Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Non-Parametric Bayesian Methods For Linear System Identification

Ph.D. candidate
Giulia Prando

Advisor
Prof. Alessandro Chiuso

Co-Advisor
Prof. Gianluigi Pillonetto

Director & Coordinator
Prof. Matteo Bertocco

Ph.D. School in
Information Engineering

Department of
Information Engineering
University of Padova
2016

Abstract

Recent contributions have tackled the linear system identification problem by means of non-parametric Bayesian methods, which are built on largely adopted machine learning techniques, such as Gaussian Process regression and kernel-based regularized regression. Following the Bayesian paradigm, these procedures treat the impulse response of the system to be estimated as the realization of a Gaussian process. Typically, a Gaussian prior accounting for stability and smoothness of the impulse response is postulated, as a function of some parameters (called hyper-parameters in the Bayesian framework). These are generally estimated by maximizing the so-called marginal likelihood, i.e. the likelihood after the impulse response has been marginalized out. Once the hyper-parameters have been fixed in this way, the final estimator is computed as the conditional expected value of the impulse response w.r.t. the posterior distribution, which coincides with the minimum variance estimator. Assuming that the identification data are corrupted by Gaussian noise, the above-mentioned estimator coincides with the solution of a regularized estimation problem, in which the regularization term is the ℓ_2 norm of the impulse response, weighted by the inverse of the prior covariance function (a.k.a. kernel in the machine learning literature).

Recent works have shown how such Bayesian approaches are able to jointly perform estimation and model selection, thus overcoming one of the main issues affecting parametric identification procedures, that is complexity selection.

While keeping the classical system identification methods (e.g. Prediction Error Methods and subspace algorithms) as a benchmark for numerical comparison, this thesis extends and analyzes some key aspects of the above-mentioned Bayesian procedure. In particular, four main topics are considered.

Prior design. Adopting Maximum Entropy arguments, a new type of ℓ_2 regularization is derived: the aim is to penalize the rank of the block Hankel matrix built with Markov coefficients, thus controlling the complexity of the identified model, measured by its McMillan degree. By accounting for the coupling between different input-output

channels, this new prior results particularly suited when dealing for the identification of MIMO systems.
To speed up the computational requirements of the estimation algorithm, a tailored version of the Scaled Gradient Projection algorithm is designed to optimize the marginal likelihood.

Characterization of uncertainty. The confidence sets returned by the non-parametric Bayesian identification algorithm are analyzed and compared with those returned by parametric Prediction Error Methods. The comparison is carried out in the impulse response space, by deriving “particle” versions (i.e. Monte-Carlo approximations) of the standard confidence sets.

Online estimation. The application of the non-parametric Bayesian system identification techniques is extended to an on-line setting, in which new data become available as time goes. Specifically, two key modifications of the original “batch” procedure are proposed in order to meet the real-time requirements. In addition, the identification of time-varying systems is tackled by introducing a forgetting factor in the estimation criterion and by treating it as a hyper-parameter.

Post processing: model reduction. Non-parametric Bayesian identification procedures estimate the unknown system in terms of its impulse response coefficients, thus returning a model with high (possibly infinite) McMillan degree. A tailored procedure is proposed to reduce such model to a lower degree one, which appears more suitable for filtering and control applications. Different criteria for the selection of the order of the reduced model are evaluated and compared.

Acknowledgements

During the last three years I have met special people who have helped me in reaching the results presented in this thesis. Many of them deserve my thanks.

My greatest thanks go to my advisor, Prof. Alessandro Chiuso: thank you for the things you taught me and for the passion you put in guiding me during these three years.

I would like to thank Prof. Gianluigi Pillonetto for sharing his inspiring ideas.

My sincere thanks go to my colleague Diego Romeres: working together has been productive and also very funny! I will not forget our long and interesting discussions, which also gladdened our office mates (now they all know what Empirical Bayes and MCMC are. . .)

I also would like to thank Prof. Michael I. Jordan for hosting me in his group at UC Berkeley.

Prof. Lennart Ljung and Prof. Simone Formentin, the reviewers of the first version of this thesis, also deserve my acknowledgement for their insightful comments.

Special thanks go to my office mates, who made these three years unforgettable and have become very good friends in my life. Thanks to Nicoletta and Giulia, who proofread the first draft of this thesis.

Last but not the least, I would like to thank my family: my grandmother, for having supported me in her own way and for having waited for me every evening; my parents for having continuously tried to transmit their faith in my potentialities, even if they still have to understand what I am doing...

Contents

Notational Conventions

Operators

$:=$	The left side is defined by the right side
$=:$	The right side is defined by the left side
$\|x\|_2$	Euclidean norm of vector x
$\|x\|_Q^2$	$x^\top Qx$ with Q symmetric positive definite weighting matrix
$\|A\|_F$	Frobenius norm of matrix A
$\otimes$	Kronecker product
0_d	Zero-vector of size d
$\{x(t)\}$	$\{x(t); t \in \mathbb{Z}\}$
$\arg\min\ f(x)$	Value of x which minimizes $f(x)$
$\mathrm{blockdiag}(A, B)$	Block diagonal matrix built with matrices A and B
$\mathrm{Cov}(x)$	Covariance matrix of the random vector x
$\frac{\partial}{\partial\theta}\ f(\theta, x)$	Partial derivative of f with respect to θ
$\det(A)$	Determinant of matrix A
$\mathbb{E}[x]$	Expectation of the random variable (or vector) x
$\delta_{t,s}$	Kronecker delta
$\mathrm{Df}_{\hat{y}}(x)$	Matricial degrees of freedom of estimator $\hat{y}$ expressed as function of x
$\mathrm{diag}(v)$	s Diagonal matrix having the vector v on the diagonal
$\xrightarrow{\text{dist}}$	Convergence in distribution
$\dim\,\theta$	Dimension of the column vector θ
$f'(x)$	Gradient of $f(x)$; a row vector of dimension $\dim\, x$ if f is scalar valued
$f''(x)$	Hessian of $f(x)$
$\mathbf{G}$	Square Hankel matrix built with the coefficients of the sequence $\{g(k)\}_{k=1}^N$
$\bar{\mathbf{G}}$	Shifted square Hankel matrix built with the coefficients of the sequence $\{g(k)\}_{k=1}^N$
$\overline{H}(\cdot)$	Entropy function
I_d	Identity matrix of size $d \times d$
A^{-1}	Inverse of matrix A
$A^\dagger$	Moore-Penrose pseudo-inverse of A
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and covariance Σ
q	Shift operator
$\mathbb{N}$	Set of natural numbers

$\mathbb{N}_+$	$\mathbb{N} \setminus \{0\}$
$\Pr\{A\}$	Probability of event A
$p_x(\cdot)$	PDF of random vector x
$\mathbb{R}$	Set of real numbers
$\mathbb{R}_+$	Set of positive real numbers: $\mathbb{R}_+ := [0, \infty)$
$\mathbb{R}^d$	Euclidean d-dimensional space
$\mathbb{R}^{m \times n}$	Space of real matrices with m rows and n columns
$\mathrm{range}(A)$	Column space of matrix A
$\mathrm{rank}(A)$	Rank of matrix A
$\mathrm{Tr}[A]$	Trace of matrix A
$A^\top$	Transpose of matrix A
$\mathrm{Var}(x)$	Variance of the random vector x
$\mathrm{vec}(A)$	Vectorization (column-wise) of matrix A
$x \sim p(x)$	The random variable x is distributed according to the probability distribution $p(x)$
$x \overset{i.i.d}{\sim} p(x)$	x is identically independently distributed according to distribution $p(x)$
y_s^t	$\{y(s), y(s+1), ..., y(t)\}$
y^t	$\{y(1), y(2), ..., y(t)\}$
$\mathbb{Z}$	Set of integer numbers

Symbols

$D_{\mathcal{M}}$	Set of values over which θ ranges for the model structure $\mathcal{M}$
$e(t)$	Disturbance variable at time t; typically $\{e(t)\}_{t=0}^{\infty}$ is assumed to be white noise
$\varepsilon(t,\theta)$	Prediction error $y(t) - \hat{y}(t\|\theta)$
$G(q)$	Transfer function from u to y
$G(q;\theta)$	Transfer function from u to y in a model $\mathcal{M}(\theta)$
$g(k)$	k-th sample of the impulse response from u to y
$\mathscr{G} = (A, B, C, D)$	Compact notation for a state-space system described by the matrices $A,\ B,\ C$ and D
$H(q)$	Transfer function from e to y
$H(q;\theta)$	Transfer function from e to y in a model $\mathcal{M}(\theta)$
$h(k)$	k-th sample of the impulse response from e to y
$\mathcal{M}$	Model structure (i.e. a mapping from the parameter space to a set of models)
$\mathcal{M}(\theta)$	Model corresponding to the parameter value θ
M	Set of models
$p(x)$	Probability distribution of the random variable x
$p_x(\cdot)$	Probability density function of the random variable x
$\varphi(t)$	Regressors vector at time t
$\psi(t,\theta)$	Gradient of $\hat{y}(t\|\theta)$ w.r.t θ
$\mathcal{S}$	The true data generating system
$S_x(\omega)$	Spectrum of the signal $\{x(t)\}$
Σ	Innovation variance for multi-variable systems, defined as $\Sigma = \mathrm{diag}([\sigma_1, \cdots, \sigma_p])$
$\widetilde{\Sigma}$	$\Sigma \otimes I_N$
σ	Innovation variance for SISO systems
θ	Parameter vector
$\hat{\theta}$	Estimate of θ computed using the dataset Z^N
$u(t)$	Input variable at time t
$v(t)$	Disturbance variable at time t
$V_N(\theta, \mathcal{D}^N)$	Parametric loss function defined on the dataset $\mathcal{D}^N$
$y(t)$	Output variable at time t
$\hat{y}(t)$	Predicted output at time t based on the data $\{y(t-1),\ u(t-1),\ ...,\ y(1),\ u(1)\}$
$\hat{y}(t\|\theta)$	Predicted output at time t using a model $\mathcal{M}(\theta)$ and based on the data $\{y(t-1),\ u(t-1),\ ...,\ y(1),\ u(1)\}$
$\mathcal{D}^N$	Dataset composed of N input-output data pairs, namely $\mathcal{D}^N = \{u(1), y(1),\ ...,\ u(N), y(N)\}$

Acronyms

ARMA	AutoRegressive Moving Average
ARMAX	AutoRegressive Moving Average with eXternal input
BIBO	Bounded Input Bounded Output
EM	Expectation Maximization
FIR	Finite Impulse Response
GLS	Generalized Least Squares
GP	Gaussian Process
GPR	Gaussian Process Regression
i.i.d.	identically independently distributed
LTI	Linear Time-Invariant
LS	Least Squares
MAP	Maximum A Posteriori
MCMC	Markov Chain Monte Carlo
MIMO	Multi Input Multi Output
MLE	Maximum Likelihood Estimate
MSE	Mean Square Error
OE	Output Error
PE	Prediction Error
PEM	Prediction Error Methods
ReLS	Regularized Least Squares
RKHS	Reproducing Kernel Hilbert Space
RLS	Recursive Least Squares
RPEM	Recursive Prediction Error Methods
SISO	Single Input Single Output
SGP	Scaled Gradient Projection
s.t.	subject to
SVD	Singular Value Decomposition
w.p.	With probability
w.r.t.	With respect to

1
Introduction

Control systems engineering aims at forcing a dynamical system to have a desired behaviour. The success of the discipline is highly dependent on the availability of an accurate mathematical model of the system to be controlled. In the continuous-time domain, such model consists of differential equations, while in the discrete-time regime it is described by a set of difference equations. A model may not only be used for the design of a desired controller, but also for simulation purposes, fault detection, quality control, etc. In addition, the presence of a model becomes essential when experiments performed through the real system are too expensive or too dangerous.
Physics first principles may provide a tool to derive such models; however, while in most cases the dynamical behaviour of interest could be too complex to be described through physical laws, in other cases, the physical model could not be suitable for its intended use. Indeed, the quality of a model should always be assessed in terms of its purpose: while a model may be good for simulation, it may not be the best one for control. Model complexity also plays a crucial role in control system engineering, where accuracy should always be traded-off with complexity: a complex model will lead to a complex controller and in turn to implementation and robustness issues. These considerations explain the development of techniques allowing to infer the mathematical model of a dynamical system from experimental data. *System Identification* is the discipline collecting all these procedures. As such, system identification appears as a preliminary step of any control system application, ranging from industrial plants to aeronautical vehicles, from home automation to humanoid robots.

The standard set-up of a system identification problem involves a set of input data, which are fed into the system under consideration, and a set of corresponding output data, recording the response of the system to the chosen input signal. The measurements, provided by suitable sensors, are typically affected by disturbances, whose presence has to be accounted for in the subsequent estimation stage. Most research in system identification has considered only noisy output data, while less attention has been devoted to the presence of disturbances on both input and output measurements (*errors-in-variables models*).
The described set-up can be fixed by the user (*experiment design*) according to the intended application. For instance, the user may choose the signals to measure and the excitation signal (*input design*) in order to maximize the information acquired from the performed experiment. Once the data are recorded, a pre-processing stage may be performed in order to remove undesired artefacts (e.g high-frequency disturbances, missing data, outliers, etc.).
The acquisition and pre-processing of the data is followed by the so-called inference step,

during which the data drive the search of the best model within the chosen model class. At this stage, a crucial role is played by the selection of the model class: it may be dictated by some a priori knowledge or, more frequently, by specific statistical procedures or by the chosen inference approach. This step is of primary concern not only in the context of system identification, but also in many statistical and learning applications, giving rise to a wide literature on this topic. Due to its importance, the theme of model selection will be widely discussed in the remainder of the manuscript.
The quality of the model returned by the inference procedure is then assessed (*model validation*). If the model does not properly describe the observed data or if it does not appear suitable for its intended use, the identification procedure has to be reviewed and a new model should be estimated.

The distinguishing tract of the estimation performed in system identification is the temporal relation present in the data: since the future output of a dynamical system depends on past input values, the prediction performed by the estimated model will be based on past measured input and outputs. System identification shares this characteristics with *econometrics*, the discipline which analyses economic data, trying to extract information from them. With the first works dated back at the end of the 19th century, econometrics has a longer tradition than system identification, which instead arose at the end of the 1950s, when the term was coined by Zadeh. However, the roots of system identification lie on the theory of stationary stochastic processes, which was mainly developed by the econometrics and times series communities between 1920 and 1970.
Two seminal papers, both published in 1965, paved the way for the future development of the two most common system identification techniques. The first work, due to Ho and Kalman, gave birth to the deterministic realization theory, thus laying the foundation of the subspace identification algorithms which blossomed in the Nineties. Åström and Bohlin, authors of the second seminal paper, introduced into the control community concepts and terminology coming from the econometrics field, specifically the Maximum Likelihood estimation of the coefficients of difference equation models (known as ARMA, ARMAX, etc.). The whole family of Prediction Error identification methods originated from this work and dominated the system identification field until the Nineties, when the lack of robust tools for the estimation of MIMO systems brought new interest on the realization approach. This renewed appeal led to the the development of subspace algorithms, which became the main focus of system identification research in the 1990s and in the early 2000s.
In parallel with these two main approaches, the Nineties awoke the interest for frequency domain identification with the aim of meeting the progresses reached by robust control

community, whose tools applied in the frequency domain. Another important research line arising in that period regarded the goal-oriented identification: the experiment design and the estimation stages were optimally designed in order to directly take into account the intended use of the model; thus, identification for control and optimal experiment design for control became hot topics around 1990.

The 1990s and the 2000s were also characterized by the wide development of the statistical learning and machine learning fields, with the introduction of new types of regularization, of the Support Vector Machines and with the application of neural networks. Even if many tools adopted by these communities could have been relevant for the system identification problem, only around 2010 some of them were extended to the control community for the estimation of dynamical systems. In particular, non-parametric Bayesian approaches relying on Gaussian Process Regression and on RKHS (Reproducing Kernel Hilbert Space) theory were introduced with the main goal of solving one of the crucial limitations affecting the older system identification techniques, that is the search for the best model structure. Indeed, while subspace methods overcame the issue of model parametrization through the estimation of a state-space model, model order (equivalently, complexity) selection still remained an open problem. Differently from the well-established system identification procedures, which require an a-priori choice of model complexity, Gaussian Process Regression provides an implicit way of dealing with the well-known bias-variance trade-off, allowing to jointly perform estimation and complexity selection.

This manuscript intends to offer new insights on the recently developed non-parametric Bayesian technique for system identification: analysis of some key properties as well as extensions of the original procedure will be provided. In an attempt to give continuity to the research in system identification, the investigation will consider the older approaches (specifically, Prediction Error Methods and subspace algorithms) as a benchmark for comparison.

In-line with the approach taken in the machine learning community, the innovative results will be mainly presented in an experimental way, meaning that the effectiveness of the proposed techniques will be mainly numerically validated.

1.1 Outline

The thesis aims at providing an overview of the three main system identification techniques, which have so far populated the literature of the field (that is, Prediction Error Methods, subspace algorithms and the recently developed non-parametric Bayesian approach). Special attention will be given to the latter with the purpose of understanding its pros

and cons, as well as of extending it in order to satisfy specific estimation requirements (such as real-time constraints or model complexity constraints). In addition, several links with the other two main families of identification algorithms will be highlighted.
A brief outline of the manuscript is provided in the following.

Chapter 2 is dedicated to the formal presentation of the linear system identification problem and to the illustration of the three main approaches to deal with it, i.e. the abovementioned Prediction Error Methods, subspace techniques and non-parametric Bayesian approaches. The description is enriched by details on the algorithmic implementation and on the choices that have to be taken by the user. The chapter concludes with a brief overview of classical model validation techniques.

Chapter 3 focuses on the role of regularization in system identification. After a brief introduction on the use of regularization in statistics and learning applications, an overview of the system identification approaches relying on ℓ_2- and ℓ_1-type regularization is provided. While ℓ_2-type penalties are adopted in order to enforce both numerical robustness and BIBO stability of the estimated system, ℓ_1-type regularization is mainly exploited for structure detection.
Recalling that the regularizer choice translates into the prior design when a Bayesian (probabilistic) framework is adopted, a maximum entropy argument is exploited to derive a new type of prior distribution to be used in the non-parametric Bayesian approach. Following the idea of elastic net in statistical learning, the proposed prior leads to a combination of ℓ_1 and ℓ_2 regularization, thus enforcing stability and structure constraints. This chapter is based on the results presented on the papers:

Prando G., Pillonetto G., and Chiuso A. The role of rank penalties in linear system identification. In *Proc. of 17th IFAC Symposium on System Identification, SYSID, Beijing*, 2015

Prando G., Chiuso A., and Pillonetto G. Bayesian and regularization approaches to multivariable linear system identification: the role of rank penalties. In *Proc. of IEEE CDC*, 2014

Prando G., Chiuso A., and Pillonetto G. Maximum entropy vector kernels for mimo system identification. *arXiv preprint arXiv:1508.02865, Automatica (accepted as regular paper)*, 2017

Chapter 4 is devoted to the analysis of the statistical properties of the estimate returned by a system identification procedure. Here the main focus will be on Prediction Error Methods and non-parametric Bayesian techniques: a comparison of the uncertainty (measured in terms of confidence sets) characterizing the obtained estimators will be drawn. The intrinsic difference between the two approaches (namely, the parametric/non-parametric nature) makes the comparison a bit tricky. To overcome the issue, a sampling approach is adopted, leading to the definition of "particle" confidence sets. The reported comparison is based on the results presented on the paper:

Prando G., Romeres D., Pillonetto G., and Chiuso A. Classical vs. bayesian methods for linear system identification: point estimators and confidence sets. In *Proc. of ECC*, 2016a

Chapter 5 deals with the problem of real-time identification, which would allow to update the system estimate as soon as new data arrive, as well as to track possible changes of the system parameters. This problem has been largely considered in the system identification literature, leading to the development of recursive algorithms both for Prediction Error Methods and for subspace algorithms. The first part of the chapter briefly reviews the real-time methods which have been proposed in the literature. The second part introduces a real-time reformulation of the "off-line" algorithm used to compute the non-parametric Bayesian estimator. By means of efficient updates of the data-related entities and of numerical expedients, a fast and robust algorithm is developed. The on-line reformulation of non-parametric Bayesian methods is based on the papers:

Romeres D., Prando G., Pillonetto G., and Chiuso A. On-line bayesian system identification. In *Proc. of ECC*, 2016

Prando G., Romeres D., and Chiuso A. Online identification of time-varying systems: a bayesian approach. In *Proc. of IEEE CDC*, 2016b

Chapter 6 considers the possibility of combining parametric and non-parametric approaches in order to jointly take advantage of their benefits. The aim is achieved by means of a two-steps procedure: first, a non-parametric Bayesian estimator is computed and secondly, it is converted into a lower order model estimated through Prediction Error Methods. Since the whole procedure can be regarded as a model reduction routine, the beginning of the chapter briefly reviews the role played by model reduction in system identification, with a particular focus on previously proposed two-steps procedures.
Part of the results of the chapter are based on the paper:

Prando G. and Chiuso A. Model reduction for linear bayesian system identification. In *Proc. of IEEE CDC*, 2015

Chapter 7 summarizes the main contributions of the thesis and outlines some possible future research directions.

2
System Identification Methods

This chapter intends to provide an overview of the three families of techniques which have dominated the system identification literature in the last fifty years. Section 2.1 introduces the problem faced by system identification methods and briefly discusses the different approaches taken by parametric and non-parametric techniques. Section 2.2 reviews the origins and main traits of Prediction Error Methods (PEM), introducing also the so-called transfer function models (Section 2.2.1). Section 2.3 is devoted to subspace algorithms and to the illustration of state-space models (Section 2.3.1). Non-parametric Bayesian methods are illustrated in Section 2.4: while the presentation is based on the Gaussian Process Regression (GPR) framework, connections with the theory of Reproducing Kernel Hilbert Spaces (RKHS) and with common Regularized Least-Squares (ReLS) practices are highlighted. Section 2.5 discusses several model validation procedures which are commonly adopted for model class selection. Some bibliographical notes are provided in Section 2.6.

2.1 System Identification Problem

This manuscript considers the identification of discrete-time causal linear systems: in particular, Linear Time-Invariant (LTI) systems will constitute the main focus of the thesis, while the Time-Varying framework will be shortly treated only in Chapter 5. In order to simplify the explanation, this introductory section will be dealing only with LTI systems.

The output signal $y(t) \in \mathbb{R}^p$ of an LTI system in response to an input $u(t) \in \mathbb{R}^m$ is defined as

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k), \qquad t = 0, 1, 2, ..., \quad g(k) \in \mathbb{R}^{p \times m} \tag{2.1}$$

Equation (2.1) makes clear how an LTI system is completely characterized by its impulse response $\{g(k)\}_{k=1}^{\infty}$; specifically, the ij-th element of $g(k)$ is the response detected at time k at the i-th output to a unit impulse applied at time 0 to input j.

In the classical system identification problem the input u is known exactly, while the output y may be corrupted by disturbance, due to e.g. measurement noise or to uncontrollable inputs. Their effect is accounted for through an additive term:

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + v(t), \qquad t = 0, 1, 2, ... \tag{2.2}$$

In addition $v(t) \in \mathbb{R}^p$ is assumed to be the output of another LTI system fed with white noise $e(t) \in \mathbb{R}^p$, namely:

$$v(t) = \sum_{k=0}^{\infty} h(k)e(t-k), \qquad t = 0, 1, 2, ..., \quad h(k) \in \mathbb{R}^{p \times p} \tag{2.3}$$

For normalization reasons, the assumption $h(0) = I_p$ is done. $\{e(t)\}$ is supposed to be a white noise sequence with probability density function $p_e(\cdot)$ such that

$$\mathbb{E}[e(t)] = 0_p \tag{2.4}$$

$$\mathbb{E}[e(t)e^\top(s)] = \Sigma\delta_{t,s}, \qquad \Sigma \in \mathbb{R}^{p \times p} \tag{2.5}$$

with $\delta_{t,s}$ denoting the Kronecker delta.[1] Throughout the manuscript, $e(t)$ and $u(s)$ are assumed to be independent for all $t, s \in \mathbb{Z}$, meaning that only open-loop operation conditions will be considered.

According to the previous assumptions, a general model of an LTI system is defined as

$$y(t) = G(q)u(t) + H(q)e(t), \qquad p_e(\cdot), \text{ PDF of } e \tag{2.6}$$

where $G(q) \in \mathbb{R}^{p \times m}$ and $H(q) \in \mathbb{R}^{p \times p}$ are the transfer function matrices

$$G(q) = \sum_{k=1}^{\infty} g(k)q^{-k}, \qquad H(q) = I_p + \sum_{k=1}^{\infty} h(k)q^{-k} \tag{2.7}$$

In the remainder of the manuscript $G(q)$ and $H(q)$ will be equivalently referred to as *transfer function matrices* or, simply, *transfer functions.* The two processes $\{y(t)\}$ and $\{u(t)\}$ are here assumed to be jointly stationary, thus implying the BIBO stability of the transfer function $G(q)$ (that is, it is analytic on and outside the unit disc of the complex plane, $|q| \geq 1$). Furthermore, both $H(q)$ and $1/H(q)$ are assumed to be BIBO stable.

Given a set of N input-output measurements $\mathcal{D}^N = \{u(t), y(t)\}_{t=1}^N$, system identification procedures aim at estimating the transfer function matrices $G(q)$ and $H(q)$ (or, equivalently, the impulse responses $\{g(k)\}_{k=1}^\infty$ and $\{h(k)\}_{k=1}^\infty$).

System identification appears as the art of *learning* the input-output behaviour of a dynamical system starting from a set of input-output data collected from the system itself. Any learning task is generally composed of three main stages: first, a *model class* M has to be chosen, i.e. a collection of *models* $\mathcal{M}$ through which the relationship of interest is described (a model may be e.g. a mathematical expression, a graph, etc.);

[1] When dealing with SISO systems, Σ will be denoted as σ.

secondly, the available data are used to select a specific model $\widehat{\mathcal{M}}$ within the set M and lastly, a *validation* stage is performed in order to assess whether $\widehat{\mathcal{M}}$ is able to correctly describe the input-output relationship of unseen data (Vapnik, 1998; Bishop, 2006).
The first and the latter stages of the described procedure are strictly connected, since a negative outcome of the latter may be an indicator of wrong decisions taken at the first stage, thus suggesting to review them and to perform again the whole "learning routine" (Ljung (1999) Ch.1, 16; Hastie, Tibshirani, and Friedman (2009)).
Obviously, the model class selection done at the first step also determines which estimation procedure will be adopted in the second stage. In particular, the choice between *parametric* and *non-parametric* models leads to two different families of system identification techniques. Parametric approaches specify a set of models completely characterized by a finite number of parameters, collected in the vector $\theta \in D_\theta \subset \mathbb{R}^{d_\theta}$; namely,

$$M = \left\{ \mathcal{M}(\theta) | \ \theta \in D_\theta \subset \mathbb{R}^{d_\theta} \right\} \tag{2.8}$$

with

$$\mathcal{M}(\theta): \quad y(t) = G(q,\theta)u(t) + H(q,\theta)e(t), \qquad p_e(\cdot,\theta), \text{ PDF of } e \tag{2.9}$$

and the system identification problem is thus reduced to the estimation of θ. Two classical parametric system identification techniques will be illustrated in the remainder of this chapter, specifically Prediction Error Methods (PEM) (Section 2.2) and subspace approaches (Section 2.3).
On the other hand, non-parametric models could be described through a function, a curve or even a table: for instance, the model class M may be the set of functions of class $\mathcal{C}^n$ (i.e. functions whose first n derivatives are continuous). Well-established non-parametric techniques working both in frequency and in time domain exist (see Ch. 6 in Ljung (1999) and Ch. 3 in Söderström and Stoica (1989)): some of them experimentally estimate the impulse response or the step response of the system by stressing it with a pulse or a step input, respectively (Rake (1980)); the *Empirical Transfer Function Estimate* (EFTE) estimates the system transfer function as the ratio of the Discrete Fourier Transforms of the given output and input signal measurements Kay (1988); Stoica and Moses (1997). Further details on this type of techniques are provided in Ljung (1999) (Ch. 6), Söderström and Stoica (1989) (Ch. 3) and in the survey Wellstead (1981). Recently, non-parametric approaches relying on statistical learning methods such as Gaussian Process Regression and kernel smoothing have been introduced into the system identification community Pillonetto and De Nicolao (2010); Pillonetto, Dinuzzo,

Chen, Nicolao, and Ljung (2014). They will be largely treated in Section 2.4 and in the remainder of the thesis: extensions of the original estimation routine will be proposed and several comparisons with classical parametric approaches will be carried out.

It should be pointed out that the previous discussion about parametric and non-parametric approaches has been confined to the system identification field; however, these two families of methods are widely applied both in statistical learning and econometric literature (Sheskin, 2003; Zhao et al., 2008).

The choice between parametric and non-parametric models is just the first step for a complete characterization of the selected model class. The model type has to be selected: parametric approaches involve a choice between e.g. transfer function or state-space models (see Sections 2.2.1 and 2.3.1), while function or table models could be estimated when applying non-parametric methods. Another important choice regards the complexity of the model class, here denoted as $C(M)$, which measures the flexibility of M. It could be the state-space size for state-space models, the polynomials degree for transfer function models or the kernel width when kernel smoothing techniques are exploited. Finally, the use of parametric methods also requires to specify an appropriate parametrization, i.e. a differentiable mapping $\mathcal{M}(\cdot) : D_\theta \to M$ from the parameter space to the chosen model class (this mapping is referred to as *model structure* in Ljung (1999)). As above-mentioned, while these choices have to be done at the first stage of any identification procedure, their validity is assessed at a later stage through model validation. The most common tools for model class selection and validation will be discussed in Section 2.5.

2.2 Prediction Error Methods

Prediction Error Methods (PEM) represent the original approach to the system identification problem; nowadays, they are a well-established parametric technique which has been largely treated in both control and econometrics textbooks (Ljung (1999); Söderström and Stoica (1989); Box and Jenkins (1970); Brockwell and Davis (2013); Hannan and Deistler (1988)).

The introduction of these techniques into the system identification field is strictly connected with the adoption of the so-called transfer function models: originally developed in the context of time series, starting from the Sixties they were extended to the field of dynamical systems by accounting also for the presence of an exogenous input (Aström, 1968; Mendel, 1973; Åström and Bohlin, 1966; Clarke, 1967; Kailath, 1980). A careful description of this family of models will be provided in Section 2.2.1.

Prediction Error Methods arise from the observation that the primary use of any identified model is prediction: for instance, the synthesis of a controller relies on the possibility of knowing at time $t-1$ what the output of the plant will be at time t. However, when the system is stochastic, an exact knowledge of this type is not achievable. These considerations suggest that the quality of an identified model could be evaluated in terms of its prediction ability, i.e. the capability of predicting the system output at time t using input and output data collected until time $t-1$. A suitable criterion for estimating the parameter vector θ would therefore try to minimize the so-called *prediction error* incurred at time t using the model $\mathcal{M}(\theta)$, i.e.

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta), \qquad \hat{y}(t|\theta) := w(t,\mathcal{D}^{t-1};\theta) \tag{2.10}$$

where $\hat{y}(t|\theta) := w(t,\mathcal{D}^{t-1};\theta)$ denotes the prediction of $y(t)$ given the data up to $t-1$, i.e. $\{y(t-1),\ u(t-1),...,\ y(1),\ u(1)\}$. The most commonly adopted predictor is the so-called *mean-square predictor*, which minimizes the variance of the prediction error (see Söderström and Stoica (1989), Sec. 7.3 and Ljung (1999), Sec. 3.2 for its derivation); for the general model (2.9), this is defined as

$$\begin{aligned} \hat{y}(t|\theta) &= F_u(q,\theta)u(t) + F_y(q,\theta)y(t) \qquad (2.11)\\ F_u(q,\theta) &: = H^{-1}(q,\theta)G(q,\theta) \\ F_y(q,\theta) &: = \left\{I_p - H^{-1}(q,\theta)\right\} \end{aligned}$$

Consequently, the prediction error (2.10) is given by

$$\varepsilon(t,\theta) = H^{-1}(q,\theta)\left\{y(t) - G(q,\theta)u(t)\right\} \tag{2.12}$$

Once the one-step ahead predictor has been defined, the probabilistic description of an LTI system given in (2.9) can be reformulated in terms of prediction as

$$\begin{aligned} \mathcal{M}(\theta): \qquad & \hat{y}(t|\theta) = w(t,\mathcal{D}^{t-1};\theta) \qquad (2.13)\\ & \varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta), \quad \varepsilon(t,\theta) \text{ independent and with PDF } p_e(\cdot,t;\theta) \end{aligned}$$

Given a dataset $\mathcal{D}^N$, PEM return an estimate of θ by minimizing a scalar function $V_N(\theta,\mathcal{D}^N)$ of the prediction errors $\{\varepsilon(t,\theta)\}_{t=1}^N$; specifically

$$\hat{\theta}_N = \underset{\theta\in D_\theta}{\arg\min}\ V_N(\theta,\mathcal{D}^N) \tag{2.14}$$

To enforce a desired frequency weighting, Ljung (1999) suggests to apply the function $V_N(\theta, \mathcal{D}^N)$ after having filtered the prediction errors with a stable linear filter.
The remainder of this section is organized as follows. Section 2.2.1 introduces the classical transfer function models which are adopted in connection with PEM. The choices that the user has to take when applying PEM are discussed in Section 2.2.2, while the connection between PEM and ML estimation is illustrated in Section 2.2.3. Finally, algorithmic details are provided in Section 2.2.4.

