n_a}(0, N_f)$
$\partial B_{n_a}(f^{\|}, T_f)$
$f^{\|}$
O
$\mathcal{A}_a$
$\mathcal{F}$
$\partial B_{n_a}(0, N_f)$

(a)

(b)

Figure 4.2: Case $n_a = 2$. **(a)**: Case $n_a = 2$, $m = n = 3$, with $A_a x_a = [A_a(1) A_a(2)][x_a(1) x_a(2)]^T$. In the figure: the true decomposition (blue), the orthogonal decomposition (red), another possible decomposition of the infinite ones (green); **(b)**: Case $n_a = 2$, $m = n = 3$. Projection of the two circumferences on the subspace $\mathcal{A}_a$, and projections of the possible decompositions of f (red, blue and green).

When there are two solutions, we have no sufficient information to determine which one of the two solutions is the true one, i.e., the one that gives $f_a = A_a \bar{x}_a$: we cannot choose the one that has minimum residual, neither the vector f_a that has the minimum angle with f, because both solutions have the same values of these two quantities. However, since we are supposing the linear system to be originated by an input/output system, where the matrix A_a is a function also of the input and f are the measurements of the output, we can take two tests with different inputs. Since all the solution sets contain the true parameter vector, we can determine the true solution from their intersubsection, unless the solutions of the two tests are coincident. The condition for coincidence is expressed in Lemma 3.

Let us call $A_{a,i} \in \mathbb{R}^{n\times n_a}$ the matrix of the test $i = 1,2$, to which correspond a vector f_i. The line on which lie the two feasible vectors f_a of the same test i is $\mathcal{F}_i$ and $\mathcal{S}_i = A^{\dagger}_{a,i}\mathcal{F}_i$ is the line through the two solution points. To have two tests with non-coincident solutions, we need that these two lines $\mathcal{S}_1, \mathcal{S}_2$ do not have more than one common point, that in the case $n_a = 2$ is equivalent to $\mathcal{S}_1 \neq \mathcal{S}_2$, i.e., $A^{\dagger}_{a,1}\mathcal{F}_1 \neq A^{\dagger}_{a,2}\mathcal{F}_2$, i.e., $\mathcal{F}_1 \neq A_{a,1}A^{\dagger}_{a,2}\mathcal{F}_2 =: \mathcal{F}_{12}$. We represent the lines $\mathcal{F}_i$ by means of their orthogonal vector from the origin $f^{ort,i} = l_{ort,i}\frac{f_i^{||}}{\| f_i^{||} \|}$, where $l_{ort,i} = \|f^{ort,i}\|$. We introduce the matrices C_a, C_f, C_{fp} such that

$$\begin{cases} A_{a,2} &= C_a A_{a,1} \\ f_2 &= C_f f_1 \\ f_2^{||} &= C_{fp} f_1^{||} \end{cases}$$

and k_f such that $\| f_2^{||} \| = k_f \| f_1^{||} \|$.

Lemma 3. *Consider two tests $i = 1,2$ from the same system with $n_a = 2$ with the above notation. Then it holds $\mathcal{F}_1 = \mathcal{F}_{12}$ if and only if $C_a = C_{fp}$.*

Proof. From the relation $f_i^{||} = \mathcal{P}_{A_{a,i}}(f_i) = A_{a,i}(A^T_{a,i}A_{a,i})^{-1}A^T_{a,i}f_i$, we have

$$f_2^{||} = A_{a,2}(A^T_{a,2}A_{a,2})^{-1}A^T_{a,2}f_2 = C_a A_{a,1}(A^T_{a,1}C_a^T C_a A_{a,1})^{-1}A^T_{a,1}C_a^T C_f f_1. \tag{4.10}$$

It holds $\mathcal{F}_1 = \mathcal{F}_{12} \iff f^{ort,1} = f^{ort,12} := A_{a,1}A^{\dagger}_{a,2}f^{ort,2}$, hence we will show this second equivalence. We note that $l_{ort,2} = k_f l_{ort,1}$ and calculate

$$\begin{aligned} f^{ort,12} &= A_{a,1}A^{\dagger}_{a,2}f^{ort,2} = A_{a,1}A^{\dagger}_{a,1}C_a^{\dagger}\left(l_{ort,2}\frac{f_2^{||}}{\| f_2^{||} \|}\right) = \\ &= A_{a,1}A^{\dagger}_{a,1}C_a^{\dagger}\left(k_f l_{ort,1}\frac{C_{fp}f_1^{||}}{k_f \| f_1^{||} \|}\right) = A_{a,1}A^{\dagger}_{a,1}C_a^{\dagger}C_{fp}f^{ort,1}. \end{aligned} \tag{4.11}$$

Now let us call $s^{ort,1}$ the vector such that $f^{ort,1} = A_{a,1}s^{ort,1}$, then, using the fact that $C_a = C_{fp}$ we obtain

$$f^{ort,12} = A_{a,1}A^{\dagger}_{a,1}C_a^{\dagger}C_{fp}A_{a,1}s^{ort,1} = A_{a,1}(A^{\dagger}_{a,1}A_{a,1})s^{ort,1} = A_{a,1}s^{ort,1} \tag{4.12}$$

where the last equality is given by the fact that $A^{\dagger}_{a,1}A_{a,1} = I_{n_a}$.

Hence we have

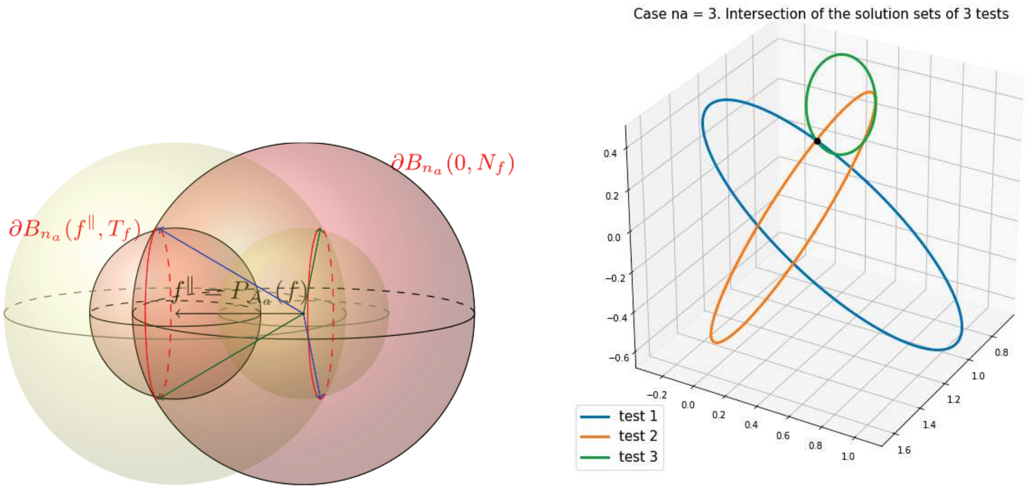
$$\mathcal{F}_{12} = \mathcal{F}_1 \iff A_{a,1}A^{\dagger}_{a,1}C_a^{\dagger}C_{fp}f^{ort,1} = f^{ort,1} \iff C_a^{\dagger}C_{fp} = I.$$

□

Case $n_a \geq 3$**.** More generally, for the case $n_a \geq 3$, consider the vectors $\hat{f}_a, \hat{f}^{\|} \in \mathbb{R}^{n_a}$ as defined previously, in particular we are looking for $\hat{f}_a = [\xi_1, \dots, \xi_{n_a}] \in \mathbb{R}^{n_a}$. Conditions (4.8) can be written as

$$\begin{cases} \sum_{i=1}^{n_a} \xi_i^2 = N_f^2 \\ \sum_{i=1}^{n_a} (\xi_i - \hat{f}^{\|}_{\xi_i})^2 = T_f^2 \end{cases} \longrightarrow \quad \mathcal{F}: \quad \sum_{i=1}^{n_a} ((\hat{f}^{\|}_{\xi_i})^2 - 2\hat{f}^{\|}_{\xi_i}\xi_i) = N_f^2 - T_f^2, \tag{4.13}$$

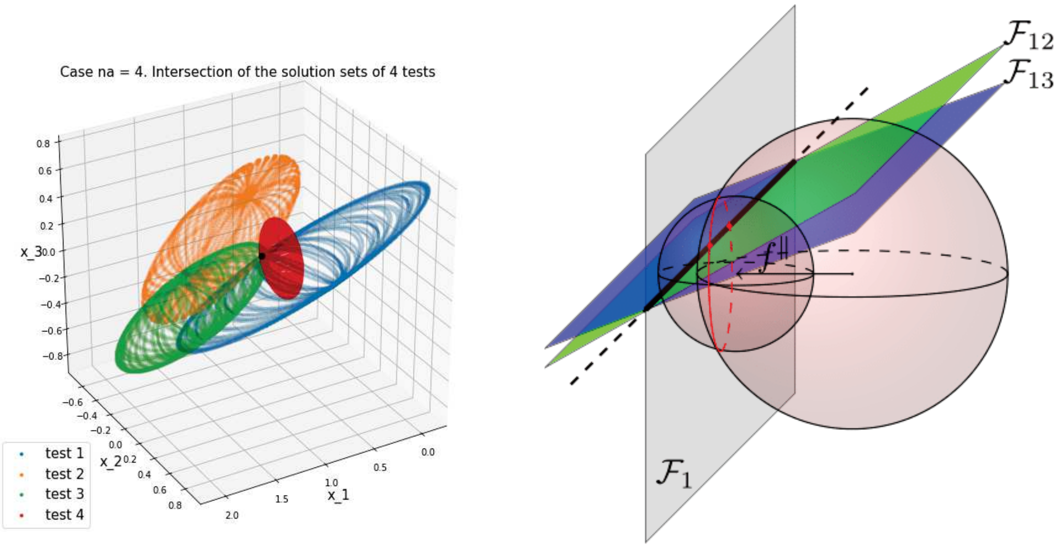
where the two equations on the left are two $(n_a - 1)$-spheres, i.e., the boundaries of two n_a-dimensional balls. Analogously to the case $n_a = 2$, the intersubsection of these equations can be empty, one point or the boundary of a $(n_a - 1)$-dimensional ball (with the same conditions on $(N_f - \| f^{\|} \|) - T_f$). The equation on the right of (4.13) is the $(n_a - 1)$-dimensional subspace $\mathcal{F}$ on which lies the boundary of the $(n_a - 1)$-dimensional ball of the feasible vectors f_a, and is obtained subtracting the first equation to the second one. In Figure 4.3a the graphical representation of the decomposition $f^{\|} = f_a + f_u^{\|}$ for the case $n_a = 3$ is shown, and in Figure 4.3b the solution ellipsoids of 3 tests whose intersubsection is one point. Figure 4.4a shows the solution hyperellipsoids of 4 tests whose intersubsection is one point, in the case $n_a = 4$.


$\partial B_{n_a}(0, N_f)$
$\partial B_{n_a}(f^{\|}, T_f)$
$f^{\|} = P_{\mathcal{A}_a}(f)$
Case na = 3. Intersection of the solution sets of 3 tests
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
0.8
1.0
1.2
1.4
1.6
test 1
test 2
test 3

(a)

(b)

Figure 4.3: Case $n_a = 3$. **(a)** Case $n_a = 3$, $m = n = 4$, $n - n_a = 1$: in the picture $\bar{f}^{\|}$, i.e. the projection of f on $\mathcal{A}_a$. The decompositions that satisfies the conditions on I_f and N_f are the ones with f_a that lies on the red circumference on the left. The spheres determined by the conditions are shown in yellow for the vector f_a and in blue for the vector $f^{\|} - a_a$. Two feasible decompositions are shown in blue and green; **(b)** Case $n_a = 3$. Intersubsection of three hyperellipsoids, set of the solutions x_a of three different tests, in the space $\mathbb{R}^{n_a=3}$.



(a) (b)

Figure 4.4: Case $n_a \geq 3$. **(a)** Case $n_a = 4$. Intersubsection of four hyperellipsoids, set of the solutions x_a of four different tests, in the space $\mathbb{R}^{n_a=4}$; **(b)** Case $n_a = 3$. Example of three tests for which the solution has an intersubsection bigger than one single point. The three $(n_a - 1)$-dimensional subspaces $\mathcal{F}_1, \mathcal{F}_{12}, \mathcal{F}_{13}$ in the space generated by $A_{a,1}$ intersect in a line and their three orthogonal vectors are not linearly independent.

We note that, to obtain one unique solution x_a we must intersect the solutions of at least two tests. Let us give a more precise idea of what happens in general. Given $i = 1, \ldots, n_a$ tests we call, as in the previous case, $f^{ort,i}$ the vector orthogonal to the $(n_a - 1)$-dimensional subspace $\mathcal{F}_i$ that contains the feasible f_a, and $\mathcal{S}_i = A^{\dagger}_{a,i}\mathcal{F}_i$. We project this subspace on $\mathcal{A}_{a,1}$ and obtain $\mathcal{F}_{1i} = A_{a,1}A^{\dagger}_{a,i}\mathcal{F}_i$ that we describe through its orthogonal vector $f^{ort,1i} = A_{a,1}A^{\dagger}_{a,i}f^{ort,i}$. If the vectors $f^{ort,1}, f^{ort,12}, \ldots f^{ort,1n_a}$ are linearly independent, it means that the $(n_a - 1)$-dimensional subspaces $\mathcal{F}_1, \mathcal{F}_{12}, \ldots \mathcal{F}_{1n_a}$ intersect themselves in one point. In Figure 4.4b it is shown an example in which, in the case $n_a = 3$ the vectors $f^{ort,1}, f^{ort,12}, f^{ort,13}$ are not linearly independent. The three solution sets of this example will intersect in two points, hence, for $n_a = 3$, three tests are not always sufficient to determine a unique solution.

Lemma 4. *For all $n_a > 1$, the condition that, given $i = 1, \ldots, n_a$ tests, the n_a hyperplanes $\mathcal{S}_i = A^{\dagger}_{a,i}\mathcal{F}_i$ previously defined have linearly independent normal vectors is sufficient to determine one unique intersubsection, i.e., one unique solution vector $\bar{x}_a$, that satisfies the system of conditions* (4.4) *for each test.*

Proof. The intersubsection of n_a independent hyperplanes in $\mathbb{R}^{n_a}$ is a point. Given a test i and $\mathcal{S}_i = A^{\dagger}_{a,i}\mathcal{F}_i$ the affine subspace of that test

$$\mathcal{S}_i = v_i + W_i = \{v_i + w \in \mathbb{R}^{n_a} \ : \ w \cdot \mathbf{n}_i = 0\} = \{x \in \mathbb{R}^{n_a} \ : \ \mathbf{n}_i^T(x - v_i) = 0\},$$

where $\mathbf{n}_i$ is the normal vector of the linear subspace and v_i the translation with respect to the origin.

The conditions on $\mathcal{S}_i$ relative to n_a tests correspond to a linear system $Ax = b$, where $\mathbf{n}_i$ is the i-th row of A and each component of the vector b given by $b_i = \mathbf{n}_i^T v_i$. The matrix A has full rank because of the linear independence condition of the vectors $\mathbf{n}_i$, hence the solution of the linear system is unique.

The unique intersubsection is due to the hypothesis of full column rank of the matrices $A_{a,i}$: this condition implies that the matrices $A_{a,i}$ map the surfaces $\mathcal{F}_i$ to hyperplanes $\mathcal{S}_i = A_{a,i}\mathcal{F}_i$. □

For example, with $n_a = 2$ (Lemma 3) this condition is equal to considering two tests with non-coincident lines $\mathcal{S}_1, \mathcal{S}_2$, i.e., two non-coincident $\mathcal{F}_1, \mathcal{F}_{12}$.

The Case of Approximate Knowledge of I_f and N_f Values

Let us consider N tests and call $I_{f,i}$, $N_{f,i}$ and $T_{f,i}$ the values as defined in Lemma 2, relative to test i. Since the system of conditions

$$\begin{cases} N_{f,i} = \|A_{a,i}x_a\| \\ I_{f,i} = \frac{\|A_{a,i}x_a\|}{\|z_i - A_{a,i}x_a\|} \end{cases} \quad \text{and} \quad \begin{cases} N_{f,i} = \|A_{a,i}x_a\| \\ T_{f,i} = \|f_i^{\|} - A_{a,i}x_a\| \end{cases} \tag{4.14}$$

is equivalent, as shown in Lemma 2, we will take into account the system on the right for its simplicity: the equation on $T_{f,i}$ represents an hyperellipsoid, translated with respect to the origin.

In a real application, we can assume to know only an interval in which the true values of I_f is contained and, analogously, an interval for N_f values. Supposing we know the bounds on I_f and N_f, then the bounds on T_f can be easily computed. Let us call these extreme values $N_f^{max}, N_f^{min}, T_f^{max}, T_f^{min}$, we will assume it always holds

$$\begin{cases} N_f^{max} \geq max_i(N_{f,i}), \\ N_f^{min} \leq min_i(N_{f,i}), \end{cases} \quad \text{and} \quad \begin{cases} T_f^{max} \geq max_i(T_{f,i}), \\ T_f^{min} \leq min_i(T_{f,i}), \end{cases} \tag{4.15}$$

for each i-th test of the considered set $i = 0, \ldots, N$.

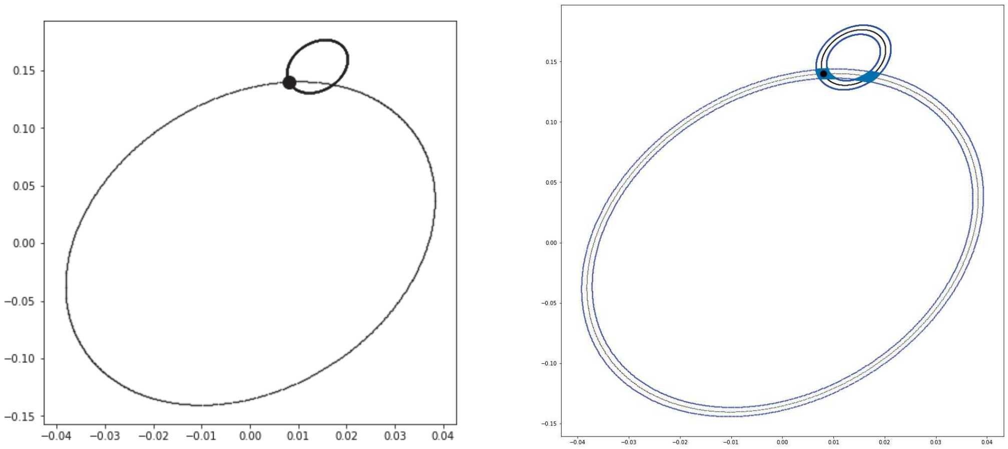
Condition (4.4) is now relaxed as follows: the true solution $\bar{x}_a$ satisfies

$$\begin{cases} \|A_{a,i}\bar{x}_a\| \leq N_f^{max}, \\ \|A_{a,i}\bar{x}_a\| \geq N_f^{min}, \end{cases} \quad \text{and} \quad \begin{cases} \|A_{a,i}\bar{x}_a - f_i^{\|}\| \leq T_f^{max}, \\ \|A_{a,i}\bar{x}_a - f_i^{\|}\| \geq T_f^{min}, \end{cases} \tag{4.16}$$

for each i-th test of the considered set $i = 0, \ldots, N$.

Assuming the extremes to be non-coincident ($N_f^{min} \neq N_f^{max}$ and $T_f^{min} \neq T_f^{max}$), these conditions do not define a single point, i.e., the unique solution $\bar{x}_a$ (as in (4.4) of subsection 4.2.3), but an entire closed region of the space that may be even not connected, and contains infinite possible solutions x different from $\bar{x}_a$.

In Figure 4.5 two examples, with $n_a = 2$, of the conditions for a single test are shown: on the left in the case of exact knowledge of the $N_{f,i}$ and $T_{f,i}$ values, and on the right with the knowledge of two intervals containing the right values.



(a)

(b)

Figure 4.5: Examples of the exact and approximated conditions on a test with $n_a = 2$. In the left equation the two black ellipsoids are the two constraints of the right system of (4.14), while in the right figure the two couples of concentric ellipsoids are the borders of the thick ellipsoids defined by (4.16) and the blue region Z_{r_i} is the intersubsection of (4.18) and (4.19). The black dot in both the figures is the true solution. **(a)** Exact conditions on N_f and T_f; **(b)** Approximated conditions on N_f and T_f.

Given a single test, the conditions (4.16) on a point x can be easily characterized. Given the condition

$$\|f_a\| = \|A_a x_a\| = N_f,$$

we write $x_a = \sum \chi_i v_i$ with v_i the vectors of the orthogonal basis, given by the columns V of the SVD decomposition $A_a = USV^T$. Then

$$f_a = A_a x_a = USV^T(\sum_i \chi_i v_i) = US(\sum_i \chi_i e_i) = U(\sum_i s_i \chi_i e_i) = \sum_i s_i \chi_i \mathbf{u}_i.$$

Since the norm condition $\|f_a\|^2 = \sum_i (s_i \chi_i)^2 = N_f^2$ holds, then we obtain the equation of the hyperellipsoid for x_a as :

$$\sum_i (s_i\chi_i)^2 = \sum_i \frac{\chi_i^2}{(\frac{1}{s_i})^2} = N_f^2. \tag{4.17}$$

The bounded conditions hence gives the region inside the two hyperellipsoids centered in the origin:

$$N_f^{min} \leq \sum_i \frac{\chi_i^2}{(\frac{1}{s_i})^2} \leq N_f^{max}. \tag{4.18}$$

We can proceed in an analogous way for the condition on x given by the system on the right of (4.16)

$$T_f^{min} \leq \|f_a - f^{\|}\| \leq T_f^{max}.$$

If we write $f^{\|}$ in the coordinates of the columns of the matrix U of the SVD decomposition of A_a, as we have done for f_a, we obtain $f^{\|} = \sum_i f_{u_i}^{\|}\mathbf{u}_i$. Now, we can see that the bounded conditions on T_f describe the region inside the two translated hyperellipsoids

$$T_f^{min} \leq \sum_i \left(s_i\chi_i - f_{u_i}^{\|}\right)^2 \leq T_f^{max}$$

and in a more explicit form

$$T_f^{min} \leq \sum_i \frac{\left(\chi_i - \frac{f_{u_i}^{\|}}{s_i}\right)^2}{(\frac{1}{s_i})^2} \leq T_f^{max}. \tag{4.19}$$

Given a test i, each of the conditions (4.18) and (4.19), constrain $\bar{x}_a$ to lie inside a thick hyperellipsoid, i.e., the region between the two concentric hyperellipsoids. The intersubsection of these two conditions for test i is a zero-residual region that we call Z_{r_i}

$$Z_{r_i} = \{x \in \mathbb{R}^{n_a} \mid \text{(4.18) and (4.19) hold}\}. \tag{4.20}$$

It is easy to verify that if $N_{f,i}$ is equal to the assumed N_f^{min} or N_f^{max}, or $T_{f,i}$ is equal to the assumed T_f^{min} or T_f^{max}, the true solution will be on a border of the region Z_{r_i}, and if it holds for both $N_{f,i}$ and $T_{f,i}$ it will lie on a vertex.

When more tests $i = 1, \ldots, N$ are put together, we have to consider the points that belong to the intersubsection of all these regions Z_{r_i}, i.e.,

$$I_{zr} = \bigcap_{i=0,\ldots,N} Z_{r_i}. \tag{4.21}$$

These points minimize, with zero residual, the following optimization problem:

$$\begin{aligned}\min_x \sum_{i=1}^{N} min(0, \|A_{a,i}x\| - N_f^{min})^2 + \sum_{i=1}^{N} max(0, \|A_{a,i}x\| - N_f^{max})^2 + \\ + \sum_{i=1}^{N} min(0, \|A_{a,i}x - f_i^{\|}\| - T_f^{min})^2 + \sum_{i=1}^{N} max(0, \|A_{a,i}x - f_i^{\|}\| - T_f^{max})^2.\end{aligned} \tag{4.22}$$

It is also easy to verify that, if the true solution lies on an edge/vertex of one of the regions Z_{r_i}, it will lie on an edge/vertex of their intersubsection.

The intersected region I_{zr} tends to monotonically shrink in a way that depends from the properties of the added tests. We are interested to study the conditions that make it reduce to a point, or at least to a small region. A sufficient condition to obtain a point is given in Theorem 2.

Let us first consider the function that, given a point in the space $\mathbb{R}^{n_a}$, returns the squared norm of its image through the matrix A_a:

$$\begin{aligned}N_f^2(x) = \|A_a x\|_2^2 = \|U\Sigma V^T x\|_2^2 = \|\Sigma V^T x\|_2^2 = (\Sigma V^T x)^T(\Sigma V^T x) = x^T(V\Sigma^T\Sigma V^T)x = \\ = \left\| \begin{bmatrix} \sigma_1 v_1^T x \\ \sigma_2 v_2^T x \\ \vdots \end{bmatrix} \right\|_2^2 = \sigma_1^2(v_1^T x)^2 + \sigma_2^2(v_2^T x)^2 + \dots,\end{aligned} \tag{4.23}$$

where v_i are the columns of V and $x = [x(1)\, x(2) \dots, x(n_a)]$.

