Thesis for the Degree of Doctor of Philosophy

Inference techniques for stochastic nonlinear system identification with applications to the Wiener-Hammerstein models

Giuseppe Giordano

CHALMERS

Department of Electrical Engineering
Chalmers University of Technology

Göteborg, Sweden 2018

Inference techniques for stochastic nonlinear system identification with applications to the Wiener-Hammerstein models

Giuseppe Giordano

ISBN: 978-91-7597-790-4

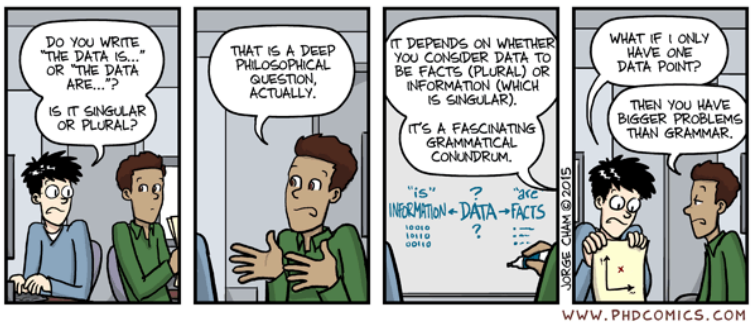
Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 4471
ISSN 0346-718X

Department of Electrical Engineering
Division of Systems and Control, Mechatronics Group
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone: +46 (0)31 – 772 1000
Email: giuseppe.giordano@chalmers.se; giordano.gius@gmail.com

Typeset by the author using LaTeX.

Chalmers Reproservice
Göteborg, Sweden 2018

Alla mia famiglia



WWW.PHDCOMICS.COM

Abstract

Stochastic nonlinear systems are a specific class of nonlinear systems where unknown disturbances affect the system's output through a nonlinear transformation. In general, the identification of parametric models for this kind of systems can be very challenging. A main statistical inference technique for parameter estimation is the Maximum Likelihood estimator. The central object of this technique is the likelihood function, i.e. a mathematical expression describing the probability of obtaining certain observations for given values of the parameter. For many stochastic nonlinear systems, however, the likelihood function is not available in closed-form. Several methods have been developed to obtain approximate solutions to the Maximum Likelihood problem, mainly based on the Monte Carlo method. However, one of the main difficulties of these methods is that they can be computationally expensive, especially when they are combined with numerical optimization techniques for likelihood maximisation.

This thesis can be divided in three parts. In the first part, a background on the main statistical techniques for parameter estimation is presented. In particular, two iterative methods for finding the Maximum Likelihood estimator are introduced. They are the gradient-based and the Expectation-Maximisation algorithms.

In the second part, the main Monte Carlo methods for approximating the Maximum Likelihood problem are analysed. Their combination with gradient-based and Expectation-Maximisation algorithms is considered. For ensuring convergence, these algorithms require the use of enormous Monte Carlo effort, i.e. the number of random samples used to build the Monte Carlo estimates. In order to reduce this effort and make the algorithms usable in practice, iterative solutions alternating *local* Monte Carlo estimates and maximisation steps are derived. In particular, a procedure implementing an efficient samples simulation across the steps of a Newton's method is developed. The procedure is based on the sensitivity of the parameter search with respect to the Monte Carlo samples and it results into an accurate and fast algorithm for solving the MLE problem.

The considered Maximum Likelihood estimation methods proceed through local explorations of the parameter space. Hence, they have guaranteed convergence only to a local optimizer of the likelihood function. In the third part of the thesis, this issue is addressed by deriving initialization algorithms. The purpose is to generate initial guesses that increase the chances of con-

verging to the global maximum. In particular, initialization algorithms are derived for the Wiener-Hammerstein model, i.e. a nonlinear model where a static nonlinearity is sandwiched between two linear dynamical systems. For this type of model, it can be proved that the best linear approximation of the system provides a consistent estimate of the concatenation of the linear dynamics. Based on this result, two main initialization algorithms are derived. The first one is the *Exhaustive Search* approach, where all the combinations of the dynamics, expressed in terms of poles and zeros, of the best linear approximation are tested as initial guesses of a Maximum Likelihood estimation problem in the parameters of the nonlinearity. The main drawback of this approach resides in its combinatorial complexity in the number of poles and zeros of the linear approximation. The second one is the *Expanded Fractional Approach*, i.e. an improvement of the original Fractional Approach. In the original approach, the estimated dynamics are parametrized in a fractional way and only one optimization problem retrieves the initial guess for the linear dynamics. However, ill-conditioning problems can arise from specific configurations of the dynamics of the linear parts. With the Expanded Fractional Approach, the ill-conditioning issue is addressed and solved via series expansion of the fractional parametrization. Furthermore, a *lifted* formulation of the optimization problem resulting from the Expanded Fraction Approach allows for a faster convergence when using Newton-type methods.

Keywords: system identification, Maximum Likelihood, nonlinear stochastic models, Monte Carlo, Newton's method, initialization algorithm, Wiener-Hammerstein

Preface

This thesis is in partial fulfillment for the degree of Doctor of Philosophy at Chalmers University of Technology, Gothenburg, Sweden.

Acknowledgments

With this thesis, an important chapter of my life comes to an end. It is then time to sit down for a moment and think about all the people who, in a way or another, have contributed to write this chapter with me. I will try to extend my gratitude to all of them.

My first thanks goes to my Master thesis supervisor, Professor Luigi Iannelli, who helped me finding the Ph.D. position at Chalmers, and to the control group at the University of Sannio in Benevento.

I am deeply grateful to my Ph.D. supervisor, Professor Jonas Sjöberg, for giving me the opportunity to work on a very interesting and challenging research topic. Thanks for all the fruitful discussions, technical and not, that helped me growing as a researcher and, more importantly, as a person.

A great thanks goes to my Ph.D. co-supervisor, Professor Sébastien Gros, for helping me in many occasions, for always being available for discussions and coffee breaks, and for always pushing me to the deep understanding of the true reason behind every aspect of my research.

I am thankful to Professor Paolo Falcone, who firstly introduced me to Chalmers and to the research environment. Thanks for always giving me precious advice and support.

A huge thanks goes to my first-ever Ph.D. colleague, my first-ever office-mate, and my great friend, Robert. Thanks for all the funny moments, but also for the serious discussions. Directly and not, you have been very helpful over these five years.

I am equally thankful to Mario, who joined the Mechatronics group as a post-doc and who supported and helped me with my research. Thanks also for all the non-work related, mainly food-based, projects and, above all, for the great friend you are now.

A big thanks goes to my other post-doc friend Josip, for all the coffee breaks and the nice discussions we had during his time at Chalmers. Thanks to Elena, always supportive when it came to coffee breaks and lunches in the Swedish sun.

Thanks to all the other colleagues, post-docs, professors, and staff who have been or are still part of the System and Control Division, and to all

the other researchers I met during the conferences. In particular, thanks to Hakan, Azita, Mitra, Sahar, Roozbeh, Maliheh, Christine, Yujiao, Yiannis, Balázs, Simon, Emil, Mohammed, Fredrik, Ankit, Ivo, Angelos, Ramin, Julio, Mohammad, Martin, Sabino, Sarmad, Boaz, and Giulio.

During these five years in Göteborg, I had the luck to get to know very nice people who then became what I consider my family in Sweden. I want to thank you all because, in a way or another, you have contributed to this achievement of mine. Hence, a great thanks goes to Livia, Gabriel, and Carlo, for always being there for me since the beginning of this experience, and to Riccardo and Flavia, for all the good dinners, the nice talks, and the all the moments spent together. To all of you, thanks for your friendship. And then Madeleine, Marie, Gabo, Sadegh, Gerardo, Francesca, Oana, Roberto, Rossella, and the little Alessia. Thank you for everything.

A special thanks goes to Salvatore, for being the other *cerratano* in Göteborg and a great friend, and to Riccardo and Marco, for sharing the fantastic experience of the *Club Napoli*.

Finally, none of this would have been possible without the strength and the love of the people waiting for me every time I was coming back to Italy.

Thanks to my best friends Giovanni, Monica, Nicola, Filomena, and Butra, the *Spunzoballs* team. I am the luckiest person in the world to have you as friends. No matter the distance, you will always be my *happy island*.

And the greatest thanks goes to my family. A mio padre Nicola, a mia madre Nicla e alle mie sorelle, Cristina e Maria Anna. Ai miei nipoti, Alessandro e Leonardo, e a Francesco. Il vostro amore e il vostro supporto mi hanno aiutato a superare anche i momenti più difficili. I vostri sorrisi all'areoporto, ogni volta che tornavo, sono stati la cosa più bella di questi ultimi cinque anni.

Giuseppe
Göteborg, August 2018

Acronyms

BLA	Best Linear Approximation
EFA	Expanded Fractional Approach
EM	Expectation-Maximization
ESA	Exhaustive Search Approach
FA	Fractional Approach
FIM	Fisher's Information Matrix
i.i.d.	independent and identically distributed
LTI	Linear Time Invariant
MC	Monte Carlo
MCEM	Monte Carlo Expectation Maximisation
MCML	Monte Carlo Maximum Likelihood
MH	Metropolis Hastings
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimator/Estimate
PDF	Probability Density Function
PEM	Prediciton Error Method
SMC	Sequential Monte Carlo
SS	State Space
WH	Wiener-Hammerstein

Contents

II Maximum Likelihood estimate for stochastic nonlinear models

III Initialization algorithms for Wiener-Hammerstein model identification

Part I

Background

Chapter 1

Introduction

Observations, measurements, and experiments are at the basis of the scientific method formally introduced by Galileo Galilei in 1638, with his publication *Discorsi e Dimostrazioni Matematiche Intorno a Due Nuove Scienze* (Discourses and Mathematical Demonstrations Relating to Two New Sciences) [1]. In his work, Galileo argued the importance of using observations to formulate hypotheses and, subsequently, testing them via experiments. The subject of *system identification* strictly follows the same basic scientific methodology: it attempts to formulate hypotheses, the *models*, based on observed data from a phenomenon, the *system*.

Finding accurate and reliable models of complex systems is one of the main challenges of modern science and engineering, since they are needed, for example, to predict and control the outcome of events and phenomena of interest. These models are often expressed in a mathematical form. The simulation analysis carried out using mathematical models may avoid the construction of expensive experiments or prototypes. Moreover, mathematical models are at the basis of most modern control techniques.

A *system* is a real-world's object producing signals depending on the internal interaction of variables of different kinds [2]. The produced signals are called *outputs*. In many cases, an observer can affect the behaviour of the system by applying external signals, which are called *inputs*. Other external signals that cannot be manipulated by the observer are called *disturbances*. They are mainly of two kinds: disturbances that can be directly measured and disturbances that can only be indirectly observed through the system outputs. In this context, the system identification task can be described as the following: given observed signals of a system, find a mathematical model that explains the observations as accurately as possible.

A *mathematical model* is an abstract representation of the system's behaviour. It mathematically relates the inputs, the outputs, and the disturbances. Most of the physical phenomena and engineering applications

around us are of dynamical nature: the output of the system at a certain time does not only depend on the inputs and disturbances at the same time, but also on their history. Hence, a dynamical system can be modelled by a set of differential or difference equations. This set of equations, defining the *model structure*, is then indexed by a parameter θ. A common subset of models is the set of linear time-invariant models. These models assume linear relationships between the input, the disturbances, and the outputs. Linear models are widely used since the identification and the control theory for this class of models is extensively developed [2]. However, when the actual behaviour of the system cannot be captured by linear models, nonlinear models have to be selected. Once a model structure has been selected, the system identification task consists of finding a value for the parameter θ that allows the model to explain the observations of the system. This is the scope of the *parameter estimation method*. Since, in many cases, the observations are affected by disturbance signals that cannot be measured directly, the estimation method has to take into account the concept of *uncertainty*: the model should describe the behaviour of the system as it was *not* affected by disturbances. In order to achieve this result, a characterization of the uncertainty is required.

A common approach to the uncertainty characterization is the *stochastic approach*: the uncertainty is assumed to be of random nature and it is characterized by probability distributions. Hence, with this approach, the estimation method is an application of classical statistical inference techniques. These techniques deal with the problem of extracting information from observations affected by disturbances, making them unreliable. Depending on the assumptions on the nature of θ, the stochastic approach can be further divided into two main frameworks: the *frequentist* framework [3] and the *Bayesian* framework. The first one assumes a deterministic nature, i.e. no probability is attached to the unknown parameter. The inference method, in this case, analyses what would happen if several experiments were to be repeated. In this way, a point estimate of the parameter with some sort of accuracy measure, e.g. confidence regions, is provided. The second framework, instead, assumes a random nature of the parameter itself: some *prior* information, expressed in terms of probability distribution, is attached to the parameter and the inference method *transforms* the prior information to *posterior* information by making use of the observations and the *Bayes' theorem*. Hence, in this case, the outcome of the estimation method is the posterior probability distribution of the parameter. In this thesis, the main focus will be on estimation methods belonging to the frequentist framework.

In the area of statistical inference, the most commonly used estimation method is the Maximum Likelihood (ML) method [2], [4], [5], [6]. As

mentioned before, given the presence of the disturbances, the observations are described as realization of stochastic variables. This description can be mathematically expressed with the joint Probability Density Function (PDF) of the observations. The *likelihood function* is then defined as this joint PDF evaluated at the available observations and seen as a function of the parameter θ. The ML method provides a point estimate for the parameter by maximizing the likelihood function over θ.

The ML method is commonly used because of its desirable statistical properties. In general, the search for a *good* model should be driven by criteria based on its *usefulness* rather than its similarity to the real/physical system. Some aspects of the system can be compared with its mathematical description, but an exact connection between them can never be establish. Nevertheless, statistical properties of estimation methods can be discussed by assuming that a *true system*, defined in terms of a mathematical description, exists. Hence, it is assumed that a *true parameter* θ_0 exists and the observations are a realization of the model output with $\theta = \theta_0$. This is never true in practice, but the fiction of a true system is very useful for the theoretical analysis of the estimation methods and the assessment of the quality of the model. In this way, it is also possible to classify systems by using the mathematical models classification: with *nonlinear system*, for example, it is intended that the fictitious true system is a nonlinear model. This is also why the terms *model* and *system* are often interchangeable.

At least two basic properties need to be considered when assessing the quality of an estimator: consistency and efficiency. Consistency means that the estimates of θ get closer to the true parameter as the observations length increases. Since they are affected by disturbances, the estimates are of random nature. Hence, the convergence to the true parameter is considered in the probabilistic sense, see [4]. Efficiency is a property that relates the errors of consistent estimates. A consistent estimator is asymptotically efficient if the normalized error between the estimates results into a covariance matrix no larger than the covariance matrix of any other consistent estimator.

It is proven that, under weak assumptions, the Maximum Likelihood Estimator (MLE) is consistent and asymptotically efficient. Thanks to its statistical properties, the MLE is often preferred as parameter estimation method. However, it is not always possible to derive the likelihood function in analytic form. This is, for example, the case of systems containing nonlinear relationships between disturbances and outputs. Often, these systems are called *stochastic nonlinear system.* Hence, in these cases, some approximate solutions to the problem need to be adopted. Furthermore, the Maximum Likelihood estimation problem is often a non-convex optimization problem. Thus, numerical optimization algorithms based on local explorations are

used. These methods, however, only guarantee the convergence to a local maximizer of the non-convex cost function.

The central object of this thesis is to address these two main issues: finding the MLE when the likelihood function is not available in analytic form and providing initialization algorithms for MLE in order to increase the chances of finding a global maximizer. These issues are addressed, in particular, for a specific class of nonlinear systems: the Wiener-Hammerstein system. In the following sections, a detailed description of the research problems and of the contribution of the thesis is presented.

1.1 Research Motivation

In this thesis, we are concerned with the problem of deriving a Maximum Likelihood estimate for general nonlinear models. When a disturbance or a unobserved (or *latent*) process is affecting the output of the model through a nonlinear transformation, it is not always possible to derive an analytic expression of the likelihood function. With *latent* process, we mean a set of signals within our system that cannot be directly measured. This issue is further explained by discussing the following two cases. In the first one, we discuss the case when the disturbances affects the outputs of the system only in an additive form. In these cases, the likelihood can be derived in closed-form. The second case, instead, deals with the *stochastic* nonlinear model. A disturbance enters the system through a nonlinear transformation. Deriving the likelihood function in this case is very challenging.

1. **Additive noise on the outputs**. The output of the nonlinear model is only affected by additive disturbances,

$$y_t = f(u_t; \theta) + e_t \quad t = 1, \dots, N \tag{1.1}$$

where $f(\cdot, \theta) : \mathbb{R} \to \mathbb{R}$ is a nonlinear static function parametrized in θ, u_t and y_t are, respectively, scalar inputs and outputs, and e_t represents the stochastic disturbance, modelled as i.i.d. random variables for each t, with a known PDF, $e_t \sim p_e(e_t)$. This is the typical case of measurement noise: the true system's output is usually observed by sensors that introduce additive measurement noise. By defining the vector of the observations $\mathbf{y} = [y_1, \dots, y_N]$, it is possible to construct the joint PDF of the model outputs. Given the independence of the disturbance e_t for each t, the output y_t is also independent over t, hence the joint PDF can be written as

$$p(\mathbf{y}; \theta) = \prod_{t=1}^{N} p(y_t; \theta) = \prod_{t=1}^{N} p(e_t; \theta) = \prod_{t=1}^{N} p(y_t - f(u_t; \theta)). \tag{1.2}$$

The ML estimate of θ can be therefore defined as

$$\hat{\theta}_{ML} := \arg\max_{\theta} \prod_{t=1}^{N} p(y_t - f(u_t; \theta)), \tag{1.3}$$

and computed by solving an optimization problem. The complexity of the PDF $p(e_t)$ and of the nonlinear function $f(\cdot, \theta)$ will determine the complexity of the optimization problem, and whether numerical techniques will be required. For example, if the disturbances are normally distributed with zero mean and variance σ_e^2, $e_t \sim \mathcal{N}(0, \sigma_e^2)$, solving the optimization problem (1.3) is equivalent to

$$\min_{\theta} \; -\log p(\mathbf{y}; \theta) = \min_{\theta} \sum_{t=1}^{N} (y_t - f(u_t; \theta))^2, \tag{1.4}$$

which is a nonlinear least-squares problem. If $f(u_t; \theta)$ is defined as the predictor of the model, see [2], then (1.4) is also the Prediction Error Method (PEM) estimator [2], a widely used estimator that inherits the good statistical properties from the ML estimator. In the system identification context, $f(u_t; \theta)$ can be replaced by a regressor function $\psi(t, \theta)$ which also takes into accout past inputs and outputs of the system. In this way, it is possible to represent dynamical models. Depending on the underlying system's behaviour, several choices of the regressor function are possible. For instance, the regressor function representation is suitable for Nonlinear AutoRegressive with eXogenous input (NARX) models and Nonlinear AutoRegressive Moving Average with eXogenous input (NARMAX) models [2], nonlinear state-space models in predictor form, block-oriented models [7], Volterra series models [8]. The particular choice of model representation is usually based on prior knowledge about the system, but also on the final use of the model. This stage of the identification process is called *model structure selection*, and it is a fundamental and difficult step. In this thesis, we do not address the model structure selection problem, but we assume that a parametrized model is available and we are concerned with the parameter estimation task only. In literature, several approaches address the selection of a model structure for nonliner systems, see e.g. [9] and [10].

The general model representation (1.1) only assumes additive measurement noise. Although this situation is common in practice, it does not cover all the possibilities of disturbances affecting the systems in the real world. A more general representation is given by the second case, discussed next.

2. **Latent process**. We assume that a disturbance or a latent process x_t affects the system's output through a non-invertible nonlinear transformation. This is described by the following equation

$$y_t = f(x_t, u_t; \theta) + e_t \quad t = 1, \ldots, N \tag{1.5}$$

where x_t is a random variable distributed according to $p(x_t; \theta)$. The PDF of the latent process can also be function of θ or of a subset of it. In order to find the likelihood function, it is required to marginalize the unknown process out. By defining $\mathbf{x} = [x_1, \ldots, x_N]$, the PDF of $\mathbf{y}$ is

$$p(\mathbf{y}; \theta) = \int p(\mathbf{y}, \mathbf{x}; \theta) \mathrm{d}\mathbf{x}, \tag{1.6}$$

where $p(\mathbf{y}, \mathbf{x}; \theta)$ is the joint PDF of $\mathbf{y}$ and $\mathbf{x}$. Even in case $p(\mathbf{y}, \mathbf{x}; \theta)$ has a known analytical form, the integral (2.5) is multidimensional and, in general, *intractable*, i.e. it has no closed-form solution. Hence, the challenge is to come up with approximate solutions to the ML problem for this case. The main approaches addressing this issue were originally based on Monte Carlo integration, see e.g. [11], [12], [13], [14]. By extending and improving these approaches, some ongoing research from the system identification community can be found in [15], [16], [17], [18], [19], where the ML estimation is based on Sequential Monte Carlo (SMC) methods. The main drawback of approaches based on Monte Carlo methods is that they can be computationally expensive and their convergence can be very slow. In this thesis, we will present the available solutions and analyse their issues. The goal is to improve some computational and convergence aspects.

Assuming for a moment that the likelihood function, or an approximation of it, has been found, a second issue needs to be addressed. In case of nonlinear models the MLE is obtained by solving a nonlinear optimization problem, which may be intractable too (not solvable in closed-form). Hence, this requires the deployment of numerical optimization techniques. Most of these techniques proceed through local explorations of the cost function, starting from an initial guess for θ. A widely used technique for numerical optimization is the Newton's method. By using information from the gradient and the Hessian of the cost function to optimize, the method is proved to converge to a stationary point of the gradient. Hence, depending on the starting point (initial guess), the method may end its search in the global or one of the local optimizers of the non-convex cost function. Of course, the latter case is not desirable. Thus, the choice of a *good* initial guess for the search algorithm is of crucial importance.

In case of likelihood intractability, the Newton's algorithm cannot be deployed on the analytic likelihood, but approximate solutions have to be derived. For this, many approaches available in the literature suggest the deployment of numerical integration techniques to estimate the likelihood or its gradient. In this case, the key issue to address is to come up with integration and optimization routines that do not compromise the convergence to a local optimizer of the true likelihood. Only when this convergence is ensured, the problem of finding a good initial guess can be addressed.

In this thesis, the two issues regarding the derivation of approximate solution for intractable likelihood function and the search for a good initial guess are addressed. In particular, the first issue is addressed in general terms. A general stochastic nonlinear model is considered and a Newton-based method for solving the approximate ML problem is developed. The method is then tested on a particular model structure: the Wiener-Hammerstein (WH) model with process noise. This is a block-oriented model structure consisting of the interconnection of two LTI blocks with a static nonlinearity in the middle,

$$\begin{aligned} x_t &= G_W(q,\theta)u_t + w_t, \quad w_t \sim p_w(w_t), \\ z_t &= f(x_t,\theta), \\ y_t &= G_H(q,\theta)z_t + e_t, \quad e_t \sim p_e(e_t), \end{aligned} \tag{1.7}$$

where $G_W(q,\theta)$, $G_H(q,\theta)$ are the two LTI systems, represented as discrete-time transfer functions (q denotes the time-shift operator), parametrized in θ, and $f(\cdot,\theta)$ is a static nonlinearity. Due to the presence of the disturbance w_t (process noise), the intermediate signal x_t can be seen as a latent process. Since x_t is then filtered through the nonlinear transformation $f(\cdot,\theta)$, the likelihood $p(\mathbf{y};\theta)$ cannot be computed in closed-form. Hence, the WH model with process noise is a special instance of the stochastic nonlinear models class, and it will be referred to as *stochastic WH model*.

The second issue is harder to consider in general terms. Thus, initialization algorithms for the search of the global optimizer is developed for WH models only. For this kind of block-oriented model structure, in fact, it is shown that linear approximations can be effectively used to initialize the nonlinear ML optimization problem. Both the cases of stochastic WH model and WH model (no process noise) are considered.

In the next section, the outline of the thesis is presented and the contributions are detailed.

1.2 Outline of the Thesis and Contributions

Most of the results presented in this thesis are based upon well-established concepts in system identification and statistical estimation theory. A brief overview of these basic concepts is therefore given in Chapter 2. In particular, the Maximum Likelihood method is presented and explained in details. Then, we discuss the standard solutions for finding the MLE when the likelihood function is available or can be computed in closed-form. This special class of models is often referred to as *tractable* models. Although, in this case, the likelihood is available in closed-form, the optimization problem may still require approximate solutions. Standard numerical optimization methods are used in this case, such as the *gradient-based* algorithms and the *Expectation-Maximization* algorithm.

In the next two parts, the contributions of this thesis are presented. The first part (Chapters 3-4) addresses the case of likelihood intractability. This issue is usually addressed by computing Monte Carlo estimates of the intractable quantity. Maximum Likelihood methods based on Monte Carlo estimates, however, may show high complexity or stability problems. Hence, in Chapter 3, we analyse the main issues of methods based on Monte Carlo estimates and we propose, in Chapter 4, some modifications addressing the main computational and stability issues. The derived solutions reduce the overall complexity of the estimation methods, by implementing an efficient use of the Monte Carlo samples. The results presented in this part are extensions of

G. Giordano, S. Gros, J. Sjöberg, "A Newton-based method for Maximum Likelihood estimation from incomplete data", to be submitted to *Automatica*, 2018.

G. Giordano, J. Sjöberg, "Maximum Likelihood identification of Wiener-Hammerstein system with process noise", *18th IFAC Symposium on System Identification*, Stockholm, Sweden, July 2018.

The second part (Chapters 5-6) addresses the initial guess problem for WH models. Initialization algorithms based on linear approximations are derived. In the first chapter, the concept of Best Linear Approximation (BLA) of nonlinear system is introduced. For a WH model not affected by process noise, it is proved that the BLA provides a consistent estimate of the linear parts of the WH model. In this thesis, the consistency of the BLA is extended to the case of stochastic WH models, where both

measurement and process noise are present. The BLA is then used to initialize the two LTI systems of the WH model structure. Thereafter, a ML estimation problem can be formulated to estimate the nonlinearity too. The initialization algorithms are combined with the methods for ML estimation presented in the first part. Finally, the special case of absence of process noise is also addressed. For this, we prove that standard approaches for MLE initialization suffer of ill-conditioning problems. Thus, a modification of the standard approach is proposed, which alleviates the conditioning issues and improves the algorithmic performance of the initialization method. The results presented in this part are based upon

G. Giordano, S. Gros, J. Sjöberg, "An improved method for Wiener-Hammerstein system identification based on the Fractional Approach", in *Automatica*, Vol 94, pp. 349-360, 2018.

G. Giordano, J. Sjöberg, "Consistency aspects of Wiener-Hammerstein model identifcation in presence of process noise", *IEEE 55th Conference on Decision and Control (CDC)*, Las Vegas, USA, December 2016.

G. Giordano, J. Sjöberg, "A time-domain Fractional Approach for Wiener-Hammerstein systems identification", *17th IFAC Symposium on System Identification*, Beijing, China, October 2015.

Other related publications by the Author not included in this thesis:

G. Giordano, V. Klass, M. Behm, G. Lindbergh, J. Sjöberg, "Model-based Lithium-Ion Battery Resistance Estimation from Electric Vehicle Operating Data", in *IEEE Transactions on Vehicular Technology*, Volume: 67, Issue: 5, May 2018, Pages 3720 - 3728.

G. Giordano, J. Sjöberg, "Black- and white-box approaches for cascaded tanks benchmark system identification", in *Mechanical Systems and Signal Processing*, Volume 108, August 2018, Pages 387-397.

Chapter 2

Background

In this chapter, a background on basic estimation theory and system identification concepts is introduced. The focus is on the Maximum Likelihood method and its statistical properties. The problem of finding the MLE for stochastic nonlinear models is then introduced and discussed. Finally, standard methods for the case of tractable models are presented.

2.1 Signals and models

The objective of system identification is to model dynamical systems given a set of observed data

$$Z^N = \{u_t, y_t\}_{t=1}^N = \{\mathbf{u}, \mathbf{y}\} \tag{2.1}$$

where $\mathbf{u} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^N$ are, respectively, input and output signals of the system. A basic assumption in system identification is that the behaviour of the system is well approximated by some parametrized mathematical model. The choice of the parametrization depends on the application. Experience and prior information are important factors in selecting a proper model parametrization. Of course, the mathematical model cannot provide an exact description of the system, but modelling errors are always present. Furthermore, other disturbances may affect the system's output. Hence, it is important to extend the model definition by using a stochastic framework. In this way, it is possible to model the *uncertainty* present in the system and use statistical inference techniques for parameter estimation. In the following, we introduce the main assumptions and definitions on signals and models, required by the stochastic framework.

2.1.1 Basic assumptions and definitions

The output signal $\mathbf{y}$ is therefore modelled as a set of random variables distributed according to a joint probability density function. In this thesis, we assume that the input signal is known exactly and the joint PDF of the output is parametrized by a finite-dimensional real vector $\theta \in \Theta \subset \mathbb{R}^{N_\theta}$,

$$\mathbf{y} \sim p(\mathbf{y}; \theta). \tag{2.2}$$

Hence, the PDF describes the model behaviour by *wrapping* a deterministic part of the model within a stochastic *envelope*. Based on the nature of the mathematical relationships contained in the model, different model definitions are possible. The mathematical models considered in this thesis are part of the class of stochastic parametric nonlinear dynamical model. Using the formulations introduced by [20], we have the following definition and assumption.

Definition 2.1.1. *(Stochastic parametric nonlinear dynamical models) The models are defined by the following discrete-time output relations*

$$y_t = f(\{u\}_{k=1}^{t-1}, \{v\}_{k=1}^{t}; \theta), \quad t = 1, 2, \ldots, N, \tag{2.3}$$

where f is a nonlinear function, θ is the parameter to be estimated, and $\{v\}_{k=1}^{t}$ is a sequence of latent random variables.

The dependence of the output on past values of inputs and unknown signals ($\{v\}_{k=1}^{t}$) makes the model dynamic. The signals $\{v\}_{k=1}^{t}$ summarize all the stochastic contributions to the model output. For stochastic nonlinear dynamical systems, the main sources of disturbances are the latent process $\mathbf{x} = \{x_t \in \mathbb{R}^{d_x}\}_{t=1}^{N}$ and the measurement noise $\mathbf{e} = \{e_t\}_{t=1}^{N}$. Hence, in this thesis, we will make the assumption that

$$v_t = [x_t, e_t]. \tag{2.4}$$

Since x_t is a latent process, the PDF of the outputs has to be calculated by marginalization, i.e. solving the multi-dimensional integral

$$p(\mathbf{y}; \theta) = \int_{\mathbb{R}^{d_x N}} p(\mathbf{y}, \mathbf{x}; \theta) \mathrm{d}\mathbf{x} \tag{2.5}$$

where $p(\mathbf{y}, \mathbf{x}; \theta)$ is the joint PDF of outputs and latent process.

Finally, as mentioned in the introduction, it is useful to describe the real system in terms of mathematical relations, in order to compare and analyze the quality of the estimators. Hence, we assume that a *true system* exists, which we define in the following.

Assumption 2.1.1. *(True system and true parameter) The observed data are generated by the following mathematical relation*

$$y_t = f(\{u\}_{k=1}^{t-1}, \{v\}_{k=1}^{t}; \theta_0), \quad t = 1, 2, \ldots, N, \tag{2.6}$$

where $\theta_0 \in \Theta \subset \mathbb{R}^{N_\theta}$ is defined as the true parameter.

2.1.2 State-space models

A special class of nonlinear dynamical models is the class of discrete-time nonlinear state-space models. These models are composed by a set of first order difference equations. Thanks to its flexibility and generality, a state-space model can describe the behaviour of a wide range of nonlinear systems. The stochastic state-space model can be defined by the following state and output equations

$$\begin{aligned} x_{t+1} &= h(x_t, u_t, w_t; \theta), \\ y_t &= g(x_t, e_t; \theta), \quad t = 1, 2, \ldots, N, \end{aligned} \tag{2.7}$$

where w_t and e_t are, respectively, process and measurement noise. A particular instance of this model class considers only additive disturbances on the state and output equations

$$\begin{aligned} x_{t+1} &= h(x_t, u_t; \theta) + w_t, \\ y_t &= g(x_t; \theta) + e_t, \quad t = 1, 2, \ldots, N. \end{aligned} \tag{2.8}$$

Given the presence of the process noise w_t, the state x_t can be viewed as a latent/hidden process, in accordance with Definition 2.1.1.

2.1.3 Wiener-Hammerstein models

In this thesis, ML estimates and initialization algorithms are developed for another important class of stochastic nonlinear dynamical models: the Wiener-Hamemrstein (WH) model. The WH model is a single-input/single output model and it is part of the block-oriented structures family. Block-oriented models represent a more structured approach to nonlinear modelling, see [7]. In particular, nonlinear systems whose behaviour can be easily decomposed in linear and static nonlinear contributions are well-described by block-oriented model structure. The linear contributions are modelled by Linear Time-Invariant (LTI) dynamical blocks. They can have several mathematical representations, such as rational transfer functions, linear state-space models, basis function expansions, or nonparametric descriptions, e.g. nonparametric frequency responses. The static nonlinearity can be

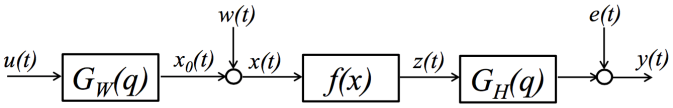


Figure 2.1: The Wiener-Hammerstein system

expressed by basis function models, polynomial expressions, nonparametric kernel models, etc.

The LTI block and the static nonlinearities can be combined in many ways. The simplest combinations are the Wiener model and the Hammerstein model. In the first case, the input of the system is filter by an LTI block, whose output goes through a static nonlinearity. The Hammerstein model is the reverse: the nonlinearity is present at the input, followed by an LTI model. By combining these two models, we obtain the Wiener-Hammerstein model. It is composed by two LTI blocks with a static non-linearity in the middle, see also Figure 2.1,

$$\begin{aligned} x_t &= G_W(q, \theta_W)u_t + w_t, \\ z_t &= f(x_t, \theta_{NL}), \\ y_t &= G_H(q, \theta_H)z_t + e_t, \end{aligned} \tag{2.9}$$

with $\theta = [\theta_W, \theta_{NL}, \theta_H]$. Similarly to the state-space model, also in this case w_t and e_t are, respectively, process and measurement noise. The function f is a static nonlinear function, while G_W and G_H are two LTI systems expressed in terms of discrete-time transfer functions, parametrized in θ. The operator q defines the forward time shift operation, i.e.

$$qu_t = u_{t+1}.$$

A generic discrete-time transfer function $G(q;\theta)$ is defined in terms of rational functions

$$G(q, \theta) = \frac{B(q, \theta)}{F(q, \theta)}, \tag{2.10}$$

where $B(q, \theta)$ and $F(q, \theta)$ are polynomials in the shift operator whose coefficients are functions of θ.