2.2.1 Transfer Function Models

Transfer function models (also known as *black-box* models) parametrize $G(q,\theta)$ and $H(q,\theta)$ in (2.9) as rational functions, thus collecting in θ the numerator and the denominator coefficients.
In its more general form, a transfer function model is given by

$$A(q,\theta)y(t) = F^{-1}(q,\theta)B(q,\theta)u(t) + D^{-1}(q,\theta)C(q,\theta)e(t) \tag{2.15}$$

The matrix polynomials in (2.15) are defined as

$$A(q,\theta) = I_p + A_1 q^{-1} + \cdots + A_{n_a} q^{-n_a}, \qquad A_i \in \mathbb{R}^{p\times p},\ i=1,...,n_a \tag{2.16}$$
$$B(q,\theta) = B_1 q^{-1} + \cdots + B_{n_b} q^{-n_b}, \qquad B_i \in \mathbb{R}^{p\times m},\ i=1,...,n_b \tag{2.17}$$
$$C(q,\theta) = I_p + C_1 q^{-1} + \cdots + C_{n_c} q^{-n_c}, \qquad C_i \in \mathbb{R}^{p\times p},\ i=1,...,n_c \tag{2.18}$$
$$D(q,\theta) = I_p + D_1 q^{-1} + \cdots + D_{n_d} q^{-n_d}, \qquad D_i \in \mathbb{R}^{p\times p},\ i=1,...,n_d \tag{2.19}$$
$$F(q,\theta) = I_p + F_1 q^{-1} + \cdots + F_{n_f} q^{-n_f}, \qquad F_i \in \mathbb{R}^{p\times p},\ i=1,...,n_f \tag{2.20}$$

Starting from the general model (2.15), 32 different model structures can be derived, according to which polynomials are estimated. The most common ones are listed in the following.

FIR: The FIR model structure contains only the matrix polynomial $B(q,\theta)$ (corresponding to $n_a = n_c = n_d = n_f = 0$),

$$y(t) = B(q,\theta)u(t) + e(t) \tag{2.21}$$

and $\theta \in \mathbb{R}^{mn_bp}$ consists of the coefficients of the B_i polynomials:

$$\theta = \left[\text{vec}^\top(B_1)\ \text{vec}^\top(B_2)\ \cdots\ \text{vec}^\top(B_{n_b})\right]^\top \tag{2.22}$$

OE: When $n_a = n_c = n_d = 0$ the OE model structure arises:

$$y(t) = F^{-1}(q,\theta)B(q,\theta)u(t) + e(t) \tag{2.23}$$

with $\theta \in \mathbb{R}^{(n_b m + n_f p)p}$ given by

$$\theta = \begin{bmatrix} \mathrm{vec}^\top(B_1) & \cdots & \mathrm{vec}^\top(B_{n_b}) \; \mathrm{vec}^\top(F_1) \cdots \mathrm{vec}^\top(F_{n_f}) \end{bmatrix}^\top \tag{2.24}$$

ARX: The ARX model structure arises when $n_c = n_d = n_f = 0$, leading to

$$A(q,\theta)y(t) = B(q,\theta)u(t) + e(t) \tag{2.25}$$

In this case, the parameter vector $\theta \in \mathbb{R}^{(n_a p + n_b m)p}$ contains the coefficient matrices

$$\theta = \begin{bmatrix} \mathrm{vec}^\top(A_1) \; \mathrm{vec}^\top(A_2) & \cdots & \mathrm{vec}^\top(A_{n_a}) \; \mathrm{vec}^\top(B_1) & \cdots \mathrm{vec}^\top(B_{n_b}) \end{bmatrix}^\top \tag{2.26}$$

ARMAX: Setting $n_d = n_f = 0$ coincides with defining an ARMAX model structure

$$A(q,\theta)y(t) = B(q,\theta)u(t) + C(q,\theta)e(t) \tag{2.27}$$

In this case $\theta \in \mathbb{R}^{(n_a p + n_b m + n_c p)p}$ is given by

$$\theta = \begin{bmatrix} \mathrm{vec}^\top(A_1) & \cdots & \mathrm{vec}^\top(A_{n_a}) \; \mathrm{vec}^\top(B_1) & \cdots \mathrm{vec}^\top(B_{n_b}) \; \mathrm{vec}^\top(C_1) & \cdots & \mathrm{vec}^\top(C_{n_c}) \end{bmatrix}^\top \tag{2.28}$$

BJ: The Box-Jenkins structure is defined by choosing $n_a = 0$,

$$y(t) = F^{-1}(q,\theta)B(q,\theta)u(t) + D^{-1}(q,\theta)C(q,\theta)e(t) \tag{2.29}$$

with $\theta \in \mathbb{R}^{(n_b m + n_c p + n_d p + n_f p)p}$ accordingly defined.

The choice of a parametrization for transfer function models involves the selection of one of the above-listed model structures, while the model complexity is determined by the polynomials degrees. In an identification procedure, these properties are typically selected by means of the tools illustrated in Section 2.5.

2.2.2 User's Choices

The brief introduction to PEM provided in Section 2.2 highlights how their adoption needs to be accompanied by some user's choices which are outlined in the following.

Model Class Selection. As discussed in Section 2.1, this choice can be split into three decisions. For what regards the type of models, the previous discussion already mentioned that Prediction Error approaches are commonly used to estimate transfer function models. Concerning the choice of the model class complexity and of its parametrization, the reader is referred to the discussion in Section 2.5.

Choice of the criterion. The scalar-valued function $V_N(\theta, \mathcal{D}^N)$ may be chosen in multiple ways. When dealing with multi-input-multi-output (MIMO) systems, a typical choice is

$$V_N(\theta, \mathcal{D}^N) = f_V(R_N(\theta, \mathcal{D}^N)), \qquad R_N(\theta, \mathcal{D}^N) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t,\theta)\varepsilon^\top(t,\theta) \tag{2.30}$$

with $R_N(\theta, \mathcal{D}^N)$ being the sample covariance matrix of $\varepsilon(t,\theta)$ and $f_V(\cdot)$ a monotonically increasing scalar-valued function defined on the set of positive definite matrices. The choice $f_V(R_N(\theta, \mathcal{D}^N)) = \det R_N(\theta, \mathcal{D}^N)$ guarantees optimal accuracy of the parameter estimate under weak conditions and is optimal for Gaussian distributed disturbances. An alternative definition of $f_V(\cdot)$ exploits a positive definite weighting matrix S, namely $f_V(R_N(\theta, \mathcal{D}^N)) = \mathrm{Tr}[SR_N(\theta, \mathcal{D}^N)]$: despite providing computational advantages when on-line identification is performed, this formulation of $f_V(\cdot)$ gives optimal accuracy of the parameter estimate only if $S = \Sigma^{-1}$; however, since the true value of the noise variance Σ is unknown, optimality is never guaranteed.
It has been shown (Caines (1978)) that for multivariable systems, in case the true system does not belong to the chosen model class, the loss function $f_V(\cdot)$ highly influences the properties of the estimated model, even when in the asymptotic regime (i.e. for $N \to \infty$).

A more general formulation of $V_N(\theta, \mathcal{D}^N)$ is given by

$$V_N(\theta, \mathcal{D}^N) = \frac{1}{N}\sum_{t=1}^{N} \ell(t, \theta, \varepsilon(t,\theta)), \qquad \ell : \mathbb{R} \times D_\theta \times \mathbb{R}^p \to \mathbb{R} \tag{2.31}$$

with $\ell(t, \theta, \cdot)$ being typically a norm function. The dependence of $\ell(\cdot,\cdot,\cdot)$ on t may be exploited when dealing with time-varying systems, when old data are considered less relevant w.r.t. more recent ones. In these cases, it is common practice to shape the function $\ell(\cdot,\cdot,\cdot)$ in order to give more weight to more reliable data. Furthermore, by a suitable choice of $\ell(\cdot,\cdot,\cdot)$ in (2.31), the estimation criterion can be made robust to outliers.

2.2.3 Connection with Maximum Likelihood Estimation

The success of Prediction Error Methods in the system identification field is partially due to their strict relationship with Maximum Likelihood estimation approaches, which estimate the parameter vector θ by maximizing the maximum likelihood, i.e. the probability distribution function of the observations conditioned on θ. The connection with PEM becomes clear when considering the prediction model (2.13), which generates the measured output data as

$$y(t) = w(t, \mathcal{D}^{t-1}; \theta) + \varepsilon(t, \theta), \qquad p_e(\cdot, t; \theta), \text{ PDF of } \varepsilon(t, \theta) \tag{2.32}$$

Given the dataset $\mathcal{D}^N = \{y(t), u(t)\}_{t=1}^N$ with $u^N = \{u(1), ..., u(N)\}$ being a deterministic sequence, the likelihood function for y^N (given u^N) is defined as

$$p_y(y^N; \theta) = \prod_{t=1}^{N} p_e(y(t) - w(t, \mathcal{D}^{t-1}; \theta),\ t;\ \theta) = \prod_{t=1}^{N} p_e(\varepsilon(t, \theta), t; \theta) \tag{2.33}$$

The maximum likelihood estimator (MLE) is computed as

$$\begin{aligned} \hat{\theta}_{ML}(y^N) &:= \arg\max_{\theta \in D_\theta}\ p_y(y^N; \theta) \\ &\equiv \arg\min_{\theta \in D_\theta}\ \frac{1}{N} \sum_{t=1}^{N} (-\ln p_e(\varepsilon(t, \theta), t; \theta)) \\ &= \arg\min_{\theta \in D_\theta}\ \frac{1}{N} \sum_{t=1}^{N} \ell(t, \theta, \varepsilon(t, \theta)) \end{aligned} \tag{2.34}$$

where the second equation has been derived by taking the negative logarithm of $p_y(y^N; \theta)$ and dividing by N, while the last one exploits the definition

$$\ell(t, \theta, \varepsilon(t, \theta)) = -\ln p_e(\varepsilon(t, \theta), t; \theta) \tag{2.35}$$

The loss function appearing in (2.34) coincides with the general formulation of $V_N(\theta, \mathcal{D}^N)$ given in (2.31), thus showing the equivalence between the MLE and the PE estimate if $\ell(t, \theta, \varepsilon(t, \theta))$ is chosen as in (2.35).
Further assuming that $p_e(\cdot, t; \theta)$ in (2.13) is normally distributed, namely

$$p_e(\cdot, \theta) = \mathcal{N}(0, \Sigma(\theta)\delta_{t,s}), \qquad \Sigma(\theta) \in \mathbb{R}^{p \times p} \tag{2.36}$$

and that $\Sigma(\theta)$ is independently parametrized w.r.t. the predictor's parameters (i.e. $\Sigma(\theta) = \Sigma$), the Maximum Likelihood estimator of θ is obtained by minimizing the loss

(2.30) with $f_V(R_N(\theta, \mathcal{D}^N)) = \det\, R_N(\theta, \mathcal{D}^N)$ (Söderström and Stoica (1989), Sec. 7.4).

2.2.4 Algorithmic Details

This section intends to provide an overview of the computational approaches which are commonly adopted to solve the optimization problem (2.14) arising in PEM. Since the literature on the topic is extensive, the interested reader is referred to the classical textbooks (Ljung (1999), Ch. 10 and Söderström and Stoica (1989), Sec. 7.6) for a more detailed summary.

The model class selection mentioned in Section 2.2 does not only influence the goodness of the final estimated model but also determines the complexity of the algorithmic procedure that has to be used to solve the problem (2.14). A first obvious observation is that the choice of a complex system leads to a large number of parameters to be estimated, thus enlarging the search space of problem (2.14). A second consideration regards the selected parametrization: for some of the model structures listed in Section 2.2.1, the predictor $\hat{y}(t|\theta)$ in (2.11) depends linearly on θ, thus giving rise to a linear regression model:

$$\hat{y}(t|\theta) = \varphi^\top(t)\theta \tag{2.37}$$

In particular, equation (2.37) holds for FIR and ARX model structures with $\varphi(t)$ respectively depending on past input data and on past input and output data. In this case, if the function $\ell(t, \theta, \cdot)$ in (2.31) is a quadratic norm, the Prediction Error estimate can be computed using the Least-Squares (LS) method (Lawson and Hanson, 1995; Aström, 1968; Hsia, 1977).

Whenever problem (2.14) can't be solved analytically, numerical iterative routines have to be adopted. Starting from an initial estimate $\hat{\theta}_N^{(0)}$, these routines iteratively update it according to the general rule

$$\hat{\theta}_N^{(i+1)} = \hat{\theta}_N^{(i)} - \alpha_N^{(i)} \left[H_N^{(i)}\right]^{-1} \left[V_N'(\hat{\theta}_N^{(i)}, \mathcal{D}^N)\right]^\top \tag{2.38}$$

where $V_N'(\theta, \mathcal{D}^N)$ denotes the gradient of the loss function $V_N(\theta, \mathcal{D}^N)$ in (2.31),

$$V_N'(\theta, \mathcal{D}^N) = -\frac{1}{N}\sum_{t=1}^{N}\left\{\frac{\partial}{\partial \varepsilon}\, \ell(t, \theta, \varepsilon(t,\theta))\psi^\top(t,\theta) - \frac{\partial}{\partial \theta}\, \ell(t, \theta, \varepsilon(t,\theta))\right\} \tag{2.39}$$

$$\psi(t,\theta) := -\left(\frac{d}{d\theta}\varepsilon(t,\theta)\right)^\top = \left(\frac{d}{d\theta}\hat{y}(t|\theta)\right)^\top \in \mathbb{R}^{d_\theta \times p} \tag{2.40}$$

while $\alpha_N^{(i)} \in \mathbb{R}$ is the step-size chosen so that

$$V_N(\hat{\theta}_N^{(i+1)}, \mathcal{D}^N) < V_N(\hat{\theta}_N^{(i)}, \mathcal{D}^N) \tag{2.41}$$

The matrix $H_N^{(i)} \in \mathbb{R}^{d_\theta \times d_\theta}$ is selected in order to modify the search direction; when a quadratic loss is adopted, the optimal choice for $R_N^{(i)}$ would be

$$H_N^{(i)} = V_N''(\hat{\theta}_N^{(i)}, \mathcal{D}^N) \tag{2.42}$$

with $V_N''(\theta, \mathcal{D}^N) \in \mathbb{R}^{d_\theta \times d_\theta}$ denoting the Hessian of $V_N(\theta, \mathcal{D}^N)$. Setting $H_N^{(i)}$ as in (2.42) corresponds to the *Netwon algorithm*. However, since the computation of $V_N''(\hat{\theta}_N^{(i)}, \mathcal{D}^N)$ may be prohibitive, approximations of the Hessian are typically adopted, giving rise to the so-called *quasi-Newton methods*. Among them, when a quadratic loss as (2.30) is adopted, one of the most common approximations is

$$V_N''(\theta, \mathcal{D}^N) \approx \frac{2}{N} \sum_{t=1}^{N} \psi(t,\theta) F_V \psi^\top(t,\theta) =: \Delta_N(\theta), \qquad F_V = \frac{\partial f_V(Q)}{\partial Q}\Big|_{Q=\Sigma} \tag{2.43}$$

The choice $H_N^{(i)} = \Delta_N(\hat{\theta}_N^{(i)})$ in (2.38) leads to the so-called *Gauss-Newton algorithm*, which is guaranteed to converge to a stationary point, thanks to the positive semidefiniteness of $\Delta_N(\hat{\theta}_N^{(i)})$.

The family of quasi-Newton algorithms, as well as the one of iterative search routines, is huge and a detailed treatment of these methods is certainly out of the scope of this thesis. To gain further insights on these topics, the reader is referred to the textbooks Nocedal and Wright (2006); Bertsekas (2014); Dennis Jr and Schnabel (1996).

Before proceeding, it should be observed that the computational effort of the above illustrated search methods when applied to system identification problems strictly depends on the chosen model class. In particular, this selection reflects on the amount of computations required for computing the gradient $V_N'(\theta, \mathcal{D}^N)$ and, specifically, the quantity $\psi(t, \theta)$. Ljung (1999) (Sec. 10.3) and Söderström and Stoica (1989) (Sec. 7.6) provide some examples of gradient evaluations; see also Hill (1985) and Van Zee and Bosgra (1982).

Another remark regards the solutions returned by iterative optimization methods: when adopted to solve the general problem (2.14), they are only guaranteed to converge to a local minimum. Even if the goodness of local minima may be assessed in the successive validation phase, the initialization plays a crucial role for the success of these search routines. In system identification applications, the a-priori physical knowledge may be exploited to derive good initializations. When such information is not available, a model fitted through a LS procedure or through the subspace method of Section 2.3 (which exploits more robust numerical routines) could be valid alternatives. The latter approach is actually implemented in the MATLAB System Identification Toolbox.

Some results regarding the presence of local minima in the asymptotic loss function (for $N \to \infty$) are provided in Ljung (1999) (Sec. 10.5) and in Söderström and Stoica (1989) (Sec. 12.8).

The system identification community has also considered some alternatives to the iterative optimization routines previously mentioned. Clarke (1967) and Goodwin and Payne (1977) proposed the so-called *generalized LS (GLS)*, which decomposes the non-linear optimization problem (2.14) arising when an ARARX (Ljung (1999), Sec. 4.2) model structure is chosen into a sequence of LS problems. The approach was later extended to general model structures by Söderström, Stoica, and Friedlander (1991), who introduced the so-called *indirect PEM*.

Solbrand, Ahlén, and Ljung (1985) and Ljung and Söderström (1983) (Sec. 7.2) proposed to solve the PEM problem by using off-line recursive techniques, which are more suited for on-line estimation (see Section 5.1 for more details on these methods). When applied off-line, recursive algorithms have to be run over the data multiple times: in this case they are guaranteed to have the same convergence properties of the iterative procedures in (2.38).

2.3 Subspace Methods

Starting from the beginning of the Nineties, subspace algorithms have managed to overcome some well-known shortcomings of Prediction Error Methods. Thanks to the estimation of state-space models and to the use of robust numerical routines, subspace procedures have constituted a sound alternative to PEM especially for the identification of MIMO systems, where the use of numerical optimization algorithms had often proved to be unreliable. Specifically, subspace methods estimate state-space models in a non-iterative way by resorting to standard linear algebra tools, such as matrix decompositions (SVD and QR) or the resolution of LS problems.

More details on the origins and the development of subspace approaches will be provided in Section 2.6.3.

Before proceeding with the description of subspace methods, the class of state-space models is briefly introduced in Section 2.3.1. Section 2.3.2 details the implementation of subspace algorithms, while related user's choices are discussed in Section 2.3.3. Finally, algorithmic details are provided in Section 2.3.4.

2.3.1 State-Space Models

Despite Section 2.2.1 has introduced multi-variable transfer function models, they are more commonly adopted to describe SISO systems. Indeed, the models illustrated in Section 2.2.1 contain an impulse response description for each input-output channel, thus not allowing to account for joint effects between different input-output channels. For this reason, state-space models are typically preferred to transfer function ones when MIMO systems have to be characterized. Furthermore, recalling that most optimal controllers are computed in terms of state-space models, this representation appears convenient also for controller design.

The adoption of state-space models may also be dictated by the availability of some a-priori physical knowledge about the system to be identified. Recalling that physical laws are expressed in terms of differential equations, one can collect the variables involved in such equations into a state vector $x(t) \in \mathbb{R}^n$ and discretize them, obtaining a representation of the type (assuming a sampling period equal to 1):

$$x(t+1) = A(\theta)x(t) + B(\theta)u(t), \qquad A(\theta) \in \mathbb{R}^{n\times n},\ B(\theta) \in \mathbb{R}^{n\times m} \tag{2.44}$$

Here the parameter vector θ may contain some unknown physical coefficients or simply the elements of the matrices $A(\theta)$ and $B(\theta)$. It is clear that the parametrization, i.e. the way in which θ enters the matrices $A(\theta)$ and $B(\theta)$ is not trivial as for transfer function models but may be dictated by specific properties of the system to be identified. Canonical parametrizations are a usual choice: for a n-th order system with m inputs and p outputs, they require $n(2p + m) + mp$ free parameters. Another possibility is to include parameters with an immediate physical interpretation, building so-called *gray-box* models.

Assuming that the noise-free measurements obtained from the system are given by linear combinations of the state and the input vectors, namely:

$$y(t) = C(\theta)x(t) + D(\theta)u(t) \tag{2.45}$$

an input-output description is derived in terms of the transfer function $G(q, \theta)$ as

$$y(t) = G(q, \theta)u(t) \tag{2.46}$$

$$G(q, \theta) = C(\theta)[qI_n - A(\theta)]^{-1}B(\theta) + D(\theta) \tag{2.47}$$

In the case of state-space models, a widespread convention is to split the additive output disturbance $v(t) \in \mathbb{R}^p$ into the measurement noise $\nu(t) \in \mathbb{R}^p$ (acting on the outputs) and the process noise $w(t) \in \mathbb{R}^n$ (acting on the states), leading to the following general state-space model:

$$\begin{aligned} x(t+1) &= A(\theta)x(t) + B(\theta)u(t) + w(t) \\ y(t) &= C(\theta)x(t) + D(\theta)u(t) + \nu(t) \end{aligned} \tag{2.48}$$

Furthermore, $\{w(t)\}$ and $\{\nu(t)\}$ are assumed to be white noise sequences with zero-mean and covariances

$$\mathbb{E}\left[\begin{bmatrix} w(t) \\ \nu(t) \end{bmatrix} \begin{bmatrix} w(s) \\ \nu(s) \end{bmatrix}^\top\right] = \begin{bmatrix} R_{ww}(\theta) & R_{w\nu}(\theta) \\ R_{w\nu}^\top(\theta) & R_{\nu\nu}(\theta) \end{bmatrix} \delta_{t,s} \tag{2.49}$$

It is well-known from classical system theory that the description (2.48) is not unique, but different realizations (leading to the same transfer function (2.48)) can be derived by means of similarity transforms. Among the possible realizations, the one using the lowest number n of states is called minimal. Correspondingly, the block Hankel matrix built with the impulse response coefficients $\{g(k)\}_{k=1}^\infty$

$$\mathbf{G} = \begin{bmatrix} g(1) & g(2) & \cdots & g(n) \\ g(2) & g(3) & \cdots & g(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ g(n) & g(n+1) & \cdots & g(2n-1) \end{bmatrix} \tag{2.50}$$

has rank equal to the order n of the system (also referred to as the Mc Millan degree) (Brockett, 1970; Kailath, 1980).

Equations (2.48) define the so-called *process form* of a stochastic linear system; an equivalent representation is provided by the so-called *innovation form*

$$\begin{aligned} x(t+1) &= A(\theta)x(t) + B(\theta)u(t) + K(\theta)e(t) \\ y(t) &= C(\theta)x(t) + D(\theta)u(t) + e(t) \end{aligned} \tag{2.51}$$

where $K(\theta) \in \mathbb{R}^{n\times p}$ is the steady state Kalman gain, while $\{e(t)\}$ is the innovation process, i.e. a white noise process independent of past input and output data, with second order moment $\mathbb{E}[e(t)e^\top(s)] = \Sigma\delta_{t,s}$. From (2.51), the general description (2.9) is readily derived with

$$G(q,\theta) = C(\theta)[qI_n - A(\theta)]^{-1}B(\theta) + D(\theta) \tag{2.52}$$

$$H(q,\theta) = C(\theta)[qI_n - A(\theta)]^{-1}K(\theta) + I_p \tag{2.53}$$

2.3.2 Subspace Methods in Practice

Given a set of input-output data $\mathcal{D}^N$, subspace algorithms return an estimate of the system matrices (A, B, C, D) up to within a similarity transform; additionally, also the covariance matrices R_{ww}, $R_{w\nu}$ and $R_{\nu\nu}$ are estimated. A key property of subspace approaches is that no parametrization is required, meaning that all the elements of the system matrices are directly estimated. Hence, for these techniques $d_\theta = \dim \theta = n(2n + m + 2p) + p(m + p)$ and

$$\theta = \begin{bmatrix} \mathrm{vec}^\top(A) & \mathrm{vec}^\top(B) & \mathrm{vec}^\top(C) & \mathrm{vec}^\top(D) & \mathrm{vec}^\top(R_{ww}) & \mathrm{vec}^\top(R_{\nu\nu}) & \mathrm{vec}^\top(R_{w\nu}) \end{bmatrix}^\top$$

The adoption of this trivial parametrization is made possible by the use of numerically reliable routines, which do not perform a nonlinear search on the space in which θ lies Viberg (1995).

Subspace methods basically consist of two steps. First, the given input-output data are exploited to retrieve a characteristic subspace, which coincides with the column space of the extended observability matrix O_i $(i > n)$

$$O_i := \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{bmatrix} \tag{2.54}$$

This range space has dimension n (the order of the system) and is often referred to as the *signal subspace*, because of its strict connection with the space adopted in sensor array signal processing (Schmidt, 1981; Viberg and Ottersten, 1991). Once the signal subspace is reconstructed, the second stage of any subspace algorithm consists in the estimation of the system matrices.

The most common procedures proposed in the literature to accomplish the first step have been unified under a common framework in the classical work Van Overschee and De Moor (1995b). The authors observe that the retrieval of the characteristic subspace is performed through an oblique projection, followed by a weighted complexity reduction step. A different choice of these weightings is basically what distinguishes the most famous subspace algorithms. Viberg, Wahlberg, and Ottersten (1997) provides a new interpretation of this first step, showing that the *signal subspace* can be retrieved by means of so-called *instrumental variables.*

Multiple procedures have been proposed to compute the system matrices starting from the estimated extended observability matrix. Some algorithms (Verhaegen, 1993b, 1994) determine A and C by exploiting the so-called *shift-invariance* structure of O_i and estimate the remaining matrices from A and C; alternatively, a so-called *state approach* is followed (Larimore, 1990; Van Overschee and De Moor, 1994), where two state sequences are derived from the extended observability matrix and used to compute the system matrices in a subsequent LS problem (involving also the original input-output data). A third technique, the so-called *subspace fitting* relies on a parametric model of the null-space of O_i to optimally estimate the matrix A; it was introduced by Swindlehust, Roy, Ottersten, and Kailath (1995) and subsequently developed Ottersten, Sensorer, Ottersten, Viberg, et al. (1994); Viberg et al. (1997).

The following description of subspace algorithms is split according to the two aforementioned steps.

2.3.2.1 Estimation of the Signal Subspace

Before proceeding, the vector $Y_r(t) \in \mathbb{R}^{pr}$ of stacked output values is introduced

$$Y_r(t) = \begin{bmatrix} y^\top(t) & y^\top(t+1) & \cdots & y^\top(t+r-1) \end{bmatrix}^\top \tag{2.55}$$

Analogously, the vectors $U_r(t)$, $W_r(t)$ and $\mathrm{N}_r(t)$ are defined by respectively stacking inputs, process and measurement noises. The basic equation which is exploited by subspace methods is easily derived from the state-space description (2.48):

$$Y_r(t) = O_r x(t) + S_r U_r(t) + V_r(t) \tag{2.56}$$

where O_r was defined in (2.54),

$$V_r(t) := \Omega_r W_r(t) + \mathrm{N}_r(t) \tag{2.57}$$

and

$$S_r = \begin{bmatrix} D & 0_{p\times m} & \cdots & 0_{p\times m} & 0_{p\times m} \\ CB & D & \cdots & 0_{p\times m} & 0_{p\times m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{r-2}B & CA^{r-3}B & \cdots & CB & D \end{bmatrix} \tag{2.58}$$

$$\Omega_r = \begin{bmatrix} 0_{p\times n} & 0_{p\times n} & \cdots & 0_{p\times n} & 0_{p\times n} \\ C & 0_{p\times n} & \cdots & 0_{p\times n} & 0_{p\times n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{r-2} & CA^{r-3} & \cdots & C & 0_{p\times n} \end{bmatrix} \tag{2.59}$$

Assuming that the data $\mathcal{D}^N = \{u(t), y(t)\}_{t=1}^N$ are available, equation (2.56) can be rewritten in order to include the whole dataset $\mathcal{D}^N$:

$$\mathbf{Y} = O_r X + S_r \mathbf{U} + \mathbf{V} \tag{2.60}$$

where

$$\begin{aligned} X &:= \begin{bmatrix} x(1) & x(2) & \cdots & x(N) \end{bmatrix} \\ \mathbf{Y} &:= \begin{bmatrix} Y_r(1) & Y_r(2) & \cdots & Y_r(N) \end{bmatrix} \\ \mathbf{U} &:= \begin{bmatrix} U_r(1) & U_r(2) & \cdots & U_r(N) \end{bmatrix} \\ \mathbf{V} &:= \begin{bmatrix} V_r(1) & V_r(2) & \cdots & V_r(N) \end{bmatrix} \end{aligned} \tag{2.61}$$

Subspace methods exploit algebraic properties to estimate the column space of O_r from equation (2.60). This procedure will be first outlined according to the unified framework proposed in Van Overschee and De Moor (1995b, 2012). In a second stage, the description will be based on the so-called *instrumental variable* interpretation provided in Viberg et al. (1997) and recalled in Ljung (1999) (Sec. 10.6).