The direction of maximum increase of this function is given by its gradient

$$\nabla N_f^2(x) = 2(V\Sigma^2 V^T)x = \begin{bmatrix} 2\sigma_1^2 v_1^T x v_1(1) + 2\sigma_2^2 v_2^T x v_2(1) + \dots + 2\sigma_{n_a}^2 v_{n_a}^T x v_{n_a}(1) \\ 2\sigma_1^2 v_1^T x v_1(2) + 2\sigma_2^2 v_2^T x v_2(2) + \dots + 2\sigma_{n_a}^2 v_{n_a}^T x v_{n_a}(2) \\ \vdots \end{bmatrix}. \tag{4.24}$$

Analogously, define the function $T_f^2(x)$ as

$$\begin{aligned}T_f^2(x) = \|A_a x - f^{\|}\|_2^2 = \|U\Sigma V^T x - f^{\|}\|_2^2 = \|\Sigma V^T x - f^{\|}\|_2^2 = \\ = (\Sigma V^T x - f^{\|})^T(\Sigma V^T x - f^{\|}) = (\Sigma V^T x)^T(\Sigma V^T x) - 2(\Sigma V^T x)^T f^{\|} + (f^{\|})^T(f^{\|}) \\ = x(V\Sigma^2 V^T)x - 2(x)^T V\Sigma f^{\|} + (f^{\|})^T(f^{\|}) = \\ = \left\| \begin{bmatrix} \sigma_1 v_1^T x \\ \sigma_2 v_2^T x \\ \vdots \end{bmatrix} - f^{\|} \right\|_2^2\end{aligned} \tag{4.25}$$

with gradient

$$\nabla T_f^2(x) = 2(V\Sigma^2 V^T)x - 2V\Sigma f^{\|} =$$

$$= \begin{bmatrix} 2\sigma_1^2 v_1^T x v_1(1) + 2\sigma_2^2 v_2^T x v_2(1) + \cdots + 2\sigma_{n_a}^2 v_{n_a}^T x v_{n_a}(1) \\ \vdots \\ 2\sigma_1^2 v_1^T x v_1(j) + 2\sigma_2^2 v_2^T x v_2(j) + \cdots + 2\sigma_{n_a}^2 v_{n_a}^T x v_{n_a}(j) \\ \vdots \end{bmatrix} - \begin{bmatrix} -2\sigma_i^2 \sum_i f^{\|}(i) v_i(1) \\ \vdots \\ -2\sigma_i^2 \sum_i f^{\|}(i) v_i(j) \\ \vdots \end{bmatrix}. \tag{4.26}$$

Definition 14. *(Upward/Downward Outgoing Gradients) Take a test i, and the functions $N_f^2(x)$ and $T_f^2(x)$ as in* (4.23) *and* (4.25)*, with the formulas of the gradient vectors of these two functions $\nabla N_{f,i}(x), \nabla T_{f,i}(x)$ as in* (4.24) *and* (4.26). *Given the two extreme values $N_f^{min/max}$ and $T_f^{min/max}$ for each test, let us define*

- *the downward outgoing gradients as the set of gradients calculated on the points on the minimum hyperellipsoid*

$$\{-\nabla N_{f,i}(x) \mid N_{f,i}(x) = N_f^{min}\} \quad \text{and} \quad \{-\nabla T_{f,i}(x) \mid T_{f,i}(x) = T_f^{min}\} \tag{4.27}$$

  they point inward to the region of the thick hyperellipsoid.

- *the Upward Outgoing Gradients as the set of negative gradients of points on the maximum hyperellipsoid*

$$\{\nabla N_{f,i}(x) \mid N_{f,i}(x) = N_f^{max}\} \quad \text{and} \quad \{\nabla T_{f,i}(x) \mid T_{f,i}(x) = T_f^{max}\} \tag{4.28}$$

  they point outward the region.

Note that the upward/downward outgoing gradient of function $N_f^2(x)$ (or $T_f^2(x)$) on point x is the normal vector to the tangent plane on the hyperellipsoid on which the point lies. Moreover, these vectors point outward the region defined by Equation (4.18) (and (4.19) respectively). In Figure 4.6, an example of some upward/downward outgoing gradients of function $N_f^2(x)$ is shown.

Theorem 2. *Given N tests with values $I_{f,i}$ and $N_{f,i}$ in the closed intervals $[I_f^{min}, I_f^{max}]$ and $[N_f^{min}, N_f^{max}]$, take the set of all the upward/downward outgoing gradients of functions $N_{f,i}^2(x)$ and $T_{f,i}^2(x)$ calculated in the true solution $\bar{x}_a$, i.e.,*

$$\begin{aligned} &\{\nabla N_{f,i}(\bar{x}_a) \, \textit{for } i = 1, \ldots, N \mid N_{f,i}(\bar{x}_a) = N_f^{max}\} \cup \\ \cup \; &\{\nabla N_{f,i}(\bar{x}_a) \, \textit{for } i = 1, \ldots, N \mid N_{f,i}(\bar{x}_a) = N_f^{min}\} \cup \\ \cup \; &\{\nabla T_{f,i}(\bar{x}_a) \, \textit{for } i = 1, \ldots, N \mid T_{f,i}(\bar{x}_a) = T_f^{max}\} \cup \\ \cup \; &\{\nabla T_{f,i}(\bar{x}_a) \, \textit{for } i = 1, \ldots, N \mid T_{f,i}(\bar{x}_a) = T_f^{min}\}. \end{aligned} \tag{4.29}$$

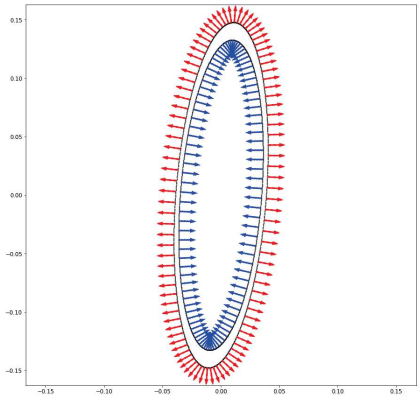


Figure 4.6: In the figure some upward/downward outgoing gradients are shown: the blue internal ones are downward outgoing gradients calculated on points x on the internal ellipsoid with $N_{f,i}(x) = N_f^{min}$, while the external red ones are upward outgoing gradients calculated on points x on the external ellipsoid with $N_{f,i}(x) = N_f^{max}$.

If there is at least one outgoing gradient of this set in each orthant of $\mathbb{R}^{n_a}$, then the intersubsection region I_{zr} of Equation (4.21) *reduces to a point.*

Proof. What we want to show is that given any perturbation δ_x of the real solution $\bar{x}_a$, there exists at least one condition among (4.18) and (4.19) that is not satisfied by the new perturbed point $\bar{x}_a + \delta_x$.

Any sufficiently small perturbation δ_x in an orthant in which lies an upward/downward outgoing gradient (from now on ”Gradient”), determines an increase/decrease in the value of the hyperellipsoid function relative to that Gradient, that makes the relative condition to be unsatisfied.

Hence, if the Gradient in the orthant considered is upward, it satisfies $N_{f,i}(\bar{x}_a) = N_f^{max}$ (or analogously with $T_{f,i}$) and for each perturbation δ_x in the same orthant we obtain

$$N_{f,i}(\bar{x}_a + \delta_x) > N_{f,i}(\bar{x}_a) = N_f^{max}$$

(or analogously with $T_{f,i}$). In the same way, if the Gradient is downward we obtain

$$N_{f,i}(\bar{x}_a + \delta_x) < N_{f,i}(\bar{x}_a) = N_f^{min}$$

(or analogously with $T_{f,i}$).

When in one orthant there are more than one Gradient, it means that more than one condition will be unsatisfied by the perturbed point $\bar{x}_a + \delta_x$ for a sufficiently small δ_x in that orthant. □

4.2.4 Problem Solution

The theory previously presented allows us to build a solution algorithm that can deal with different a-priori information. We will start with the ideal case, i.e., with exact knowledge of I_f and N_f. Then, we generalize to a more practical setting, where we suppose to know an interval that contains the T_f values of all the experiments considered and an interval for the N_f values. Hence, the estimate solution will satisfy Equations (4.18) and (4.19). In this case we describe an algorithm for computing an estimate of the solution, that we will test in subsection 4.2.5 against a toy model.

Exact Knowledge of I_f and N_f

When the information about I_f and N_f is exact, with the minimum amount of experiments indicated in subsection 4.2.3 we can find the unbiased parameter estimate as the intersubsection I_{zr} of the zero-residual sets Z_{r_i} corresponding to each experiment. In principle this could be done also following the proof of Lemma 4, but the computation of the v_i vectors is quite cumbersome. Since this is an ideal case, we solve it by simply imposing the satisfaction of the various N_f and T_f conditions (equation (4.14)) as an optimization problem:

$$\min_x F(x) \quad \text{with} \quad F(x) = \sum_{i=1}^{N} (\|A_{a,i}x\| - N_{f,i})^2 + \sum_{i=1}^{N} (\|A_{a,i}x - f_i^{\|}\| - T_{f,i})^2. \quad (4.30)$$

The solution of this problem is unique when the tests are in a sufficient number and satisfies the conditions of Lemma 4.

This nonlinear least-squares problem can be solved using a general nonlinear optimization algorithm, like Gauss–Newton method or Levenberg–Marquardt [61].

Approximate Knowledge of I_f and N_f

In practice, as already pointed out in subsection 4.2.3, it is more realistic to know the two intervals that contain all the $N_{f,i}$ and $I_{f,i}$ values for each test i. Then, we know that within the region I_{zr} there is also the exact unbiased parameter solution $\bar{x}_a$, that we want at least to approximate. We introduce here an Unbiased Least-Squares (ULS) Algorithm 3 for the computation of this estimate.

In general, the zero-residual region Z_{r_i} of each test contains the true point of the parameters vector, while the estimated iterates with the local optimization usually start from a point outside this region and converge to a point on the boundary of the region.

The ULS estimate can converge to the true solution in two cases:

1. the true solution lies on the border of the region I_{zr} and the estimate reach the border on that point;

Algorithm 3 An Unbiased Least-Squares (ULS) algorithm.

```
Given a number $n_{tests}$ of available tests, indexed with a number between 1 and $n_{tests}$, and two intervals, $\left[I_f^{min}, I_f^{max}\right]$ and $\left[N_f^{min}, N_f^{max}\right]$, containing the $I_f$ and $N_f$ values of all tests.
At each iteration we will consider the tests indexed by the interval $[1, i_t]$; set initially $i_t = n_a$.
while $i_t \leq n_{tests}$ do
    1) compute a solution with zero residual of the problem (4.22) with a nonlinear least-squares optimization algorithm,
    2) estimate the size of the zero-residual region as described below in (4.31),
    3) increment by one the number $i_t$ of tests.
end while
Accept the final solution if the estimated region diameter is sufficiently small.
```

2. the region I_{zr} reduces to a dimension smaller than the required accuracy, or reduces to a point.

The size of the intersubsection set I_{zr}, of the zero-residual regions Z_{r_i}, is estimated in the following way.

Let us define an index, that we call region shrinkage estimate, as follows:

$$\hat{s}(x) = min\{n \mid \sum_{\delta \in \mathcal{P}} \mathbf{1}_{I_{zr}}(x + \mu^{-n}\delta) > 0\}, \tag{4.31}$$

where we used $\mu = 1.5$ in the experiments below, $\mathcal{P} = \{\delta \in \mathbb{R}^{n_a} \mid \delta(i) \in (-1, 0, 1)\ \forall i = 1, \ldots, n_a\}$ and $\mathbf{1}_{I_{zr}}$ is the indicator function of the subset I_{zr} of the set, i.e. the function $\mathbf{1}_{I_{zr}} : I_{zr} \subseteq \mathbb{R}^{n_a} \to \{0, 1\}$ such that

$$\mathbf{1}_{I_{zr}}(x) = \begin{cases} 1 & \text{if } x \in I_{zr}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.32}$$

4.2.5 Numerical Examples

Let us consider a classical application example, the equations of a DC motor with a mechanical load, where the electrical variables are governed by the following ordinary differential equation

$$\begin{cases} L\dot{I}(t) & = -K\omega(t) - RI(t) + V(t) - f_u(t) \\ I(t_0) & = I_0, \end{cases} \tag{4.33}$$

where I is the motor current, ω the motor angular speed, V the applied voltage, and $f_u(t)$ a possible unmodeled component

$$f_u(t) = -m_{err}cos(n_{poles}\theta(t)), \tag{4.34}$$

where n_{poles} is the number of poles of the motor, i.e., the number of windings or magnets , m_{err} the magnitude of the error model and θ the angle, given by the system

$$\begin{cases} \dot{\omega}(t) & = \theta(t) \\ \omega(t_0) & = \omega_0. \end{cases} \tag{4.35}$$

Note that the unknown component f_u of this example can be seen as a difference in the potential that is not described by the approximated model. We are interested in the estimation of parameters $[L, K, R]$. In our test the true values were constant values $[L = 0.0035, K = 0.14, R = 0.53]$.

We suppose to know the measurements of I and ω at equally spaced times $t_0, \ldots, t_{\bar{N}}$ with step h, such that $t_k = t_0 + kh$, and $t_{k+1} = t_k + h$. In Figure 4.7 we see the plots of the motor speed ω and of the unknown component f_u for this experiment.

We compute the approximation of the derivative of the current signal $\hat{\dot{I}}(t_k)$ with the forward finite difference formula of order one

$$\hat{\dot{I}}(t_k) = \frac{I(t_k) - I(t_{k-1})}{h}, \qquad \text{for} \quad t_k = t_1, \ldots, t_{\bar{N}}$$

with a step $h = 4 \times 10^{-4}$. The applied voltage is held constant to the value $V(t) = 30.0$.

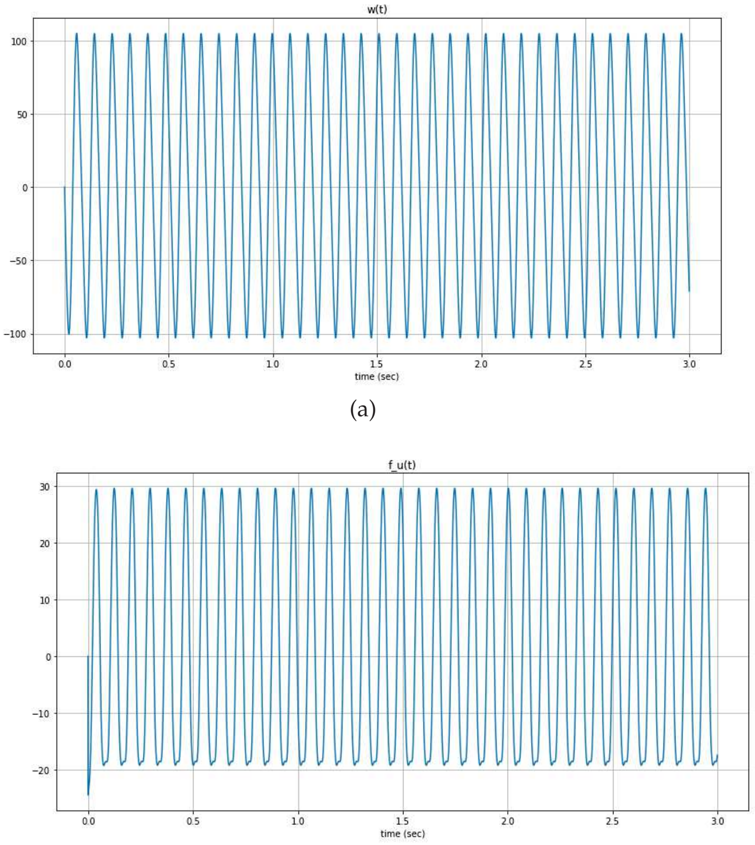
To obtain a more accurate estimate, or to allow the possibility of using higher step size values h, finite differences of higher order can be used, for example the fourth order difference formula

$$\hat{\dot{I}}(t_k) = \frac{I(t_k - 2h) - 8I(t_k - h) + 8I(t_k + h) - I(t_k + 2h)}{12h}, \qquad \text{for} \quad t_k = t_2, \ldots, t_{\bar{N}-2}.$$

With the choice of the finite difference formula, we obtain the discretized equations

$$L\hat{\dot{I}}(t_k) = -K\omega(t_k) - RI(t_k) + V(t_k) - f_u(t_k), \qquad \text{for} \quad t_k = t_1, \ldots, t_{\bar{N}}. \tag{4.36}$$

We will show a possible implementation of the method explained in the previous subsections, and the results we get with this toy-model example. The comparison is made against the standard least-squares. In particular, we will show that when the information about I_f and N_f is exact, we have an exact removal of the bias. In case this information is only approximate, which is common in a real application, we will show how the bias asymptotically disappears when the number of experiments increases.



(a)

(b)

Figure 4.7: The plots of **(a)** $\omega(t)$ and **(b)** $f_u(t)$ in the experiment.

We build each test taking the Equation (4.36) for n samples in the range $t_1, \ldots, t_{\bar{N}}$, obtaining the linear system

$$\begin{bmatrix} \hat{I}(t_k) & \omega(t_k) & I(t_k) \\ \hat{I}(t_{k+1}) & \omega(t_{k+1}) & I(t_{k+1}) \\ \vdots & \vdots & \vdots \\ \hat{I}(t_{k+n}) & \omega(t_{k+n}) & I(t_{k+n}) \end{bmatrix} \begin{bmatrix} L \\ K \\ R \end{bmatrix} + \begin{bmatrix} f_u(t_k) \\ f_u(t_{k+1}) \\ \vdots \\ f_u(t_{k+n}) \end{bmatrix} = \begin{bmatrix} V(t_k) \\ V(t_{k+1}) \\ \vdots \\ V(t_{k+n}) \end{bmatrix} \tag{4.37}$$

so that the first matrix in the equation is $A_a \in \mathbb{R}^{n \times n_a}$ with $n_a = 3$, the number of parameters to be estimated.

To measure the estimation relative error $\hat{e}_{rel}$ we will use the following formula, where $\hat{x}_a$ is the parameter estimate:

$$\hat{e}_{rel} = \frac{1}{n_a} \sum_{i=1}^{n_a} \frac{\|\hat{x}_a(i) - \bar{x}_a(i)\|_2}{\|\bar{x}_a(i)\|_2}. \tag{4.38}$$

Note that the tests that we built in the numerical experiments below are simply small chunks of consecutive data, taken from one single simulation for each experiment.

The results have been obtained with a Python code developed by the authors, using `NumPy` for linear algebra computations and `scipy.optimize` for the nonlinear least-squares optimization.

Exact Knowledge of I_f and N_f

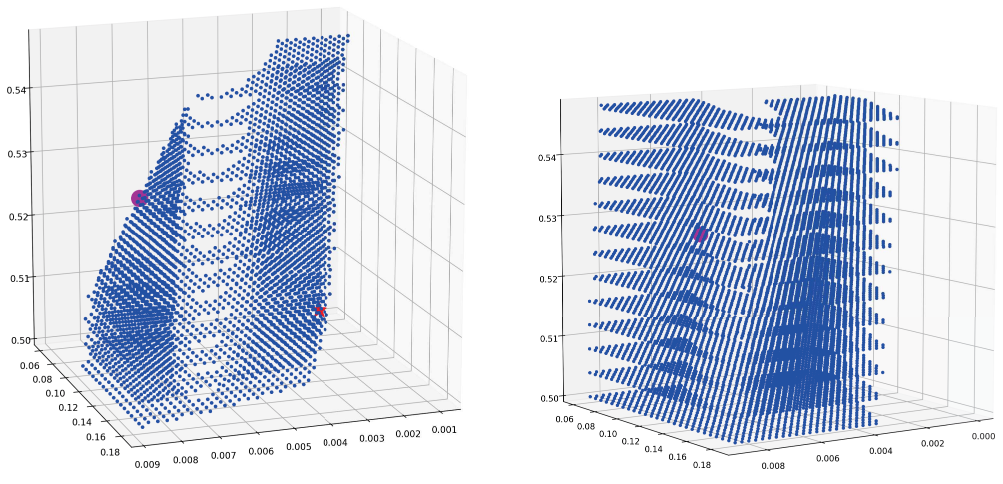
As analyzed in subsection 4.2.4, the solution of the minimization problem (4.30) is computed with a local optimization algorithm.

Here the obtained results show an error $\hat{e}_{rel}$ with an order of magnitude of 10^{-7} in every test we made. Note that it is also possible to construct geometrically the solution, with exact results.

Approximate Knowledge of I_f and N_f

When I_f and N_f are known only approximately, i.e., we know only an interval that contains all the I_f values and an interval that contains all the N_f values, we lose the unique intersubsection of Lemma 4, that would require only n_a tests. Moreover, with a finite number of tests we cannot guarantee in general to satisfy the exact hypotheses of Theorem 2. As a consequence, various issues open up. Let's start by showing in Figure 4.8 that when all the four conditions of (4.15) hold with equality, the true solution lies on the boundary of the region I_{zr} as already mentioned in 4.2.3. If this happens, then with the conditions of Theorem 2 on the upward/downward outgoing gradients, the region I_{zr} is a point. When all the four conditions of (4.15) hold with strict inequalities, the true solution lies inside the region I_{zr} (Figure 4.8b). From a theoretical point of view this distinction has a big importance, since it means that the zero-residual region can or cannot be reduced to a single point. From a practical point of view it becomes less important, for the moment, since we cannot guarantee that the available tests will reduce I_{zr} exactly to a single point and we will arrive most of the times to an approximate estimate. This can be more or less accurate, but this depends on the specific application, and this is out of the scope of the present work.

To be more precise, when the conditions of Theorem 2 are not satisfied, there is an entire region of the parameters space which satisfies exactly problem (4.30), but only one point of this region is the true solution $\bar{x}_a$. As more tests are added and intersected together, the zero-residual region I_{zr} tends to reduce, simply because it must satisfy an increasing number of inequalities. In Figure 4.9 we can see four iterations taken from an example, precisely with 3, 5, 9 and 20 tests intersected and



(a)

(b)

Figure 4.8: Two examples of (zero-residual) intersubsection regions $I_{zr} \subset \mathbb{R}^3$ with different location of the true solution: inside the region or on its border. For graphical reasons the region has been discretized and the dots are the grid nodes; the bigger ball (thick point) is the true solution. **(a)** The true solution (ball) is on the border of I_{zr}; **(b)** The true solution (ball) is internal to I_{zr}.

$m_{err} = 19$. With only three tests (Figure 4.9a), there is a big region I_{zr} (described by the mesh of small dots), and here we see that the true solution (thick point) and the current estimate (star) stay on opposite sides of the region, as accidentally happens. With five tests (Figure 4.9a) the region has shrunk considerably and the estimate is reaching the boundary (in the plot it is still half-way), and even more with nine tests (Figure 4.9c). The convergence arrives here before the region collapses to a single point, because accidentally the estimate has approached the region boundary at the same point where the true solution is located.

In general, the zero-residual region Z_{r_i} (4.20) of each test contains the true solution, while the estimate arrives from outside the region and stops when it bumps the border of the intersubsection region I_{zr} (4.21). For this reason we have convergence when the region that contains the true solution is reduced to a single point, and the current estimate $\hat{x}_a$ does not lie in a disconnected sub-region of I_{zr} different from the one in which the true solution lies. Figure 4.10 shows an example of an intersubsection region I_{zr} which is the union of two closed disconnected regions: this case creates a local minimum in problem (4.30).

In Figure 4.11 we see the differences $N_f^{max} - N_f^{min}$ and $T_f^{max} - T_f^{min}$ vs m_{err}. The differences are bigger for higher values of the model error. It seems that this is the cause of a more frequent creation of local minima.

Figure 4.12 synthesizes the main results that we have experienced with this new approach. Globally it shows a great reduction of the bias contained in the standard least-squares estimates; indeed, we had to use the logarithmic scale to enhance the differences in the behaviour of the proposed method while varying m_{err}. In particular,

- with considerable levels of modelling error, let us say m_{err} between 2 and 12, the parameter estimation error $\hat{e}_{rel}$ is at least one order of magnitude smaller that that of least-squares; this is accompanied by high levels of shrinkage of the zero-residual region (Figure 4.12b);

- with higher levels of m_{err}, we see a low shrinkage of the zero-residual region and consequently an estimate whose error is highly oscillating, depending on where the optimization algorithm has brought it to get in contact with the zero-residual region;

- at $m_{err} = 18$ we see the presence of a local minimum, due to the falling to pieces of the zero-residual region as in Figure 4.10: the shrinkage at the true solution is estimated to be very high, while at the estimated solution it is quite low, since it is attached to a disconnected, wider sub-region.

- the shrinking of the zero-residual region is related to the distribution of the outgoing gradients, as stated by Theorem 2: in Figure 4.12d we see that in the experiment with $m_{err} = 18$ they occupy only three of eight orthants, while in the best results of the other experiments the gradients distribute themselves in almost all orthants (not shown).