In this thesis, we will consider the following parametrization for the blocks of the WH model. The two linear parts are parametrized with two stable linear time-invariant transfer functions. In the discrete-time domain,

they can be expressed as

$$G_W(q,\theta_W) = \frac{\sum_{k=0}^{n_B^W} b_k^W q^{-k}}{1+\sum_{k=1}^{n_A^W} a_k^W q^{-k}}, \tag{2.11}$$

$$G_H(q,\theta_H) = \frac{\sum_{k=0}^{n_B^H} b_k^H q^{-k}}{1+\sum_{k=1}^{n_A^H} a_k^H q^{-k}}, \tag{2.12}$$

where $\theta_W = [b_0^W, ..., b_{n_B^W}^W, a_1^W, ..., a_{n_A^W}^W]$ and $\theta_H = [b_0^H, ..., b_{n_B^H}^H, a_1^H, ..., a_{n_A^H}^H]$ are the parameter vectors. The static non-linearity is expressed as a basis functions expansion,

$$f(x_t,\theta_{NL}) = \sum_{k=1}^{d} \theta_{NL}^k f_k(x_t), \tag{2.13}$$

where f_k are the basis functions, θ_{NL}^k are the parameters entering linearly in f, and d is the number of basis functions.

An advantage of the block-oriented models is that, under some assumptions, the best linear approximation (BLA) [21], [22], [23] of the nonlinear system is strictly related to the LTI blocks, see [10], [24]. This result is used to separate the estimation of the linear and nonlinear parts of the model. In this thesis, we will use and extend this result, in order to derive initialization algorithm for the stochastic Wiener-Hammerstein model identification. In some cases, we will test the derived algorithms on experimental data provided by a real system that can be modelled as a WH. This is presented next.

2.1.4 The Wiener-Hammerstein benchmark system

In this thesis, real experimental data from a Wiener-Hammerstein benchmark example are used to test some of the derived algorithms and methods. The benchmark was originally proposed by [25]. The real system is an electronic WH system built by sandwiching a resistor-diode network in between two third-order Cheyshev filters. A picture of the electronic system is reported in Figure 2.2. The benchmark data presented in [25] were collected from the real WH system when only the measurement noise e_t was affecting the system's output.

More recently, a modification of the benchmark example has been presented, see [26]. In this case, an additional disturbance has been introduced at the input of the resistor-diode network, to simulate the effect of the process noise. From the information available in [26], we know that the noises sources e_t and w_t can be considered to be white and Gaussian. The dominant noise source is w_t, the measurement noise e_t is very small. We

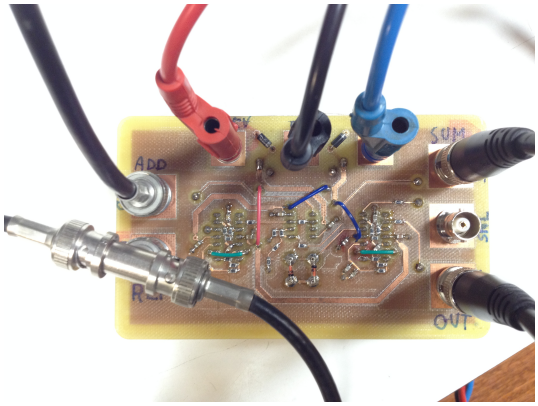


Figure 2.2: The Wiener-Hammerstein benchmark system

will use this second benchmark set of data to test the algorithms derived for the stochastic WH model.

2.2 Parameter estimation methods

Once a mathematical model and a stochastic framework have been determined, the next step of the system identification procedure is to choose a parameter estimation method. In the frequentist framework, two methods are mainly used for parameter estimation. They are the Maximum Likelihood Method and the Prediction Error Method (PEM). In this thesis, the focus will be on the first one, introduced in the following. Nevertheless, we also briefly present the PEM and its connection with the ML method.

2.2.1 The Maximum Likelihood method

The Maximum Likelihood method is a statistical inference method introduced by Fisher [27]. It is based on the likelihood function, which is defined in the following.

Definition 2.2.1. *(Likelihood function) Given the joint PDF of the outputs of the system*

$$p(\boldsymbol{y};\theta) = p(y_1, y_2, \ldots, y_N;\theta), \tag{2.14}$$

the likelihood function is defined as the joint PDF $p(\boldsymbol{y};\theta)$ evaluated at a particular realization (observation) $\boldsymbol{y}^*$ of the random vector $\boldsymbol{y}$,

$$p(\boldsymbol{y}^*;\theta). \tag{2.15}$$

This is a deterministic function of θ, once the numerical value $\boldsymbol{y}^*$ is inserted.

Hence, a reasonable estimator of θ is the one following the likelihood principle: pick the value of θ so that the observed event becomes *as likely as possible*. In mathematical terms, this can be defined as in the following.

Definition 2.2.2. *(Maximum Likelihood Estimator) The random variable*

$$\hat{\theta}_{ML} = \arg\max_{\theta \in \Theta} \; p(\boldsymbol{y}^*; \theta). \tag{2.16}$$

is the Maximum Likelihood Estimator (MLE) of θ.

With these definitions, some statistical properties of the MLE can be discussed. Since the estimator is a random variable, its quality can be assessed by its mean-square error matrix

$$P = \mathbb{E}[\hat{\theta}_{ML} - \theta_0][\hat{\theta}_{ML} - \theta_0]^T. \tag{2.17}$$

where θ_0 is the true parameter. It is desirable to come up with estimators with small P. For unbiased estimators, see [28], [29], it exists a lower limit to the value of P. This limit is defined by the *Cramér-Rao inequality*.

Theorem 2.2.1. *(Cramér-Rao inequality) Let $\hat{\theta}$ be an unbiased estimator of θ, and assume that the PDF of the observations $\boldsymbol{y}^*$ is $p(\boldsymbol{y}^*; \theta_0)$. Then*

$$\mathbb{E}[\hat{\theta} - \theta_0][\hat{\theta} - \theta_0]^T \geq M^{-1} \tag{2.18}$$

where M is the Fisher Information Matrix (FIM), defined by

$$M = -\mathbb{E}[\nabla_\theta^2 \log p(\boldsymbol{y}^*; \theta)]\big|_{\theta=\theta_0}, \tag{2.19}$$

in which $\nabla_\theta^2 \log p(\boldsymbol{y}^*; \theta)$ denotes the Hessian of the log-likelihood function of θ.

A proof of the Cramér-Rao inequality is given in [2]. In case of independent observations, the ML estimator benefits from the following asymptotic properties [30], [31].

Theorem 2.2.2. *Suppose that the random variables $y_1, y_2, \ldots, y_N$ are independent and identically distributed, so that*

$$p(\boldsymbol{y}; \theta) = \prod_{t=1}^{N} p(y_t; \theta). \tag{2.20}$$

Suppose also that the distribution of the observation $\boldsymbol{y}^*$ is given by $p(\boldsymbol{y}^*; \theta_0)$. Then

$$\lim_{N \to \infty} \hat{\theta}_{ML} = \theta_0 \quad w.p.\ 1, \tag{2.21}$$

and

$$\sqrt{N}[\hat{\theta}_{ML} - \theta_0] \sim \mathcal{N}(0, M^{-1}). \tag{2.22}$$

The two results of Theorem 2.2.2 are also referred to as, respectively, *consistency* and *asymptotic efficiency*. Thanks to these properties, the MLE is thus the best possible estimator and it is widely used in many scientific fields, including system identification. It is however important to underline that those properties hold asymptotically, as N approaches infinity. In the finite case, there are no general guarantees. Nevertheless, in many practical cases, the MLE shows its asymptotic properties even for finite but *long enough* data sequence. This is also why the MLE is widely used in practice.

Once the MLE has been defined, the next step is to compute it. For general nonlinear models, the optimization problem (2.16) cannot be solved in closed-form. Hence, numerical optimization techniques are employed. Furthermore, we will discuss the problem of finding the MLE for stochastic nonlinear models, i.e. when the likelihood function has to be calculated as marginalization of the latent process, see Equation (2.5). The methods for the computation of the MLE are presented in Section 2.3.

2.2.2 Connection with the Prediction Error Method

Another widely used family of parameter estimation methods is the family of Prediction Error Methods (PEMs), see [2], [6], [32]. In this case, the main idea is to write the parametrized model in terms of parametrized predictor: known inputs and previous observed outputs are used to predict *future* outputs. Then, the parameter estimation is performed via minimization of some metric ℓ defined on the prediction error $e_t(\theta)$, i.e. the distance between the observed and predicted output.

The one-step-ahead predictor is defined by the function

$$\hat{y}_{t|t-1}(\theta) = \phi(Z^{t-1}, t, \theta), t = 1, \dots, N \tag{2.23}$$

where $Z^{t-1} = \{u_i, y_i\}_{i=1}^{t-1}$ and it is assumed the presence of one-time input delay. Also in this case, the function ϕ can be selected based on prior knowledge of the system and on the stochastic description of it. Once this predictor is defined, the parameter estimation is done by solving a minimization problem.

Definition 2.2.3. *(Prediction Error Method estimator) Given a predictor function ϕ and a nonnegative scalar-valued function ℓ, the random variable*

$$\hat{\theta}_{PEM} := \min_{\theta} \sum_{t=1}^{N} \ell(e_t(\theta), t; \theta), \tag{2.24}$$

where $e_t(\theta) = y_t - \hat{y}_{t|t-1}(\theta),\ \forall t = 1, \dots, N$, is the prediction error method estimator.

The particular choice of ℓ and ϕ defines the particular instance of the estimator within the family of PEMs. When this metric and predictor can be chosen according to the exact probabilistic nature of the data, the PEM coincides with the ML method. This is true for the following additive measurement noise case. Assume that the data are generated according to

$$y_t = \phi(Z^{t-1}, t, \theta) + e_t, \ t = 1, \ldots, N \tag{2.25}$$

where e_t is an independent zero mean process, with PDF $p(e_t; \theta)$. Thus, it is easy to define the likelihood function of the observed output as a reflection of the PDF of e_t,

$$p(\mathbf{y}; \theta) = \prod_{t=1}^{N} p(e_t; \theta) = \prod_{t=1}^{N} p(y_t - \phi(Z^{t-1}, t, \theta); \theta) \tag{2.26}$$

The MLE can thus be found by

$$\min_{\theta} \ -\log(p(\mathbf{y}; \theta)) = \min_{\theta} -\sum_{t=1}^{N} \log(p(y_t - \phi(Z^{t-1}, t, \theta); \theta)). \tag{2.27}$$

If we choose the predictor for the PEM as

$$\hat{y}_{t|t-1}(\theta) = \phi(Z^{t-1}, t, \theta) \tag{2.28}$$

and the metric ℓ as

$$\ell(e_t, t, \theta) := -\log(p(e_t; \theta)), \tag{2.29}$$

where $e_t(\theta) = y_t - \hat{y}_{t|t-1}(\theta)$, the PEM estimator would be provided by

$$\min_{\theta} -\sum_{t=1}^{N} \log(p(y_t - \phi(Z^{t-1}, t, \theta); \theta)), \tag{2.30}$$

which coincides with the ML estimator, defined in (2.27). This entails that the PEM estimator inherits the statistical properties of the ML estimator, i.e. consistency and asymptotic efficiency. The equivalence to the MLE is due to the fact that, in this case, it is possible to choose the predictor for the PEM estimator according to the actual stochastic description of the data, defined in (2.25). As already discussed in the introduction, this is not the general case for the stochastic nonlinear models, for which the presence of the latent process makes the equivalence invalid. Hence, the focus of this thesis will be on the derivation of ML estimates and, in case of stochastic WH model, we will show that a standard PEM approach, as defined in (2.30), leads to inconsistent estimates.

Nevertheless, it is important to stress that when PEM is not equivalent to the MLE, this does not directly imply its inconsistency. In fact, there are other ways to define the predictor and the metric in order to obtain consistent PEM estimates, even when it does not coincide with the MLE. Recent studies in this direction can be found in [20], [33]. A consistent PEM estimator might be harder to define, but the related optimization problem would be easier to solve. On the other hand, a ML estimator is always possible to define, but finding the MLE solution might be difficult. From this point of view, this thesis is an attempt to simplify the search for the MLE.

2.3 Finding the MLE for stochastic models

In this section, an overview of the main numerical techniques for finding the MLE, when the optimization problem (2.16) cannot be solved in closed-form, is presented. In particular, we consider the case of stochastic nonlinear models. The key quantity of the ML problem is the likelihood function. For stochastic model, this is calculated via marginalization of the latent process

$$p(\mathbf{y};\theta) = \int_{\mathbb{R}^{d_x N}} p(\mathbf{y},\mathbf{x};\theta)\mathrm{d}\mathbf{x}, \tag{2.31}$$

where $\mathbf{y} \in \mathbb{R}^N$ and $\mathbf{x} \in \mathbb{R}^{d_x N}$. The methods presented in this chapter and in the rest of the thesis require a basic assumption regarding the joint PDF $p(\mathbf{y},\mathbf{x};\theta)$.

Assumption 2.3.1. *The joint PDF $p(\boldsymbol{y},\boldsymbol{x};\theta)$ has a known analytical form, parametrized in θ.*

This assumption holds for many nonlinear dynamical models, e.g. state-space models with known process and measurement noise distributions, or block-oriented models with similar characteristics. The joint PDF can be factorized as

$$p(\mathbf{y},\mathbf{x};\theta) = p(\mathbf{y}|\mathbf{x};\theta)p(\mathbf{x};\theta). \tag{2.32}$$

In this case, Assumption 2.3.1 imposes that both $p(\mathbf{y}|\mathbf{x};\theta)$ and $p(\mathbf{x};\theta)$ have a known analytical form. In an analogous way, it is possible to factorize the joint PDF as

$$p(\mathbf{y},\mathbf{x};\theta) = p(\mathbf{x}|\mathbf{y};\theta)p(\mathbf{y};\theta), \tag{2.33}$$

where $p(\mathbf{x}|\mathbf{y};\theta)$ is the *posterior* distribution of $\mathbf{x}$ given $\mathbf{y}$. If this posterior is known, then the likelihood function can be computed as

$$p(\mathbf{y};\theta) = \frac{p(\mathbf{y},\mathbf{x};\theta)}{p(\mathbf{x}|\mathbf{y};\theta)}. \tag{2.34}$$

Hence, the computation of the likelihood function of stochastic models requires either the solution in closed-form of the integral in (2.31) or the availability in analytical form of the posterior $p(\mathbf{x}|\mathbf{y};\theta)$. Thus, we can define the *intractable* stochastic nonlinear models in the following way.

Definition 2.3.1. *(Intractable stochastic nonlinear models) A stochastic nonlinear model, see Definition 2.1.1, is defined intractable if the likelihood function of its outputs, $p(\boldsymbol{y};\theta)$, is not available in analytical form.*

This happens when both the integral (2.31) cannot be solved in closed-form and the posterior $p(\mathbf{x}|\mathbf{y};\theta)$ is not available. While Assumption 2.3.1 holds for many nonlinear systems, the tractability of the integral (2.31) or the availability of the posterior $p(\mathbf{x}|\mathbf{y};\theta)$ are satisfied very rarely in case of nonlinear systems. Approximate solutions for the ML problems, in this case, will be discussed in the next chapter.

Nevertheless, it is still important to present the available solutions to the ML problem when either the likelihood function (and its gradient) or the posterior $p(\mathbf{x}|\mathbf{y};\theta)$ possess a known analytical form. When this happens, the stochastic models are often referred to as *tractable* models. The two main approaches for finding the MLE for tractable models are the gradient-based and the Expectation-Maximization algorithms. The first one is mainly used when the likelihood function and/or its gradient are available, the second one when the same is true for the posterior $p(\mathbf{x}|\mathbf{y};\theta)$.

2.3.1 Gradient-based algorithms

Gradient-based algorithms are widely used in numerical optimization, see [34], [35]. The two main algorithms in this class are the steepest-ascent algorithm and the Newton's method. They are iterative algorithms that, starting from an initial guess of θ, proceed through local explorations of the quantity to optimize. The key quantity used in the iterations is the gradient of the likelihood function

$$\nabla_\theta p(\mathbf{y};\theta) = \frac{\partial}{\partial\theta} p(\mathbf{y};\theta). \tag{2.35}$$

The steepest-ascent algorithm updates the guesses of θ by iterating

$$\theta^{(i+1)} = \theta^{(i)} + \alpha \nabla_\theta p(\mathbf{y};\theta)|_{\theta=\theta^{(i)}} \tag{2.36}$$

where α is the a small non-negative real number and it is used to control the size of the update. In this formulation, only the gradient of the likelihood is evaluated at each iteration. In practice, line-search techniques [34] are also implemented, in order to guarantee that the update in θ provides an

increase of the value of the cost function and that the algorithm is stable. In this case, the evaluation of the likelihood function is also required.

The Newton's method makes use of the gradient and the Hessian of the likelihood function. The iterative update is

$$\theta^{(i+1)} = \theta^{(i)} - [\nabla^2_\theta p(\mathbf{y};\theta)|_{\theta=\theta^{(i)}}]^{-1} \nabla_\theta p(\mathbf{y};\theta)|_{\theta=\theta^{(i)}}. \tag{2.37}$$

Also in this case, a small step size α can be introduced

$$\theta^{(i+1)} = \theta^{(i)} - \alpha[\nabla^2_\theta p(\mathbf{y};\theta)|_{\theta=\theta^{(i)}}]^{-1} \nabla_\theta p(\mathbf{y};\theta)|_{\theta=\theta^{(i)}}. \tag{2.38}$$

The Newton's method has quadratic convergence, see [34]. Therefore, its use is desirable. However, its application hinges on the possibility of efficient computation and use of the Hessian matrices. Fortunately, recently developed algorithmic tools allow for fast computation of sensitivity information, see [36]. Moreover, when non-invertible or ill-conditioned Hessian matrices render the Newton's scheme difficult to implement, other methods can be used to alleviate such difficulties, i.e. the quasi-Newton methods [34].

When using gradient-based algorithms for solving ML problems, it is also quite common to work with the log of the likelihood function instead of the likelihood itself. This may provide some numerical benefits. In fact, the likelihood of dynamical system's outputs is usually expressed in terms of product of single realization distributions, $p(\mathbf{y};\theta) = \prod_{t=1}^{N} p(y_t;\theta)$. For big N, this value may become very small. The use of logarithm turns the product into a sum and the produced values may be more tractable. Furthermore, the gradient of log-likelihood is usually well-scaled compared to the gradient of the likelihood, especially for the exponential family distributions.

Finally, both gradient-based and Newton's methods are proved to converge to a stationary point of the gradient. Hence, in case of non-convex likelihood function, only local convergence is guaranteed. This is why the choice of the initial guess θ_0 is of crucial importance, and it is one of the issues addressed in this thesis. In fact, in case of Wiener-Hammerstein models, we will discuss a particular initialization strategy based on linear approximations of the nonlinear system.

2.3.2 The Expectation-Maximization algorithm

Similarly to the gradient-based algorithms, the Expectation-Maximization (EM) algorithm is another iterative procedure for solving ML problems. It is mainly employed when a latent process or, more generally, *incomplete data* are present, see [37]. The algorithm requires the knowledge of the posterior $p(\mathbf{x}|\mathbf{y};\theta)$ but it assumes that the likelihood function is not available. In fact, it makes use of intermediate quantities, function of the joint PDF

$p(\mathbf{y}, \mathbf{x}; \theta)$, in order to maximize the likelihood function. This is explained in the following. Consider the likelihood factorization

$$p(\mathbf{x}|\mathbf{y}; \theta) = \frac{p(\mathbf{y}, \mathbf{x}; \theta)}{p(\mathbf{y}; \theta)}. \tag{2.39}$$

By taking the log of these quantities, we obtain

$$\log p(\mathbf{y}; \theta) = \log p(\mathbf{y}, \mathbf{x}; \theta) - \log p(\mathbf{x}|\mathbf{y}; \theta). \tag{2.40}$$

Given a guess value $\theta^{(i)}$ of the parameter, and knowing the posterior density $p(\mathbf{x}|\mathbf{y}; \theta)$, the latent process can be marginalized over this posterior, providing

$$\log p(\mathbf{y}; \theta) = \int \log p(\mathbf{y}, \mathbf{x}; \theta) p(\mathbf{x}|\mathbf{y}; \theta^{(i)}) \mathrm{d}\mathbf{x} - \int \log p(\mathbf{x}|\mathbf{y}; \theta) p(\mathbf{x}|\mathbf{y}; \theta^{(i)}) \mathrm{d}\mathbf{x}. \tag{2.41}$$

By defining

$$\begin{aligned} Q(\theta, \theta^{(i)}) &= \int \log p(\mathbf{y}, \mathbf{x}; \theta) p(\mathbf{x}|\mathbf{y}; \theta^{(i)}) \mathrm{d}\mathbf{x}, \\ -S(\theta, \theta^{(i)}) &= - \int \log p(\mathbf{x}|\mathbf{y}; \theta) p(\mathbf{x}|\mathbf{y}; \theta^{(i)}) \mathrm{d}\mathbf{x}, \end{aligned} \tag{2.42}$$

it is possible to write

$$\log p(\mathbf{y}; \theta) - \log p(\mathbf{y}; \theta^{(i)}) = (Q(\theta, \theta^{(i)}) - Q(\theta^{(i)}, \theta^{(i)})) + (S(\theta^{(i)}, \theta^{(i)}) - S(\theta, \theta^{(i)})).$$

The quantity $(S(\theta^{(i)}, \theta^{(i)}) - S(\theta, \theta^{(i)}))$ is defined as the relative entropy and it is always non-negative, see [38]. Hence, in order to obtain a positive increment of the log-likelihood function, we have to seek for another value $\theta^{(j)}$ of θ such that

$$Q(\theta^{(j)}, \theta^{(i)}) - Q(\theta^{(i)}, \theta^{(i)}) \geq 0. \tag{2.43}$$

That is the basic principle of the EM algorithm, whose steps are formalized in the following.

- E-step: Compute $Q(\theta, \theta^{(i)}) = \mathbb{E}_{p(\mathbf{x}|\mathbf{y}; \theta^{(i)})}[\log p(\mathbf{y}, \mathbf{x}; \theta)]$
- M-step : Compute $\theta^{(i+1)} = \arg\max_{\theta} Q(\theta, \theta^{(i)})$

By iterating these two steps, the algorithm is proved to converge to a local maximum of the likelihood function, see [39]. The rate of convergence of standard EM algorithm is linear [37].

Compared to the gradient-based algorithm, the use of the EM algorithm is highly suggested when the E-step is cheap to compute and the M-step has a simple expression allowing a closed-form solution. When the E-step is

computationally expensive and/or the M-step requires the deployment of numerical optimization techniques, there is no clear advantage of using the EM algorithm instead of a gradient-based algorithm. However, it has been proved that each M-step does not need full convergence to a local maximizer, but any parameter guess $\theta^{(i+1)}$ satisfying the condition $Q(\theta^{(i+1)}, \theta^{(i)}) > Q(\theta^{(i)}, \theta^{(i)})$ is enough. Thus, some approaches propose to use only one iteration of a gradient-based algorithm to solve the M-step, see e.g. [40]. On the other hand, the use of gradient-based algorithm is highly suggested when second-order information are easy to compute, allowing quadratic convergence, which is clearly a benefit compared to the linear convergence of the EM methods.

2.4 Summary

In this chapter, the main theoretical and algorithmic tools for system identification and statistical estimation have been introduced. Given its statistical properties, a widely used parameter estimation method is the Maximum Likelihood estimator. For stochastic nonlinear models, the likelihood function is not available directly but it is the result of a marginalization operation. In case of tractable stochastic models, two main numerical algorithms are adopted to solve the ML problem (2.16). They are the gradient-based and the Expectation-Maximisation algorithms. General guidelines on the use of the these two methods have been presented. The next part of the thesis will deal, instead, with the intractable case.

Part II

Maximum Likelihood estimate for stochastic nonlinear models

Chapter 3

Methods for the intractable case

In this chapter, we introduce the problem of finding approximate solutions to the MLE problem, for the general case of intractable stochastic nonlinear models. The main object required for computation of the MLE is the likelihood function defined in (2.15). For stochastic nonlinear models, the presence of the latent process $\mathbf{x}$ makes the computation of this likelihood problematic. In fact, the likelihood can be either computed by solving the multi-dimensional integral (2.31) or using the factorization (2.34), which requires the knowledge of the posterior $p(\mathbf{x}|\mathbf{y};\theta)$.

With *intractable case*, we mean that both it is not possible to solve (2.31) in closed-form and the posterior $p(\mathbf{x}|\mathbf{y};\theta)$ is not available. Hence, approximate solutions to the MLE need to be used in this case. Most of the approaches available in the literature address the issue by deploying numerical integration techniques. In particular, the gradient-based and the EM algorithms, described in the previous chapter, are modified using a random numerical integration technique, the Monte Carlo method [41]. We give a brief overview of the Monte Carlo method in Appendix 8.1. In the following, instead, we present and analyze the main modifications to the gradient-based and EM algorithms available in the literature, addressing the intractable problem. The goal of this chapter is to understand which are the main difficulties of obtaining the MLE in the intractable case.

3.1 Methods based on Monte Carlo simulations

The main advantage of the Monte Carlo methods, compared to analytic or other numerical integration techniques, is that, asymptotically, they can be considered as exact methods for numerical integration. In fact, the approximation errors tend to zero has the Monte Carlo effort tends to infinity, where, with effort, we mainly mean the number of random points

used to evaluate the integrand function. Hence, the integration errors are only function of the available computational time and budget needed to generate the random points and evaluate the function, see 8.1.

In most cases, when Monte Carlo methods are used for solving the MLE problem, the theoretical convergence to the true parameter can be established under mild conditions. In practice, however, the required effort might be too expensive. The focus of the next sections will be to analyze the problems related to the complexity and computational issues of the available approaches based on Monte Carlo simulations, in order to propose new solutions to them.

We will distinguish between two main categories addressing the ML problem based on Monte Carlo simulations. They are the Monte Carlo approximations of gradient-based methods and the Monte Carlo Expectation-Maximization (MCEM) methods. Although both of them address the ML problem, in literature the first category is also known as the Monte Carlo Maximum Likelihood (MCML) method, since it directly tries to maximize the likelihood function.

3.1.1 The Monte Carlo Maximum Likelihood methods

These methods approximate directly the likelihood function and/or its gradient. Recall that the likelihood function is given by the marginalization integral

$$p(\mathbf{y};\theta) = \int_{\mathbb{R}^{d_x N}} p(\mathbf{y},\mathbf{x};\theta)\mathrm{d}\mathbf{x} = \int_{\mathbb{R}^{d_x N}} p(\mathbf{y}|\mathbf{x};\theta)p(\mathbf{x};\theta)\mathrm{d}\mathbf{x}. \tag{3.1}$$

Hence the integral can be reformulated as expectation of functions over a probability distribution,

$$p(\mathbf{y};\theta) = \mathbb{E}_{p(\mathbf{x};\theta)}[p(\mathbf{y}|\mathbf{x};\theta)]. \tag{3.2}$$

If $\mathbf{X}^M = \{X^{(m)}\}_{m=1}^M$ is a set of M i.i.d. random samples, distributed according $p(\mathbf{x};\theta)$, then an unbiased Monte Carlo estimate of this quantity is

$$p(\mathbf{y};\theta) \approx \hat{p}(\mathbf{y};\theta,\mathbf{X}^M) = \frac{1}{M}\sum_{m=1}^{M} p(\mathbf{y}|X^{(m)};\theta), \quad X^{(m)} \sim p(\mathbf{x};\theta). \tag{3.3}$$

Thus, in principle it is possible to deploy the gradient-based algorithm, described in Section 2.3.1, using this likelihood approximation. The gradient can be computed by analytical derivation of the function (3.3), or by numerical differences. The main difficulty is, however, that the sampling

PDF $p(\mathbf{x};\theta)$ requires the knowledge of the unknown parameter θ. To address this problem, [11], [42] resort the ML problem into a likelihood ratio maximization problem,

$$\hat{\theta}_{ML} := \arg\max_{\theta} \frac{p(\mathbf{y};\theta)}{p(\mathbf{y};\psi)}, \tag{3.4}$$

and the ratio is approximated via Monte Carlo integration

$$\frac{p(\mathbf{y};\theta)}{p(\mathbf{y};\psi)} \approx r(\theta, \mathbf{X}^M) = \frac{1}{M}\sum_{m=1}^{M} \frac{p(\mathbf{y}|X^{(m)};\theta)}{p(\mathbf{y}|X^{(m)};\psi)}, \quad X^{(m)} \sim p(\mathbf{x};\psi). \tag{3.5}$$

In this way, the samples simulation can be performed using any arbitrary value of the parameter space, ψ in this case, and the a priori knowledge of θ is not required. Another solution is presented in [13] where, instead, importance sampling is used to address the sampling issue,

$$\hat{p}(\mathbf{y};\theta,\mathbf{X}^M) = \frac{1}{M}\sum_{m=1}^{M} p(\mathbf{y}|X^{(m)};\theta)\frac{p(X^{(m)};\theta)}{p(X^{(m)};\psi)}, \quad X^{(m)} \sim p(\mathbf{x};\psi). \tag{3.6}$$

Also in this case, the sampling is performed using ψ instead of θ. Importance sampling also helps in reducing the variance of the Monte Carlo estimate, see Appendix 8.1. In both cases, convergence of the ML estimate to a local optimizer of the true likelihood holds when the sample size M goes to infinity, see [42]. If $M < \infty$, convergence may still be achieved but the sample size has to grow exponentially fast with the dimension N of the signals $\mathbf{x}$ and $\mathbf{y}$. The main reason for this is that the samples simulation should provide the whole picture of the likelihood, i.e. how the likelihood changes as a function of the parameter, for all possible values in the parameter space. With only one samples simulation, from an arbitrary parameter value, enormous effort (M) is required in order to achieve this level of approximation.

Based on these results, the authors of [13] and [42] conclude that these MCML methods are effective in practice only when ψ is already in a close, local neighbourhood of the true parameter θ_0. In this case, in fact, it is arguable that the convergence results are actually achieved for a reasonably finite M, which can provide a local but accurate approximation of the likelihood. However, θ_0 is typically unknown, and ψ may be far from it. Hence, they suggest to use many runs of these MCML methods, where the Monte Carlo integration stage and the gradient-based search are nested in a loop, and intermediate solutions $\theta^{(k)}$ are used for the sampling stage, instead of ψ. The local solution of one MCML run can be used to locally approximate the unknown sampling distribution in the next run (similarly to the EM algorithm) and the parameter search can move towards θ_0. In

this way, a smaller sample size M would be enough to build an accurate, local approximation the the likelihood, since the portion of the parameter space to explore is a local, limited neighbourhood of the current parameter estimation.

This idea is implemented in practice in [43], [44], [45], where the authors propose to iteratively alternate the Monte Carlo integration stage with one iteration of a gradient-based method. In particular, Newton's method is used in [43], [45], and steepest-ascent method in [44]. Based on the local sampling from the current guess $\theta^{(k)}$, the methods proceed through local approximations of the likelihood. Furthermore, instead on relying on the numerical differentiation of the Monte Carlo likelihood estimates, for obtaining the gradient, these methods make use of Monte Carlo estimates of the gradient of the log-likelihood directly. This is presented next.

MCML methods based on gradient approximations

Since the main object of gradient-based method is the gradient of the likelihood, Monte Carlo simulations can be used to estimate this quantity directly,

$$\nabla_\theta p(\mathbf{y};\theta) = \frac{\partial}{\partial\theta}\int p(\mathbf{y}|\mathbf{x};\theta)p(\mathbf{x};\theta)\mathrm{d}\mathbf{x}. \tag{3.7}$$

However, given the presence of the intractable integral, it is not always possible to change the order of differentiation and integration, in order to obtain a suitable form for Monte Carlo approximation. Nevertheless, if we consider the gradient of the log-likelihood instead, Fisher's identity can be used to obtain a suitable form. Under some regularity conditions, see [46], the identity states that

$$\nabla_\theta \log p(\mathbf{y};\theta) = \int \nabla_\theta[\log p(\mathbf{x},\mathbf{y};\theta)]p(\mathbf{x}|\mathbf{y};\theta)\mathrm{d}\mathbf{x}. \tag{3.8}$$

The derivation of (3.8) can be found in Appendix 8.2. The joint likelihood can be factorised, leading to

$$\begin{aligned}
\nabla_\theta \log p(\mathbf{y};\theta) &= \int_{\mathcal{X}} \frac{\partial}{\partial\theta}[\log p(\mathbf{y}|\mathbf{x};\theta)p(\mathbf{x};\theta)]p(\mathbf{x}|\mathbf{y};\theta)\mathrm{d}\mathbf{x} \\
&= \int_{\mathcal{X}} \frac{\partial}{\partial\theta}[\log p(\mathbf{y}|\mathbf{x};\theta) + \log p(\mathbf{x};\theta)]p(\mathbf{x}|\mathbf{y};\theta)\mathrm{d}\mathbf{x} \\
&= \int_{\mathcal{X}} \Psi(\mathbf{x};\theta)p(\mathbf{x}|\mathbf{y};\theta)\mathrm{d}\mathbf{x}, \\
\Psi(\mathbf{x};\theta) &= \frac{\partial}{\partial\theta}\log p(\mathbf{y}|\mathbf{x};\theta) + \frac{\partial}{\partial\theta}\log p(\mathbf{x};\theta).
\end{aligned} \tag{3.9}$$

The integral can be reformulated as expectation of functions over a probability distribution,

$$\nabla_\theta \log p(\mathbf{y};\theta) = \mathbb{E}_{p(\mathbf{x}|\mathbf{y};\theta)}[\Psi(\mathbf{x};\theta)]. \tag{3.10}$$

This form is suitable for Monte Carlo integration,

$$\nabla_\theta \log p(\mathbf{y};\theta) \approx \hat{G}(\theta, \mathbf{X}^M) = \frac{1}{M}\sum_{m=1}^{M} \Psi(X^{(m)};\theta), \quad X^{(m)} \sim p(\mathbf{x}|\mathbf{y};\theta). \tag{3.11}$$

This quantity can be used to derive an iterative gradient-based algorithm for likelihood maximization, which makes use of local approximations of the gradient. At each guess $\theta^{(k)}$ of the parameter, simulated samples are drawn from the posterior distribution $p(\mathbf{x}|\mathbf{y};\theta^{(k)})$, used to locally approximate $p(\mathbf{x}|\mathbf{y};\theta)$. Then the Newton's scheme is deployed,

$$\theta^{(k+1)} = \theta^{(k)} - [\nabla_\theta \hat{G}(\theta, \mathbf{X}^M)|_{\theta=\theta^{(k)}}]^{-1} \hat{G}(\theta^{(k)}, \mathbf{X}^M), \ k \geq 0, \tag{3.12}$$

where the Hessian is computed by derivation of (3.11) or by using numerical differences. When the derivation of the Hessian is problematic, other approximations can be used. In literature, it is often referred to this approach as *stochastic Newton's method* or *stochastic gradient ascent method*, depending on the approximation used for the Hessian, see e.g. [11], [43], [45].

The term *stochastic* comes from the fact that noisy estimates of the gradient are used. At each iteration, in fact, a new Monte Carlo integration is performed and different random numbers (the samples) are generated and used to estimate the likelihood function gradient. In Chapter 4, we will discuss in more details these stochastic algorithms.