Unifying Framework. According to the approach introduced in Van Overschee and De Moor (1995b), the first goal is to determine the optimal linear prediction of future outputs $\mathbf{Y}$ based on all the information contained in the available data, namely using past input and output data and future input values (contained in the matrix $\mathbf{U}$). To this purpose, the following matrices need to be defined

$$U_s^-(t) := \begin{bmatrix} u^\top(t-s) & \cdots & u^\top(t-2) & u^\top(t-1) \end{bmatrix}^\top \tag{2.62}$$

$$Y_s^-(t) := \begin{bmatrix} y^\top(t-s) & \cdots & y^\top(t-2) & y^\top(t-1) \end{bmatrix}^\top \tag{2.63}$$

with the corresponding block Hankel matrices

$$\mathbf{U}^- := \begin{bmatrix} U_s^-(1) & U_s^-(2) & \cdots & U_s^-(N) \end{bmatrix} \tag{2.64}$$

$$\mathbf{Y}^- := \begin{bmatrix} Y_s^-(1) & Y_s^-(2) & \cdots & Y_s^-(N) \end{bmatrix} \tag{2.65}$$

By collecting the past information into the matrix

$$\Phi := \begin{bmatrix} \mathbf{U}^- \\ \hline \mathbf{Y}^- \end{bmatrix} \tag{2.66}$$

the prediction problem can be formally stated as

$$(\widehat{L}_p, \widehat{L}_u) = \underset{L_p \in \mathbb{R}^{pr \times (p+m)s},\ L_u \in \mathbb{R}^{pr \times mr}}{\arg\min} \left\| \mathbf{Y} - \begin{bmatrix} L_p & L_u \end{bmatrix} \begin{bmatrix} \Phi \\ \mathbf{U} \end{bmatrix} \right\|_F^2 \tag{2.67}$$

where $\| \cdot \|_F$ denotes the Frobenius norm. The following derivation is based on the assumptions:

1. The process noise $\{w(t)\}$ and the measurement noise $\{\nu(t)\}$ are not identically zero.

2. The input $\{u(t)\}$ is uncorrelated with the process noise $\{w(t)\}$ and the measurement noise $\{\nu(t)\}$.

3. The input $\{u(t)\}$ is persistently exciting of order $r + s$.

4. An infinite number of measurements are available, i.e. $N \to \infty$.

According to the previous assumptions, it turns out that the optimal prediction of future outputs $\widehat{\mathbf{Y}}$ is the orthogonal projection of $\mathbf{Y}$ onto the combined row spaces of Φ and $\mathbf{U}$, which is equal to (Van Overschee and De Moor (2012), Th. 11)

$$\widehat{\mathbf{Y}} := \widehat{L}_p \Phi + \widehat{L}_u \mathbf{U} = O_r \widehat{X} + S_r \mathbf{U} \tag{2.68}$$

$$= O_r(\Delta_X \widehat{X}_0 + \Delta_\Phi \Phi) + S_r \mathbf{U} \tag{2.69}$$

with

$$\widehat{X} := \begin{bmatrix} \hat{x}(1) & \hat{x}(2) & \cdots & \hat{x}(N) \end{bmatrix} \tag{2.70}$$

Each column $\hat{x}(i)$ of $\widehat{X}$ is the output of a non-steady-state Kalman filter built from the system matrices, while $\widehat{X}_0$ contains the sequence of initial states. Δ_X and Δ_Φ are suitable matrices depending on the system matrices (their definition can be found in Van Overschee and De Moor (2012) , A.7). Define the vector

$$X^- = \begin{bmatrix} x(1-s) & x(2-s) & \cdots & x(N-s) \end{bmatrix} \tag{2.71}$$

and set $\widehat{X}_0$ in (2.69) equal to the orthogonal projection of X^- onto the combined row spaces of $\mathbf{U}^-$ and $\mathbf{U}$, then it follows from (2.68) that

$$\widehat{L}_p \Phi = O_r \widetilde{X} \tag{2.72}$$

Last equation shows that the optimal output prediction based only on past input and output data is given by the product of the extended observability matrix with the vector $\widetilde{X}$, which contains the Kalman filter sequence initialized with the oblique projection of X^- onto $\mathbf{U}^-$ along $\mathbf{U}$. Since $\widetilde{X}$ depends on the unknown system matrices, equation (2.72) can't be computed as stated. However, Van Overschee and De Moor (1995b) proved that $O_r\widetilde{X}$ equals the oblique projection of the row space of $\mathbf{Y}$ along the row space of $\mathbf{U}$ on the row space of Φ; namely

$$O_r \widetilde{X} = \mathbf{Y}^{\mathbf{U}}_{\Phi} \tag{2.73}$$

The quantity $\mathbf{Y}^{\mathbf{U}}_{\Phi}$ can be computed from the given input-output data $\mathcal{D}^N$ as

$$\mathbf{Y}^{\mathbf{U}}_{\Phi} = \mathbf{Y}\Pi^{\perp}_{\mathbf{U}^\top}\Phi^\top(\Phi\Pi^{\perp}_{\mathbf{U}^\top}\Phi^\top)^{-1}\Phi, \qquad \mathbf{Y}^{\mathbf{U}}_{\Phi} \in \mathbb{R}^{pr\times N} \tag{2.74}$$

where $\Pi^{\perp}_{\mathbf{U}^\top}$ denotes the orthogonal projection matrix onto the null-space of $\mathbf{U}$:

$$\Pi^{\perp}_{\mathbf{U}^\top} = I_N - \mathbf{U}^\top(\mathbf{U}\mathbf{U}^\top)^{-1}\mathbf{U} \tag{2.75}$$

Equation (2.73) proves that the row space of $\widetilde{X}$ equals the row space of $\mathbf{Y}^{\mathbf{U}}_{\Phi}$; analogously, the column space of the extended observability matrix O_r equals the column space of $\mathbf{Y}^{\mathbf{U}}_{\Phi}$. Therefore, the so-called *signal subspace* can be reconstructed by computing $\mathbf{Y}^{\mathbf{U}}_{\Phi}$. Recalling that this subspace has dimension n and that the rows of $\mathbf{Y}^{\mathbf{U}}_{\Phi}$ span a pr-dimensional space, a reduction step could be performed in order to reduce this subspace dimension to n. In turn, this will allow to reduce the amount of information of the "past" that has to be considered in order to optimally predict the "future". Formally, the complexity reduction step can be formulated as

$$\begin{aligned} \widehat{R} = &\operatorname*{arg\,min}_{R\in\mathbb{R}^{pr\times N}} \|W_1(\mathbf{Y}^{\mathbf{U}}_{\Phi} - R)W_2\|^2_F \\ &\text{s.t. } \operatorname{rank}(R) = n \end{aligned} \tag{2.76}$$

where the weighting matrices $W_1 \in \mathbb{R}^{rp\times rp}$ and $W_2 \in \mathbb{R}^{N\times\alpha}$ determine which part of the information contained in $\mathbf{Y}^{\mathbf{U}}_{\Phi}$ has to be retained. Even if W_1, W_2 and the number of columns α of W_2 are chosen by the user, they have to guarantee that

$$\mathrm{rank}(W_1 \mathbf{Y}_{\Phi}^{\mathbf{U}} W_2) = \mathrm{rank}(\mathbf{Y}_{\Phi}^{\mathbf{U}}) \tag{2.77}$$

Specifically, W_1 has to be of full rank, while W_2 must guarantee that $\mathrm{rank}(\Phi) = \mathrm{rank}(\Phi W_2)$. As will be detailed in Section 2.3.3, specific choices of these matrices give rise to the different subspace algorithms proposed in the literature (Van Overschee and De Moor, 1995b). The solution to problem (2.76) can be computed by properly partitioning the SVD of $W_1 \mathbf{Y}_{\Phi}^{\mathbf{U}} W_2$:

$$W_1 \mathbf{Y}_{\Phi}^{\mathbf{U}} W_2 = QDP^\top = \begin{bmatrix} Q_s & Q_n \end{bmatrix} \begin{bmatrix} D_s & 0 \\ 0 & D_n \end{bmatrix} \begin{bmatrix} P_s^\top \\ P_n^\top \end{bmatrix} \tag{2.78}$$

Retaining in D_s the n largest singular values of $W_1 \mathbf{Y}_{\Phi}^{\mathbf{U}} W_2$ and in Q_s the corresponding singular vectors, it follows that

$$\widehat{R} = W_1^{-1} Q_s D_s P_s^\top W_2^\dagger \tag{2.79}$$

If assumption 4 above is satisfied, $\widehat{R} = \mathbf{Y}_{\Phi}^{\mathbf{U}}$, since $\mathbf{Y}_{\Phi}^{\mathbf{U}}$ is exactly of rank n and has only n non-zero singular values (meaning that $D_n = 0$). However, when only a finite number of data is available, the singular values of $W_1 \mathbf{Y}_{\Phi}^{\mathbf{U}} W_2$ are all different from zero and order n has to be selected by the user according to one of the procedures discussed in Section 2.3.3 and 2.5.

Moreover, according to (2.73), the extended observability matrix O_r and the Kalman filter sequence $\widetilde{X}$ can be estimated as

$$\widehat{O}_r = W_1^{-1} Q_s \Gamma \tag{2.80}$$

$$\widehat{\widetilde{X}} = \begin{bmatrix} \hat{\tilde{x}}(1) & \hat{\tilde{x}}(2) & \cdots & \hat{\tilde{x}}(N) \end{bmatrix} = \widehat{O}_r^\dagger \, \mathbf{Y}_{\Phi}^{\mathbf{U}} \tag{2.81}$$

where $\Gamma \in \mathbb{R}^{n \times n}$ is an arbitrary invertible matrix which determines the coordinate basis of the estimated state-space representation. Furthermore, the part of the Kalman state sequence $\widehat{\widetilde{X}}$ which lies on the range of W_2 can be recovered as

$$\widehat{\widetilde{X}} W_2 = \Gamma^{-1} P_s^\top \tag{2.82}$$

"Instrumental Variables" Perspective. Recalling that the objective is to estimate the range space of O_r, the idea is to adopt so-called *instrumental variables* to eliminate the influence of the input and noise matrices in equation (2.60), thus retrieving the column space of O_r from that of $\mathbf{Y}$.

Remark 2.3.1. It should be mentioned that the use of instrumental variables is very popular in system identification, especially in connection with Prediction Error Methods. The reader is referred to e.g. Söderström and Stoica (1983) for an extensive treatment. The term “ instrumental variables” was first associated to subspace approaches by Aoki (1990); De Moor, Van Overschee, and Suykens (1991); Verhaegen (1991).

To eliminate the effect of the inputs, the original subspace methods (also called *direct subspace*, De Moor, Vandewalle, Moonen, Van Mieghem, and Vandenberghe (1988); Verhaegen (1991)) right-multiply equation (2.60) by $\Pi^{\perp}_{\mathbf{U}^{\top}}$, the orthogonal projection matrix onto the null-space of $\mathbf{U}$ (defined in (2.75))

$$\mathbf{Y}\Pi^{\perp}_{\mathbf{U}^{\top}} = O_r X \Pi^{\perp}_{\mathbf{U}^{\top}} + \mathbf{V}\Pi^{\perp}_{\mathbf{U}^{\top}} \tag{2.83}$$

Neglecting the noise term (i.e. supposing $\mathbf{V} = 0_{pr\times N}$) and assuming that the product $X\Pi^{\perp}_{\mathbf{U}^{\top}}$ has full rank n or, equivalently that,

$$\operatorname{rank}\begin{bmatrix} X \\ \mathbf{U} \end{bmatrix} = n + \operatorname{rank}(\mathbf{U}) \tag{2.84}$$

the column space of O_r is spanned by $\mathbf{Y}\Pi^{\perp}_{\mathbf{U}^{\top}}$, i.e.

$$\operatorname{range}(O_r) = \operatorname{range}(\mathbf{Y}\Pi^{\perp}_{\mathbf{U}^{\top}}) \tag{2.85}$$

In presence of noise ($\mathbf{V} \neq 0_{pr\times N}$), equation (2.85) holds only approximately and the range space of the extended observability matrix can be reconstructed by choosing a large value of r (the number of block rows in the matrix $\mathbf{Y}$) and by performing the SVD of $\mathbf{Y}\Pi^{\perp}_{\mathbf{U}^{\top}}$ and retaining only the first n singular vectors. However, this procedure has been proved to be consistent only if the noise sequence contained in $V_r(t)$ is white Verhaegen (1993b); Viberg, Ottersten, Wahlberg, and Ljung (1991).

To account for coloured noise an additional instrument matrix has to be adopted in order to decorrelate out the noise term $\mathbf{V}$. Let $\Psi \in \mathbb{R}^{j\times N}$ ($j \geq N$) denote such matrix and multiply equation (2.83) from the right by $\Psi^{\top}$:

$$\frac{1}{N}\mathbf{Y}\Pi^{\perp}_{\mathbf{U}^{\top}}\Psi^{\top} = O_r \frac{1}{N} X\Pi^{\perp}_{\mathbf{U}^{\top}}\Psi^{\top} + \frac{1}{N}\mathbf{V}\Pi^{\perp}_{\mathbf{U}^{\top}}\Psi^{\top} \tag{2.86}$$

A normalization by N has also been introduced in (2.86). The matrix Ψ has to be chosen in order to satisfy the two following asymptotic conditions:

$$\lim_{N\to\infty} \frac{1}{N} \mathbf{V}\Pi^{\perp}_{\mathbf{U}^\top} \Psi^\top = 0_{pr\times j} \tag{2.87}$$

$$\operatorname{rank}\left(\lim_{N\to\infty} \frac{1}{N} X\Pi^{\perp}_{\mathbf{U}^\top} \Psi^\top \right) = n \tag{2.88}$$

The second equation guarantees that the so-called *signal subspace* is not destroyed, namely that the range of $\mathbf{Y}\Pi^{\perp}_{\mathbf{U}^\top}\Psi^\top$ provides a consistent estimate of the column space of O_r. Assuming that the given input data u^N are generated in an open loop situation and that the input signal is persistently exciting of order r (see Ljung (1999), Sec. 13.2), it has been shown that the conditions (2.87)-(2.88) are satisfied by setting Ψ equal to the matrix Φ defined in (2.66) (Ottersten et al., 1994; Van Overschee and De Moor, 2012). The number s of past input and output values contained in Φ has to be chosen by the user (see Section 2.3.3 for a further discussion).
Following the approach in (2.78), a consistent estimate of the signal subspace can be obtained by computing the following SVD:

$$\frac{1}{N}\widetilde{W}_1 \mathbf{Y}\Pi^{\perp}_{\mathbf{U}^\top}\Phi^\top \widetilde{W}_2 = QDP^\top = \begin{bmatrix} Q_s & Q_n \end{bmatrix} \begin{bmatrix} D_s & 0 \\ 0 & D_n \end{bmatrix} \begin{bmatrix} P_s^\top \\ P_n^\top \end{bmatrix} \tag{2.89}$$

where the weighting matrices $\widetilde{W}_1 \in \mathbb{R}^{rp\times rp}$ and $\widetilde{W}_2 \in \mathbb{R}^{s(p+m)\times\alpha}$ play the same role of W_1 and W_2 introduced in (2.76). Collecting in D_s the n largest singular values and in Q_s the corresponding singular vectors, an estimate of the extended observability matrix is readily given by

$$\widehat{O}_r = \widetilde{W}_1^{-1} Q_s \Gamma \tag{2.90}$$

where, as before, $\Gamma \in \mathbb{R}^{n\times n}$ is an arbitrary invertible matrix fixing the basis of the state-space representation.
If condition (2.88) is satisfied, the estimate (2.90) is guaranteed to converge to the true observability matrix for some state-space realization which depends on the input sequence u^N provided in the data $\mathcal{D}^N$ (Van Overschee and De Moor, 1995b, 2012). However, as previously observed, in practice the true order n of the system is not a-priori known and the user has to choose the number of singular vectors to be retained in Q_s.

Remark 2.3.2. Comparing equations (2.78) and (2.89), it is clear that the SVD performed in the unifying framework of Van Overschee and De Moor (1995b) coincides with the one computed according to the “instrumental variables” perspective if $\widetilde{W}_1$ and $\widetilde{W}_2$ in (2.89) are chosen as:

$$\widetilde{W}_1 = W_1, \qquad \widetilde{W}_2 = \left(\frac{1}{N}\Phi\Pi^{\perp}_{\mathbf{U}^\top}\Phi^\top\right)^{-1} \Phi W_2 \tag{2.91}$$

2.3.2.2 Estimation of the System Matrices

Once the extended observability matrix O_r has been estimated, the corresponding system matrices have to be computed. Three popular approaches can be found in the literature and will be outlined in the following.

Shift Invariance. This is probably the most common procedure and is based on the so-called *shift invariance* property of the extended observability matrix O_r (Kung, 1978). If O_r in (2.54) is partitioned into r block rows $O_{r,i} \in \mathbb{R}^{p\times n}$, $i = 1, ..., r$, then it readily follows that

$$C = O_{r,1}, \qquad O_{r,i} = O_{r,i-1}A \tag{2.92}$$

Define the analogous partition $\widehat{O}_{r,i} \in \mathbb{R}^{p\times \hat{n}}$, $i = 1, ..., r$ for the estimated extended observability matrix $\widehat{O}_r$ (see (2.80) and (2.90)), with $\hat{n}$ denoting the estimated system order. C and A can be estimated as

$$\widehat{C} = \widehat{O}_{r,1}, \qquad \widehat{A} = \underset{A\in\mathbb{R}^{\hat{n}\times\hat{n}}}{\arg\min} \sum_{i=2}^{r} \|\widehat{O}_{r,i} - \widehat{O}_{r,i-1}A\|_F^2 \tag{2.93}$$

Once $\widehat{C}$ and $\widehat{A}$ have been computed, B and D can be determined using the equation (compare with (2.47))

$$y(t) = \widehat{C}(qI_{\hat{n}} - \widehat{A})^{-1}Bu(t) + Du(t) + v(t) \tag{2.94}$$

and hence the predictor

$$\begin{aligned}
\hat{y}(t|B, D, x_0) &= \widehat{C}(qI_{\hat{n}} - \widehat{A})^{-1}x_0\delta(t) + \widehat{C}(qI_{\hat{n}} - \widehat{A})^{-1}Bu(t) + Du(t) && (2.95)\\
&= \widehat{C}\widehat{A}^t x_0 + (u^\top(t) \otimes I_p)\text{vec}(D) + \left(\sum_{k=0}^{t-1} u^\top(t) \otimes (\widehat{C}\widehat{A}^{t-k-1})\right) \text{vec}(B) \\
&= \varphi^\top(t) \begin{bmatrix} x_0 \\ \text{vec}(B) \\ \text{vec}(D) \end{bmatrix} && (2.96)
\end{aligned}$$

In equation (2.95), x_0 and $\delta(t)$ respectively denote the initial state and the unit pulse at time 0, while the symbol $\otimes$ is the Kronecker product. Equation (2.96) suggests to estimate x_0 and the matrices B and D by solving the following weighted least squares problem (Van Overschee and De Moor, 2012)

$$
\begin{aligned}
(\widehat{B}, \widehat{D}) &= \arg\min_{B,D,x_0} \frac{1}{N} \sum_{t=1}^{N} \|y(t) - \hat{y}(t|B, D, x_0)\|_W^2 \\
&= \arg\min_{B,D,x_0} \frac{1}{N} \sum_{t=1}^{N} \left\| y(t) - \varphi^\top(t) \begin{bmatrix} x_0 \\ \mathrm{vec}(B) \\ \mathrm{vec}(D) \end{bmatrix} \right\|_W^2
\end{aligned} \tag{2.97}
$$

where $\|x\|_W^2 = x^\top W x$ with W denoting a suitable weighting matrix (optimal values for W have been investigated by Chiuso and Picci (2004a)).

State Estimation. This approach is based on the reformulation of the state-space model (2.48) as a linear regression. In fact, defining

$$
Y(t) = \begin{bmatrix} x(t+1) \\ y(t) \end{bmatrix}, \quad \Theta = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \quad \varphi(t) = \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \quad E(t) = \begin{bmatrix} w(t) \\ \nu(t) \end{bmatrix} \tag{2.98}
$$

the model (2.48) can be rewritten as

$$
Y(t) = \Theta\varphi(t) + E(t) \tag{2.99}
$$

Hence, using the input-output data $\mathcal{D}^N$ and the state sequence $\widehat{\widehat{X}}$ computed in (2.81), the system matrices can be estimated solving the LS problem

$$
\widehat{\Theta} = \begin{bmatrix} \widehat{A} & \widehat{B} \\ \widehat{C} & \widehat{D} \end{bmatrix} = \arg\min_{\Theta} \sum_{t=1}^{N} \left\| \begin{bmatrix} \widehat{\widehat{x}}(t+1) \\ y(t) \end{bmatrix} - \Theta \begin{bmatrix} \widehat{\widehat{x}}(t) \\ u(t) \end{bmatrix} \right\|_F^2 \tag{2.100}
$$

The procedure here illustrated follows the approach in Larimore (1983), where $x(t+1)$ is replaced by the shifted version of $\widehat{\widehat{x}}(t)$ (in (2.81)), namely $x(t+1) = \widehat{\widehat{x}}(t+1)$. Van Overschee and De Moor (2012) (Sec. 4.4) propose two different choices of $x(t+1)$, leading to other two algorithms for the estimation of the system matrices. The reader is referred to Ljung and McKelvey (1996); Chiuso and Picci (2004a, 2005) and to Van Overschee and De Moor (2012) for further details on the proposed procedures based on a state estimate.

Subspace Fitting. This approach exploits the structure of the extended observability matrix to obtain a statistically optimal estimation of matrix $\widehat{A}$. Compared to the previous techniques which re-use the given input-output data $\mathcal{D}^N$, this method simply uses the estimated $\widehat{O}_r$ (computed in (2.80) or equivalently in (2.90)). Specifically, denoting with $O_r(\theta)$ the parametrized observability matrix of some realization, the estimated

$\widehat{O}_r$ can be rewritten as

$$\widehat{O}_r = O_r(\theta)T + E_{O_r} \tag{2.101}$$

where T represents an unknown transformation matrix and E_{O_r} is the error matrix. The *subspace-fitting* approach aims at estimating the parameters θ and the elements of T by minimizing the distance between the range spaces of $\widehat{O}_r$ and $O_r(\theta)$, namely

$$(\hat{\theta}, \widehat{T}) = \underset{\theta,\ T}{\arg\min} \|\mathrm{vec}(\widehat{O}_r - O_r(\theta)T)\|_W^2 \tag{2.102}$$

for some positive definite weighting matrix W. An asymptotic best consistent (ABC) estimate would be achieved by setting W equal to a consistent estimate of $\mathrm{Cov}(\mathrm{vec}(E_{O_r}))$. However, while (2.102) can be easily solved w.r.t. T for fixed θ, the optimal θ has to be found through a non-linear search. As shown in Ottersten et al. (1994); Viberg et al. (1997), the problem can be circumvented by estimating θ as

$$\hat{\theta} = \underset{\theta}{\arg\min} \|\mathrm{vec}(\Upsilon^\top(\theta) W_1^{-1} Q_s)\|_W^2 \tag{2.103}$$

$$W = \mathrm{Cov}(\mathrm{vec}(\Upsilon^\top(\theta) W_1^{-1} Q_s)) \tag{2.104}$$

where Υ denotes a parametrized basis for the null-space of $O_r^\top(\theta)$. Since this null-space can be linearly parametrized w.r.t. θ, problem (2.103) can be solved through a non-iterative (two-step) procedure (Viberg et al., 1997).

2.3.2.3 Estimation of the Noise Model

Once the system matrices have been estimated through one of the three techniques detailed in Section2.3.2.2, a noise model can be retrieved by first estimating the process and the measurement noises as

$$w(t) = \widehat{\tilde{x}}(t+1) - \widehat{A}\widehat{\tilde{x}}(t) - \widehat{B}u(t) \tag{2.105}$$

$$\nu(t) = y(t) - \widehat{C}\widehat{\tilde{x}}(t) - \widehat{D}u(t) \tag{2.106}$$

where $\widehat{\tilde{x}}(t)$ denotes the Kalman filter sequence computed in (2.81). The corresponding covariance matrices can then be readily estimated as

$$\widehat{R}_{ww} = \frac{1}{N-1}\sum_{t=1}^{N} w(t)w^\top(t), \qquad \widehat{R}_{\nu\nu} = \frac{1}{N-1}\sum_{t=1}^{N} \nu(t)\nu^\top(t) \tag{2.107}$$

$$\widehat{R}_{w\nu} = \frac{1}{N-1}\sum_{t=1}^{N} w(t)\nu^\top(t) \tag{2.108}$$

2.3.3 User's choices

Compared to PEM, subspace methods have always been considered less demanding not only from the computational point of view, but also w.r.t. to the choices that the user has to make. In particular, many authors have contemplated the selection of the system order $\hat{n}$ as the only decision left to the user. While this constitutes for sure the most relevant user's choice, the previous discussion highlights how the use of a subspace algorithm requires the user to take some other decisions. These will be pointed out in this section together with the corresponding recommendations that can be found in the literature. The following discussion will show how clear guidelines for most of these choices still don't exist, despite the interest that the system identification community has devoted to this topic in the last decade.

Choice of the system order $\hat{n}$. This decision represents the analogous of the model class selection for Prediction Error Methods. The introductory discussion to subspace algorithms in Section 2.3.2 has pointed out how the model type and the parametrization are implicitly selected, once subspace approaches are used. Thus, the model complexity selection appears as the only decision on the model class which is left to the user. The discussion on this choice is postponed to Section 2.5, where an overview of model class selection techniques will be presented. However, it is worth to mention here that specific approaches for the estimation of the order n have been introduced in the context of subspace methods (Bauer, 2005, 2001): most of them are based on the singular values computed in (2.78) and (2.89) and will be further mentioned in Section 2.5.

Choice of the weighting matrices W_1 and W_2. Together with the selection of the system order $\hat{n}$, the choice of W_1 and W_2 represents the most important decision for the application of a subspace algorithm. Indeed, they affect the variance and the possible bias of the estimates due to under-modelling (Jansson and Wahlberg, 1995; Van Overschee and De Moor, 1995b, 2012). In particular, it has been proved that the most common choices (which lead to the classical algorithms unified by Van Overschee and De Moor (1995b)) return the same system estimate (up to within a similarity transform) whenever the exact order n is selected and the number of available data goes to infinity (since all the algorithms are asymptotically unbiased). On the other hand, if the selected order $\hat{n}$ is smaller than the true one, the corresponding bias error is affected by the weighting matrices. Further details can be found in Van Overschee and De Moor (1995a) and in Section 4.2 of this manuscript.

As proved in Van Overschee and De Moor (1995b), the existing algorithms correspond to the following choices of W_1 and W_2:

- *N4SID* (Van Overschee and De Moor, 1994): $W_1 = I_{rp}$, $W_2 = I_{s(p+m)}$
- *MOESP* (Verhaegen, 1994): $W_1 = I_{rp}$, $W_2 = \Pi^{\perp}_{\mathbf{U}^\top}$
- *CVA* (Larimore, 1990): $W_1 = \left(\frac{1}{N}\mathbf{Y}\Pi^{\perp}_{\mathbf{U}^\top}\mathbf{Y}^\top\right)^{-1/2}$, $W_2 = \Pi^{\perp}_{\mathbf{U}^\top}$
- *IVM* (Viberg, 1995): $W_1 = \left(\frac{1}{N}\mathbf{Y}\Pi^{\perp}_{\mathbf{U}^\top}\mathbf{Y}^\top\right)^{-1/2}$, $W_2 = \Pi^{\perp}_{\mathbf{U}^\top}\Phi^\top\left(\frac{1}{N}\Phi\Phi^\top\right)^{-1/2}$

Some results have been derived on the choices above. Larimore (1994) shows that the weighting used in CVA is optimal for the estimation of the system order using a finite amount of data. Van Overschee and De Moor (1995b) investigate the selection of W_1 according to a frequency domain criterion and Van Overschee and De Moor (1995a) provide an interpretation of the choice of W_1 in line with the weighted model reduction of Enns (1985).
Further results will be reported in Section 4.2, where the optimal selection of W_1 and W_2 w.r.t. to the accuracy of the estimates is investigated.

Choice of the future horizon r. Since the value of r determines the number of block rows in the estimated observability matrix O_r, $r > n$ is required. Many algorithms set $r = s$, with s denoting the past horizon contained in the instrumental variables matrix Φ in equation (2.66) (Van Overschee and De Moor, 1994; Verhaegen, 1993b, 1994). Despite the effort that has been devoted to determine the influence of r on the accuracy of the subspace estimate, no clear conclusion has been drawn, as will be highlighted also in Section 4.2.

Choice of the past horizon s. A necessary condition for recovering the true observability matrix O_r is $s > \frac{n}{p+m}$ (Viberg, 1995). Some algorithms also adopt two different past horizons for the input and the output signals; the *OE-MOESP* of Verhaegen (1994) uses only past inputs, thus leading to the estimation of an Output-Error model. Analogously to the selection of r, no clear guideline for the value of s has been derived in the literature.

Choice of the matrix Γ. As already mentioned in the previous discussion, the value of Γ only determines the coordinate basis of the estimated state-space realization. Typical choices are $\Gamma = I_n$, $\Gamma = D_s$ or $\Gamma = D_s^{1/2}$.

Choice of the procedure to estimate the system matrices. The described techniques lead to different estimates; consequently, the analysis of the asymptotic properties of subspace estimators heavily depends on this choice, as will be clear from the overview of Section 4.2.

The interested reader is referred also to Ljung (2003), where the impact of the mentioned user's choices is investigated through numerical simulations.

2.3.4 Algorithmic Details

One of the main advantages of subspace algorithms w.r.t. PEM regards the computational complexity: thanks to the use of simple linear algebra tools (such as the computation of projections and of SVD), subspace approaches avoid the use of iterative optimization routines, thus being immune from convergence issues. In particular, the benefit w.r.t. to PEM is relevant when MIMO systems have to be estimated.

However, compared to PEM, the lack of a cost function to be minimized complicates the statistical analysis of subspace estimates, as will be clarified in Chapter 4.

From a computational point of view, the most demanding step of a subspace algorithm is the SVD of equation (2.78) or (2.89). Efficient implementations compute the SVD of a low dimensional matrix, arising after a preliminary QR decomposition of the data matrix $[\mathbf{U}^\top \; \Phi^\top \; \mathbf{Y}^\top]^\top$ (Verhaegen (1994); Verhaegen and Verdult (2007), Sec. 9.6.1).

2.4 Non-Parametric Bayesian Methods

Non-parametric Bayesian methods have been introduced into the system identification community at the beginning of the 2010s with the aim of overcoming a well-known issue affecting both PEM and subspace approaches, i.e. the requirement of model class selection. To this end, subspace algorithms only demand to choose the model complexity, while the application of PEM also involves to fix a suitable parametrization. As will be clear from the careful discussion of Section 2.5, these decisions may not only require a significant computational effort (especially when multiple models have to be estimated), but they also highly influence the quality of the returned estimators (Pillonetto and De Nicolao, 2012; Ljung, 1999). Differently from the techniques presented in Sections 2.2 and 2.3, when applying the method here illustrated, a model class selection stage is not needed, since the mathematical tool exploited for the description of the system does not contain a set of parameters, as highlighted by the adjective "non-parametric" in the name. Furthermore, model complexity is implicitly chosen during the estimation step.

The non-parametric approach here presented directly estimates the impulse responses appearing in the one-step ahead predictor defined in (2.11). Namely, recalling that it is defined as

$$\hat{y}(t) = F_u(q)u(t) + F_y(q)y(t) \tag{2.109}$$

with

$$F_u(q) = H^{-1}(q)G(q) = \sum_{k=1}^{\infty} f_u(k)q^{-k} \tag{2.110}$$

$$F_y(q) = I_p - H^{-1}(q) = \sum_{k=1}^{\infty} f_y(k)q^{-k} \tag{2.111}$$

the aim is to infer $\{f_u(k)\}_{k=1}^{\infty}$ and $\{f_y(k)\}_{k=1}^{\infty}$ as (vector-valued) functions over $\mathbb{N}$. This is accomplished by resorting to the theory of *Gaussian Process Regression (GPR)*, i.e. by treating $\{f_u(k)\}$ and $\{f_y(k)\}$ as Gaussian processes and inferring their distribution according to the available input-output data $\mathcal{D}^N$. Many authors have pointed out the relationship between the *Gaussian Process (GP)* framework and the function estimation performed through regularized kernel methods (according to the theory of Reproducing Kernel Hilbert Spaces (RKHS)) (Kimeldorf and Wahba, 1970; Wahba, 1990; Rasmussen and Williams, 2006). Following this tradition, Section 2.4.1.1 introduces how Gaussian Process Regression is applied in the context of system identification: it turns out that the resulting approach relies on the so-called Bayesian inference, thus clarifying the classification as “Bayesian” methods. In the subsequent Section 2.4.1.2, the equivalent formulation as regularized estimation in RKHS is provided, while Section 2.4.1.3 describes the practical implementation of such methods, as *Regularized Least Squares (ReLS)* techniques. To favour the understanding of such approaches, the identification of SISO systems is first considered (Section 2.4.1), while the estimation of MIMO systems is treated in a second stage (Section 2.4.2).

As a further simplification, the following description only considers the identification of Output-Error models, meaning that the noise model is neglected ($H(q) \equiv I_p$). For OE models, the predictor (2.112) becomes

$$\hat{y}(t) = F_u(q)u(t) = G(q)u(t) \tag{2.112}$$

and the impulse response $\{g(k)\}_{k=1}^{\infty}$ is directly estimated. Such simplification has been adopted in the seminal paper Pillonetto and De Nicolao (2010) and can be found in several works on non-parametric Bayesian methods for system identification. The extension of the approach here presented to the identification of complete predictor models (2.112) is straightforward. The interested reader is referred to Pillonetto, Chiuso, and De Nicolao (2011a).

2.4.1 Non-Parametric Bayesian Methods for SISO systems

This section illustrates the use of non-parametric Bayesian methods for the identification of SISO systems (namely $p = m = 1$). In this case the impulse response $g(\cdot)$ is a scalar function over $\mathbb{N}$.

Recalling the setting introduced in Section 2.1, the given input-output data $\mathcal{D}^N = \{u(t), y(t)\}_{t=1}^N$ are generated according to

$$y(t) = G(q)u(t) + H(q)e(t), \qquad e(t) \sim p(e) \tag{2.113}$$

As previously anticipated, the estimations of the noise model is not considered here, thus postulating $H(q) \equiv 1$. Therefore, by introducing the functional $\mathcal{L}_t[g]$ over functions $g : \mathbb{N} \to \mathbb{R}$

$$\mathcal{L}_t[g] := \sum_{k=1}^{\infty} g(k)u(t-k) \tag{2.114}$$

the data-generating model can be rewritten as

$$y(t) = \mathcal{L}_t[g] + e(t), \qquad t = 1, ..., N \tag{2.115}$$

For future use, let

$$Z_N := \begin{bmatrix} \mathcal{L}_1[g] & \mathcal{L}_2[g] & \cdots & \mathcal{L}_N[g] \end{bmatrix}^\top, \qquad Z_N \in \mathbb{R}^N \tag{2.116}$$

2.4.1.1 Gaussian Process Regression Framework

In this setting the process $\{e(t)\}$ is assumed to be zero-mean Gaussian white noise with variance $\sigma \in \mathbb{R}$, namely $\mathbb{E}[e(t)e(s)] = \sigma\delta_{t,s}$. According to the GPR procedure Rasmussen and Williams (2006), the system impulse response g is assumed to be a zero-mean Gaussian process on $\mathbb{N}$, independent of $\{e(t)\}$ with covariance

$$K_\eta(t,s) := \text{Cov}(g(t), g(s)) = \mathbb{E}[g(t)g(s)], \qquad K_\eta : \mathbb{N} \times \mathbb{N} \to \mathbb{R} \tag{2.117}$$

Equivalently, adopting a Bayesian terminology, one could say that a zero-mean Gaussian prior with covariance K_η is postulated for g.