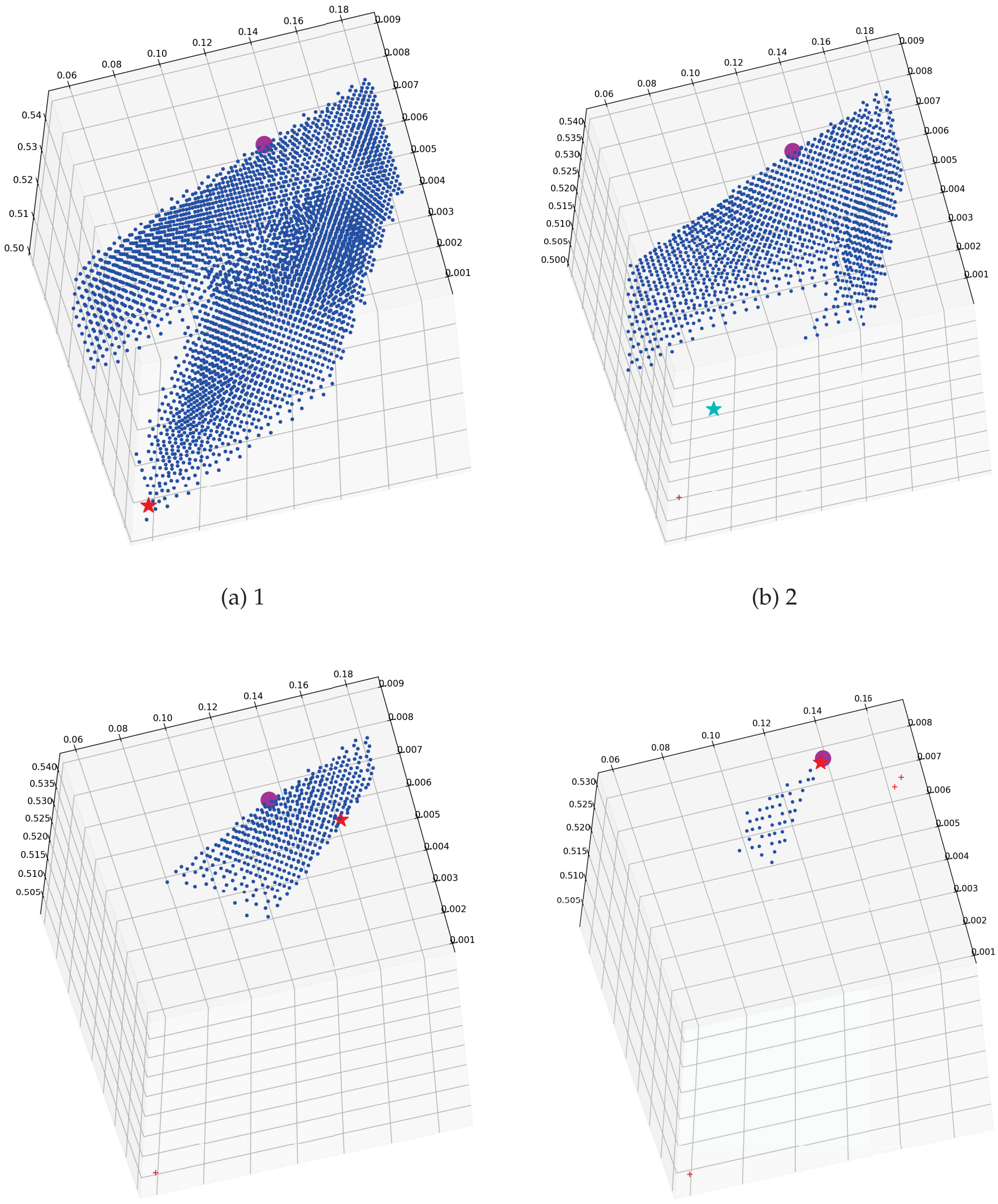
It is evident from these results that for lower values of modelling error m_{err}, it is much easier to produce tests that reduce the zero-residual region to a quite small interval of R^{n_a}, while for high values of m_{err} it is much more difficult and the region I_{zr} can even fall to pieces, thus creating local minima. It is also evident that a simple estimate of the I_{zr} region size, like (4.31), can reliably assess the quality of the estimate produced by the approach here proposed, as summarized in Figure 4.12c.

4.2.6 Future work

We have analyzed the bias commonly arising in parameter estimation problems where the model is lacking some deterministic part of the system. This result is useful in applications where an accurate estimation of parameters is important, e.g., in physical (grey-box) modelling typically arising in the model-based design of multi-physical systems, see e.g., the motivations that the authors did experience in the design of digital twins of controlled systems [6, 5, 4] for virtual prototyping, among an actually huge literature.

At this point, the method should be tested in a variety of applications, since the ULS approach here proposed is not applicable black-box as Least-Squares are. Indeed, it requires some additional a-priori information. Moreover, since the computational complexity of the method here presented is relevant, efficient computational methods must be considered and will be a major issue in future investigations.

Another aspect that is even worth to deepen is also the possibility to design tests that contribute optimally to the reduction of the zero-residual region.



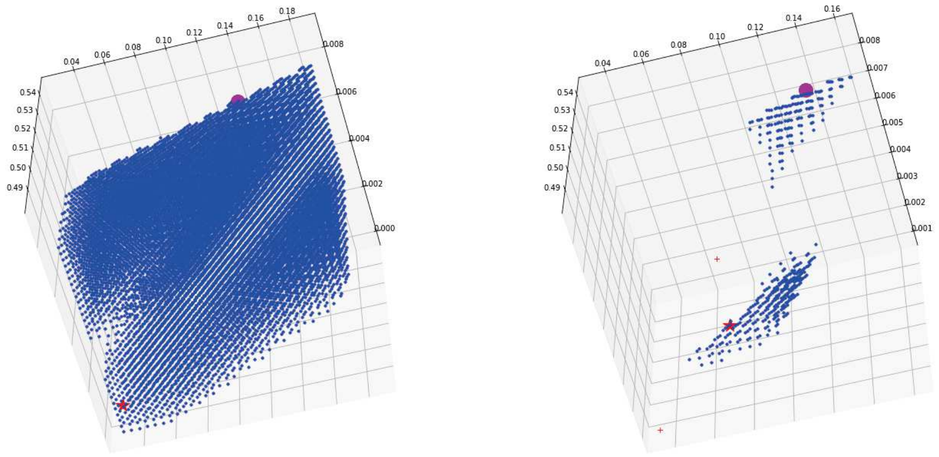
(a) 1

(b) 2

(c) 3

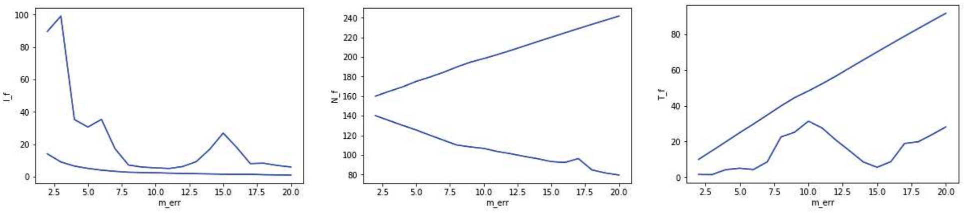
(d) 4

Figure 4.9: The intersubsection region $I_{zr} \subset \mathbb{R}^3$ at different number of tests involved. For graphical reasons the region has been discretized and the dots are the grid nodes; the bigger ball is the true solution and the star is the current estimate in the experiment. **(a)** 3 tests; **(b)** 5 tests; **(c)** 9 tests; **(d)** 20 tests.



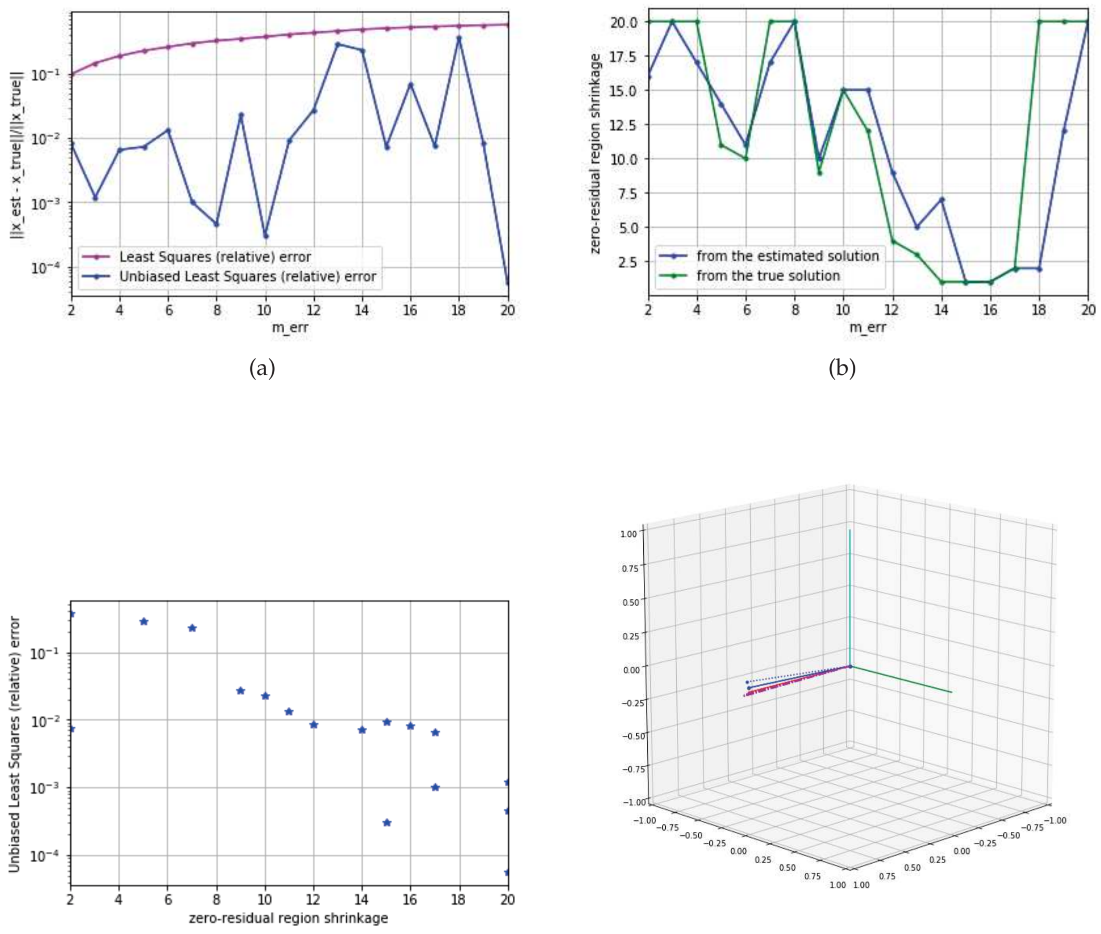
(a) (b)

Figure 4.10: The intersubsection region $I_{zr} \subset \mathbb{R}^3$ at different number of tests involved. On the left a few tests have created a single connected region while, on the right, adding more tests have splitted it into two subregions. For graphical reasons the region has been discretized and the dots are the grid nodes; the bigger ball is the true solution and the star is the current estimate in the experiment. **(a)** A (portion of a) connected region I_{zr}; **(b)** A region I_{zr} split into two not connected sub regions.



(a) (b) (c)

Figure 4.11: The three plots show the values assumed by the extreme values (4.15) as a function of m_{err}. **(a)** $\{I_f^{min}, I_f^{max}\}$ *vs* m_{err}; **(b)** $\{N_f^{min}, N_f^{max}\}$ *vs* m_{err}; **(c)** $\{T_f^{min}, T_f^{max}\}$ *vs* m_{err}.



(a)

(b)

(c)

(d)

Figure 4.12: The plots summarize the results obtained by the ULS approach to parameter estimation no the model problem explained at the beginning of this subsection. **(a)** The relative estimation error (4.38) vs m_{err}; **(b)** The I_{zr} region shrinkage estimate (4.31) vs m_{err}; **(c)** The relative estimation error (4.38) vs the estimate of the I_{zr} region shrinkage, considering the experiments with $m_{err} \in [2, 20]$; **(d)** A three dimensional view of the Outgoing Gradients at the last iteration of the experiment with $m_{err} = 18$.

Part II

Model-Based Denoising: *when the model is known*

Chapter 5

Denoising of I/O signals of DLTI Models

5.1 Introduction

It is well known that experimental measurements generally contain noise that corrupts the (usually deterministic) real quantities to be measured. Many denoising algorithms have been proposed and analyzed in the literature, and differ from each other for the hypothesis on signal and noise properties and for the prior knowledge required about them [12]. Comparisons on these methods have been carried out, for example in [53] for signal denoising and [55] for image denoising, and different kind of noisy data and functions for the measure of denoising results have been considered. Some of the most important techniques are: moving average filters, linear filters, nonlinear filters, Fourier decomposition, Wavelet decomposition, Total Variation Minimization, Neural Networks (e.g. [17]).

Denoising methods are mainly studied for individual signals, isolated from the system from which they arise. When more variables are measured from the same physical dynamical system, the noise not only corrupts each single quantity but also the causal (input-output) relation between the data. Therefore, applying the methods previously mentioned separately to each noisy signal is not sufficient to recover the correct input-output relations: this work aims at introducing a new algorithm that satisfies this requirement. In particular, we will consider the simultaneous denoising of input-output signals from a physical dynamical system, that we suppose can be modeled by a discrete linear time-invariant (DLTI) system. Taking into account the system relations we will obtain a linear denoising method that we therefore classify as *"DLTI model-based denoising"*.

The availability of data with a precise input-output relation is of fundamental importance in real-life applications, e.g. when quantities of interest derived from input/output variables (like the mechanical/electric power) are required, or in the resolution of computational inverse problems. In such a context the use of a mathematical model of the system allows to recover unknown (and not measurable) quantities, for example model parameters and boundary conditions for distributed parameters systems [19], [56]. In particular, good input-output algebraic relations in the data are required to estimate continuous system parameters from discrete measures [4], [57].

The term *Model-Based* in the image and signal processing literature, is often used to refer not only to physical models, but also to (abstract) mathematical model structures. This is the case of Candy, who develops in [12] a general model-based approach to signal processing for different kinds of problems, using not only *physical-based* models but also ”black-box” ones.

In this work, published in [22], we focus on model-based denoising where models arise from the physical equations that describe the system. Studies in this direction are, for example [78], where a ”Physically Consistent Denoising” is described and an algorithm for the denoising and ”missing data recovery” of 2D physically plausible vector fields is introduced, and [67], where a Model-Based Image Reconstruction that satisfies the physical Cahn-Hilliard equation is presented.

We will consider signals corrupted by additive noise, also called in literature *”Error-in-variables (EIV) framework”* [28].

The studies in [20] and [58], [59], where the *”noisy I/O problem”* is introduced, are the nearest to the approach of this work. In this setting, the matrices of a DLTI system are supposed known, and the aim is the denoising of input-output data. The method is based on the a-priori knowledge of mean and covariances of the noise signals added to the inputs and the outputs. Unlike this approach, we are interested in the more general case in which these quantities are unknown, as it happens in many applications [28].

In the literature, a more general topic has also been investigated, in the case of additive noise with known properties: the denoising of input-output data and simultaneous identification of the DLTI system that links them. This problem shows a certain affinity with the Total Least Squares (TLS) problem, see e.g. [70], [69], [74]. Moreover, the problem of denoising of the input-output data used for the identification of the model that relates them, has been studied with optimization methods, more precisely the minimization of the nuclear-norm of a matrix generated by data and model predictions, see e.g. [51].

The approach presented here differs from the ones in the literature because it assumes a DLTI physical model to be available and aims at denoising the input-output signals, of which only noisy measurements are available with unknown values of means and variances, i.e. we generalize the problem of [58] and [59] in the case in which the statistical properties of the input and output noises are unknown. This comes at the price of a bigger computational effort that makes its applicability to real-time not straightforward, since we use global optimization and smoothing.

It is worth noticing that this generalization is important in applications; for example, when these properties are constant but good estimates of them are not available, or when the sensors (or estimators) generate noises with non constant properties, which variability cannot be described a-priori by a model.

We will show that in these cases the approach of [58] is not optimal and we will see in more details which are the additional conditions required when the estimates of noise means and covariances are unknown. In this work we propose a method based on the resolution of a linear system generated by the model equations (that we want to be satisfied precisely), and four other parameterized regularization terms (that are not required to be satisfied exactly). Hence, the problem is reduced to the choice of the regularization parameters of a multi-parametric linear system. With this approach, several methods have been presented in the literature and are divided in two categories: methods that use the noise variance value and, viceversa, *heuristic* methods also called *noise level free rules*. If the noise covariance is known (at least approximatively), the parameters can be determined with the discrepancy principle, and in the multi-parameter case with its generalization, introducing the *discrepancy hypersurface* ([24], [54]). The optimal parameters are found on that hypersurface through the optimization of a function of the parameters, for example the norm of the solution can be maximized [24] or the quasi-optimality criterion (introduced in [73]) can be considered [21].

On the other side, *heuristic* methods possess bad convergence properties. In fact, they don't converge in the "worst case scenario", i.e. it is not true that the regularized solution converges to the true one for every noise realization with noise level that tends to zero. Although this result, called *Bakushinskii veto* [3], convergence results have been demonstrated under appropriate conditions, that are usually satisfied in real situations (see [45] and references therein). Among heuristic methods for multi-parameter regularization, we can find the generalizations of the L-curve [7], of the Generalized Cross Validation (GCV) [10], a balancing principle [39] and parameter learning for denoising [47],[37].

Since we suppose the noise means and variances to be unknown, we rely upon an additional criterion based on other statistical properties of the noise. More precisely, we will use the Normalized Cumulative Periodogram (NCP), also known as Bartlett test, to measure the whiteness of the estimated noises, a well known method for the one-parameter regularization choice ([34], [32]).

The organization of this Chapter is as follows. In Section 5.2 we introduce the Kalman filter, used for the estimation of the DLTI state, but also for the denoising of ouput measurements. This is important for us because we will see that a reformulation of the input/output denoising problem with known covariances can be reduced to the Kalman filter problem. In Section 5.3 we will define the problem and the proposed approach to solve it; in Section 5.4 some general issues are discussed. In Section 5.5 an algorithm for the selection of parameters will be introduced and in Section 5.6 some numerical results will be shown. Concluding remarks and possible future work are provided in Section 5.7.

5.2 Kalman filter as output measurement denoising

With the probabilistic concepts recalled in Section 1.1.1, in particular the definitions of Gaussian and White Gaussian noises, we can introduce noise in the DLTI model (1.12). The most common formulation is the one in which some error in the model and in the output measure y is added, while the input measure u is supposed exact, with no noise.

Definition 15 (DLTI State-Space Models with process and measure noise)**.**

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + v(k) \\ y(k) = Cx(k) + Du(k) + w(k) \end{cases} \quad \textit{with} \quad \begin{cases} v(k) \sim \mathcal{N}(0, R_{vv}) \\ w(k) \sim \mathcal{N}(0, R_{ww}) \end{cases} \tag{5.1}$$

where $v(k)$ is the process or model noise and $w(k)$ is the measurement noise, and both $v(k)$ and $w(k)$ are White Gaussian noises.

The problem we want to solve in this Section is to recover a state estimate of (5.1) using both the information of the model and the measurements. Since models are inaccurate and sensors have errors, we must take into consideration the noise values. The Kalman Filter gives us the right way (in a sense we will define) to weight both the information, i.e. it gives us the weights for the combination of the two information. It is a very famous algorithm since its successful use in the navigation systems for the Apollo mission. From its discover, the applications for which it has been used cover different fields like signal processing, voice recognition, video stabilization, and automotive, control, global positioning, computer vision and lots more. Moreover, various generalizations of this method have been studied and are still open problems.

The term *filtering* is referred to the fact that it is an online algorithm, i.e. it works one discrete instant at a time, in contrast to the offline procedures, which are called *smoothing*. More precisely, at each time instant k we have an estimate of the state at that instant, calculated with the model, and the measurements up to the previous time instant. In Figure 5.1 the prediction-correction procedure is shown.

Let us define the Kalman Problem more precisely [77, 12].

Problem 2 (Kalman Filter Problem)**.** *We are given the signal-generation model*

$$\begin{cases} x(k+1) & = A(k)x(k) + B(k)u(k) + w(k) \\ y(k) & = C(k)x(k) + D(k)u(k) + v(k) \end{cases} \tag{5.2}$$

with the process noise $w(k)$ and measurement noise $v(k)$ assumed to be zero mean white-noise sequences with joint covariance matrix

$$E\left[\begin{bmatrix} v(k) \\ w(k) \end{bmatrix} \begin{bmatrix} v(j)^T & w(j)^T \end{bmatrix}^T\right] = \begin{bmatrix} R_{vv}(k) & R_{wv}(k)^T \\ R_{wv}(k) & R_{ww}(k) \end{bmatrix} \Delta(k-j) \geq 0$$

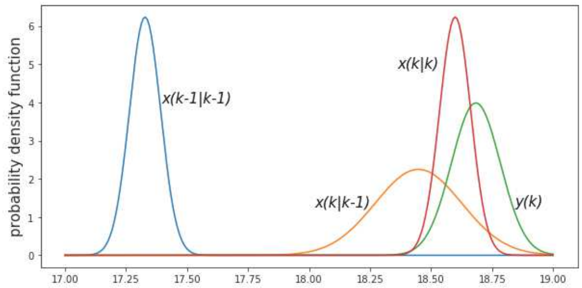


Figure 5.1: Kalman Filter iteration of predictor-corrector form, where $x(k|k-1)$ is the estimate of state x at time k with known measurements up to time $k-1$, and $x(k|k)$ is the estimate of state x at time k with known measurements up to time k.

with $R_{vv}(k) > 0$ and where $\Delta(k)$ is the unit pulse. At time instant $k-1$, we have an estimate of $x(k)$, which is denoted by $\hat{x}(k|k-1)$ with properties

$$E[x(k)] = E[\hat{x}(k|k-1)],$$

$$E[(x(k) - \hat{x}(k|k-1))(x(k) - \hat{x}(k|k-1))^T] = P(k|k-1) \geq 0.$$

This estimate is uncorrelated with the noise $w(k)$ and $v(k)$. The problem is to determine a linear estimate of $x(k)$ and $x(k+1)$ based on the given data $u(k)$, $y(k)$, and $\hat{x}(k|k-1)$, such that both estimates are minimum variance unbiased estimates*:*
minimum variance

$$\begin{cases} \min \; E[(\hat{x}(k) - \hat{x}(k|k))(x(k) - \hat{x}(k|k))^T], \\ \min \; E[(x(k+1) - \hat{x}(k+1|k))(x(k+1) - \hat{x}(k+1|k))^T], \end{cases} \tag{5.3}$$

unbiased

$$E[\hat{x}(k|k)] = E[x(k)], \quad E[\hat{x}(k+1|k)] = E[x(k+1)]. \tag{5.4}$$

Note that, with the condition of the mean, for the unbiased estimate, asking for the minimum variance of the error is equivalent to asking the minimum cross-correlation of the error, and the minimum variance of the solution. The conditions on means and variances of the errors give the right Kalman weights for the combination of the measurements.

There are lots of formulations and derivation of this result, we will present here the “conventional Kalman filter”.

Predictor-Corrector form:

$$\begin{cases} \hat{x}(k|k-1) & = A(k-1)\hat{x}(k-1|k-1) + B(k-1)u(k-1) \\ \tilde{P}(k|k-1) & = A(k-1)\tilde{P}P(k-1|k-1)A'(k-1) + R_{ww}(k-1) \end{cases} \qquad \textbf{predictor}$$

$$\begin{cases} e(k) & = y(k) - \hat{y}(k|k-1) = y(k) - C(k)\hat{x}(k|k-1) \\ R_{ee}(k) & = C(k)\tilde{P}(k|k-1)C'(k) + Rvv(k) \end{cases}$$

$$K(k) = \tilde{P}(k|k-1)C'(k)R_{ee}^{-1}(k)$$

$$\begin{cases} \hat{x}(k|k) & = \hat{x}(k|k-1) + \mathbf{K}(k)e(k) \\ \tilde{P}(k|k) & = [I - K(k)C(k)]\tilde{P}(k|k-1) \end{cases} \qquad \textbf{corrector}$$

$$\hat{x}(0|0), \tilde{P}(0|0)$$

where $K(k)$ is called *Kalman gain or weight*, that gives us exactly that correct combination of the two information we started with. The equations of the algorithm descend directly from the conditions on means (5.3) and covariances (5.4).

Note that the Kalman filter can be seen not only as a state estimation algorithm, but also as an output denoising algorithm. We will see in the following Sections the problem of denoising both input and output for particular DLTI models and how it is related to the Kalman filter problem.