One final consideration regards the practical implementation of the gradient estimate (3.11). In fact, it relies on the assumption that it is possible to draw samples according to the posterior $p(\mathbf{x}|\mathbf{y};\theta)$. However, we are assuming that, for intractable models, this posterior is not available and direct samples drawing is not possible, even in case a guess $\theta^{(k)}$ for θ is available. The same problem affects the implementation of the Monte Carlo Expectation-Maximization methods, presented in the next section. Hence, we will address this common issue later on, in Section 3.2.

3.1.2 The Monte Carlo Expectation-Maximization methods

The Monte Carlo Expectation-Maximization (MCEM) algorithm was first introduced by [14], who considered an intractable E-step. Thus, the principle consists of estimating the quantity in the E-step introduced in Section 2.3.2,

$$Q(\theta, \theta^{(k)}) = \mathbb{E}_{p(\mathbf{x}|\mathbf{y};\theta^{(k)})}[\log p(\mathbf{x}, \mathbf{y};\theta)], \tag{3.13}$$

via Monte Carlo integration,

$$Q(\theta, \theta^{(k)}) \approx \hat{Q}(\theta, \theta^{(k)}, \mathbf{X}^M) = \frac{1}{M} \sum_{m=1}^{M} \log p(X^{(m)}, \mathbf{y}; \theta), \;\; X^{(m)} \sim p(\mathbf{x}|\mathbf{y}; \theta^{(k)}). \tag{3.14}$$

The M-step is then replaced by the maximization of $\hat{Q}(\theta, \theta^{(k)}, \mathbf{X}^M)$, which provides $\theta^{(k+1)}$,

$$\theta^{(k+1)} = \arg\max_{\theta} \hat{Q}(\theta, \theta^{(k)}, \mathbf{X}^M). \tag{3.15}$$

The procedure is repeated in loop, until convergence. General convergence results are available in [47]. Similarly to the MCML methods, convergence to the true MLE can be easily shown when M goes to infinity, see also [48]. In this case, the MCEM sequence is viewed as a convergent Monte Carlo approximation of the ordinary EM. In [49], instead, the authors treat the samples size as increasing across the MCEM iterations, and establish convergence of the sequence as the iterations count goes to infinity.

For general stochastic nonlinear models, the M-step is intractable, and MCEM methods must be combined with numerical optimization methods. Thus, at each EM iterate, many iterations may be required just to converge to the next parameter guess. This is why, in [40], the author proposes to use only one iteration of a gradient-based method, after each E-step, proving local equivalence to classical EM. Given the current guess $\theta^{(k)}$, a gradient-based optimization scheme for solving the M-step in one iteration can be deployed,

$$\theta^{(k+1)} = \theta^{(k)} - [\nabla^2_\theta \hat{Q}(\theta, \theta^{(k)}, \mathbf{X}^M)|_{\theta=\theta^{(k)}}]^{-1} \nabla_\theta \hat{Q}(\theta, \theta^{(k)}, \mathbf{X}^M)|_{\theta=\theta^{(k)}}, \;\; k \geq 0. \tag{3.16}$$

In this case, gradient and Hessian are computed by differentiating the estimate (3.14). Also in this case, however, the Monte Carlo estimate used for the E-step renders the method a stochastic algorithm, similar to [44].

Furthermore, there is a strong connection between the MCEM method based on (3.16) and the MCML method based on gradient approximations, described in Section 3.1.1. In fact, if we look at the definitions of the Monte Carlo estimates in (3.11) and (3.14), we note that

$$\nabla_\theta \log p(\mathbf{y}; \theta) \approx \hat{G}(\theta, \mathbf{X}^M) = \nabla_\theta \hat{Q}(\theta, \theta^{(k)}, \mathbf{X}^M) \approx \nabla_\theta Q(\theta, \theta^{(k)}). \tag{3.17}$$

Thus, we can conclude that the two methods based on Monte Carlo simulations are practically equivalent. This is why, in the following, we will only consider the MCML methods, but we will have in mind that the use Monte Carlo estimates of the log-likelihood gradient entails the equivalence with MCEM methods.

Finally, also the MCEM methods assume the possibility to obtain i.i.d. samples from the posterior $p(\mathbf{x}|\mathbf{y};\theta^{(k)})$. However this is not true by assumption. Hence, approximate methods for sampling have to be developed. We address this issue in the next section.

3.2 Sampling from the posterior

Both MCML and MCEM methods require samples from the posterior distribution $p(\mathbf{x}|\mathbf{y};\theta^{(k)})$. For intractable stochastic nonlinear models, this distribution is not available. Thus, an indirect approach to the simulation of complex distributions needs to be implemented. Sample simulations based on Markov chain can be adopted. In particular, Markov Chain Monte Carlo (MCMC) methods [41] preserve the asymptotic results of the Monte Carlo methods, see Appendix 8.1, via the weak law of large numbers, for the number of passages through a recurrent state in an ergodic Markov chain, see [50], [51]. Moreover, almost sure convergence holds via the Ergodic Theorem, see [12].

These methods return samples from a Markov chain which is ergodic and stationary w.r.t. the distribution to approximate. Hence, simulating this Markov chain is intrinsically equivalent to a standard simulation from the distribution, with the difference being in the necessity to simulate more samples to achieve a given accuracy. An example of MCMC methods is the Metropolis-Hastings (MH) technique [52], [53]. When we deal with nonlinear state-space models, a very efficient technique for sample simulation is represented by the Sequential Monte-Carlo (SMC) techniques [19], [54], [55]. Those techniques are also known as Particle Filters and they implement an efficient simulation of high dimension random variables. MCML and MCEM methods that make use of particle filters can be found in [19], [56].

In the following, we will assume that MCMC methods are available for sample simulation from $p(\mathbf{x}|\mathbf{y};\theta)$. Hence, when we write

$$X^{(m)} \sim p(\mathbf{x}|\mathbf{y};\theta^{(k)}), \tag{3.18}$$

we actually imply the implementation of a MCMC sampling routine, providing the samples set $\{X^{(m)}\}_{m=1}^{M}$. We make this assumption because the methods we developed in this thesis for ML estimation do not rely on a particular implementation of the sampling method. In this way, depending on the specific problem and model definition, one can choose the most suitable sampling method, e.g. particle filters, Metropolis-Hastings, Gibbs sampling, etc. The only requirement is that the simulated samples have to guarantee the asymptotic results of the Monte Carlo estimate.

Nevertheless, for completeness, we provide a practical implementation of the samples simulation from $p(\mathbf{x}|\mathbf{y};\theta)$ based on Metropolis-Hastings. This is detailed in the next section.

3.2.1 The Metropolis-Hastings technique

The Metropolis-Hastings algorithm is a Markov chain Monte Carlo (MCMC) method for obtaining a sequence of random samples from a probability distribution for which direct sampling is difficult. The distribution to sample from is defined *target* distribution. In our case, the target is the posterior $p(\mathbf{x}|\mathbf{y};\theta)$ that can be expressed as

$$p(\mathbf{x}|\mathbf{y};\theta) = \frac{p(\mathbf{x},\mathbf{y};\theta)}{p(\mathbf{y};\theta)} = \frac{p(\mathbf{y}|\mathbf{x};\theta)p(\mathbf{x};\theta)}{p(\mathbf{y};\theta)} \tag{3.19}$$

where at the denominator we find the unknown likelihood function. However, the knowledge of this normalizing constant is not required by the MH algorithm. The algorithm is explained in the following.

1. Starting from an arbitrary initial sample $X^{(0)}$, a candidate is sampled from a *proposal* distribution $X^{(m+1)} \sim q(\mathbf{x}|X^{(m)})$.

2. An acceptance ratio is computed as

$$a = \frac{p(X^{(m+1)}|\mathbf{y};\theta)}{p(X^{(m)}|\mathbf{y};\theta)}. \tag{3.20}$$

3. The posterior distribution is further factorized and expressed as ratio between the joint likelihood of $\mathbf{x}$ and $\mathbf{y}$, available by assumption, and the likelihood of $\mathbf{y}$, which is unknown,

$$a = \frac{p(X^{(m+1)}|\mathbf{y};\theta)}{p(X^{(m)}|\mathbf{y};\theta)} = \frac{p(X^{(m+1)},\mathbf{y};\theta)}{p(\mathbf{y};\theta)} \frac{p(\mathbf{y};\theta)}{p(X^{(m)},\mathbf{y};\theta)}. \tag{3.21}$$

   The unknown quantity simplifies and the ratio is only function of known, analytical available quantities,

$$a = \frac{p(X^{(m+1)},\mathbf{y};\theta)}{p(X^{(m)},\mathbf{y};\theta)} = \frac{p(\mathbf{y}|X^{(m+1)};\theta)p(X^{(m+1)};\theta)}{p(\mathbf{y}|X^{(m)};\theta)p(X^{(m)};\theta)}, \tag{3.22}$$

4. The ratio a is translated into a probability $p = \min\{1, a\}$.

5. The candidate is accepted with probability p, otherwise it is rejected.

6. A new candidate is then generated and the procedure is repeated in loop.

The dimension of the sample is the dimension of the signals $\mathbf{x} \in \mathbb{R}^N$. For system identification problems, this dimension corresponds to the length of the data used of estimation. In order to ensure consistency results of the ML estimate, a long data sequence is desirable. This makes the sample simulation more problematic. In fact, already with $N \geq 50$, the probability of generating samples within the multi-dimension 6-σ sphere is almost zero, see [53]. However, a modification of the Metropolis-Hastings techniques can be used in this case, i.e. the Component-Wise Metropolis-Hastings [53]. Instead of drawing one N-dimension sample from the proposal, the algorithm draws one scalar component of the sample at the time. Then, the acceptance/rejection routine is applied to single components. Similar ideas are at the base of the particle filter, where the samples simulation is done by propagating the sample components through the system dynamics.

Another drawback of the MH method is that the generated samples may show high correlation, since the simulation is based on a Markov chain. This may compromise the asymptotic results of the MC integration, valid for independent random variables. However, solutions to the problem were already proposed by [53], and more recent approaches can be found in [57], where a smart choice of the proposal distribution $q(x)$ addresses the correlation problem.

3.3 Summary

In this chapter, we presented the main methods for ML estimation in case of intractable stochastic nonlinear models. They are the Monte Carlo Maximum Likelihood and the Monte Carlo Expectation-Maximization methods. With the use of Monte Carlo simulations, it is possible to establish asymptotic convergence results, at least to a local optimizer of the true likelihood function. However, the methods have proven convergence properties only in case of infinite sample size or when it grows across the iterations count. In practice, these conditions are hardly achievable because the sample size to be used is mainly imposed by the available computational budget.

The MCML methods attempt to overcome this issue by nesting the integration and the optimization stages. The nested procedure uses the current guess of the parameter for a local samples simulation. From an algorithmic perspective, this makes the MCML similar to the MCEM, where the nesting and the local sampling is already implemented. In the next chapter, we will discuss in more details the nested algorithms, by analysing advantages and drawbacks.

Chapter 4

Nested methods for MLE

The MCML and MCEM methods presented Chapter 3 address the intractability problem of stochastic nonlinear models identification by making use of Monte Carlo methods. These methods provide exact estimates of the intractability quantities when the sample size goes to infinity. In practice, however, only a finite sample size can be used. Thus, the algorithms are modified by nesting the MC estimate and the numerical optimization stages, allowing to work with local approximations of the intractable quantities.

In this chapter, we analyse the nested algorithms and we explain how they are useful for the finite sample size problem. Then, we address the main issues emerging from the use of a nested algorithm, i.e. noisy parameter search and increasing Monte Carlo errors, by proposing two modifications.

4.1 A nested MCML method

In this section, we analyse in details the pros and cons of nesting the MC integration and the numerical optimization stages, in case of finite sample size M available for integration. At the end of Section 3.1.2, we showed the equivalence between the MCML method based on gradient approximations and the MCEM method where the M-step is approximated with one iteration of a gradient-based scheme. Thus, we will focus on the MCML method in the following.

The gradient $\nabla_\theta \log p(\mathbf{y};\theta)$ of the log-likelihood is approximated via Monte Carlo integration, see Equation (3.11),

$$\hat{G}(\theta, \mathbf{X}_k^M) = \frac{1}{M}\sum_{m=1}^{M} \Psi(X_k^{(m)};\theta), \tag{4.1}$$

where $\{X_k^{(m)}\}_{m=1}^M$ are samples simulated from $p(\mathbf{x}|\mathbf{y};\theta^{(k)})$. This integration

introduces a *Monte Carlo error*, defined as

$$\varepsilon_k(\theta) = \nabla_\theta \log p(\mathbf{y};\theta) - \frac{1}{M}\sum_{m=1}^{M} \Psi(X_k^{(m)};\theta), \tag{4.2}$$

where the subscript k indicates that the samples set $\mathbf{X}_k^M$ is simulated from $p(\mathbf{x}|\mathbf{y};\theta^{(k)})$. The main purpose of the MC estimate is to keep this error as small as possible. Clearly, if $M \to \infty$, the error tends to 0. For a finite sample size M, however, the Monte Carlo error is a function of the parameter θ. In fact, since the random samples are simulated according to $p(\mathbf{x}|\mathbf{y};\theta^{(k)})$, their distribution is able to provide an accurate estimate of $\nabla_\theta \log p(\mathbf{y};\theta)$ in a close neighbourhood of $\theta^{(k)}$. At any other value $\theta^{(j)} \neq \theta^{(k)}$, only few samples from the distribution $p(\mathbf{x}|\mathbf{y};\theta^{(k)})$ may be useful. This results into a reduction of the effective number M of samples used for estimation. Hence, in general we have that

$$|\varepsilon_k(\theta^{(k)})| \leq |\varepsilon_k(\theta^{(j)})|, \ \forall\ \theta^{(j)} \in \Theta,\ \theta^{(j)} \neq \theta^{(k)}. \tag{4.3}$$

Thus, if the samples are only simulated once, from an initial guess of θ, and then the MC estimate of the gradient is used in an optimization loop, the Monte Carlo error would grow across the iterations, resulting into an increasing error of the parameter estimate too.

This is the main reason behind the nesting idea: a new samples set has to be generated from each new guess $\theta^{(k+1)}$ of the parameter during the optimization loop. In this way, at each iteration, all the M samples contribute to build local approximations of the gradient. The actual magnitude of each error, then, will depend on the value of M used. Also the error of the parameter estimate will then only depend on M, which can be defined based on the desired accuracy of the estimate.

Based on these considerations, we introduce the *nested MCML* method, whose main steps are described in the following.

1. Start from initial guess $\theta^{(k)}$, with $k = 0$.

2. Simulate new samples $\mathbf{X}_k^M = \{X_k^{(m)}\}_{m=1}^M$, where $X_k^{(m)} \sim p(\mathbf{x}|\mathbf{y};\theta^{(k)})$.

3. Compute a new MC estimate of the log-likelihood gradient

$$\hat{G}(\theta, \mathbf{X}_k^M) = \frac{1}{M}\sum_{m=1}^{M} \Psi(X_k^{(m)};\theta) \tag{4.4}$$

   where $\Psi(\mathbf{x};\theta)$ is defined in (3.9).

4. Take one Newton's step

$$\theta^{(k+1)} = \theta^{(k)} - [\nabla_\theta \hat{G}(\theta, \mathbf{X}_k^M)|_{\theta=\theta^{(k)}}]^{-1} \hat{G}(\theta, \mathbf{X}_k^M)|_{\theta=\theta^{(k)}}. \tag{4.5}$$

5. Set $k = k + 1$ and check if the parameter search converged.

6. If convergence is achieved, terminate the procedure; otherwise go to step 2.

Hence, the nested MCML method allows to find accurate estimation of the parameters, in the finite sample size case. The accuracy depends on M. However, a main drawback emerges from this formulation of method, explained in the following. Since at each iterate, a new, random samples simulation is performed, the function approximating the gradient is altered every time. This results into an indirect modification of the ML criterion that we are trying to maximise. As a consequence, the parameter search may show a noisy behaviour, poor convergence rate, and, in extreme cases, instability.

In summary, the parameter search shows different behaviours based on how often, across the iterations, the samples simulation is performed. The two extreme cases are:

- **Only one samples simulation** at the start of the algorithm. In this case, the function estimating the gradient is not altered across the iterations, providing a stable behaviour of the search. If full Newton's steps are deployed at each iterate, the method can also benefit from the quadratic convergence. However, since M is finite, the Monte Carlo error will increase across the iterations, resulting into an increasing error of the parameter estimate too.

- **Re-sampling at each iterate**. In this case, the final parameter estimation error only depends on the chosen M. On the other hand, the function approximating the gradient is changed at each iteration and the search is noisy.

Thus, the challenge is to come up with intermediate solutions addressing both the large error and the noisy search problem. In the following, we present two new methods addressing these issues: the *deterministic method* and the *partial re-sampling method*. The first one tries to avoid the re-sampling stage at each iterate by working with local, deterministic approximations of the likelihood function. The second one, instead, is a direct modification of the nested MCML method. In order to stabilize the search, a re-sampling rule is derived in order to understand which samples, at each iterate, can be useful for future iterates too. This second solution shows superior performance

in terms of accuracy and convergence rate, as it will be explained in the following.

4.2 The deterministic method

The main idea is to fix, at each iterate, the samples set $\mathbf{X}_k^M$ and to deploy the optimization procedure. The solution of this optimization is then used as a next guess of the parameter search, and a new samples simulation is performed from it. In this way, the MC estimate can be considered as a deterministic function of θ that does not change during the optimization phase, resulting into a stable algorithm. The procedure is then repeated in loop until final convergence. The scheme is detailed in the following.

1. Start from initial guess $\theta^{(k)}$, with $k = 0$.
2. Simulate samples $\mathbf{X}_k^M = \{X_k^{(m)}\}_{m=1}^M$, where $X_k^{(m)} \sim p(\mathbf{x}; \theta^{(k)})$.
3. Compute the MC estimate of the likelihood with importance sampling, see (3.6),

$$\hat{p}(\mathbf{y}; \theta, \mathbf{X}_k^M) = \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{y}|X_k^{(m)}; \theta) \frac{p(X_k^{(m)}; \theta)}{p(X_k^{(m)}; \theta^{(k)})}. \tag{4.6}$$

   With a fixed set $\mathbf{X}_k^M$, $\hat{p}(\mathbf{y}; \theta, \mathbf{X}_k^M)$ can be considered as a deterministic function of θ.

4. Solve an optimization problem in order to obtain the new guess $\theta^{(k+1)}$,

$$\theta^{(k+1)} = \arg\max_{\theta} \hat{p}(\mathbf{y}; \theta, \mathbf{X}_k^M). \tag{4.7}$$

5. Set $k = k + 1$ and check if the parameter search converged. This can be assessed by measuring the relative variation among consecutive guesses of the parameter. If this variation is within some pre-defined tolerance bound, convergence is achieved.
6. If convergence is achieved, terminate the procedure; otherwise go to step 2).

Hence, the method proceeds through local, deterministic approximations of the likelihood function and the optimization in (4.7) can be solved by a standard deterministic optimization tool. The method was originally presented in [58]. However, it is important to mention that a similar method has been independently developed for nonlinear state-space models, see [59].

The main drawback of this method is that we introduced inner optimization loops. Hence, the overall convergence rate may still be very poor. For these reasons, a more advanced solution has been developed and it will be presented in Section 4.3.

Nevertheless, the deterministic method shows good performance in finding accurate, consistent estimates for a stochastic WH model.

4.2.1 Application to stochastic WH models

In this section, we apply the deterministic method to a stochastic WH model identification. In particular, we want to illustrate two main characteristics of the deterministic method, compared to other methods used for the identification of this type of models:

1. As the deterministic method is computing the MLE, we expect it to find unbiased estimates of the parameters of the model. To illustrate this, we also compare the method with the standard PEM estimator, where we derive the output's predictor neglecting the presence of process noise. We show that, in this case, the PEM always leads to biased estimates, even in the case it can be formulated as a Linear Least Squares problem.

2. The deterministic method proceeds through local approximations of the likelihood. Hence, we expect that the method is able to find the solution with a reduced Monte Carlo sample size. For this, we compare the deterministic method with a standard MCML method for Wiener-Hammerstein ML estimation, as the one presented in [61], which implements a global sampling, i.e. only one samples simulation with enormous M.

A stochastic WH model

We consider the stochastic WH model, as defined in Section 2.1.3. For simplicity, we assume that the parameters of the linear parts are known, and the estimation only concerns the parameters on the nonlinearity. We recall the model structure,

$$\begin{aligned} x_t &= G_W(q, \theta_W)u_t + w_t, \\ z_t &= f(x_t, \theta_{NL}), \\ y_t &= G_H(q, \theta_H)z_t + e_t, \end{aligned} \tag{4.8}$$

and we assume that θ_W and θ_H do not need to be estimated. The signal w_t is the *process noise* and it is filtered via the nonlinear function f. This makes the WH model a stochastic nonlinear model.

Inconsistency of the PEM estimator

In this section we show that, for the stochastic WH model (4.8), a standard PEM estimator is not consistent. We first assume that the true system is within the model class, i.e. there exist parameters (θ_W^0, θ_{NL}^0, θ_H^0) such that the true output can be defined as

$$y_t = G_H(q, \theta_H^0) f(G_W(q, \theta_W^0) u_t + w_t, \theta_{NL}^0) + e_t. \tag{4.9}$$

A standard one-step-ahead predictor for this system is defined by neglecting the presence of the noise w_t,

$$\hat{y}_{t|t-1} = G_H(q, \theta_H) f(G_W(q, \theta_W) u_t, \theta_{NL}). \tag{4.10}$$

This definition of the predictor leads to the following PEM criterion

$$V_N(\theta_W, \theta_{NL}, \theta_H) = \frac{1}{N} \sum_{t=1}^{N} (y_t - G_H(q, \theta_H) f(G_W(q, \theta_W) u_t, \theta_{NL}))^2. \tag{4.11}$$

Consistency of the parameters means that

$$\hat{\theta}_W, \hat{\theta}_{NL}, \hat{\theta}_H \to \theta_W^0, \theta_{NL}^0, \theta_H^0 \quad \text{when} \quad N \to \infty, \tag{4.12}$$

where

$$\hat{\theta}_W, \hat{\theta}_{NL}, \hat{\theta}_H = \arg \min_{\theta_W, \theta_{NL}, \theta_H} V_N(\theta_W, \theta_{NL}, \theta_H). \tag{4.13}$$

We assume that the linear parameters are known and fixed in the estimation, i.e. $\theta_W = \theta_W^0$, $\theta_H = \theta_H^0$. Then we have the following theorem.

Theorem 4.2.1. *(Inconsistency of a standard PEM estimator) Let the nonlinearity $f(x_t, \theta_{NL})$ be polynomial or well approximated by a polynomial function. Under the assumption of ergodicity, the estimate of θ_{NL}, obtained by minimization of the PEM criterion (4.11) is inconsistent.*

The proof can be found in Appendix 8.3. The PEM criterion (4.11) is the standard least squares error cost function that is usually chosen as criterion of fit. However, here we showed that in the presence of process noise, this is clearly the wrong criterion for parameter estimation. In order to get a consistent PEM estimator, the predictor (4.10) and the metric used in (4.11) should take into account the correct stochastic description of the data.

Numerical example

In this section, the deterministic method implementing the ML estimator and the PEM estimator are tested on a numerical example. Noisy data are

simulated using the stochastic W-H system,

$$x_t = \frac{1}{1-\alpha q^{-1}} u_t + w_t \tag{4.14a}$$

$$z_t = f(x_t, \theta_{NL}) \tag{4.14b}$$

$$y_t = \frac{1}{1-\beta q^{-1}} z_t + e_t. \tag{4.14c}$$

with $f(x_t, \theta_{NL})$ being a third degree polynomial $f(x_t, \theta_{NL}) = c_0 + c_1 x_t + c_2 x_t^2 + c_3 x_t^3$, and $\theta_{NL} = [c_0, c_1, c_2, c_3] \in \mathbb{R}^4$. The process and output noise are respectively white and Gaussian with standard deviations $\sigma_w = 4$, $\sigma_e = 1$. The signals u, w, and e are mutually independent. Knowing this, the distributions needed for the deterministic method, see (4.6), can be defined,

$$\begin{aligned} p(y_t|x_t;\theta) &= \frac{1}{\sqrt{2\pi\sigma_e^2}} \mathrm{e}^{-\frac{1}{2\sigma_e^2}(y_t - G_H(q,\beta) f(x_t,\theta_{NL}))^2}, \\ p(x_t;\theta) &= \frac{1}{\sqrt{2\pi\sigma_w^2}} \mathrm{e}^{-\frac{1}{2\sigma_w^2}(x_t - G_W(q,\alpha)u_t)^2}. \end{aligned} \tag{4.15}$$

The PEM estimator is defined and computed as minimization of

$$V_N(\theta_{NL}) = \frac{1}{N} \sum_{t=1}^{N} (y_t - G_H(q,\beta) f(G_W(q,\alpha)u_t, \theta_{NL}))^2, \tag{4.16}$$

which, given the parametrization of $f(x_t, \theta_{NL})$, results into a linear least squares problem.

The parameter of the linear parts are known and fixed during the estimation, $[\alpha, \beta] = [0.4, 0.8]$, while the parameters of the nonlinearity are estimated. The estimated parameters with PEM and with the deterministic method are reported in Figure 4.1. A data-set of 1000 input-output points have been used for estimation. For the deterministic method a sample size of $M = 100$ has been used for the MC estimate (4.6). Since the deterministic method implements the ML estimator, it is labelled ML in the figure. As expected, the the PEM estimate is biased, while the ML is not. In particular, we want to stress the fact that, in this case, the PEM estimator is a linear least squares problem, since the parameters of the linear parts are fixed and the nonlinearity is linearly parametrized in θ_{NL}. Hence, the biased estimates of the PEM are not the result of local minima effects, but they are a consequence of the inconsistency of the PEM. On the other hand, the MLE is a nonlinear problem, thus it may suffer of local optimizer effects. Hence, to illustrate consistency, we initialized the MLE problem in a close neighbourhood of the true solution. We will address the initialization problem of the MLE in the next part of the thesis. We can conclude that, with a

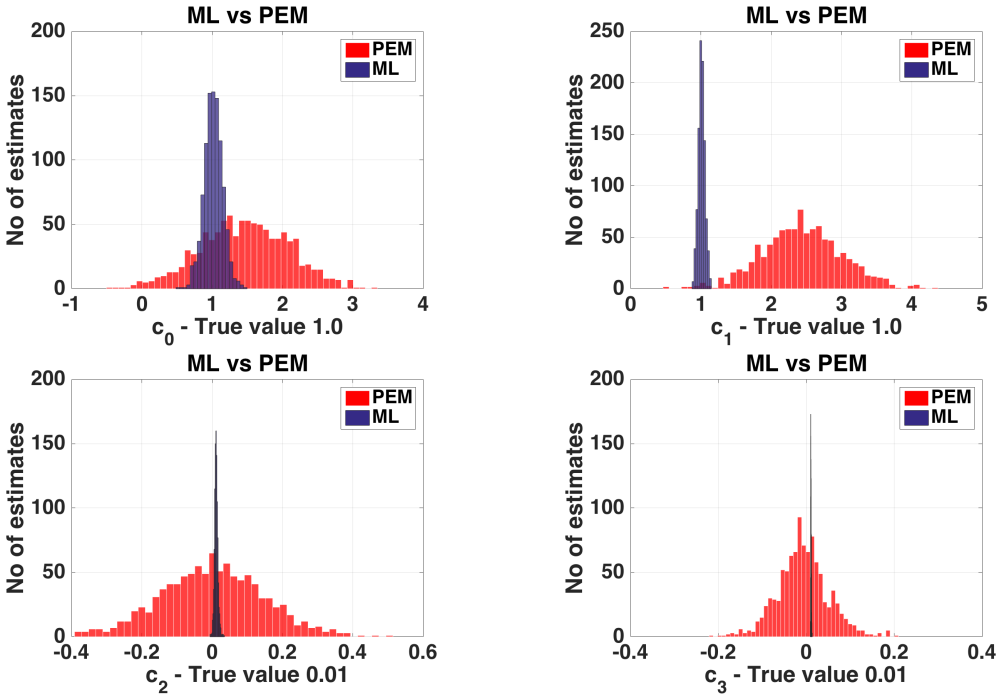


Figure 4.1: PEM vs ML (deterministic method). Estimates of the nonlinear parameters. Histograms over 1000 Monte Carlo simulations. The ML estimates are unbiased and they also show quite narrow variance. The PEM estimates are biased.

good initial guess, the ML method is able to find the true solution, while the standard PEM will always provide biased results.

From a computational perspective, the deterministic method shows a significant reduction of the required sample size. In fact, compared to a standard MCML method, as the one presented in [61], the sample size is reduced from $M = 30000$ to $M = 100$. The main reason is that the deterministic method works with local approximations of the likelihood, by the use of a local (around the current value of θ) sampling. The MCML method of [61], instead, is an implementation of the method described in [42], [13], where a whole picture of the likelihood is built from only one samples simulation, requiring enormous sample size. Furthermore, none of the estimations performed using the deterministic method led to instability, indicating that noisy behaviour is actually reduced.

On the other hand, a drawback of the deterministic method is that, at each iteration, a full optimization problem has to be solved till convergence, see (4.7). In Table 4.1, the main indicators for computational analysis are reported. Compared to the standard MCML method of [61], the overall number of required iterations for convergence is higher for the deterministic

Table 4.1: Indicators for computational analysis. M = Sample size for MC estimate; Iters = Average number of iterations for convergence; Inner iters = Average number of iterations for convergence of the inner optimization loop (4.7); Time = Average time per iteration (inner loop iterations in case of deterministic method)

Method	M	Iters	Inner iters	Time (s)
Deterministic Method	100	9.52	4.3	0.45
Standard MCML	30000	21.4	-	5.21

method. This is mainly due to the inner optimization loops. However, at each iteration the deterministic method solves a smaller optimization problem, thanks to the smaller sample size, resulting into a reduction of the time per iteration.

4.3 The partial re-sampling method

The partial re-sampling method is a direct modification of the nested MCML method, discussed in Section 4.1. The main drawback of the nested MCML method is that a new, samples simulation is performed at each iteration, altering the function approximating the gradient.

The main idea is then to keep, at each iteration, a considerable part of the samples set, and reuse it in the next iteration. In this way, the samples sets are only slightly altered, resulting into a smoother change among the functions approximating the gradient. Furthermore, the samples that cannot be reused allow to simulate new samples from the current guess of the parameter. In this way, all the M samples contribute to build the MC estimate of the gradient and the estimation error does not increase.

The challenge is then to come up with a rule deciding whether samples generated at iteration k can be used at iteration $k+1$.

In literature, similar ideas have been developed in [15] and [62], with focus on noise reduction for MCEM methods. The proposed solutions implement a smooth, stochastic update of the quantity to maximise: all the samples from the previous iteration are used again, and only few new ones are simulated from the current guess of the parameter. This is the Stochastic Approximation version of the EM method (SAEM). Since, in this way, all samples are kept from one iteration to another, the noisy behaviour is drastically reduced. However, using all samples from previous iterations

might also be quite inefficient. The sample size has to increase at each iterate with a related increase of the required computational resources for storage and evaluation. With our idea, instead, the samples size M can be kept fixed.

In the following, the key aspects of the new idea are presented. Firstly, in order to correctly reuse samples simulated at old guesses of θ for computing MC estimates at the current guess, a correction to the MC estimates, based on importance sampling, has to be implemented. Then, we present the re-sampling rule for the selection of old samples that can be reused in the future iterations. Finally, we discuss some convergence and complexity aspects of the algorithm.

4.3.1 Importance sampling for correction

As already discussed in Section 3.1.1, the MC estimates require a correction when the sampled functions are evaluated at a different parameter value from the the one used for samples simulation. This correction can be done by using importance sampling, e.g. see Equation (3.6). In a similar fashion, the proposed solution requires that samples simulated for old values of the parameter are reused to evaluate the log-likelihood gradient of the nested MCML method, see (4.4), at the new values of θ. Hence, we correct (4.4) via importance sampling. At each iteration, the distribution $p(\mathbf{x}|\mathbf{y};\theta^{(k)})$ acts as a *proposal* and (4.4) is corrected via

$$\hat{G}(\theta, \mathbf{X}_k^M) = \frac{1}{M}\sum_{m=1}^{M} \Psi(X_k^{(m)};\theta)\frac{p(X_k^{(m)}|\mathbf{y};\theta)}{p(X_k^{(m)}|\mathbf{y};\theta^{(k)})} \tag{4.17}$$

Clearly, when $\theta = \theta^{(k)}$, the important ratio simplifies to one and the original MC estimate (4.4) is retrieved. The importance sampling correction used in (4.17) requires the evaluation of the posterior $p(\mathbf{x}|\mathbf{y};\theta)$ on the samples $\{X_k^{(m)}\}_{m=1}^M$. By assumption, the posterior is not available in closed-form, since it requires the knowledge of the likelihood function $p(\mathbf{y};\theta)$,

$$p(\mathbf{x}|\mathbf{y};\theta) = \frac{p(\mathbf{x},\mathbf{y};\theta)}{p(\mathbf{y};\theta)}. \tag{4.18}$$

Nevertheless, this issue is solved by implementing a *self-normalizing* importance sampling operation, which is often used for weighting distributions that are only known up to a normalizing constant, see Appendix 8.4. As a result, we obtain that (4.17) is reformulated as

$$\hat{G}(\theta, \mathbf{X}_k^M) = \frac{\frac{1}{M}\sum_{m=1}^{M} \Psi(X_k^{(m)};\theta) w(X_k^{(m)})}{\frac{1}{M}\sum_{m=1}^{M} w(X_k^{(m)})}, \tag{4.19}$$

where the weights $w(\mathbf{x})$ are only function of the joint PDF $p(\mathbf{x}, \mathbf{y}; \theta)$, available for evaluation, see Assumption 2.3.1,

$$w(\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{y}; \theta)}{p(\mathbf{x}, \mathbf{y}; \theta^{(k)})}. \tag{4.20}$$

The derivation of (4.19) is detailed in Appendix 8.4.

4.3.2 The samples selection

The sample selection procedure has to decide which samples from previous iterations can be reused in the future. The idea is to keep, at each iterate, only samples that have a *high impact* on the parameter search in the next update, and simulate new samples in place of the *low impact* ones, which do not contribute to the search. Assume that, at iteration k, a sample set $\mathbf{X}_k^M$ is available to compute (4.19). Then (4.5) can be deployed,

$$\theta^{(k+1)} = \theta^{(k)} - [\nabla_\theta \hat{G}(\theta, \mathbf{X}_k^M)|_{\theta=\theta^{(k)}}]^{-1} \hat{G}(\theta^{(k)}, \mathbf{X}_k^M). \tag{4.21}$$

The ideal samples selection procedure will now select a set of *high impact* samples, denoted by $\mathbf{X}_k^H \subseteq \mathbf{X}_k^M$ and composed of $H_k \leq M$. The $L_k = M - H_k$ remaining samples are defined as *low impact* samples, and collected in the set $\mathbf{X}_k^L \subseteq \mathbf{X}_k^M$. In their place, $M - H_k$ new samples are simulated from $\theta^{(k+1)}$,

$$\{X_{k+1}^{(m)}\}_{m=1}^{M-H_k} \sim p(\mathbf{x}|\mathbf{y}; \theta^{(k+1)}). \tag{4.22}$$

This set of new simulated samples is appended to the old H_k samples, and used to define the samples set for $\theta^{(k+1)}$,

$$\mathbf{X}_{k+1}^M \triangleq \{\mathbf{X}_k^H, \{X_{k+1}^{(m)}\}_{m=1}^{M-H_k}\}, \tag{4.23}$$

in order to compute $\theta^{(k+2)}$. In this way, all the M samples are useful to compute the MC estimate and the estimation error does not increase. Furthermore, thanks to $\mathbf{X}_k^H$, the MC estimates between two consecutive iterations is altered less compared to the full re-sampling case, as the one in the nested MCML method, see Step 2.