The scalar function K_η is typically called *kernel* (for reasons which will become clear in Section 2.4.1.2) and is here specified through some parameters $\eta \in D_\eta \subset \mathbb{R}^{d_\eta}$, called *hyper-parameters* in this context. These are unknown and have to be estimated using the data $\mathcal{D}^N$ through one of the procedures illustrated in Section 2.4.3. The parametrization of function K_η through η allows the user to account for some desired properties of the impulse response g that has to be estimated. In particular, in the context of dynamical systems, features as smoothness and stability are sought. According to the Bayesian formalism, the shaping of K_η is referred to as *prior design* and will be further discussed in Chapter 3.

Thanks to the properties of Gaussian distributions, the vector Z_N in (2.116) is a multivariate zero-mean normal vector, since it consists of linear transformation of the Gaussian process g. Furthermore,

$$\mathrm{Cov}([Z_N]_t, [Z_N]_s) = \mathbb{E}\left[\mathcal{L}_t[g], \mathcal{L}_s[g]\right] = \Lambda(t,s) \tag{2.118}$$

where $\Lambda : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ is the so-called *output kernel*, defined as

$$\begin{aligned} \Lambda(t,s) :&= \sum_{k=1}^{\infty} u(t-k) \sum_{j=1}^{\infty} u(s-j) K_\eta(k,j) && (2.119)\\ &= \sum_{k=1}^{\infty} u(t-k)\mathcal{L}_s[K_\eta(k,\cdot)] && (2.120)\\ &= \mathcal{L}_t\left[\mathcal{L}_s[K_\eta(\cdot,\cdot)]\right] = \mathcal{L}_t\left[\mathcal{L}_s[K_\eta]\right] && (2.121) \end{aligned}$$

For future convenience, it is useful to define the corresponding *output kernel matrix* $\bar{\Lambda} \in \mathbb{R}^{N\times N}$ with the ij-th entry given by

$$\bar{\Lambda}_{ij} := \Lambda(i,j) = \mathcal{L}_i\left[\mathcal{L}_j[K_\eta]\right] \tag{2.122}$$

Due to the independence of the processes $\{g(k)\}$ and $\{e(t)\}$, the vector

$$Y_N := \begin{bmatrix} y(1) & y(2) & \cdots & y(N)\end{bmatrix}^\top, \qquad Y_N \in \mathbb{R}^N \tag{2.123}$$

and the impulse response $g(t)$ are jointly Gaussian for any $t \in \mathbb{N}$ (Papoulis and Pillai, 2002). The joint distribution is defined as

$$\begin{bmatrix} g(t) \\ Y_N \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0_N \end{bmatrix}, \begin{bmatrix} P_{g_t} & P_{g_t, Y_N} \\ P_{Y_N, g_t} & P_{Y_N} \end{bmatrix} \right), \qquad t \in \mathbb{N} \tag{2.124}$$

where

$$P_{g_t} := K_\eta(t,t) \tag{2.125}$$

$$\begin{aligned} P_{gt,Y_N} &:= \mathrm{Cov}(g(t), Y_N) = \mathrm{Cov}(g(t), Z_N) \\ &= [\mathcal{L}_1[K_\eta(t,\cdot)] \quad \mathcal{L}_2[K_\eta(t,\cdot)] \quad \cdots \quad \mathcal{L}_N[K_\eta(t,\cdot)]] \qquad (2.126) \\ P_{Y_N} &:= \mathrm{Cov}(Y_N, Y_N) = \mathrm{Cov}(Z_N, Z_N) + \sigma I_N = \bar{\Lambda} + \sigma I_N \qquad (2.127) \end{aligned}$$

If the hyper-parameters are known, the conditional distribution $p(g(t)|Y_N,\eta)$ is Gaussian, $p(g(t)|Y_N,\eta) \sim \mathcal{N}(\mu_{gt}^{post}, P_{gt}^{post})$, and its mean μ_{gt}^{post} and covariance P_{gt}^{post} can be computed through standard rules for conditional Gaussian variables:

$$\begin{aligned} \hat{g}(t) := \mu_{gt}^{post} &= P_{gt,Y_N} P_{Y_N}^{-1} Y_N \qquad (2.128) \\ &= [\mathcal{L}_1[K_\eta(t,\cdot)] \quad \mathcal{L}_2[K_\eta(t,\cdot)] \quad \cdots \quad \mathcal{L}_N[K_\eta(t,\cdot)]] \left(\bar{\Lambda} + \sigma I_N\right)^{-1} Y_N \\ P_{gt}^{post} &= P_{gt} - P_{gt,Y_N} P_{Y_N}^{-1} P_{Y_N,gt} \qquad (2.129) \end{aligned}$$

According to the Bayesian paradigm, $p(g(t)|Y_N,\eta)$ is the so-called *posterior*, i.e. the distribution of the unknown g conditioned on the observed data Y_N (and the hyper-parameters η). Using the Bayes' rule, this can be expressed as

$$p_g(g(t)|Y_N,\eta) = \frac{p_y(Y_N|g(t))\ p_g(g(t)|\eta)}{p_y(Y_N|\eta)}, \qquad t \in \mathbb{N} \qquad (2.130)$$

where the probability density function of Y_N given $g(t)$ is the *likelihood* function $p_y(Y_N; g(t))$, while $p_g(g(t)|\eta)$ denotes the PDF of the *prior* distribution. The PDF $p_y(Y_N|\eta)$ is the so-called *marginal likelihood* function, such defined:

$$p_y(Y_N|\eta) = \int_{\mathbb{R}} p_y(Y_N|g(t))\ p_g(g(t)|\eta) dg(t) \qquad (2.131)$$

As expression (2.131) clarifies, the name *marginal likelihood* is due to the marginalization over the unknown g.

In the Bayesian setting, the posterior mean μ_{gt}^{post} is also known as the *maximum a posteriori (MAP)* estimator of $g(t)$ (DeGroot, 2005) and it also coincides with the *minimum variance estimator.*

For future developments, it should be observed that μ_{gt}^{post} in (2.128) can be computed as

$$\mu_{gt}^{post} = \sum_{i=1}^{N} \hat{c}_i \mathcal{L}_i[K_\eta(t,\cdot)] \qquad (2.132)$$

where $\hat{c}_i$ is the i-th component of the vector

$$\hat{c} = (\bar{\Lambda} + \sigma I_N)^{-1} Y_N, \qquad \hat{c} \in \mathbb{R}^N \qquad (2.133)$$

Remark 2.4.1. The Bayesian inference procedure illustrated in equations (2.128)-(2.131) follows the so-called *Empirical Bayes* paradigm (Berger, 2013; Maritz and Lwin, 1989): the hyper-parameters η are assumed to be fixed to a certain estimated value, thus allowing to compute mean and covariance of the posterior distribution $p(g(t)|Y_N, \eta)$.
Alternatively, a *Full Bayes* approach could be used, where also η is treated as a random variable and the posterior PDF

$$p_g(g(t)|Y_N) = \int_{D_\eta} p_g(g(t)|Y_N, \eta) p_\eta(\eta|Y_N) d\eta \tag{2.134}$$

is inferred. Due to the intractability of the above integral, a sampled approximation of $p_g(g(t)|Y_N)$ needs to be computed by means of stochastic simulation techniques, such as the Markov Chain Monte Carlo (MCMC) algorithm (Gilks, 2005; Andrieu, Doucet, and Holenstein, 2010; Ninness and Henriksen, 2010).
The thorough discussion of these two alternative approaches is postponed to Section 2.4.3, where several techniques for the estimation of η from the data will be illustrated.

Remark 2.4.2. Besides assuming the knowledge of the hyper-parameters η, the previous derivation has also implicitly supposed that the noise variance σ is known. However, such hypothesis is unrealistic, since σ has to be somehow estimated through the available data $\mathcal{D}^N$. This can be done by following two possible routes, which will be detailed in Section 2.4.4.

2.4.1.2 Connection with Regularization in RKHS

The theory of Reproducing Kernel Hilbert Spaces (RKHS) provides a powerful mathematical tool for regularized function estimation (Aronszajn, 1950), i.e. for the reconstruction of a function starting from a finite set of input-output data pairs. In particular, regularization in RKHS represents an alternative to parametric approaches, where the function of interest is modelled through a set of parameters to be inferred from the given data. It should be recalled that in the literature of statistical learning, and specifically of inverse problems, regularization was introduced with the aim of solving the possible ill-posedness affecting the parametric estimators (Hoerl and Kennard, 1970; Tikhonov and Arsenin, 1977). This was the cause of the high variance affecting such estimators, especially in the case of complex models; as a consequence, the derived solutions resulted to be highly sensitive to data perturbations.
Exploiting the theory of RKHS, the unknown function can be searched for within an infinite dimensional space and ill-posedness (or, equivalently, overfitting) is avoided by adding a *regularization* term, designed in order to penalize undesired solutions. Specifi-

cally, given a set of data pairs $\{(x_i, y_i)\}_{i=1}^N$, generated according to the unknown function $g : \mathcal{X} \mapsto \mathbb{R}$ (i.e. $y_i = g(x_i)$, $x_i \in \mathcal{X}$ $\forall i = 1, ..., N$), g is estimated as

$$\min_{g \in \mathcal{H}} \quad \sum_{i=1}^{N} (y_i - g(x_i))^2 + \lambda \|g\|_{\mathcal{H}}^2, \qquad \gamma \in \mathbb{R} \tag{2.135}$$

In equation (2.135) $\mathcal{H}$ denotes the RKHS of functions $g : \mathcal{X} \mapsto \mathbb{R}$ within which the search is conducted and $\|\cdot\|_{\mathcal{H}} = \sqrt{\langle \cdot, \cdot \rangle_{\mathcal{H}}}$ is the associated norm. The functional $\|g\|_{\mathcal{H}}^2$ plays the role of the *regularization* term and penalizes solutions having a large norm in the space $\mathcal{H}$. The scalar λ is the so-called *regularization parameter*, which controls the relative influence of the loss and the penalty term. It has been proved that problem (2.135) is well-posed, meaning that there exists a unique solution with scarce sensitivity to data perturbations (Tikhonov and Arsenin, 1977).
The unfamiliar reader with the theory of RKHS is referred to Appendix A, where some basic concepts are reviewed. It is worth to recall here that every RKHS $\mathcal{H}$ is associated with a positive semidefinite kernel $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, called *reproducing kernel* (Aronszajn, 1950). K completely characterizes the space $\mathcal{H}$, meaning that both the functions belonging to $\mathcal{H}$ and the associated inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ are specified through K (Cucker and Smale, 2002).

The connection between GPR and regularized function estimation in RKHS has its origins in the work of Parzen (Parzen, 1961, 1970), who proved the duality between the Hilbert space spanned by a Gaussian process and its associated RKHS. In the statistical learning literature such relationship was first pointed out by Kimeldorf and Wahba (1970), and it has been later resumed by Girosi, Jones, and Poggio (1995) and in the textbooks Wahba (1990); Rasmussen and Williams (2006). In the following such relationship will be clarified in the context of system identification.

Differently from the previous section, in the RKHS framework the measurement noise $\{e(t)\}$ is simply assumed to be zero-mean white noise with variance σ; hence, the Gaussianity assumption is not required.

According to the measurement model (2.115), the unknown impulse response g is observed through the convolution functional $\mathcal{L}_t[g]$ (2.114). As observed by Twomey (1977), the inversion of such convolution results in an inverse problem which may lead to ill-conditioned solutions, especially when the input $\{u(t)\}$ is a low-pass signal or many measurements are given. The previous observations about the use of regularization in inverse problems suggest that g should be estimated as

$$\hat{g} = \arg\min_{g\in\mathcal{H}} \sum_{t=1}^{N} (y(t) - \mathcal{L}_t[g])^2 + \lambda\|g\|_{\mathcal{H}}^2, \qquad \lambda \in \mathbb{R} \tag{2.136}$$

In this case the impulse response g is treated as an element of the RKHS $\mathcal{H}$ of functions $g : \mathbb{N} \mapsto \mathbb{R}$ associated to the kernel $K_\eta : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$. It has been shown that if the linear functional $\mathcal{L}_t : \mathcal{H} \mapsto \mathbb{R}$ is continuous on $\mathcal{H}$, then the variational problem (2.136) admits a solution which can be expressed as a linear combination of a finite number of terms. The theory of RKHS tells that a linear functional $\mathcal{L}_t$ is continuous if and only if $\mathcal{L}_t[K_\eta(x,\cdot)]$ is a function in $\mathcal{H}$ (Aronszajn, 1950). It follows that the solution $\hat{g}$ can be computed as

$$\hat{g}(t) = \sum_{i=1}^{N} \hat{c}_i \; \mathcal{L}_i[K_\eta(t,\cdot)], \qquad \hat{c} = (\bar{\Lambda} + \lambda I_N)^{-1} Y_N \tag{2.137}$$

with $\bar{\Lambda}$ as defined in (2.122). Equation (2.137) coincides with the solution (2.132) obtained through GPR if $\lambda = \sigma$ and the kernel K_η is chosen equal to the covariance function defined in (2.117).

The result (2.137) is a consequence of the so-called *representer theorem* (Kimeldorf and Wahba (1971); Wahba (1990), Theorem 1.3.1) and of its extension provided by Yuan, Cai, et al. (2010), where the case of functional linear regression is treated. The interested reader is referred to Appendix A, where the representer theorem for the case in which direct observations of the unknown functions are available is stated.

Remark 2.4.3. Some authors have considered the estimation of the impulse response in an enlarged space, defined as $\mathcal{H} + \text{span}\{\phi_1, \phi_2, ..., \phi_r\}$, with $\{\phi_j\}_{j=1}^r$ playing the role of basis functions. According to this setting, the impulse response is assumed to be expressed as $g + \sum_{i=1}^r \theta_i\phi_i$, where $\{\theta_j\}_{j=1}^r$ are suitable parameters which can be jointly estimated with g by solving

$$\min_{g\in\mathcal{H},\theta\in\mathbb{R}^r} \sum_{t=1}^{N} \left(y(t) - \mathcal{L}_t\left[g + \sum_{j=1}^{r} \theta_j\phi_j \right] \right)^2 + \lambda\|g\|_{\mathcal{H}}^2, \qquad \lambda \in \mathbb{R} \tag{2.138}$$

Theorem 1.3.1 in Wahba (1990) proves that the corresponding impulse response estimate is given by

$$\sum_{i=1}^{N} \hat{c}_i \; \mathcal{L}_i[K_\eta(t,\cdot)] + \sum_{j=1}^{r} \hat{\theta}_j\phi_j \tag{2.139}$$

where

$$\hat{\theta} = (\Phi_N^\top A^{-1}\Phi_N)^{-1}\Phi_N^\top A^{-1} Y_N, \qquad \hat{c} = A^{-1}(Y_N - \Phi_N \hat{\theta}) \tag{2.140}$$

with $\Phi_N \in \mathbb{R}^{N \times r}$, $[\Phi_N]_{ij} := \mathcal{L}_i[\phi_j]$, and $A := \bar{\Lambda} + \lambda I_N$ (with $\bar{\Lambda}$ as defined in (2.122)). By means of this additional parametric component the flexibility of the obtained estimator is enhanced: for instance, the fast dynamics due to high-frequency poles could be captured. Such approach has been utilized e.g. by Pillonetto and De Nicolao (2010); Pillonetto et al. (2011a); Chen, Ohlsson, and Ljung (2012); Pillonetto et al. (2014).

2.4.1.3 Connection with Regularized LS

When the estimation procedure illustrated in Sections 2.4.1.1 and 2.4.1.2 is numerically implemented, only a finite number T of the estimated impulse response samples is actually computed. Such simplification, dictated by computational reasons, does not negatively affect the quality of the returned estimator. Indeed, if the system to be identified is BIBO stable, its impulse response is exponentially decaying. Therefore, by choosing a large enough value of T, the relevant system dynamics can be completely captured by retaining the first T impulse response coefficients $\{g(k)\}_{k=1}^T$. Such values are collected in the vector $\mathbf{g} \in \mathbb{R}^T$:

$$\mathbf{g} = [g(1) \quad g(2) \quad \cdots \quad g(T)]^\top \tag{2.141}$$

The notation $\mathbf{g}$ will be used in the remainder of the manuscript to denote the vector containing the first T impulse response coefficients, while $\mathbf{g}_i$ will indicate the i-th coefficient.

Assuming that $g(k) = 0, \; k = T+1, ..., \infty$, the data generating model (2.115) can be rewritten as the following FIR model

$$y(t) = \sum_{k=1}^{T} g(k)u(t-k) + e(t) = \sum_{k=1}^{T} \mathbf{g}_k u(t-k) + e(t), \qquad t = 1, ..., N \tag{2.142}$$

Recalling the definition of Y_N in equations (2.123) and defining the matrix $\Phi_N \in \mathbb{R}^{N \times T}$

$$\Phi_N := \begin{bmatrix} \varphi(1) & \varphi(2) & \cdots & \varphi(N) \end{bmatrix}^\top \tag{2.143}$$

$$\varphi(t) := \begin{bmatrix} u(t-1) & u(t-2) & \cdots & u(t-T) \end{bmatrix}^\top \tag{2.144}$$

equation (2.142) can be reformulated as a linear regression model

$$Y_N = \Phi_N \mathbf{g} + E \tag{2.145}$$

where $E := [e(1)\ e(2)\ \cdots\ e(N)]^\top$.

Recalling the Bayesian framework adopted in Section 2.4.1.1, a Gaussian prior distribution is postulated for the vector $\mathbf{g}$:

$$\mathbf{g} \sim \mathcal{N}\left(0_T, \bar{K}_\eta\right), \qquad \bar{K}_\eta \in \mathbb{R}^{T\times T} \tag{2.146}$$

with $\bar{K}_\eta$ denoting the covariance matrix, $\bar{K}_\eta = \mathbb{E}[\mathbf{g}\mathbf{g}^\top]$. Since $\mathbf{g}$ is assumed to be independent from the Gaussian innovation $\{e(t)\}$, the random vectors $\mathbf{g}$ and Y_N are jointly Gaussian, with joint distribution

$$\begin{bmatrix}\mathbf{g}\\ Y_N\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}0_T\\ 0_N\end{bmatrix}, \begin{bmatrix}\bar{K}_\eta & \bar{K}_\eta\Phi_N^\top\\ \Phi_N\bar{K}_\eta & \Phi_N\bar{K}_\eta\Phi_N^\top + \sigma I_N\end{bmatrix}\right) \tag{2.147}$$

Assuming the hyper-parameters η to be known, the conditional distribution $p(\mathbf{g}|Y_N,\eta)$ is Gaussian with mean and covariance given by

$$\hat{\mathbf{g}} := \mu_{\mathbf{g}}^{post} = \bar{K}_\eta\Phi_N^\top(\Phi_N\bar{K}_\eta\Phi_N^\top + \sigma I_N)^{-1}Y_N \tag{2.148}$$

$$P_{\mathbf{g}}^{post} = \bar{K}_\eta - \bar{K}_\eta\Phi_N^\top(\Phi_N\bar{K}_\eta\Phi_N^\top + \sigma I_N)^{-1}\Phi_N\bar{K}_\eta \tag{2.149}$$

Simple algebraic manipulations show that the MAP estimator (2.148) coincides with the solution of the following regularized LS problem:

$$\underset{\mathbf{g}\in\mathbb{R}^T}{\arg\min}\ \|Y_N - \Phi_N\mathbf{g}\|_2^2 + \sigma\mathbf{g}^\top\bar{K}_\eta\mathbf{g} \tag{2.150}$$

With regard to the RKHS framework, it can be easily shown that there exists a suitable RKHS $\mathcal{H}$ such that the t-th component of the solution to (2.150), $\hat{\mathbf{g}}_t$, is equal to $\hat{g}(t)$ computed according to (2.137). Such RKHS consists of functions $g : \mathcal{X} \to \mathbb{R}$, with $\mathcal{X} = \{1, 2, ..., T\}$ and is associated to the reproducing kernel $K_\eta : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$, defined by $K_\eta(i,j) = [\bar{K}_\eta]_{ij}$. $\bar{K}_\eta$ is the covariance matrix introduced in (2.146): its positive semidefiniteness guarantees the positive semidefiniteness of kernel K_η, and in turn the uniqueness of the RKHS associated to it (Theorem A.0.5 in Appendix A).

From the definition of $\mathcal{X}$, it follows that $\mathcal{L}_t[g] = \varphi(t)\mathbf{g}$, where $\mathbf{g}$ is the impulse response vector (2.141) while $\varphi(t)$ is defined in (2.144); consequently, the sum of squared prediction errors can be rewritten as $\sum_{t=1}^N (y(t) - \mathcal{L}_t[g])^2 = \|Y_N - \Phi_N\mathbf{g}\|_2^2$.

Furthermore, using the formula for function evaluation in $\mathcal{H}$ provided in (A.1), it is possible to write

$$\mathbf{g} = \begin{bmatrix} g(1) \\ g(2) \\ \vdots \\ g(T) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{T} a_i K_\eta(i,1) \\ \sum_{i=1}^{T} a_i K_\eta(i,2) \\ \vdots \\ \sum_{i=1}^{T} a_i K_\eta(i,T) \end{bmatrix} = \bar{K}_\eta a \tag{2.151}$$

for some vector $a = [a_1 \; a_2 \; \cdots \; a_T]^\top$; according to (A.2),

$$\|g\|^2_{\mathcal{H}} = a^\top \bar{K}_\eta a \tag{2.152}$$

Combining equations (2.151) and (2.152), it results $\|g\|^2_{\mathcal{H}} = \mathbf{g}^\top \bar{K}_\eta^{-1} \mathbf{g}$, thus showing that problems (2.136) and (2.150) coincide, once λ is set equal to σ.

Alternatively, the equivalence between the two frameworks could have been established by directly inspecting the estimator formula (2.137).

2.4.2 Non-Parametric Bayesian Methods for MIMO systems

The algorithm illustrated in Section 2.4.1 is here extended to the identification of MIMO systems ($p > 1$ and $m > 1$), meaning that $g : \mathbb{N} \to \mathbb{R}^{p \times m}$. To simplify the treatment, a vector-valued version $\underline{g}$ of the impulse response function is considered:

$$\begin{aligned} \underline{g} &: \mathbb{N} \to \mathbb{R}^{pm} \\ & k \mapsto \underline{g}(k) := \mathrm{vec}(g(k)) \end{aligned} \tag{2.153}$$

Furthermore, in order to maintain a simple notation, the same symbols of Section 2.4.1 will be here adopted even if the definition of the corresponding operators differ from the previous ones.

According to the definition of $\underline{g}$, the functional $\mathcal{L}_t[\cdot]$ is formulated over the space $\mathcal{H}$ of functions $\underline{g} : \mathbb{N} \to \mathbb{R}^{pm}$:

$$\begin{aligned} \mathcal{L}_t : \mathcal{H} &\to \mathbb{R}^p \\ \underline{g} &\mapsto \sum_{k=1}^{\infty} \phi^\top(t-k)\underline{g}(k) \end{aligned} \tag{2.154}$$

where

$$\phi(t) := \left[\; u_1(t) I_p \; \middle| \; u_2(t) I_p \; \middle| \; \cdots \; \middle| \; u_m(t) I_p \; \right]^\top, \qquad \phi(t) \in \mathbb{R}^{pm \times p} \tag{2.155}$$

In the equation above $u_i(t)$ denotes the i-th component of the input signal at time t.

Postulating $H(q) \equiv 1$, the data-generating model (2.6) can be rewritten as

$$y(t) = \mathcal{L}_t[\underline{g}] + e(t) \tag{2.156}$$

with $\{e(t)\}$ here assumed to be white noise with

$$\mathbb{E}[e(t)e^\top(s)] = \Sigma\delta_{t,s}, \qquad \Sigma := \operatorname{diag}([\sigma_1, ..., \sigma_p]) \tag{2.157}$$

As in the SISO case, define the vector $z(t) = \mathcal{L}_t[\underline{g}]$, $z(t) \in \mathbb{R}^p$, and let

$$Z_N := \begin{bmatrix} z^\top(1) & z^\top(2) & \cdots & z^\top(N) \end{bmatrix}^\top, \qquad Z_N \in \mathbb{R}^{Np} \tag{2.158}$$

For future use, let $A : \mathbb{N} \to \mathbb{R}^{pm\times\alpha}$, $\alpha \in \mathbb{N}$, and define the following column-wise decomposition of $A(t)$:

$$A(t) = \begin{bmatrix} A_1(t) & A_2(t) & \cdots & A_\alpha(t) \end{bmatrix}, \qquad A_i : \mathbb{N} \to \mathbb{R}^{pm} \tag{2.159}$$

Furthermore, the operator $\underline{\mathcal{L}}_t[\cdot]$ over the space $\underline{\mathcal{H}}^\alpha$ of functions $\mathbb{N} \to \mathbb{R}^{pm\times\alpha}$ is defined as:

$$\begin{aligned} \underline{\mathcal{L}}_t : \underline{\mathcal{H}}^\alpha &\to \mathbb{R}^{p\times\alpha} \\ A &\mapsto [\mathcal{L}_t[A_1] \quad \mathcal{L}_t[A_2] \quad \cdots \quad \mathcal{L}_t[A_\alpha]] \end{aligned} \tag{2.160}$$

Accordingly, let

$$\begin{aligned} \underline{\mathcal{L}}_t^\top : \underline{\mathcal{H}}^\alpha &\to \mathbb{R}^{\alpha\times p} \\ A &\mapsto \begin{bmatrix} \mathcal{L}_t^\top[A_1] \\ \mathcal{L}_t^\top[A_2] \\ \vdots \\ \mathcal{L}_t^\top[A_\alpha] \end{bmatrix} \end{aligned} \tag{2.161}$$

where $\mathcal{L}_t^\top[A_i] = \sum_{k=1}^\infty A_i^\top(k)\phi(t-k)$.

2.4.2.1 Gaussian Process Regression Framework

In this section the noise $\{e(t)\}$ is assumed to be Gaussian white noise with covariance Σ. Following the Bayesian paradigm of Section 2.4.1.1, $\{\underline{g}(k)\}$ is considered as a realization of a vector-valued zero-mean Gaussian processes, independent of $\{e(t)\}$, with covariance

$$K_\eta(t,s) := \operatorname{Cov}(\underline{g}(t), \underline{g}(s)) = \mathbb{E}[\underline{g}(t)\underline{g}^\top(s)], \qquad K_\eta : \mathbb{N}\times\mathbb{N} \to \mathbb{R}^{pm\times pm} \tag{2.162}$$

Let introduce the following column-wise decomposition of the kernel function K_η:

$$K_\eta(t,s) = \begin{bmatrix} K_{\eta,1}(t,s) & K_{\eta,2}(t,s) & \cdots & K_{\eta,pm}(t,s) \end{bmatrix}, \qquad K_{\eta,i} : \mathbb{N} \times \mathbb{N} \to \mathbb{R}^{pm} \tag{2.163}$$

Denoting with $\underline{g}_i(t)$ the i-th component of $\underline{g}(t)$, it follows that

$$\begin{aligned} \operatorname{Cov}(y(t), \underline{g}_i(s)) &= \operatorname{Cov}(z(t), \underline{g}_i(s)) \qquad (2.164) \\ &= \mathbb{E}\left[\sum_{k=1}^{\infty} \phi^\top(t-k)\underline{g}(k)\underline{g}_i(s)\right] \\ &= \sum_{k=1}^{\infty} \phi^\top(t-k) K_{\eta,i}(k,s) \\ &= \mathcal{L}_t[K_{\eta,i}(\cdot,s)] \end{aligned}$$

and

$$\begin{aligned} \operatorname{Cov}(y(t), \underline{g}(s)) &= \operatorname{Cov}(z(t), \underline{g}(s)) \qquad (2.165) \\ &= \mathbb{E}\left[\sum_{k=1}^{\infty} \phi^\top(t-k)\underline{g}(k)\underline{g}^\top(s)\right] \\ &= \sum_{k=1}^{\infty} \phi^\top(t-k) K_\eta(k,s) \\ &= [\mathcal{L}_t[K_{\eta,1}(\cdot,s)] \quad \mathcal{L}_t[K_{\eta,2}(\cdot,s)] \quad \cdots \quad \mathcal{L}_t[K_{\eta,pm}(\cdot,s)]] \\ &= \underline{\mathcal{L}}_t[K_\eta(\cdot,s)] \end{aligned}$$

where the operator $\underline{\mathcal{L}}_t[\cdot]$ is here applied on the function $K_\eta(\cdot,s)$ belonging to $\underline{\mathcal{H}}^{pm}$. Consequently,

$$\operatorname{Cov}(Y_N, g(s)) = \begin{bmatrix} \underline{\mathcal{L}}_1[K_\eta(\cdot,s)] \\ \cdots \\ \underline{\mathcal{L}}_N[K_\eta(\cdot,s)] \end{bmatrix} \tag{2.166}$$

Now define the *output kernel* function $\Lambda : \mathbb{N} \times \mathbb{N} \to \mathbb{R}^{p \times p}$ as

$$\begin{aligned} \Lambda(t,s) &:= \sum_{k=1}^{\infty} \sum_{l=1}^{\infty} \phi^\top(t-k) K_\eta(k,l) \phi(s-l) \\ &= \sum_{k=1}^{\infty} \phi^\top(t-k) \underline{\mathcal{L}}_s^\top [K_\eta(k,\cdot)] \\ &= \underline{\mathcal{L}}_t\left[\underline{\mathcal{L}}_s^\top [K_\eta]\right] \qquad (2.167) \end{aligned}$$

where $\underline{\mathcal{L}}_s^\top$ is here applied on $\underline{\mathcal{H}}^{pm}$, while $\underline{\mathcal{L}}_t$ is applied on $\underline{\mathcal{H}}^p$. Correspondingly, define the *output kernel matrix* $\bar{\Lambda} \in \mathbb{R}^{Np\times Np}$ as

$$\bar{\Lambda} := \begin{bmatrix} \Lambda(1,1) & \cdots & \Lambda(1,N) \\ \vdots & \ddots & \vdots \\ \Lambda(N,1) & \cdots & \Lambda(N,N) \end{bmatrix} \tag{2.168}$$

It follows that

$$\begin{aligned} \operatorname{Cov}(y(t), y(s)) &= \operatorname{Cov}(z(t), z(s)) + \Sigma \qquad (2.169) \\ &= \mathbb{E}\left[\sum_{k=1}^{\infty}\sum_{l=1}^{\infty} \phi^\top(t-k)\underline{g}(k)\underline{g}^\top(l)\phi(s-l)\right] + \Sigma \\ &= \sum_{k=1}^{\infty}\sum_{l=1}^{\infty} \phi^\top(t-k)K_\eta(k,l)\phi(s-l) + \Sigma \\ &= \Lambda(t,s) + \Sigma \end{aligned}$$

and

$$\operatorname{Cov}(Y_N, Y_N) = \bar{\Lambda} + \widetilde{\Sigma}_N \tag{2.170}$$

$$\widetilde{\Sigma}_N := \Sigma \otimes I_N, \qquad \widetilde{\Sigma}_N \in \mathbb{R}^{Np\times Np} \tag{2.171}$$

Thanks to the independence of the processes $\{\underline{g}(k)\}$, and $\{e(t)\}$, the vector $Y_N \in \mathbb{R}^{Np}$ (defined as in (2.123)) and $\underline{g}(t)$ are jointly normally distributed for any $t \in \mathbb{N}$. Assuming the hyper-parameters η to be known and using the rules of conditioned Gaussian variables, the minimum variance estimator of $\underline{g}(t)$ is given by:

$$\begin{aligned} \mu_{\underline{g}_t}^{post} = \mathbb{E}\left[\underline{g}(t)|Y_N,\eta\right] &= \operatorname{Cov}(\underline{g}(t), Y_N)\left\{\operatorname{Cov}(Y_N,Y_N)\right\}^{-1} Y_N \qquad (2.172) \\ &= \left[\underline{\mathcal{L}}_1^\top[K_\eta(t,\cdot)] \quad \underline{\mathcal{L}}_2^\top[K_\eta(t,\cdot)] \quad \cdots \quad \underline{\mathcal{L}}_N^\top[K_\eta(t,\cdot)]\right](\bar{\Lambda}+\widetilde{\Sigma}_N)^{-1}Y_N \\ &= \sum_{i=1}^{N} \underline{\mathcal{L}}_i^\top[K_\eta(t,\cdot)]\hat{c}_{(i)} \qquad (2.173) \end{aligned}$$

where $\hat{c}_{(i)}$ denotes the i-th block of size p of the vector $\hat{c} = (\bar{\Lambda} + \widetilde{\Sigma}_N)^{-1}Y_N$. Furthermore, the posterior covariance is computed as

$$P_{\underline{g}_t}^{post} = \operatorname{Cov}(\underline{g}(t), \underline{g}(t)) - \operatorname{Cov}(\underline{g}(t), Y_N)\left\{\operatorname{Cov}(Y_N,Y_N)\right\}^{-1}\operatorname{Cov}(Y_N, \underline{g}(t)) \tag{2.174}$$

2.4.2.2 Connection with Regularization in RKHS

The illustration of the Bayesian procedure for the identification of MIMO systems has clarified how such problem involves the joint estimation of several functions, namely the pm impulse responses connecting each input-output channel. In the literature of learning through kernel methods, the joint estimation of multiple functions is known as *multi-task learning*. This kind of problems has been treated e.g. by Caruana (1998); Evgeniou and Pontil (2004); Micchelli and Pontil (2005b),Evgeniou, Micchelli, and Pontil (2005). According to the framework introduced in Section 2.4.2, the aim is to estimate the vector-valued function $\underline{g} : \mathbb{N} \to \mathbb{R}^{pm}$ using the available data $\mathcal{D}^N$. Hence, $\underline{g}$ is searched for within a RKHS $\mathcal{H}$ consisting of functions $\underline{f} : \mathcal{X} \to \mathcal{Y}$, with $\mathcal{X} = \mathbb{N}$ and $\mathcal{Y} = \mathbb{R}^{pm}$. The reproducing kernel $K_\eta : \mathbb{N} \times \mathbb{N} \to \mathbb{R}^{pm\times pm}$ is associated to $\mathcal{H}$. Exploiting this setting, the impulse response function is estimated by solving

$$\underline{\hat{g}} := \arg\min_{\underline{g}\in\mathcal{H}} \sum_{t=1}^{N} \|y(t) - \mathcal{L}_t[\underline{g}]\|^2 + \lambda \|\underline{g}\|^2_{\mathcal{H}}, \qquad \lambda \in \mathbb{R} \tag{2.175}$$

The generalized version of the representer theorem exploited in Section 2.4.1.2 applies also when dealing with RKHS of vector-valued functions. Therefore, if the linear functional $\mathcal{L}_t$ in (2.154) (and in turn $\underline{\mathcal{L}}_t$ in (2.160)) is continuous on $\mathcal{H}$ the solution to problem (2.175) is given by

$$\underline{\hat{g}} = \sum_{i=1}^{N} \underline{\mathcal{L}}_i^\top [K_\eta(t, \cdot)] \hat{c}_{(i)} \tag{2.176}$$

with $\hat{c}_{(i)} \in \mathbb{R}^p$, $i = 1, ..., N$, being the unique solution to the set of linear equations

$$\sum_{i=1}^{N} (\Lambda(t,i) + \lambda\delta_{t,i}) c_{(i)} = y(t), \qquad t = 1, ..., N \tag{2.177}$$

Equivalently, $\hat{c}_{(i)}$ is the i-th block of size p of the vector $\hat{c} = (\bar{\Lambda} + \lambda I_{Np})^{-1} Y_N$ (Micchelli and Pontil, 2005a). It follows that the solution of problem (2.175) coincides with (2.172) if the output noise variance is assumed to be equal throughout the channels (i.e. $\Sigma = \sigma I_p$ in equation (2.157)) and if λ is set equal to σ.