5.3 DLTI Model-Based denoising

Consider a linear dynamical system with deterministic and measurable input and output, described by a discrete linear time-invariant (DLTI) system [40] in state-space form:

$$\begin{cases} x(k+1) & = A\,x(k) + B\,u(k) \\ y(k) & = C\,x(k) + D\,u(k). \end{cases} \qquad \text{for } k = 0, \ldots, N-1, \tag{5.5}$$

where $x(k) \in \mathbb{R}^{n_x \times 1}$, $u(k) \in \mathbb{R}^{n_u \times 1}$, $y(k) \in \mathbb{R}^{n_y \times 1}$, and A, B, C, D are the system matrices, $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$, $D \in \mathbb{R}^{n_y \times n_u}$.

The following definitions will be useful

$$\begin{aligned} u &:= [u^T(0) \cdots u^T(N-1)]^T && \in \mathbb{R}^{Nn_u \times 1} \\ y &:= [y^T(0) \cdots y^T(N-1)]^T && \in \mathbb{R}^{Nn_y \times 1} \\ x &:= [x^T(0) \cdots x^T(N)]^T && \in \mathbb{R}^{(N+1)n_x \times 1} \end{aligned} \tag{5.6}$$

and the superscript notation will be used to denote the components of the signals, i.e. for example $u(k) = [u^1(k), \ldots, u^{n_u}(k)]^T$ and $u^i = [u^i(0) \cdots u^i(N-1)]^T$.

In this work we will consider the common choice of $D = 0_{n_y \times n_u}$ and a more restrictive, but often acceptable, condition that the matrix C be invertible so that the system reduces to

$$y(k+1) = CAC^{-1}\,y(k) + CB\,u(k), \qquad k = 0, \ldots, N-2. \tag{5.7}$$

We suppose to know only noisy measurements of input and output signals u_e, y_e (defined in the same way as (5.6)), corrupted with white noise

$$\begin{cases} u_e(k) = u(k) + e_u(k) \\ y_e(k) = y(k) + e_y(k) \end{cases} \tag{5.8}$$

where the error vectors relative to each component of input and output signals e_{u^i}, e_{y^i} are White Gaussian noise vectors that satisfy

$$\begin{cases} \mathbb{E}\{e_{u^i}\} = 0 \\ \mathbb{E}\{e_{u^i} e_{u^i}^T\} = \sigma_{eu^i}^2 I \end{cases} \quad \text{for } i = 1, \ldots, n_u, \qquad \begin{cases} \mathbb{E}\{e_{y^i}\} = 0 \\ \mathbb{E}\{e_{y^i} e_{y^i}^T\} = \sigma_{ey^i}^2 I \end{cases} \quad \text{for } i = 1, \ldots, n_y. \tag{5.9}$$

In the following we will consider also a more general hypothesis in which e_u and e_y are biased noises with mean values different from zero, i.e. we will consider Gaussian noises with biases $\mu_{eu^i} \neq 0, \mu_{ey^i} \neq 0,\ \forall i$ (hence not White noises):

$$\begin{cases} \mathbb{E}\{e_{u^i}\} = \mu_{eu^i} \\ \mathbb{E}\{e_{u^i} e_{u^i}^T\} = \sigma_{eu^i}^2 I \end{cases} \quad \text{for } i = 1, \ldots, n_u, \qquad \begin{cases} \mathbb{E}\{e_{y^i}\} = \mu_{ey^i} \\ \mathbb{E}\{e_{y^i} e_{y^i}^T\} = \sigma_{ey^i}^2 I \end{cases} \quad \text{for } i = 1, \ldots, n_y. \tag{5.10}$$

We observe that the noisy measurements u_e, y_e do not satisfy the model constraints. Hence, any other quantity calculated from them will be corrupted, even nonlinearly, by input/output noise. This fact motivates the need of an additional requirement for the denoising algorithm: it must be model-based, i.e. the denoised data must satisfy the deterministic model.

Our approach for the model-based denoising problem is the following: we impose the model constraints, together with other additional conditions on input-output data, as described in the following subsections.

5.3.1 Problem formulation

Model-based constraining We define the fundamental constraint of model-based denoising in the following problem, that by itself is ill-posed and has infinitely many solutions, as will be shown later:

Problem 3 (”model-based constraining”)**.** *Given the vectors $y_e(k), u_e(k)$ with $k = 0, 1, \ldots, N-1$, measures of signals with white* (5.9) *or Gaussian* (5.10)*, additive noise* (5.8)*, determine vectors $\hat{y}(k), \hat{u}(k)$ that satisfy the model* (5.7)*.*

Following [59], we can impose model equations (5.7) writing them in matrix form

$$\begin{bmatrix} \bar{A} & \bar{B} \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = 0$$

where the matrices $\bar{A} \in \mathbb{R}^{(N-1)n_y \times Nn_y}$ and $\bar{B} \in \mathbb{R}^{(N-1)n_y \times Nn_u}$ are the following

$$\bar{A} = \begin{bmatrix} CAC^{-1} & -I_{n_u} & 0 & \dots & 0 \\ 0 & CAC^{-1} & -I_{n_u} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & CAC^{-1} & -I_{n_u} \end{bmatrix} \qquad \bar{B} = \begin{bmatrix} CB & & \dots & 0 \\ 0 & CB & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & CB \end{bmatrix}.$$

We write the system with respect to noise variables, recalling (5.8), and substituting u with $u_e - e_u$ and y with $y_e - e_y$, so that we obtain the system:

$$\begin{bmatrix} \bar{A} & \bar{B} \end{bmatrix} \begin{bmatrix} e_y \\ e_u \end{bmatrix} = d \tag{5.11}$$

where

$$d = \begin{bmatrix} \bar{A} & \bar{B} \end{bmatrix} \begin{bmatrix} y_e \\ u_e \end{bmatrix}.$$

Calling

$$G = \begin{bmatrix} \bar{A} & \bar{B} \end{bmatrix} \in \mathbb{R}^{(N-1)n_y \times N(n_y+n_u)} \qquad \text{and} \qquad z = \begin{bmatrix} e_y \\ e_u \end{bmatrix} \in \mathbb{R}^{N(n_y+n_u) \times 1}$$

we obtain the linear system

$$Gz = d.$$

Since this system is underdetermined, and infinitely many solutions $\hat{z}$ exist, we can consider the least-squares solution

$$\begin{aligned} z_{ls} = \ & \underset{z}{\text{argmin}} \ \|z\|_2^2 \quad = G^T(GG^T)^{-1}d \\ & \text{s.t.} \quad Gz = d \end{aligned}$$

that is the minimum norm solution. It is worth noticing that, if the variances of the noise signals e_u and e_y are unitary and their covariance as well as the means μ_{eu^i}, μ_{ey^i} $\forall i$ are zero, then z_{ls} is the solution of the problem given by equation (5) of [59] (reported in subsection 5.3.3, Problem 4), therein demonstrated to be optimal. We will deal with the case of unknown and not necessarily unitary variances and not necessarily zero means.

White noise and bias separation Since we are interested not only in the case of white noise (5.9), but also Gaussian one (5.10), we reformulate the problem separating the noisy signals in the sum of a zero mean signal (that we will call $\tilde{e}_{u,y}(k)$) and its mean ($\bar{e}_{u,y}$):

$$\begin{cases} \hat{e}_y(k) = \tilde{e}_y(k) + \bar{e}_y, \\ \hat{e}_u(k) = \tilde{e}_u(k) + \bar{e}_u. \end{cases} \tag{5.12}$$

Because of this separation we must add to the system the zero mean conditions on the vectors $\tilde{e}_{u^i}$ and $\tilde{e}_{y^i}$ for every i :

$$\sum_{k=0}^{N-1} \tilde{e}_{u^i}(k) = 0, \quad \text{for } i = 1, \ldots, n_u, \qquad \sum_{k=0}^{N-1} \tilde{e}_{y^i}(k) = 0, \quad \text{for } i = 1, \ldots, n_y.$$

We call the new unknown vector

$$\tilde{z} = [\tilde{e}_u, \tilde{e}_y, \bar{e}_u, \bar{e}_y]^T$$

and consider the new system with respect to the new variables and call it with the same notation

$$G\tilde{z} = d \tag{5.13}$$

where now $G \in \mathbb{R}^{(Nn_y+n_u)\times(N+1)(n_y+n_u)}$ and $d \in \mathbb{R}^{(Nn_y+n_u)\times 1}$.

5.3.2 Regularization

The considered system (5.13) is underdetermined, and the minimum norm solution is not able to determine a sufficiently good denoising, as we will see in subsection 5.3.3. In fact, the optimal solution is the weighted least squares solution with weights that depends on the covariance values, as explained in [59] and recalled in subsection 5.3.3. Therefore, since we are assuming unknown values of variances, and bad estimates of these values can produce bad denoised signals, the minimum norm solution (obtained assuming unitary variances) is not optimal.

For these reasons, we must consider additional conditions that regularize the problem, in order to obtain a good, unique denoised solution. Since there are no other known conditions to be minimized exactly, we must add adequately weighted regularization terms.

Single Signal Denoising One of the main signal denoising methods is based on the Tikhonov regularization ([32], [43], [9]): given a signal $u_e \in \mathbb{R}^{N\times 1}$ corrupted with white noise, the denoised signal $\hat{u} \in \mathbb{R}^{N\times 1}$ is given by

$$\hat{u} = \min_u \|u - u_e\|_2^2 + \|\lambda L u\|_2^2 \tag{5.14}$$

for a certain value of the regularization parameter λ, with L a regularization matrix that represent the discrete approximation of a derivative operator. In the case of the second derivative, the matrix L is

$$L = \begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \end{bmatrix} \in \mathbb{R}^{(N-2)\times N}$$

and the obtained method is called Hodrick-Prescott filter [36] in economics and statistics, and is used for trend estimation of time series.

This technique is used for many applications, e.g. for the denoising of ECG signals [68]. In that work in particular, it is shown how the parameter λ is linked with the value of the variance of the noise signal: if its value is known it is possible to determine the optimal parameter λ with eq. (30) of that paper.

As pointed out in [44], the solution of problem (5.14) is smooth with respect to the regularization parameter λ that varies in the interval $[0, \infty)$:

- for λ that tends to zero, the solution tends to the measured signal u_e,
- for λ that tends to infinity, the solution converges to the affine subspace that best approximate the time series ("ba"= best affine):

$$u^{ba}(t) = \alpha^{ba} + \beta^{ba} t.$$

Now we generalize the problem above to multiple signals with $u \in \mathbb{R}^{N n_u \times 1}$ defined as in subsection 5.3:

$$u := [u^T(0) \cdots u^T(N-1)]^T \quad \in \mathbb{R}^{N n_u \times 1}.$$

In this case the matrix L becomes

$$L = \begin{bmatrix} I_{n_u} & -2\,I_{n_u} & I_{n_u} & & & & \\ & I_{n_u} & -2\,I_{n_u} & I_{n_u} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & I_{n_u} & -2\,I_{n_u} & I_{n_u} & \\ & & & & I_{n_u} & -2\,I_{n_u} & I_{n_u} \end{bmatrix} \in \mathbb{R}^{(N-2)\,n_u \times N\,n_u}$$

and we must consider n_u parameters $\lambda_{u_1}, \ldots, \lambda_{u_{n_u}}$, one for each input variable, so that the minimization problem (5.14) becomes

$$\hat{u} = \underset{u}{\operatorname{argmin}} \; \|u - u_e\|_2^2 + \|\Lambda\, L\, u\|_2^2 \tag{5.15}$$

with the matrix $\Lambda \in \mathbb{R}^{(N-2)n_u \times (N-2)n_u}$ as follows

$$\Lambda = I_{N-2} \otimes \operatorname{diag}(\lambda_{u_1}, \ldots, \lambda_{u_{n_u}}) = \begin{bmatrix} \operatorname{diag}(\lambda_{u_1}, \ldots, \lambda_{u_{n_u}}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ddots & \operatorname{diag}(\lambda_{u_1}, \ldots, \lambda_{u_{n_u}}) \end{bmatrix}$$

where $\otimes$ is the Kronecker product.

We note also that, since $e_u = u_e - u$ from the definitions above, problem (5.15) is equivalent to the following problem

$$\hat{e}_u = \underset{e_u}{\operatorname{argmin}} \; \|e_u\|_2^2 + \|\Lambda\, L\,(u_e - e_u)\|_2^2 \tag{5.16}$$

which is in the form we will use in the following.

DLTI Model-Based Denoising In this paragraph we apply the regularization just considered to the underdetermined model-based denoising problem (5.13). Differently from the previous case, we must add two regularization terms for each input and output signal, and we obtain:

$$\min_{\tilde{z}} \left(\|G\tilde{z} - d\|_2^2 + \|\Lambda_{eu}^{min}\tilde{e}_u\|_2^2 + \|\Lambda_{ey}^{min}\tilde{e}_y\|_2^2 + \|\Lambda_{eu}^{curv}L_{n_u}(u_e - \hat{e}_u)\|_2^2 + \|\Lambda_{ey}^{curv}L_{n_y}(y_e - \hat{e}_y)\|_2^2 \right) \tag{5.17}$$

where $\tilde{e}_u, \tilde{e}_y, \hat{e}_u, \hat{e}_y$ are defined as in (5.12), the matrices $\Lambda_{<eu,ey>}^{<min,curv>}$ have the form of diagonal matrices, as described in the multiple signal case of previous subsection, more precisely

$$\begin{array}{ll} \Lambda_{eu}^{min} \in \mathbb{R}^{Nn_u \times Nn_u}, & \Lambda_{ey}^{min} \in \mathbb{R}^{Nn_y \times Nn_y}, \\ \Lambda_{eu}^{curv} \in \mathbb{R}^{(N-2)n_u \times (N-2)n_u}, & \Lambda_{ey}^{curv} \in \mathbb{R}^{(N-2)n_y \times (N-2)n_y}, \end{array}$$

and where $L_{n_u} \in \mathbb{R}^{(N-2)n_u \times Nn_u}$ and $L_{n_y} \in \mathbb{R}^{(N-2)n_y \times Nn_y}$ are rectangular matrices generated by the discretization of the second derivative operator, as defined previously.

Note that the terms of equation (5.17) have the following meaning

- the first term weights the model residual,
- the second and third terms weight the "distance" of the estimated denoised i/o signals to the measured ones,
- the last two terms weight the curvatures of the input-output denoised signals.

We can rewrite the least-squares regularized problem (5.17) as a least-squares problem as

$$\tilde{z}^* = \text{argmin}\ \|G_{reg}\tilde{z} - d_{reg}\|_2^2$$

that in more explicit form is

$$\tilde{z}^* = \text{argmin} \left\| \left[\begin{array}{cccc} \multicolumn{4}{c}{G} \\ \Lambda_{eu}^{min} & 0 & 0 & 0 \\ 0 & \Lambda_{ey}^{min} & 0 & 0 \\ \Lambda_{eu}^{curv}L_{n_u} & 0 & \Lambda_{eu}^{curv}L_{n_u} & 0 \\ 0 & \Lambda_{ey}^{curv}L_{n_y} & 0 & \Lambda_{ey}^{curv}L_{n_y} \end{array} \right] \begin{bmatrix} \tilde{e}_u \\ \tilde{e}_y \\ \bar{e}_u \\ \bar{e}_y \end{bmatrix} - \begin{bmatrix} d \\ 0 \\ 0 \\ \Lambda_{eu}^{curv}L_{n_u}u_e \\ \Lambda_{ey}^{curv}L_{n_y}y_e \end{bmatrix} \right\|_2^2 . \tag{5.18}$$

The matrix of the linear system, that we will use for the analysis in Section 5.4, is

$$G_{reg} = \left[\begin{array}{cccc} \multicolumn{4}{c}{G} \\ \Lambda_{eu}^{min} & 0 & 0 & 0 \\ 0 & \Lambda_{ey}^{min} & 0 & 0 \\ \Lambda_{eu}^{curv}L_{n_u} & 0 & \Lambda_{eu}^{curv}L_{n_u} & 0 \\ 0 & \Lambda_{ey}^{curv}L_{n_y} & 0 & \Lambda_{ey}^{curv}L_{n_y} \end{array} \right] \in \mathbb{R}^{n_G \times m_G} \tag{5.19}$$

with number of rows and columns respectively

$$\begin{cases} n_G & = (N(n_y + n_u) + N(n_u + n_y) + (N-2)(n_u + n_y)) = (3N-2)(n_y + n_u) \\ m_G & = (N+1)(n_y + n_u) \end{cases}$$

while the well-known term is

$$d_{reg} = \begin{bmatrix} d \\ 0 \\ 0 \\ \Lambda_{eu}^{curv} L_{n_u} u_e \\ \Lambda_{ey}^{curv} L_{n_y} y_e \end{bmatrix} \in \mathbb{R}^{n_G}$$

and the solution $\tilde{z}^* = [\tilde{e}_u^*, \tilde{e}_y^*, \bar{e}_u^*, \bar{e}_y^*] \in \mathbb{R}^{m_G}$.

We give now a simplification of problem (5.17) in the case of single input-single output (SISO), i.e. the scalar case with $n_u = n_y = 1$, $A = a \in \mathbb{R}$, $B = b \in \mathbb{R}$, which allow us a more precise analysis: in that case we obtain

$$\min_{\tilde{z}} \left(\|G\tilde{z} - d\|_2^2 + \|\lambda_{eu}^{min} \tilde{e}_u\|_2^2 + \|\lambda_{ey}^{min} \tilde{e}_y\|_2^2 + \|\lambda_{eu}^{curv} L_{n_u}(u_e - \hat{e}_u)\|_2^2 + \|\lambda_{ey}^{curv} L_{n_y}(y_e - \hat{e}_y)\|_2^2 \right). \tag{5.20}$$

Analogously to the single signal denoising case, we can highlight the following cases:

- for $\lambda_{eu^i}^{min}, \lambda_{ey^j}^{min} \to \infty\ \forall i, j$ the vectors $\tilde{e}_u^*$ and $\tilde{e}_u^*$ of the solution $\tilde{z}^*$ of the problem will be trivially zero. In this case the model is not satisfied since the denoised signals u_{den}^*, y_{den}^* coincide with the measured signals but for additive constants

$$u_{den}^* = u_e - \hat{e}_u^* = u_e - \bar{e}_u^*, \qquad y_{den}^* = y_e - \hat{e}_y^* = y_e - \bar{e}_y^*.$$

- for $\lambda_{eu^i}^{curv}, \lambda_{ey^j}^{curv} \to \infty\ \forall i, j$ the solution $\tilde{z}^*$ of the problem is such that the denoised signals are

$$u_{den}^* = \alpha^{ba} + \beta^{ba} t, \quad y_{den}^* = \alpha^{ba} + \beta^{ba} t$$

  i.e. are the best affine approximations of the time series. Hence the noise signals estimates tend to

$$e_u^* = u_e - u_{den}^*, \quad e_y^* = y_e - y_{den}^*.$$

  In this case, likewise the previous one, the model constraints are not satisfied.

- for $\lambda_{eu^i}^{curv} = 0, \lambda_{ey^j}^{curv} = 0\ \forall i, j$ and parameters $\lambda_{<eu^i, ey^i>}^{min}$ sufficiently small so that the model constraints are accurately satisfied, the problem is equivalent to the problem in [58, 59] in the case in which the covariance of the input and output noise is zero: the parameters $\lambda_{<eu,ey>}^{min}$ have the same role of the variances of the two noise signals.

- for $\lambda^{min}_{eu^i} = 0, \lambda^{min}_{ey^j} = 0\ \forall i, j$ and $\lambda^{curv}_{<eu^i,ey^i>}$ sufficiently small so that the model constraints are accurately satisfied, we obtain a regularized problem that doesn't have conditions on the norm of the solution that could explode.

Although each single regularization term makes the problem determined or overdetermined, they are not sufficient to generate an acceptable solution. Therefore, it is necessary to find the right trade-off of the parameters and consider a method for this multi-parameter choice problem.

The parameters cannot be determined by the minimization of the residual of problem (5.17), because that condition gives bad parameters of the kind listed above. This happens because the real solution is not the one that gives the minimum of the objective function in (5.17). For this reason we consider the Normalized Cumulative Periodogram ([34], [32]), also known as Bartlett test, as the quantity to be minimized. We use this method to test the whiteness of the vectors $\tilde{e}^*_u$ and $\tilde{e}^*_y$, i.e. the distance of these vectors from white noises, as are supposed to be in this work. None of the degenerate cases listed above gives a good value of whiteness of the solution, hence this method is able to keep the solution away from those critical situations and find acceptable solutions.

We propose an iterative algorithm that minimizes the residual and the whiteness of the solution of (5.17), with iterative optimizations on single parameters $\lambda^{<min,curv>}_{<eu^i,ey^i>}$. We will call this algorithm "Whiteness and Minimum-Curvature Model-Based Denoising" (WMC-MBD) and we will describe it in detail in subsection 5.5.

5.3.3 Comparison with previous works

In [58] and [59] a deterministic DLTI system in state-space form is considered (analogous to (5.5)) with noisy inputs and outputs, and is referred to as "noisy I/O model". The problem of estimating the state and the denoised input and output signals is then addressed, in the online and offline cases, defined respectively *filtering* and *smoothing* problems. In our case we are interested in the second of these problems that we quote here:

Problem 4 ("Optimal noisy I/O smoothing problem")**.** *Call $\tilde{u}, \tilde{y}$ measurement errors random, centered, normal, uncorrelated and white, with known covariance matrices*

$$cov(\tilde{u}) =: V_{\tilde{u}}(t), \quad cov(\tilde{y}) =: V_{\tilde{y}}(t)$$

and suppose that the initial condition $x_0 = \hat{x}(0)$ is known. Then, given the matrices A, B, C, D, the optimal noisy I/O smoothing problem is defined as

$$\begin{aligned} \min_{\hat{u},\hat{y},\hat{x}} & \left\| \begin{bmatrix} V_{\tilde{u}} & \\ & V_{\tilde{y}} \end{bmatrix}^{-1/2} \begin{bmatrix} \hat{u} - u_d \\ \hat{y} - y_d \end{bmatrix} \right\|_2^2 \\ s.t.\ & \hat{x}(t+1) = A\hat{x}(t) + B\hat{u}(t) \qquad for\ \ t = 0, 1, \ldots, t_f - 1. \\ & \hat{y}(t) = C\hat{x}(t) + D\hat{u}(t) \end{aligned} \tag{5.21}$$

The optimal smoothed state estimate $\hat{x}(\cdot, t_f)$ *is the solution of* (5.21).

In the notation of this work $t_f = N$. The estimate $\hat{x}(\cdot, t_f)$ is the optimal smoothed state estimate with a time horizon t_f, i.e. with the input and output signals supposed known in all the time instants. Hence, in this case, the resolution of the underdetermined system given by the model equations is found imposing the minimum of a weighted norm of the solution.

We observe here that our formulation have the model-constraining equations in common with the more general one considered in [59]. On the other hand we are assuming a particular case of the problem with C invertible and $D = 0$ and we are not interested in the state estimation. Moreover, in the approach of [59] the regularization terms considered are the norms of the noises, weighted with the covariance matrices known values. In this work, we don't suppose variance values to be known and we add regularization terms based on the curvature of these signals (second derivative).

5.4 General issues

5.4.1 Uncertainty of the noise bias values

Equations of system (5.13) only determine the ratio of the biases $\bar{e}_u, \bar{e}_y$, but not their independent values. In particular, every couple $(e_{y_{offset}}, e_{u_{offset}})$ such that

$$e_{y_{offset}} - M_0 \, e_{u_{offset}} = 0 \tag{5.22}$$

satisfies the model equations (5.11), where M_0 is the static gain of the model (steady-state value of the unit step response): $M_0 = (I - A)^{-1} B$ (where we recall that we are considering the case $C = I$ and $D = 0$).

Moreover, none of the regularization terms introduced in paragraph 5.3.2 gives any additional information to resolve this uncertainty, that remains in the regularized problem (5.17). This situation can be dangerous since the bias estimates calculated can explode, especially when the conditioning number of the regularized matrix G_{reg} obtained from (5.17) is big.

For this reason, we introduce an additional constraint equation, that weights the proximity of the bias $\bar{e}_u$ to an a-priori estimate of it, obtained with the truncated SVD method applied to the regularized matrix G_{reg}.

In fact, the bias of the noise e_u can be estimated as follows. Solving the regularized system with zero $\lambda^{curv}_{<eu,ey>}$ values and relatively big $\lambda^{min}_{<eu,ey>}$ values, truncating at the first singular value, i.e. considering only the first principal component, the solution $\hat{e}_u, \hat{e}_y$ is almost constant (i.e. $\tilde{e}_u, \tilde{e}_y$ are almost zero).

5.4.2 Singular values and ill-conditioning

If we consider small regularization parameters, $\lambda^{curv}_{<eu,ey>}$ and $\lambda^{min}_{<eu,ey>}$, the singular values of the regularized matrix G_{reg} have a constant trend for different kind of noise signals. As shown in Figure 5.2, this trend is characterized by some singular values of bigger amplitude, relative to the model-constraining, a gap, a group of smaller singular values, another gap, and a group of nearly zero singular values; this last group is the responsible for the uncertainty on the bias estimation of subsection 5.4.1.