In the next section, we propose a practical procedure to select the high and low impact samples.

4.3.3 The sensitivity-based re-sampling rule

The practical implementation for selection of high and low impact samples can be based on the sensitivity of the parameter search with respect to the samples. In fact, each parameter guess $\theta^{(k)}$ is retrieved by implementing

the optimization procedure described in Section 4.1, i.e. the nested MCML method. Thus, the parameter guess can be seen as an implicit function of the samples, i.e. $\theta^{(k)} = \theta^{(k)}(\mathbf{X}_k^M)$. Assuming that we are at the solution of the optimization procedure, $\theta^{(k)} = \theta^*$, then the following must hold,

$$\hat{G}(\theta^{(k)}(\mathbf{X}_k^M), \mathbf{X}_k^M) = 0, \ \forall \ \mathbf{X}_k^M. \tag{4.24}$$

Using the implicit function theorem [63], we can then retrieve the following relation,

$$\frac{\partial \theta^{(k)}}{\partial \mathbf{X}_k^M} = - \left[\nabla_\theta \hat{G}(\theta, \mathbf{X}_k^M)|_{\theta=\theta^{(k)}}\right]^{-1} \left[\frac{\partial \hat{G}(\theta, \mathbf{X}_k^M)}{\partial \mathbf{X}_k^M}\bigg|_{\theta=\theta^{(k)}}\right]. \tag{4.25}$$

Although the previous relation is derived for $\theta^{(k)} = \theta^*$, it still provides a good approximation to the sensitivity of the parameter $\theta^{(k)}$ w.r.t. the samples, also during the search. Hence, we have found a rule to select high and low impact samples. We call this rule the *sensitivity-based re-sampling* rule. At each iterate, the sets of high and low impact samples can be defined in the following way

$$\mathbf{X}_k^H \triangleq \left\{ X_k^{(m)} \in \mathbf{X}_k^M : \left\| \frac{\partial \theta^{(k)}}{\partial X_k^{(m)}} \right\|_2 \geq \bar{s} \right\}, \tag{4.26}$$

and

$$\mathbf{X}_k^L \triangleq \left\{ X_k^{(m)} \in \mathbf{X}_k^M : \left\| \frac{\partial \theta^{(k)}}{\partial X_k^{(m)}} \right\|_2 < \bar{s} \right\}. \tag{4.27}$$

The scalar $\bar{s}$ is a non-negative threshold for the sensitivity, and it can be used as a tuning parameter to control the ratio between the number of old samples to keep and new ones to simulate.

The threshold $\bar{s}$ plays an important role. In fact, if $\bar{s} = 0$, then all samples are kept, $\mathbf{X}_{k+1}^M \equiv \mathbf{X}_k^H \equiv \mathbf{X}_k^M$, and no samples are simulated from $\theta^{(k+1)}$. As discussed before, in this case the MC error will increase across the iterations. On the other hand, if $\bar{s}$ is too high, then all samples are discarded and, at each iterate, a whole new set of samples is simulated, as in the standard nested MCML method, which suffers of stability problem. Hence, $\bar{s}$ should be chosen in order to keep a substantial part of old samples, for stability reasons, and to simulate new samples from the current guess of the parameter, for keeping the MC error limited.

Some practical guidelines on how to define $\bar{s}$ are given at the end of the next section.

4.3.4 Final identification algorithm

In this section we summarize the main steps of on the partial re-sampling method. Compared to the nested MCML methods, the main modification affects the samples simulation part; at each iterate, not a whole new samples set is simulated but the sensitivity-based re-sampling rule decides which samples can be kept from the previous iterate. The procedure is illustrated in the following.

1. Start from initial guess $\theta^{(k)}$, with $k = 0$.
2. Set indexes $H_k = 0$, $L_k = M$, and decide the value of the threshold $\bar{s}$.
3. Set the *high* impact samples set to $\mathbf{X}_{k-1}^H = \emptyset$.
4. Simulate samples set with dimension L_k,
   $\mathbf{X}_k^L = \{X_k^{(m)}\}_{m=1}^{L_k}$, where $X_k^{(m)} \sim p(\mathbf{x}|\mathbf{y};\theta^{(k)})$.
5. Define $\mathbf{X}_k^M$ by appending the high impact samples set $\mathbf{X}_{k-1}^H$ to $\mathbf{X}_k^L$,
   $\mathbf{X}_k^M = \{\mathbf{X}_{k-1}^H, \mathbf{X}_k^L\}$.
6. Compute the MC estimate of the log-likelihood gradient, with the importance sampling correction, see Equation (4.19), using $\mathbf{X}_k^M$,
$$\hat{G}(\theta, \mathbf{X}_k^M) = \frac{\frac{1}{M}\sum_{m=1}^{M} \Psi(X_k^{(m)};\theta) w(X_k^{(m)})}{\frac{1}{M}\sum_{m=1}^{M} w(X_k^{(m)})}, \quad X_k^{(m)} \in \mathbf{X}_k^M. \tag{4.28}$$
7. Take one full Newton's step
$$\theta^{(k+1)} = \theta^{(k)} - [\nabla_\theta \hat{G}(\theta, \mathbf{X}_k^M)|_{\theta=\theta^{(k)}}]^{-1} \hat{G}(\theta^{(k)}, \mathbf{X}_k^M). \tag{4.29}$$
8. Check the sensitivity of the paramter w.r.t the samples and define $\mathbf{X}_k^H$,
   $\mathbf{X}_k^H \triangleq \left\{ X_k^{(m)} \in \mathbf{X}_k^M : \left\| \frac{\partial \theta^{(k+1)}}{\partial X_k^{(m)}} \right\|_2 \geq \bar{s} \right\}$,
9. Set the index $H_k = \dim\{\mathbf{X}_k^H\}$, and $L_k = M - H_k$
10. Set $k = k + 1$.
11. Check if the parameter search converged. This can be assessed by measuring the relative variation among consecutive guesses of the parameter. If this variation is within some pre-defined tolerance bound, convergence is achieved. In this case, terminate the procedure; otherwise go to step 4).

We now discuss important aspects regarding the convergence and the complexity of the presented algorithm.

Convergence aspects

- **How to define** $\bar{s}$. As discussed before, the threshold $\bar{s}$ should guarantee both a stable search and the simulation of new samples from the current parameter. Therefore, a first idea would be to allow the threshold $\bar{s}$ to change across the iterations, i.e. $\bar{s} = \bar{s}_k$. In this way, at the beginning of the search, $\bar{s}_k$ can be kept quite low, favouring the generation of more samples during the first iterations. Then, when the search has moved in the neighbourhood of the solution, the relative variation among the parameter guesses is smaller and more samples can be kept among the iterations. Hence, $\bar{s}_k$ can be decreased, providing a stable convergence. Another idea, which has been implemented and tested in practice, is to define $\bar{s}$ as a fraction of the median sensitivity of all samples. In this way, $\bar{s}$ can still change during the iterations, but it does that in accordance with the overall sensitivity of the samples. As we will illustrate in Section 4.3.5, this choice of $\bar{s}$ shows some desirable behaviour. In particular, we observe that the number of samples that are changed automatically decreases across the iterations. This is a consequence of the fact that, close to convergence, the parameter does not vary a lot between the iterations and, thus, more samples can be reused.

- **Control of the Newton's step and rate of convergence**. The re-sampling routine implemented by the partial re-sampling method can be seen as a way to control the stability of the Newton's step. In fact, in presence of noisy behaviour, other techniques are usually deployed for stabilizing the search. E.g., in [11], [43], and [45], line-search techniques are implemented. From this point of view, the re-sampling rule has the effect of stabilizing the search, and full Newton's steps can be deployed at each iterate. Thanks to this, when the number of replaced samples considerably decreases, the algorithm can benefit from the quadratic convergence of the Newton's method. Furthermore, in this case, the value of the gradient can be used for assessing the convergence of the parameter search.

Complexity and computational aspects

- **Complexity of the posterior sampling routine**. As the other MCML and MCEM methods, a samples simulation from the posterior distribution $p(\mathbf{x}|\mathbf{y})$ is required. However, for intractable stochastic nonlinear model, this posterior is not available and the sample simulation cannot be performed directly. We discussed alternative solutions to this issue in Section 3.2. The idea is to use MCMC techniques to

approximate the samples simulation. In case of stochastic nonlinear state-space models, efficient samples simulation can be performed via SMC techniques. In any case, the complexity of this operation is in the order of $O(M \times N)$, where M is the number of samples to simulate and N is the dimension of each sample. Hence, for a standard MCML or MCEM method, where a full re-sampling is performed at each iteration, the operation of complexity $O(M \times N)$ has to be repeated at each iterate, until convergence. For the partial re-sampling method, instead, the complexity is $O(L_k \times N)$, where L_k denotes the number of samples which are discarded from one iteration to another. In general we observed that, except for the first few iterations, $L_k << M$, and it decreases across the iterations. Thus, the overall complexity of the algorithm is reduced.

- **Hessian computation**. The used of an exact Hessian is desireble in order to guarantee quadratic convergence. So far we assumed that it is possible to derive this second order information simply by differentiation of the gradient estimate. Actually, modern automatic Algorithmic Differentiation (AD) tools allow this operation by making use of symbolic expressions, see e.g. [36]. The only limitation may reside in the computational complexity required by these tools. In our case, the developed methods address the problem of reducing the sample size M in order to reduce the overall complexity. Hence, this reduction can be also beneficial for this sensitivity computational tools. When the complexity is anyway too high, common Hessian approximation techniques can be used. E.g., in [45] and [64], unbiased estimates of the Hessian are computed only using MC estimates of the gradient.

- **Sensitivity computation**. Some extra complexity is introduced by the implementation of the sensitivity-based re-sampling rule. However, the computation of this sensitivity is not more expensive than other techniques to control the stability of the Newton's iteration. As discussed before, some step-length control is usually implemented to stabilize the search, and this requires extra evaluation of cost function and sensitivities. In our case, the rule (4.25) requires the evaluation of the Hessian and the derivative of the gradient w.r.t. the samples. The first one is obtained at zero cost, since it has already been computed in the Newton's update. The second one may be more costly, as it may involve complicated expressions. However, a small modification can be performed to simplify this operation. It consists of introducing

a *virtual* weight $W_k^{(m)}$ for each sample $X_k^{(m)}$ used in the MC estimate,

$$\hat{G}(\theta, \mathbf{X}_k^M, \mathbf{W}_k^M) = \frac{\frac{1}{M}\sum_{m=1}^{M} W_k^{(m)} \Psi(X_k^{(m)}; \theta) w(X_k^{(m)})}{\frac{1}{M}\sum_{m=1}^{M} w(X_k^{(m)})}, \tag{4.30}$$

with $W_k^{(m)} \in \mathbf{W}_k^M$. In this way, the sensitivity (4.25) can be computed w.r.t. the virtual samples instead, which enters linearly in the expression of the MC estimate,

$$\frac{\partial \theta^{(k)}}{\partial \mathbf{W}_k^M} = -\left[\nabla_\theta \hat{G}(\theta^{(k)}, \mathbf{X}_k^M, \mathbf{W}_k^M)\right]^{-1} \left[\frac{\partial \hat{G}(\theta^{(k)}, \mathbf{X}_k^M, \mathbf{W}_k^M)}{\partial \mathbf{W}_k^M}\right]. \tag{4.31}$$

In the evaluations, the value of the virtual weights can be set to 1 for all the samples, in this way the MC estimate is not altered. The tool presented in [36] can be used to derive the sensitivity (4.31) by symbolically differentiation of the modified gradient expression (4.30).

One final consideration concerns the generality of the re-sampling rule implemented by this method. The rule mainly affects the sample simulation stage, where we assumed that a generic MCMC technique is implemented. Hence, the sensitivity-based re-sampling rule can be in principle adopted by any MLE method based on Monte Carlo approximations where a samples simulation stage is nested with the numerical optimization iterations. For example, this could be the case of the methods based on Sequential Monte Carlo, see [17], [19].

4.3.5 Numerical examples

In this section, we evaluate the partial re-sampling method (PRM) by considering a stochastic nonlinear state-space model. Firstly, we describe the model under test. Then, we apply the partial re-sampling method and we compare it with other two MCML methods, described in the following:

1. A nested MCML (NMCML) method, as the one described in Section 4.1, where a new, random samples simulation is performed at each iteration;

2. A one-sampling MCML (OMCML) method, i.e. a modification of the nested MCML method, where the samples simulation is only performed from the initial guess of the parameter.

From this comparisons, we expect the following results:

1. The NMCML should show small estimation error but quite noisy search behaviour.

2. The OMCML should show a more stable search behaviour, but a higher estimation error.

3. The PRM should inherits the benefits of the two previous methods, i.e. a stable search and a small estimation error.

Finally, we also compare the PRM with the MCML proposed in [45], i.e. a nested method implementing Monte Carlo estimates of the gradient and local sampling based on particle filters. The authors label this method as ALG3FL. The method is a stochastic Newton's scheme, where the stability problem is addressed by making use of a line-search algorithm. This methods performs a new samples simulation at each iterate. Hence, it is able to provide small estimation errors. However, it addresses the stability problem via line-search. Compared to our PRM, it shows slower convergence rate.

A stochastic nonlinear state-space model

The state-space model under test is the following,

$$\begin{aligned} x_{t+1} &= \theta_1 \arctan x_t + w_t, \quad && w_t \sim \mathcal{N}(0,1) \\ y_t &= \theta_2 x_t + e_t, && e_t \sim \mathcal{N}(0, 0.1^2). \end{aligned} \tag{4.32}$$

The unknown parameter vector is $\theta = [\theta_1, \theta_2]$. For analysis, we simulated 100 data sets, consisting of $N = 500$ input-output data points each. The true parameter value is $\theta_0 = [0.7, 0.5]$ and each estimation experiment is initialized with $\theta^{(0)} = [0.5, 0.7]$.

Firstly, in Figure 4.2, we compare estimates from the PRM, the NMCML, and the OMCML, on the same data-set. Those estimates are representative of the average behaviour over the 100 data sets. As expected the NMCML shows noisy behaviour but small error; the OMCML is more stable but the error is bigger; the PRM is both stable and accurate. The reason of these different behaviours is further explained by the value of the infinity norm of the gradient estimates across the iteration, reported in Figure 4.3. The NMCML gradient stays noisy across all the iterations; the OMCML gradient is stable, but the introduced MC errors drift away its solution; the PRM gradient is a bit noisy at the beginning of the search, but then it stabilizes. This is mainly due to the re-sampling rule implemented by the PRM, which reuses a different number of samples across the iterations. In Figure 4.4, the gradient of the PRM is reported together with the number of replaced samples per iteration.

From the previous figures, we can conclude that the PRM converges to a stable solution after 14-15, i.e. when the infinity norm of the gradient is lower than 10^{-10}. The ALG3FL method from [45] provides similar solutions

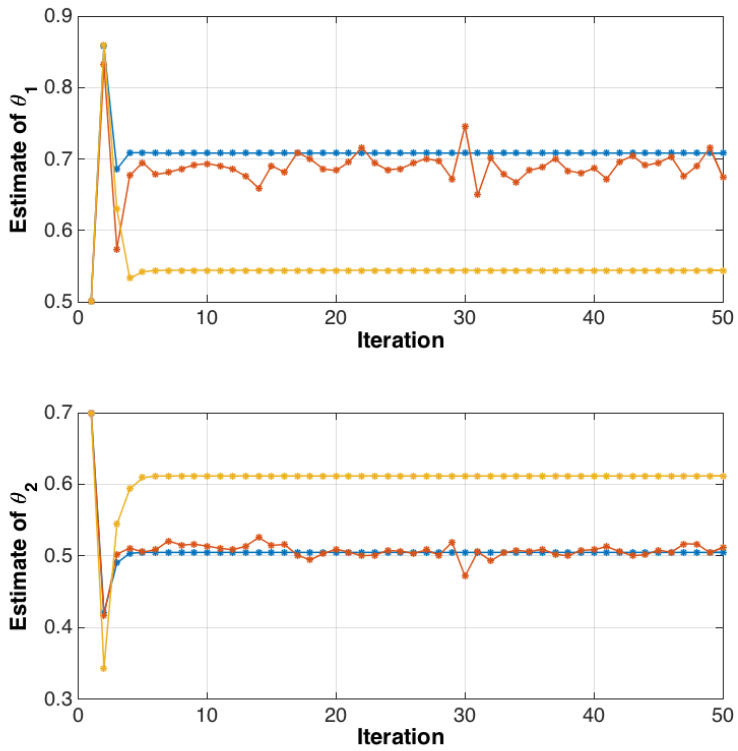


Figure 4.2: State-space model (4.32). Estimates of θ_1, θ_2 (true value [0.7, 0.5]) using the NMCML (orange), the OMCML (yellow), and the PRM (blue), on one data-set. The NMCML shows noisy behaviour but small error; the OMCML is more stable but the error is bigger; the PRM is both stable and accurate.

in terms of accuracy, but convergence is achieved after 25-30 iterations. We argue that the improvement in the convergence rate of the PRM is mainly due to the re-sampling rule which allows the deployment of full Newton's steps at least in the last iterations, when the number of replaced samples is almost zero and the method benefits of the quadratic convergence typical of the deterministic Newton's method.

In Figure 4.5, the final estimates from the 100 data-sets are reported, in order to show bias and variance information. The bias is very small and the variance is comparable to the one from the ALG3FL method, reported in [45].

Finally, the PRM implements the Monte Carlo integration with a sample size $M = 500$. The ALG3FL method is implemented with $M = 2000$. The PRM requires a full sample simulation only at the first iteration, see Figure

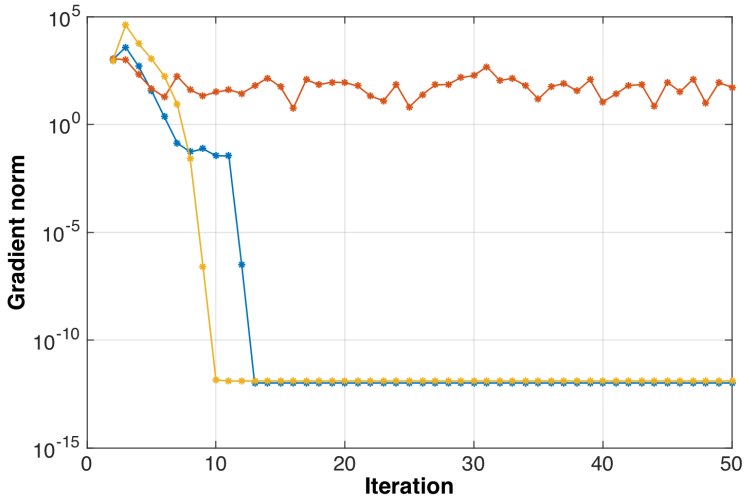


Figure 4.3: State-space model (4.32). Infinity norm of the gradient estimates using the NMCML (orange), the OMCML (yellow), and the PRM (blue), on one data-set. The NMCML gradient is very noisy; the OMCML gradient is stable, but the introduced MC errors drift away its solution; the PRM gradient shows noisy behaviour at the start, but then it stabilizes.

4.4. From the second iteration, the number of samples to be simulated reduces, resulting into an overall reduction of complexity of the method.

4.4 Summary

In this chapter, two methods addressing the finite sample size problem have been presented. When $M < \infty$, two main issues affect the performance of a nested MCML method: increasing Monte Carlo error and noisy search problem.

The first one occurs when the samples simulation is performed only once, from an arbitrary or initial value of the parameter. Since the parameter is updated during the iterative search, the function resulting from the MC estimate is evaluated at values of θ different from the one used for samples simulation. This results into an increase of the MC errors and, hence, of the final parameter estimate. The second issue occurs when the samples simulation is performed at each iteration of the iterative search. In this case, the MC error does not increase, but the function resulting from the MC estimate is altered at each iteration. This leads to a noisy search.

Two new methods have been proposed to address these two issues. The first one is the deterministic method: at each iteration a new sample set is simulated but it is kept fixed during the successive optimization stage. In

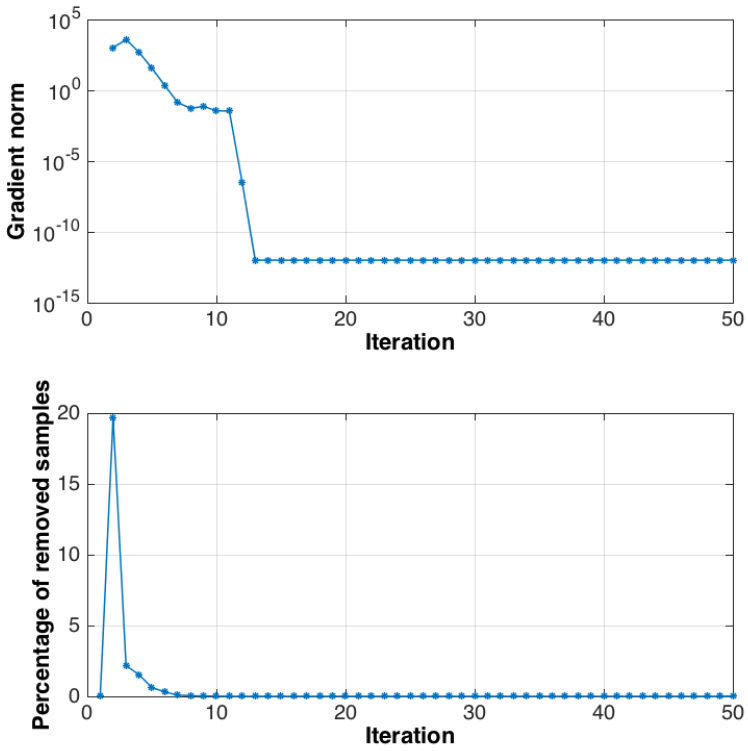


Figure 4.4: State-space model (4.32). Infinity norm of the gradient estimate and number of replaced samples per iteration of the PRM. The behaviour of the gradient is noisy as long as many samples are changed across the iterations. When this number decreases, the PRM shows the quadratic convergence of the Newton's method.

this way, each optimization is deterministic and the search is more stable. However, at each iteration, an optimization problem has to be solved till full convergence. This increases the total number of iterations required for convergence.

The second method is the partial re-sampling method. It is based on the nested MCML method, where a samples simulation is performed at each iteration. However, a sensitivity-based re-sampling rule is derived in order to decide which samples from the current iteration can be reused in future iterations too. In this way, a lower number of new samples is simulated each time, stabilizing the search. Furthermore, the efficient use of the samples, based on their sensitivity w.r.t to the parameter search, reduces the complexity of the samples simulation operation and improves the convergence rate of the method.

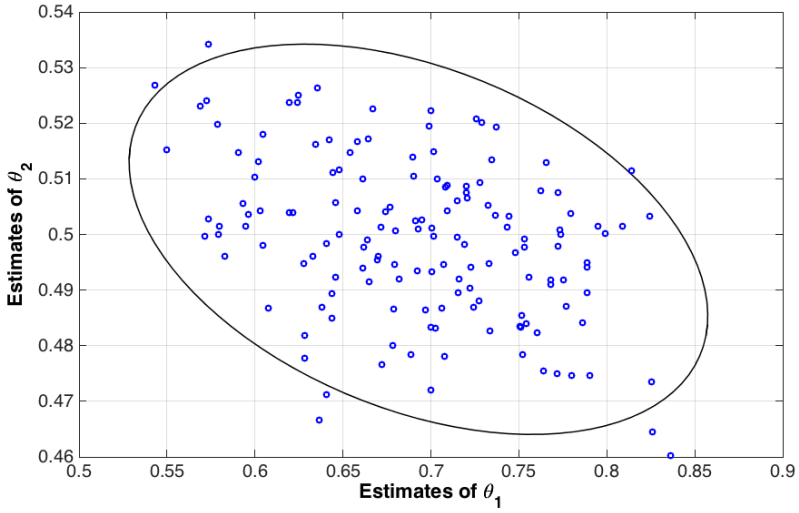


Figure 4.5: State-space model (4.32). Estimates of θ_1, θ_2 (true value [0.7, 0.5]) using the PRM from 100 data-sets. The solid like denotes an empirical covariance ellipse (95% confidence interval).

Part III

Initialization algorithms for Wiener-Hammerstein model identification

Chapter 5

Introduction

All the methods discussed and presented in the previous chapter address the problem of finding the ML estimate in case of intractable stochastic nonlinear models. In this case, the resulting MLE problem can be highly nonlinear and non-convex, and all the methods have proven convergence results only to a local optimizer of the likelihood function. Although the two new methods, proposed in Chapter 4, improve some stability and computational issue related to the ML and EM methods based on Monte Carlo simulations, they cannot solve the global optimization problem. Furthermore, even when the problem is tractable and no Monte Carlo approximations are needed, the resulting ML criterion may still be non-convex, since the underlying model structure is nonlinear. Hence, the search for the global maximum of the likelihood has to be addressed separately.

General algorithms for global convergence make use of random global search strategies, see e.g. [41], [65]. However, these strategies may be extremely computational expensive and time-consuming. Hence, in many cases, it is common to derive *ad hoc* initialization methods with the sole purpose of finding a good initial guess. With good initial guess, we mean an initialization point for the parameter search that increases the chances of converging to a global optimizer, using local exploration methods. This is the central object of this third part of the thesis, where we attempt to derive initialization methods for a specific class of block-oriented models: the Wiener-Hammerstein model. For this class of models, in fact, we will show that linear approximations can be used to retrieve a good initial guess.

In this introductory chapter, we introduce the initialization problem for MLE of WH models and we present the main properties of the linear approximations for this type of models. In the next chapter, we present and discuss two intialization algorithms based on linear approximations.

5.1 The MLE initialization problem for WH models

In this section, we recall the definition of the WH model and discuss the initialization problem for ML estimation. In Section 4.2.1, we already presented the MLE of an example WH model, using the deterministic method. However, we did not address the initialization problem. We start by introducing the stochastic WH model first, i.e. the presence of both measurement and process noise. Then, we focus on the special case where the process noise is not present.

5.1.1 Presence of measurement and process noise

When both measurement and process noise are present, we define the WH model as stochastic WH model, see also Section 2.1.3. The model equations are

$$\begin{aligned} x_t &= G_W(q, \theta_W)u_t + w_t, \\ z_t &= f(x_t, \theta_{NL}), \\ y_t &= G_H(q, \theta_H)z_t + e_t. \end{aligned} \tag{5.1}$$

where w_t and e_t are, respectively, process and measurement noise, independently and identically distributed according to some distributions $p_{\mathbf{e}}(e_t)$ and $p_{\mathbf{w}}(w_t)$. For this model, the MLE is defined as

$$\hat{\theta}_{ML} \coloneqq \arg\max_{\theta} p(\mathbf{y}; \theta), \tag{5.2}$$

where $\mathbf{y} = \{y_t\}_{t=1}^N$ and $p(\mathbf{y}; \theta)$ is the likelihood function, see Definition 2.2.1. Given the presence of the process noise w_t, this likelihood function has to be computed by marginalizing the unknown signal $\mathbf{x} = \{x_t\}_{t=1}^N$ out from the joint probability,

$$p(\mathbf{y}; \theta) = \int_{\mathbb{R}^N} p(\mathbf{x}, \mathbf{y}; \theta) \mathrm{d}\mathbf{x}, \tag{5.3}$$

where the joint probability is factorized as $p(\mathbf{x}, \mathbf{y}; \theta) = p(\mathbf{y}|\mathbf{x}; \theta)p(\mathbf{x}; \theta)$. The input $\mathbf{u} = \{u_t\}_{t=1}^N$ is assumed to be known exactly. Hence, we have that $p(\mathbf{y}|\mathbf{x}; \theta)$ and $p(\mathbf{x}; \theta)$ are direct reflections of, respectively, the measurement and process noise distributions,

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}; \theta) &= p_{\mathbf{e}}(\mathbf{y} - G_H(q, \theta_H)f(\mathbf{x}, \theta_{NL})), \\ p(\mathbf{x}; \theta) &= p_{\mathbf{w}}(\mathbf{x} - G_W(q, \theta_W)\mathbf{u}). \end{aligned} \tag{5.4}$$

Given the presence of the non-invertible nonlinear transformation $f(\cdot; \theta_{NL})$, the integral (5.3) is intractable and one of the Monte Carlo methods described

in Chapter 4 has to be implemented. In particular, we consider the nested MCML method described in Section 4.1. Starting from an initial guess $\theta^{(0)}$, this method implements a local exploration of the likelihood function by making use of Monte Carlo estimates of its log-gradient. In general, because of the presence of the nonlinearity $f(\cdot;\theta_{NL})$ and of the process noise, the likelihood function is nonlinear and non-convex. Thus, $\theta^{(0)} = [\theta_W^{(0)}, \theta_{NL}^{(0)}, \theta_H^{(0)}]$ has to be chosen carefully, in order to avoid local maxima.

Initialization algorithms for this type of model are at their early stage of development. The algorithm presented in this thesis (Section 6.1), based on [66], is the first attempt to combine the ML estimates with the linear approximations of the system. After this, [67] has generalized and extended the framework of the linear approximations to general nonlinear models affected by process noise.

5.1.2 Presence of measurement noise only

When process noise is not present, the only stochastic contribution to the model outputs comes from the measurement noise e_t, distributed according to $p_{\mathbf{e}}(e_t)$. Since the signal $\mathbf{x}$ is not affected by disturbances, the marginalization integral can be easily solved

$$p(\mathbf{y};\theta) = \int_{\mathbb{R}^N} p(\mathbf{x},\mathbf{y};\theta)\delta(\mathbf{x}-\mathbf{x}_0)\mathrm{d}\mathbf{x} = \int_{\mathbb{R}^N} p(\mathbf{y}|\mathbf{x};\theta)p(\mathbf{x};\theta)\delta(\mathbf{x}-\mathbf{x}_0)\mathrm{d}\mathbf{x}, \tag{5.5}$$

where $\delta(\mathbf{x}-\mathbf{x}_0)$ is a Dirac delta function centred in $\mathbf{x}_0$, which is defined as

$$\mathbf{x}_0 = G_W(q,\theta_W)u_t, \tag{5.6}$$

i. e. the actual, undisturbed output of the first linear block. Thus, the probability $p(\mathbf{x};\theta)$ boils down to a Dirac delta function in $\mathbf{x}_0$ and

$$\begin{aligned} p(\mathbf{y};\theta) &= p(\mathbf{y}|\mathbf{x}=\mathbf{x}_0;\theta) \\ &= p_{\mathbf{e}}(\mathbf{y} - G_H(q,\theta_H)f(\mathbf{x}_0,\theta_{NL})) \\ &= p_{\mathbf{e}}(\mathbf{y} - G_H(q,\theta_H)f(G_W(q,\theta_W)\mathbf{u},\theta_{NL})). \end{aligned} \tag{5.7}$$

Hence, the likelihood function $p(\mathbf{y};\theta)$ is a direct reflection of the measurement noise only. Since the measurement noise is independently distributed over t, the likelihood can be expressed as product of the probabilities of the single realizations,

$$p(\mathbf{y};\theta) = \prod_{t=1}^{N} p_{\mathbf{e}}(y_t - G_H(q,\theta_H)f(G_W(q,\theta_W)u_t,\theta_{NL})), \tag{5.8}$$

and the MLE problem can be formulated as the minimization of the negative log-likelihood,

$$\hat{\theta}_{ML} \coloneqq \arg\min_{\theta} \; -\sum_{t=1}^{N} \log \; p_{\mathbf{e}}(y_t - G_H(q,\theta_H) f(G_W(q,\theta_W)u_t, \theta_{NL})). \quad (5.9)$$

Although the MLE problem simplified a lot, compared to stochastic case, the cost function of the minimization problem is still nonlinear and, depending on the shape of the measurement noise distribution and of the nonlinearity, many local minima can be present. Thus, also in this case, the choice of $\theta^{(0)} = [\theta_W^{(0)}, \theta_{NL}^{(0)}, \theta_H^{(0)}]$ is crucial.

A typical assumption for the measurement noise is that it is normally distributed with zero mean and variance σ_e^2,

$$e_t \sim \mathcal{N}(0, \sigma_e^2), \quad \forall t. \quad (5.10)$$

In this case, the MLE problem simplified even further to

$$\hat{\theta}_{ML} \coloneqq \arg\min_{\theta} \; \sum_{t=1}^{N} \frac{1}{2\sigma_e^2} [y_t - G_H(q,\theta_H) f(G_W(q,\theta_W)u_t, \theta_{NL})]^2, \quad (5.11)$$

which is equivalent to a weighted least square error criterion. Furthermore, as already discussed in Section 2.2.2, if we define the predictor of the model output as

$$\hat{y}_t(\theta) = G_H(q,\theta_H) f(G_W(q,\theta_W)u_t, \theta_{NL}), \quad (5.12)$$

then (5.11) is equivalent to the PEM estimator. However, the cost function to minimize is still nonlinear and iterative search algorithms may end up in local minima, see e.g. [68]. Thus, initialization algorithms need to be developed for this special case as well. We will address this problem in Section 6.2.

Unlike the stochastic WH model case, many approaches for initializing the WH model identification problem, when process noise is not present, are available in literature. Many of them are based on linear approximations and, in particular, they rely on the asymptotic results of the Best Linear Approximation (BLA) of a nonlinear system. In the next section, we will formally define the BLA and we will explain how it can be used to initialize the WH model identification problem.

5.2 The Best Linear Approximation

The Best Linear Approximation of a nonlinear system is defined in the following.

Definition 5.2.1. *(The Best Linear Approximation) The BLA of a time-invariant nonlinear system to a given class of stationary input signals $\mathcal{U}$, containing sequences of length N, is defined as the best linear system approximating the system's output in the mean square sense [69], [70],*

$$G_{BLA}(q,\hat{\theta}_N) = \arg\min_{G\in\mathcal{G}} \frac{1}{N}\sum_{t=1}^{N}(y_t - G(q,\theta)u_t)^2, \tag{5.13}$$

where $G(q,\theta)$ is a linear model belonging to the class of linear systems $\mathcal{G}$.