2.4.2.3 Connection with Regularized LS

In practice, the quality of the returned estimator remains (almost) unaltered if only the first T impulse response coefficients are chosen, provided that T is chosen sufficiently large. Hence, by collecting such coefficients in the vector $\mathbf{g} \in \mathbb{R}^{pmT}$,

$$\mathbf{g} = \left[\text{vec}^\top(g(1)) \quad \text{vec}^\top(g(2)) \quad \cdots \quad \text{vec}^\top(g(T))\right]^\top \tag{2.178}$$

the model (2.156) can be approximated through the following linear regression model

$$Y_N = \Phi_N \mathbf{g} + E \tag{2.179}$$

where $E = [e^\top(1)\ e^\top(2)\ \cdots\ e^\top(N)]^\top$, $E \in \mathbb{R}^{Np}$ and

$$\Phi_N = \left[\varphi(1) \quad \varphi(2) \quad \cdots \quad \varphi(N)\right]^\top, \qquad \Phi_N \in \mathbb{R}^{Np \times pmT} \tag{2.180}$$

$$\varphi(t) = \left[\phi^\top(t-1) \quad \phi^\top(t-2) \quad \cdots \quad \phi^\top(t-T)\right]^\top, \qquad \varphi(t) \in \mathbb{R}^{pmT \times p} \tag{2.181}$$

where $\phi(t)$ was defined in equation (2.155). When a Gaussian prior is postulated for $\mathbf{g}$, i.e. $\mathbf{g} \sim \mathcal{N}(0_{pmT}, \bar{K}_\eta)$, $\bar{K}_\eta \in \mathbb{R}^{pmT \times pmT}$, the Bayesian inference procedure for the MIMO case follows straightforwardly from the one previously illustrated for SISO systems. In particular, using the matrices above defined, the minimum variance estimator and the posterior covariance can be computed through equations (2.148) and (2.149).

2.4.3 Hyperparameters Tuning

The computation of the non-parametric Bayesian estimate (2.150) relies on the knowledge of the hyper-parameters η. However, these are a-priori unknown and they have to be somehow estimated from the given data $\mathcal{D}^N$. As observed in Remark 2.4.1, several techniques could be adopted for such estimation. These can be clustered into two main families, according to the interpretation given to the impulse response to be estimated. Namely, if it is interpreted as a random process, the Bayesian perspective used in Sections 2.4.1.1 and 2.4.2.1 provides two main approaches: the *Empirical Bayes* and the *Full Bayes*. While the first approximates the posterior distribution $p_\eta(\eta|Y_N)$ with a delta-function, the latter exploits stochastic simulation algorithms to obtain a sampled approximation of $p_\eta(\eta|Y_N)$; in such a way, the Full Bayes approach also accounts for the uncertainty of the hyper-parameters (Magni, Bellazzi, and De Nicolao, 1998).

Neglecting the probabilistic interpretation of the impulse response to be estimated and thus considering it a deterministic function, the Bayesian inference procedure turns out to be a regularization problem (as shown in Sections 2.4.1.2, 2.4.2.2 and 2.4.1.3, 2.4.2.3). Therefore, procedures such as cross-validation or C_p-statistics can be exploited for the estimation of η.

These two families of techniques are now detailed. In favour of a practical implementation, the illustration is based on the finite-dimensional notation introduced in Sections 2.4.1.3 and 2.4.2.3. In particular, the more general MIMO case will be treated.

2.4.3.1 Hyper-parameters Tuning in a Bayesian framework

Empirical Bayes. Such approach relies on the approximation of the hyper-parameters posterior $p_\eta(\eta|Y_N)$ in terms of a delta-function (Berger, 2013; Maritz and Lwin, 1989). The tuning of the hyper-parameters thus reduces to the estimation of the delta location. A widely used method assumes that such delta-function is located at the mode of $p_\eta(\eta|Y_N)$. To estimate it, it should first be observed that, when a non-informative prior is fixed for η,

$$p_\eta(\eta|Y_N) = \frac{p_y(Y_N|\eta)p_\eta(\eta)}{p_y(Y_N)} \propto p_y(Y_N|\eta) \tag{2.182}$$

Hence, the hyper-parameters are tuned by maximizing the *marginal likelihood* function, which was defined in (2.131):

$$\hat{\eta}_{EB} = \arg\max_{\eta\in D_\eta} p_\eta(\eta|Y_N) \equiv \arg\max_{\eta\in D_\eta} p_y(Y_N|\eta) \tag{2.183}$$

$p_y(Y_N|\eta)$ is also known to as *type-II likelihood* (Berger, 2013) or as *evidence for the hyper-parameters* (MacKay, 1992), while the "marginal likelihood maximization" approach is sometimes referred to as "evidence procedure".

When the measurement noise $\{e(t)\}$ is assumed to be Gaussian white noise, and the impulse response $\mathbf{g}$ is assigned a zero-mean Gaussian prior with covariance $\bar{K}_\eta$, $p(Y_N|\eta)$ is Gaussian too; namely,

$$f_{ML}(\eta) := -\ln p_y(Y_N|\eta) = Y_N^\top(\Phi_N\bar{K}_\eta\Phi_N^\top + \widetilde{\Sigma}_N)^{-1}Y_N + \ln\det(\Phi_N\bar{K}_\eta\Phi_N^\top + \widetilde{\Sigma}_N) \tag{2.184}$$

where Φ_N, Y_N and $\widetilde{\Sigma}_N$ have been respectively defined in (2.143), (2.123) and (2.171), while $\bar{K}_\eta$ is the kernel matrix.

Some authors have investigated the goodness of such approach in the literature of Bayesian learning. MacKay (1992) discusses about the tendency of marginal likelihood maximization to automatically penalize unnecessarily complex models, thus embedding the derived estimator with the so-called *Occam's razor principle.*

In the field of system identification, a recent work has proved the robustness of such method, even when undermodelling is present (Pillonetto and Chiuso, 2015). The properties of such estimator have been also investigated by Aravkin, Burke, Chiuso, and Pillonetto (2012).

Numerical routines to solve problem (2.183) include constrained gradient methods (Nocedal and Wright, 2006) and the Expectation-Maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977; McLachlan and Krishnan, 2007). The reader is referred to Section 2.4.5.2 for a more detailed discussion of these techniques.

Full Bayes. As observed in Remark 2.4.1, the Full Bayes approach slightly differs from the procedure illustrated in Sections 2.4.1 and 2.4.2, which assumes the availability of a punctual estimate of η. On the contrary, the methodology here described computes a Monte-Carlo approximation of the posterior PDF:

$$p_{\mathbf{g}}(\mathbf{g}|Y_N) = \int_{D_\eta} p_{\mathbf{g}}(\mathbf{g}|Y_N,\eta)p_\eta(\eta|Y_N)d\eta \approx \frac{1}{N_{sp}}\sum_{i=1}^{N_{sp}} p_{\mathbf{g}}(\mathbf{g}|Y_N,\eta^{(i)}) \tag{2.185}$$

where $p_{\mathbf{g}}(\mathbf{g}|Y_N,\eta^{(i)})$ denotes the posterior density when the hyper-parameters are fixed to $\eta^{(i)}$. $p(\mathbf{g}|Y_N,\eta^{(i)})$ is a Gaussian distribution with mean and covariance respectively given by (2.148) and (2.149). For approximation (2.185) to hold, the values $\eta^{(i)}$ have to be drawn from $p(\eta|Y_N)$. This can be achieved by designing a suitable MCMC algorithm (Gilks, 2005), whose implementation will be detailed in Section 2.4.5.2.

Once the posterior approximation (2.185) is computed, the minimum variance impulse response estimate could then be taken as

$$\hat{\mathbf{g}}_{FB} = \frac{1}{N_{sp}}\sum_{i=1}^{N_{sp}} \mathbf{g}^{(i)} \tag{2.186}$$

with $\mathbf{g}^{(i)}$ drawn from $p_{\mathbf{g}}(\mathbf{g}|Y_N,\eta^{(i)})$. It should be pointed out that (2.186) is only a possible way to compute the impulse response estimator; for instance a MAP estimator could be defined as

$$\hat{\mathbf{g}}_{MAP} = \max_{i=1,..,N_{sp}} \mathbb{E}[p_{\mathbf{g}}(\mathbf{g}|Y_N,\eta^{(i)})] \tag{2.187}$$

2.4.3.2 Hyper-parameters Tuning in a deterministic framework

Hyper-parameters tuning in a deterministic setting is performed by minimizing an estimate of the so-called *generalization error*, given by

$$\mathbb{E}\left[\frac{1}{N}\sum_{t=1}^{N}\|\varphi^\top(t)\hat{\mathbf{g}} - z(t)\|^2\right], \qquad z(t) := \sum_{k=1}^{\infty} g(k)u(t-k) \tag{2.188}$$

The expectation above is taken w.r.t. the measurement noise affecting the given data $\mathcal{D}^N$, while $z(t)$ denotes the noiseless system output and $\varphi(t)$ contains past input values, as defined in (2.144). Commonly used approaches to estimate (2.188) are briefly detailed in the following.

C_p statistics. The C_p statistics provides an unbiased estimator of (2.188) if the noise variance Σ is known (Mallows, 1973). Consequently, the hyper-parameters can be tuned as

$$\hat{\eta} = \arg\min_{\eta \in D_\eta} \frac{1}{N} \|Y_N - \Phi_N \hat{\mathbf{g}}\|^2 + \frac{2}{N} \mathrm{Tr}\{\mathrm{Df}_{\mathbf{g}}(\eta) \widetilde{\Sigma}_N\} \tag{2.189}$$

where $\widetilde{\Sigma}_N$ has been defined in (2.171), while

$$\mathrm{Df}_{\mathbf{g}}(\eta) := \Gamma(\eta)\widetilde{\Sigma}_N^{-1} \qquad\qquad \Gamma(\eta) := \Phi_N \bar{K}_\eta \Phi_N^\top (\Phi_N \bar{K}_\eta \Phi_N^\top + \widetilde{\Sigma}_N)^{-1} Y_N \tag{2.190}$$

are the so-called *matricial degrees of freedom*, which measure the flexibility of the estimator $\hat{\mathbf{g}}$ as a function of η (Pillonetto and Chiuso, 2015; Tibshirani, 2014). Equation (2.190) makes clear the role played by the hyper-parameters in the non-parametric estimation here discussed: the tuning of η represents the counterpart of complexity selection in parametric methods. However, differently from that setting, here complexity can be continuously controlled by changing the value of η.
As a final remark, it should be observed that the C_p statistics in equation (2.189) coincides with the *Stein Unbiased Risk Estimation (SURE)* criterion, when the measurement noise is assumed to be normally distributed (Stein, 1981).

Cross-Validation. This is a widely used approach for estimating (2.188). It first requires to split the data $\mathcal{D}^N$ into two parts: $\mathcal{D}^{N_{tr}} = \{(u_{tr}(t), y_{tr}(t)\}_{t=1}^{N_{tr}}$ and $\mathcal{D}^{N_{val}} = \{(u_{val}(t), y_{val}(t)\}_{t=1}^{N_{val}}$, $N_{val} + N_{tr} := N$. The hyper-parameters are then tuned by solving

$$\hat{\eta} = \arg\min_{\eta \in D_\eta} \frac{1}{N} \|Y_{N_{val}} - \Phi N_{val}\ \hat{\mathbf{g}}_{tr}(\eta)\|^2 \tag{2.191}$$

$$\hat{\mathbf{g}}_{tr}(\eta) : = \bar{K}_\eta \Phi {N_{tr}}^\top (\Phi_{N_{tr}} \bar{K}_\eta \Phi_{N_{tr}}^\top + \widetilde{\Sigma}_{N_{tr}})^{-1} Y_{N_{tr}} \tag{2.192}$$

where $Y_{N_{tr}} \in \mathbb{R}^{pN_{tr}}$ and $Y_{N_{val}} \in \mathbb{R}^{pN_{val}}$ contain output values belonging respectively to the training and the validation dataset; analogously, $\Phi_{N_{tr}} \in \mathbb{R}^{pN_{tr} \times Tpm}$ and $\Phi_{N_{val}} \in \mathbb{R}^{pN_{val} \times Tpm}$ contain past input values from $\mathcal{D}^{N_{tr}}$ and $\mathcal{D}^{N_{val}}$.
In practice, dataset $\mathcal{D}^{N_{tr}}$ is used to compute the estimate $\hat{\mathbf{g}}_{tr}(\eta)$, while the generalization error is approximated by evaluating the prediction capabilities of $\hat{\mathbf{g}}_{tr}(\eta)$ on the data contained in $\mathcal{D}^{N_{val}}$.
Several variants of cross-validation exist in the literature of statistical learning, such as *k-fold* cross-validation, where k disjoint datasets (folds) are extracted from the data and k different estimations are performed. An extreme case of such procedure is the so-called

leave-one-out, where N folds are used, meaning that each validation set consists of only one sample. Special cases of *leave-one-out* are *PRESS* and *Generalized Cross Validation (GCV)*; see Allen (1974); Golub, Heath, and Wahba (1979); Wahba (1990).

2.4.4 User's Choices

Analogously to the parametric techniques reviewed in Sections 2.2 and 2.3, the non-parametric Bayesian methods here illustrated also involve some choices that have to be taken by the user. These are briefly discussed in the following.

Choice of the impulse response length T. As mentioned in Section 2.4.1.3, such choice is not critical for the quality of the returned estimate. T needs to be simply chosen large enough in order to guarantee that the relevant system dynamics is captured. If the value of T does not crucially affect the goodness of the identified model, it significantly impacts the computational effort of the methods here considered. Section 2.4.5 will make clear such dependence.

Choice of the kernel (Prior Design). Since prior design represents the main topic of Chapter 3, the reader is referred to that chapter for a thorough discussion about such choice. Here it should simply be recalled that the prior has to be designed in order to account for the desired properties of the impulse response to be estimated. Currently, the most commonly adopted kernels are adaptations of the classical spline-kernels used in the statical learning literature (Wahba, 1990; Hastie et al., 2009). Specifically, the modified versions of such kernels allow to describe the exponentially decaying profile of BIBO stable impulse responses (Pillonetto and De Nicolao, 2010). This type of kernels is commonly referred to as *stable-spline* kernel.

Choice of the procedure for the hyper-parameters tuning. Representing the counterpart of complexity selection in parametric methods, hyper-parameters tuning stands as an important step of any non-parametric Bayesian identification routine. This task can be accomplished through several procedures, as illustrated in Section 2.4.3. The recent literature on non-parametric methods for system identification mainly adopts the Empirical Bayes approach through marginal likelihood maximization (Pillonetto and De Nicolao, 2010; Pillonetto et al., 2011a; Chen et al., 2012). While such technique has been often criticized in the classical literature on spline models (Wahba, 1990; Evgeniou, Pontil, and Poggio, 2000), recent theoretical contributions have tried to explain the effectiveness of the evidence maximization in the context of system identification

(Aravkin et al., 2012; Pillonetto and Chiuso, 2015). In particular, the investigation conducted by Pillonetto and Chiuso (2015) relies on the introduction of the concept of *excess degrees of freedom*, which measure the additional complexity associated to an estimator that has to determine the hyper-parameters from the data. Pillonetto and Chiuso (2015) carry out a comparison between estimators derived through minimization of the SURE criterion (2.189) and through maximization of the marginal likelihood: the results show that the latter guarantee a better balance between fit and parsimony, thanks to a better control of the so-called excess degrees of freedom.

Recently, Prando, Romeres, Pillonetto, and Chiuso (2016a) have numerically compared Empirical Bayes (through marginal likelihood maximization) and Full Bayes approaches: while the results do not highlight a significant performance gap, the computational effort appears much more favourable to the Empirical Bayes approach. The outcomes of such comparison are reported in Chapter 4.

Following the recent trend in the literature of non-parametric Bayesian methods, the results presented in the following chapters of the thesis rely on marginal likelihood maximization procedure for the hyper-parameters tuning.

Choice of the procedure for the noise variance Σ estimation. The noise variance could be treated as a hyper-parameter and hence estimated with η, through one of the procedures described in Section 2.4.3 (MacKay, 1992; Chen, Andersen, Ljung, Chiuso, and Pillonetto, 2014). Alternatively, Σ could be estimated as the sample variance of the prediction error achieved through a LS estimate (Goodwin, Gevers, and Ninness, 1992; Ljung, 1999), such as an ARX or a FIR model (Pillonetto and De Nicolao, 2010; Chen et al., 2012).

Choice about the estimation of a parametric component. The user should decide whether or not the non-parametric estimate should be equipped with a parametric component, as illustrated in Remark 2.4.3. This feature was originally proposed in order to let the estimator capture some system dynamics (e.g. high-frequency behaviour), which were difficulty reproduced by the smooth kernels inherited from the machine learning literature (Pillonetto and De Nicolao, 2010; Pillonetto et al., 2011a; Chen et al., 2012). However, recent contributions have tried to address this limitation by directly designing enriched kernels, thus enabling them to capture some desired dynamics (see e.g. the multiple kernels introduced by Chiuso, Chen, Ljung, and Pillonetto (2014) and the discussion in Section 3.3).

2.4.5 Algorithmic Details

From an algorithmic point of view, the non-parametric Bayesian identification procedure can be split into two main steps: the hyper-parameters tuning and the computation of the impulse response estimate. The discussion which follows is therefore based on this scheme.

2.4.5.1 Impulse Response Estimate

Treating the impulse response to be estimated as an infinite-dimensional object (i.e. using the viewpoint of Sections 2.4.1.1 and 2.4.2.2), the computation of $\hat{g}$ in equation (2.132) (or, equivalently, (2.137)) requires to solve the system of Np linear equations (2.133). The resulting computational complexity of $O((Np)^3)$ can be significant if N is particularly large. Several contributions have dealt with this problem in the machine learning literature: the proposed solutions mainly rely on approximations of the kernel function. These range from the use of the Nyström method (Zhang and Kwok, 2010) or of greedy algorithm (Smola and Schölkopf, 2000) to the truncation of the kernel eigen-decomposition (Zhu, Williams, Rohwer, and Morciniec, 1997; Rahimi and Recht, 2007). The latter approach has been also successfully applied in a system identification setting (Carli, Chiuso, and Pillonetto, 2012).

In the context of system identification, the practical approach of treating the impulse response as a finite-dimensional object allows to compute $\hat{\mathbf{g}}$ at a cost of $O\left((Np)(Tmp)^2\right)$ through the following rewriting of $\hat{\mathbf{g}}$ in equation (2.148)

$$\hat{\mathbf{g}} = (\Phi_N^\top \widetilde{\Sigma}_N^{-1} \Phi_N + \bar{K}_\eta^{-1})^{-1} \Phi_N^\top \widetilde{\Sigma}_N^{-1} Y_N \tag{2.193}$$

The plain analysis here conducted needs to be modified if so-called reweighted techniques are exploited (Chartrand and Yin, 2008; Daubechies, DeVore, Fornasier, and Güntürk, 2010), as will be done in the identification algorithm proposed in Section 3.4. These procedures require to iterate the hyper-parameters tuning and the impulse response estimation until a certain stopping condition is met.

2.4.5.2 Hyper-parameters Tuning

From a computational perspective, the hyper-parameters tuning constitutes the most involved step in the non-parametric Bayesian identification routine. Moreover, according to the procedure adopted for the tuning, the computational effort may vary significantly. The following discussion is mainly focused on the probabilistic approaches illustrated in

Section 2.4.3, since they are more popular within the system identification community.

Empirical Bayes. The Empirical Bayes paradigm illustrated in Section 2.4.3 involves the resolution of the constrained optimization problem (2.183). Under Gaussian assumptions on the noise and on the impulse response to be estimated, such problem reduces to the constrained minimization of function (2.184). The following discussion will specifically treat this case.

Resorting to numerical search routines (such as gradient or Netwton's methods) represents a natural way to minimize function (2.184). Compared to the optimization stage required by a Prediction Error Method (2.14), the marginal likelihood maximization (2.183) turns out to be a simpler problem, because of the smaller dimension of the search space: indeed, the number of hyper-parameters is typically much smaller than the size of the parameter vector θ, that is, $d_\eta < d_\theta$. However, some criticality arise when optimizing function (2.184). Firstly, the objective function is non-convex, thus leading to local minima matters; secondly, the computation of the Hessian may be costly; thirdly, the evaluation of the objective function and of its gradient may suffer of ill-conditioning and finally, when the number of data N is large, the matrix inversions appearing in (2.184) may be particularly inefficient. The latter two issues have been considered by Chen and Ljung (2013), in the FIR case (i.e. when a finite-length impulse response is estimated). They show that pointwise evaluation of (2.184) can be robustly and efficiently performed using the equivalent reformulation

$$\begin{aligned} f_{ML}(\eta) :&= -\ln p_y(Y_N|\eta) \\ &= Y_N^\top \widetilde{\Sigma}_N^{-1} Y_N - Y_N^\top \widetilde{\Sigma}_N^{-1} \Phi_N (\bar{K}_\eta^{-1} + \Phi_N^\top \widetilde{\Sigma}_N^{-1} \Phi_N)^{-1} \Phi_N^\top \widetilde{\Sigma}_N^{-1} Y_N \\ &\quad + \ln\det(\widetilde{\Sigma}_N) + \ln\det(\bar{K}_\eta) + \ln\det(\bar{K}_\eta^{-1} + \Phi_N^\top \widetilde{\Sigma}_N^{-1} \Phi_N) \qquad (2.194) \\ &= Y_N^\top \widetilde{\Sigma}_N^{-1} Y_N - Y_N^\top \widetilde{\Sigma}_N^{-1} \Phi_N L (I_{Tmp} + L^\top \Phi_N^\top \widetilde{\Sigma}_N^{-1} \Phi_N L)^{-1} L^\top \Phi_N^\top \widetilde{\Sigma}_N^{-1} Y_N \\ &\quad + N(\sum_{i=1}^{p} \ln\sigma_i) + \ln\det(I_{Tmp} + L^\top \Phi_N^\top \widetilde{\Sigma}_N^{-1} \Phi_N L) \qquad (2.195) \end{aligned}$$

Equation (2.194) exploits the matrix inversion lemma and the Sylvester's determinant identity (Harville, 1998), while in (2.195) the Cholesky decomposition $\bar{K}_\eta := LL^\top$ is used. By means of expression (2.194), pointwise evaluation of (2.184) takes $O(Np(Tmp)^2 + (Tmp)^3)$.

Concerning the second issue above mentioned, the computation of the Hessian can be avoided by resorting to *quasi-Netwon* methods, which replace the Hessian by suitable approximations (see also the discussion in Section 2.2.4). Such methods are iterative routines which update the hyper-parameters according to the rule

$$\eta^{(k+1)} = \eta^{(k)} - \alpha^{(k)}[H_N^{(k)}]^{-1}[f'_{ML}(\eta^{(k)})]^\top \tag{2.196}$$

where $f'_{ML}(\cdot)$ denotes the gradient of $f_{ML}(\cdot)$, while $H_N^{(k)}$ denotes an approximation of the Hessian of the objective function (Nocedal and Wright, 2006).

A recent contribution has proposed a version of the so-called *Scaled Gradient Projection (SGP)* algorithm, which has been adapted to the minimization of function $f_{ML}(\eta)$ in (2.194) (Bonettini, Chiuso, and Prato, 2015). Despite the theoretical linear convergence of such type of routines, this tailored version has proved to be superior to second order state-of-the-art methods in terms of computational effort. Such outcomes will be further confirmed by the results reported in Section 3.5.

Algorithm 1 reports the pseudo-code of the SGP version proposed by Bonettini et al. (2015). Line 5 represents the crucial step of the routine: at each iteration the hyper-parameters $\eta^{(k)}$ are updated through a gradient scaling involving a scalar $\alpha^{(k)}$ and the diagonal matrix $D^{(k)} \in \mathbb{R}^{d_\eta \times d_\eta}$; such candidate update is then projected onto the feasible set D_η through the projection operator

$$\Pi_{D_\eta, W} = \underset{x \in D_\eta}{\arg\min} (x-z)^\top W (x-z) \tag{2.197}$$

Algorithm 1 Scaled Gradient Projection (SGP) Algorithm

```
 1: **Initialization:** Choose the starting point $\eta^{(0)} \in D_\eta$.
 2: Set the parameters $\kappa$, $\rho \in (0,1)$, $0 < \alpha_{min} < \alpha_{max}$, $0 < L_{min} < L_{max}$.
 3: **for** $k = 0, 1, 2...$ **do**
 4:     Choose $\alpha^{(k)} \in [\alpha_{min}, \alpha_{max}]$ and the diagonal scaling matrix $D^{(k)}$ such that $L_{min} < \left[D^{(k)}\right]_{ii} < L_{max}$, $i = 1, .., d_\eta$.
 5:     Projection:
            $$x^{(k)} = \Pi_{D_\eta, D^{(k)^{-1}}} \left(\eta^{(k)} - \alpha^{(k)} D^{(k)} [f'_{ML}(\eta^{(k)})]^\top\right)$$
 6:     Descent direction: $\Delta\eta^{(k)} = x^{(k)} - \eta^{(k)}$.
 7:     Set $\epsilon = 1$.
 8:     **if** $f_{ML}(\eta^{(k)} + \epsilon\Delta\eta^{(k)}) \le f_{ML}(\eta^{(k)}) + \kappa\epsilon f'_{ML}(\eta^{(k)})\Delta\eta^{(k)}$ **then**
 9:         Go to step 13.
10:     **else**
11:         Set $\epsilon = \rho\epsilon$ and go to step 8.
12:     **end if**
13:     Set $\eta^{(k+1)} = \eta^{(k)} + \epsilon\Delta\eta^{(k)}$.
14: **end for**
```

The stepsize $\alpha^{(k)}$ is chosen by means of an alternation strategy based on the Barzilai-Borwein rules (Barzilai and Borwein, 1988), which aims at finding $\alpha^{(k)}$ such that $\alpha^{(k)}D^{(k)}$ approximates the inverse Hessian of the objective function. Specifically, at each iteration k, $\alpha^{(k)}$ is set equal to one of the two values

$$\alpha_1^{(k)} = \frac{r^{(k-1)^\top} D^{(k)^{-1}} D^{(k)^{-1}} r^{(k-1)}}{r^{(k-1)^\top} D^{(k)^{-1}} w^{(k-1)}}, \qquad \alpha_2^{(k)} = \frac{r^{(k-1)^\top} D^{(k)} w^{(k-1)}}{w^{(k-1)^\top} D^{(k)} D^{(k)} w^{(k-1)}} \tag{2.198}$$

where

$$r^{(k-1)} := \eta^{(k)} - \eta^{(k-1)}, \qquad w^{(k-1)} := [f'_{ML}(\eta^{(k)}) - f'_{ML}(\eta^{(k-1)})]^\top \tag{2.199}$$

The quantities (2.198) are respectively the solutions of the two problems

$$\min_{\alpha\in\mathbb{R}} \|(\alpha D^{(k)})^{-1} r^{(k-1)} - w^{(k-1)}\|, \qquad \min_{\alpha\in\mathbb{R}} \|r^{(k-1)} - \alpha D^{(k)} w^{(k-1)}\| \tag{2.200}$$

Algorithm 2 provides a detailed description of the alternation procedure developed by Bonettini et al. (2015) for the selection of the stepsize.