The condition number of the matrix with small values of the regularization parameters is high and for this reason the matrix is numerically singular, in double precision. Since regularization parameters are chosen a-posteriori, with a criterion based on the whiteness of the estimated noise vector, it is necessary to solve the least-squares problem derived from (5.17) with a regularization approach to remain in the dynamic range of double precision, in particular we choose the Truncated SVD (TSVD).

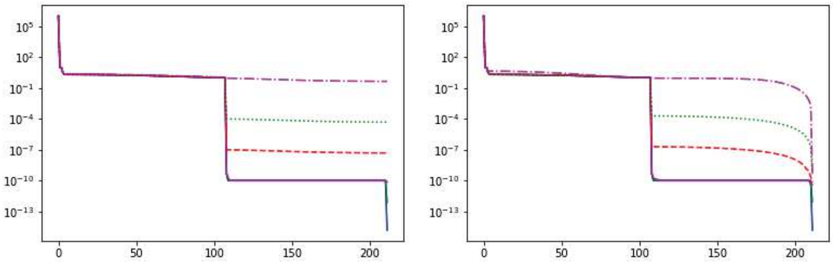


Figure 5.2: Singular values of the regularized matrix G_{reg} for different values of the regularization parameters: increasing λ^{min} (left) we see, going from the solid to the dashed line, a Tikhonov effect on singular values, while increasing λ^{curv} (right) we notice that the condition number remains the same.

5.5 The WMC-MBD algorithm

We propose here a denoising algorithm based on the considerations of subsection 5.4.

5.5.1 Iterative Algorithm

We consider an algorithm in which we iterate single-parameter optimizations to maximize the whiteness of the estimated noise signals. Given an estimated noise signal e, we minimize its deviation from white noise using the following measure of whiteness loss:

Definition 16 (Whiteness Loss)**.** *Given an estimated noise signal $e \in \mathbb{R}^N$, we call Whiteness Loss of e*

$$w_e := \|l - NCP(e)\|_2 \tag{5.23}$$

where $l \in \mathbb{R}^N$ is the vector of equispaced values from 0 *to* 1 *and $NCP(e)$ is the Normalized*

Cumulative Periodogram of the signal e, defined (see [32]) as the vector

$$NCP(e)_i := \frac{(p_e)_2 + (p_e)_3 + \cdots + (p_e)_{i+1}}{(p_e)_2 + (p_e)_3 + \cdots + (p_e)_{q+1}} \qquad \textit{for } i = 1, \ldots, q = \lfloor N/2 \rfloor$$

where

$$p_e = [|(f_e)_1|^2, |(f_e)_2|^2, \ldots, , |(f_e)_N|^2]^T$$

is the power spectrum density and $f_e = dft(e) = [(f_e)_1, (f_e)_2, \ldots, (f_e)_N]^T \in \mathbb{C}^N$ *is the discrete Fourier transform of e.*

Definition 17 (Curvature). *Given a signal* $u \in \mathbb{R}^{N \times 1}$*, we call curvature of u the signal*

$$L\,u \quad \in \mathbb{R}^{(N-2)\times 1}$$

where

$$L = \begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \end{bmatrix} \quad \in \mathbb{R}^{(N-2)\times N}.$$

In the iterative algorithm we will use definition (5.23) as a distance measure of the estimated noise signals from white noise. Hence, we will consider the functions

$$\begin{cases} w_{eu^i} = \|l - NCP(eu^i)\|_2 \\ w_{ey^i} = \|l - NCP(ey^i)\|_2 \end{cases}$$

where eu, ey are obtained from solving problem (5.17) for certain values of the parameters. The algorithm aims at finding such parameters $\Lambda^{<min,curv>}_{<eu,ey>}$ that produce estimated noise signals with a minimum value of the whiteness loss.

We can now introduce the ”Whiteness and Minimum Curvature Model-Based Denoising” algorithm:

where w_{eu} and w_{ey} are respectively the functions that calculate the whiteness loss values (equation (5.23)) of the estimated signals e_u and e_y obtained from solving (5.17) with the current parameters $\Lambda^{<min,curv>}_{<eu,ey>}$. The variables r_{eu} and r_{ey} are respectively the ratios between the parameters of each signal of input and output

$$r_{eu} = \frac{\lambda^{curv}_{eu}}{\lambda^{min}_{eu}}, \qquad r_{ey} = \frac{\lambda^{curv}_{ey}}{\lambda^{min}_{ey}}.$$

We consider adaptive grids to perform each global minimization in Algorithm 4. This is because the whiteness loss function is not a convex function of the parameters, hence an optimization algorithm (such as `scipy.optimize.minimize` of the Scipy library for Python) can easily stop on local minima.

Algorithm 4 ”Whiteness and Minimum Curvature Model-Based Denoising” (WMC-MBD)

```
A-priori estimate of the input noise bias $\bar{e}_u$ as described in subsection 5.4.1
Initialization of the current regularization parameters $\lambda^{min}_{<eu^i,ey^i>}$, $r_{<eu^i,ey^i>} = \frac{\lambda^{curv}_{<eu^i,ey^i>}}{\lambda^{min}_{<eu^i,ey^i>}}$ and initial guess of their optimal values $r^*_{eu^i}, \lambda^{min*}_{eu^i}, r^*_{ey^i}, \lambda^{min*}_{ey^i}$ to the initial mid-range value $10^{-7}$
for $k = 1, \ldots, K_{maxiter}$ do
    for $i = 1, \ldots, n_u$ do
        $\bar{r}_{eu^i} = \underset{r_{eu^i}}{\text{argmin}}\, w_{eu^i}$
            where $eu, ey$ are the solutions of (5.17) with the
            current regularization parameters $\lambda^{min}_{<eu^i,ey^i>}, r_{<eu^i,ey^i>}$
    end for
    for $i = 1, \ldots, n_u$ do
        $\bar{\lambda}^{min}_{eu^i} = \underset{\lambda^{min}_{eu^i}}{\text{argmin}}\, w_{eu^i}$
            where $eu, ey$ are the solutions of (5.17) with the
            current regularization parameters $\lambda^{min}_{<eu^i,ey^i>}, r_{<eu^i,ey^i>}$
    end for
    for $i = 1, \ldots, n_y$ do
        $\bar{r}_{ey^i} = \underset{r_{ey^i}}{\text{argmin}}\, w_{ey^i}$
            where $eu, ey$ are the solutions of (5.17) with the
            current regularization parameters $\lambda^{min}_{<eu^i,ey^i>}, r_{<eu^i,ey^i>}$
    end for
    for $i = 1, \ldots, n_y$ do
        $\bar{\lambda}^{min}_{ey^i} = \underset{\lambda^{min}_{ey^i}}{\text{argmin}}\, w_{ey^i}$
            where $eu, ey$ are the solutions of (5.17) with the
            current regularization parameters $\lambda^{min}_{<eu^i,ey^i>}, r_{<eu^i,ey^i>}$
    end for
    Curvature check: if the curvature of both I/O denoised signals is less than the previous iteration, update the optimal values $r^*_{eu^i}, \lambda^{min*}_{eu^i}, r^*_{ey^i}, \lambda^{min*}_{ey^i}$ to their current values
end for
```

We consider grids obtained in the following way: at each iteration we calculate the grid by multiplying the actual value of the considered parameter to the following vector

$$10^{k_{exp}} * [1, \frac{1}{2^1}, \frac{1}{2^2}, \frac{1}{2^3}, \ldots, \frac{1}{2^j}, 1, \frac{1}{2^{j+1}}, \ldots, \frac{1}{2^{j_{max}+1}}]$$

where $k_{exp} = (K_{maxiter} + 1 - k)$ for each iteration k, $K_{maxiter}$ is the maximum number of iterations, and $j_{max} = log_2(10^{k_{exp}-(-k_{exp})})$ so that this vector span values from $10^{k_{exp}}$ to $10^{-k_{exp}}$. Moreover, such grids are restricted so that the values considered are only the ones for which the parameters belong to the interval $[10^{-15}, 10^{-2}]$: on one hand the value of the parameters must be sufficiently bigger than machine precision, on the other hand a value of the parameters too big would compromise the model constraints.

After each group of four kinds of minimizations, the optimal parameters are conditionally updated, so that the tuple that generates a solution of minimum curvature among the ones obtained during the iteration is kept as the final solution. This is done because among the tuples that minimize the whiteness function, we are interested in the smoothest one, and it allows us to exclude over-whitened solutions which curvature can be high.

As already introduced, the assumption of unknown value of the norm of the noise implies that the proposed method belongs to the heuristic class. Moreover as we observed, the true signals do not satisfy the minimum of the regularization terms of equation (5.17). Therefore Algorithm 4 can reach satisfying denoised signals but not, in general, the original ones.

At each grid point of the optimization the following computations are performed: an SVD of matrix G_{reg}, the TSVD solution of the system and one FFT for the calculation of the whiteness loss, with a total computational cost

$$\mathcal{O}(\min(n_{reg}\, m_{reg}^2, n_{reg}^2\, m_{reg})) + \mathcal{O}(N \ \log(N))$$

where we call n_{reg}, m_{reg} the dimensions of matrix G_{reg}.

In the end, we observe that, in this work, both biases and variances of the noise signals are supposed to be unknown. However, in case these quantities are available, the algorithm can take advantage of that. In fact, if one of the biases of the noises is known, there's no need to calculate the a-priori estimate of the input bias. Moreover, if one of the variance values is available, it is possible to determine the ratio between the parameters $r_{<eu,ey>} = \lambda^{curv}_{<eu,ey>} / \lambda^{min}_{<eu,ey>}$ as in [68] and skip the calculation of these parameters in Algorithm 4.

5.6 Numerical experiments

In this subsection we will see the numerical comparison between the denoising carried out with the "modified Kalman filter" of [59] and the WMC-MBD (4). We will show the effectiveness of algorithm 4 through a Monte Carlo simulation, likewise [20].

5.6.1 Results and numerical comparisons of algorithms

We will show in the first two paragraphs the results for two particular examples, with sinusoidal and piecewise constant input, and after these we will show the results obtained for a bigger group of tests. For ease of presentation, to test the proposed method we reduce to the single input-single output (SISO) case, hence $n_u = n_y = 1$, $A = a \in \mathbb{R}$, $B = b \in \mathbb{R}$.

Example 1: sinusoidal input In this example we consider a sinusoidal input $u(t) = A\,sin(3\,t)$ with amplitude $A_u = 10$, sampled with sampling time $dt = 0.1$ in the interval $[0, 10]$. The additive Gaussian noises of this example have the following statistical properties: standard deviations $\sigma_{eu} = 5$, $\sigma_{ey} = 10$, and means $\mu_{eu} = 0$, $\mu_{ey} = 1$, and are generated with the Python function
`numpy.random.normal(bias, std, size=N)` from the Numpy library.

In Figure 5.3 we can see the noisy Input and Output signals and the signal obtained as their product, that is an example of a quantity of interest derived from the system I/O signals. In these graphs the sinusoidal trends are altered and difficult to recognize.

In Figures 5.4a, 5.4b and 5.4c we show the noisy and denoised input, output and product signals respectively, as follows:

- in the left figures the noisy signal (dashed line), and the comparison of the true signal with the denoised signal from algorithm 4 (WMC-MBD), in which the statistical properties of the noise are supposed unknown. For this test 8 iterations of the proposed algorithm are used;
- in the right figures the noisy signal (dashed line), and the comparison of the true signal with the denoised signal with the ”modified Kalman filter” of [59], in the case in which correct variances values are used, but biases are assumed zero (since it is an assumption of the method).

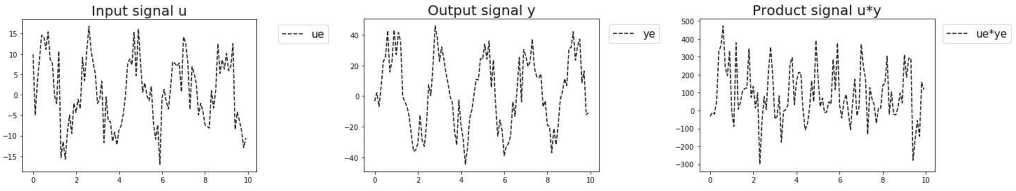
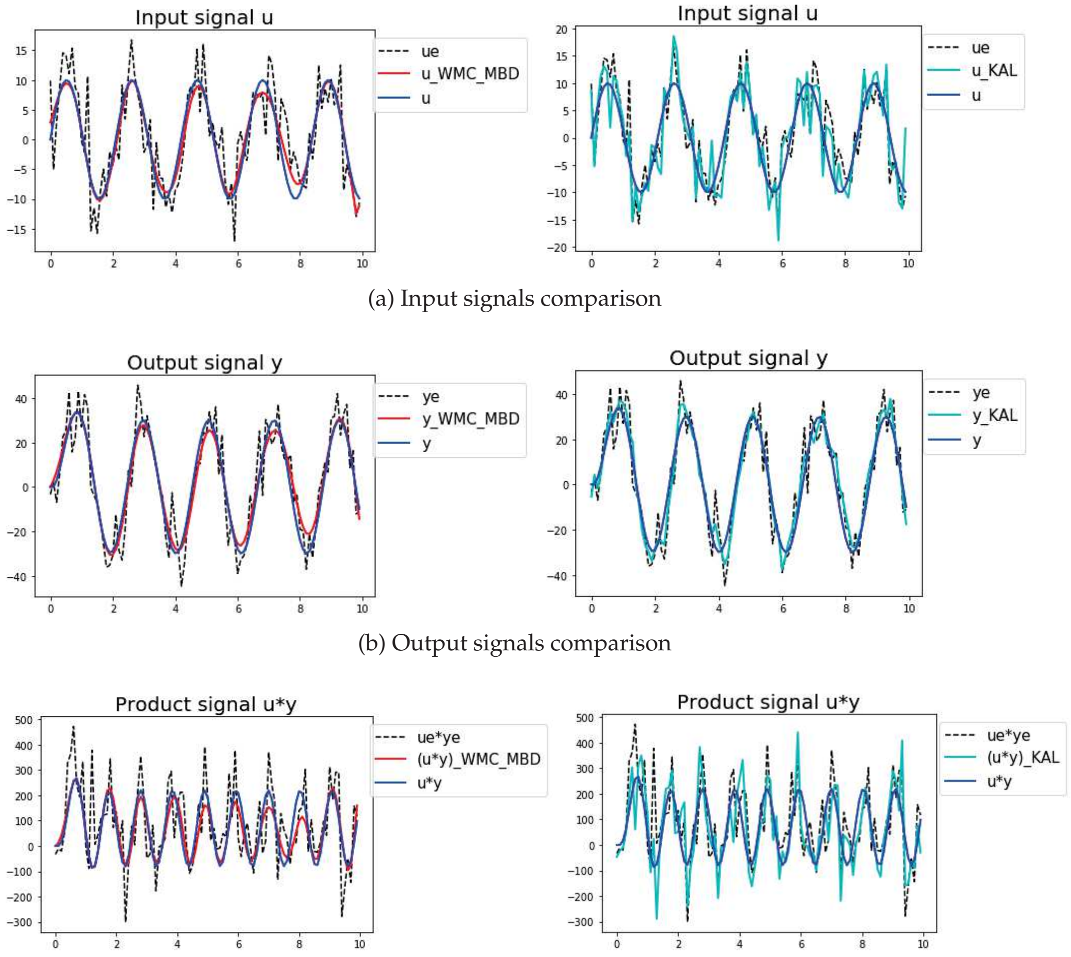


Figure 5.3: Example 1: Noisy Input, Output and Product signals.

Example 2: piecewise constant input In our second example we consider a piecewise constant input, with additive Gaussian noises with the same properties as the previous example. Analogously to the previous case we show in Figure 5.6a, 5.6b and 5.6c respectively the results for the input, output and product signals. As in the previous example, eight iterations of the proposed algorithm are used $K_{maxiter} = 8$.



(a) Input signals comparison

(b) Output signals comparison

(c) Product signals comparison

Figure 5.4: Comparison of the Input, Output and product signals for Example 1 respectively in figures 5.4a, 5.4b and 5.4c: in the left figures the noisy signal (dashed line), and the comparison of the true signal with the denoised signal from algorithm 4 (WMC-MBD), in which the statistical properties of the noise are supposed unknown; in the right figures the noisy signal (dashed line), and the comparison of the true signal with the denoised signal with the ”modified Kalman filter” of [59], in the casein which correct variances values are used, but biases are assumed zero (since it is an assumption of the method).

We considered this example since for these kind of signals other type of norms and regularizations are usually preferred, for example Total Variation ([29], [66], [79], [15]). However the obtained results show that this method is able to perform a useful denoising also in this critical situation. In Figure 5.5 we can see the noisy Input, Output and product signals, in which the discontinuous trend is altered and difficult to recognize.

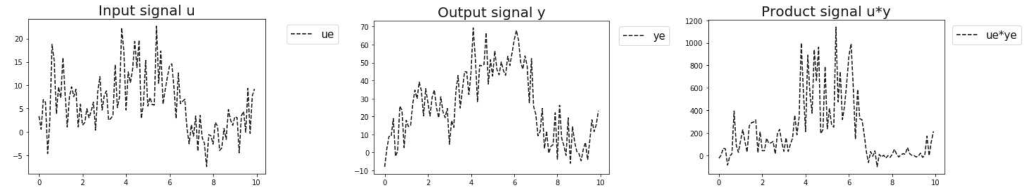


Figure 5.5: Example 2: Noisy Input, Output and Product signals.

Numerical results of the comparisons In this paragraph we show the results of the algorithm 4 on a total of 100 tests for different kinds of input signals and different values of variances and biases of I/O signals.

We consider 4 different kinds of input signals: a sinusoid, a piecewise constant signal (as in the two previous examples) and other two kinds of sum of two sinusoids with different amplitude and frequency values.

The couple of standard deviation values (square root of the variances) considered are the following

$$(\sigma_{eu}, \sigma_{ey}) = [(5, 10), (5, 12), (2.5, 15), (2.5, 7), (1.5, 10)].$$

For each of these, the following couples of biases have been tested:

$$(\mu_{eu}, \mu_{ey}) = [(0, 0), (0, 1), (1, 0), (1, -1), (-2, 3)].$$

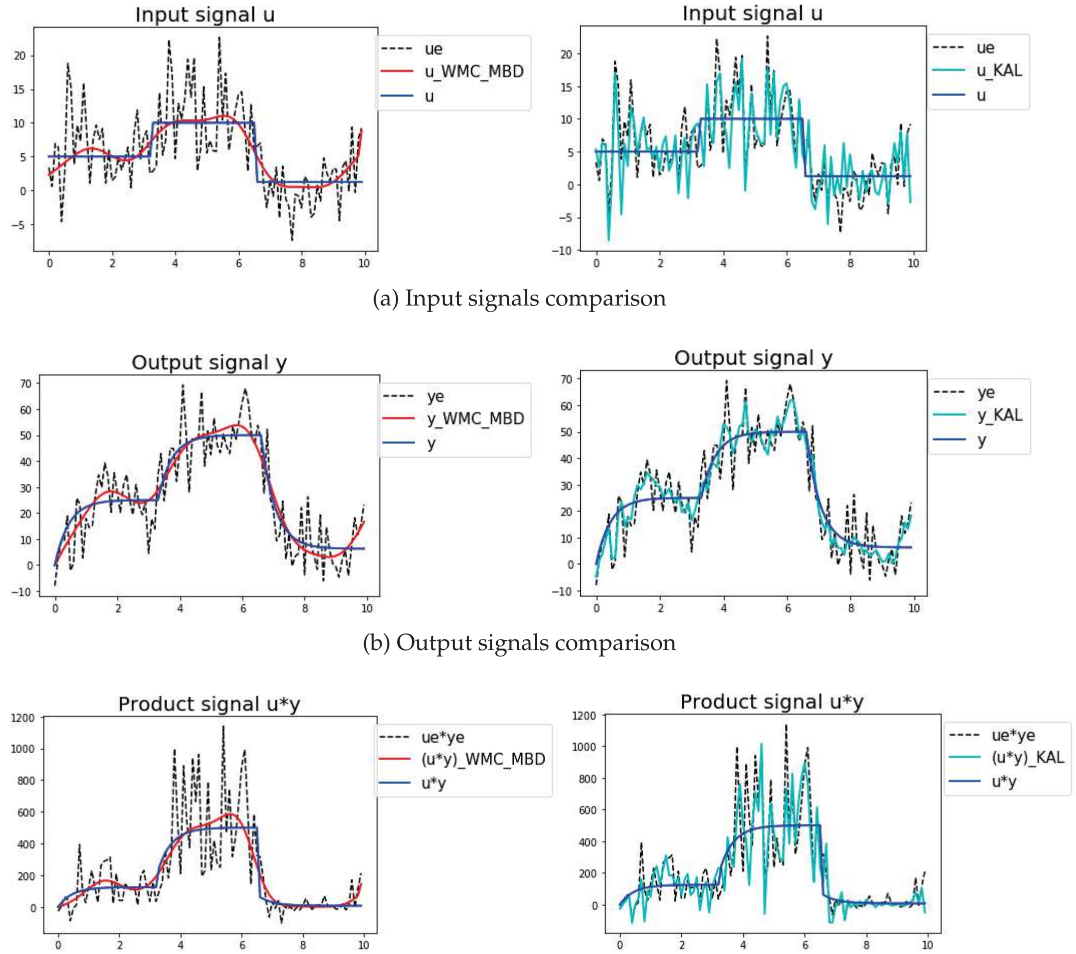
Hence 25 kinds of input-output noises have been tested for each of the 4 input signals.

In Tables 5.1, 5.2, 5.3 and 5.4, the values of mean and standard deviation of the relative errors for input and output signals are shown, for each type of input:

$$re_u = \frac{\|u - u^*\|}{\|u\|}, \qquad re_y = \frac{\|y - y^*\|}{\|y\|}.$$

Relative errors are shown in order for noisy measurements (”Noisy”), for I/O signals obtained with the ”modified Kalman filter” of [59], in the case in which the variances are supposed known (”KAL exact vars”) and unknown with hypothetic unitary values (”KAL vars=1”), and for I/O signals denoised with algorithm 4 (”WMC-MBD”). As in the previous examples, 8 iterations of the proposed algorithm are used.

Table 5.5 contains the results for all the tests in which the noise biases are zero, for all the four kinds of inputs. From these results, it is possible to compare the denoising obtained with the ”modified Kalman filter” with known and unknown variances values. In this second case we suppose hypothetic unitary values, and the results show that the denoising errors in this case are higher.



(a) Input signals comparison

(b) Output signals comparison

(c) Product signals comparison

Figure 5.6: Comparison of the Input, Output and product signals for Example 2 respectively in figures 5.4a, 5.4b and 5.4c: in the left figures the noisy signal (dashed line), and the comparison of the true signal with the denoised signal from algorithm 4 (WMC-MBD), in which the statistical properties of the noise are supposed unknown; in the right figures the noisy signal (dashed line), and the comparison of the true signal with the denoised signal with the ”modified Kalman filter” of [59], in the case in which correct variances values are used, but biases are assumed zero (since it is an assumption of the method).

5.7 Conclusions

We have presented a new algorithm, called “Whiteness Minimum-Curvature Model-Based Denoising” (WMC-MBD), for the denoising of data affected by white

data	re_u mean	re_u std	re_y mean	re_y std
Noisy	0.493	0.194	0.515	0.126
KAL (exact vars)	0.484	0.157	0.216	0.059
KAL (vars=1)	0.743	0.166	0.293	0.073
WMC-MBD	0.231	0.109	0.197	0.07

Table 5.1: Sinusoid Input Comparisons of relative errors *re*

data	re_u mean	re_u std	re_y mean	re_y std
Noisy	0.462	0.189	0.338	0.09
KAL (exact vars)	0.432	0.157	0.141	0.036
KAL (vars=1)	0.671	0.166	0.189	0.05
WMC-MBD	0.286	0.104	0.139	0.045

Table 5.2: Sum of sinusoids ($u = (A_u/2)\, sin(6\, t) + A_u\, sin(t)$) Input Comparisons of relative errors *re*

data	re_u mean	re_u std	re_y mean	re_y std
Noisy	0.442	0.168	0.532	0.139
KAL (exact vars)	0.433	0.14	0.229	0.063
KAL (vars=1)	0.66	0.15	0.309	0.079
WMC-MBD	0.308	0.104	0.242	0.085

Table 5.3: Sum of sinusoids ($u = A_u\, sin(6\, t) + (A_u/2)\, sin(t)$) Input Comparisons of relative errors *re*

data	re_u mean	re_u std	re_y mean	re_y std
Noisy	0.525	0.196	0.349	0.086
KAL (exact vars)	0.508	0.17	0.151	0.039
KAL (vars=1)	0.791	0.187	0.201	0.048
WMC-MBD	0.246	0.104	0.132	0.053

Table 5.4: Discontinuous Input Comparisons of relative errors *re*

data ($\mu_{<eu,ey>} = 0$)	re_u mean	re_u std	re_y mean	re_y std
Noisy	0.454	0.198	0.424	0.143
KAL (exact vars)	0.462	0.168	0.182	0.064
KAL (vars=1)	0.71	0.193	0.242	0.084
WMC-MBD	0.255	0.098	0.169	0.076

Table 5.5: Comparisons of relative errors *re* for the tests with zero bias noises ($\mu_{eu} = 0, \mu_{ey} = 0$) for all the different kind of signals

Gaussian noise whose statistical description, in terms of mean and variance, is not known. The algorithm produces denoised data that satisfies the input/output relations of the model used to describe the physical system.