When process noise is not present, it is proved that the BLA of a WH system provides a consistent estimate of the concatenation of the two linear dynamic blocks, when the input belongs to the Riemann equivalence class of asymptotically normally distributed signals [5], [70]. Consider the following WH model with no process noise,

$$\begin{aligned} x_t &= G_W(q,\theta_W)u_t, \\ z_t &= f(x_t,\theta_{NL}), \\ y_t &= G_H(q,\theta_H)z_t + e_t, \end{aligned} \tag{5.14}$$

and assume that real data are generated when the true value of the parameter vector is used, i.e. $\theta_0 = [\theta_W^0, \theta_{NL}^0, \theta_H^0]$. By using Definition 5.2.1, the consistency result from [5] and [70] entails that

$$G_{BLA}(q,\hat{\theta}_N) \to kG_W(q,\theta_G^0)G_H(q,\theta_H^0) \quad \text{w.p. } 1 \quad \text{as} \quad N\to\infty \tag{5.15}$$

where the constant scaling factor k depends on the input amplitude and on the nonlinearity. Clearly, this result provides a strong tool for deriving initialization algorithms for WH model identification. A simple linear least squares problem provides consistent estimates of the concatenation of the linear parameters. Therefore, the problem of retrieving a good approximation of θ_W^0, θ_H^0 is reduced to a partitioning problem, where it is required to correctly divide the dynamics contained in the BLA between the two linear parts. Once this partitioning problem has been solved, θ_W^0, θ_H^0 can be used to define a new MLE problem for identification of the θ_{NL}. This is the main idea at the basis of many WH system identification algorithms, see e.g. [10], [68], [71], [72], [73], [74].

Hence, the question is whether we could use the BLA also in case of stochastic WH model, i.e. when the process noise is present. The answer is positive and it is based on the results originally presented in [66], where we extended the BLA consistency results to the stochastic WH model case. We recall the main consistency result in the following section.

5.2.1 Best Linear Approximations of stochastic WH models

We first recall the stochastic WH model define in Section 2.1.3,

$$\begin{aligned} x_t &= G_W(q, \theta_W)u_t + w_t, \\ z_t &= f(x_t, \theta_{NL}), \\ y_t &= G_H(q, \theta_H)z_t + e_t, \end{aligned} \tag{5.16}$$

and we assume that real data are generated when the true value of the parameter vector is used, i.e. $\theta_0 = [\theta_W^0, \theta_{NL}^0, \theta_H^0]$. Then we have the following theorem.

Theorem 5.2.1. *(Consistency of the BLA of stochastic WH models) If the following assumptions are satisfied,*

1. *The input u_t and the process noise w_t are independent, Gaussian, stationary processes;*

2. *The measurement noise e_t is a stationary stochastic process, independent of u_t and w_t;*

3. *$G(q, \theta)$ is an arbitrary transfer function parametrization with freely adjustable gain, such that $G(q, \theta_0) = G_W(q, \theta_W^0)G_H(q, \theta_H^0)$, for some parameter value θ_0;*

4. *The parameter θ is estimated from $\boldsymbol{u}$ and $\boldsymbol{y}$ using an output error method,*

$$\hat{\theta}_N = \arg\min_{\theta} \frac{1}{N}\sum_{t=1}^{N}(y(t) - G(q, \theta)u(t))^2, \tag{5.17}$$

then

$$G(q, \hat{\theta}_N) \to kG_W(q, \theta_W^0)G_H(q, \theta_H^0) \quad w.p.\ 1 \quad as \quad N \to \infty \tag{5.18}$$

The proof can be found in Appendix 8.5. The static k depends on the variance of the input u_t and the process noise w_t, see [67]. In the next section, we illustrate the consistency results of the BLA on numerical examples. In the next chapter, instead, we will use the BLA to derive an initialization algorithm for stochastic WH model identification.

Table 5.1: BLA estimation - Example 1

Poles/Zeros	True	Estimated ($\mu \pm \sigma$)
α	0.4	0.3988 ± 0.1000
β	0.8	0.7998 ± 0.0401

5.2.2 Numerical examples

To support the theoretical results, we provide here two numerical examples illustrating the consistency of the BLA estimation. Two different stochastic WH systems are simulated to obtain data. The generated data are used to estimate the best linear approximation of the systems. Monte Carlo analysis is used to generate estimation distributions over 1000 sets of 1000 data points for each set.

Example 1: Two first order systems with polynomial nonlinearity

The first WH model is the same model that we used in Section 4.2.1 to illustrate the ML estimation of the parameters of the nonlinearity. In that case, we assumed known linear parts. Now, we estimate the parameters of the linear parts using the BLA. The WH model is

$$x_t = \frac{1}{1-\alpha q^{-1}} u_t + w_t \tag{5.19a}$$
$$z_t = f(x_t, \theta_{NL}) \tag{5.19b}$$
$$y_t = \frac{1}{1-\beta q^{-1}} z_t + e_t. \tag{5.19c}$$

with $f(x_t, \theta_{NL})$ being a third degree polynomial $f(x_t, \theta_{NL}) = c_0 + c_1 x_t + c_2 x_t^2 + c_3 x_t^3$. The process and output noise are respectively white and Gaussian with standard deviations $\sigma_w = 4$, $\sigma_e = 1$. The signals u, w, and e are mutually independent. The BLA estimates are reported in Table 5.1.

Table 5.2: BLA estimation - Example 2

Poles/Zeros	True	Estimated ($\mu \pm \sigma$)
$p_{1,2}$	$0.8 \pm 0.4\mathbf{i}$	$(0.7965 \pm 0.012) \pm (0.3999 \pm 0.0154)\mathbf{i}$
$p_{3,4}$	$0.4 \pm 0.7\mathbf{i}$	$(0.3954 \pm 0.037) \pm (0.6997 \pm 0.0504)\mathbf{i}$
z_1	0.6	0.5714 ± 0.1015

Example 2: Two second order systems with polynomial nonlinearity

The second example is a Wiener-Hammerstein system consisting in two second order linear systems with a polynomial nonlinearity in the middle:

$$\begin{aligned} x_t &= \frac{q - b1}{q^2 + a_1 q + a_2} u_t + w_t \\ z_t &= f(x_t, \theta_{NL}) \\ y_t &= \frac{q}{q^2 + a_3 q + a_4} z_t + e_t \end{aligned} \tag{5.20}$$

where the $f(x_t, \theta_{NL})$ is a third degree polynomial:

$$f(x_t, \theta_{NL}) = c_0 + c_1 x_t + c_2 x_t^2 + c_3 x_t^3 \tag{5.21}$$

Signals u, w and e have same statistical properties as previous example and standard deviations, respectively, 5, 5 and 1. Linear parameters b_1, a_1, a_2, a_3, a_4 correspond, respectively, to a real zero in $z_1 = 0.6$ and two pairs of complex poles in $p_{1,2} = 0.8 \pm 0.4\mathbf{i}$ and $p_{3,4} = 0.4 \pm 0.7\mathbf{i}$. Also in this case the BLA provides good estimates of the linear parameters, see Table 5.2.

5.3 Summary

In this chapter, we introduced the initialization problem of WH models identification. For both the case of presence of process noise (stochastic WH model) and of its absence, the choice of an initial guess for the parameter vector $\theta = [\theta_W, \theta_{NL}, \theta_H]$ is crucial in order to avoid local minima. In fact, in both cases, the MLE problem can be highly nonlinear and local explorations methods need to be implemented.

When the process noise is not present, many approaches for initializing the MLE problem rely on the fact the the Best Linear Approximation of the WH system provides a consistent estimate of the concatenation of θ_W and

θ_H. Hence, the problem of finding the initial guess for these two parameters boils down to a partitioning problem, consisting of deciding how to split the dynamics contained in the BLA between the two linear blocks of the WH model.

In order to use the same idea also in case of stochastic WH model, the consistency result of the BLA has been extended to the general case of presence of process noise. In this way, initialization algorithms solving the partitioning problem can be adapted for this case too. Once the initial estimates for the parameters of the linear blocks are found, the only remaining problem is the initialization and identification of θ_{NL}. We will discuss this point in the next chapter.

Chapter 6

Initialization algorithms

In this chapter, the BLA is used to derive initialization algorithms for WH models. The main idea is to use the consistent estimates of the linear dynamics contained in the BLA, to initialize the two linear blocks of the WH model. The initialization problem becomes actually a partitioning problem of the BLA dynamics between the two linear parts.

A common solution to the partitioning problem is the *exhaustive search approach* (ESA), where all the possible combinations of the dynamics contained in the BLA are tested as initial guess. In this way, the approach is formulated as a discrete optimization problem. The resulting initialization algorithm might be very expensive for high order models, since it shows a combinatorial complexity. However, it shows good performances in obtaining the dynamics of the two linear parts and it is easy to implement. The approach was presented in [74] and it was developed for the identification of WH model structure affected by measurement noise only. In this thesis, we extend and apply the exhaustive search approach to the stochastic WH model case, by combining the initialization algorithm with the new MCML methods developed in Chapter 4.

Next, we focus on the initialization problem of the WH model, when the process noise is not present. In this case, in fact, the MLE results into a simpler optimization problem, see Section 5.1.2, and a more efficient initialization algorithm can be deployed. It is the *fractional approach* (FA) and it performs a relaxation of the discrete optimization problem into a continuous one. The linear dynamics obtained from the BLA are parametrized in a fractional way and only one optimization problem is required to retrieve the partitioning of the dynamics. The FA was originally presented in [75], and its main advantage is that the original discrete partitioning problem is replaced by a single continuous one, resulting in a reduced complexity. In this thesis we show that the FA becomes ill-conditioned for some particular configurations of the linear dynamics, causing identifiability issues. Hence,

we propose a modification of this approach, based on series expansion of the fractional dynamics. This modification shares most of the properties of the fractional approach. Nevertheless, it provides an implicit regularization of the identification problem. It addresses the ill-conditioning problem while preserving meaningful statistical properties of the estimation.

6.1 Initialization algorithm for stochastic WH models

In this section, we first describe the exhaustive search approach for initializing WH models. Then, based on the result about the consistency of the BLA, derived in Section 5.2.1, we combine the exhaustive search approach with the ML estimation algorithms derived in Chapter 4. In particular, we make use of the *partial re-sampling* method, see Section 4.3. Finally we test the initialization algorithm with simulated and benchmark data.

6.1.1 The Exhaustive Search approach

The Exhaustive Search approach [74] was originally developed for initializing WH models with no process noise. In this case, we showed that ML and PEM estimators are equivalent, see Section 5.1.2. Hence, we formulate the identification problem as a PEM problem,

$$\hat{\theta}_{ML} \coloneqq \arg\min_{\theta} \frac{1}{N} \sum_{t=1}^{N} [y_t - G_H(q, \theta_H) f(G_W(q, \theta_W) u_t, \theta_{NL})]^2, \tag{6.1}$$

where $\theta = [\theta_W, \theta_{NL}, \theta_H]$. Given the consistency of the BLA, we can separate the identification of the parameters of the linear and nonlinear parts. Hence, the idea is to split the BLA model into two sub-models in all possible ways, and to initialize a WH model with each of these splits, for estimating the parameters of the nonlinearity. The steps of the algorithm are the presented in the following.

1. Estimate the BLA $G_{BLA}(q, \hat{\theta}_N)$ of the system, see Section 5.2.1.

2. Split the BLA into all possible $G_W(q, \hat{\theta}_W^i)$ and $G_H(q, \hat{\theta}_H^i)$ so that $G_{BLA}(q, \hat{\theta}_N) = G_W(q, \hat{\theta}_W^i) G_H(q, \hat{\theta}_H^i)$, with $i = 1, \ldots, C$ and C is the total number of possible partitions. For this, the poles and the zeros of the BLA need to be calculated. Depending on prior knowledge of the system, e.g. order of one or both linear parts, some split can be avoided.

3. For all partitions, fix the linear parameters in the model's predictor and estimate the parameters of the nonlinearity θ_{NL} using PEM,

$$\hat{\theta}_{NL} \coloneqq \arg\min_{\theta_{NL}} \frac{1}{N} \sum_{t=1}^{N} [y_t - G_H(q, \theta_H^i) f(G_W(q, \theta_W^i) u_t, \theta_{NL})]^2 \quad (6.2)$$

4. Order the partitions with respect to the criterion of fit.

The best partition is selected and the corresponding parameters, defined as $\theta^b = [\theta_W^b, \theta_{NL}^b, \theta_H^b]$, are used to initialize a final parameter estimation. Hence, a final minimization using (6.1), with $\theta^{(0)} = \theta^b$, is performed.

It is possible to prove that the identification algorithm initialized with the exhaustive search is able to provide consistent estimates of the parameters. To illustrate this, we first note that the PEM estimator defined in (6.2) is consistent, since it corresponds to the MLE. Since one of the tested combination corresponds to the right split of the linear dynamics, for that combination the global solution of (6.2) also corresponds to the true value of θ_{NL} and it provides the lowest value of the criterion. Hence, proving consistency boils down to showing that the global solution of (6.2) can be retrieved for each combination. This is clearly true if $f(\cdot, \theta_{NL})$ is linearly parametrized in θ_{NL}. In this case, in fact, the optimization problem (6.2) is a linear least squares problem, with a global, unique solution. Formal proof of the consistency of the algorithm can be found in [74].

6.1.2 Adaptation to the stochastic WH model case

In order to make use of the exhaustive search approach for initializing stochastic WH model structure, we implement one important modification, originally introduced in [58]. The consistency result of the exhaustive search algorithm mainly relies on the consistency of the estimation problem in Step 3. However, when process noise is present, the PEM criterion used in (6.2) is proved to be inconsistent, see Section 4.2.1. The main reason for the inconsistency is that the criterion used in (6.2) does not take into account the stochastic contribution coming from the process noise. Hence, in order to be able to get consistent estimates, we replace the PEM estimator in step 3 with a ML estimator,

$$\hat{\theta}_{NL} \coloneqq \arg\max_{\theta_{NL}} p(\mathbf{y}; \theta_{NL}), \quad (6.3)$$

where $p(\mathbf{y}; \theta_{NL})$ is the likelihood function, derived for stochastic WH models in Section 5.1.1, Equation (5.3). In this case, however, we are only interested in the estimation of the parameter θ_{NL}. In fact, the parameters of the linear

parts are fixed and given by the BLA. Nevertheless, given the presence of the process noise, the likelihood function is still computed via a marginalization operation. Hence, one of the methods developed in Chapter 4 have to be used in this case. In particular, we make use of the partial re-sampling method, presented in 4.3.

With these modifications, the main steps of the exhaustive search approach for stochastic WH model initialization are:

1. Estimate the BLA $G_{BLA}(q, \hat{\theta}_N)$ of the system, see Section 5.2.1.

2. Split the BLA into all possible $G_W(q, \theta_W^i)$ and $G_H(q, \theta_H^i)$ so that $G_{BLA}(q, \hat{\theta}_N) = G_W(q, \theta_W^i)G_H(q, \theta_H^i)$, with $i = 1, \ldots, C$ and C is the total number of possible partitions.

3. For all partitions, fix the linear parameters in the deterministic parts of the measurement and process noise distributions

$$\begin{aligned} p(\mathbf{y}|\mathbf{x};\theta) = p(\mathbf{y}|\mathbf{x};\theta_{NL}) = p_{\mathbf{e}}(\mathbf{y} - G_H(q,\theta_H^i)f(\mathbf{x},\theta_{NL})), \\ p(\mathbf{x};\theta) = p(\mathbf{x}) = p_{\mathbf{w}}(\mathbf{x} - G_W(q,\theta_W^i)\mathbf{u}). \end{aligned} \tag{6.4}$$

   and define ML estimator for the parameters of the nonlinearity θ_{NL} only,

$$\hat{\theta}_{NL} \coloneqq \arg\max_{\theta_{NL}} p(\mathbf{y};\theta_{NL}) = \arg\max_{\theta_{NL}} \int_{\mathbb{R}^N} p(\mathbf{y}|\mathbf{x};\theta_{NL})p(\mathbf{x})\mathrm{d}\mathbf{x} \tag{6.5}$$

4. Solve the ML problem by using the partial re-sampling method of Section 4.3.

5. Order the partitions with respect to the criterion of fit.

Again, the best partition is selected and the corresponding parameters, defined as $\theta^b = [\theta_W^b, \theta_{NL}^b, \theta_H^b]$, are used to initialize a final parameter estimation. This time, the final parameter estimation is the MLE,

$$\hat{\theta}_{ML} \coloneqq \arg\max_{\theta} \int_{\mathbb{R}^N} p(\mathbf{y}|\mathbf{x};\theta)p(\mathbf{x};\theta)\mathrm{d}\mathbf{x}, \tag{6.6}$$

where the linear parameters θ_W, θ_H are not fixed and they can be re-estimated. The partial re-sampling method can be deployed again and the search is initialized with $\theta^{(0)} = \theta^b$.

Also in this case, the identification algorithm based on the exhaustive search is able to provide consistent estimates of the true parameter. However, a formal proof of consistency in this case is harder to derive, since the optimization problem in Step 3 is a nonlinear ML problem. Hence, the

partial re-sampling method, used in Step 4, requires a good initial guess in order to obtain a consistent estimate of the parameters of the nonlinearity. At this point, one may argue that the derived initialization algorithm does not actually solve the global optimization problem of the MLE estimate for stochastic WH model. From a general perspective, this is true. However, this initialization algorithm is a first step towards the derivation of consistent estimates, since it makes use of consistent estimators in all its steps. In fact, if at Step 3 the standard PEM were used, it would be impossible to obtain unbiased estimate of θ_{NL}, even in case of linear parametrization of $f(\cdot, \theta_{NL})$. Hence, we argue that the combination of the exhaustive search and the ML estimates for θ_{NL} is still advantageous in terms of finding a good initial guess, because the ML problem (6.5), given its reduced parameter dimension, is simpler and easier to initialize than the ML problem in all models parameters (6.6). Our statement is also supported by practical numerical examples, where the derived initialization algorithm has been used to retrieve the right split of the dynamics of the BLA, see next section.

6.1.3 Numerical examples

In this section, we illustrate the effectiveness of the initialization algorithm combining the exhaustive search and the ML estimates on two numerical examples.

Simple stochastic WH model

The first example is the stochastic WH model used in Section 4.2.1 and Section 5.2.2,

$$x_t = \frac{1}{1 - \alpha q^{-1}} u_t + w_t, \tag{6.7a}$$

$$z_t = f(x_t, \theta_{NL}), \tag{6.7b}$$

$$y_t = \frac{1}{1 - \beta q^{-1}} z_t + e_t, \tag{6.7c}$$

with $f(x_t, \theta_{NL}) = c_0 + c_1 x_t + c_2 x_t^2 + c_3 x_t^3$. The same system, in fact, has been used to test the consistency of the ML estimates of the nonlinear parameters using the deterministic method (Section 4.2.1), and the consistency of the BLA (Section 5.2.2). Thus, we have all the ingredients to implement the initialization algorithm described in 6.1.2.

When estimating the nonlinear parameters only (Section 4.2.1), we assumed that the parameters α, β of the linear parts were known and fixed during the estimation, and θ_{NL} was the only unknown parameter vector. Here, instead, we use the estimated values from the BLA to initialize α, β.

Table 6.1: Example 1 - The BLA estimation of a stochastic WH model provides unbiased estimates of the linear dynamics.

Poles/Zeros	True	Estimated ($\mu \pm \sigma$)
α	0.4	0.3988 ± 0.1000
β	0.8	0.7998 ± 0.0401

Table 6.2: Example 1 - BLA partitioning - For both combinations of the linear parts, the nonlineraty of the WH model is estimated with ML and PEM. The RMSE is used to decide a partitioning of the linear parts. The ML manages to find the right split. The PEM, inconsistent, provides a lower RMSE for the wrong split.

	ML	PEM
True split	**1.6934**	8.1125
Wrong split	4.3872	**8.0762**

These values are reported in Table 6.1. Since the data are generated from a simulated system, we actually know the true values of the linear parameters. Hence, in order to test the initialization algorithm, we assume we do not know which of the estimated poles from the BLA corresponds to α and which to β, and we deploy the exhaustive search algorithm. Since there are only two poles, only two combinations are possible. For each combination, we estimate θ_{NL} using the partial re-sampling method implementing the MLE. We use $\theta_{NL}^{(0)} = [0, 1, 0, 0]$ as initial guess for θ_{NL}. We also compare this estimation with a standard PEM estimator. In Table 6.2, we report the RMS of the data-fitting errors obtained when using the nonlinear parameters estimated with ML and with the PEM, for the two possible splits of the BLA. We observe that the ML estimate finds the right split, while the PEM, inconsistent for the process noise case, shows lower value of the criterion for the wrong split.

Benchmark example

In this second example, we test the derived algorithm for BLA partitioning on the benchmark WH system introduced in Section 2.1.4. The available data from the system affected by process and measurement noise are used. The

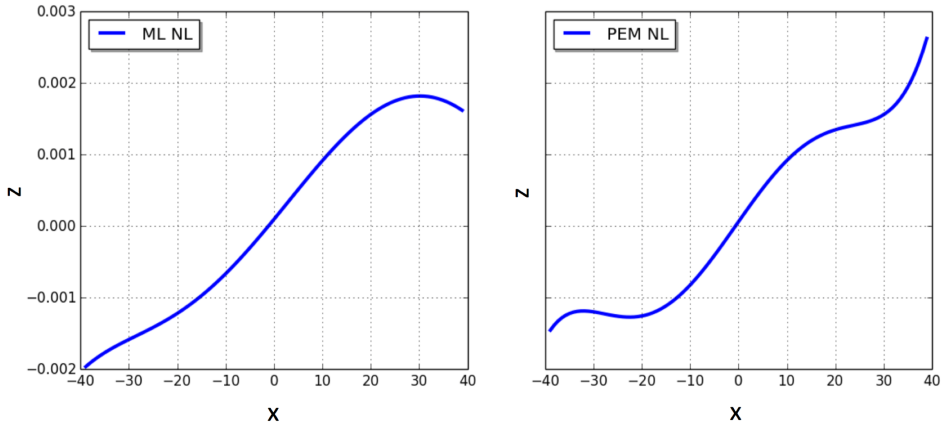


Figure 6.1: Estimation of the WH nonlinearity using ML (partial re-sampling method) and PEM. The ML estimate tries to capture the true saturation behaviour of the true nonlinearity.

BLA is estimated by fitting a 6-th order OE model, with 10000 input/output data. The input used for estimation is the *Multisine*, provided by the benchmark. A set of data not affected by process noise is also available for validation. The BLA provides a RMSE of 35 mV on the validation data. The linear model is then split into two sub-models in all possible ways, but avoiding the combination providing improper transfer functions. For each split, a 5-th order polynomial function is estimated as static non-linearity, via the partial re-sampling method. In this case, 3000 data and a sample size of M = 300 are used. The RMSE achieved by the best split, i.e. lowest value of the ML criterion, is 16.2 mV. This result is comparable with other recently developed methods for stochastic WH model identification, see [59], [76].

The identification of the nonlinearity, providing the best split, is reported in Figure 6.1, and it is compared with the one estimated using PEM. From prior knowledge of the WH benchmark system, we know that the nonlinearity is a diode-resistor network implementing a saturation effect. The ML estimate tries to capture this behaviour.

Finally, in Figure 6.2, the estimated model output, using ML and PEM, is reported. The estimated output is compared with the validation data set, i.e. a process noise-free data. Also in this case, the ML estimate shows better results.

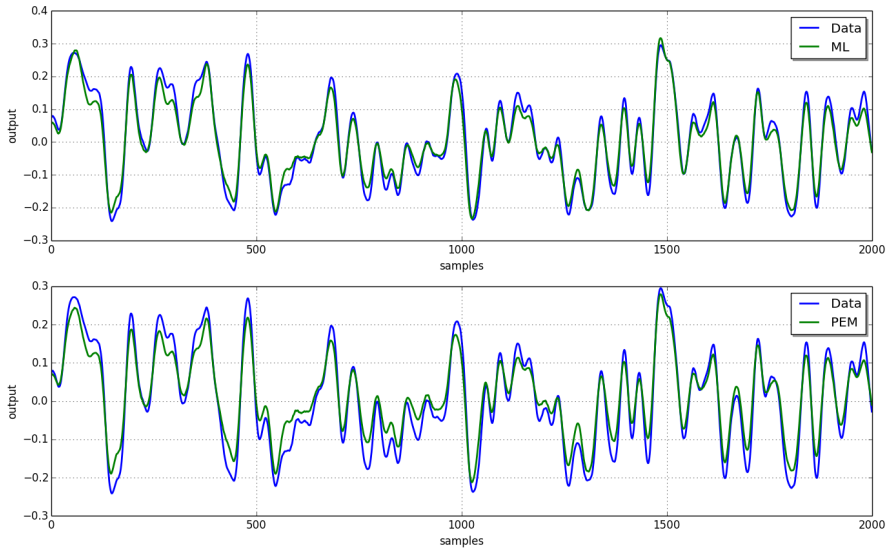


Figure 6.2: ML and PEM estimated outputs for the corresponding identified split and nonlinearity. Validation on a process noise-free data-set. The ML estimate shows better fitting results.

6.2 Initialization algorithm for WH models

In this final part of the thesis, we focus on the case of WH models whose outputs are only affected by measurement noise. For this type of models we showed that the MLE is actually equivalent to the PEM estimator, see Section 5.1.2. Hence, consistent estimates of the parameters can be obtained by implementing a nonlinear least squares problem. Compared to the stochastic WH model case, where approximate solutions based on Monte Carlo estimates are required for solving the MLE problem, the overall difficulty of solving the estimation problem is considerably reduced and no approximate solutions are required. This also affects the design and implementation of an initialization algorithm. In this case, in fact, a more efficient approach can be used, i.e. the *fractional approach* (FA), originally presented in [75]. With this approach, the discrete optimization problem with combinatorial complexity, implemented by the exhaustive search approach, is replaced by only one optimization problem in a new set of variables. In fact, the FA uses the BLA to initialize both linear parts of the WH model and integer exponents in the set $\{0, 1\}$ are introduced for every pole/zero present in the BLA. In this way, it is possible to describe the partition between the two linear parts. A relaxation of the set $\{0, 1\}$ to the continuous interval $[0, 1]$ is then performed. The resulting real-valued exponents can be

identified via a continuous optimization problem. When all the identified real-valued exponents are close to their integer values $\{0, 1\}$, a partition of the pole or the zero can be decided. Iterative methods can be used to solve the continuous problem efficiently, in the considered noise framework.

However, the FA shows some identifiability and conditioning issues, arising from the continuous relaxation of the discrete problem. In the following sections, we present in details the FA and we analyse its identifiability issues. Then, we present a modification of the FA to solve these issues. The relaxation of the set of the integer exponents yields fractional dynamics. The proposed modification treats these fractional dynamics via series expansion. The expansion method naturally introduces a form of regularization in the estimation problem, which alleviates its potential ill-conditioning. Moreover, since no artificial regularization is introduced, the identification problem retains a meaningful description of the (local) statistical properties of the estimation. Finally, a novel formulation of the identification problem based on lifting techniques [77] is proposed, yielding advantageous properties in the resulting continuous optimization algorithm, which allow for a faster and more reliable convergence to the solution when using Newton-type methods. The results presented in the following are mainly based on [78].

6.2.1 The Fractional Approach

To present the fractional approach, we recall the parametrization of the linear parts of a WH model, introduced in Section 2.1.3,

$$G_W(q, \theta_W) = \frac{\sum_{k=0}^{n_B^W} b_k^W q^{-k}}{1 + \sum_{k=1}^{n_A^W} a_k^W q^{-k}}, \tag{6.8}$$

$$G_H(q, \theta_H) = \frac{\sum_{k=0}^{n_B^H} b_k^H q^{-k}}{1 + \sum_{k=1}^{n_A^H} a_k^H q^{-k}}, \tag{6.9}$$

and we assume that the BLA of the WH system has been identified and expressed in terms of poles (p) and zeros (z) factorization,

$$G_{BLA}(q, p, z) = k \frac{\prod_{i=1}^{n_B}(1 - z_i q^{-1})}{\prod_{i=1}^{n_A}(1 - p_i q^{-1})}, \tag{6.10}$$

with $z = [z_1, \ldots, z_{n_B}]$, $p = [p_1, \ldots, p_{n_A}]$, and n_A, n_B being the total number of poles and zeros of the system.

With the FA, the poles and zeros of the BLA are used to initialize two new linear blocks $\hat{G}_W$ and $\hat{G}_H$, where the partition of poles and zeros is parametrized through a new vector of real parameters $[\alpha, \beta]$ in the following

way,

$$\hat{G}_W(q,\alpha,\beta) = \frac{\prod_{i=1}^{n_B}(1-z_i q^{-1})^{\beta_i}}{\prod_{i=1}^{n_A}(1-p_i q^{-1})^{\alpha_i}}, \tag{6.11a}$$

$$\hat{G}_H(q,\alpha,\beta) = \frac{\prod_{i=1}^{n_B}(1-z_i q^{-1})^{1-\beta_i}}{\prod_{i=1}^{n_A}(1-p_i q^{-1})^{1-\alpha_i}}, \tag{6.11b}$$

The indices n_A and n_B also denote the dimension of vectors α and β,

$$\alpha = [\alpha_1, \cdots, \alpha_{n_A}], \quad \beta = [\beta_1, \cdots, \beta_{n_B}]. \tag{6.12}$$

For a real pole (zero) one α_i (β_i) is introduced, and complex pairs of poles/zeros share the same parameter, so that they are kept together during the estimation, see [75]. As a result, $\alpha_i = 1 \quad (\beta_i = 1)$ locates the *i-th* pole or pair of complex poles (zero or pair of complex zeros) at G_W, while $\alpha_i = 0 \quad (\beta_i = 0)$ locates the *i-th* pole or pair of complex poles (zero or pair of complex zeros) at G_H. The estimation of α and β provides the splitting of the dynamics and, hence, the initial guesses $\theta_W^{(0)}$, $\theta_H^{(0)}$. Thus, by using the PEM criterion, consistent for the considered noise framework, the initialization problem consits of finding $\hat{\theta}_N = [\hat{\alpha}, \hat{\beta}, \hat{\theta}_{NL}]$ minimizing

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t(\alpha,\beta,\theta_{NL}))^2. \tag{6.13}$$

The minimum of the criterion (6.13) is obtained at the true value $\theta_0 = [\alpha^0, \beta^0, \theta_{NL}^0]$, which represents the parameter vector describing the true splitting and the true non-linear function. Once the splitting of the dynamics is retrieved, a final optimization can be performed over all parameters, i.e. poles, zeros, and non-linearity. In this way, possible estimation errors from the BLA can be reduced, see [75]. In the next section, we show that the problem of minimizing (6.13) is ill-conditioned for certain configurations of poles and zeros.

6.2.2 Conditioning problem of the fractional approach

In this section, we show that the conditioning problem of the FA derives from a lack of identifiability of the α, β parameters, when particular pole/zero configurations occur in the BLA. Since in the considered noise framework the PEM criterion (6.13) is also the Maximum Likelihood criterion, see Section 2.2.2, the *Cramér-Rao* lower bound for the covariance C of the estimate $\hat{\theta}_N$ is

$$C(\hat{\theta}_N) \geq -M(\theta_0)^{-1}, \tag{6.14}$$

where M is the *Fisher Information Matrix* (FIM), defined by

$$M(\theta_0) = \mathbb{E}\left[\nabla^2_\theta[V_N(\theta)]\Big|_{\theta=\theta_0}\right]. \tag{6.15}$$

The argument of the expectation operator is the Hessian of the criterion,

$$\begin{aligned}\nabla^2_\theta[V_N(\theta)] = \sum_{t=1}^{N}\left(\frac{\mathrm{d}\hat{y}_t(\theta)}{\mathrm{d}\theta}\right)^T \frac{\mathrm{d}\hat{y}_t(\theta)}{\mathrm{d}\theta} \\ + \sum_{t=1}^{N}\frac{\mathrm{d}^2\hat{y}_t(\theta)}{\mathrm{d}\theta^2}(\hat{y}_t(\theta)-y_t).\end{aligned} \tag{6.16}$$

At the true solution θ_0, the residuals $\hat{y}_t(\theta_0) - y_t$ behave as random numbers distributed according the noise distribution $p_{\mathbf{e}}(e_t)$. As discussed before, this noise is the measurement noise for which the common assumption is the normal distribution $\mathcal{N}(0, \sigma_e^2)$. However, for our purposes it is enough that the distribution is zero mean. In this case, in fact, the residuals $\hat{y}_t(\theta_0) - y_t$ are zero mean and the FIM becomes

$$M(\theta_0) = \sum_{t=1}^{N}\left(\frac{\mathrm{d}\hat{y}_t(\theta)}{\mathrm{d}\theta}\right)^T \frac{\mathrm{d}\hat{y}_t(\theta)}{\mathrm{d}\theta}\Big|_{\theta=\theta_0}. \tag{6.17}$$

Thus, the FIM can be ill-conditioned if there is linear dependence among the elements of the sensitivity of model output $\frac{\mathrm{d}\hat{y}_t(\theta)}{\mathrm{d}\theta}$, for example if

$$\frac{\mathrm{d}\hat{y}_t(\theta)}{\mathrm{d}\theta_i} + \mu\frac{\mathrm{d}\hat{y}_t(\theta)}{\mathrm{d}\theta_j} = 0, \quad \forall t, \tag{6.18}$$

for some constant μ and $i \neq j$. In the following we prove that the matrix $M(\theta)$, defined in (6.19), can have an arbitrarily bad conditioning (or even be rank-deficient) for some non-trivial configurations of pole-zeros in the linear subsystems, resulting from some elements $\frac{\mathrm{d}\hat{y}_t(\theta)}{\mathrm{d}\theta_i}$ and $\frac{\mathrm{d}\hat{y}_t(\theta)}{\mathrm{d}\theta_j}$ being nearly linearly dependent.

Theorem 6.2.1. *Assume the fractional parametrization of the Wiener-Hammerstein dynamics as in (6.11) and assume a static non-linearity $f(x_t, \theta_{NL})$ in the middle. Consider the following cases:*

1. *One pole p_i (zero z_i) from G_W is in the neighbourhood of a zero z_j (pole p_j) from G_H;*
2. *One pole p_i (zero z_i) from G_W or G_H is in the neighbourhood of another pole p_j (zero z_j) from G_W or G_H.*

Let λ_{min} be the smallest eigenvalue of the matrix

$$M(\theta) = \sum_{t=1}^{N} \left(\frac{d\hat{y}_t(\theta)}{d\theta} \right)^T \frac{d\hat{y}_t(\theta)}{d\theta}, \tag{6.19}$$

then $0 \leq \lambda_{min} \leq |\Gamma(\theta)\Delta|^2, \quad \forall\theta,$

where Δ is the distance between the pole (zero) and the zero (pole) of case (1) or the distance between the two poles (zeros) of case (2). The function $\Gamma(\theta)$ is defined in Appendix 8.6.

Corollary 6.2.1. *If $\Delta \equiv 0$, then Equation (6.18) holds with $\mu = 1$ for Case 1 and $\mu = -1$ for Case 2, and with θ_i, θ_j being the α and/or β parameters corresponding to poles and/or zeros considered in Case 1 and Case 2 of Theorem 6.2.1.*

The proof of Theorem 6.2.1 and Corollary 6.2.1 can be found in Appendix 8.6. The conditioning problem, described by Theorem 6.2.1, can cause both numerical and identifiability issues. This is explained by the following remark and corollary.