Algorithm 2 Barzilai-Borwein Alternation Strategy

```
Inputs: $\tau^{(k)}, r^{(k-1)}, w^{(k-1)}$
Set $0 < \alpha_{min} < \alpha_{max}$
$\alpha_1 \leftarrow \left(r^{(k-1)^\top} D^{(k)^{-1}} r^{(k-1)}\right) / \left(r^{(k-1)^\top} D^{(k)^{-1}} D^{(k)^{-1}} w^{(k-1)}\right)$
$\alpha_2 \leftarrow \left(r^{(k-1)^\top} D^{(k)} w^{(k-1)}\right) / \left(w^{(k-1)^\top} D^{(k)} D^{(k)} w^{(k-1)}\right)$
$\tilde{\alpha}_1 \leftarrow \min\{\max\{\alpha_{min}, \alpha_1\}, \alpha_{max}\}$
$\tilde{\alpha}_2 \leftarrow \min\{\max\{\alpha_{min}, \alpha_2\}, \alpha_{max}\}$
if $\tilde{\alpha}_2/\tilde{\alpha}_1 \leq \tau^{(k)}$ then
    $\alpha^{(k)} \leftarrow \tilde{\alpha}_2$
    $\tau^{(k+1)} \leftarrow 0.9\tau^{(k)}$
else
    $\alpha^{(k)} \leftarrow \tilde{\alpha}_1$
    $\tau^{(k+1)} \leftarrow 1.1\tau^{(k)}$
end if
Return: $\alpha^{(k)}$, $\tau^{(k+1)}$
```

The choice of the scaling matrix $D^{(k)}$ strictly depends on the objective function $f_{ML}(\eta)$ and on the structure of the constraint set D_η. The definition of $D^{(k)}$ proposed in Bonettini et al. (2015) exploits the following decomposition of the gradient of $f(\eta)$

$$f'_{ML}(\eta) = V(\eta) - U(\eta), \qquad V(\eta) > 0, \quad U(\eta) \geq 0 \tag{2.201}$$

The specific choice of $D^{(k)}$ is here reported only for non-negative constraints on η, namely $D_\eta = \mathbb{R}_+^{d_\eta}$. The interested reader is referred to Bonettini et al. (2015) for the dealing of box constraints on η.
By means of the splitting (2.201), the first order optimality conditions for the i-th component of η,

$$\eta_i[f'_{ML}(\eta)]_i = 0, \qquad \eta \geq 0, \quad [f'_{ML}(\eta)]_i \geq 0 \tag{2.202}$$

can be rewritten as the fixed point equation $\eta_i = \eta_i U_i(\eta) V_i(\eta)^{-1}$, thus suggesting the following update for $\eta_i^{(k)}$:

$$\eta_i^{(k+1)} = \eta_i^{(k)} \frac{U_i(\eta^{(k)})}{V_i(\eta^{(k)})} = \eta_i^{(k)} - \frac{\eta^{(k)}}{V_i(\eta^{(k)})} \left[f'_{ML}(\eta^{(k)}) \right]_i \tag{2.203}$$

It follows that the scaling matrix $D^{(k)}$ could be defined as

$$[D^{(k)}]_{ii} = \min \left(\max \left(L_{min}, \frac{\eta^{(k)}}{V_i(\eta^{(k)})} \right) L_{max} \right) \tag{2.204}$$

When Gaussianity does not hold or when an informative prior is postulated also for the hyper-parameters, the maximization of the evidence function is more involved than what has been described so far. In such cases, the *Expectation-Maximization (EM)* algorithm represents a valid alternative to gradient methods (Bottegal, Aravkin, Hjalmarsson, and Pillonetto, 2016). EM is a widely used algorithm for the optimization of likelihood functions in presence of latent variables (Dempster et al., 1977; McLachlan and Krishnan, 2007). When maximizing the marginal likelihood, the impulse response $\mathbf{g}$ plays the role of the latent variable in the complete likelihood function $p_{y\mathbf{g}}(Y_N, \mathbf{g}|\eta)$. For simplicity, the FIR implementation of Bayesian approaches is here considered.
The EM algorithm exploits the following decomposition of the evidence function (Bishop, 2006):

$$\ln p_y(Y_N|\eta) = \mathfrak{L}(q(\mathbf{g}), \eta) + KL(q(\mathbf{g})||p_{\mathbf{g}}(\mathbf{g}|Y_N, \eta)) \tag{2.205}$$

$$\mathfrak{L}(q(\mathbf{g}), \eta) : = \int_{\mathbb{R}^{Tmp}} q(\mathbf{g}) \ln \left\{ \frac{p_{y\mathbf{g}}(Y_N, \mathbf{g}|\eta)}{q(\mathbf{g})} \right\} d\mathbf{g} \tag{2.206}$$

$$KL(q(\mathbf{g})||p_{\mathbf{g}}(\mathbf{g}|Y_N, \eta)) = - \int_{\mathbb{R}^{Tmp}} q(\mathbf{g}) \ln \left\{ \frac{p_{\mathbf{g}}(\mathbf{g}|Y_N, \eta)}{q(\mathbf{g})} \right\} d\mathbf{g} \tag{2.207}$$

where $\mathfrak{L}(q, \eta)$ denotes a lower bound for $\ln p_y(Y_N|\eta)$ based on the distribution $q(\mathbf{g})$, while $KL(\cdot||\cdot)$ denotes the Kullback-Leibler divergence between two probability distributions.

The EM algorithm finds the optimal value for η by keeping alternating between two steps, namely the Expectation (E) and the Maximization (M) steps, until convergence is reached. At the k-th iteration, the E-step computes the lower bound $\mathfrak{L}(q(\mathbf{g}), \eta)$ as

$$\mathfrak{L}\left(p(\mathbf{g}|Y_N, \eta^{(k)}), \eta\right) = \mathbb{E}\left[\ln \frac{p_y(Y_N|\mathbf{g}, \eta)p_{\mathbf{g}}(\mathbf{g}|\eta)}{p_{\mathbf{g}}(\mathbf{g}|Y_N, \eta^{(k)})}\right] \tag{2.208}$$

where the expectation is taken w.r.t. $p(\mathbf{g}|Y_N, \eta^{(k)})$. Notice that this step corresponds to solve

$$\mathfrak{L}\left(p(\mathbf{g}|Y_N\eta^{(k)}), \eta\right) = \max_{q(\mathbf{g})} \ \mathfrak{L}(q(\mathbf{g}), \eta^{(k)}) \tag{2.209}$$

since $KL(q(\mathbf{g})||p(\mathbf{g}|Y_N, \eta)) = 0$ when $q(\mathbf{g})$ is the posterior distribution computed for $\eta^{(k)}$. The M-step of the EM algorithm updates the hyper-parameters value according to:

$$\eta^{(k+1)} = \arg\max_{\eta \in D_\eta} \ \mathfrak{L}(p(\mathbf{g}|Y_N, \eta^{(k)}), \eta) \tag{2.210}$$

Algorithm 3 reports the pseudo-code of the EM algorithm adapted to solve problem (2.183). Such routine is guaranteed to converge to a stationary point of the evidence function which has to be maximized. Furthermore, except for very unlucky initializations, it will converge to a local (or global) optimum of the likelihood function.

Algorithm 3 EM Algorithm to optimize $p_y(Y_N|\eta)$

```
**Initialization:** Choose the starting point $\eta^{(0)} \in D_\eta$
**for** $k = 0, 1, 2, ...$ **do**
    *E-step:* Compute $\mathfrak{L}\left(p(\mathbf{g}|Y_N, \eta^{(k)}), \eta\right)$ as in (2.208)
    *M-step:* $\eta^{(k+1)} \leftarrow \arg\max_{\eta \in D_\eta} \mathfrak{L}(p(\mathbf{g}|Y_N, \eta^{(k)}), \eta)$
**end for**
**Return:** $\hat{\eta}$
```

Remark 2.4.4. Appendix B highlights a connection between the EM routine and gradient algorithms, arising when $\bar{K}_\eta = \eta\bar{K}$, $\eta \in \mathbb{R}_+$, i.e. when only a scaling factor needs to be estimated. $\bar{K}$ is here assumed to be a fixed matrix. Under the same assumption, a connection between the EM and reweighted algorithms discussed by Wipf and Nagarajan (2010) is drawn.

Full Bayes. The Full Bayes approach illustrated in Section 2.4.3.1 relies on the design of a stochastic simulation algorithm (such as an MCMC) to draw samples from the hyper-parameters posterior $p(\eta|Y_N)$.

The MCMC routine exploits a proposal distribution, from which the samples are iteratively drawn; each sample is then kept or rejected by evaluating the PDF of $p(\eta|Y_N)$ at such sample. By means of this procedure a Markov chain having $p(\eta|Y_N)$ as stationary distribution is built; therefore, after a *burn-in* period consisting of N_{bi} iterations, the accepted samples are guaranteed to be distributed as $p(\eta|Y_N)$ (Gilks, 2005).

From the above description, it is clear that the implementation of an MCMC for drawing samples from $p(\eta|Y_N)$ requires to be able to evaluate $p_\eta(\eta|Y_N)$: recalling (2.182), it turns out that $p_\eta(\eta|Y_N)$ can be evaluated through the marginal likelihood $p_y(Y_N|\eta)$, apart from the normalization constant $p_y(Y_N)$. Algorithm 4 illustrates an MCMC algorithm designed to sample from $p(\eta|Y_N)$. Since it adopts a Gaussian (and thus symmetric) proposal, Algorithm 4 is actually a Metropolis-Hastings routine (Gilks, 2005). Concerning the initialization, $\eta^{(0)}$ can be set equal to the estimate $\hat{\eta}_{EB}$ computed in (2.183), while a typical choice for $\tilde{P}^{(0)}$ is

$$\tilde{P}^{(0)} = -\left[\frac{d^2 \ln[p_y(Y_N|\hat{\eta}_{EB})p_\eta(\hat{\eta}_{EB})]}{d\eta d\eta^T}\right]^{-1} \tag{2.211}$$

Multiple methodologies exist to set the burn-in period N_{bi} at step 2; a good overview is provided by Raftery and Lewis (1996).

In order to obtain a reliable approximation of the distribution $p(\eta|Y_N)$ (and in turn of the posterior $p(\mathbf{g}|Y_N)$), a large number of samples N_{sp} has to be drawn, meaning that a high number of iterations of Algorithm 4 has to be performed. This makes the *Full Bayes* approach for hyper-parameters tuning particularly inefficient in terms of computational effort, thus explaining the scarce popularity of such approach within the system identification community.

Algorithm 4 MCMC algorithm to draw samples from $p(\eta|Y_N)$

```
1: Initialization:  Set maximum number of iterations $N_{max}$
2: Initialization:  Set burn-in period $N_{bi}$
3: Initialization:  Choose the proposal distribution $\tilde{p}(\cdot)$: $p(\cdot) \sim \mathcal{N}(\eta^{(0)}, \tilde{P}^{(0)})$
4: for $i = 1, 2, ..., n_{max}$ do
5:     Sample $\eta$ from $\tilde{p}(\cdot|\eta^{(i-1)}) \sim \mathcal{N}(\eta^{(i-1)}, \tilde{P}^{(0)})$
6:     Sample $\upsilon$ from a uniform distribution on $[0, 1]$
7:     Set
           $\eta^{(i)} = \begin{cases} \eta & \text{if } \upsilon \leq \frac{p_y(Y_N|\eta)p_\eta(\eta)}{p_y(Y_N|\eta^{(i-1)})p_\eta(\eta^{(i-1)})} \\ \eta^{(i-1)} & \text{otherwise} \end{cases}$
8: end for
9: Return: $\{\eta^{(i)}\}_{i=N_{bi}+1}^{N_{max}}$
```

2.5 Model Selection and Validation

The discussion of the previous sections has highlighted how the implementation of the described system identification methods is necessarily accompanied by some user's choices. Some of them are strictly related to the specification of the model class within which the estimated model lies. Regarding PEM, these decisions mainly involve the choice of model complexity and of its parametrization, that is, the specification of the polynomials that will be estimated in the general transfer function model (2.15) as well as of their degrees. Subspace algorithms instead only require to fix the state-space size, since the parametrization is implicitly specified by the method itself. Finally, complexity selection in the non-parametric Bayesian paradigm is somehow performed through the hyper-parameters tuning described in Section 2.4.3: consequently, no clear-cut decision about the system order is left to the user, who simply needs to specify the kernel. However, such choice is not as crucial as the model structure selection required by parametric methods, since it has been proved that the space of functions associated with the standard kernels adopted by the system identification community is rich enough to include the impulse responses of any BIBO stable LTI system (Pillonetto and De Nicolao, 2010; Chen et al., 2012).

The selection of a specific model class fixes a trade-off between *flexibility* and *parsimony*: on the one hand, a complex model would allow a more accurate reproduction of the given data but, on the other hand, a simple model would be more handleable in its estimation stage and also in its eventual future use, guaranteeing better generalization capabilities (i.e. a better description of unseen data). In particular, the choice of a simple model structure would provide computational advantages during the estimation phase: for instance, with regard to PEM, Section 2.2.4 has pointed out how some of the transfer function models illustrated in Section 2.2.1 admit a linear representation of the predictor w.r.t the parameter vector, thus allowing the use of simple computational procedures (such as LS when a quadratic loss function is used). On the other hand, when a linear predictor is not available, more involved algorithms (e.g. iterative routines) need to be adopted.

In the statistical learning literature, the trade-off between flexibility and parsimony is traditionally formulated in terms of *bias* and *variance* of the derived estimator (Hastie et al., 2009; Burnham, Anderson, and Burnham, 2002). These two terms arise from a decomposition of the so-called *Mean Square Error* (MSE). Specifically, let $\mathcal{S}$ and $\widehat{\mathcal{M}}$ respectively denote the true system description and the model estimated trough a certain identification algorithm. The MSE for $\widehat{\mathcal{M}}$ is defined as

$$\begin{aligned} \mathrm{MSE}(\widehat{\mathcal{M}}) = \mathbb{E}[(\mathcal{S} - \widehat{\mathcal{M}})^2] &= (\mathcal{S} - \mathbb{E}[\widehat{\mathcal{M}}])^2 + \mathbb{E}[(\widehat{\mathcal{M}} - \mathbb{E}[\widehat{\mathcal{M}}])^2] \\ &=: \mathfrak{B}^2(\widehat{\mathcal{M}}) + \mathfrak{V}(\widehat{\mathcal{M}}) \end{aligned} \tag{2.212}$$

where $\mathfrak{B}(\cdot)$ represents the bias, i.e. the gap between the true system and the average of the estimates (the expectation is taken w.r.t. measurement noise in the data), while $\mathfrak{V}(\cdot)$ is the variance of the estimated model. Both of them are functions of the model complexity: while the bias decreases as it increases, a complex model leads to a large variance. The bias term can also be further decomposed as

$$\mathfrak{B}^2(\widehat{\mathcal{M}}) = (\mathcal{S} - \mathcal{M}^*)^2 + (\mathcal{M}^* - \mathbb{E}[\widehat{\mathcal{M}}])^2 \tag{2.213}$$

The first term is the so-called squared *model bias* (or *model error*), i.e. the error between the true system description and its closest approximation lying within the chosen model class M. The second term denotes the *estimation bias*, i.e. the gap between such optimal approximation $\mathcal{M}^*$ and the average of the estimated models (see also Hastie et al. (2009), Ch. 7). While model bias is a measure of the eventual inadequacy of the chosen model class M, the estimation bias may be due to little informative data or to the implementation of the identification routine (for instance, when iterative search routines are used, convergence to local minima of the objective function could give rise to estimation bias).
According to the previous considerations on non-parametric Bayesian methods, the stable-spline kernels commonly adopted in system identification define a model class M which guarantees a null model bias, thus making possible to recover the true system description $\mathcal{S}$ if the hyper-parameters η are suitably tuned. Concerning parametric methods, the quantification of the *model error* arising from the misspecification of the model class has been investigated by several contributions in the system identification literature, through the development of so-called *model error models* (Goodwin et al., 1992; Ljung, Goodwin, and Agüero, 2014).

For parametric methods, model class selection also determines the *identifiability* properties of the identification procedure. Such concept, which is also influenced by experimental conditions, has been widely treated in the system identification literature, receiving many different connotations. Since a thorough discussion of the topic is out of scope for this manuscript, the interested reader is referred to Ljung (1999), Sec. 4.5 and 4.6, Bellman and Åström (1970) and to the survey Nguyen and Wood (1982). Identifiability of multivariable model structures has been discussed in Ljung and Rissanen (1976); Kailath

(1980); Gevers and Wertz (1984).

The choices regarding the specification of a model class can be taken at different stages of an identification procedure, namely: (1) a preliminary data analysis may give some hints on the complexity of the system to be estimated; (2) a selection can be performed during the inference stage, by directly comparing different models or by some specific procedure connected with the chosen identification algorithm; (3) a post-processing analysis, known as *model validation*, may highlight some deficiencies of the estimated model and thus suggest to reconsider the choices done in the previous stages.

2.5.1 A Priori Model Class Selection

Pre-processing tools include the spectral analysis estimate (which may highlight the number of resonance peaks, thus suggesting the order of the system at hand), tests on the rank of the sample covariance matrices of past input and output data (Woodside (1971); Wellstead and Rojas (1982); Tse and Weinert (1975)) and canonical correlation analysis (Hotelling (1936)) to assess whether one more variable should be included or not in a model structure (Draper (1998); Larimore (1990)). A more detailed discussion on the mentioned tools can be found in Ljung (1999), Sec. 16.3.

2.5.2 Model Class Selection during the Estimation Stage

As observed throughout Section 2.4, the model class selection (specifically, the complexity choice) for non-parametric Bayesian methods is implicitly performed during the estimation stage through the hyper-parameters tuning. On the other hand, model class selection can be accomplished during the estimation phase also when parametric techniques are adopted. In particular, for PEM the choice is based on criteria which compare the generalization capabilities of different model classes, while subspace approaches exploit the information contained in the singular values computed in equations (2.78) and (2.89).

Prediction Error Methods. When adopting PEM, model class selection is performed through standard techniques inherited from the statistical learning literature Hastie et al. (2009). Specifically, the chosen model structure minimizes a certain approximation of the generalization error, i.e. the error observed on a new set of data (*validation data*):

$$(\hat{\theta}, \widehat{M}) = \underset{\theta \in D_\theta, M}{\arg\min}\ \widehat{\text{Err}}(\theta, M, \mathcal{D}^N) \tag{2.214}$$

The above equation highlights how such procedures allow to solve the joint problem of model class selection and parameter estimation. In particular, $\widehat{M}$ denotes the choice of a parametrization $\widehat{\mathcal{M}}(\cdot)$ and of the model complexity $\hat{d}_\theta$, that is $\widehat{M} = \{\hat{d}_\theta, \widehat{\mathcal{M}}(\cdot)\}$.
The approximations $\widehat{\mathrm{Err}}(\theta, M, \mathcal{D}^N)$ which are found in the literature can be grouped into two main families (Efron, 2012): on the one hand, the so-called *covariance penalties* and on the other hand, *cross-validation* and *bootstrap* methods. Covariance penalty approaches arise when quadratic error measures are used: they resort to the following approximation of the generalization error

$$\widehat{\mathrm{Err}}(\theta, M, \mathcal{D}^N) = \frac{1}{N}\sum_{t=1}^{N} \|y(t) - \hat{y}(t|\theta)\|_2^2 + \frac{2}{N}\sum_{t=1}^{N} \mathrm{Tr}\left\{\widehat{\mathrm{Cov}}(\hat{y}(t|\theta), y(t))\right\} \tag{2.215}$$

$$= \frac{1}{N}\sum_{t=1}^{N} \|y(t) - \hat{y}(t|\theta)\|_2^2 + \frac{2}{N}\mathrm{Tr}\left\{\widehat{\Sigma}\ \widehat{\mathrm{Df}}_{\hat{y}}(\theta)\right\} \tag{2.216}$$

where equation (2.216) derives from the extension provided by Ye (1998), with $\mathrm{Df}_{\hat{y}}(\theta)$ denoting the so-called *matricial degrees of freedom* of the predictor $\hat{y}(t|\theta)$ (see also definition (2.190)):

$$\mathrm{Df}_{\hat{y}}(\theta) = \sum_{t=1}^{N} \mathrm{Cov}(\hat{y}(t|\theta), y(t))\Sigma^{-1} \tag{2.217}$$

Hence, to obtain an approximation of the generalization error, the empirical squared prediction error on the estimation dataset is penalized with a term depending on the covariance between the obtained estimator and the given data. For predictors which are linear in the observations, i.e. $\hat{y}(t|\theta) = \Gamma(\theta)Y_N$, $\mathrm{Tr}\{\mathrm{Cov}(\hat{y}(t|\theta), y(t))\} = \mathrm{Tr}\{\Gamma(\theta)\} = d_\theta$ and (2.216) coincides with the C_p-statistics. Under Gaussian assumptions on the measurement noise, the C_p statistics coincides with the well-known *Akaike Information Criterion (AIC)* (Akaike, 1998).
An alternative approximation of the generalization error arises by resorting to Bayesian arguments, leading to the so-called *Minimum Description Length (MDL)* (or BIC) criterion (Rissanen, 1978):

$$\mathrm{BIC}(\theta, M, \mathcal{D}^N) = \frac{1}{N}\sum_{t=1}^{N} \|y(t) - \hat{y}(t|\theta)\|_2^2 + \frac{\ln N}{N} \cdot d_\theta \tag{2.218}$$

The second family of procedures, which include *cross-validation* and *bootstrap* methods estimate $\mathrm{Err}(\theta, M)$ by means of suitable resamplings of the given dataset.
So-called *parametric bootstrap* uses an available estimate $\hat{\theta}$ to build B new datasets

according to

$$\begin{aligned} \widetilde{\mathcal{D}}_{(i)}^N &= \{\tilde{y}_{(i)}(t), u(t)\}_{t=1}^N, & i &= 1, ..., B \\ \tilde{y}_{(i)}(t) &:= \hat{y}(t|\hat{\theta}) + \tilde{e}_{(i)}(t), & \tilde{e}_{(i)}(t) &\sim \mathcal{N}(0_p, \widehat{\Sigma}) \end{aligned} \tag{2.219}$$

where $u(t)$ denotes the given input data, $\hat{y}(t|\hat{\theta})$ is the predictor computed on the data $\mathcal{D}^N$ using the estimate $\hat{\theta}$ and $\widehat{\Sigma}$ is an available noise variance estimate. The model class is then selected according to criterion (2.215) with $\widehat{\mathrm{Cov}}(\hat{y}(t|\theta), y(t))$ computed as

$$\widehat{\mathrm{Cov}}(\hat{y}(t|\theta), y(t)) = \frac{1}{B-1}\sum_{i=1}^{B} \hat{y}(t|\hat{\theta}_{(i)})(\tilde{y}_{(i)}(t) - \bar{\tilde{y}}(t))^\top, \qquad \bar{\tilde{y}}(t) := \frac{1}{B}\sum_{i=1}^{B} \tilde{y}_{(i)}(t) \tag{2.220}$$

where $\hat{y}(t|\hat{\theta}_{(i)})$ denotes the predictor computed using dataset $\widetilde{\mathcal{D}}_{(i)}^N$.

The cross-validation procedure require to split the data $\mathcal{D}^N$ into two sets, $\mathcal{D}^{N_{tr}}$ and $\mathcal{D}^{N_{val}}$, respectively composed of N_{tr} and N_{val} samples. The approximated generalization error in equation (2.188) is then given by

$$\widehat{\mathrm{Err}}(\theta, M, \mathcal{D}^N) = \frac{1}{N_{val}}\sum_{t=1}^{N_{val}} \|y_{val}(t) - \hat{y}(t|\hat{\theta}_{tr})\|^2 \tag{2.221}$$

where $\hat{\theta}_{tr}$ denotes the parameters estimate computed using data $\mathcal{D}^{N_{tr}}$. Once the model class is chosen, the parameter vector can be re-estimated using all the available data $\mathcal{D}^N$. Another class of selection criteria resorts to sequential statistical tests based on the F-distribution (see Ljung (1999), Sec. 16.4 and Söderström and Stoica (1989), Ch. 11).

Subspace Methods. Differently from PEM which require to select also a suitable parametrization $\mathcal{M}(\cdot)$ of the model class M, subspace algorithms simply need to fix a certain state-space size n. This is typically accomplished by inspecting the singular values computed in equations (2.78) and (2.89). Let denote them as $\hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \cdots$. If n_0 is the true system order, then, under suitable assumptions on the data generating process and on the weighting matrices W_1, W_2:

$$\begin{aligned} &\lim_{N\to\infty} \hat{\sigma}_i = \sigma_i, & i &= 1, ..., n_0 \\ &\lim_{N\to\infty} \hat{\sigma}_i = 0, & i &> n_0 \end{aligned}$$

Such observation has been exploited to derive the following general selection criteria, respectively proposed by Peternell (1995) and Bauer (2001):

$$NIC(n) = \sum_{i=n+1}^{v} \hat{\sigma}_i^2 + \frac{C(N)d(n)}{N} \tag{2.222}$$

$$SVC(n) = \hat{\sigma}_{n+1}^2 + \frac{C(N)d(n)}{N} \tag{2.223}$$

In the above equations $d(n) = n(m+p) + np + pm$ denotes the number of parameters in the state-space model, while $v = \min\{rp, (p+m)s\}$, with r and s being the future and past horizons, respectively. Furthermore, $C(N)$ is a penalty term chosen so that $C(N)/N \to 0$ as $N \to \infty$.

The `n4sid` routine implemented in the MATLAB System Identification Toolbox (Ljung, 2007) selects the index of the singular value which in logarithm is closest to the logarithmic mean of the maximum and minimum singular values.

From a computational point of view, it should be noticed that the criteria here illustrated significantly differ from those described for PEM: while $IVC(n)$ and $SVC(n)$ simply require to compute one SVD, the criteria adopted by PEM demand the estimation of several models, thus resulting computationally expensive.

Alternative approaches exist which consist in sequential tests (Sorelius, 1999; Camba-Mendez and Kapetanios, 2001) or on criteria resembling the AIC, which exploit an estimate of the innovation covariance (Bauer, 2001, 2005).

2.5.3 Model Validation

The principal goal of model validation is to check whether the estimated model achieves the desired performance in the applications it was designed for: for instance, if the intended use of the model was controller design, the performance of the designed closed-loop system are evaluated. Model validation also aims at assessing whether the estimated model is too complex: to this purpose, the confidence intervals built around the estimate may be evaluated (see also chapter 4); alternatively, the approximation of the inferred model with a simpler one may reveal an unnecessary over-parametrization (the well-known zero-pole cancellation technique proposed by Söderström (1975) may e.g. be applied). Model reduction will be further discussed in chapter 6.

An important class of model validation methods is based on the analysis of the residuals, i.e. of the part of the data that is not caught by the estimated model:

$$\varepsilon(t, \widehat{\mathcal{M}}) = y(t) - \hat{y}(t|\widehat{\mathcal{M}}), \qquad t = 1, .., N \tag{2.224}$$

Statistical tests are performed in order to assess their whiteness and their independence from the given dataset $\mathcal{D}^N$. Indeed, if correlation among $\varepsilon(t, \widehat{\mathcal{M}})$ and $\varepsilon(t-\tau, \widehat{\mathcal{M}})$ is detected for $\tau > 0$, it is reasonable to think that part of $\varepsilon(t)$ could have been better predicted from past data. Analogously, if independence from $\mathcal{D}^N$ is verified, it is probable that the model would be able to correctly reproduce also unseen data.
Residuals whiteness is assessed through a statistical test on the sample covariance

$$\hat{R}^N_{\varepsilon_i}(\tau, \widehat{\mathcal{M}}) = \frac{1}{N} \sum_{t=1}^{N} \varepsilon_i(t, \widehat{\mathcal{M}}) \varepsilon_i(t-\tau, \widehat{\mathcal{M}}) \tag{2.225}$$

where $\varepsilon_i(t, \widehat{\mathcal{M}})$ denotes the residual on the i-th output component. If $\{\varepsilon_i(t, \widehat{\mathcal{M}})\}$ is a white noise zero-mean sequence with variance σ_i, then it can be proved that

$$\frac{N}{\sigma_i} \sum_{\tau=1}^{\bar{\tau}} \left(\hat{R}^N_{\varepsilon_i}(\tau, \widehat{\mathcal{M}}) \right)^2 \tag{2.226}$$

is asymptotically $\chi^2(\bar{\tau})$-distributed. Therefore, let $\chi^2_d(\cdot)$ denote the quantile function of the χ^2-distribution with d degrees of freedom: $\chi^2_d(p)$ is equal to the value x for which $\Pr(\chi^2(d) \leq x) = p$. The null hypothesis stating residuals whiteness for model $\widehat{\mathcal{M}}$ is accepted if

$$\zeta^{N,\bar{\tau}}_{\varepsilon_i}(\widehat{\mathcal{M}}) = \frac{N}{\left(\hat{R}^N_{\varepsilon_i}(0, \widehat{\mathcal{M}}) \right)^2} \sum_{\tau=1}^{\bar{\tau}} \left(\hat{R}^N_{\varepsilon_i}(\tau, \widehat{\mathcal{M}}) \right)^2 < \chi^2_{\bar{\tau}}(1-\alpha), \qquad i = 1, ..., p \tag{2.227}$$

where α is the so-called *significance level* of the test (also called type-I risk). A typical value for α is 0.05. An independent test is performed on each output channel, because of the diagonal structure assumed for the noise covariance matrix Σ. The interested reader is referred to Ljung (1999) (Sec. 16.6) and to Söderström and Stoica (1989) (Sec. 11.2) for a more detailed illustration of such test.
Following a similar reasoning, the independence from the given dataset can be verified by studying the covariance between residuals and past input:

$$\hat{R}^N_{\varepsilon_i, u_j}(\tau, \widehat{\mathcal{M}}) = \frac{1}{N} \sum_{t=1}^{N} \varepsilon_i(t, \widehat{\mathcal{M}}) u_j(t-\tau) \tag{2.228}$$

where $u_j(t)$ denotes the value measured at time t at the j-th input channel. The test is based on the following quantities

$$\zeta^{N,\bar{\tau}}_{\varepsilon_i,u_j}(\widehat{\mathcal{M}}) = N r_{ij}^\top \left[\hat{R}^N_{\varepsilon_i}(0,\widehat{\mathcal{M}})\hat{R}^N_{u_j}\right]^{-1} r_{ij}, \qquad i=1,...,p, \quad j=1,...,m \tag{2.229}$$

where

$$\hat{R}_{u_j} = \frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix} u_j(t-1) \\ u_j(t-2) \\ \vdots \\ u_j(t-\bar{\tau})\end{bmatrix}\begin{bmatrix} u_j(t-1) & u_j(t-2) & \cdots & u_j(t-\bar{\tau})\end{bmatrix}$$

$$r_{ij} := \frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix} u_j(t-\xi-1) \\ u_j(t-\xi-2) \\ \vdots \\ u_j(t-\xi-\bar{\tau})\end{bmatrix}\varepsilon_i(t,\widehat{\mathcal{M}})$$

Specifically, model $\widehat{\mathcal{M}}$ passes the test on the independence between its residuals and past input data in $\mathcal{D}^N$ if

$$\zeta^{N,\bar{\tau}}_{\varepsilon_i,u_j}(\widehat{\mathcal{M}}) < \chi^2_{\bar{\tau}}(1-\alpha), \qquad i=1,...,p, \quad j=1,...,m \tag{2.230}$$

for a specified significance level α. More details on such tests are given in Ljung (1999) (Ch. 16) and Söderström and Stoica (1989) (Ch. 11).

A sequential application of such tests represents a practical way of exploiting them: specifically, models of increasing complexities should be tested until the null hypothesis is accepted for one of them.

2.6 Bibliographical Notes

2.6.1 System Identification Problem

The term *system identification* was coined by Zadeh (1956), whereas the concepts of model set, model structure and identification methods were first discussed by Zadeh (1962) and Ljung (1976). A quite recent overview of the achievements reached by the system identification community and of the existing open questions has been given by Ljung (2010).

Section 2.1 has introduced the distinction between parametric and non-parametric approaches to system identification. Concerning the latter, some references of early works in the field are listed in Söderström and Stoica (1989) (Ch. 3). Other overviews have been provided by Rake (1980, 1987) and Ljung and Glover (1981).

2.6.2 Prediction Error Methods

The roots of Prediction Error Methods in the system identification field go back to the seminal paper Åström and Bohlin (1966), which imported from the time series literature the Maximum Likelihood approach for the estimation of parameters of difference equation models. In the time series literature early references on such approaches include Cramér (1945); Grenander (1950); Whittle (1953). The works by Box and Jenkins (1970) and by Åström and Eykhoff (1971) provide a comprehensive survey of the identification methods developed at that time. Until the beginning of the nineties, the interest of the system identification community was mainly focused on Prediction Error Methods, thus making them a well-established techniques even in practical applications. The large attention devoted to such approaches in the classical textbooks Söderström and Stoica (1989) and Ljung (1999) is a clear proof of their impact into the system identification community. Other comprehensive treatments of PE methods can be found in Brockwell and Davis (2013) and Hannan and Deistler (1988).
Frequency-domain Prediction Error Methods have been also largely investigated in the literature: see e.g. Pintelon, Guillaume, Rolain, Schoukens, Van Hamme, et al. (1994); Pintelon and Schoukens (2012). A comparison between frequency- and time- domain approaches has been conducted by Ljung (2006).