The algorithm has been tested with success on various kind of signals: sinusoids, sums of sinusoids at distant frequencies and discontinuous signals as well.

There are two main questions that are difficult to solve and still open because of the complexity of the algorithm:

- it is not clear if (or under which conditions) there exists a set of parameters $\Lambda^{<min,curv>}_{<eu,ey>}$ for which problem (5.17) gives the real signals;
- the properties of the whiteness functions are difficult to analyse, in particular the existence-uniqueness of a global minimum of the four whiteness minimizations inside the loop of Algorithm 4.

Natural extensions of this work could be to consider different norms used for the regularization. For example, the norms used to regularize u and y could be different: u may be discontinuous, therefore the Total Variation could attain better results than the 2-norm, while y may be substantially smoother, being the system's output. Moreover, in the case of discontinuous signals it could be more useful to consider norms ℓ_p with $0 < p < 1$ and, therefore, methods that approximate such norms, like the Iterative Reweighting Least Squares or the Robust Least Squares.

The analysis of the extension to the case of nonlinear models is treated in the next Chapter.

Chapter 6

Denoising of I/O signals of Nonlinear Dynamic Models

6.1 Introduction

What we are going to tackle in this Chapter is the nonlinear extension of the problem of input and output denoising of a dynamical system treated in the previous Chapter 5, in which the model is known and the covariances of the noises are unknown.

As seen in the previous Chapter, the Kalman Problem (Problem 2 of Section 5.2) is related to the denoising of input-output signals, since the latter problem can be reduced to a particular case of the first one. The nonlinear extension of the Kalman problem has been studied in literature and is still an open field of research [12], even in the case with known covariances of the two noise terms.

Other related problems, similar to the one we are considering, are for example the unknown input estimation and the nonlinear noise reduction [18, 46], which, however, are different in the formulation.

We start this Chapter with the setting of the nonlinear denoising problem in Section 6.2, and then present two different formulations in Sections 6.3 and 6.4. The second formulation is the one we will analyse more in detail and of which we will give some numerical results in Subsection 6.4.4, followed by conclusions and future work.

6.2 Problem setting

Given the continuous ODE

$$\dot{y}(t) = f^c_{NL}(u(t), y(t)) \tag{6.1}$$

with f^c_{NL} a nonlinear function, we call f_{NL} the nonlinear map of the discrete system

$$y(k+1) = f_{NL}(u(k), y(k)) \tag{6.2}$$

and we suppose that the discretization error is small and negligible w.r.t. the scope of this analysis. Therefore, at discrete time, $u(k)$ is the true input and $y(k)$ the true solution with $k = 0, \ldots, N-1$.

The following definitions will be useful

$$\begin{aligned} u &:= [u^T(0) \cdots u^T(N-1)]^T \qquad && \in \mathbb{R}^{Nn_u \times 1} \\ y &:= [y^T(0) \cdots y^T(N)]^T && \in \mathbb{R}^{(N+1)n_y \times 1} \end{aligned} \tag{6.3}$$

Given some noisy measures u_e, y_e

$$\begin{cases} u_e(k) = u(k) + e_u(k) \\ y_e(k) = y(k) + e_y(k) \end{cases}$$

with e_u, e_y Gaussian noises with biases $\mu_{eu^i} \neq 0, \mu_{ey^i} \neq 0, \ \forall i$:

$$\begin{cases} \mathbb{E}\{e_{u^i}\} = \mu_{eu^i} \\ \mathbb{E}\{e_{u^i} e_{u^i}^T\} = \sigma^2_{eu^i} I \end{cases} \quad \text{for } i = 1, \ldots, n_u, \qquad \begin{cases} \mathbb{E}\{e_{y^i}\} = \mu_{ey^i} \\ \mathbb{E}\{e_{y^i} e_{y^i}^T\} = \sigma^2_{ey^i} I \end{cases} \quad \text{for } i = 1, \ldots, n_y. \tag{6.4}$$

Our aim in general is to obtain denoised signals $\hat{u}, \hat{y}$ minimizing the norm of the model residual vector given by:

$$r(\hat{y}, \hat{u}) = \begin{bmatrix} \hat{y}(1) - f_{NL}(\hat{y}(0), \hat{u}(0)) \\ \vdots \\ \hat{y}(i) - f_{NL}(\hat{y}(i-1), \hat{u}(i-1)) \\ \vdots \\ \hat{y}(N) - f_{NL}(\hat{y}(N-1), \hat{u}(N-1)) \end{bmatrix} \in \mathbb{R}^{n_y N \times 1} \tag{6.5}$$

i.e. we are looking for estimates of input/output that satisfy the model. The related estimates of the noise are

$$\begin{cases} \hat{e}_u(k) = \tilde{e}_u(k) + \bar{e}_u \\ \hat{e}_y(k) = \tilde{e}_y(k) + \bar{e}_y \end{cases} \tag{6.6}$$

where $\tilde{e}_u, \tilde{e}_y$ are the white gaussian noises (with zero mean), and $\bar{e}_u, \bar{e}_y$ are constants, estimated means of the input/output noises. Hence the denoised signals $\hat{u}, \hat{y}$ are related to the estimates of the noise in the following way

$$\begin{cases} \hat{u}(k) = u_e(k) - \hat{e}_u(k) \\ \hat{y}(k) = y_e(k) - \hat{e}_y(k). \end{cases} \tag{6.7}$$

Hence, we obtain a nonlinear problem, which is ill-posed since there may be more solutions: we have not introduced yet the minimization of the residual with respect to the measured values.

Let us call

$$\tilde{z} = [\tilde{e}_u, \tilde{e}_y, \bar{e}_u, \bar{e}_y] \in \mathbb{R}^{n_u N + n_y(N+1) + n_u + n_y = n_u(N+1) + n_y(N+2) =: n_z}$$

the vector of *error variables*.

In the linear case, treated in the previous Section, the linear system (5.13) given by the model constraint was under-determined and its solution was unique up to multiplicative terms. We added four regularization terms to the least-squares problem generated by the linear system, to control the smoothness of the input and output variables. After choosing the weight of each regularization term, it was possible to calculate the unique optimal input/output signals that solve the regularized linear least squares optimization problem (5.17).

We would like to extend Algorithm 4 to include the nonlinear case: with the same structure of the algorithm, we want to substitute at each step of the iterations the resolution of the linear problem (5.17) with a nonlinear formulation of it.

We will analyze two formulations (or criteria) of this problem as optimizations with respect to two different sets of variables: the error variables and the initial condition on y and input variables. The first formulation is the direct consequence of the one used in the previous Section for the linear case, but has a number of parameters higher than the second one, on which methods introduced in Section 3.3.1 can be used as we will see.

6.3 Formulation 1: Optimization w.r.t the error variables

We first study the direct generalization of the WMC-MBD of the previous Section to the nonlinear case.

In the nonlinear case, the minimization of the residual (6.5) gives multiple solutions, so we still need some regularization terms to obtain smooth signals. Note that, in contrast to the linear case, the solutions of the non-regularized problem here are not related by a multiplicative term.

If we add the regularization terms as in the linear case we obtain:

$$\min_{\tilde{z}} \left(\|r(\hat{y}, \hat{u})\|_2^2 + \|\Lambda_{eu}^{min} \tilde{e}_u\|_2^2 + \|\Lambda_{ey}^{min} \tilde{e}_y\|_2^2 + \|\Lambda_{eu}^{curv} L_{n_u}(u_e - \hat{e}_u)\|_2^2 + \|\Lambda_{ey}^{curv} L_{n_y}(y_e - \hat{e}_y)\|_2^2 \right) \tag{6.8}$$

where

- we recall that the regularization matrices Λ are of the form:

$$\Lambda = I_{<\cdot>} \otimes \mathrm{diag}(\lambda_{u_1}, \ldots, \lambda_{u_{n_u}}) = \begin{bmatrix} \mathrm{diag}(\lambda_{u_1}, \ldots, \lambda_{u_{n_u}}) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ddots & \mathrm{diag}(\lambda_{u_1}, \ldots, \lambda_{u_{n_u}}) \end{bmatrix}$$

  where $I_{<\cdot>}$ is I_N in the case of Λ^{min} and I_{N-2} in the case of Λ^{curv} and with $\Lambda_{eu}^{min} \in \mathbb{R}^{Nn_u \times Nn_u}$, $\Lambda_{ey}^{min} \in \mathbb{R}^{(N+1)n_y \times (N+1)n_y}$, $\Lambda_{eu}^{curv} \in \mathbb{R}^{(N-2)n_u \times (N-2)n_u}$, and $\Lambda_{ey}^{curv} \in \mathbb{R}^{(N-1)n_y \times (N-1)n_y}$,

- and the matrices L_{n_u} and L_{n_y} have the following form

$$L_{n_u} = \begin{bmatrix} I_{n_u} & -2\,I_{n_u} & I_{n_u} & & & & \\ & I_{n_u} & -2\,I_{n_u} & I_{n_u} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & I_{n_u} & -2\,I_{n_u} & I_{n_u} & \\ & & & & I_{n_u} & -2\,I_{n_u} & I_{n_u} \end{bmatrix} \in \mathbb{R}^{(N-2)\,n_u \times N\,n_u}$$

and analogously $L_{n_y} \in \mathbb{R}^{(N-1)\,n_y \times (N+1)\,n_y}$.

Moreover, we recall that, as in the linear case, the terms have the following meaning:

1. the first term $\|r\|_2^2$ weights the model residual,
2. the second and third terms weight the "distance" of the denoised input/output signals to the measured ones,
3. the last two terms weight the curvatures of the input/output denoised signals.

If we define

$$r_\Lambda(\hat{y},\hat{u}) = \begin{bmatrix} r(\hat{u},\hat{y}) \\ \Lambda_{eu}^{min}\tilde{e}_u \\ \Lambda_{ey}^{min}\tilde{e}_y \\ \Lambda_{eu}^{curv}L_{n_u}(u_e - \hat{e}_u) \\ \Lambda_{ey}^{curv}L_{n_y}(y_e - \hat{e}_y) \end{bmatrix} \in \mathbb{R}^{n_y N + n_u N + n_u(N-2) + n_y(N+1) + n_y(N-1) \times 1}$$

the previous problem (6.8) is equivalent to

$$\min_{\tilde{z}} \|r_\Lambda(\tilde{z})\|_2^2 \tag{6.9}$$

that we can solve with Gauss-Newton iterations

$$J^T(\tilde{z}^i)\,J(\tilde{z}^i)\,\delta\tilde{z}^i = -J^T(\tilde{z}^i)\,r_\Lambda(\tilde{z}^i) \tag{6.10}$$

$$\tilde{z}^{i+1} = \tilde{z}^i + \delta\tilde{z}^i \tag{6.11}$$

where J is the Jacobian of the residual vector r_Λ

$$J \in \mathbb{R}^{(n_y N + 2 n_u (N-1) + 2 n_y N) \times n_z}$$

$$J = \frac{\partial r_\Lambda}{\partial \tilde{z}} = \begin{bmatrix} \frac{\partial r}{\partial \tilde{z}} \\ \frac{\partial(\Lambda_{eu}^{min}\tilde{e}_u)}{\partial \tilde{z}} \\ \frac{\partial(\Lambda_{ey}^{min}\tilde{e}_y)}{\partial \tilde{z}} \\ \frac{\partial(\Lambda_{eu}^{curv}L_{nu}(u_e - \hat{e}_u))}{\partial \tilde{z}} \\ \frac{\partial(\Lambda_{ey}^{curv}L_{ny}(y_e - \hat{e}_y))}{\partial \tilde{z}} \end{bmatrix} = \tag{6.12}$$

$$= \begin{bmatrix} \frac{\partial r}{\partial \tilde{e}_u} & \frac{\partial r}{\partial \tilde{e}_y} & \frac{\partial r}{\partial \bar{e}_u} & \frac{\partial r}{\partial \bar{e}_y} \\ \frac{\partial(\Lambda_{eu}^{min}\tilde{e}_u)}{\partial \tilde{e}_u} & 0 & 0 & 0 \\ 0 & \frac{\partial(\Lambda_{ey}^{min}\tilde{e}_y)}{\partial \tilde{e}_y} & 0 & 0 \\ \frac{\partial(\Lambda_{eu}^{curv}L_{nu}(u_e - \hat{e}_u))}{\partial \tilde{e}_u} & 0 & \frac{\partial(\Lambda_{eu}^{curv}L_{nu}(u_e - \hat{e}_u))}{\partial \bar{e}_u} & 0 \\ 0 & \frac{\partial(\Lambda_{ey}^{curv}L_{ny}(y_e - \hat{e}_y))}{\partial \tilde{e}_y} & 0 & \frac{\partial(\Lambda_{ey}^{curv}L_{ny}(y_e - \hat{e}_y))}{\partial \bar{e}_y} \end{bmatrix} \tag{6.13}$$

We show the explicit formulation of the partial derivatives in the following.

The partial derivative w.r.t the vector of white noise $\tilde{e}_y$ is

$$\begin{aligned} \frac{\partial r}{\partial \tilde{e}_y}(\hat{y}, \hat{u}) &= \begin{bmatrix} \frac{\partial \hat{y}(1)}{\partial \tilde{e}_y} - \frac{\partial f_{NL}(\hat{y}(0), \hat{u}(0))}{\partial \tilde{e}_y} \\ \frac{\partial \hat{y}(2)}{\partial \tilde{e}_y} - \frac{\partial f_{NL}(\hat{y}(1), \hat{u}(1))}{\partial \tilde{e}_y} \\ \vdots \\ \frac{\partial \hat{y}(N)}{\partial \tilde{e}_y} - \frac{\partial f_{NL}(\hat{y}(N-1), \hat{u}(N-1))}{\partial \tilde{e}_y} \end{bmatrix} = \\ &= \begin{bmatrix} -\frac{\partial f_{NL}}{\partial \tilde{e}_y(0)}(\hat{y}(0), \hat{u}(0)) & \frac{\partial \hat{y}(1)}{\partial \tilde{e}_y(1)} & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & -\frac{\partial f_{NL}}{\partial \tilde{e}_y(N-1)}(\hat{y}(N-1), \hat{u}(N-1)) & \frac{\partial \hat{y}(N)}{\partial \tilde{e}_y(N)} \end{bmatrix} = \\ &= \begin{bmatrix} -\frac{\partial f_{NL}}{\partial \tilde{e}_y(0)}(\hat{y}(0), \hat{u}(0)) & -I_{n_y} & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & -\frac{\partial f_{NL}}{\partial \tilde{e}_y(N-1)}(\hat{y}(N-1), \hat{u}(N-1)) & -I_{n_y} \end{bmatrix} \in \mathbb{R}^{n_y N \times n_y(N+1)} \end{aligned} \tag{6.14}$$

where the last equality comes from the substitution of (6.6) in (6.7).

The partial derivative w.r.t the mean $\bar{e}_y$ is

$$\frac{\partial r}{\partial \bar{e}_y}(\hat{y},\hat{u}) = \begin{bmatrix} \frac{\partial \hat{y}(1)}{\partial \bar{e}_y} - \frac{\partial f_{NL}(\hat{y}(0),\hat{u}(0))}{\partial \bar{e}_y} \\ \frac{\partial \hat{y}(2)}{\partial \bar{e}_y} - \frac{\partial f_{NL}(\hat{y}(1),\hat{u}(1))}{\partial \bar{e}_y} \\ \vdots \\ \frac{\partial \hat{y}(N)}{\partial \bar{e}_y} - \frac{\partial f_{NL}(\hat{y}(N-1),\hat{u}(N-1))}{\partial \bar{e}_y} \end{bmatrix} = \begin{bmatrix} -I_{n_y} - \frac{\partial f_{NL}(\hat{y}(0),\hat{u}(0))}{\partial \bar{e}_y} \\ -I_{n_y} - \frac{\partial f_{NL}(\hat{y}(1),\hat{u}(1))}{\partial \bar{e}_y} \\ \vdots \\ -I_{n_y} - \frac{\partial f_{NL}(\hat{y}(N-1),\hat{u}(N-1))}{\partial \bar{e}_y} \end{bmatrix} \quad \in \mathbb{R}^{n_y N \times n_y}. \tag{6.15}$$

The partial derivatives w.r.t $\tilde{e}_u$ and $\bar{e}_u$ can be computed analogously.

Recalling that $\tilde{e}_u \in \mathbb{R}^{n_u N}$ and $\Lambda_{eu}^{min} \in \mathbb{R}^{n_u N \times n_u N}$, we have the following matrices dimensions

$$\frac{\partial(\Lambda_{eu}^{min}\tilde{e}_u)}{\partial \tilde{e}_u} = \Lambda_{eu}^{min} * I_{n_u N \times n_u N} = \Lambda_{eu}^{min} \in \mathbb{R}^{n_u N \times n_u N}, \tag{6.16}$$

$$\frac{\partial(\Lambda_{eu}^{curv} L_{n_u}\hat{u})}{\partial \tilde{e}_u} = \begin{bmatrix} \frac{\partial(\Lambda_{eu}^{curv} L_{n_u}\hat{u})}{\partial \tilde{e}_u(0)} & \dots & \frac{\partial(\Lambda_{eu}^{curv} L_{n_u}\hat{u})}{\partial \tilde{e}_u(N-1)} \end{bmatrix} = -\Lambda_{eu}^{curv} L_{n_u} \in \mathbb{R}^{n_u(N-2)\times n_u N} \tag{6.17}$$

and the last one is a band matrix, more precisely a block tridiagonal matrix.

The optimization of equation (6.9) is to be embedded in a brute-force optimization on the values of $\Lambda_{<eu,ey>}^{<min,curv>}$, as in Algorithm 4, i.e. has to be done for a grid of values of each $\Lambda_{<eu,ey>}^{<min,curv>}$ to find the correct values that minimize the whiteness function.

6.4 Formulation 2: Optimization w.r.t the initial condition and input

In this second formulation we will write the input/output denoising problem as a constrained optimization problem. In this case the unknown vector of variable to estimate is made of the input signals $\hat{u}(k)$ and the initial output value $\hat{y}(0)$, and the constraints are the state equations of the nonlinear model.

Let us call the variables to estimate

$$w = [\hat{y}(0), \hat{u}(0), \dots, \hat{u}(N-1), \bar{e}_y, \bar{e}_u] \quad \in \mathbb{R}^{2n_y + (N+1)n_u =: n_w}$$

and impose the discrete system equations

$$\hat{y}(k+1) - f_{NL}(\hat{u}(k), \hat{y}(k)) = 0 \qquad \text{for} \quad k = 0, \dots, N-1 \tag{6.18}$$

i.e. from the initial condition $y(0)$ and the input $u(k)$ we can deduce the remaining denoised vector $\hat{y}(k)$. For simplicity we will call

$$f_{k+1}(\hat{y}(k+1), \hat{y}(k), \hat{u}(k)) = \hat{y}(k+1) - f_{NL}(\hat{u}(k), \hat{y}(k)) \qquad \text{for} \quad k = 0, \dots, N-1.$$

The problem we are going to consider is

$$\begin{aligned} \min_{w=[\hat{y}(0),\hat{u}(0),\dots,\hat{u}(N),\bar{e}_y,\bar{e}_u]} \quad & \left(\|\Lambda_{eu}^{min}(u_e - (\hat{u}+\bar{e}_u)\|_2^2 + \|\Lambda_{ey}^{min}(y_e - (\hat{y}+\bar{e}_y)\|_2^2 + \|\Lambda_{eu}^{curv} L_{n_u}\hat{u}\|_2^2 + \|\Lambda_{ey}^{curv} L_{n_y}\hat{y}\|_2^2\right) \\ \text{s.t.} \quad & f_{k+1}(y(k+1), y(k), u(k)) = 0 \qquad \text{for} \quad k = 0,\dots,N-1 \end{aligned} \tag{6.19}$$

where the Nonlinear Least-Squares residual is the second term $\|\Lambda_{ey}^{min}(y_e - (\hat{y}+\bar{e}_y)\|_2^2$ and the remaining are regularization terms.

Observation 3. *Note that problem* (6.19) *is equivalent to the following problem*

$$\begin{aligned} \min_{[\tilde{e}_y(0),\tilde{e}_u(0),\dots,\tilde{e}_u(N),\bar{e}_y,\bar{e}_u]} \quad & \left(\|\Lambda_{eu}^{min}\tilde{e}_u\|_2^2 + \|\Lambda_{ey}^{min}\tilde{e}_y\|_2^2 + \|\Lambda_{eu}^{curv} L_{n_u}(u_e - \hat{e}_u)\|_2^2 + \|\Lambda_{ey}^{curv} L_{n_y}(y_e - \hat{e}_y)\|_2^2\right) \\ \textit{s.t.} \quad & f_{k+1}(y(k+1), y(k), u(k)) = 0 \qquad \textit{for} \quad k = 0,\dots,N-1 \end{aligned}$$

Note that, if we consider the input samples $u(k)$ as parameters, we can see this problem as the identification of initial condition and parameters of the continuous ODE (6.1), that is a Nonlinear Least Squares Inverse Problem which is well studied [16], [42], [30]. With respect to the classical formulation (for example the problem in [30]) in our case two regularization terms are added and we have a Nonlinear Least Squares Inverse Problem regularized with the Tikhonov regularization method.

Calling the residual vector

$$q_\Lambda(w) = \begin{bmatrix} (\Lambda_{eu}^{min}(u_e - (\hat{u}+\bar{e}_u)) \\ (\Lambda_{ey}^{min}(y_e - (\hat{y}+\bar{e}_y)) \\ (\Lambda_{eu}^{curv} L_{n_u}\hat{u}) \\ (\Lambda_{ey}^{curv} L_{n_y}\hat{y}) \end{bmatrix} \in \mathbb{R}^{n_u N + n_y(N+1) + n_u(N-2) + n_y(N-1) = 2n_u(N-1) + 2n_y N =: n_q} \tag{6.20}$$

problem (6.19) is equivalent to

$$\begin{aligned} \min_{w=[\hat{y}(0),\hat{u}(0),\dots,\hat{u}(N),\bar{e}_y,\bar{e}_u]} \quad & \|q_\Lambda(w)\|_2^2 \\ \text{s.t.} \quad & f_{k+1}(y(k+1), y(k), u(k)) = 0 \qquad \text{for} \quad k = 0,\dots,N-1 \end{aligned} \tag{6.21}$$

on which we can apply again Gauss-Newton iterations:

$$J^T(w^i)J(w^i)\delta w^i = -J^T(w^i)q_\Lambda(w^i) \tag{6.22}$$

$$w^{i+1} = w^i + \delta w^i \tag{6.23}$$

where $J = J(w) = \frac{\partial q_\Lambda}{\partial w}(w) \in \mathbb{R}^{n_q \times n_w}$.

Note that the number of parameters to estimate in this second formulation is $(2n_y + (N+1)n_u)$, smaller than in the first formulation $(n_y(N+2) + n_u(N+1))$.