Remark 6.2.1. *Matrix (6.19) also defines the Gauss-Newton approximation of the Hessian for least-squares optimization problems. Hence, the bad conditioning of this matrix can cause numerical issues when the Gauss-Newton approximation is used in the iterative optimization algorithm.*

Corollary 6.2.2. *The parameters α and/or β corresponding to poles and/or zeros considered in Case 1 and Case 2 of Theorem 6.2.1 are not identifiable using the fractional approach.*

Proof. The identifiability of the parameters can be assessed by checking the FIM for singularities. The FIM is the matrix (6.19) evaluated at θ_0. Thus the conditioning results of Theorem 6.2.1 extend to the FIM, causing the identifiability issue.

The conditioning problem generated by Case 2 of Theorem 6.2.1 can be intuitively understood by considering the fractional parametrization (6.11). Indeed, the fractional exponents allow for changing the multiplicity of a pole (zero) arbitrarily. Thus, when two poles (two zeros) are very close to each other, or in case of double pole (double zero), having multiplicity 1 for both of them is equivalent to increase the multiplicity of one while decreasing the multiplicity of the other. On the other hand, Case 1 is less intuitive and more challenging from the identification perspective.

In the next section, we develop and argue for a solution to the identifiability issue generated from Case 1. For the sake of completeness and

comparison, however, we also consider two standard tricks addressing the conditioning problem. The first one is to add an explicit regularizing function to the cost function (6.13). The purpose of the regularization is to accentuate the curvature around the possible minima of the cost function, which correspond to the integer values of α, β and for which Theorem 6.2.1 shows conditioning problems. The second one is to add the inequality constraints $0 \leq \theta_i, \theta_j \leq 1$. In this way, the flat directions of the cost function, indicated by Corollary 6.2.1, would intersect the constrained variable space at a single point, i.e. one of the corner of the constrained box, which would also be the solution of the optimization problem. This is what is also done in [75], but without any further explanation regarding the need of inequality constraints.

Both explicit regularization and inequality constraints, while fixing the conditioning problem, artificially alter the covariance properties of the parameters at the solution. Furthermore the explicit regularization requires some user-tuning of the regularizing function. Our solution is represented by the Expanded Fractional Approach (EFA) which, for Case 1, provides an *implicit* and natural regularization effect for the FA conditioning problem. The EFA and its properties are presented in the next sections.

6.2.3 The Expanded Fractional Approach

In this section, the novel expansion approach is presented and the initialization problem using the Expanded Fractional Approach (EFA) is introduced. The implicit regularization effect of the EFA is finally established.

The expansion idea

In order to get a formulation of the fractional approach which makes use of integer values of the delay operator q, a series expansion is made separately for the numerator and denominator of $\hat{G}_W$ and $\hat{G}_H$, see Expressions (6.11). Numerator and denominator are expanded separately in order to fulfil the following property.

Property 6.2.1. *At the integer values* $\{0, 1\}$ *of the parameters* α *and* β*, the series expansions of numerator and denominator of* $\hat{G}_W$ *and* $\hat{G}_H$ *(6.11) are exact.*

This property does not hold in the original expansion approach presented in [79], where numerator and denominator are expanded together. Property 6.2.1 is important because it allows the model derived from the expanded formulation to recover the original parametrization (2.11)-(2.12) for integer values of α and β. Numerator and denominator, in their fractional form (6.11), are analytic in the complex variable q everywhere except in 0, and

can therefore be expanded as an ordinary Laurent power series about 0, in the variable q^{-1}, see [80]. For example, the n^{th}-order expansion of a single pole p is given by:

$$(1 - pq^{-1})^{\alpha} \approx 1 + A_1(p, \alpha)q^{-1} + \ldots + A_n(p, \alpha)q^{-n}, \tag{6.20}$$

where α is a scalar, n is the expansion order, and the coefficients $A_1(p, \alpha)$, ..., $A_n(p, \alpha)$ are uniquely determined and equal to the Taylor coefficients of the expansion of the function $f(x) = (1 - px)^{\alpha}$, where $x = q^{-1}$, see [81]. Thus, the functions $\hat{G}_W$ and $\hat{G}_H$ can be approximated by $\hat{G}_W^{EXP}$ and $\hat{G}_H^{EXP}$, defined as

$$\hat{G}_W^{EXP}(q, \alpha, \beta) = \frac{1 + \sum_{l=1}^{n_2^W} B_l^W(z, \beta)q^{-l}}{1 + \sum_{l=1}^{n_1^W} A_l^W(p, \alpha)q^{-l}}, \tag{6.21a}$$

$$\hat{G}_H^{EXP}(q, \alpha, \beta) = \frac{1 + \sum_{l=1}^{n_2^H} B_l^H(z, \beta)q^{-l}}{1 + \sum_{l=1}^{n_1^H} A_l^H(p, \alpha)q^{-l}}, \tag{6.21b}$$

where the indices $n_1^W, n_2^W, n_1^H, n_2^H$ are the expansion orders for denominators and numerators, and

$$\begin{aligned} A^W(\alpha) &= [A_1^W(p, \alpha), \ldots, A_{n_1^W}^W(p, \alpha)], \\ B^W(\beta) &= [B_1^W(z, \beta), \ldots, B_{n_2^W}^W(z, \beta)], \\ A^H(\alpha) &= [A_1^H(p, \alpha), \ldots, A_{n_1^H}^H(p, \alpha)], \\ B^H(\beta) &= [B_1^H(z, \beta), \ldots, B_{n_2^H}^H(z, \beta)] \end{aligned} \tag{6.22}$$

are the coefficients provided by the Taylor expansion, depending on the vectors α, β, p, z, see Expression (6.11). The expansion orders $n_1^W, n_2^W, n_1^H, n_2^H$ have to be chosen high enough in order to reproduce all possible pole/zero combinations of the BLA, see Property 6.2.1 . This sets a lower bound for the orders. While one may consider a high order desirable, in order to obtain high accuracy of the expansion, in the next section we show that choosing an order higher than the lower bound can be counter-productive.

Conditioning property of the expanded approach

In this section, it is shown that a low-order expansion of the problematic pole/zero pair described in Case 1 of Theorem 6.2.1 results in rectifying the ill-conditioning problem. This observation is formally stated in the following theorem.

Theorem 6.2.2. *Consider the fractional parametrization as in (6.11) and Case 1 of Theorem 6.2.1. If the two factors containing the zero and the pole*

close to each other are expanded with a first order series expansion, then the matrix $M(\theta)$ is well-conditioned, independently of Δ.

Proof. See Appendix 8.7.

Thus, a low-order expansion of the problematic pole/zero pair addresses the ill-conditioning of the FA problem, providing a well-conditioned Fisher Information Matrix. In general, it has been observed that the EFA, i.e. the expansions (6.21), preserves the result of Theorem 6.2.2 if sufficiently low expansion orders are used. In order to fulfil Property 6.2.1, the lowest possible orders are given by the total number of zeros (n_B) for the expansion of the numerator, and total number of poles (n_A) for the expansion of the denominator. This becomes a general guideline for choosing $n_1^W, n_2^W, n_1^H, n_2^H$. Finally, the EFA introduces algebraic expressions in α, β of increasing complexity in the expansion orders, see (6.22). However, a particular structure resides in these algebraic expressions. By using this structure, a reformulation of the optimization problem allows to improve the algorithmic performance of the optimization problem. We detail these observations next.

6.2.4 Properties of the Expanded Fractional Approach

In this section, firstly we show that a particular structure relates the variables α and β to the coefficients of the expansions in (6.21). A structure that we label as *pseudo-linearity* property. Then, a reformulation of the initialization problem is presented, which makes use of this property. Finally, convergence aspects of Newton-type methods applied to the new formulation are discussed.

The pseudo-linearity property

The pseudo-linearity property is defined in the following.

Definition 6.2.1. *(Pseudo-linearity). A function $g(x) = [g_1(x) \cdots g_N(x)]^T$ is pseudo-linear in $x \in \mathbb{R}^n$ if, $\forall k$, $g_k(x) = A_k x + b_k$, $\forall x \in S_{k-1}$ (for some constant matrices A_k and vectors b_k), where $S_{k-1} = \{x | g_i(x) = 0, i = 1, ..., k-1\}$.*

The following lemma provides results of uniqueness of the solution for a pseudo-linear system of equations.

Lemma 6.2.1. *Consider the system of M equations $g(x) = \bar{g}$, with $g(x)$ being pseudo-linear, $\bar{g}$ given, and $x \in \mathbb{R}^n$. The system admits a unique solution if $M = n$.*

The proof can be found in Appendix 8.8. A useful remark related to the computation of the solution of a pseudo-linear equations system follows next.

Remark 6.2.2. *The proof of Lemma 6.2.1 also shows that the solution to a pseudo-linear system of equations can be analytically computed by substitution, solving one equation at the time, from $k = 1$ to M.*

We show next that the Taylor coefficients (6.22) are pseudo-linear functions of the parameters α and β. For simplicity, we consider the generic fractional function $G(\eta, x)$, representing either the numerator ($\eta \equiv \beta$) or the denominator ($\eta \equiv \alpha$) of (6.21), and where the variable change $x = q^{-1}$ is made. Furthermore, only real zeros (poles) are considered, but all the results can be generalized to complex pairs of zeros and poles, by introducing the same exponent for both the complex zero (pole) and its conjugate. Based on these considerations, we introduce the following theorem.

Theorem 6.2.3. *Consider a generic fractional function G in the form*

$$G(\eta, x) = \prod_{i=1}^{n} (1 + a_i x)^{\eta_i}, \tag{6.23}$$

where $\eta = [\eta_1, \cdots, \eta_n]$ are real exponents, and the k-th order partial derivative of $G(\eta, x)$ w.r.t x is defined as

$$G^{(k)}(\eta, x) = \frac{\partial^k}{\partial x^k} G(\eta, x). \tag{6.24}$$

The generic k-th coefficient of the Taylor expansion of $G(\eta, x)$ w.r.t x and about 0 is

$$A_k(\eta) = \frac{1}{k!} G^{(k)}(\eta, x)\big|_{x=0}. \tag{6.25}$$

Then, the function $A(\eta)$ defined as

$$A(\eta) = \begin{bmatrix} A_1(\eta) \\ \vdots \\ A_M(\eta) \end{bmatrix}, \tag{6.26}$$

where M is the finite order of the Taylor expansion of $G(\eta, x)$ about 0, is pseudo-linear in η.

Proof. See Appendix 8.9.

Corollary 6.2.3. *Consider the function $A(\eta)$, given by the Taylor coefficients of the finite series expansion of a fractional function in the form (6.23), with expansion order M. Then the system of equations*

$$A(\eta) = \bar{A}, \tag{6.27}$$

with $\bar{A}$ being a given vector in $\mathbb{R}^M$, admits a unique solution if $M = n$, where n is the dimension of the variables vector η.

Proof. It directly follows from Lemma 6.2.1 and Theorem 6.2.3.

Theorem 6.2.3 and Corollary 6.2.3, applied to the EFA problem, entail that it is possible to build a pseudo-linear system of equations admitting a unique solution in the α, β variables. Moreover, the proof of Theorem 6.2.3 also provides a computationally efficient procedure to build the pseudo-linear functions, i.e. the recursive law (8.64) in Appendix 8.9. Those results can be used to reformulate the EFA initialization problem, by introducing a set of equality constraints for the Taylor coefficients. This is presented in the next section.

Reformulation of the EFA

With the EFA, the initialization problem is the identification of α, β, which now appear in the coefficients of the expansions of the BLA numerator and denominator, and θ_{NL}, the parameters of the non-linearity, see (6.21). In order to exploit the pseudo-linear structure of the Taylor coefficients, a *lifted* formulation of the optimization problem is proposed. With the lifting procedure, intermediate optimization variables and suitable constraints, which ensure the equivalence with the original problem, are introduced. In general, this procedure offers advantages in terms of convergence rates and region of attraction [77]. In our case, the proposed lifting procedure is the following:

- the coefficients $A^W(\alpha)$, $B^W(\beta)$, $A^H(\alpha)$, $B^H(\beta)$, see (6.22), are re-parametrized by introducing the α-, and β-independent quantities A^W, B^W, A^H, B^H, in the functions (6.21);
- a set of equality constraints is introduced in order to force $A^W(\alpha)$, $B^W(\beta)$, $A^H(\alpha)$, $B^H(\beta)$ to match the newly introduced quantities A^W, B^W, A^H, B^H.

Thus, the lifted initialization problem is

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t(A^W, B^W, A^H, B^H, \theta_{NL}))^2 \\ \text{s.t.} \quad & A^W(\alpha) = A^W \in \mathbb{R}^{n_1^W} \\ & B^W(\beta) = B^W \in \mathbb{R}^{n_2^W} \\ & A^H(\alpha) = A^H \in \mathbb{R}^{n_1^H} \\ & B^H(\beta) = B^H \in \mathbb{R}^{n_2^H} \end{aligned} \tag{6.28}$$

where the vector θ contains both the parameters and the newly introduced optimization variables, i.e. $\theta = [\alpha,\, \beta,\, \theta_{NL},\, A^W,\, B^W,\, A^H,\, B^H\,]$.

The set of equality constraints in (6.28) is, thus, a set of pseudo-linear equations which, according to Corollary 6.2.3, admits a unique solution in α, β if the number of equations equals the number of variables α, β. Hence, the expansion orders ought to match the number of poles and zeros. For comparison, we will refer with *lifted* formulation to problem (6.28), while with *unlifted* formulation we will denote the unconstrained version of (6.28), where the terms $A^W(\alpha)$, $B^W(\beta)$, $A^H(\alpha)$, $B^H(\beta)$ are directly substituted in the model predictor $\hat{y}_t$. Finally, while Remark 6.2.2 entails that it is possible to compute the solution of the system of equality constraints analytically, numerical approaches are far superior for high order systems. This aspect is addressed in the next section, where we show that a Newton-based method converges to the analytical solution of the system of equations.

6.2.5 Convergence of the Newton's method

In this section, we show that the Newton's method applied to the set of equations (6.27) converges to the analytic solution of the system in a number of full steps equal to the number of equations to solve. To do so, we will make use of the pseudo-linearity definition, together with the Newton's step update equation. First we introduce two useful lemmas.

Lemma 6.2.2. *If $g(x)$ is pseudo-linear, then every set S_k is an affine hyperplane.*

Proof. By using induction we have that for $k = 1$, $S_0 \equiv x \in \mathbb{R}^N$. Since the hypothesis is that $g(x)$ is pseudo-linear, then $g_1(x) = A_1 x + b_1$. Therefore S_1 will be the set of x solving an affine system, hence it is an affine hyperplane. Assume now that S_k is a hyperplane and consider the set $\gamma_{k+1} = \{x | A_{k+1} x + b_{k+1} = 0\}$, which is an affine hyperplane. Then S_{k+1} would be the intersection of the two affine hyperplanes S_k and γ_{k+1}, hence it is an affine hyperplane. □

Lemma 6.2.3. *If $S = \{x | g(x) = 0\}$ is a hyperplane, then $\forall \bar{x}$ s.t. $g(\bar{x}) = 0$, the set $T = \{x | \frac{\partial g(\bar{x})}{\partial \bar{x}}(x - \bar{x}) = 0\}$ coincides with S.*

Proof. At any point $\bar{x} \in S$, the tangent set, i.e. $T_S(\bar{x})$ is equivalent to S,

$$T_S(\bar{x}) = \left\{ x | \frac{\partial g(\bar{x})}{\partial \bar{x}}(x - \bar{x}) = 0 \right\} = S. \tag{6.29}$$

□

We can now establish the following proposition, about the Newton convergence.

Lemma 6.2.4. *Let $g(x) \in \mathbb{R}^k$ be pseudo-linear. The solution to the system $g(x) = 0$ is obtained after k full Newton steps.*

Proof. By induction, if $k = 1$, $g_1(x)$ is linear and a full Newton's step provides an update $x^+ \in S_1$. Assume now a generic k and $\bar{x} \in S_{k-1}$. We have to prove that, after k Newton steps, $x^+ \in S_k$. From full Newton's step update we have that

$$\frac{\partial g(\bar{x})}{\partial \bar{x}}(x^+ - \bar{x}) + g(\bar{x}) = 0. \tag{6.30}$$

By separating the first $k-1$ equations from the last one, we have that

$$\frac{\partial}{\partial \bar{x}} \begin{bmatrix} g_1(\bar{x}) \\ \vdots \\ g_{k-1}(\bar{x}) \end{bmatrix} (x^+ - \bar{x}) + \begin{bmatrix} g_1(\bar{x}) \\ \vdots \\ g_{k-1}(\bar{x}) \end{bmatrix} = 0 \tag{6.31}$$

and

$$\frac{\partial}{\partial \bar{x}} g_k(\bar{x})(x^+ - \bar{x}) + g_k(\bar{x}) = 0 \tag{6.32}$$

From (6.31), since $\bar{x} \in S_{k-1}$ and from Lemma 3, we have that $x^+ \in S_{k-1}$. While, in (6.32), we have that

$$\frac{\partial}{\partial \bar{x}} g_k(\bar{x}) = A_k, \quad g_k(\bar{x}) = A_k \bar{x} + b_k \tag{6.33}$$

Thus, (6.32) becomes

$$A_k(x^+ - \bar{x}) + A_k \bar{x} + b_k = 0 \ \rightarrow A_k x^+ + b_k = 0, \tag{6.34}$$

which is the definition of affine hyperplane, hence $x^+ \in S_k$. □

Lemma 6.2.4 shows that the k-th iteration of the Newton method, solving the pseudo-linear equation $g(x) = 0$, provides a solution lying in the hyperplane S_k, i.e. the analytic solution manifold of $g_k(x) = 0$. Therefore, at each iteration, the Newton method fixes the solution manifold of one equation and, after k iterations, it converges to the final and unique solution of the system. Unfortunately, in the EFA case, $g(x) = 0$ is replaced by the set of constraints of (6.28), where the right-hand side consists of the optimization variables A^W, B^W, A^H, B^H. Since those variables are not fixed during the iterations, convergence and uniqueness results do not automatically extend to the full optimization problem (6.28). However, we show next in simulations that problem (6.28) inherits numerically advantageous properties from the pseudo-linear constraints, resulting in superior algorithmic performances when compared to an unlifted formulation.

6.3 Numerical examples

In this section, first the conditioning problems and the possible solutions are illustrated. Then, algorithmic performance of Problem (6.28) for a class of WH systems are shown and discussed. Finally, the EFA is tested on real experimental benchmark data.

6.3.1 Illustration of the conditioning problem and its solution

In this example, the initialization problem for Case (1) of Theorem 6.2.1 is considered. The data are generated with the following true WH system

$$G_W(q) = \frac{1 - 0.4q^{-1}}{1 - 0.2q^{-1}}, \quad G_H(q) = \frac{1}{1 - 0.42q^{-1}}, \tag{6.35}$$

with a cubic polynomial as non-linearity

$$f(x_t) = x_t + 0.01x_t^2 + 0.03x_t^3. \tag{6.36}$$

Hence, the zero in G_W is very close to the pole in G_H. The relative distance is $\Delta = 0.02$. The system is excited by white Gaussian noise with variance 5, and the output is corrupted by white Gaussian noise with variance 1. The input/output data samples generated in this way ($N = 3000$) are used to identify the BLA first. The identified poles and zeros are then used to initialized the transfer functions for FA,

$$\begin{aligned} G_W^{FA} &= \frac{(1 - z_1 z^{-1})^{\beta_1}}{(1 - p_1 z^{-1})^{\alpha_1}(1 - p_2 z^{-1})^{\alpha_2}}, \\ G_H^{FA} &= \frac{(1 - z_1 z^{-1})^{1-\beta_1}}{(1 - p_1 z^{-1})^{1-\alpha_1}(1 - p_2 z^{-1})^{1-\alpha_2}}, \end{aligned} \tag{6.37}$$

where z_1, p_1, p_2 are the identified poles/zeros from the BLA, and $\beta_1, \alpha_1, \alpha_2$ are the corresponding fractional exponents, see (6.11). For the EFA, according Corollary 6.2.3, the expansion order for the numerator is $n_B^W = n_B^H = 1$ (one real zero in the BLA) and for the denominator is $n_A^W = n_A^H = 2$ (2 real poles in the BLA). Thus,

$$\begin{aligned} G_W^{EFA} &= \frac{1 - B_1^W(\beta_1) z^{-1}}{1 + A_1^W(\alpha_1, \alpha_2) z^{-1} + A_2^W(\alpha_1, \alpha_2) z^{-2}}, \\ G_H^{EFA} &= \frac{1 - B_1^H(\beta_1) z^{-1}}{1 + A_1^H(\alpha_1, \alpha_2) z^{-1} + A_2^H(\alpha_1, \alpha_2) z^{-2}}. \end{aligned} \tag{6.38}$$

These transfer functions are re-parametrized and constraints are introduced in order to get the lifted formulation (6.28). The true solution for the parameter vector $\theta = [\alpha_1, \alpha_2, \beta_1]$ is $\theta_0 = [1, 0, 1]$.

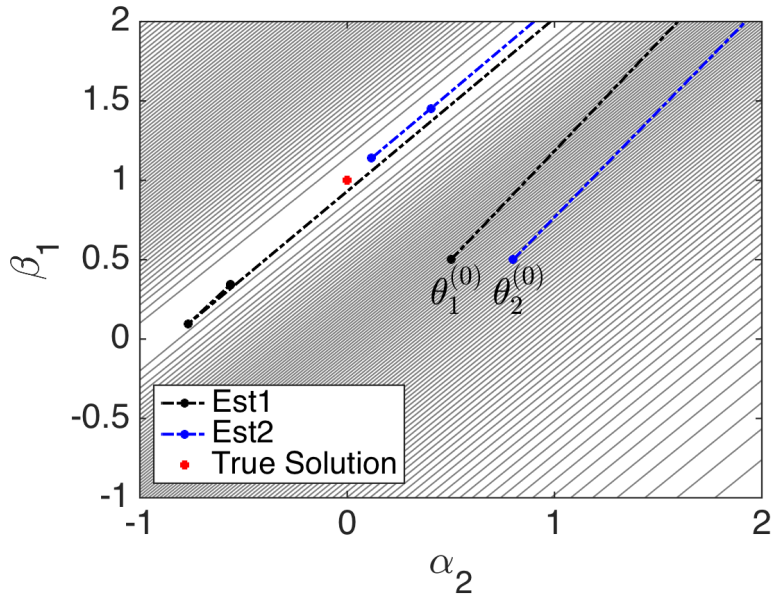


Figure 6.3: Contour plot of the cost function for FA. The small Δ is the cause of almost flat directions, which can be expressed by $\beta_1 = \alpha_2 + 1$. Two iterative estimations, starting from different initial guess $(\theta_1^{(0)}, \theta_2^{(0)})$, are plotted. They converge to two different solutions along the ill-conditioned direction, due to the lack of identifiability.

In the following, for a clear illustration of the identifiability problem, we will first consider the ideal case of exact identification of poles and zeros from the BLA, hence $z_1 = 0.4, p_1 = 0.2, p_2 = 0.42$. In this case, $\Delta = 0.02$ and Theorem 6.2.1 predicts very flat curvature of the Hessian of the cost function along the α_2, β_1 direction for the FA problem. This can be visualized by the contour plot of the cost function of the FA problem in Figure 6.3. In the same figure, the iterations of two estimation problems are depicted. Each estimation has a different initial guess for θ. In Figure 6.4, instead, the contour plot for the EFA problem is reported. In this case, no linear dependence occurs between the sensitivities w.r.t. β_1, α_2, and the optimization converges to the minimum of the cost function, which is not ill-conditioned. In Figure 6.5 and Figure 6.6, the solutions with explicit regularizing function (*reg*), and added inequality constraints (*box*), discussed for completeness at the end of Section 6.2.2, are tested on the same estimation problems. Table 6.3 shows the smallest eigenvalue and variance of the parameters for each estimation. Finally, the identified poles and zeros from the BLA are used. A Monte Carlo study has been conducted and 100 BLAs have been identified. In each case, due to estimation errors,

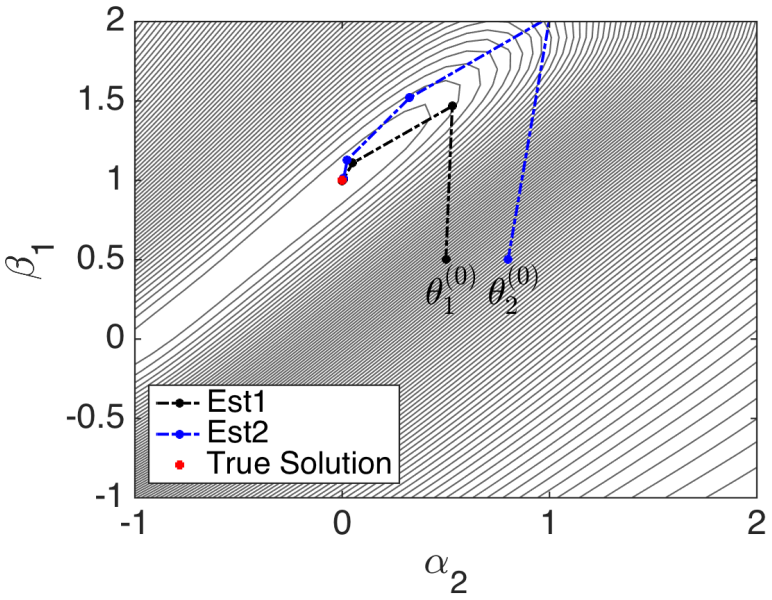


Figure 6.4: Contour plot of the cost function for EFA. In this case, the iterative two estimations, starting from different initial guess ($\theta_1^{(0)}$, $\theta_2^{(0)}$), converge to the well-conditioned minimum of the EFA cost function.

	λ_{min}	$\sigma^2_{\beta_1}$	$\sigma^2_{\alpha_2}$
FA Est1	0.2043	2.04	2.75
FA Est2	0.2697	1.54	2.08
EFA Est1	138.3	6×10^{-4}	2×10^{-4}
EFA Est2	163.8	4×10^{-4}	1×10^{-4}
FA+reg Est1	20.07	-	-
FA+reg Est2	20.13	-	-
FA+box Est1	0.2043	-	-
FA+box Est2	0.2697	-	-

Table 6.3: Smallest eigenvalue and variance of the parameters for all the estimations. For the FA, the idenfiability issue results in high variances of the parameters. The EFA shows well-conditioned eigenvalues and meaningful variances. The eigenvalue for the regularized FA is mainly due to the regularizing function. The FA with inequality constraints shows the same low eigenvalues of the FA and variance information is lost due to constraints.

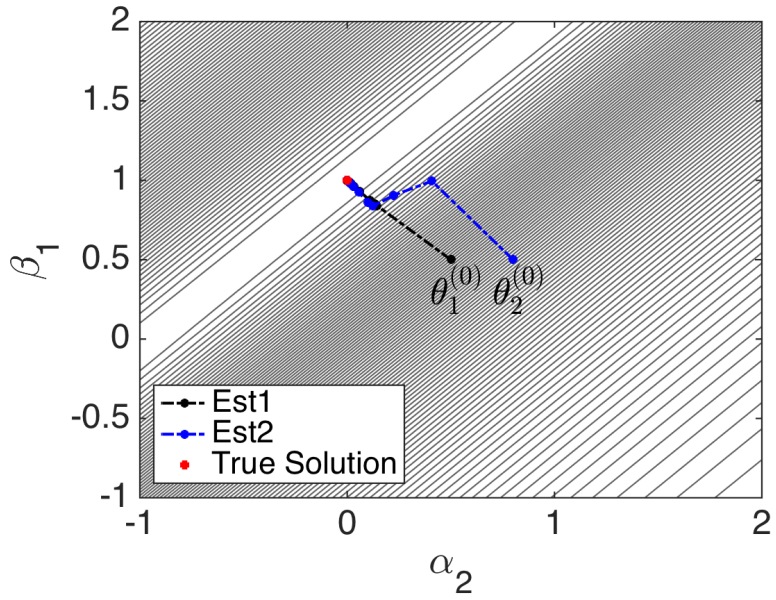


Figure 6.5: Contour plot of the cost function for FA with explicit regularization. Two iterative estimations converge to the same minimum, due to the artificial curvature introduced by the regularizing function. Such curvature is too tiny to be visible, but big enough to create a region of attraction around the solution.

the identified poles and zeros were lying in a neighbourhood of the true ones, given the finite amount of data used for identification. However, in all cases the right split was retrieved. Furthermore, in some cases, the distance between the identified z_1 and p_2 was even smaller than the true one. Hence, for these cases, the use of the EFA is even more justified.

6.3.2 Algorithmic performance

In this section, we illustrate the effectiveness of the lifted formulation of the EFA as in (6.28), by using some performance indices, such as computational time, average step-length, memory occupation (in terms of number of nodes of the symbolic expressions to store). As discussed at the end of Section 6.2.5, convergence and uniqueness results do not automatically extend to the full optimization problem (6.28). However, the pseudo-linearity of the constraints in (6.28) improves the performance, especially in the presence of strong non-linearity of the WH system. Furthermore, an efficient derivation of the pseudo-linear expressions in α, β is performed, which makes use of the recursive law (8.64) given in the proof of Theorem 6.2.3. Table 6.4 compares

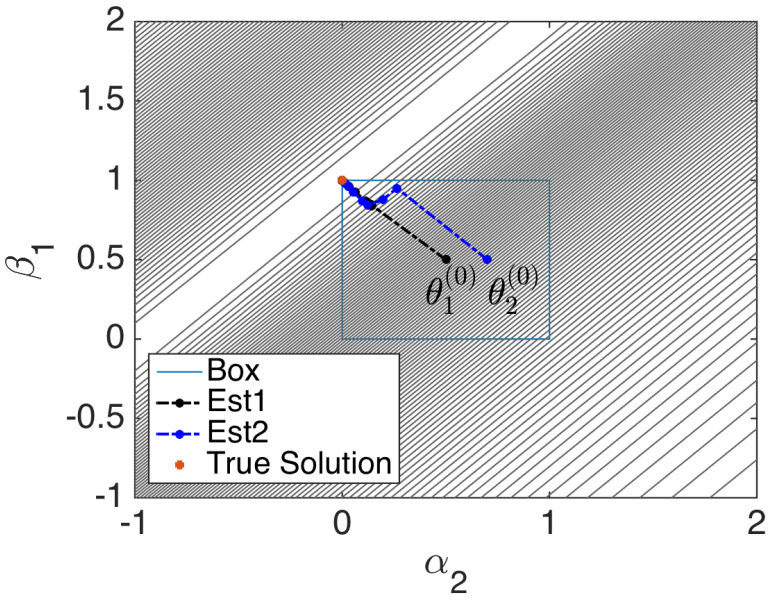


Figure 6.6: Contour plot of the cost function for FA with inequality constraints on the α, β parameters. Two iterative estimations converge to the corner $(0, 1)$ of the box, thanks to the gradient along the non-ill conditioned direction.

the EFA problem implemented by using the lifted formulation (6.28) (L) with the unlifted one (U), for which the filters coefficients are directly function of α, β, see Section 6.2.4. In Table 6.4, some key algorithmic indices are reported, regarding the estimation of different WH systems. Each tested WH system has the same linear parts but different polynomial non-linearities, which are parametrized in the following way

$$f(x_t, \theta_{NL}) = \sum_{k=0}^{d} \theta_{NL}^k x_t^k, \tag{6.39}$$

where d is the degree of the polynomial. For each tested WH system, 50 sets of 500 samples each are generated to derive average performance. The examples confirm that, for more complicated non-linearities, the overall complexity of the optimization problem is lower for the lifted formulation. In particular, almost always full Newton steps are taken in the lifted formulation and the overall computation time is, on average, half of the unlifted one. In terms of memory occupation, the symbolic expression for the unlifted version are composed by 17633 nodes and 8288 for the lifted one. Both lifted and unlifted formulations are solved using the Newton's method with Gauss-Newton

θ_{NL}	L/U	SL	Iters	T (s)
$[0, 1, 0.01, 0.03]$	L	1.0	8	8.41
	U	1.0	8	14.11
$[0, -25, 5, 1]$	L	1.0	10	15.05
	U	0.817	16	26.85
$[5, 5, -5, -1]$	L	1.0	8	10.56
	U	0.412	17	45.78
$[1, -3, -1, -5, 0.5]$	L	0.986	14	13.69
	U	0.525	24	53.02

Table 6.4: Algorithmic performance. L: lifted, U: unlifted formulation of the EFA. SL: average step-length, as fraction of full Newton's step-length (1.0). Iters: average number of iterations to get to convergence. T: average total computation time. The average is computed over 50 data sets of 500 samples each, for each non-linearity.

approximation of the Hessian. All the expressions (cost function, constraints, sensitivities) are built using CasADi, a symbolic framework for algorithmic differentiation and numeric optimization [82], and the optimization problems are solved with IPOpt, Interior Point OPTimizer, [83] in Python.

6.3.3 Benchmark example

In this section, the EFA is tested on the benchmark example introduced in Section 2.1.4. In this case, however, we use the data affected by measurement noise only. The available dataset is divided into estimation and validation sets. As in [84], the BLA is fitted with 6 poles, 5 zeros and one sample delay. The estimated BLA pole/zero pattern is shown in Figure 6.7. It can be seen that the pattern does not really suffer of the conditioning problems described by Theorem 6.2.1 (no poles/zeros very close to each other). For this reason, the EFA provides the same split as the FA, see [75]. Once this initialization problem is solved, poles and zeros are placed in the identified positions and a final optimization is performed over all parameters. The final RMS error, as defined in [25], on validation data, is 0.291 mV, in line with other methods: 0.295 mV for the FA in [75], 0.286 mV in [84], 0.27 mV in [74]. Hence, since the benchmark does not suffer of conditioning problems, the overall performance of the identification procedure is in line with other methods. However, we can still compare the lifted formulation of the EFA

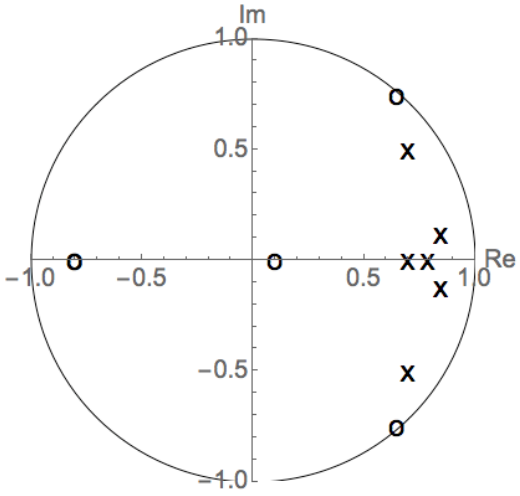


Figure 6.7: Poles and zeros of the estimated BLA for the WH benchmark example. One real zero lies outside the unit circle, at -3.2.