2.6.3 Subspace Methods

Subspace methods originate from the state-space theory developed in the 1960s. In particular, the work of Ho and Kalman (1966) is considered the main contribution for the origin of such approaches. By extending the work of Akaike (1974), they provided a solution for determining the minimal state-space representation from impulse response data. Refinements of such theory were provided by Zeiger and McEwen (1974) and Kung (1978). Until the end of the Nineties, such techniques did not receive a significant attention from the system identification community, because of the difficulty in the treatment of data containing also a measured input. Such obstacle was overcome at the beginning of the Nineties, when several research teams proposed different solutions: the survey by Viberg (1995) distinguishes between *realization-based* subspace methods , *direct* subspace algorithms (De Moor et al., 1988; Verhaegen, 1991) and *instrumental variable* techniques (Verhaegen, 1993b, 1994; Van Overschee and De Moor, 1994). Van Overschee (1995) and McKelvey (1995) provide reviews of such early works on subspace identification, while

Van Overschee and De Moor (2012) is based on the unifying framework briefly reviewed in Section 2.3.2. A recent overview is also provided in Verhaegen and Verdult (2007).
The time series case has been dealt by Aoki (1990); Van Overschee and De Moor (1993); Deistler, Peternell, and Scherrer (1995).
The identification of Linear Parameter Varying (LPV) systems is considered by Verdult and Verhaegen (2002), while an overview of the application of subspace algorithms for the estimation on non-linear systems is given in Verdult (2002).
Frequency-domain identification is treated e.g. by McKelvey (1995); McKelvey, Akçay, and Ljung (1996); Van Overschee and De Moor (1996).
The extension of subspace algorithms to systems operating in closed-loop has been treated by Verhaegen (1993a); Ljung and McKelvey (1996); Qin and Ljung (2003); Chiuso and Picci (2005) and Chiuso (2010).

2.6.4 Non-Parametric Bayesian Methods

Non-parametric Bayesian methods (or equivalently, kernel-based approaches) have been introduced into the system identification community by the seminal paper Pillonetto and De Nicolao (2010) and further developed by the follow-up papers Pillonetto et al. (2011a) and Chen et al. (2012). Most of the research in this area has focused on the design of the kernel K_η (Dinuzzo, 2015; Chen and Ljung, 2014; Chen et al., 2014) and on the understanding of the properties of the Empirical Bayes estimator (Aravkin et al., 2012; Pillonetto and Chiuso, 2015). Stability issues have been considered by Pillonetto, Chen, Chiuso, Ljung, and Nicolao (2016) and Romeres, Pillonetto, and Chiuso (2015).
Recent surveys on regularization methods for system identification have been published: Pillonetto et al. (2014) focus on the connection with regularization in Reproducing Kernel Hilbert Spaces, while Chiuso (2016) provides several connections with econometrics and time-series literature.
Frequency-domain extensions of such techniques have been proposed by Bottegal and Pillonetto (2013) and Lataire and Chen (2016), while Pillonetto, Quang, and Chiuso (2011b) and Risuleo, Bottegal, and Hjalmarsson (2015) have applied such methods for the identification of non-linear systems.
Extensions to the field of network identification have been considered by Chiuso and Pillonetto (2012) and Zorzi and Chiuso (2015).
Several contributions relying on the Bayesian paradigm for time series estimation exist in the econometrics literature: see e.g. Doan, Litterman, and Sims (1984); De Mol, Giannone, and Reichlin (2008); Knox, Stock, and Watson (2001); Giannone, Lenza, and Primiceri (2015).

A classical reference for Gaussian Processes Regression is the book by Rasmussen and Williams (2006), while the theory of RKHS is developed in Aronszajn (1950). Applications of such theory in the machine learning field have been widely dealt by Cucker and Smale (2002); Schölkopf and Smola (2002); Wahba (1990).

3
Prior Design

Chapter 2 has highlighted how the non-parametric Bayesian methods admit an equivalent interpretation in terms of regularization. According to such interpretation, the prior designed by the user following the Bayesian paradigm acts as a penalty term in the regularization framework. In the previous chapter no details have been given about how the prior (or equivalently, the regularizer) should be designed in order to properly account for desired properties of the impulse response to be estimated. The current chapter intends to provide an overview of the priors commonly adopted when the methods illustrated in Section 2.4 are applied in system identification. To draw connections with the other identification approaches, the regularization perspective will be taken in the first part of the chapter: indeed, the role played by regularization in PEM and subspace techniques will be also discussed. Accordingly, the estimation problem here considered takes the general form

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{X}} J_F(\mathbf{x}, \mathcal{D}^N) + J_R(\mathbf{x}, \eta) \tag{3.1}$$

where $\mathbf{x}$, lying in the inner product space $\mathcal{X}$, represents an unknown quantity related to the system description which needs to be estimated. $\mathbf{x}$ could e.g. denote the parameter vector $\theta \in D_\theta$ for PEM, the impulse response function $g(\cdot)$ for the kernel-based regularization methods discussed in Section 2.4.1.2-2.4.2.2, or the impulse response vector $\mathbf{g} \in \mathbb{R}^{pmT}$ for the regularized LS techniques of Sections 2.4.1.3-2.4.2.3.

According to problem (3.1), $\mathbf{x}$ is estimated by trading-off the data fitting term $J_F(\mathbf{x}, \mathcal{D}_N)$ and the regularization term $J_R(\mathbf{x}, \eta)$, which acts as a penalty discouraging certain undesired solutions. $J_R(\mathbf{x}, \eta)$ depends on some regularization parameters η (called hyper-parameters in the Bayesian framework), which have to be tuned using the available data. Consequently, regularization deals with the well-known bias/variance trade-off using a continuous set of regularization parameters.
Historically, regularization was introduced to render the inverse problem of finding $\mathbf{x}$ from the measured data well-posed: indeed, problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{X}} J_F(\mathbf{x}, \mathcal{D}^N) \tag{3.2}$$

is ill-posed if $\mathbf{x}$ denotes e.g. a function belonging to a suitable Hilbert space; if $\mathbf{x}$ represents a finite-dimensional object (e.g. a vector or a matrix), problem (3.2) may be ill-posed, unless the number of data N is much larger than the size of $\mathbf{x}$. To overcome this issue, the so-called Tykhonov regularization was proposed by setting $J_R(\mathbf{x}, \eta) = \eta\|\mathbf{x}\|^2_{\mathcal{X}}$, with $\|\cdot\|_{\mathcal{X}}$ denoting the norm associated to the inner product defined in $\mathcal{X}$ (Tikhonov and Arsenin, 1977; Hoerl and Kennard, 1970).

Extensions of this basic regularization have been proposed in the statistical learning literature. To keep the general notation of problem (3.1), $J_R(\mathbf{x}, \eta)$ will be specified as a function of a bounded linear operator $A : \mathcal{X} \to \mathcal{Y}$, that is $J_R(\mathbf{x}, \eta) = f_R\left(A(\mathbf{x}), \eta\right)$. A large attention has been devoted to the development of penalty functions which favour certain structures on $A(\hat{\mathbf{x}})$, e.g. which force some elements of $A(\hat{\mathbf{x}})$ to be zero or equivalently, which enforce *sparsity* in $A(\hat{\mathbf{x}})$. To this purpose, Tibshirani (1996) proposed to set $f_R(\cdot)$ equal to the L^1-norm, i.e. $J_R(\mathbf{x}, \eta) = \eta \|A(\mathbf{x})\|_1$, thus guaranteeing the convexity of the optimization problem (3.1). The seminal work of Tibshirani (1996) gave rise to the broad family of so-called *LASSO* estimators, i.e. of learning algorithms relying on L^1-type penalties.

Another formulation of $J_R(\mathbf{x}, \eta)$ adopts the so-called *nuclear norm* (also known as *Schatten 1-norm*), defined as

$$\|A\|_* := \sum_{i=1}^{\infty} \sigma_i(A) \tag{3.3}$$

where $\sigma_i(A)$ denotes the i-th singular value of A. Consequently, the penalty $J_R(\mathbf{x}, \eta) = \eta \|A(\hat{\mathbf{x}})\|_*$ will induce sparsity on the singular values of $A(\mathbf{x})$. This type of regularizer has been introduced by Fazel, Hindi, and Boyd (2001) as a convex surrogate to the rank function in matrix rank minimization problems. Fazel et al. (2001) also prove that the nuclear norm $\|A\|_*$ is the convex envelope of the rank of A on the ball $\|A\|_2 < 1$. The quality of the nuclear norm heuristic as a replacement of the rank function has been analytically proved for certain applications (such as low-rank matrix completion) (Candès and Recht, 2009; Recht, Fazel, and Parrilo, 2010); moreover, it has been empirically observed that minimum nuclear norm solutions often have low rank.

The aforementioned types of regularization have found application also in the system identification field. Specifically, Tykhonov regularization has been used to overcome the issue of ill-posedeness, as well as to equip the estimator with smoothness and stability properties. LASSO penalties have been considered mainly for the problem of structure detection. For instance, in a MIMO system, a certain output may be affected by only a subset of the corresponding inputs: hence, a sparsity inducing estimator would set to zero unnecessary model components and simply estimate the relevant ones. The adoption of nuclear norm regularization in system identification is instead connected to a well-known property coming from realization theory, which appear relevant especially for MIMO systems. Indeed, as mentioned in Section 2.2.1, the order n of a minimal state-space realization

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \qquad x(t) \in \mathbb{R}^n \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{3.4}$$

equals the McMillan degree of the impulse response g. In turn it equals the rank of the block-Hankel matrix $\mathbf{G} \in \mathbb{R}^{pn \times mn}$ built with the Markov coefficients $g(k) = CA^{k-1}B + D$, which are expressed through a relation coupling the impulse responses $g_{ij}(k)$ (with i and j denoting different input and output channels). Consequently, imposing a nuclear norm penalty on $\mathbf{G}$ allows to account for this coupling, by controlling the complexity (measured in terms of McMillan degree) of the estimated system.

The use of these regularizers in system identification will be clarified in the next sections. Specifically, Section 3.1 will focus on PEM, while Section 3.2 is devoted to the role played by regularization in connection with subspace algorithms. Finally, Section 3.3 will describe the more common penalties adopted in the non-parametric estimation illustrated in Section 2.4. According to the connections drawn in Section 2.4, the design of such penalties could be equivalently interpreted as the design of the prior distribution in a Bayesian framework; however, Section 3.3 will be based on a regularization perspective in order to highlight the connections with the methods illustrated in the preceding sections.

The innovative contribution of this chapter is described in Section 3.4, where the Bayesian framework of Section 2.4 is exploited to derive a prior inducing a joint L^1 and L^2 penalty, thus controlling at the same time complexity, stability and smoothness of the estimated models. Exploiting such prior, an iterative identification algorithm is developed. Section 3.5 contains an extensive numerical comparison between the newly introduced identification algorithm and several classical methods, including those illustrated in Sections 3.1-3.3.

3.1 Regularization in Prediction Error Methods

3.1.1 ℓ_2 Regularization

The use of classical ridge regression in PEM is suggested by Ljung (1999) to solve the ill-conditioning which may arise when the number d_θ of parameters to be estimated is particularly high. In such cases, the Hessian of the loss function $V_N''(\theta, \mathcal{D}^N)$ may be ill-conditioned; thus, solving the regularized problem

$$\hat{\theta}_N = \arg\min_{\theta \in D_\theta} V_N(\theta, \mathcal{D}^N) + \eta\, \theta^\top \theta \tag{3.5}$$

will add a term ηI_{d_θ} to the Hessian V_N'', thus making it better conditioned. In addition, when d_θ is large, an accurate reconstruction of the true parameters becomes difficult: in such situations, the estimation benefits of the use of regularization, since it allows to fixes a better bias/variance trade-off. Further details on this topic are also provided by

Sjöberg, McKelvey, and Ljung (1993).

Note that in some cases also the Bayesian approaches detailed in Section 2.4 can reduce to regularized LS. In particular, this happens when they are implemented through regularized LS, i.e. by estimating a FIR model as the one in equation (2.142) (or an ARX model, if also $H(q)$ is determined). Nevertheless, the author's choice is to treat them in Section 3.3, since such methods have been introduced in the system identification literature as non-parametric techniques. Consequently, some of the penalties which will be later illustrated are adaptations of classical penalties adopted for non-parametric estimation in the statistical learning literature (e.g. those arising from splines kernels, Wahba (1990)). Other regularizers detailed in Section 3.3 have been instead developed according to a regularized LS framework: however, they will be treated in Section 3.3, because their roots lie in the non-parametric framework introduced by the seminal papers Pillonetto and De Nicolao (2010) and Pillonetto et al. (2011a).

3.1.2 ℓ_1 Regularization

Ljung, Hjalmarsson, and Ohlsson (2011) list four encounters between system identification and other research fields. One of them regards the exploitation of ℓ_1 regularization in connection with PEM to perform the aforementioned structure detection, or to estimate so-called segmented models (Ohlsson, Ljung, and Boyd, 2010).

Rojas and Hjalmarsson (2011) apply LASSO to LS PEM: the proposed algorithm first computes an LS estimate $\hat{\theta}_N^{LS}$, which is then made sparse by solving the following constrained ℓ_1 minimization problem

$$\begin{aligned} \hat{\theta}_N^{SP} := \arg\min_{\theta \in D_\theta} \; &\|\theta\|_1 \\ \text{s.t.} \;\; V_N(\theta) &\leq V_N(\hat{\theta}_N^{LS}) \left(1 + \frac{2n}{N}\right) \end{aligned} \tag{3.6}$$

where V_N is the LS loss function. A new LS estimation is then performed by removing the regressors corresponding to null entries in vector $\hat{\theta}_N^{SP}$. Conditions for consistency and and for sparsity are derived.

Tóth, Sanandaji, Poolla, and Vincent (2011) combine ideas from the compressive sensing literature (Baraniuk, 2007) with PEM in system identification. They consider the estimation of ARX models through a LASSO penalty and they show that the proposed method returns a consistent estimation of sparse models in terms of the so-called oracle property.

Another example of ℓ_1 type regularization regards nuclear norm penalties. To the best of the author's knowledge, the combination of PEM with nuclear norm regularization has been first proposed to deal with situations of missing output data. In such cases, classical approaches first reconstruct the missing measurements through the interpolation of the available data and then they estimate a model by minimizing the cost function (2.30). The alternative paradigm proposed by Grossmann, Jones, and Morari (2009) considers fitting the data only at the available measurements and adopts nuclear norm minimization as an interpolating method for the missing data. The authors consider the FIR model class (2.21) and estimate θ by solving

$$\hat{\theta}_N = \arg\min_{\theta \in D_\theta} \sum_{t \in T_o} \|y(t) - \hat{y}(t|\theta)\|_2^2 + \eta \|\mathbf{\Theta}\|_*, \qquad \mathbf{\Theta} \in \mathbb{R}^{p\frac{n_b}{2} \times p\frac{n_b}{2}} \tag{3.7}$$

where T_o denotes the set of time instants at which the output data are available, while $\mathbf{\Theta}$ is the (square) block Hankel matrix built with $\theta = \{B_1, ..., B_{n_b}\}$.
Through the resolution of problem (3.7), interpolation of missing output data is done by fitting a model in the class of low-order dynamic systems. Differently from standard approaches, the one due to Grossmann et al. (2009) does not require any assumption on the way in which the available measurements should be interpolated.
Grossmann et al. (2009) exploit the reformulation of nuclear norm minimization as an SDP (Fazel et al. (2001), equation (4)) to solve problem (3.7).

Hjalmarsson, Welsh, and Rojas (2012) takle the estimation of high-order ARX models by including a noise model in the convex optimization framework considered in the work of Grossmann et al. (2009). A high-order ARX model (2.25) is estimated by solving

$$\hat{\theta}_N = \arg\min_{\theta \in D_\theta} \sum_{t=1}^{N} \|y(t) - \hat{y}(t|\theta)\|_2^2 + \eta_A \|\mathbf{\Theta}_A\|_* + \eta_B \|\mathbf{\Theta}_B\|_* \tag{3.8}$$

where $\mathbf{\Theta}_A \mathbb{R}^{p\frac{n_a}{2} \times p\frac{n_a}{2}}$ and $\mathbf{\Theta}_B \in \mathbb{R}^{p\frac{n_b}{2} \times p\frac{n_b}{2}}$ respectively denote the block Hankel matrices built with the coefficients of the polynomials $A(q,\theta)$ and $B(q,\theta)$. This regularization serves as a penalty on complexity and pushes the estimated long ARX model to be close to a low-order model, thus reducing the variance error.
Hjalmarsson et al. (2012) also exploit the SDP reformulation of the nuclear norm minimization problem to solve (3.8); the regularization parameters η_A and η_B are determined through cross-validation. A reweighted algorithm to solve problem (3.8) has been proposed by Ha, Welsh, Blomberg, Rojas, and Wahlberg (2015).

An alternative type of regularization which has been recently introduced for PEM is based on the so-called *atomic norm* (Shah, Narayan Bhaskar, Tang, and Recht, 2012;

Bekiroglu, Yilmaz, Lagoa, and Sznaier, 2014). The authors propose the penalty

$$\|G(z)\|_{\mathcal{A}} = \inf \left\{ \sum_{w \in \mathbb{B}} |c_w| \ : \ G(z) = \sum_{w \in \mathbb{B}} \frac{c_w (1-|w|)^2}{z-w} \right\} \tag{3.9}$$

where $\mathbb{B}$ denotes the open unit ball in the complex plane $\mathbb{C}$, while z takes values on the unit circle of $\mathbb{C}$. In Shah et al. (2012) it is shown that the atomic norm is equivalent to the nuclear norm of the Hankel operator associated with $G(z)$. Hence, the penalty defined through (3.9) will prefer models having low-rank Hankel operators, and in turn low McMillan degrees. A convex optimization problem is formulated to approximatively solve the atomic norm minimization (Shah et al., 2012).

The stability of the derived estimators has been recently analyzed by Pillonetto et al. (2016).

3.2 Regularization in Subspace Methods

3.2.1 ℓ_2 Regularization

ℓ_2 regularization has not found large application in connection with subspace identification. Van Gestel, Suykens, Van Dooren, and De Moor (2001) propose the addition of an ℓ_2 penalty in the LS objective (2.100) adopted to estimate the system matrices. Namely,

$$\begin{bmatrix} \widehat{A} & \widehat{B} \end{bmatrix} = \underset{A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}}{\arg\min} \sum_{t=1}^{N} \left\| \hat{\hat{x}}(t+1) - \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \hat{\hat{x}}(t) \\ u(t) \end{bmatrix} \right\|_F^2 + \|AW^{1/2}\|_F^2 \tag{3.10}$$

Such regularized estimation should enforce the stability of matrix $\widehat{A}$. Indeed, it is well-known that for a finite number of data, the estimated system matrix $\widehat{A}$ is not guaranteed to be stable, even when the true linear system is known to be stable. The value of the regularization parameters W is determined through a generalized eigenvalue problem. Similarly, Lacy and Bernstein (2003) reformulate the LS problem (2.100) as a constrained convex linear programming problem.

3.2.2 ℓ_1 Regularization

The introduction of nuclear norm regularization in the context of subspace algorithms is quite recent and involves the SVD step performed in equations (2.78) (or (2.89)). Indeed, the truncation of the SVD entails an "hard" decision on the order of the system, which may often be difficult in the presence of noise and short data records. In fact this step has always been regarded as a critical one in subspace methods, see e.g. Bauer (2001).

As an alternative, Liu and Vandenberghe (2009) propose to adopt nuclear norm regularization as a complementary method for computing the above-mentioned low-rank approximations. In particular it is suggested that "surrogate" output data $\hat{y}^N$ are estimated solving

$$\hat{y}^N = \arg\min_{\tilde{y}^N} \sum_{t=1}^{N} \|y(t) - \tilde{y}(t)\|_2^2 + \eta\ \|\widetilde{\mathbf{Y}}\Pi^{\perp}_{\mathbf{U}^\top}\|_* \tag{3.11}$$

where $\widetilde{\mathbf{Y}}$ denotes the block Hankel matrix (2.61) built with unknown data, while $\Pi^{\perp}_{\mathbf{U}^\top}$ is the orthogonal projector onto the null-space of $\mathbf{U}$, defined in (2.75).
Once the optimal solution $\hat{y}^N$ of (3.11) is determined, the SVD of $\widehat{\mathbf{Y}}\Pi^{\perp}_{\mathbf{U}^\top}$ can be computed, where $\widehat{\mathbf{Y}}$ is here built with the data estimated through (3.11). Thanks to the nuclear norm minimization step (3.11), a clear gap between the relevant and the non-relevant singular values of $\widehat{\mathbf{Y}}\Pi^{\perp}_{\mathbf{U}^\top}$ should be detected, thus making the order selection a straightforward choice.

Liu and Vandenberghe (2009) adopt an interior-point method to solve problem (3.11), while Mohan and Fazel (2010) propose a variation inspired by so-called "iterative-reweighted" schemes, which has been named "reweighted nuclear norm heuristic" (RNH). RNH is based on the so-called reweighted trace heuristic (RTH, equation (5) in Mohan and Fazel (2010)), which derives from the reformulation of the rank minimization problem as a positive semidefinite one (equation (4) in Fazel et al. (2001)). At each iteration RNH solves the problem

$$\left(\hat{y}^N\right)^{(k+1)} = \arg\min_{\tilde{y}^N} \sum_{t=1}^{N} \|y(t) - \tilde{y}(t)\|_2^2 + \eta\|W_l^{(k)}\widetilde{\mathbf{Y}}\Pi^{\perp}_{\mathbf{U}^\top}W_r^{(k)}\|_* \tag{3.12}$$

and updates the weights $W_l^{(k)}$ and $W_r^{(k)}$ according to the current $\left(\hat{y}^N\right)^{(k+1)}$. Mohan and Fazel (2010) prove through numerical experiments that RNH makes model order selection easier and returns lower model orders w.r.t. standard nuclear norm minimization (3.11). The nuclear norm minimization of matrices with linear structure (e.g. Hankel, Toeplitz) is further discussed in Fazel, Kei, Sun, and Tseng (2013), where various first-order optimization methods are compared: these include alternating direction method of multipliers (ADMM), proximal-point algorithms and gradient projection methods.

The original idea introduced by Liu and Vandenberghe (2009) to combine nuclear norm regularization and subspace methods has been further developed by many authors during the last years. Hansson, Liu, and Vandenberghe (2012) reformulate the optimization problem (3.11) as

$$\hat{y}^N = \arg\min_{\tilde{y}^N} \quad \sum_{t=1}^{N} \|y(t) - \tilde{y}(t)\|^2 + \eta \, \|G_{IV}(\tilde{y}^N)\|_* \tag{3.13}$$

$$G_{IV}(\tilde{y}^N) = \widetilde{W}_1 \widetilde{\mathbf{Y}} \Pi^{\perp}_{\widetilde{\mathbf{U}}^\top} \Phi(u^N, \tilde{y}^N)^\top \widetilde{W}_2 \tag{3.14}$$

where $\Phi(u^N, \tilde{y}^N)$ denotes the instrumental variables matrix define in (2.66) (here the dependence on the input and output data has been made explicit), while $\widetilde{W}_1$ and $\widetilde{W}_2$ are the weighting matrices appearing in the SVD (2.89). The authors formulate an ADMM algorithm in order to solve problem (3.13). Experiments performed in Hansson et al. (2012) show that replacing the optimization problem (3.11) with (3.13) improves the accuracy of the estimated model and also reduces the dimension of the optimization problem, thus speeding up the problem resolution.

Liu, Hansson, and Vandenberghe (2013) adapt the subspace method combined with a nuclear norm optimization step to identification problems with partially missing input and output data. In this case problem (3.13) needs to be reformulated in order to account for the non-linear dependence of the matrix $G_{IV}(\tilde{y}^N)$ in (3.14) w.r.t. to the inputs:

$$(\hat{y}^N, \hat{u}^N) = \min_{\tilde{y}^N, \tilde{u}^N} \; \|\Phi(\tilde{y}^N, \tilde{u}^N)\|_* + \eta_1 \sum_{t \in T_o} \|y(t) - \tilde{y}(t)\|_2^2 + \eta_2 \sum_{t \in T_i} \|u(t) - \tilde{u}(t)\|_2^2 \tag{3.15}$$

In (3.15) T_o and T_i contain the time instants at which output and input measurements are available, while $\Phi(\tilde{y}^N, \tilde{u}^N)$ is the instrumental variables matrix built with unknown input and output data. The optimization variables in (3.15) are defined as: $\tilde{y}^N = \{\tilde{y}(-s), ..., \tilde{y}(N)\}$, $\tilde{u}^N = \{\tilde{u}(-s), ..., \tilde{u}(N)\}$. The choice of minimizing the nuclear norm of matrix $\Phi(\tilde{y}^N, \tilde{u}^N)$ is motivated by the fact that

$$\text{rank} \begin{bmatrix} \widetilde{\mathbf{U}} \\ \widetilde{\mathbf{Y}} \end{bmatrix} = n + \text{rank } \widetilde{\mathbf{U}} \tag{3.16}$$

provided $\tilde{u}^N$ and $\tilde{y}^N$ are, respectively, the input and output of a (noise free) linear system and the input is persistently exciting. Thus, the rank of $\Phi(\tilde{y}^N, \tilde{u}^N)$ equals the true system order plus a constant term.

After solving the optimization (3.15), the range of the extendend observability matrix can be estimated from $\Phi(\hat{y}^N, \hat{u}^N)$, e.g. through an LQ factorization. Again, the authors adopt a version of the ADMM algorithm to solve problem (3.15).

A further variation to the criterion (3.13) has been proposed by Sadigh, Ohlsson, Sastry, and Seshia (2013) in order to detect output outliers in the training data. The authors assume that the measured output data $y^N = \{y(1), ..., y(N)\}$ have a sparse number of outliers; no further assumptions on the specific time at which the outliers occur are done. By introducing an error term $\tilde{e}(t) \in \mathbb{R}^p$ which should represent the outlier appearing at time t, the optimization (3.13) is modified as follows:

$$(\hat{y}^N, \hat{e}^N) = \arg \min_{\tilde{y}^N, \tilde{e}^N} \sum_{t=1}^{N} \|\tilde{y}(t) - y(t)\tilde{e}(t)\|_2^2 + \eta_1 \|G_{IV}(\tilde{y}^N)\|_* + \eta_2 \sum_{t=1}^{N} \|\tilde{e}(t)\|_1 \tag{3.17}$$

The idea is to estimate both $\tilde{y}^N$ and the error term $\tilde{e}^N$ such that the error vector is sparse and accounts for the outliers that occur in the measured data.

A recent contribution (Verhaegen and Hansson, 2014) proposes a modification of the standard subspace algorithm in order to take into account the highly structured nature of equation (2.60). The structural properties on which the authors focus are the low-rank nature of the product $O_r X$, the block-Toeplitz structure of S_r and the block-Hankel structure of $\mathbf{V}$. The authors observe that these properties are not exploited in the first step of standard subspace methods, which typically use instrumental variables or projections to transform the original data. To avoid the loss of information which could arise because of these pre-processing steps, Verhaegen and Hansson (2014) suggest to consider the above-mentioned structural properties in order to constrain the estimation. Therefore, the first step of the N2SID algorithm they introduce consists in solving the following problem

$$(\widehat{\mathbf{Y}}, \widehat{S}_r) = \arg\min_{\widetilde{\mathbf{Y}} \in \mathbb{H}_p,\ \tilde{S}_r \in \mathbb{T}_{p,m}} \sum_{t=1}^{N} \|y(t) - \tilde{y}(t)\|_2^2 + \eta \|\widetilde{\mathbf{Y}} - \widetilde{S}_r \mathbf{U}\|_* \tag{3.18}$$

where $\mathbb{T}_{p,m}$ denotes the set of lower-triangular block-Toeplitz matrices having $p \times m$ matrices as block entries, while $\mathbb{H}_p$ is the set of block-Hankel matrices with block entries of p column vectors. The idea is to recover the low-rank approximation of the extended observability matrix by imposing the desired structural constraints.

Once the optimization (3.18) has been solved, the system order is then estimated through the SVD of the matrix $\widehat{\mathbf{Y}} - \widehat{S}_r \mathbf{U}$. The authors also design an appropriate ADMM algorithm to solve problem (3.18).

Smith (2014) extends the nuclear norm minimization to the frequency domain subspace identification. A further extension of the concepts illustrated in this section has been provided by Sznaier and Camps (2011), where rank minimization is exploited in order to establish whether two vector time sequences could have been generated by the same unknown LTI system. The proposed approach finds applications in computer vision and image processing problems.

3.3 Regularization in Non-Parametric Bayesian Methods

When a Gaussian prior distribution is adopted in presence of Gaussian noise, Bayesian regression coincides with ℓ_2 regularization (as widely discussed in Section 2.4). Therefore, the following distinction between ℓ_2 and ℓ_1 regularization may be misleading. What actually distinguishes the penalties illustrated in Sections 3.3.1 and 3.3.2 is the sparsity inducing property which characterizes the latter: specifically, Section 3.3.2 illustrates ℓ_1-type penalties, which are designed in order to induce sparsity in the returned estimator.

3.3.1 ℓ_2 Regularization

As observed in Section 2.4.1.2, regularization is a necessary tool for function regression, i.e. for the non-parametric system identification that is here treated. The explanation in Section 2.4 has highlighted how the desired properties of the impulse response to be estimated should be somehow encoded in the reproducing kernel associated to the RKHS within which $g(\cdot)$ is searched for. Recalling that such kernel admits an equivalent interpretation in terms of a covariance function in the Bayesian setting, such properties could be equivalently encoded into the design of a suitable Gaussian prior distribution for the stochastic process $\{g(k)\}$. To be in line with previous sections, the following discussion adopts the regularization point of view, even if some comments arising from the probabilistic perspective will be given.

For ease of notation, this section considers the SISO case ($p = m = 1$), while the MIMO case will be treated in Section 3.3.2. In addition, the discrete-time domain treated so far will be temporarily abandoned in order to faithfully follow the derivation of so-called stable-spline kernels provided by Pillonetto and De Nicolao (2010). Indeed, their presentation is based on the classical setting considered by the machine learning community, where a continuous function has to be estimated using the available observations.

The key idea of the seminal paper Pillonetto and De Nicolao (2010) is the adaptation of the spline kernels typically adopted in the statistical learning literature (Wahba, 1990) to the purposes of system identification. The use of such kernels ensures that the computed estimate is sufficiently smooth, according to the degree of smoothness encoded in the kernel. Specifically, considering the continuous domain $\mathcal{X} = [0, 1]$, a spline kernel of order p is defined as

$$K^{(p)}(s,t) = \int_0^1 G_p(s,u)G_p(t,u)du, \qquad G_p(r,u) = \frac{(r-u)_+^{p-1}}{(p-1)!} \tag{3.19}$$

where $(r-u)_+ = \max\{r-u, 0\}$. When this kernel is used to define the space $\mathcal{H}$ in

problem (2.135), the estimated function $\hat{g}$ is a so-called *smoothing spline*, because its derivatives upto order $2p-2$ are continuous.
However, when the impulse response of a BIBO stable LTI system has to be reconstructed, smoothness does not represent the unique desired property. It is well known that a sufficient and necessary condition for BIBO stability is that the system impulse response be absolutely integrable. As a consequence, it turns out that for system identification applications the chosen kernel should induce a space of functions $\mathcal{H}$ contained in the space of absolutely integrable functions. Considering the input space $\mathcal{X} = \mathbb{R}_+$, it can been proved that a necessary and sufficient condition for this to happen is that the kernel itself is absolutely summable, that is

$$\int_{\mathbb{R}_+} \int_{\mathbb{R}_+} K_+^{(p)}(s,t)\ ds\ dt < \infty \tag{3.20}$$

where $K_+(\cdot,\cdot)$ denotes the positive part of the kernel function. It turns out that the spline kernel (as well as the Gaussian kernel (Rasmussen and Williams, 2006)) is not stable when they are defined over $\mathcal{X} = \mathbb{R}_+$. Stability for this kernel could be easily ensured by truncating them, i.e. by setting $K^{(p)}(s,t) = 0,\ s,t > T$. However, this trick does not make the variability of the functions belonging to the space associated with $K^{(p)}$ exponentially decreasing, which is instead a distinctive feature of stable kernels.
Adopting a Bayesian perspective and interpreting the impulse response as a realization of a stochastic process with covariance (3.19), the previous considerations imply that the variance of g is not asymptotically decreasing (actually, it is asymptotically increasing, as observed in Figure 3.1 (left plot)).
To render the spline kernel stable, Pillonetto and De Nicolao (2010) introduces an exponential change of coordinates to map $\mathbb{R}_+$ into $[0,1]$ and then to apply the spline kernel there. Specifically, the proposed change of variables is

$$K_\tau^{(p)}(s,t) = K^{(p)}(e^{-\tau s}, e^{-\tau t}), \qquad (s,t) \in \mathbb{R}_+ \times \mathbb{R}_+, \qquad \tau \in \mathbb{R}_+ \tag{3.21}$$

with τ, playing the role of a hyper-parameter. By means of transformation (3.21), the so-called *first-order stable-spline* kernel is derived:

$$K_\tau^{(1)}(s,t) = e^{-\tau \max\{s,t\}} \tag{3.22}$$

Analogously, the *second-order stable-spline* kernel is defined as

$$K_\tau^{(2)}(s,t) = \frac{e^{-\tau(s+t+\max\{s,t\})}}{2} - \frac{e^{-3\tau \max\{s,t\}}}{6} \tag{3.23}$$

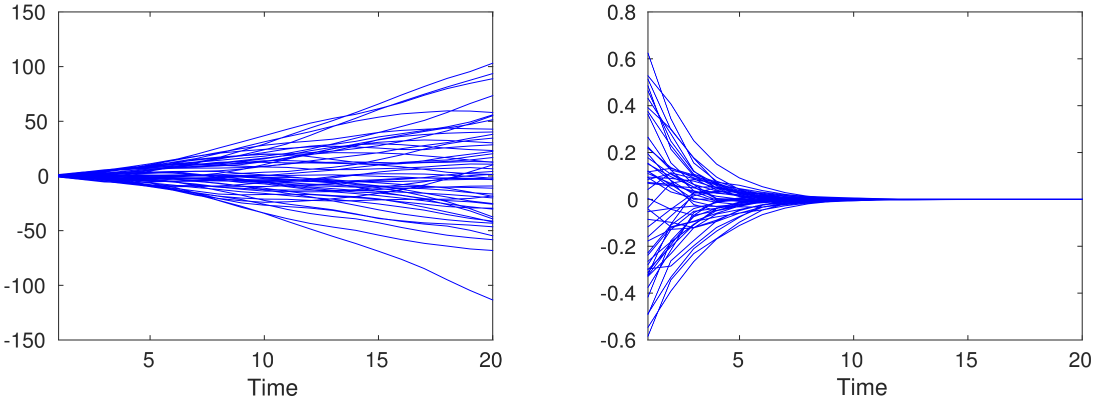


Figure 3.1: *Left:* Realizations of a zero-mean Gaussian process with covariance (3.19) and $p = 2$. *Right:* Realizations of a zero-mean Gaussian process with covariance (3.23).