Hence, the Jacobian matrix of this case is smaller. It is calculated on w at each of the iterations of the optimization method, and has the following form

$$J = \frac{\partial q_\Lambda}{\partial w} = \begin{bmatrix} \frac{\partial(\Lambda_{eu}^{min}(u_e - (\hat{u} + \bar{e}_u))}{\partial w} \\ \frac{\partial(\Lambda_{ey}^{min}(y_e - (\hat{y} + \bar{e}_y))}{\partial w} \\ \frac{\partial(\Lambda_{eu}^{curv} L_{n_u} \hat{u})}{\partial w} \\ \frac{\partial(\Lambda_{ey}^{curv} L_{n_y} \hat{y})}{\partial w} \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & \frac{\partial(\Lambda_{eu}^{min}(u_e - (\hat{u} + \bar{e}_u))}{\partial \hat{u}(0)} & \cdots & \frac{\partial(\Lambda_{eu}^{min}(u_e - (\hat{u} + \bar{e}_u))}{\partial \hat{u}(N-1)} & \frac{\partial(\Lambda_{eu}^{min}(u_e - (\hat{u} + \bar{e}_u))}{\partial \bar{e}_y} & \frac{\partial(\Lambda_{eu}^{min}(u_e - (\hat{u} + \bar{e}_u))}{\partial \bar{e}_u} \\ \frac{\partial(\Lambda_{ey}^{min}(y_e - (\hat{y} + \bar{e}_y))}{\partial \hat{y}(0)} & \frac{\partial(\Lambda_{ey}^{min}(y_e - (\hat{y} + \bar{e}_y))}{\partial \hat{u}(0)} & \cdots & \frac{\partial(\Lambda_{ey}^{min}(y_e - (\hat{y} + \bar{e}_y))}{\partial \hat{u}(N-1)} & \frac{\partial(\Lambda_{ey}^{min}(y_e - (\hat{y} + \bar{e}_y))}{\partial \bar{e}_y} & \frac{\partial(\Lambda_{ey}^{min}(y_e - (\hat{y} + \bar{e}_y))}{\partial \bar{e}_u} \\ 0 & \frac{\partial(\Lambda_{eu}^{curv} L_{n_u} \hat{u})}{\partial \hat{u}(0)} & \cdots & \frac{\partial(\Lambda_{eu}^{curv} L_{n_u} \hat{u})}{\partial \hat{u}(N-1)} & \frac{\partial(\Lambda_{eu}^{curv} L_{n_u} \hat{u})}{\partial \bar{e}_y} & \frac{\partial(\Lambda_{eu}^{curv} L_{n_u} \hat{u})}{\partial \bar{e}_u} \\ \frac{\partial(\Lambda_{ey}^{curv} L_{n_y} \hat{y})}{\partial \hat{y}(0)} & \frac{\partial(\Lambda_{ey}^{curv} L_{n_y} \hat{y})}{\partial \hat{u}(0)} & \cdots & \frac{\partial(\Lambda_{ey}^{curv} L_{n_y} \hat{y})}{\partial \hat{u}(N-1)} & \frac{\partial(\Lambda_{ey}^{curv} L_{n_y} \hat{y})}{\partial \bar{e}_y} & \frac{\partial(\Lambda_{ey}^{curv} L_{n_y} \hat{y})}{\partial \bar{e}_u} \end{bmatrix}. \tag{6.24}$$

Extended form of the Jacobian components We give here an extended formulation of the components of the Jacobian matrix (6.24) for each row.

First row

$$\frac{\partial(\Lambda_{eu}^{min}(u_e - (\hat{u} + \bar{e}_u))}{\partial \hat{u}(i)} = \frac{\partial(\Lambda_{eu}^{min}(u_e - (\hat{u} + \bar{e}_u))}{\partial \hat{u}} \frac{\partial \hat{u}}{\partial \hat{u}(i)} = -\Lambda_{eu}^{min} \begin{bmatrix} 0_{n_u i \times n_u} \\ I_{n_u \times n_u} \\ 0_{n_u(N-1-i) \times n_u} \end{bmatrix} \in \mathbb{R}^{n_u N \times n_u} \tag{6.25}$$

since $\Lambda_{eu}^{min} \in \mathbb{R}^{Nn_u \times Nn_u}$.

Second row:

$$\frac{\partial(\Lambda_{ey}^{min}(y_e - (\hat{y} + \bar{e}_y))}{\partial \hat{y}(0)} = \frac{\partial(\Lambda_{ey}^{min}(y_e - (\hat{y} + \bar{e}_y))}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \hat{y}(0)} = -\Lambda_{ey}^{min} \frac{\partial \hat{y}}{\partial \hat{y}(0)} \in \mathbb{R}^{n_y(N+1) \times n_y} \tag{6.26}$$

since $\Lambda_{ey}^{min} \in \mathbb{R}^{(N+1)n_y \times (N+1)n_y}$ and where $\frac{\partial \hat{y}}{\partial \hat{y}(0)}$ is calculated with the variational equations below.

$$\frac{\partial(\Lambda_{ey}^{min}(y_e - (\hat{y} + \bar{e}_y))}{\partial \hat{u}(i)} = \frac{\partial(\Lambda_{ey}^{min}(y_e - (\hat{y} + \bar{e}_y))}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \hat{u}(i)} = -\Lambda_{ey}^{min} \frac{\partial \hat{y}}{\partial \hat{u}(i)} \in \mathbb{R}^{n_y(N+1) \times n_u} \tag{6.27}$$

Third row:

$$\frac{\partial(\Lambda_{eu}^{curv} L_{n_u} \hat{u})}{\partial \hat{u}(i)} = \frac{\partial(\Lambda_{eu}^{curv} L_{n_u} \hat{u})}{\partial \hat{u}} \frac{\partial \hat{u}}{\partial \hat{u}(i)} = \Lambda_{eu}^{curv} L_{n_u} \begin{bmatrix} 0_{n_u i \times n_u} \\ I_{n_u \times n_u} \\ 0_{n_u(N-1-i) \times n_u} \end{bmatrix} \in \mathbb{R}^{n_u(N-2) \times n_u} \tag{6.28}$$

since $\Lambda_{eu}^{curv} \in \mathbb{R}^{(N-2)n_u \times (N-2)n_u}$ and $L_{n_u} \in \mathbb{R}^{(N-2)n_u \times Nn_u}$.

Fourth row

$$\frac{\partial(\Lambda_{ey}^{curv} L_{n_y} \hat{y})}{\partial \hat{y}(0)} = \frac{\partial(\Lambda_{ey}^{curv} L_{n_y} \hat{y})}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \hat{y}(0)} = \Lambda_{ey}^{curv} L_{n_y} \begin{bmatrix} I_{n_y \times n_y} \\ 0_{n_y N \times n_y} \end{bmatrix} \in \mathbb{R}^{n_y(N-1) \times n_y} \tag{6.29}$$

since $\Lambda_{ey}^{curv} \in \mathbb{R}^{(N-1)n_y \times (N-1)n_y}$ and $L_{n_y} \in \mathbb{R}^{(N-1)n_y \times (N+1)n_y}$.

$$\frac{\partial(\Lambda_{ey}^{curv} L_{n_y} \hat{y})}{\partial \hat{u}(i)} = \frac{\partial(\Lambda_{ey}^{curv} L_{n_y} \hat{y})}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \hat{u}(i)} = \Lambda_{ey}^{curv} L_{n_y} \frac{\partial \hat{y}}{\partial \hat{u}(i)} \in \mathbb{R}^{n_y(N-1) \times n_u}. \tag{6.30}$$

6.4.1 Sensitivity equations for the computation of the Jacobian

In the previous Section we wrote the entries of the matrix J as functions of the partial derivatives $\frac{\partial \hat{y}(k)}{\partial \hat{y}(0)}$ and $\frac{\partial \hat{y}(k)}{\partial \hat{u}(i)}$.

For the calculation of these derivative terms we use in this paragraph the sensitivity equations introduced in Section 3.3.1.

The procedure is the same as in [30], but the residual function $q_\Lambda(w)$ defined in Equation (6.20) has more entries due to the regularization terms.

In our case $p = w = [y(0), u(0), \ldots, u(N-1), \bar{e}_y, \bar{e}_u]$.

From the equation

$$\left[\frac{\partial \dot{y}}{\partial y_0}, \frac{\partial \dot{y}}{\partial u_0}, \ldots, \frac{\partial \dot{y}}{\partial u_{N-1}}, \frac{\partial \dot{y}}{\partial \bar{e}_y}, \frac{\partial \dot{y}}{\partial \bar{e}_u}\right] =$$

$$= \left[\frac{\partial f_{NL}}{\partial y}, \frac{\partial f_{NL}}{\partial u_0}, \ldots, \frac{\partial f_{NL}}{\partial u_{N-1}}, \frac{\partial f_{NL}}{\partial \bar{e}_y}, \frac{\partial f_{NL}}{\partial \bar{e}_u}\right] \cdot \begin{bmatrix} \frac{\partial y}{\partial y_0} & \frac{\partial y}{\partial u_0} & \ldots & \frac{\partial y}{\partial u_{N-1}} & \frac{\partial y}{\partial \bar{e}_y} & \frac{\partial y}{\partial \bar{e}_u} \\ 0 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & & & \vdots \\ \vdots & \vdots & & \ddots & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & 1 \end{bmatrix} \tag{6.31}$$

we can compute

$$z = [z_0, z_1, \ldots, z_{N+3}] = \left[\frac{\partial y}{\partial y_0}, \frac{\partial y}{\partial u_0}, \ldots, \frac{\partial y}{\partial u_{N-1}}, \frac{\partial y}{\partial \bar{e}_y}, \frac{\partial y}{\partial \bar{e}_u}\right]$$

in the discrete points $z(k)$ for $k = 0, \ldots, N$, hence at each k we can calculate

- $\frac{\partial y}{\partial y(0)}(k)$,
- $\frac{\partial y}{\partial \hat{u}(j)}(k)$ for $j = 0, \ldots, N-1$,

- $\frac{\partial y}{\partial \bar{e}_y}(k)$ and $\frac{\partial y}{\partial \bar{e}_u}(k)$.

At each iteration of the Gauss-Newton minimization (6.23) on $w_i = [\hat{y}(0), \hat{u}(0), \dots, \hat{u}(N-1), \bar{e}_y, \bar{e}_u]$ we need to calculate the Jacobian $J(w_i)$. To do so we must solve the variational equations with

$$z(w_i) = \left[\frac{\partial y}{\partial y_0}, \frac{\partial y}{\partial u_0}, \dots, \frac{\partial y}{\partial u_{N-1}}, \frac{\partial y}{\partial \bar{e}_y}, \frac{\partial y}{\partial \bar{e}_u}\right](w_i),$$

i.e. calculated in w_i, the i-th iteration of the optimization.

The initial conditions of the variable equation (6.31) are always $[1, 0, \dots, 0]$.

This is to be done for each value of the grid on the Λ's for the brute-force optimization, as in Algorithm 4.

6.4.2 Optimization with the adjoint-state method

We recall that we already introduced the adjoint-state method for the computation of the gradient in Section 3.3.1; here we will apply that approach to our problem.

We start from the formulation of [49] and then generalize it to our problem. The general basic formulation of the problem is

$$\begin{aligned} \min_{p} \quad & F(x,p) = \frac{1}{2}\sum_{i=0}^{N-1} \eta_i \left(s(x_i) - \bar{s}(t_i)\right)^2 = \frac{1}{2}\|r\|_2^2 \\ \text{s.t.} \quad & f(x_{i+1}, x_i, t_{i+1}, t_i, p) = 0 \quad \text{for } i = 0, 1, \dots, N-1, \\ & x_0 \text{ given} . \end{aligned} \tag{6.32}$$

In our case, with the notation of [49] and with

$$w = [\hat{y}(0), \hat{u}(0), \dots, \hat{u}(N-1), \bar{e}_y, \bar{e}_u] \in \mathbb{R}^{n_w}$$

as already defined above, we have

$$\begin{aligned} \min_{w} \quad F(w) = \frac{1}{2}\|q_\Lambda\|_2^2 = \frac{1}{2}\Big\{ & \sum_{i=0}^{N} (\lambda_{ey}^{min} \hat{e}_y(i))^2 + \sum_{i=0}^{N-1} (\lambda_{eu}^{min} \hat{e}_u(i))^2 + \\ & + \sum_{i=1}^{N-1} [\lambda_{ey}^{curv} (\hat{y}_{i-1} - 2\hat{y}_i + \hat{y}_{i+1})]^2 + \\ & + \sum_{i=1}^{N-2} [\lambda_{eu} (\hat{u}_{i-1} - 2\hat{u}_i + \hat{u}_{i+1}))]^2 \Big\} \end{aligned} \tag{6.33}$$

$$\text{s.t.} \qquad f_{k+1} = \hat{y}(k+1) - \hat{y}(k) - h * f_{NL}(\hat{y}(k), \hat{u}(k)) = 0 \quad \text{for } k = 0, \dots, N-1.$$

The Lagrangian function is

$$\mathcal{L}(y,w) = F(y,w) + \sum_{i=1}^{N} \lambda_i f_i(y(i), y(i-1), u(i-1)) \tag{6.34}$$

where λ_i are the *adjoint variables*.

We write more explicitly the components in the case of scalar input-output signals (SISO case), so that $\Lambda^{<min,curv>}_{<eu,ey>} = \lambda^{<min,curv>}_{<eu,ey>} I_{<n_u,n_y>}$

$$\begin{aligned} \mathcal{L} =& \frac{1}{2}\sum_{i=0}^{N-1}(\lambda_{ey}^{min}\hat{e}_y(i))^2 + \sum_{i=0}^{N}(\lambda_{eu}^{min}\hat{e}_u(i))^2 + \\ &+ \sum_{i=1}^{N-1}[\lambda_{ey}^{curv}(\hat{y}_{i-1} - 2\hat{y}_i + \hat{y}_{i+1})]^2 + \\ &+ \sum_{i=1}^{N-2}[\lambda_{eu}^{curv}(\hat{u}_{i-1} - 2\hat{u}_i + \hat{u}_{i+1}))]^2 + \\ &+ \sum_{i=0}^{N-1}\lambda_{i+1} f_{i+i}(\hat{y}(i+1), \hat{u}(i), \hat{y}(i)). \end{aligned} \tag{6.35}$$

The variation of the cost function (and of the Lagrangian) in this case is

$$\begin{aligned} \delta F = \delta\mathcal{L} =& 2 * \frac{1}{2}\sum_{i=0}^{N}\Big[(\lambda_{ey}^{min})^2\left(y_e(i) - \hat{y}(i) - \bar{e}_y\right)^T\left(-\delta\hat{y}(i) - \delta\bar{e}_y\right)\Big] + \\ &+ \sum_{i=0}^{N-1}\Big[(\lambda_{eu}^{min})^2\left(u_e(i) - \hat{u}(i) - \bar{e}_u\right)^T\left(-\delta\hat{u}(i) - \delta\bar{e}_u\right)\Big] + \\ &+ \sum_{i=1}^{N-1}\Big[(\lambda_{ey}^{curv})^2\left(\hat{y}(i-1) - 2\hat{y}(i) + \hat{y}(i+1)\right)^T\left(\delta\hat{y}(i-1) - 2\delta\hat{y}(i) + \delta\hat{y}(i+1)\right)\Big] + \\ &+ \sum_{i=1}^{N-2}\Big[(\lambda_{eu}^{curv})^2\left(\hat{u}(i-1) - 2\hat{u}(i) + \hat{u}(i+1)\right)^T\left(\delta\hat{u}(i-1) - 2\delta\hat{u}(i) + \delta\hat{u}(i+1)\right)\Big] + \\ &+ \sum_{i=0}^{N-1}\lambda_{i+1}\left(\frac{\partial f_{i+1}}{\partial\hat{y}(i+1)}\delta\hat{y}(i+1) + \frac{\partial f_{i+1}}{\partial\hat{y}(i)}\delta\hat{y}(i) + \frac{\partial f_{i+1}}{\partial\hat{u}(i)}\delta\hat{u}(i)\right). \end{aligned} \tag{6.36}$$

We can reformulate this as

$$\begin{aligned}\delta F = \delta\mathcal{L} = &\sum_{i=2}^{N-2}\left\{\frac{\partial\mathcal{L}}{\partial\hat{y}(i)}\delta\hat{y}(i) + \frac{\partial\mathcal{L}}{\partial\hat{u}(i)}\delta\hat{u}(i)\right\}+\\ &+\frac{\partial\mathcal{L}}{\partial\hat{y}(0)}\delta\hat{y}(0) + \frac{\partial\mathcal{L}}{\partial\hat{y}(1)}\delta\hat{y}(1) + \frac{\partial\mathcal{L}}{\partial\hat{y}(N-1)}\delta\hat{y}(N-1) + \frac{\partial\mathcal{L}}{\partial\hat{y}(N)}\delta\hat{y}(N)+\\ &+\frac{\partial\mathcal{L}}{\partial\hat{u}(0)}\delta\hat{u}(0) + \frac{\partial\mathcal{L}}{\partial\hat{u}(1)}\delta\hat{u}(1) + \frac{\partial\mathcal{L}}{\partial\hat{u}(N-1)}\delta\hat{u}(N-1)+\\ &+\frac{\partial\mathcal{L}}{\partial\bar{e}_y}\delta\bar{e}_y + \frac{\partial\mathcal{L}}{\partial\bar{e}_u}\delta\bar{e}_u.\end{aligned} \tag{6.37}$$

We recall from Section 3.3.1 that the adjoint equations are obtained from imposing

$$\frac{\partial\mathcal{L}}{\partial\hat{y}(i)} = 0 \qquad \text{for } i = 1,\dots,N.$$

We give now the explicit formulations of the partial derivatives. The derivatives with respect to the mean of the noisy signals are

$$\frac{\partial\mathcal{L}}{\partial\bar{e}_y} = -\sum_{i=0}^{N}(\lambda_{ey}^{min})^2\left(y_e(i) - \hat{y}(i) - \bar{e}_y\right)^T \tag{6.38}$$

$$\frac{\partial\mathcal{L}}{\partial\bar{e}_u} = -\sum_{i=0}^{N-1}(\lambda_{eu}^{min})^2\left(u_e(i) - \hat{u}(i) - \bar{e}_u\right)^T. \tag{6.39}$$

For $i = 2,\dots,N-2$ it holds

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\hat{y}(i)} =&\Big[(\lambda_{ey}^{min})^2\left(y_e(i) - \hat{y}(i) - \bar{e}_y\right)^T\frac{-\partial\hat{y}(i)}{\partial\hat{y}(i)}+\\ &+(\lambda_{ey}^{curv})^2\big[\left(\hat{y}(i-2) - 2\hat{y}(i-1) + \hat{y}(i)\right) - 2\left(\hat{y}(i-1) - 2\hat{y}(i) + \hat{y}(i+1)\right)+\\ &+\left(\hat{y}(i) - 2\hat{y}(i+1) + \hat{y}(i+2)\right)\big]+\\ &+\lambda_i\frac{\partial f_i}{\partial\hat{y}(i)} + \lambda_{i+1}\frac{\partial f_{i+1}}{\partial\hat{y}(i)}\Big]\end{aligned}$$

while for the remaining derivatives ($i = 0, 1, N-1, N$)

$$\frac{\partial \mathcal{L}}{\partial \hat{y}(0)} = -(\lambda_{ey}^{min})^2 \tilde{e}_y(0) + (\lambda_{ey}^{curv})^2 (\hat{y}(0) - 2\hat{y}(1) + \hat{y}(2)) + \lambda_1^T \frac{\partial f_1}{\partial \hat{y}(0)}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{y}(1)} =& -(\lambda_{ey}^{min})^2 \tilde{e}_y(1) + (\lambda_{ey}^{curv})^2 \big[-2(\hat{y}(0) - 2\hat{y}(1) + \hat{y}(2)) + \\ &+ (\hat{y}(1) - 2\hat{y}(2) + \hat{y}(3))\big] + \\ &+ \lambda_1^T \frac{\partial f_1}{\partial \hat{y}(1)} + \lambda_2^T \frac{\partial f_2}{\partial \hat{y}(1)}\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{y}(N-1)} =& -(\lambda_{ey}^{min})^2 \tilde{e}_y(N-1) + \\ &+ (\lambda_{ey}^{curv})^2 \big[(\hat{y}(N-3) - 2\hat{y}(N-2) + \hat{y}(N-1)) + \\ &- 2(\hat{y}(N-2) - 2\hat{y}(N-1) + \hat{y}(N))\big] + \\ &+ \lambda_{N-1}^T \frac{\partial f_{N-1}}{\partial \hat{y}(N-1)} + \lambda_N^T \frac{\partial f_N}{\partial \hat{y}(N-1)}\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{y}(N)} =& -(\lambda_{ey}^{min})^2 \tilde{e}_y(N) + \\ &+ (\lambda_{ey}^{curv})^2 (\hat{y}(N-2) - 2\hat{y}(N-1) + \hat{y}(N)) + \\ &+ \lambda_N^T \frac{\partial f_N}{\partial \hat{y}(N)}.\end{aligned}$$

Analogously for the partial derivatives with respect to the input signal $\hat{u}$ we have for $i = 2, \ldots, N-3$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{u}(i)} =& \Big[(\lambda_{eu}^{min})^2 \, (u_e(i) - \hat{u}(i) - \bar{e}_u)^T \, \frac{-\partial \hat{u}(i)}{\partial \hat{u}(i)} + \\ &+ (\lambda_{eu}^{curv})^2 \big[\, (\hat{u}(i-2) - 2\hat{u}(i-1) + \hat{u}(i)) - 2\,(\hat{u}(i-1) - 2\hat{u}(i) + \hat{u}(i+1)) + \\ &+ (\hat{u}(i) - 2\hat{u}(i+1) + \hat{u}(i+2)) \, \big] + \\ &+ \lambda_{i+1} \frac{\partial f_{i+1}}{\partial \hat{u}(i)} \Big]\end{aligned}$$

and the remaining derivatives ($i = 0, 1, N-2, N-1$)

$$\frac{\partial \mathcal{L}}{\partial \hat{u}(0)} = -(\lambda_{eu}^{min})^2 \tilde{e}_u(0) + (\lambda_{eu}^{curv})^2 (\hat{u}(0) - 2\hat{u}(1) + \hat{u}(2)) + \lambda_1^T \frac{\partial f_1}{\partial \hat{u}(0)}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{u}(1)} = &-(\lambda_{eu}^{min})^2 \tilde{e}_u(1) + \\ &+ (\lambda_{eu}^{curv})^2 \big[-2(\hat{u}(0) - 2\hat{u}(1) + \hat{u}(2)) + \\ &+ (\hat{u}(1) - 2\hat{u}(2) + \hat{u}(3)) \big] + \\ &+ \lambda_2^T \frac{\partial f_2}{\partial \hat{u}(1)}\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{u}(N-2)} = &-(\lambda_{eu}^{min})^2 \tilde{e}_u(N-1) + \\ &+ (\lambda_{eu}^{curv})^2 \big[-2(\hat{u}(N-3) - 2\hat{u}(N-2) + \hat{u}(N-1)) + \\ &(\hat{u}(N-4) - 2\hat{u}(N-3) + \hat{u}(N-2)) \big] + \\ &+ \lambda_N^T \frac{\partial f_N}{\partial \hat{u}(N-1)}\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{u}(N-1)} = &-(\lambda_{eu}^{min})^2 \tilde{e}_u(N-1) + \\ &+ (\lambda_{eu}^{curv})^2 (\hat{u}(N-3) - 2\hat{u}(N-2) + \hat{u}(N-1)) + \\ &+ \lambda_N^T \frac{\partial f_N}{\partial \hat{u}(N-1)}.\end{aligned}$$

Unlike the problem of [48], in our case y_0 is not given, hence $\delta\hat{y}(0)$ is not zero.

Note that this approach allows to avoid the computation of the partial derivative $\frac{\partial \hat{y}(i)}{\partial \hat{u}(i)}$ that was instead calculated in the sensitivity equations approach.

Recalling the equation of the Lagrangian function as in (6.34), the adjoint equations in this case can be written as

$$\begin{cases} \left(\frac{\partial f_i}{\partial \hat{y}(i)}\right)^T \lambda_i = -\left(\frac{\partial F}{\partial \hat{y}(i)}\right)^T - \left(\frac{\partial f_{i+1}}{\partial \hat{y}(i)}\right)^T \lambda_{i+1} & \text{for} \quad i = 1, \dots, N-1. \\ \left(\frac{\partial f_i}{\partial \hat{y}(i)}\right)^T \lambda_i = -\left(\frac{\partial F}{\partial \hat{y}(i)}\right)^T & \text{for} \quad i = N \end{cases} \tag{6.40}$$

that are N equations in N unknowns $\lambda_1, \dots, \lambda_N$.

In the matrix notation we see this is a bidiagonal linear system

$$\begin{bmatrix} \frac{\partial f_1}{\partial \hat{y}(1)} & \frac{\partial f_2}{\partial \hat{y}(1)} & 0 & & 0 \\ 0 & \frac{\partial f_2}{\partial \hat{y}(2)} & \frac{\partial f_3}{\partial \hat{y}(2)} & & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & & \frac{\partial f_{N-1}}{\partial \hat{y}(N-1)} & \frac{\partial f_N}{\partial \hat{y}(N-1)} \\ 0 & & & 0 & \frac{\partial f_N}{\partial \hat{y}(N)} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \\ \lambda_{N-1} \\ \lambda_N \end{bmatrix} = \begin{bmatrix} -\frac{\partial F}{\partial \hat{y}(1)} \\ -\frac{\partial F}{\partial \hat{y}(2)} \\ \vdots \\ -\frac{\partial F}{\partial \hat{y}(N-1)} \\ -\frac{\partial F}{\partial \hat{y}(N)} \end{bmatrix} \tag{6.41}$$

The Jacobian matrices of the adjoint equation depend on the solution of the system, hence there are two steps, first the solution of the system and then the solution in reverse order of the linear adjoint equations, as summarized in Algorithm 5.

For this optimization, authors in [49] and [72] use quasi-Newton methods, the first approximating the Hessian with BFGS formula and the second with Davidon-Fletcher-Powell (DFP) method.