	SL	Iters	T (s)
Lifted EFA	0.97	19	110.13
Unlifted EFA	0.65	24	118.8
FA	0.58	44	222.45

Table 6.5: Lifted EFA vs Unlifted EFA vs FA. Algorithmic performance on the benchmark. SL: average step-length, as fraction of full Newton's step-length (1.0). Iters: number of iterations to get to convergence. T: average total computation time.

with other approaches, in terms of computation performance. Table 6.5 compares the lifted formulation EFA with its unlifted counterpart, and with the FA. Also in this benchmark case, the lifted formulation shows better performance, especially in terms of number of iterations to convergence and step-length.

6.4 Summary

In this chapter, we derived and analysed two initialization algorithms, based on the BLA, for the nonlinear identification of WH model structures. For both algorithms, the basic idea is to separate the identification of the parameters of the linear parts and of the nonlinearity. The parameters of the

two linear parts are, in fact, contained in the BLA and their identification boils down to find the correct partitioning of the BLA dynamics between G_W and G_H. Therefore, the initialization algorithm has to perform this partitioning. In this way, the linear parts can be initialized to their true value, provided by the partitioning, and a final estimation in all models parameters results into a simpler problem to initialize and solve.

The partitioning problem has been addressed separately for the case of stochastic WH models (presence of both process and measurement noise) and non-stochastic WH model (presence of measurement noise only).

In this first case, the partitioning problem has been solved via the *exhaustive search approach* combined with ML estimates of the nonlinearity. All possible poles/zeros combinations from the BLA are tested and, for each combination, the nonlinearity is estimated via the *partial re-sampling method*, implementing the MLE. The combination providing the best value of the ML criterion is selected and the corresponding poles and zeros were used to initialize a final MLE problem in all model parameters. The approach showed good results when tested on simulated and experimental data. However, the complexity of the approach is combinatorial in the number of combinations to test and, hence, its used is limited to low order models (up to 10).

In the second case, we focused on initialization algorithm for WH models when only measurement affects the model's outputs. In this case, the MLE problem boils down to nonlinear least squares problem and a more efficient algorithm for solving the partitioning problem can be used, i. e. the *fractional approach*. This approach makes use of fractional dynamics and it shows good performance in providing an initial estimate for the identification problem. However, we showed that it can become ill-conditioned for some pole/zero configuration of the two linear parts. The conditioning problem leads to a lack of identifiability of the parameters describing the splitting of the dynamics from the BLA. We showed that this issue can be solved via a series expansion of the fractional approach, which led to the development of the Expanded Fractional Approach (EFA). We proved that the EFA provides an implicit regularization in the estimation problem, which improves its conditioning.

Compared to the exhaustive search approach, the complexity of the fractional approach is lower, since it only requires the solution of one continuous optimization problem. However, so far its application is limited to the measurement noise framework only, since it has been developed for a consistent PEM estimator. Hence, one possibility to investigate is the combination of the fractional approach with the ML methods derived in Chapter 4, in order to reduce the complexity of the initialization algorithm for stochastic WH model too.

Part IV

Conclusions

Chapter 7

Conclusions and Recommendations for Future Work

7.1 Thesis conclusions and contributions

In the following, we summarize the main conclusions and contributions of the thesis.

The MLE and the intractability problem

In this thesis, we addressed the parameter estimation problem of stochastic nonlinear dynamical systems. For this class of systems, the main difficulty is the presence of a latent stochastic process affecting the system's outputs through a non-invertible nonlinear transformation. The most commonly used tool for parameter estimation is the Maximum Likelihood estimator (MLE), due to its desirable statistical properties, i.e. consistency and asymptotic efficiency. However, for the considered stochastic models, the MLE problem is intractable, since the likelihood function cannot be computed in closed-form. Ignoring the existence of the latent process may result into inconsistent estimators. Therefore, in order to obtain consistent estimates, approximate solutions to the intractable MLE problem have been developed.

Finding the MLE for tractable models

In case of tractable models, two main iterative algorithms are used to find the MLE. They are the gradient-based and the Expecatation-Maximization (EM) algorithm. The gradient-based algorithm makes use of the gradient of the likelihood to convergence to the closest optimizer. The EM, instead, uses

a proxy of the likelihood function, the Q-function, to iteratively maximize the likelihood. Unfortunately, for stochastic nonlinear models, both the likelihood (or its gradient) and the Q-function are intractable.

Finding the MLE via Monte Carlo approximations

A commonly used approach addressing the intractability problem is the Monte Carlo method. The gradient-based and the EM algorithms are modified by introducing Monte Carlo estimates of the gradient and of the Q-function. The resulting algorithms are Monte Carlo Maximum Likelihood (MCML) and the Monte Carlo Expectation Maximization (MCEM) methods. The advantage of methods based on Monte Carlo estimates is that they become exact as the Monte Carlo effort, i.e. the number of the required samples, goes to infinity. This also ensures the consistency results of the MLE. In practice, the actual sample size that can be used depends on the available computational resources. Hence, further modifications to the MCML and MCEM methods need to be developed.

Reducing the required sample size

One way to reduce the required sample size, while preserving accurate results, is to work with local Monte Carlo approximations in the parameter space. This can be achieved by making use of local sampling techniques. In this way, accurate, local approximations can be obtained from a relatively small sample size. However, these approximations depend on the current guess of the parameter, which may be far from the true solution. Thus, the local approximations have to be nested in an iterative loop: starting from an arbitrary initial guess of the parameter, a local approximation using MC integration is built, which is then maximized to obtain the next guess. This allows to keep the Monte Carlo error small. The procedure is then repeated in loop, until convergence. This idea is implemented in the nested MCML method, where MC estimates of the log-likelihood gradient are used.

The noisy search problem

The combination of MC estimates and optimization renders the nested MCML a *stochastic* parameter search algorithm. At each iterate of the loop, in fact, new random numbers (the samples) are generated and *noisy* MC estimates are used in the following optimization steps. As a result, the ML criterion is altered across the iterations. The main consequence is that the parameter search becomes noisy and, in some extreme cases, the search may become unstable.

The partial re-sampling method

An efficient solution to the noisy search problem problem is represented by the *partial re-sampling* method. In order to reduce the noise of the MC estimates used across the iterations, the proposed method tries to reuse as many samples as possible from previous iterations.

This is implemented by the *sensitivity-based re-sampling* rule, i.e. a re-sampling procedure that decides which samples from the current iteration can be used in future iterations too. The implicit function theorem provides an expression for the sensitivity of the parameter guesses w.r.t. the samples at each iteration. In this way, samples with high sensitivity are kept and reused in future iterations, while samples with low sensitivity are replaced by new generated samples. By controlling the threshold defining high and low sensitivity, it is possible to come up with an algorithm where both the noisy behaviour and the Monte Carlo error are reduced.

The optimization stage of the partial re-sampling method is solved via the Newton's method. As the stability issues of the search are solved via the re-sampling rule, no line-search techniques are required and full Newton's steps can be deployed at each iterate. This allows to achieve the quadratic convergence, typical of the Newton's method, at least for the iterations in which the number of replaced samples reduces drastically. It has been observed that this happens, in particular, in the neighbourhood of the solution.

Finally, the efficient reuse of samples generated at previous iterations results into a general reduction of the complexity of the samples simulation operation. Since the sensitivity-based re-sampling rule does not assume any specific samples simulation technique, it can be generalized and adopted by any iterative method for finding the MLE based on MCMC or SMC techniques.

The local convergence problem

In case of nonlinear systems, finding the MLE boils down to solving a nonlinear and non-convex optimization problem. In this case, all the iterative algorithms solving the MLE problem have proven convergence only to a local optimizer of the likelihood function or of its proxy. However, the ML estimate is found at the global maximum of the likelihood function. Therefore, it is important to derive initialization algorithm that have the sole purpose of finding a *good* initial guess, i.e. an initial point of the iterative search that increases the chances of converging to the global maximum of the likelihood.

Initialization algorithms for Wiener-Hammerstein systems

Addressing the problem of finding a good initial guess for general nonlinear models is a very challenging task. Thus, we focused on the Wiener-Hammerstein system. This is a block-oriented model structure where a static non-invertible nonlinearity is sandwiched between two LTI models. For this class of model, initialization algorithms rely on the fact that the best linear approximation (BLA) of the system contains the true linear dynamics WH model. In this way, initial estimates can be obtained by finding the right partition of these dynamics between the two linear parts. The remaining estimation problem in the parameters of the nonlinearity is in general easier to initialize and solve. In some cases, formal consistency of the initialization algorithm can be proved.

In this thesis, the partitioning problem has been solved via the *exhaustive search approach* and the *fractional approach*.

The exhaustive approach for stochastic WH systems

When a disturbance is present at the input of the nonlinearity, we are dealing with a stochastic nonlinear model. In this case, we proved that the BLA is still a consistent estimate of the linear dynamics and initialization algorithms based on linear approximations can be deployed. In particular, we developed an initialization algorithm by combining consistent ML estimates with the *exhaustive search* approach. For each possible combination of the linear dynamics contained in the BLA, the parameters of the nonlinearity are estimated via the partial re-sampling method. Since the partial re-sampling method is implementing the MLE, the estimated parameters are consistent and the combination providing the best value of the ML criterion corresponds to the right partition of the dynamics. These estimates are then used to initialize a final ML problem in all models parameters.

The fractional approach for WH systems

A major drawback of the exhaustive search approach is that its complexity is combinatorial in the number of possible partitions to test. An alternative solution to the partitioning problem is then represented by the *fractional approach* (FA). This initialization algorithm has been developed for the case of WH models only affected by measurement noise. The dynamics of the BLA are used to initialize both linear parts and fractional exponents are used to parametrize the partitioning. By estimating the fractional exponents, a partition of the dynamics can be decided in only one optimization problem. The approach show good performance in finding the initial guess for high

order WH models. However, we proved that it can be ill-conditioned for some particular dynamics configurations of the BLA. This conditioning problem also leads to identifiability issues. We proposed a modification of the FA, i.e. the *expanded fractional approach* (EFA). With this modification, the fractional dynamics are approximated via series expansions, which alleviate the conditioning problem.

Finally, we proved that a lifted formulation of the optimization problem resulting from the EFA allows for a faster and more reliable convergence to the solution when using Newton-type methods.

7.2 Possible future research directions

In this last section, we discuss some directions for possible future research.

Monte Carlo methods for MLE

Deriving efficient methods for MLE based on Monte Carlo approximations is a very active research area and still many issues need to be solved. In particular, it is quite challenging to scale these methods to high-dimensional models with many parameters. From this point of view, the solution derived in this thesis is a first attempt to reduce the overall complexity of the samples simulation operation, in order to allow the treatment of higher dimension problems. However, the complexity reduction was mainly a consequence of the actual goal we had in the second part of this thesis, i.e. the reduction of the noisy behaviour. Therefore, possible future work may investigate the improvement of the sensitivity-based re-sampling rule by addressing complexity issues directly. For example, the possibility of having a sample size M that changes across the iterations can be considered.

Connection with Bayesian estimation

As the sensitivity-based re-sampling rule implements an efficient use of the samples across local sampling explorations, it can be useful also in case of Bayesian estimation. The Bayesian framework, in fact, treats the parameter θ itself as a random variable and the parameter estimation method consists of computing the *posterior* distribution of θ given the observed data. Using Bayes' theorem, we have that this posterior is given by

$$p(\theta|\mathbf{y}) = \frac{p(\mathbf{y}|\theta)p(\theta)}{p(\mathbf{y})}, \tag{7.1}$$

where $p(\mathbf{y}|\theta)$ is the likelihood function that we extensively used in this thesis in case of frequentist framework, $p(\theta)$ is the *a priori* distribution describing

prior information available about the parameter, and $p(\mathbf{y})$ is known as the *marginal* likelihood, expressed as

$$p(\mathbf{y}) = \int p(\mathbf{y}|\theta)p(\theta)\mathrm{d}\theta. \tag{7.2}$$

This integral, depending on the models and distributions specification, may be intractable. Hence, Monte Carlo methods can be used for approximating the integral and samples simulation techniques may be required. In this thesis we have also seen that, for stochastic nonlinear models, the likelihood $p(\mathbf{y}|\theta)$ has to be computed via the marginalization operation,

$$p(\mathbf{y}|\theta) = \int p(\mathbf{x}, \mathbf{y}|\theta)\mathrm{d}\mathbf{x}, \tag{7.3}$$

which can be intractable too. Thus, the sensitivity-based re-sampling rule can be modified and adapted in case Monte Carlo methods based on local sampling explorations are needed to approximate the two intractable integrals (7.2) and (7.3).

Fractional approach for stochastic WH models

In this thesis, we presented and analysed the fractional approach as an initialization algorithm for the case of WH models affected by measurement noise only. One possible research direction is to extend the fractional approach to the case of stochastic WH model, i.e. when both measurement and process noise are present. To do this, the Monte Carlo Maximum Likelihood methods derived in the second part of this thesis have to combined with the fractional formulation of the dynamics required by the fractional approach. This may be challenging, since the overall optimization problem can become highly nonlinear. Hence, efficient modifications and, if needed, approximations, need to be investigated.

Part V

Appendices

Chapter 8

Appendices

8.1 The Monte Carlo method

The Monte Carlo (MC) method is a non-deterministic approach for numerical integration, see [85]. Assume we are interested in approximating the following integral,

$$I = \int_{\mathcal{X}} f(\mathbf{x}) g(\mathbf{x}) \mathrm{d}\mathbf{x} = \mathbb{E}_{g(\mathbf{x})}[f(\mathbf{x})], \tag{8.1}$$

where the function $f(\mathbf{x})$ is defined over the set $\mathcal{X}$ and it is integrable with respect to the probability distribution $g(\mathbf{x})$. When this integral cannot be solved in closed form, it is possible to approximate it via numerical integration. If $\mathbf{X}^M = \{X^{(m)}\}_{m=1}^M$ is a set of M random variables, i.i.d. according to $g(\mathbf{x})$, then a Monte Carlo estimator of the integral (8.1) is

$$\hat{I} = \frac{1}{M} \sum_{m=1}^{M} f(X^{(m)}), \quad X^{(m)} \sim g(\mathbf{x}). \tag{8.2}$$

The quantity $\hat{I}$ is an unbiased estimate of I,

$$\begin{aligned} \mathbb{E}_{g(\mathbf{x})}[\hat{I}] &= \mathbb{E}_{g(\mathbf{x})}\left[\frac{1}{M} \sum_{m=1}^{M} f(X^{(m)})\right] \\ &= \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{g(\mathbf{x})}\left[f(X^{(m)})\right] = \mathbb{E}_{g(\mathbf{x})}[f(\mathbf{x})] = I. \end{aligned} \tag{8.3}$$

Furthermore, the weak law of large numbers, see [51], tells us that for any arbitrarily small ϵ,

$$\lim_{M \to \infty} P(|\hat{I} - I| \geq \epsilon) = 0, \tag{8.4}$$

implying that we can achieve any required approximation accuracy if we use sufficiently large sample size M. Finally, the central limit theorem [51] implies

the convergence in distribution of the normalized Monte Carlo approximation errors,

$$\sqrt{M}\left(\hat{I} - I\right) \sim \mathcal{N}(0, \sigma^2), \tag{8.5}$$

where $\sigma^2 = \mathrm{Var}(X^{(m)})$. The results (8.3)-(8.4)-(8.5) are independent on the dimension of the space $\mathcal{X}$, see [54].

Importance sampling is a common technique for reducing the variance of the MC estimate or when direct sampling from $g(\mathbf{x})$ is not possible. The basic idea is to introduce a *importance sampling density* $q(\mathbf{x})$, also called *proposal*, and to rewrite the integral (8.1) in the following way,

$$I = \int_{\mathcal{X}} \frac{f(\mathbf{x})g(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{8.6}$$

In this way, a MC estimate of the integral can be obtained by drawing samples from the proposal $q(\mathbf{x})$,

$$\hat{I}_{IS} = \frac{1}{M} \sum_{m=1}^{M} f(X^{(m)}) \frac{g(X^{(m)})}{q(X^{(m)})}, \quad X^{(m)} \sim g(\mathbf{x}), \tag{8.7}$$

where $g(\mathbf{x})/q(\mathbf{x})$ is defined as the *importance ratio*. Using the Cauchy-Schwarz inequality, it can be shown that the variance of $\hat{I}_{IS}$ is lower than the variance of $\hat{I}$, see e.g. [85].

8.2 The Fisher's identity

In presence of a latent process $\mathbf{x}$, the Fisher's identity is a useful tool to express the gradient of the log-likelihood function. The identity states that

$$\nabla_\theta \log p(\mathbf{y};\theta) = \int \nabla_\theta [\log p(\mathbf{x},\mathbf{y};\theta)] p(\mathbf{x}|\mathbf{y};\theta) \mathrm{d}\mathbf{x}. \tag{8.8}$$

To prove the identity, we first recall that

$$p(\mathbf{y};\theta) = \int p(\mathbf{x},\mathbf{y};\theta) \mathrm{d}\mathbf{x}, \tag{8.9}$$

and that the differentiation rule of logarithm of a function is

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \frac{f'(x)}{f(x)}. \tag{8.10}$$

Hence, we write the gradient of the log-likelihood as

$$\nabla_\theta \log p(\mathbf{y};\theta) = \frac{\nabla_\theta p(\mathbf{y};\theta)}{p(\mathbf{y};\theta)}, \tag{8.11}$$

and we express the likelihood as integral of the joint distribution, see Equation (8.9),

$$\frac{\nabla_\theta p(\mathbf{y};\theta)}{p(\mathbf{y};\theta)} = \frac{\nabla_\theta \int p(\mathbf{x},\mathbf{y};\theta) \mathrm{d}\mathbf{x}}{p(\mathbf{y};\theta)}. \tag{8.12}$$

Under the assumption of regularity conditions, see [46], we can perform change of integration and differentiation, obtaining

$$\nabla_\theta \log p(\mathbf{y};\theta) = \int \frac{\nabla_\theta p(\mathbf{x},\mathbf{y};\theta)}{p(\mathbf{y};\theta)} \mathrm{d}\mathbf{x}. \tag{8.13}$$

We use again the differentiation rule of the logarithm again, but this time on the gradient of the log-joint probability,

$$\nabla_\theta \log p(\mathbf{x},\mathbf{y};\theta) = \frac{\nabla_\theta p(\mathbf{x},\mathbf{y};\theta)}{p(\mathbf{x},\mathbf{y};\theta)}, \tag{8.14}$$

to obtain an expression for $\nabla_\theta p(\mathbf{x},\mathbf{y};\theta)$,

$$\nabla_\theta p(\mathbf{x},\mathbf{y};\theta) = \nabla_\theta \log p(\mathbf{x},\mathbf{y};\theta) p(\mathbf{x},\mathbf{y};\theta). \tag{8.15}$$

By substituting this last expression in (8.13), we obtain,

$$\nabla_\theta \log p(\mathbf{y};\theta) = \int \frac{\nabla_\theta \log p(\mathbf{x},\mathbf{y};\theta) p(\mathbf{x},\mathbf{y};\theta)}{p(\mathbf{y};\theta)} \mathrm{d}\mathbf{x}, \tag{8.16}$$

and we recall that, according the Bayes' rule,

$$\frac{p(\mathbf{x}, \mathbf{y}; \theta)}{p(\mathbf{y}; \theta)} = p(\mathbf{x}|\mathbf{y}; \theta). \tag{8.17}$$

Finally, we have

$$\nabla_\theta \log p(\mathbf{y}; \theta) = \int \nabla_\theta [\log p(\mathbf{x}, \mathbf{y}; \theta)] p(\mathbf{x}|\mathbf{y}; \theta) \mathrm{d}\mathbf{x}. \tag{8.18}$$

8.3 Proof of Theorem 4.2.1 - Inconsistency of the standard PEM

By using the input-dependent transformation of the process noise after the nonlinearity, similarly to what has been done in [86], the true system can be written as

$$y_t = G_H(q, \theta_H^0)[f(G_W(q, \theta_W^0)u_t, \theta_{NL}^0) + \tilde{w}_t] + e_t \tag{8.19}$$

where

$$\tilde{w}_t = f(G_W(q, \theta_W^0)u_t + w_t, \theta_{NL}^0) - f(G_W(q, \theta_W^0)u_t, \theta_{NL}^0). \tag{8.20}$$

Statistical properties are not preserved in the transformation from w_t to $\tilde{w}_t$. In particular, signal $\tilde{w}_t$ is not independent of u_t, as it happens for w_t. This will be used later in the proof.

By using expression (8.20), the cost function $V_N(\theta_W, \theta_{NL}, \theta_H)$ can be written as

$$V_N(\theta_W, \theta_{NL}, \theta_H) = \frac{1}{N}\sum_{t=1}^{N}[G_H^0(f_t^0 + \tilde{w}_t) + e_t - G_H^0 f_t(\theta_{NL})]^2, \tag{8.21}$$

where, to simplify the notation, the following symbols are introduced,

$$G_H = G_H^0 \triangleq G_H(q, \theta_H^0) \tag{8.22a}$$

$$f_t^0 \triangleq f(G_W(q, \theta_W^0)u_t, \theta_{NL}^0) \tag{8.22b}$$

$$f_t(\theta_{NL}) \triangleq f(G_W(q, \theta_W^0)u_t, \theta_{NL}). \tag{8.22c}$$

The approximate PEM criterion then, becomes

$$\begin{aligned} V_N(\theta_{NL}) =& \frac{1}{N}\sum_{t=1}^{N}[G_H^0(f_t^0 + \tilde{w}_t) + e_t - G_H^0 f_t(\theta_{NL})]^2 \\ =& \frac{1}{N}\sum_{t=1}^{N}[G_H^0(f_t^0 - f_t(\theta_{NL}))]^2 \\ &+ \frac{1}{N}\sum_{t=1}^{N}[G_H^0\tilde{w}_t + e_t]^2 \\ &+ \frac{2}{N}\sum_{t=1}^{N}[G_H^0(f_t^0 - f_t(\theta_{NL}))(G_H^0\tilde{w}_t + e_t)]. \end{aligned} \tag{8.23}$$

Under the ergodicity assumption, the time averages tend to the mathematical expectations as N tends to infinity. The operator E, then, denotes

both mathematical expectation and averaging over time. Since the measurement noise e_t is zero mean and signals u_t and w_t are independent, as N tends to infinity, the criterion tends to

$$\begin{aligned}\bar{V}_N(\theta_{NL}) =&[G_H^0 E(f_t^0 - f_t(\theta_{NL}))]^2 + [G_H^0 E\tilde{w}_t]^2 + Ee_t^2 \\ &+ 2|G_H^0|^2[E(f_t^0 - f_t(\theta_{NL}))\tilde{w}_t].\end{aligned} \tag{8.24}$$

This cost is quadratic except for the last term $E(f_t^0 - f_t(\theta_{NL}))\tilde{w}_t$, linear in $f_t(\theta_{NL})$. We want to show that, due to the presence of this term, the estimation of θ_{NL} is biased, i.e.

$$\exists \theta_{NL}^* \neq \theta_{NL}^0 : \bar{V}_N(\theta_{NL}^*) < \bar{V}_N(\theta_{NL}^0). \tag{8.25}$$

Let us consider an arbitrarily small ε such that

$$\theta_{NL}^* = \theta_{NL}^0 + \varepsilon. \tag{8.26}$$

Since ε is small, we can focus on the first order Taylor approximation of the term $E(f_t^0 - f_t(\theta_{NL}))\tilde{w}_t$, at θ_{NL}^0. This will be

$$-E\left[\frac{\mathrm{d}f_t(\theta_{NL})}{\mathrm{d}\theta_{NL}}\bigg|_{\theta_{NL}^0}\tilde{w}_t\right]\varepsilon. \tag{8.27}$$

Since ε is arbitrary, it is enough to show that

$$E\left[\frac{\mathrm{d}f_t(\theta_{NL})}{\mathrm{d}\theta_{NL}}\bigg|_{\theta_{NL}^0}\tilde{w}_t\right] \neq 0. \tag{8.28}$$

Let us consider a linearly parametrized nonlinearity, i.e.

$$f_t(\theta_{NL}) = \theta_{NL}^T g(x_t^0), \tag{8.29}$$

with $x_t^0 = G_W(q, \theta_W^0)$. Thus,

$$\frac{\mathrm{d}f_t(\theta_{NL})}{\mathrm{d}\theta_{NL}}\bigg|_{\theta_{NL}^0} = g(x_t^0). \tag{8.30}$$

On the other hand, $\tilde{w}_t$ can be written, in terms of $g(x_t^0)$, as

$$\tilde{w}_t = \theta_{NL}^{0^T}[g(x_t^0 + w_t) - g(x_t^0)] \tag{8.31}$$

Since g is a polynomial nonlinearity, it exists at least one $n \geq 2$, such that $g(x^0 + w) = (x^0 + w)^n \neq 0$. This term can be expanded as

$$(x^0 + w)^n = x_0^n + nx_0^{n-1}w + \tbinom{n}{2}x_0^{n-2}(w)^2 + \mathcal{O}(w^3). \tag{8.32}$$

Therefore we have

$$\begin{aligned} g(x^0) &= x_0^n \\ \tilde{w}_t &= \theta_{NL}^{0^T}[(x_0^n + nx_0^{n-1}w + \mathcal{O}(w^3)) - x_0^n], \end{aligned} \tag{8.33}$$

and, since w is zero mean, the expression (8.28) becomes

$$\theta_{NL}^{0^T} E\left[x_0^n \left(\binom{n}{2} x_0^{n-2}(w)^2 + \mathcal{O}(w^3)\right)\right]. \tag{8.34}$$

Therefore, the argument of the expectation operator contains one term in the form $x_0^{2n-2}w^2$. Since, x_0 and w are independent and they are raised to positive powers, their expectation is different from 0.

8.4 Self-normalizing Importance Sampling

To make use of the importance sampling correction described in Section 4.3.1, we need to evaluate the posterior distribution $p(\mathbf{x}|\mathbf{y};\theta)$ on different values of θ from the one used for samples simulation. However, the posterior is only known up to a normalizing constant, i.e. the likelihood $p(\mathbf{y};\theta)$. Hence, the important weighting is implemented via a self-normalizing operation. Consider the importance sampling integration in (4.17), where we express the posterior in terms of joint and likelihood distributions, see (4.18),

$$\hat{G}(\theta, \mathbf{X}_k^M) = \frac{1}{M}\sum_{m=1}^{M} \Psi(X_k^{(m)};\theta)\frac{p(X_k^{(m)},\mathbf{y};\theta)}{p(X_k^{(m)},\mathbf{y};\theta^{(k)})}\frac{p(y;\theta^{(k)})}{p(y;\theta)}.$$

The self-normalization operation is implemented in the following way,

$$\hat{G}(\theta, \mathbf{X}_k^M) = \frac{\frac{p(\mathbf{y};\theta^{(k)})}{p(\mathbf{y};\theta)}\sum_{m=1}^{M} \Psi(X_k^{(m)};\theta)\frac{p(X_k^{(m)},\mathbf{y};\theta)}{p(X_k^{(m)},\mathbf{y};\theta^{(k)})}}{\sum_{m=1}^{M}\frac{p(X_k^{(m)},\mathbf{y};\theta)}{p(X_k^{(m)},\mathbf{y};\theta^{(k)})}\frac{p(\mathbf{y};\theta_k)}{p(\mathbf{y};\theta)}}.$$

The unknown constant ratios simplifies and we get

$$\hat{G}(\theta, \mathbf{X}_k^M) = \frac{\frac{1}{M}\sum_{m=1}^{M} \Psi(X_k^{(m)};\theta)\frac{p(X_k^{(m)},\mathbf{y};\theta)}{p(X_k^{(m)},\mathbf{y};\theta^{(k)})}}{\frac{1}{M}\sum_{m=1}^{M}\frac{p(X_k^{(m)},\mathbf{y};\theta)}{p(X_k^{(m)},\mathbf{y};\theta^{(k)})}}.$$

According to Assumption 2.3.1, all the quantities in this last expression can be evaluated. This provides Expression (4.19).

8.5 Proof of Theorem 5.2.1 - Consistency of the BLA

To prove the consistency result, we first recall Bussgang's theorem about cross spectra transformation.

Theorem 8.5.1. *(Bussgang) Let m_t and n_t be two real-valued, jointly Gaussian stationary process. Let f be a nonlinear function and g a stochastic process defined by*

$$g_t = f(n_t). \tag{8.35}$$

Then the cross spectrum between m and n is proportional to the cross spectrum between m and g:

$$\Phi_{mg}(\omega) = k\Phi_{mn}(\omega) \tag{8.36}$$

where k is areal-valued constant (that may be zero).

The proof can be found in [87]. Then, we recall the stochastic WH model equations,

$$\begin{aligned} x_t &= G_W(q,\theta_W)u_t + w_t, \\ z_t &= f(x_t,\theta_{NL}), \\ y_t &= G_H(q,\theta_H)z_t + e_t, \end{aligned} \tag{8.37}$$

and we define the following quantities, considering the true model description G_W^0, G_H^0, f^0,

$$\begin{aligned} x_t &= x_t^0 + w_t, \quad & x_t^0 &= G_W^0(q)u_t, \\ z_t &= f^0(x_t) & & \\ y_t &= y_t^0 + e_t, \quad & y_t^0 &= G_H^0(q)z_t. \end{aligned} \tag{8.38}$$

Since e and u are independent, the cross spectra between u and y, y^0 will be the same: $\Phi_{yu} = \Phi_{y^0u}$. Since u and w are Gaussian, then x_t is also Gaussian and Bussgang's theorem tells us that $\Phi_{zu} = k\Phi_{xu}$. Signals u and w are also independent, then $\Phi_{xu} = \Phi_{x^0u}$. Thus, we have the following result,

$$\Phi_{zu}(\omega) = k\Phi_{xu}(\omega) = k\Phi_{x^0u}(\omega) = kG_W^0(e^{\mathrm{i}\omega})\Phi_u(\omega). \tag{8.39}$$

Similarly, for the second linear system, we get

$$\Phi_{yu}(\omega) = G_H^0(e^{i\omega})\Phi_{zu}(\omega) = kG_H^0(e^{i\omega})G_W^0(e^{i\omega})\Phi_u(\omega). \tag{8.40}$$

Now, we know that $\hat{\theta}_N$, asymptotically, will minimize (see e.g. Chapter 8 in [2]) the function

$$\begin{aligned} V(\theta) =& E(y_t - G(q,\theta)u_t)^2 \\ =& \frac{1}{2\pi}\int (\Phi_y(\omega) - 2Re[G(e^{-\mathrm{i}\omega},\theta)\Phi_{yu}(\omega)] \\ &+ |G(e^{\mathrm{i}\omega},\theta)|^2\Phi_u(\omega))\mathrm{d}\omega. \end{aligned} \tag{8.41}$$

In (8.41), we can add and subtract the quantity $k^2|G_W^0(e^{\mathrm{i}\omega})G_H^0(e^{i\omega})|^2\Phi_u(\omega)$, obtaining

$$\begin{aligned} V(\theta) =& \frac{1}{2\pi}\int (\Phi_y(\omega) + k^2|G_W^0(e^{\mathrm{i}\omega})G_H^0(e^{i\omega})|^2\Phi_u(\omega) \\ & - k^2|G_W^0(e^{\mathrm{i}\omega})G_H^0(e^{i\omega})|^2\Phi_u(\omega) \\ & - 2Re[G(e^{-\mathrm{i}\omega},\theta)\Phi_{yu}(\omega)] \\ & + |G(e^{\mathrm{i}\omega},\theta)|^2\Phi_u(\omega))\mathrm{d}\omega. \end{aligned} \tag{8.42}$$

Since $\Phi_y(\omega)$ and $k^2|G_W^0(e^{\mathrm{i}\omega})G_H^0(e^{\mathrm{i}\omega})|^2\Phi_u(\omega)$ are θ-independent terms, minimizing $V(\theta)$ is the same as minimizing

$$\begin{aligned} W(\theta) =& \frac{1}{2\pi}\int (k^2|G_W^0(e^{\mathrm{i}\omega})G_H^0(e^{i\omega})|^2\Phi_u(\omega) \\ & - 2Re[G(e^{-\mathrm{i}\omega},\theta)\Phi_{yu}(\omega)] \\ & + |G(e^{\mathrm{i}\omega},\theta)|^2\Phi_u(\omega))\mathrm{d}\omega. \end{aligned} \tag{8.43}$$

We can substitute now the relation for $\Phi_{yu}(\omega)$ from (8.40),

$$\begin{aligned} W(\theta) =& \frac{1}{2\pi}\int (k^2|G_W^0(e^{\mathrm{i}\omega})G_H^0(e^{i\omega})|^2\Phi_u(\omega) \\ & - 2Re[kG(e^{-\mathrm{i}\omega},\theta)G_H^0(e^{i\omega})G_W^0(e^{i\omega})\Phi_u(\omega)] \\ & + |G(e^{\mathrm{i}\omega},\theta)|^2\Phi_u(\omega))\mathrm{d}\omega \end{aligned} \tag{8.44}$$

that leads to

$$W(\theta) = \frac{1}{2\pi}\int |kG_W^0(e^{\mathrm{i}\omega})G_H^0(e^{i\omega}) - G(e^{\mathrm{i}\omega},\theta)|^2\Phi_u(\omega)\mathrm{d}\omega, \tag{8.45}$$

which is minimized by $G(e^{\mathrm{i}\omega},\theta) = kG_W^0(e^{\mathrm{i}\omega})G_H^0(e^{\mathrm{i}\omega})$.

8.6 Proof of Theorem 6.2.1 and Corollary 6.2.1

For the smallest eigenvalue of $M(\theta)$, the following is true

$$\lambda_{min} = \min_{||x||_2=1} \quad x^T M(\theta) x \leq d^T M(\theta) d, \tag{8.46}$$

where x, d are two non-zero, unit vectors. Thus, by expressing $M(\theta)$ in vector form,

$$\lambda_{min} \leq d^T \left(\frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\theta} \right)^T \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\theta} d = || \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\theta} d ||^2, \tag{8.47}$$

where $\hat{\mathbf{y}}(\theta)$ is the vector collecting the model output for all time instants, $\hat{\mathbf{y}}(\theta) = [\hat{y}_1(\theta), \ldots, \hat{y}_N(\theta)]^T$. The sensitivity of $\hat{\mathbf{y}}(\theta)$ w.r.t. $\theta = [\alpha, \beta, \theta_{NL}]$ is

$$\begin{aligned} \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\theta} &= \frac{\mathrm{d}}{\mathrm{d}\theta} [\hat{G}_H(\alpha, \beta) f(\theta_{NL}, \hat{G}_W(\alpha, \beta)\mathbf{u})] \\ &= \frac{\mathrm{d}\hat{G}_H(\alpha, \beta)}{\mathrm{d}\theta} f(\theta_{NL}, \hat{G}_W(\alpha, \beta)\mathbf{u}) + \\ &+ \hat{G}_H(\alpha, \beta) \left[\frac{\partial f}{\partial \theta_{NL}} + \frac{\partial f}{\partial \hat{v}} \frac{d\hat{G}_W(\alpha, \beta)}{d\theta} \mathbf{u} \right], \end{aligned} \tag{8.48}$$

with $\mathbf{u} = [u(1), \ldots, u(N)]^T$ and $\hat{v} = \hat{G}_W(\alpha, \beta)\mathbf{u}$. The unitary vector d can be chosen in order to select two specific directions in the parameters space, related to the two cases of the theorem. Thus, we consider

$$d = [0, \ldots, 0, d_i, 0, \ldots, 0, d_j, 0, \ldots]^T, \tag{8.49}$$

where d_i, d_j refer to parameters θ_i, θ_j defined, for the two cases of the theorem, as: 1) $[\theta_i, \theta_j] = [\beta_i, \alpha_j]$, associated to a zero z_i and a pole p_j; 2) $[\theta_i, \theta_j] = [\beta_i, \beta_j]$, associated to two zeros z_i, z_j, or $[\alpha_i, \alpha_j]$, associated to two poles p_i, p_j. Thus, by using the vector (8.49), the inequality (8.47) becomes

$$\lambda_{min} \leq || \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\theta_i} d_i + \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\theta_j} d_j ||^2. \tag{8.50}$$

For Case 1, and from (8.48), it follows that

$$\begin{aligned} &\frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\beta_i} d_i + \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\alpha_j} d_j = \\ &= \left(\frac{\mathrm{d}\hat{G}_H}{\mathrm{d}\beta_i} d_i + \frac{\mathrm{d}\hat{G}_H}{\mathrm{d}\alpha_j} d_j \right) f(\theta_{NL}, \hat{G}_W \mathbf{u}) + \\ &+ \hat{G}_H \left[\frac{\partial f}{\partial \hat{v}} \left(\frac{\mathrm{d}\hat{G}_W}{\mathrm{d}\beta_i} d_i + \frac{\mathrm{d}\hat{G}_W}{\mathrm{d}\alpha_j} d_j \right) \mathbf{u} \right]. \end{aligned} \tag{8.51}$$

The fractional dynamics (6.11) can be expressed as

$$\hat{G}_W = \hat{G}'_W \frac{(1 - z_i q^{-1})^{\beta_i}}{(1 - p_j q^{-1})^{\alpha_j}},\ \hat{G}_H = \hat{G}'_H \frac{(1 - z_i q^{-1})^{1-\beta_i}}{(1 - p_j q^{-1})^{1-\alpha_j}},$$

where $\hat{G}'_W, \hat{G}'_H$ are the factors of $\hat{G}_W, \hat{G}_H$ not depending on β_i, α_j. The distance between the zero z_i and the pole p_j is Δ. Thus, by defining $z_i = p_j + \Delta$, we can linearise in Δ. Since the assumption is that Δ is small, we can focus on the first order term of the expansion. It can be easily verified that the sensitivities of the fractional dynamics w.r.t the parameter β_i and α_j can be expressed as

$$\frac{\mathrm{d}\hat{G}_W}{\mathrm{d}\beta_i} = \gamma_0^W(q) + \gamma_1^W(q)\Delta + \mathcal{O}(\Delta^2), \tag{8.52a}$$

$$\frac{\mathrm{d}\hat{G}_W}{\mathrm{d}\alpha_j} = -\gamma_0^W(q) + \gamma_2^W(q)\Delta + \mathcal{O}(\Delta^2), \tag{8.52b}$$

$$\frac{\mathrm{d}\hat{G}_H}{\mathrm{d}\beta_i} = -\gamma_0^H(q) + \gamma_1^H(q)\Delta + \mathcal{O}(\Delta^2), \tag{8.52c}$$

$$\frac{\mathrm{d}\hat{G}_H}{\mathrm{d}\alpha_j} = \gamma_0^H(q) + \gamma_2^H(q)\Delta + \mathcal{O}(\Delta^2), \tag{8.52d}$$

where the γ's functions are the zero and first order Taylor terms. By inserting (8.52) in (8.51), we have

$$\begin{aligned}
&\frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\beta_i} d_i + \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\alpha_j} d_j \approx \\
&\left(-\gamma_0^H d_i + \gamma_1^H \Delta d_i + \gamma_0^H d_j + \gamma_2^H \Delta d_j\right) f(\theta, \hat{G}_W \mathbf{u}) \\
&+ \hat{G}_H \left[\frac{\partial f}{\partial \hat{v}} \left(\gamma_0^W d_i + \gamma_1^W \Delta d_i - \gamma_0^W d_j + \gamma_2^W \Delta d_j\right) \mathbf{u}\right].
\end{aligned} \tag{8.53}$$

Choosing $d_i = d_j$, the expression simplifies and

$$\frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\beta_i} + \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\alpha_j} = \Gamma(\theta)\Delta + \mathcal{O}(\Delta^2), \tag{8.54}$$

where

$$\begin{aligned}
\Gamma(\theta) = &\left(\gamma_1^H + \gamma_2^H\right) f(\theta, \hat{G}_W \mathbf{u}) \\
&+ \hat{G}_H \left(\frac{\partial f}{\partial \hat{v}} \left(\gamma_1^W + \gamma_2^W\right) \mathbf{u}\right).
\end{aligned}$$

Hence for small Δ and by using (8.47), we conclude that

$$\lambda_{min} \leq |\Gamma(\theta)\Delta|^2,$$

which proves Theorem 6.2.1 for Case 1. By setting $\Delta = 0$, (8.54) satisfies Equation (6.18), yielding $\mu = 1$. This proves Corollary 6.2.1 for the Case 1.

In an analogous way, for Case 2, we have that, from (6.11),

$$\hat{G}_W = \hat{G}'_W \frac{1}{(1-p_i q^{-1})^{\alpha_i}(1-p_j q^{-1})^{\alpha_j}} \tag{8.55a}$$

$$\hat{G}_H = \hat{G}'_H \frac{1}{(1-p_i q^{-1})^{1-\alpha_i}(1-p_j q^{-1})^{1-\alpha_j}} \tag{8.55b}$$

with $p_j = p_i + \Delta$, in case of a pole in G_W is in the neighbourhood of another pole in G_H, and

$$\hat{G}_W = \hat{G}'_W (1 - z_i q^{-1})^{\beta_i}(1 - z_j q^{-1})^{\beta_j} \tag{8.56a}$$

$$\hat{G}_H = \hat{G}'_H (1 - z_i q^{-1})^{1-\beta_i}(1 - z_j q^{-1})^{1-\beta_j} \tag{8.56b}$$

with $z_j = z_i + \Delta$, in case of in case of a zero in G_W is in the neighbourhood of another zero in G_H. Similar derivations of the sensitivities via Taylor expansion lead to

$$\begin{aligned} &\frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\theta_i} d_i + \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\theta_j} d_j \approx \\ &\left(\gamma_0^H d_i + \gamma_1^H \Delta d_i + \gamma_0^H d_j + \gamma_2^H \Delta d_j\right) f(\theta, \hat{G}_W \mathbf{u}) \\ &+ \hat{G}_H \left[\frac{\partial f}{\partial \hat{v}} \left(\gamma_0^W d_i + \gamma_1^W \Delta d_i + \gamma_0^W d_j + \gamma_2^W \Delta d_j\right) \mathbf{u}\right]. \end{aligned} \tag{8.57}$$

where $[\theta_i, \theta_j]$ can be either $[\alpha_i, \alpha_j]$ or $[\beta_i, \beta_j]$. By setting $d_i = -d_j$, the expression simplifies and

$$\frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\beta_i} - \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\alpha_j} = \Gamma(\theta)\Delta + \mathcal{O}(\Delta^2), \tag{8.58}$$

where

$$\begin{aligned} \Gamma(\theta) = &\left(\gamma_1^H - \gamma_2^H\right) f(\theta, \hat{G}_W \mathbf{u}) \\ &+ \hat{G}_H \left(\frac{\partial f}{\partial \hat{v}} \left(\gamma_1^W - \gamma_2^W\right) \mathbf{u}\right). \end{aligned}$$

Hence, also in this case, for small Δ and by using (8.47), we conclude that $\lambda_{min} \leq ||\Gamma(\theta)\Delta||^2$, which proves Theorem 6.2.1 for Case 2. Finally, by setting $\Delta = 0$, (8.58) satisfies Equation (6.18), yielding this time $\mu = -1$. This proves Corollary 6.2.1 for the Case 2. □

8.7 Proof of Theorem 6.2.2

Similarly to the proof of Theorem 6.2.1, we can derive Equations (8.46) to (8.51), where $\hat{G}_W$ and $\hat{G}_H$ are replaced with their expanded versions. We expand only the two factors of $\hat{G}_W$ and $\hat{G}_H$ containing the pole and the zero close to each other. Thus, $n_A = n_B = 1$ and the lowest expansion orders fulfilling Property 1 are $n_1^W = n_1^H = n_2^W = n_2^H = 1$. By performing the expansions, we get

$$\hat{G}_W^{EFA} = \hat{G}'_W \tfrac{1-\beta_i z_i q^{-1}}{1-\alpha_j p_j q^{-1}}, \tag{8.59a}$$

$$\hat{G}_H^{EFA} = \hat{G}'_H \tfrac{1-(1-\beta_i) z_i q^{-1}}{1-(1-\alpha_j) p_j q^{-1}}. \tag{8.59b}$$

By defining $z_i = p_j + \Delta$, we can linearise in Δ. Since Δ is small, we focus on the first order term of the expansion. The sensitivities of the expanded fractional dynamics w.r.t the parameter β_i and α_j become

$$\frac{\mathrm{d}\hat{G}_W^{EFA}}{\mathrm{d}\beta_i} = \gamma_{\beta 0}^W(q) + \gamma_{\beta 1}^W(q)\Delta + \mathcal{O}(\Delta^2), \tag{8.60a}$$

$$\frac{\mathrm{d}\hat{G}_W^{EFA}}{\mathrm{d}\alpha_j} = \gamma_{\alpha 0}^W(q) + \gamma_{\alpha 1}^W(q)\Delta + \mathcal{O}(\Delta^2), \tag{8.60b}$$

$$\frac{\mathrm{d}\hat{G}_H^{EFA}}{\mathrm{d}\beta_i} = \gamma_{\beta 0}^H(q) + \gamma_{\beta 1}^H(q)\Delta + \mathcal{O}(\Delta^2), \tag{8.60c}$$

$$\frac{\mathrm{d}\hat{G}_H^{EFA}}{\mathrm{d}\alpha_j} = \gamma_{\alpha 0}^H(q) + \gamma_{\alpha 1}^H(q)\Delta + \mathcal{O}(\Delta^2), \tag{8.60d}$$

where the γ's functions are the zero and first order Taylor terms. Therefore, Equation (8.53) for the EFA case becomes

$$\begin{aligned}
\frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\beta_i} d_i + \frac{\mathrm{d}\hat{\mathbf{y}}(\theta)}{\mathrm{d}\alpha_j} d_j = \\
&= \left(\gamma_{\beta 0}^H d_i + \gamma_{\alpha 0}^H d_j\right) f(\theta, \hat{G}_W \mathbf{u}) \\
&\quad + \hat{G}_H \left[\frac{\partial f}{\partial \hat{v}} \left(\gamma_{\beta 0}^W d_i + \gamma_{\alpha 0}^W d_j\right) \mathbf{u}\right] \\
&\quad + \left(\gamma_{\beta 1}^H \Delta d_i + \gamma_{\alpha 1}^H \Delta d_j\right) f(\theta, \hat{G}_W \mathbf{u}) \\
&\quad + \hat{G}_H \left[\frac{\partial f}{\partial \hat{v}} \left(\gamma_{\beta 1}^W \Delta d_i + \gamma_{\alpha 1}^W \Delta d_j\right) \mathbf{u}\right] \\
&\quad + \mathcal{O}(\Delta^2).
\end{aligned}$$

Thus, for the smallest eigenvalue of $M(\theta)$, see (8.46), it holds that

$$\lambda_{min} = \min_{||d||_2 = 1} \quad ||\Gamma_0(\theta, d) + \Gamma_1(\theta, d)\Delta||^2,$$

where

$$\Gamma_0(\theta,d) = \left(\gamma^H_{\beta 0} d_i + \gamma^H_{\alpha 0} d_j\right) f(\theta, \hat{G}_W \mathbf{u}) + \hat{G}_H \left[\frac{\partial f}{\partial \hat{v}} \left(\gamma^W_{\beta 0} d_i + \gamma^W_{\alpha 0} d_j\right) \mathbf{u}\right],$$

$$\Gamma_1(\theta,d) = \left(\gamma^H_{\beta 1} d_i + \gamma^H_{\alpha 1} d_j\right) f(\theta, \hat{G}_W \mathbf{u}) + \hat{G}_H \left[\frac{\partial f}{\partial \hat{v}} \left(\gamma^W_{\beta 1} d_i + \gamma^W_{\alpha 1} d_j\right) \mathbf{u}\right].$$

Thus, the smallest eigenvalue of $M(\theta)$ does not go to zero with Δ. Furthermore, when $\Delta \equiv 0$,

$$\lambda_{min} = \min_{||d||_2=1} ||\mathbf{\Gamma}_0(\theta,d)||^2 = \min_{||d||_2=1} d^T \mathbf{\Gamma}_0(\theta)^T \mathbf{\Gamma}_0(\theta) d,$$

where

$$\mathbf{\Gamma}_0(\theta) = \begin{pmatrix} \gamma^H_{\beta 0} f(\theta, \hat{G}_W \mathbf{u}) + \hat{G}_H \left[\frac{\partial f}{\partial \hat{v}} \gamma^W_{\beta 0} \mathbf{u}\right] \\ \gamma^H_{\alpha 0} f(\theta, \hat{G}_W \mathbf{u}) + \hat{G}_H \left[\frac{\partial f}{\partial \hat{v}} \gamma^W_{\alpha 0} \mathbf{u}\right] \end{pmatrix}^T,$$

and $d = (d_i \quad d_j)^T$. The two components of $\mathbf{\Gamma}_0(\theta)$ only differ in the linear filters $\gamma^H_{\beta 0}, \gamma^H_{\alpha 0}$ and $\gamma^W_{\beta 0}, \gamma^W_{\alpha 0}$. Those filters are linear independent if

$$\begin{aligned} &\nexists \nu_H \in \mathbb{C} : \gamma^H_{\beta 0}(e^{i\omega}) + \nu_H \gamma^H_{\alpha 0}(e^{i\omega}) = 0, \forall \omega, \\ &\nexists \nu_W \in \mathbb{C} : \gamma^W_{\beta 0}(e^{i\omega}) + \nu_W \gamma^W_{\alpha 0}(e^{i\omega}) = 0, \forall \omega. \end{aligned} \tag{8.61}$$

Assuming that the input $\mathbf{u}$ is persistently exciting, the condition (8.61) holds. Hence, the two components of $\mathbf{\Gamma}_0(\theta)$ are linearly independent, and $\lambda_{min} \neq 0$ even when $\Delta = 0$.

8.8 Proof of Lemma 6.2.1

We first note that the system $g(x) = \bar{g}$ can be written in the homogeneous form $g(x) - \bar{g} = 0$, so that $\bar{g}$ is included in the b_k terms of the pseudo-linear definition. The last equation of the system is

$$g_M(x) - \bar{g}_M = 0 \tag{8.62}$$

and, according to the *pseudo-linear* definition, this equation is linear when evaluated on the solution manifold of the previous equations, i.e.

$$S_{M-1} = \{x | g_i(x) - \bar{g}_i = 0, i = 1, ..., M-1\}. \tag{8.63}$$

This manifold can be expressed as a linear combination of the M-th component x_M of x. Therefore, if $M = n$, the last equation (8.62) will be only function of x_M, and a unique value for this element can be found. By analytically propagating the solution backwards, from $i = M$ to $i = 1$, a unique solution for each element of x is found.

8.9 Proof of Theorem 6.2.3

To prove that $A(\eta)$ is pseudo-linear, first we define a recursive procedure to generate a pseudo-linear function $\tilde{A}(\eta)$ and then we prove that $A(\eta)$ can be derived by using the recursive procedure, i.e. $A(\eta) = \tilde{A}(\eta)$. Consider the set $S_0 \equiv \eta \in \mathbb{R}^n$, then the element $\tilde{A}_1(\eta) = \sum_{j=1}^n a_j \eta_j$ is linear for $\eta \in S_0$. Consider now the element $\tilde{A}_2(\eta) = \sum_{j=1}^n a_j \eta_j [\tilde{A}_1(\eta) - a_j]$. This element is linear for $\eta \in S_1 = \{\eta | A_1(\eta) = 0\}$. By generalizing this procedure we get the following recursive law, which satisfies Definition 1,

$$\tilde{A}_k(\eta) = \sum_{j=1}^n a_j \eta_j \phi_{k-1}^j(\eta) \quad \forall k \geq 1 \tag{8.64a}$$

$$\phi_k^j(\eta) = \tilde{A}_k(\eta) - k a_j \phi_{k-1}^j(\eta) \quad \forall k \geq 1 \tag{8.64b}$$

$$\phi_0^j(\eta) = 1 \tag{8.64c}$$

The k-th element of $A(\eta)$ is the k-th order Taylor coefficient of the expansion of $G(\eta, x)$, see (6.25). Thus, to prove that $A(\eta)$ can be derived using the 8.64's, we need to prove that the 8.64's hold for $G(\eta, x)$ as well. In terms of $G(\eta, x)$, the recursive procedure (8.64) is

$$\begin{aligned} G^{(k)}(\eta, x) &= \sum_{j=1}^n a_j \eta_j (1 + a_j x)^{-1} \tilde{\phi}_{k-1}^j(\eta, x) && (8.65a) \\ &\quad \forall k \geq 1 \\ \tilde{\phi}_k^j(\eta, x) &= G^{(k)}(\eta, x) && (8.65b) \\ &\quad -k a_j (1 + a_j x)^{-1} \tilde{\phi}_{k-1}^j(\eta, x) \\ \tilde{\phi}_0^j(\eta, x) &= G^{(0)}(\eta, x) = G(\eta, x) && (8.65c) \end{aligned}$$

This will, in fact, provide $\tilde{A}(\eta)$ when $x \equiv 0$. Hence, it remains to prove that the (8.65)'s also provide the k-th order derivative of $G(\eta, x)$, see (6.24), in order to get $A(\eta)$. By using induction, we get the following. For $k = 1$, the relation (8.65a) becomes

$$G^{(1)}(\eta, x) = \sum_{j=1}^n a_j \eta_j (1 + a_j x)^{-1} G(\eta, x), \tag{8.66}$$

which is the expression for the first order derivative of $G(\eta, x)$ with respect to x. Thus, for $k = 1$, the (8.65)'s hold. Assuming that the (8.65)'s hold for a generic k, we want to show that, by computing $G^{(k+1)}(\eta, x)$ as the derivative of $G^{(k)}(\eta, x)$, we get the relations as in (8.65), at $k + 1$. In the following, for simplicity, explicit dependence on η, x for $G^{(k)}$ and $\tilde{\phi}_k^j$ will be

omitted. Since (8.65a) holds for k, we have that

$$G^{(k+1)} = \frac{\partial}{\partial x} G^{(k)} = \frac{\partial}{\partial x} \left[\sum_{j=1}^{n} a_j \eta_j (1 + a_j x)^{-1} \tilde{\phi}_{k-1}^j \right]$$

By computing the partial derivative, we get that

$$G^{(k+1)} = \textstyle\sum_{j=1}^{n} a_j \eta_j (1 + a_j x)^{-1} \times \left(\frac{\partial}{\partial x} \tilde{\phi}_{k-1}^j - a_j (1 + a_j x)^{-1} \tilde{\phi}_{k-1}^j \right).$$

In order to get the relation (8.65a), we have to prove that the following is valid

$$\frac{\partial}{\partial x} \tilde{\phi}_{k-1}^j - a_j (1 + a_j x)^{-1} \tilde{\phi}_{k-1}^j = \tilde{\phi}_k^j, \tag{8.68}$$

where, from (8.65b), $\tilde{\phi}_k^j = G^{(k)} - k a_j (1 + a_j x)^{-1} \tilde{\phi}_{k-1}^j$. To show this, we can use induction again. For $k = 1$, the right-hand side of (8.68) becomes

$$\tilde{\phi}_1^j = G^{(1)} - a_j (1 + a_j x)^{-1} G^{(0)},$$

and the left-hand side

$$\frac{\partial}{\partial x} \tilde{\phi}_0^j - a_j (1 + a_j x)^{-1} \tilde{\phi}_0^j = G^{(1)} - a_j (1 + a_j x)^{-1} G^{(0)}.$$

Thus, for $k = 1$, (8.68) holds. Assuming that (8.68) is true for k, we want to show that it is also valid for $k + 1$. At $k + 1$, the right-hand side of (8.68) can be expressed, using (8.65b), as

$$\tilde{\phi}_{k+1}^j = G^{(k+1)} - (k+1) a_j (1 + a_j x)^{-1} \tilde{\phi}_k^j, \tag{8.69}$$

while the left-hand side becomes

$$\begin{aligned}
&\frac{\partial}{\partial x} \tilde{\phi}_k^j - a_j (1 + a_j x)^{-1} \tilde{\phi}_k^j = \\
&= \frac{\partial}{\partial x} \left[G^{(k)} - k a_j (1 + a_j x)^{-1} \tilde{\phi}_{k-1}^j \right] \\
&\quad - a_j (1 + a_j x)^{-1} \tilde{\phi}_k^j \\
&= G^{(k+1)} - k a_j \left(-a_j (1 + a_j x)^{-2} \tilde{\phi}_{k-1}^j \right. \\
&\quad \left. + (1 + a_j x)^{-1} \frac{\partial}{\partial x} \left[\tilde{\phi}_{k-1}^j \right] \right) - a_j (1 + a_j x)^{-1} \tilde{\phi}_k^j.
\end{aligned}$$

We can now use the assumption (8.68) to substitute $\frac{\partial}{\partial x} \left[\tilde{\phi}_{k-1}^j \right]$. Hence, we obtain that the previous expression is equal to $G^{(k+1)} - (k+1) a_j (1 + a_j x)^{-1} \tilde{\phi}_k^j$, also equivalent to (8.69). Therefore, the recursive procedure (8.65) is an alternative way to derive the k-th order derivative of $G(\eta, x)$ and, thus, $\tilde{A}(\eta) = A(\eta)$.

References

[1] G. Galilei, *Discorsi e Dimostrazioni Matematiche Intorno a Due Nuove Scienze*. Leyden, Netherlands, 1638.

[2] L. Ljung, *System Identification (2Nd Ed.): Theory for the User*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.

[3] E. Lehmann and G. Casella, *Theory of Point Estimation*. Springer Verlag, 1998.

[4] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[5] R. Pintelon and J. Schoukens, *System identification: a frequency domain approach*. New York: IEEE press, 2001.

[6] T. Söderström and P. Stoica, *System Identification*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[7] F. Giri and E. W. Bai, *Block-oriented nonlinear system identification*, ser. Lecture notes in control and information sciences, 2010, no. 404.

[8] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. Melbourne, FL, USA: Krieger Publishing Co., Inc., 2006.

[9] S. Billings, *Nonlinear System Identification: Narmax Methods in the Time, Frequency, and Spatio-Temporal Domains*. John Wiley & Sons, 2013.

[10] M. Schoukens and K. Tiels, "Identification of block-oriented nonlinear systems starting from linear approximations: A survey," *Automatica*, vol. 85, pp. 272 – 292, 2017.

[11] C. J. Geyer and E. A. Thompson, "Constrained Monte Carlo maximum likelihood for dependent data," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 54, no. 3, pp. pp. 657–699, 1992.

[12] C. J. Geyer, *Markov Chain Monte Carlo Maximum Likelihood.* Defense Technical Information Center, 1992.

[13] J. Durbin and S. J. Koopman, "Monte Carlo Maximum Likelihood estimation for non-Gaussian state space models," *Biometrika*, vol. 84, no. 3, pp. 669–684, 1997.

[14] G. C. G. Wei and M. A. Tanner, "A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 699–704, 1990.

[15] F. Lindsten, "An efficient stochastic approximation EM algorithm using conditional particle filters," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6274–6278.

[16] B. Ninness, A. Wills, and T. Schön, "Estimation of general nonlinear state-space systems," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010, pp. 6371–6376.

[17] T. B. Schön, A. Wills, and B. Ninness, "System Identification of Nonlinear State-space Models," *Automatica*, vol. 47, no. 1, pp. 39–49, Jan. 2011.

[18] A. Wills, T. B. Schön, L. Ljung, and B. Ninness, "Identification of Hammerstein-Wiener models," *Automatica*, vol. 49, no. 1, pp. 70–81, 2013.

[19] T. B. Schön, F. Lindsten, J. Dahlin, J. Wågberg, C. A. Naesseth, A. Svensson, and L. Dai, "Sequential Monte Carlo Methods for System Identification," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 775–786, 2015.

[20] M. Abdalmoaty, "Learning stochastic nonlinear dynamical systems using non-stationary linear predictors," Licentiate thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2017.

[21] M. Enqvist, "Linear models of nonlinear systems," PhD thesis, Linköping University, Institutionen för systemteknik, 2005.

[22] L. Ljung, "Estimating Linear Time-invariant Models of Nonlinear Time-varying Systems," *European Journal of Control*, vol. 7, no. 2, pp. 203 – 219, 2001.

[23] J. Schoukens, A. Marconato, R. Pintelon, Y. Rolain, M. Schoukens, K. Tiels, L. Vanbeylen, G. Vandersteen, and A. V. Mulders, "System identification in a real world," in *2014 IEEE 13th International Workshop on Advanced Motion Control (AMC)*, March 2014, pp. 1–9.

[24] M. Schoukens, "Identification of parallel block-oriented models starting from the best linear approximation," PhD thesis, Vrije Universitet Brussels, 2015.

[25] J. Schoukens and L. Ljung, "Wiener-Hammerstein benchmark," 2009. [Online]. Available: http://tc.ifac-control.org/1/1/DataRepository/sysid-2009-Wiener-hammerstein-benchmark

[26] M. Schoukens and J. P. Noel, "Wiener-Hammerstein system with process noise," *Nonlinear System Identification Benchmarks Workshop*, Brussels, Belgium, 2016.

[27] R. A. Fisher, "On an absolute criterion for fitting frequency curves," *Messenger of Mathematics*, vol. 41, pp. 155–160, 1912.

[28] H. Cramér, "A contribution to the theory of statistical estimation," *Scandinavian Actuarial Journal*, pp. 85–94, 1946.

[29] R. C. Rao, "Information and the accuracy attainable in the estimation of statistical parameters," *Bulletin of the Calcutta Mathematical Society*, vol. 37, pp. 81–91, 1945.

[30] A. Wald, "Note on the Consistency of the Maximum Likelihood Estimate," *The Annals of Mathematical Statistics*, vol. 20, no. 4, pp. 595–601, 12 1949.

[31] H. Cramér, *Mathematical Methods of Statistics*. Princeton University Press, Princeton, 1946.

[32] P. E. Caines, *Linear Stochastic Systems*. John Wiley & Sons, Inc., 1987.

[33] M. Abdalmoaty and H. Hjalmarsson, "Simulated Pseudo Maximum Likelihood Identification of Nonlinear Models," in *The 20th IFAC World Congress*, 2017, pp. 14 058–14 063.

[34] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[35] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*. Springer Publishing Company, Incorporated, 2015.

[36] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, In Press, 2018.

[37] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[38] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley-Interscience, 2006.

[39] R. A. Boyles, "On the convergence of the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 45, no. 1, pp. 47–50, 1983.

[40] K. Lange, "A Gradient Algorithm Locally Equivalent to the EM Algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 57, no. 2, pp. 425–437, 1995.

[41] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[42] O. Cappé, R. Douc, E. Moulines, and C. Robert, "On the convergence of the Monte Carlo Maximum Likelihood method for latent variable models," *Scandinavian journal of Statistics*, vol. 29, no. 1998, pp. 615–635, 2002.

[43] A. Penttinen, *Modelling interaction in spatial point patterns: Parameter estimation by the Maximum Likelihood method*. Jyvaskyla Studies in Computer Science, Economics, and Statistics 7, 1984.

[44] A. Doucet and V. B. Tadić, "Parameter estimation in general state-space models using particle methods," *Annals of the Institute of Statistical Mathematics*, vol. 55, no. 2, pp. 409–422, Jun 2003.

[45] M. Kok, J. Dahlin, T. B. Schön, and A. Wills, "Newton-based Maximum Likelihood estimation in nonlinear state space models," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 398–403, 2015.

[46] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, 2nd ed., ser. Wiley series in probability and statistics. Hoboken, NJ: Wiley, 2008.

[47] R. C. Neath, "On Convergence Properties of the Monte Carlo EM Algorithm," *Institute of Mathematical Statistic Collection*, vol. 10, pp. 43–62, 2013.

[48] K. S. Chan and J. Ledolter, "Monte Carlo EM Estimation for Time Series Models Involving Counts," *Journal of the American Statistical Association*, vol. 90, no. 429, pp. 242–252, 1995.

[49] G. Fort and E. Moulines, "Convergence of the Monte Carlo expectation maximization for curved exponential families," *The Annals of Statistics*, vol. 31, no. 4, pp. 1220–1259, 08 2003.

[50] C. Andrieu and G. O. Roberts, "The pseudo-marginal approach for efficient Monte Carlo computations," *The Annals of Statistics*, vol. 37, no. 2, pp. 697–725, 04 2009.

[51] W. Feller, *An Introduction to Probability Theory and Its Applications, 2nd Edition.* John Wiley & Sons, New York, 1957.

[52] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[53] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, p. 97, 1970.

[54] A. Doucet and A. M. Johansen, *A tutorial on particle filtering and smoothing: fifteen years later. In D. Crisan and B. Rozovsky (eds.).* Oxford University Press, 2011.

[55] G. Kitagawa, "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[56] S. Malik and M. K. Pitt, "Particle filters for continuous likelihood evaluation and maximisation," *Journal of Econometrics*, vol. 165, no. 2, pp. 190–209, 2011.

[57] J. Dahlin, A. Wills, and B. Ninness, "Constructing Metropolis-Hastings proposals using damped BFGS updates," *18th IFAC Symposium on System Identification*, Stockholm, Sweden, 2018.

[58] G. Giordano and J. Sjöberg, "Maximum Likelihood identification of Wiener-Hammerstein system with process noise," *18th IFAC Symposium on System Identification*, Stockholm, Sweden, 2018.

[59] A. Svensson, F. Lindsten, and T. B. Schön, "Learning nonlinear state-space models using smooth particle-filter-based likelihood approximations," *18th IFAC Symposium on System Identification*, Stockholm, Sweden, 2018.

[60] G. Giordano, S. Gros, and J. Sjöberg, "A Newton-based method for Maximum Likelihood estimation from incomplete data," *to be submitted to Automatica*, 2018.

[61] A. Haryanto and K. S. Hong, "Maximum Likelihood identification of Wiener-Hammerstein models," *Mechanical Systems and Signal Processing*, vol. 41, no. 1-2, pp. 54–70, 2013.

[62] B. Delyon, M. Lavielle, and E. Moulines, "Convergence of a stochastic approximation version of the EM algorithm," *The Annals of Statistics*, vol. 27, no. 1, pp. 94–128, 03.

[63] A. C. Chiang, *Fundamental Methods of Mathematical Economics*, 3rd ed. McGraw-Hill, 1984.

[64] I. Meilijson, "A Fast Improvement to the EM Algorithm on its Own Terms," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 51, no. 1, pp. 127–138, 1989.

[65] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer, 2001.

[66] G. Giordano and J. Sjöberg, "Consistency Aspects of Wiener-Hammerstein Model Identification in Presence of Process Noise," *IEEE 55th Conference on Decision and Control (CDC)*, Las Vegas, USA, 2016.

[67] M. Schoukens, R. Pintelon, T. P. Dobrowiecki, and J. Schoukens, "Extending the Best Linear Approximation Framework to the Process Noise Case," *CoRR*, vol. abs/1804.07510, 2018.

[68] A. Wills and B. Ninness, "Generalised Hammerstein-Wiener system estimation and a benchmark application," *Control Engineering Practice*, vol. 20, no. 11, pp. 1097–1108, nov 2012.

[69] R. Pintelon and J. Schoukens, *System identification: a frequency domain approach*. New York: IEEE press, 2001.

[70] M. Enqvist and L. Ljung, "Linear approximations of nonlinear FIR systems for separable input processes," in *Automatica*, vol. 41, no. 3, 2005, pp. 459–473.

[71] J. Schoukens, R. Pintelon, and M. Enqvist, "Study of the LTI relations between the outputs of two coupled Wiener systems and its application to the generation of initial estimates for Wiener-Hammerstein systems," *Automatica*, vol. 44, no. 7, pp. 1654 – 1665, 2008.

[72] R. Pintelon, J. Schoukens, G. Vandersteen, and K. Barbé, "Estimation of nonparametric noise and FRF models for multivariable systems - Part I: Theory," *Mechanical Systems and Signal Processing*, vol. 24, no. 3, pp. 573 – 595, 2010.

[73] J. Sjöberg and J. Schoukens, "Initializing Wiener-Hammerstein models based on partitioning of the best linear approximation," *Automatica*, vol. 48, no. 2, pp. 353–359, 2012.

[74] J. Sjöberg, L. Lauwers, and J. Schoukens, "Identification of Wiener-Hammerstein models: Two algorithms based on the best split of a linear model applied to the SYSID'09 benchmark problem," *Control Engineering Practice*, vol. 20, no. 11, pp. 1119–1125, nov 2012.

[75] L. Vanbeylen, "A fractional approach to identify Wiener-Hammerstein systems," *Automatica*, vol. 50, no. 3, pp. 903–909, 2014.

[76] M. Abdalmoaty and H. Hjalmarsson, "Application of a Linear PEM Estimator to a Stochastic Wiener-Hammerstein Benchmark Problem," in *The 18th IFAC Symposium on System Identification*, Stockholm, Sweden, 2018.

[77] J. Albersmeyer and M. Diehl, "The Lifted Newton Method and its Application in Optimization," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1655–1684, 2010.

[78] G. Giordano, S. Gros, and J. Sjöberg, "An improved method for Wiener-Hammerstein system identification based on the Fractional Approach," *Automatica*, vol. 94, pp. 349 – 360, 2018.

[79] G. Giordano and J. Sjöberg, "A Time-Domain Fractional Approach for Wiener-Hammerstein Systems Identification," *18th IFAC Symposium on System Identification*, vol. 48, no. 28, pp. 1232–1237, 2015.

[80] H. Cartan, *Elementary Theory of Analytic Functions of One Or Several Complex Variables*, ser. Dover books on mathematics. Dover Publications, 1995.

[81] L. V. Ahlfors, *Complex analysis*, ser. Mathematics Series. McGraw-Hill International Editions, 1987.

[82] J. Andersson, "A General-Purpose Software Framework for Dynamic Optimization," PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013.

[83] A. Wächter and L. T. Biegler, "On the Implementation of Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[84] D. T. Westwick and J. Schoukens, "Initial estimates of the linear subsystems of Wiener-Hammerstein models," *Automatica*, vol. 48, no. 11, pp. 2931–2936, 2012.

[85] R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1981.

[86] A. Hagenblad, L. Ljung, and A. Wills, "Maximum Likelihood identification of Wiener models," *Automatica*, vol. 44, no. 11, pp. 2697 – 2705, 2008.

[87] J. Bussgang, "Crosscorrelation functions of amplitude-distorted Gaussian signals," *RLE Technical Reports*, vol. 216, 1952.