Algorithm 5 Solution of the nonlinear problem (6.33) with the Adjoint-state method for fixed values of $\lambda_{<eu,ey>}^{<min,curv>}$

```
1: Initialization of the parameters $w_0 = [y(0), u(0), \dots, u(N-1), \bar{e}_y, \bar{e}_u]$
2: for $k = 1, \dots, K_{maxiter}$ do
3:     0) given the parameters $w_{k-1}$ of the $(k-1)$-th iteration of the optimization
4:     1) solve the discretized dynamical system to obtain the values of $\hat{y}$
5:     2) solve the bidiagonal linear system (6.41) and obtain the adjoint variables $\lambda_1, \dots, \lambda_N$
6:     3) use the values of $\hat{y}$ and the adjoint variables $\lambda_i$ to compute the gradient
7:     4) calculate the new parameters for the new iteration $w_k$ of the optimization
8: end for
```

6.4.3 Comparison between the calculation of the gradient with the sensitivity equation and the adjoint method

- We recall that the additional cost for the computation of the gradient with the sensitivity approach is proportional to the number of the parameters, while in the adjoint method it does not depend on it. Hence, for problems with a high number of parameters the adjoint method is recommended. The denoising problem here considered has a big number of parameters (the value of the input variables for each time step), and so the adjoint method provide a solution with a smaller computational cost.

- Moreover, the sensitivity approach results in the calculation of the Jacobian matrix of the residual q_Λ from which it is possible to calculate the gradient of the cost function as $\frac{\partial F}{\partial w} = J^T q_\Lambda$; while the adjoint method yields only the gradient $\frac{\partial F}{\partial w}$.

6.4.4 Numerical experiments

We do not present a complete set of tests as in the linear case of the previous Chapter 5, since the extension of Algorithm 4 as it is, for nonlinear models did not give good results in all cases and the computation cost was heavy. Hence, more work is needed to obtain a satisfying method.

However, we show the results on a fixed example, with regularization parameters fixed a-priori.

For the example we chose the nonlinear equation

$$f_{NL}^{c}(u,y) = -y^2 + 40 * u^2$$

and integrated the ODE $\dot{y} = f_{NL}^{c}(y,u)$ with Explicit Euler with time step $h = 0.0001$ for $N = 40$ time samples. The discrete input signal is

$$u(k) = sin(0.01 * k)$$

The noisy input/output measurements have been created adding a white noise term with standard deviation $\sigma = 0.4$. In this example the noises are supposed zero-mean white noises and the means $\bar{e}_u, \bar{e}_y$ are removed from the unknown vector of problem (6.19), which is now simplified to

$$w = [y(0), u(0), \dots, u(N-1)].$$

We will compare the results for two set of parameters fixed equal to

$$(\lambda_{eu}^{min}, \lambda_{ey}^{min}, \lambda_{eu}^{curv}, \lambda_{ey}^{curv}) = (1, 1, 50, 1)$$

and

$$(\lambda_{eu}^{min}, \lambda_{ey}^{min}, \lambda_{eu}^{curv}, \lambda_{ey}^{curv}) = (1, 1, 10, 1)$$

to show that the optimal set of parameters is the one that gives the best recovering of the true signals.

The initial value w_0 is initialized with the measured values of input and output signals for each case, i.e.

$$w_0 = [y^{meas}(0), u^{meas}(0), \dots, u^{meas}(N-1)].$$

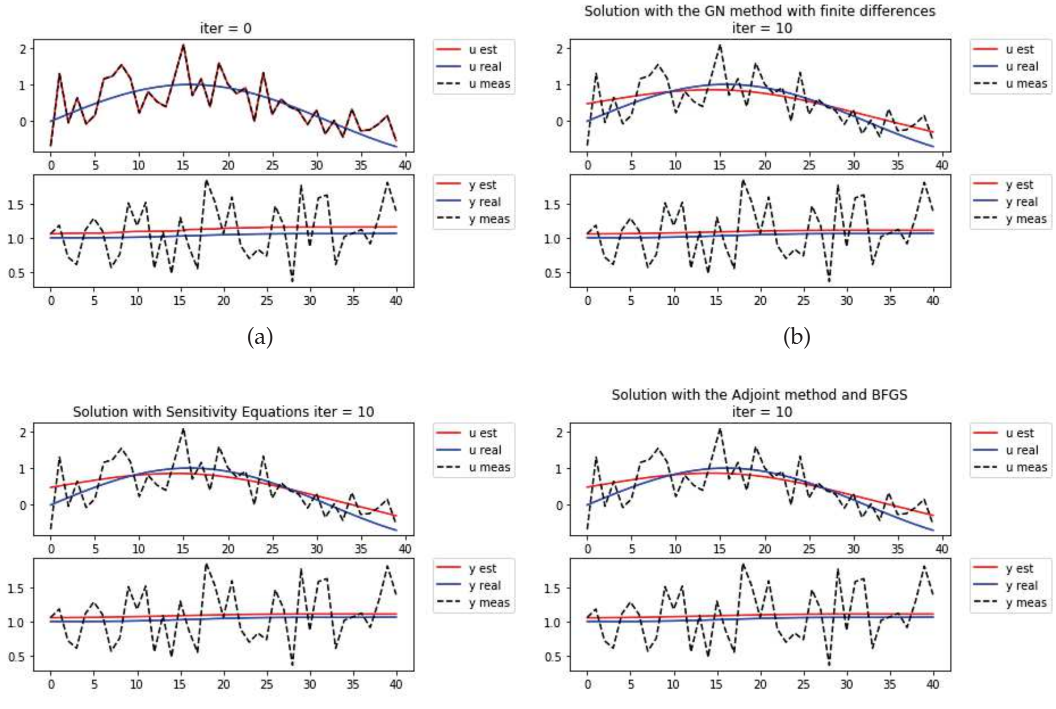
We will compare three methods for the computation of the denoised signals, fixed the two sets of parameters λ:

- Gauss-Newton method with J calculated with the finite difference method the gradient $grad = J^T q_\lambda$ and the approximated Hessian $H = J^T J$ as in the definition of the Gauss-Newton method (equations (3.1) and (3.2)),

- Gauss-Newton method with J calculated with the sensitivity equations, the gradient $grad = J^T q_\lambda$ and the approximated Hessian $H = J^T J$ as in the definition of the Gauss-Newton method (equations (3.1) and (3.2)),

- Gauss-Newton method with the gradient calculated with the adjoint method and the hessian calculated with BFGS method.

The methods for gradient computation of sensitivity equations and adjoint method for this problem have been implemented in `Python`, following the derivations obtained in the previous Sections, and validated with the finite difference method.



(a)

(b)

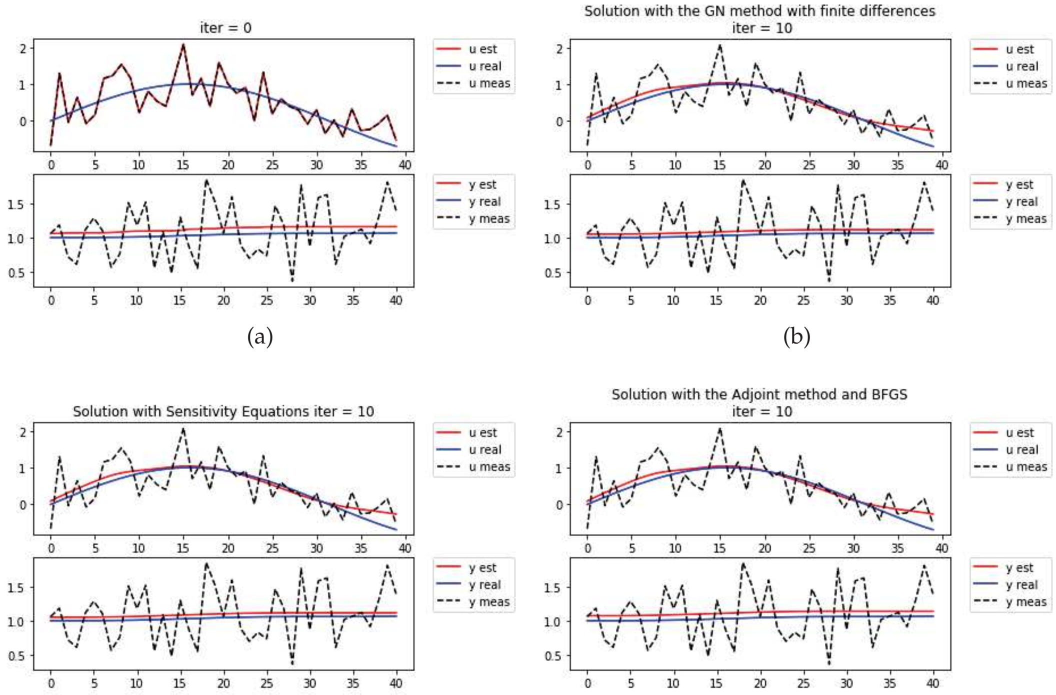
(c)

(d)

Figure 6.1: Results obtained with fixed parameters $(\lambda_{eu}^{min}, \lambda_{ey}^{min}, \lambda_{eu}^{curv}, \lambda_{ey}^{curv}) = (1, 1, 50, 1)$. **(a)** Initial condition with guess $w_0 = [y^{meas}(0), u^{meas}(0), \ldots, u^{meas}(N-1)]$ coincident with the measured values of input and initial condition of the output; **(b)** Iteration 10 of Gauss-Newton optimization method with finite differences for the calculation of J; **(c)** Iteration 10 of Gauss-Newton optimization method with sensitivity equations for the calculation of J; **(d)** Iteration 10 of Gauss-Newton optimization method with Adjoint method for the calculation of J and BFGS for the calculation of the Hessian.

We can see from the results that both methods, starting from an initial guess of the unknown parameter vector $w_0 = [y^{meas}(0), u^{meas}(0), \ldots, u^{meas}(N-1)]$, are able to find an estimate of the input/output signals near the true values. However, the performance of the result depends on the chosen parameters λ, and can be improved.

In the Tables 6.1 and 6.2 the comparisons of relative errors and whiteness values



(a) (b) (c) (d)

Figure 6.2: Results obtained with fixed parameters $(\lambda_{eu}^{min}, \lambda_{ey}^{min}, \lambda_{eu}^{curv}, \lambda_{ey}^{curv}) = (1, 1, 10, 1)$. **(a)** Initial condition with guess $w_0 = [y^{meas}(0), u^{meas}(0), \ldots, u^{meas}(N-1)]$ coincident with the measured values of input and initial condition of the output; **(b)** Iteration 10 of Gauss-Newton optimization method with finite differences for the calculation of J; **(c)** Iteration 10 of Gauss-Newton optimization method with sensitivity equations for the calculation of J; **(d)** Iteration 10 of Gauss-Newton optimization method with Adjoint method for the calculation of J and BFGS for the calculation of the Hessian.

of the noise signals are shown for the true noise vector and the estimated ones, for the two choices of fixed parameter values respectively. Moreover, we can see that in the second case, the relative errors of the estimated input/output values are smaller and the whiteness functions are closer to the true values.

data	re_u	re_y	w_{eu}	w_{ey}
Real signals	0.603	0.337	0.485	0.288
estimate GN FD	0.324	0.052	0.354	0.294
estimate SE	0.324	0.052	0.354	0.294
estimate ADJ BFGS	0.329	0.052	0.361	0.293

Table 6.1: Comparisons of relative errors *re* of the input/output signals and whiteness functions of the input/output noises for the case with $\lambda_{eu}^{curv} = 50$. The values are computed for the real noise, and the estimated values with 10 iterations of the three methods described: Gauss-Newton with Finite Differences (GN FD), Gauss-Newton with Sensitivity Equations (SE) and Gauss-Newton with Adjoint method and BFGS (ADJ BFGS).

data	re_u	re_y	w_{eu}	w_{ey}
Real signals	0.603	0.337	0.485	0.288
estimate GN FD	0.218	0.052	0.523	0.294
estimate SE	0.218	0.052	0.523	0.294
estimate ADJ BFGS	0.219	0.075	0.523	0.290

Table 6.2: Comparisons of relative errors *re* of the input/output signals and whiteness functions of the input/output noises for the case with $\lambda_{eu}^{curv} = 10$. The values are computed for the real noise, and the estimated values with 10 iterations of the three methods described: Gauss-Newton with Finite Differences (GN FD), Gauss-Newton with Sensitivity Equations (SE) and Gauss-Newton with Adjoint method and BFGS (ADJ BFGS).

6.4.5 Conclusions and future work

As already said, the problem is still open and the algorithm for the optimization of the parameters λ should be improved, to give a complete set of tests for different ODEs examples. Moreover, the same open problems of the linear case, highlighted in the conclusions of Chapter 5, should be studied for this problem.

Acknowledgments

I wish to thank many people who contributed to my doctoral path.

First of all my supervisor Prof. Fabio Marcuzzi, who believed in me and has always been supportive and positive, giving me a lot of ideas to think, study and work on.

I want to thank Electrolux for the financial opportunity, and the group of GTD CA&CCF for their welcome in the industry and the possibility to work on the applied problems of modeling.

Thanks to all my family for their presence and the kindness they always have showed me, and for the support in the academic and personal experiences of these years.

Last but not least, I thank all my friends and colleagues, to which goes my esteem, with whom I have shared these years, and that taught me a lot with their example.

Bibliography

[1] P. Amodio, J. R. Cash, G. Roussos, R. W. Wright, G. Fairweather, I. Gladwell, G. L. Kraut, and M. Paprzycki. Almost block diagonal linear systems: sequential and parallel solution techniques, and applications. *Numerical Linear Algebra with Applications*, 7(5):275–317, 2000.

[2] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Society for Industrial and Applied Mathematics, 1995.

[3] A. Bakushinsky. Remarks on choosing a regularization parameter using the quasioptimality and ratio criterion. *U.S.S.R. Comput. Math. Math. Phys.*, 24:181–182, 1984.

[4] A. Beghi, F. Marcuzzi, P. Martin, F. Tinazzi, and M. Zigliotto. Virtual prototyping of embedded control software in mechatronic systems: A case study. *Mechatronics*, 43:99 – 111, 2017.

[5] A. Beghi, F. Marcuzzi, and M. Rampazzo. A virtual laboratory for the prototyping of cyber-physical systems. *IFAC-PapersOnLine*, 49(6):63 – 68, 2016.

[6] A. Beghi, F. Marcuzzi, M. Rampazzo, and M. Virgulin. Enhancing the simulation-centric design of cyber-physical and multi-physics systems through co-simulation. In *2014 17th Euromicro Conference on Digital System Design*, pages 687–690, Aug 2014.

[7] M. Belge, M. E. Kilmer, and E. L. Miller. Efficient determination of multiple regularization parameters in a generalized L-curve framework. *Inverse Problems*, 18(4):1161–1183, jul 2002.

[8] A. Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, 1996.

[9] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, USA, 2004.

[10] C. Brezinski, M. Redivo-Zaglia, G. Rodriguez, and S. Seatzu. Multi-parameter regularization techniques for ill-conditioned linear systems. *Numerische Mathematik*, 94:203–228, 2003.

[11] J. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, June 2003.

[12] J. V. Candy. *Model-Based Signal Processing*. Wiley IEEE Press, 2005.

[13] J. V. Candy. *Model-Based Processing*. John Wiley & Sons, Ltd, 2019.

[14] J. V. Candy. *Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods, 2nd Edition*. Wiley-IEEE Press, July 2016.

[15] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, Jan 2004.

[16] G. Chavent. *Nonlinear Least Squares for Inverse Problems*. Springer Netherlands, 2010.

[17] Y. Chen, M. Akutagawa, Y. Kinouchi, and Q. Zhang. Neural network based audio signal denoising. In *Proceedings of the 2008 International Conference on Advanced Infocomm Technology*, ICAIT '08, pages 54:1–54:6, New York, NY, USA, 2008. ACM.

[18] M. Davies. Noise reduction schemes for chaotic time series. *Physica D: Nonlinear Phenomena*, 79(2):174 – 192, 1994.

[19] G. Deolmi and F. Marcuzzi. A parabolic inverse convection–diffusion–reaction problem solved using space–time localization and adaptivity. *Applied Mathematics and Computation*, 219(16):8435 – 8454, 2013.

[20] R. Diversi, R. Guidorzi, and U. Soverini. Algorithms for optimal errors-in-variables filtering. *Systems & Control Letters*, 48(1):1 – 13, 2003.

[21] M. Fornasier, V. Naumova, and S. V. Pereverzyev. Parameter choice strategies for multipenalty regularization. *SIAM J. Numerical Analysis*, 52:1770–1794, 2014.

[22] M. Gatto and F. Marcuzzi. An algorithm for model-based denoising of input-output data. *Dolomites Research Notes on Approximation*, 12(1):73–85, 2019.

[23] M. Gatto and F. Marcuzzi. Unbiased least-squares modelling. *MDPI Mathematics*, 8(6):982, 2020.

[24] S. Gazzola and L. Reichel. A new framework for multi-parameter regularization. *BIT Numerical Mathematics*, 56(3):919–949, 9 2016.

[25] G. H. Golub. Some modified matrix eigenvalue problems. *SIAM Review*, 15(2):318–334, 1973.

[26] G. H. Golub and C. F. V. Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, 1980.

[27] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, USA, 1996.

[28] R. Guidorzi, R. Diversi, and U. Soverini. Optimal errors-in-variables filtering. *Automatica*, 39(2):281 – 289, 2003.

[29] A. Haddad and Y. Meyer. An improvement of rudin–osher–fatemi model. *Applied and Computational Harmonic Analysis*, 22(3):319 – 334, 2007.

[30] F. Hamilton. Parameter estimation in differential equations: A numerical study of shooting methods. *SIAM Undergraduate Research Online*, Volume 4, 02 2011.

[31] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Society for Industrial and Applied Mathematics, USA, 1999.

[32] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2010.

[33] P. C. Hansen. Oblique projections and standard-form transformations for discrete inverse problems. *Numerical Linear Algebra with Applications*, 20(2):250–258, 2013.

[34] P. C. Hansen, M. E. Kilmer, and R. H. Kjeldsen. Exploiting residual information in the parameter choice for discrete ill-posed problems. *BIT Numerical Mathematics*, 46(1):41–59, Mar 2006.

[35] A. N. Heinz Werner Engl, Martin Hanke. *Regularization of Inverse Problems*. Springer, 2000.

[36] R. J. Hodrick and E. C. Prescott. Postwar u.s. business cycles: An empirical investigation. *Journal of Money, Credit and Banking*, 29(1):1–16, 1997.

[37] G. Holler, K. Kunisch, and R. C. Barnard. A bilevel approach for parameter learning in inverse problems. *Inverse Problems*, 34(11):115012, sep 2018.

[38] A. Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2 edition, 2008.

[39] K. Ito, B. Jin, and T. Takeuchi. Multi-parameter tikhonov regularization. *Methods and Applications of Analysis*, 18:31–46, 2011.

[40] T. Kailath. *Linear Systems*. Prentice-Hall, 1980.

[41] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Applied Mathematical Sciences. Springer-Verlag New York, 2005.

[42] M. Kern. *Numerical Methods for Inverse Problems*. John Wiley & Sons, Ltd, 2016.

[43] M. Kilmer, P. Hansen, and M. Espanol. A projection-based approach to general-form tikhonov regularization. *S I A M Journal on Scientific Computing*, 29(1):315–330, 2007.

[44] S. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009.

[45] S. Kindermann. Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems. *Electronic Transactions on Numerical Analysis*, 38:233–257, 2011.

[46] E. J. Kostelich and T. Schreiber. Noise reduction in chaotic time-series data: A survey of common methods. *Phys. Rev. E*, 48:1752–1763, Sep 1993.

[47] K. Kunisch and T. Pock. A bilevel optimization approach for parameter learning in variational models. *SIAM J. Imaging Sciences*, 6:938–983, 2013.

[48] T. Lauß, S. Oberpeilsteiner, W. Steiner, and et al. The discrete adjoint method for parameter identification in multibody system dynamics. *Multibody Syst Dyn*, 42:397–410, 2018.

[49] T. Lauß, S. Oberpeilsteiner, W. Steiner, and K. Nachbagauer. The discrete adjoint method for parameter identification in multibody system dynamics. *Multibody System Dynamics*, 42(4):397–410, Apr 2018.

[50] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, 1995.

[51] Z. Liu, A. Rantzer, and L. Vandenberghe. Nuclear norm system identification with missing inputs and outputs. *Systems & Control Letters*, 62:605–612, 2013.

[52] L. Ljung. Model validation and model error modeling. Technical Report 2125, Linköping University, The Institute of Technology, Automatic Control, 1999.

[53] D. Lorenz and T. Köhler. *A Comparison of Denoising Methods for One Dimensional Time Series*. DFG-Schwerpunktprogramm 1114, Mathematical methods for time series analysis and digital image processing. Zentrum für Technomathematik, 2005.

[54] S. Lu, S. Pereverzev, Y. Shao, and et al. Discrepancy curves for multi-parameter regularization. *Journal of Inverse and Ill-posed Problems*, 18(6):pp. 655–676, 2010.

[55] M. Mafi, H. Martin, M. Cabrerizo, J. Andrian, A. Barreto, and M. Adjouadi. A comprehensive survey on impulse and gaussian denoising filters for digital images. *Signal Processing*, 157:236 – 260, 2019.

[56] F. Marcuzzi. Space and time localization for the estimation of distributed parameters in a finite element model. *Computer Methods in Applied Mechanics and Engineering*, 198(37):3020 – 3025, 2009.

[57] F. Marcuzzi. Linear estimation of physical parameters with subsampled and delayed data. *Journal of Computational and Applied Mathematics*, 331:11 – 22, 2018.

[58] I. Markovsky and B. De Moor. Technical communique: Linear dynamic filtering with noisy input and output. *Automatica*, 41(1):167–171, Jan. 2005.

[59] I. Markovsky and B. D. Moor. Linear dynamic filtering with noisy input and output. *IFAC Proceedings Volumes*, 36(16):1711 – 1716, 2003. 13th IFAC Symposium on System Identification (SYSID 2003), Rotterdam, The Netherlands, 27-29 August, 2003.

[60] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, USA, 2000.

[61] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

[62] C. C. Peck, S. L. Beal, L. B. Sheiner, and A. I. Nichols. Extended least squares nonlinear regression: A possible solution to the "choice of weights" problem in analysis of individual pharmacokinetic data. *Journal of Pharmacokinetics and Biopharmaceutics*, 12(5):545–558, Oct 1984.

[63] T. Raus and U. Hämarik. Heuristic parameter choice in tikhonov method from minimizers of the quasi-optimality function, 2017.

[64] L. Reichel and G. Rodriguez. Old and new parameter choice rules for discrete ill-posed problems. *Numerical Algorithms*, 63:65–87, 2013.

[65] W. Reinelt, A. Garulli, and L. Ljung. Comparing different approaches to model error modeling in robust identification. *Automatica*, 38(5):787–803, May 2002.

[66] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259 – 268, 1992.

[67] M. U. Sadiq, J. P. Simmons, and C. A. Bouman. Model based image reconstruction with physics based priors. *2016 IEEE International Conference on Image Processing (ICIP)*, 2016.

[68] R. Sameni. Online filtering using piecewise smoothness priors: Application to normal and abnormal electrocardiogram denoising. *Signal Processing*, 133:52 – 63, 2017.

[69] T. Söderström, U. Soverini, and K. Mahata. Perspectives on errors-in-variables estimation for dynamic systems. *Signal Processing*, 82(8):1139 – 1154, 2002.

[70] T. Söderström, R. Diversi, and U. Soverini. A unified framework for eiv identification methods in the presence of mutually correlated noises. *IFAC Proceedings Volumes*, 47(3):4644 – 4649, 2014. 19th IFAC World Congress.

[71] K. J. Åström and B. Torsten. Numerical identification of linear dynamic systems from normal operating records. *IFAC Proceedings Volumes*, 2(2):96 – 111, 1965. 2nd IFAC Symposium on the Theory of Self-Adaptive Control Systems, Teddington, UK, September 14-17, 1965.

[72] Y. Sun, Z. Liu, and Y. Hu. Adjoint based state estimation of compressible flow in porous media. *Petroleum*, 2020.

[73] A. Tikhonov and V. Glasko. Use of the regularization method in non-linear problems. *USSR Computational Mathematics and Mathematical Physics*, 5(3):93 – 107, 1965.

[74] S. Van Huffel, I. Markovsky, R. J. Vaccaro, and T. Söderström. Total least squares and errors-in-variables modeling. *Signal Processing*, 87(10):2281 – 2282, 2007. Special Section: Total Least Squares and Errors-in-Variables Modeling.

[75] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 1991.

[76] P. Van Overschee and B. L. De Moor. *Subspace Identification for Linear Systems, Theory - Implementation - Applications*. Springer US, 1996.

[77] M. Verhaegen and V. Verdult. *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2007.

[78] A. Vlasenko and C. Schnorr. Physically consistent and efficient variational denoising of image fluid flow estimates. *IEEE Transactions on Image Processing*, 19(3):586–595, March 2010.

[79] M. Zhu, S. J. Wright, and T. F. Chan. Duality-based algorithms for total-variation-regularized image restoration. *Computational Optimization and Applications*, 47(3):377–400, Nov 2010.