Figure 3.1 (right plot) shows realizations of a zero-mean Gaussian process with covariance (3.23): clearly, its variance is exponentially decreasing. Recently, Chen, Ardeshiri, Carli, Chiuso, Ljung, and Pillonetto (2016) have provided a Maximum-Entropy interpretation of kernel (3.22), arising from the observation that stable-spline kernels are actually covariance functions of time-varying backward AR processes. For instance, kernel (3.22) can be obtained as the covariance of an AR process of order 1. Chen et al. (2016) show that, for any $\bar{t} \in \mathbb{N}$, the first-order stable-spline kernel is the solution of a Maximum Entropy problem (Cover and Thomas, 1991):

$$\max_{g(t)} \quad \overline{H}\left(g(t_0), g(t_1), \cdots, g(t_{\bar{t}})\right) \tag{3.24}$$

$$\text{s.t.} \quad \mathrm{Var}\left(g(t_{i+1}) - g(t_i)\right) = c(e^{-\tau t_i} - e^{-\tau t_{i+1}}) \tag{3.25}$$

$$\mathbb{E}[g(t_i)] = 0, \qquad i = 0, ..., \bar{t} - 1$$

where $\overline{H}(\cdot)$ denotes the entropy function.

The discrete domain can be straightforwardly recovered by setting $\beta = e^{-\tau}$:

$$K_\beta^{(p)}(s,t) = K^{(p)}(\beta^s, \beta^t), \qquad (s,t) \in \mathbb{N} \times \mathbb{N}, \qquad \beta \in \;[0,1] \tag{3.26}$$

Accordingly, the discrete version of the first-order stable-spline kernel is given by

$$K_\beta^{TC}(s,t) = \beta^{\max\{s,t\}}, \qquad \beta \in [0,1] \tag{3.27}$$

In the system identification literature (3.27) is known as the TC (tuned/correlated) kernel: this is the name with which it was originally proposed by Chen et al. (2012) in a regularized LS setting. That work also introduced the so-called DC (diagonal/correlated) kernel, defined as

$$K_{\eta}^{DC}(s,t) = \beta^{(s+t)/2}\rho^{|s-t|}, \qquad (s,t) \in \mathbb{N} \times \mathbb{N},\ \beta \in [0,1),\ \rho \in [-1,1] \tag{3.28}$$

with $\eta = [\beta, \rho]$. The Maximum Entropy interpretation of this kernel has been investigated by Carli, Chen, and Ljung (2014).

More details on the derivation of the stable-spline kernels are provided by Pillonetto et al. (2014) and Dinuzzo (2015).

Extensions of these basic kernels have been considered in the recent system identification literature. Chiuso et al. (2014) suggest a superposition of stable-spline kernels, which allows to combine structural properties (such as exponentials of exponentially modulated sinusoids) with a random process built from the Brownian bridge. Such kind of construction has proved to be particularly efficient when dealing with resonant systems.

Remark 3.3.1. Typically, the described kernels are all equipped with a scaling factor $\lambda \in \mathbb{R}_+$, which is treated as an hyper-parameter. Consequently, the adopted kernels are

$$K_{\eta}(s,t) = \lambda K_{\beta}^{(p)}(s,t), \qquad \lambda \in \mathbb{R}_+, \eta = [\lambda,\ \beta] \tag{3.29}$$

with $K_{\beta}^{(p)}(s,t)$ defined e.g. as in equation (3.26).

An alternative approach for kernel design is taken by Darwish, Tóth, and van den Hof (2014) and Chen and Ljung (2014), who construct RKHS of impulse responses spanned by orthonormal basis functions on the unit circle (e.g. Laguerre basis (Wahlberg, 1991)). The associated reproducing kernel is a combination of such bases, whose poles are treated as hyper-parameters and hence estimated from the given data. To satisfy the stability constraint imposed by BIBO stable systems, a decaying prior on the basis coefficients is postulated. The results reported in Chen and Ljung (2014) suggest that this kernel design may provide some advantages w.r.t. the classical TC kernels above-mentioned.

3.3.2 ℓ_1 Regularization

The types of penalties described in Section 3.3.1 only account for certain properties of the impulse response to be estimated (such as smoothness and stability). However, the corresponding kernels are not able to reproduce certain structural features of the system to be estimated, which are of extreme importance when MIMO systems have to be identified. Indeed, the focus of this section will go back to the reconstruction of MIMO systems, where not only the impulse responses connecting each input-output channel have to be estimated but also the interplay between different input and output channels should be taken into account. A primary goal in this setting is *structure detection* (as already observed in the introduction to this chapter), i.e. the capability of detecting which inputs influence a certain output. This means that, if the impulse responses are collected in a vector $\mathbf{g} \in \mathbb{R}^{pmT}$, with T denoting the impulse response length

$$\begin{aligned} \mathbf{g} &= [\mathbf{g}_{11}^\top \ \mathbf{g}_{12}^\top \ \cdots \ \mathbf{g}_{1m}^\top \ \cdots \ \mathbf{g}_{p1}^\top \ \cdots \ \mathbf{g}_{pm}^\top]^\top \\ \mathbf{g}_{ij} &= [g_{ij}(1) \ g_{ij}(2) \ \cdots \ g_{ij}(T)]^\top, \qquad i = 1,..,p, \ j = 1,..,m \end{aligned} \tag{3.30}$$

the identification method should return an estimate $\hat{\mathbf{g}}$ with possibly null block entries (i.e. $\hat{\mathbf{g}}_{ij} = 0_T$ for some i, j).

Another structural property which involves also SISO systems regards the possible time-varying nature of the system to be identified: for instance, if the system poles undergo certain abrupt changes, the identification procedure should be able to detect them. In this case, collecting in $\theta_t \in \mathbb{R}^{pmT}$ the impulse response coefficients at time t, a suitable identification algorithm should estimate $\{\theta_{T+1}, ..., \theta_N\}$ and return $\theta_t = \theta_{t+1}$ when no modifications happen.

These examples highlight that such problems could be tackled by resorting to sparsity inducing regularization techniques. The approach which has been mainly pursued in the system identification literature is the so-called *Sparse Bayesian Learning (SBL),* introduced by Tipping (2001). Such learning algorithm also shares several features with the *Automatic Relevance Determination (ARD)* (MacKay and Neal, 1994; Wipf and Nagarajan, 2008). The way in which SBL algorithms work can be easily understood through the following trivial example, derived by Aravkin et al. (2012) and reported in Chiuso (2016).

Example 3.3.2. Consider the data y^N, generated according to

$$y(t) = \theta + e(t), \qquad \theta \in \mathbb{R}, \quad e(t) \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma) \tag{3.31}$$

Assuming a zero-mean Gaussian prior for θ, $\theta \sim \mathcal{N}(0, \eta)$, the estimator $\hat{\eta}_{EB}$ computed by marginal likelihood maximization (as in equation (2.183)) is given by

$$\hat{\eta}_{EB} = \max \left\{ 0, \left(\frac{1}{N} \sum_{t=1}^{N} y^2(t) \right) - \sigma \right\} \tag{3.32}$$

Hence, if the sample variance of the measured data is below the noise variance, the hyper-parameter estimate is zero and in turn $\hat{\theta} = \mathbb{E}[\theta | Y_N, \hat{\eta}_{EB}] = 0$. ■

Following the SBL approach, Chiuso and Pillonetto (2010, 2012) propose to treat the impulse responses $\mathbf{g}_{ij}$, $i = 1, ..., p$, $j = 1, ..., m$, as independent random vectors, thus postulating a Gaussian prior with a block diagonal covariance matrix:

$$\mathbf{g} \sim \mathcal{N}(0_{pmT}, \bar{K}_\eta), \qquad \bar{K}_\eta = \text{blockdiag}(\bar{K}_\eta, ..., \bar{K}_\eta) \tag{3.33}$$

with

$$\bar{K}_{\eta_{ij}} = \lambda_{ij}\bar{K}_\beta, \qquad \bar{K}_\beta \in \mathbb{R}^T \tag{3.34}$$

In particular, $\bar{K}_\beta$ is set equal to one of the kernels illustrated in Section 3.3.1. According to Example 3.3.2, estimating λ_{ij}, $i = 1, ..., p$, $j = 1, ..., m$, through marginal likelihood maximization will enforce block-sparsity in the vector $\hat{\mathbf{g}}$. It should be noticed that this algorithm coincides with solving the following ℓ_2-type regularization problem

$$\hat{\mathbf{g}} = \underset{\mathbf{g}\in\mathbb{R}^{pmT}}{\arg\min} \sum_{t=1}^{N} \|y(t) - \varphi^\top(t)\mathbf{g}\|^2_{\Sigma^{-1}} + \sum_{i=1}^{p}\sum_{j=1}^{m} \mathbf{g}_{ij}^\top \frac{\bar{K}_\beta^{-1}}{\hat{\lambda}_{EB,ij}} \mathbf{g}_{ij} \tag{3.35}$$

where $\varphi(t)$ is the matrix containing past input data (defined in equation (2.181)), Σ denotes the noise variance and the regularization parameters λ_{ij} have been fixed through evidence maximization. A more general version would estimate a separate kernel $\bar{K}_{\beta_{ij}}$ for each input-output channel.

A similar approach is taken by Chen et al. (2014) for the segmentation of SISO systems. Specifically, considering the aforementioned setting, the problem is tackled by solving

$$\begin{aligned}(\hat{\theta}_{T+1}, ..., \hat{\theta}_N) &= \underset{\theta_{T+1},...,\theta_N}{\arg\min} \sum_{t=1}^{N} \|y(t) - \varphi^\top(t)\theta_t\|^2_{\Sigma^{-1}} + (\theta_t - \theta_{t-1})^\top \bar{K}(\alpha_t)^{-1}(\theta_t - \theta_{t-1}) \\ \bar{K}(\alpha_t) : &= \sum_{i=1}^{r} \lambda_{i,t}\bar{K}, \qquad \alpha_t := [\lambda_{1,t}, \cdots, \lambda_{r,t}]^\top \end{aligned} \tag{3.36}$$

where again $\lambda_{i,t}$, $i = 1, ..., r$, $t = 1, ..., N$ are fixed through marginal likelihood maximization. If $\alpha_t = 0$ for some $t = T+1, ..., N$, then $\theta_t = \theta_{t-1}$. The authors observe that problem (3.36) is actually a difference of convex programming (DCP) problems, meaning that a locally optimal solution can be efficiently found (e.g. by using a majorization-minimization algorithm or an interior-point technique).

SBL algortihms have been largely compared with LASSO (or Grouped-LASSO) estimators in the machine learning literature (see e.g. Wipf, Rao, and Nagarajan (2011)). In particular, Aravkin, Burke, Chiuso, and Pillonetto (2014) prove the superiority of SBL in terms of achieving a better trade-off between shrinkage and sparsity. Indeed, besides recovering the sparsity pattern of the unknown variable, LASSO estimators also tend to shrink the estimated non-zero coefficients, thus possibly compromising the recovery of the true unknown quantity. On the other hand, the estimates returned by SBL have proved to be more effective in the reconstruction of the sparsity pattern and in the correct estimation of the non-zero coefficients.

Modelling the impulse responses of a MIMO system as independent Gaussian processes (as in the above-detailed approach) has a major drawback: the coupling between the impulse responses g_{ij} connecting different input-output channels is not captured; this is especially true when the system has a low McMillan degree. To encode such property in a suitable identification criterion, penalties inducing low McMillan degree should be adopted: as observed in the introduction of this chapter, nuclear norm penalties on the system Hankel matrix are appropriate candidates for this task. Despite the broad application that nuclear norm regularization has recently found in the system identification literature (as detailed in the previous sections), direct use of nuclear norm (or atomic) penalties may lead to undesired behaviour, as suggested and studied in Pillonetto et al. (2016), due to the fact that nuclear norm is not able alone to guarantee stability and smoothness of the estimated impulse responses. To address this limitation, Chiuso, Chen, Ljung, and Pillonetto (2013) suggest to estimate the system impulse response by combining the stability/smoothness penalty with the nuclear norm one:

$$\hat{\mathbf{g}} = \arg\min_{\mathbf{g}\in\mathbb{R}^{pmT}} \sum_{t=1}^{N} \|y(t) - \varphi^\top(t)\mathbf{g}\|^2_{\Sigma^{-1}} + \lambda_1 \mathbf{g}^\top \bar{K}_\beta^{-1}\mathbf{g} + \lambda_2\|\mathbf{G}\|_* \tag{3.37}$$

In equation (3.37) $\varphi(t)$ denotes the matrix (2.181) containing past input data, $\mathbf{G}$ is the Hankel matrix built with impulse response coefficients and $\bar{K}_\beta$ is one of the stable-spline kernels detailed in Section 3.3.1. In this case $\bar{K}_\beta$ is not constrained to be block-diagonal. However, it should be observed that formulation (3.37) does not admit a fully Bayesian interpretation, since no Gaussian prior gives rise to a regularization function $J_R(\mathbf{g},\eta) = \eta\|\mathbf{G}\|_*$: hence, evidence maximization can not be exploited for the estimation of $\eta = [\lambda_1, \lambda_2, \beta]$. Indeed, Chiuso et al. (2013) assume that the hyper-parameters defining the stable-spline kernel $\bar{K}_\beta$ have been already fixed through a previous identification procedure, while they estimate λ_1 and λ_2 in (3.37) through cross-validation.

Next section will extend this latter idea, by adopting a Bayesian perspective and developing a Gaussian prior accounting for both stability and complexity (measured in terms of McMillan degree) of the identified system.

3.4 Combining ℓ_2 and ℓ_1 regularization in Non-parametric Bayesian system identification: a Maximum-Entropy derivation

The section will develop, by means of Maximum Entropy arguments, a vector-valued kernel accounting both for the stability of the system to be estimated and for its complexity, as measured by its McMillan degree. The prior distribution here introduced leads, as a special case, to an Hankel nuclear norm penalty.
By exploiting the newly developed prior distribution, inspired by the growing literature on iterative reweighted algorithms, an iterative procedure is designed, which alternatively updates the impulse response estimate and the hyper-parameters defining the prior. Standard iterative reweighted algorithms solve regularized estimation problems by alternatively updating the estimate and the regularization parameters (referred as “weights” in this context). These methods have been first introduced in compressive sensing applications, in order to improve the recovery of sparse solutions in presence of few measurements Candes, Wakin, and Boyd (2008); Chartrand and Yin (2008); Daubechies et al. (2010). Mohan and Fazel (2012) and Fornasier, Rauhut, and Ward (2011) have extended these algorithms to the *Affine Rank Minimization Problem* (ARMP), while Wipf and Nagarajan (2010) developed a reweighting scheme for the *Sparse Bayesian Learning* (SBL) setting (Tipping, 2001), where the weights update corresponds to a hyper-parameter update. The algorithm here designed differs from these cited above in that the regularization matrix takes on a very special structure, described by a few hyper-parameters. This special structure acts indeed as an hyper-regularizer which helps avoiding overfitting, but has the drawback that no closed-form solution is available for the weights (i.e. hyper-parameters) update. Indeed, this step is performed through marginal likelihood maximization following the so-called Empirical Bayes approach.

As the title of this section may suggest, the prior which will be here derived leads to a regularization term which could resemble the well-known elastic-net regularization (Zou and Hastie, 2005). This technique combines ℓ_1- and ℓ_2-type regularization in order to enforce shrinkage and sparsity in the returned estimate. The approach here developed differs from the standard elastic net in the implementation of the sparsity inducing term, which is here a weighted ℓ_2-type regularization (with a weighting suitably designed to enforce sparsity). This property directly results from the derivation of the regularization term by means of Bayesian arguments and, in particular, by the use of a Gaussian prior distribution.

Section 3.4.1 details how the mentioned prior is derived, while Section 3.4.2 illustrates the iterative algorithm which computes the final impulse response estimate. Finally, Section 3.4.3 describes an adaptation of the SGP routine (Algorithm 1), which has been developed to solve the marginal likelihood maximization problem arising in this setup.

For simplicity, the impulse response to be estimated will be here treated as a finite-dimensional vector $\mathbf{g} \in \mathbb{R}^{pmT}$ (as defined in equation (2.178)).

3.4.1 Maximum-Entropy design of stable Hankel-type penalties

In Section 3.3.1, the Maximum Entropy derivation of the first-order stable spline kernel (3.22) has been mentioned (Chen et al., 2016). Here, its discrete version, the TC kernel (3.27) will be considered and denoted as $\bar{K}_{S,\nu}$, with ν being its hyper-parameters. It is easy to see that the Gaussian prior with covariance (3.27) can be derived as the solution of a Maximum Entropy problem with constraints

$$\mathbb{E}\left[\mathbf{g}^\top \bar{K}_{S,\nu}^{-1}\mathbf{g}\right] = \bar{k} \tag{3.38}$$

$$\mathbb{E}\left[g(k)\right] = 0, \quad k = 1, ..., \bar{k} \tag{3.39}$$

where the expectation is taken w.r.t. the probability distribution $p(\mathbf{g})$. Note, in fact, that the constraint set (3.38) contains (3.25).

When dealing with MIMO systems, a possible approach is to consider a block-diagonal kernel (as suggested e.g. by Chiuso and Pillonetto (2012), here outlined in equation (3.33)). However, this assumption is often unreasonable, since the possible coupling between the different input-output channels is not accounted for. Recalling the introductory discussion to the chapter, such coupling could be accounted for through a suitable penalty on the block-Hankel matrix $\mathbf{G}$, built with the impulse response coefficients.

In this setting, r and c will respectively denote the number of block rows and columns appearing in $\mathbf{G}$. Their values are chosen so that $r + c - 1 = T$ and the matrix $\mathbf{G}$ is as close as possible to a square matrix. Furthermore, for the purpose of normalization, a weighted version $\widetilde{\mathbf{G}}$ of $\mathbf{G}$ is considered:

$$\widetilde{\mathbf{G}} := W_1\mathbf{G}W_2 \tag{3.40}$$

Specifically, W_1 and W_2 are chosen so that the singular values of $\widetilde{\mathbf{G}}$ are conditional canonical correlation coefficients between future outputs and near past inputs, given the future inputs and remote past inputs. We refer to Chiuso et al. (2013) for more details on the derivation of W_1 and W_2. Notice that the notation adopted for these weighting matrices is analogous to that used in equation (2.78) for subspace algorithms. Such choice is done to highlight the connections between the two approaches.

Remark 3.4.1. For Gaussian processes, there is a one-to-one correspondence between the Canonical Correlation Analysis (CCA) and mutual information. Indeed, the mutual information between past (y^-) and future (y^+) of a Gaussian process $\{y(t)\}$ is given by:

$$I(y^+; y^-) = -\frac{1}{2}\sum_{k=1}^{n} \log(1-\rho_k^2) \tag{3.41}$$

where ρ_k is the k-th canonical correlation coefficient and n is the McMillan degree of a minimal spectral factor of y.
This provides a clear interpretation of canonical correlations as well as of the impact of shrinking them in terms of mutual information. A similar interpretation holds for systems with inputs, which relates conditional mutual information and conditional canonical correlations, i.e. the singular values of (3.40).

In the following, the design of a kernel allowing to reduce the complexity of the estimated system, i.e. the rank of the corresponding block Hankel matrix, is illustrated. This kernel will derive from the covariance matrix of a Maximum Entropy distribution built under suitable constraints.
The aim now is to formulate a probability distribution $p(\mathbf{g})$ for $\mathbf{g}$, such that samples drawn from $p(\mathbf{g})$ have low rank (or close to low rank) Hankel matrices. To this purpose, some of the singular values of $\mathbf{G}$ should be favoured to be zero: this can be achieved imposing constraints on the eigenvalues of the weighted matrix $\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top$. Denoting with $u_i(\mathbf{g})$ the i-th singular vector of $\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top$, the corresponding singular value (or, equivalently, the eigenvalue) is given by

$$s_i^2(\mathbf{g}) = u_i(\mathbf{g})^\top \widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top u_i(\mathbf{h}), \qquad i = 1, ..., pr \tag{3.42}$$

In the spirit of Sparse Bayesian Learning (SBL) ideas (Tipping, 2001), a constraint of the following type can be imposed:

$$\mathbb{E}\left[s_i^2(\mathbf{g})\right] = \mathbb{E}\left[u_i(\mathbf{g})^\top \widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top u_i(\mathbf{g})\right] \leq \omega_i, \qquad i = 1, ..., pr \tag{3.43}$$

where the expectation is taken w.r.t. $p(\mathbf{g})$. The ω_i's play the role of hyper-parameters that have to be estimated from the data[1].

[1] In fact, one shall not estimate directly the ω_i's, but rather the corresponding dual variables appearing in the Maximum Entropy distribution, i.e. the λ_i's in (3.51).

To the purpose of defining a distribution for $\mathbf{g}$ which encodes the desired prior knowledge, an estimate $\hat{\mathbf{g}}$ of $\mathbf{g}$ is assumed to be available. Section 3.4.2 will detail how this "preliminary" estimate of $\mathbf{g}$ arises as an intermediate step in an alternating minimization algorithm.

Thus, the (weighted) estimated Hankel matrix $\widehat{\widetilde{\mathbf{G}}}$ and its singular value decomposition are considered:

$$\widehat{U}\widehat{S}\widehat{U}^\top := \widehat{\widetilde{\mathbf{G}}}\widehat{\widetilde{\mathbf{G}}}^\top \tag{3.44}$$

The constraints (3.43) can now be formulated as

$$\mathbb{E}\left[\hat{u}_i^\top \widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \hat{u}_i\right] \leq \omega_i, \qquad i = 1, ..., pr \tag{3.45}$$

where $\hat{u}_i$ denotes the i-th column of $\widehat{U}$. In this way the vectors $\hat{u}_i$ are fixed, thus leaving all the modelled uncertainty modelled to the prior on the weighted Hankel matrix $\widetilde{\mathbf{G}}$. However, having fixed the $\hat{u}_i$'s, which in general are not the (exact) singular vectors of the "true" Hankel matrix, introduces a perturbation on the constraints (3.45), and in turn on the resulting prior distribution. One way to make the constrains (3.45) robust to such perturbations is to group the estimated singular vectors in the so-called "signal" and "noise" subspaces. To this purpose, the first n singular vectors are grouped, while $\widehat{U}$ and $\widehat{S}$ are partitioned as follows:

$$\widehat{U} = \left[\begin{array}{cc} \widehat{U}_n & \widehat{U}_n^\perp \end{array}\right], \qquad\qquad \widehat{S} = \text{blockdiag}(\widehat{S}_n, \widehat{S}_n^\perp) \tag{3.46}$$

where $\widehat{U}_n \in \mathbb{R}^{pr\times n}$. Note that, while the $\hat{u}_i$'s corresponding to small singular values are likely to be very noisy, both the "signal" space spanned by the columns of $\widehat{U}_n$, as well as that spanned by $\hat{u}_i$, $i = n+1, .., pr$, i.e. the column space of $\widehat{U}_n^\perp$ are much less prone to noise. This is easily derived from a perturbation analysis of the singular value decomposition which shows that the error in $\widehat{U}_n^\perp$ depends on the gap between the smallest singular value of $\widehat{S}_n$ and the largest of $\widehat{S}_n^\perp$. In view of these considerations, the constraints (3.45) can be relaxed by aggregating the "signal" components (i.e. the first n singular vectors):

$$\mathbb{E}\left[\sum_{i=1}^{n} \hat{u}_i^\top \widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \hat{u}_i\right] \leq \sum_{i=1}^{n} \omega_i \tag{3.47}$$

i.e.

$$\mathbb{E}\left[\text{Tr}\left[\widehat{U}_n^\top \widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \widehat{U}_n\right]\right] \leq \sum_{i=1}^{n} \omega_i \tag{3.48}$$

Similarly, the constraints on the "noise" component (i.e. the last $pr-n$ singular vectors) are grouped:

$$\mathbb{E}\left[\sum_{i=n+1}^{pr} \hat{u}_i^\top \widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \hat{u}_i\right] \leq \sum_{i=n+1}^{pr} \omega_i \tag{3.49}$$

that is,

$$\mathbb{E}\left[\mathrm{Tr}\left[\left(\widehat{U}_n^\perp\right)^\top \widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \widehat{U}_n^\perp\right]\right] \leq \sum_{i=n+1}^{pr} \omega_i \tag{3.50}$$

Notice that these constraints are relaxed w.r.t. the ones in (3.45), since here only the sum is involved.

Exploiting a well-known result (Cover and Thomas, 1991, p. 409), the Maximum Entropy distribution subject to constraints (3.48)-(3.50) can be built as:

$$\begin{aligned} p_\zeta(\mathbf{h}) &\propto \exp\left(-\lambda_1 \mathrm{Tr}\left\{\widehat{U}_n^\top \widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \widehat{U}_n\right\}\right) \cdot \exp\left(-\lambda_2 \mathrm{Tr}\left\{\left(\widehat{U}_n^\perp\right)^\top \widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \widehat{U}_n^\perp\right\}\right) \\ &\propto \exp\left(-\mathrm{Tr}\left\{\widehat{U}^\top \widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \widehat{U} \ \mathrm{blockdiag}(\lambda_1 I_n, \lambda_2 I_{pr-n})\right\}\right) \\ &\propto \exp\left(-\mathrm{Tr}\left\{\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \widehat{Q}(\zeta)\right\}\right) \end{aligned} \tag{3.51}$$

where $\zeta := [\lambda_1, \lambda_2, n]$, $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, and the last equation uses

$$\widehat{Q}(\zeta) := \widehat{U} \ \mathrm{blockdiag}(\lambda_1 I_n, \lambda_2 I_{pr-n}) \ \widehat{U}^\top = \lambda_1 \widehat{U}_n \widehat{U}_n^\top + \lambda_2 \widehat{U}_n^\perp \left(\widehat{U}_n^\perp\right)^\top \tag{3.52}$$

Remark 3.4.2. It should be stressed that the quality of the relaxation introduced in constraints (3.48)-(3.50) depends on the relative magnitude of the Hankel singular values. Using the “normalized” Hankel matrix (3.40) plays an important role here since its singular values, being canonical correlations, are all in the interval $(0, 1]$. On the other hand, the aggregation of the singular values along the “noise” subspace resembles the role played by the regularization factor in Iterative Reweighted methods Chartrand and Yin (2008); Wipf and Nagarajan (2010). The reader is referred to Section 3.4.2.3 for a further discussion on the connection between these methods and the approach here proposed.

Remark 3.4.3. Notice that $\widehat{Q}(\zeta)$ in (3.52) is actually the weighted sum of two orthogonal projections, respectively on the so-called “signal subspace” (that would coincide with the column space of $\widetilde{\mathbf{G}}$ if n was the true system order) and on the “noise subspace”. This observation provides new insights on the design of the prior in (3.51): namely, by properly tuning the hyper-parameters ζ, the prior is intended to be stronger along certain directions of the column space of $\widetilde{\mathbf{G}}$ (referred to as the “noisy” ones) and milder along what we call the “signal” directions.

Since $\widetilde{\mathbf{G}}$ is linear in $\mathbf{g}$, $\mathrm{Tr}\left[\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \widehat{Q}(\zeta)\right]$ can be rewritten as a quadratic form in $\mathbf{g}$. Indeed, letting $\widehat{Q}(\zeta) = LL^\top$,

$$\begin{aligned}\mathrm{Tr}\left[\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \widehat{Q}(\zeta)\right] &= \mathrm{Tr}\left[L^\top\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top L\right] &(3.53)\\ &= \|\mathrm{vec}(\widetilde{\mathbf{G}}^\top L)\|_2^2\\ &= \|(L^\top W_1 \otimes W_2^\top)\mathrm{vec}(\mathbf{G}^\top)\|_2^2\\ &= \|(L^\top W_1 \otimes W_2^\top)P\mathbf{g}\|_2^2\\ &= \mathbf{g}^\top P^\top (W_1^\top \widehat{Q}(\zeta) W_1 \otimes W_2 W_2^\top)P\mathbf{g} &(3.54)\end{aligned}$$

where $P \in \mathbb{R}^{rpcm\times Tmp}$ is the matrix which vectorizes $\mathbf{G}^\top$, i.e. $\mathrm{vec}\left(\mathbf{G}^\top\right) = P\mathbf{g}$. Equation (3.51) can then be rewritten as

$$p_\zeta(\mathbf{g}) \propto \exp\left(-\mathbf{g}^\top P^\top (W_1^\top \widehat{Q}(\zeta) W_1 \otimes W_2 W_2^\top)P\mathbf{g}\right) \tag{3.55}$$

so that $p_\zeta(\mathbf{g})$ is the probability density function of a zero-mean Gaussian vector, i.e.:

$$\begin{aligned}\mathbf{g} &\sim \mathcal{N}(0_{Tmp}, \bar{K}_{H,\zeta}) &(3.56)\\ \bar{K}_{H,\zeta} &= \left[P^\top (W_1^\top \widehat{Q}(\zeta) W_1 \otimes W_2 W_2^\top)P\right]^{-1} &(3.57)\\ \zeta &= [\lambda_1, \lambda_2, n] &(3.58)\end{aligned}$$

By adopting (3.56) as a prior distribution for $\mathbf{g}$, the problem of estimating $\mathbf{g}$ can be recast under the framework outlined in Section 2.4.2.3. In particular, ζ play the role of hyper-parameters; as $\lambda_2 \to \infty$, realizations $\mathbf{g}$ from (3.56) are (close to) low order systems with (weighted) Hankel matrices $\widetilde{\mathbf{G}}$ having the n-dimensional principal subspace close to the column space of $\hat{U}_n$. Conversely, as $\lambda_1 \to 0$, the n-dimensional principal subspace of $\widetilde{\mathbf{G}}$ is not penalized, thus leading to an improper prior, flat along some directions. This feature allows to reduce the bias of the estimator along the "signal" subspace. Thus complexity (in terms of McMillan degree) is controlled by properly choosing the hyperparametrs λ_1, λ_2, n, which can be done by marginal likelihood maximization as outlined in Section 3.4.2.

It is worth to observe that the quadratic nature of (3.55) w.r.t. $\mathbf{g}$ derives from the fact that the constraints (3.45) are quadratic in $\mathbf{g}$.

A prior distribution is now formulated, which enforces both stability (imposing constraint (3.38)) and low complexity (imposing (3.48) and (3.50)) of the estimated system. Using again (Cover and Thomas, 1991, p. 409), the Maximum Entropy distribution under

(3.38), (3.48) and (3.50) takes then the form:

$$\begin{aligned} p_\eta(\mathbf{g}) &\propto \exp\left(-\lambda_0 \mathbf{g}^\top \bar{K}_{S,\nu}^{-1}\mathbf{g} - \mathbf{g}^\top \bar{K}_{H,\zeta}^{-1}\mathbf{g}\right) \\ &\propto \exp\left(-\mathbf{g}^\top\left(\lambda_0 \bar{K}_{S,\nu}^{-1} + \bar{K}_{H,\zeta}^{-1}\right)\mathbf{g}\right) \end{aligned} \tag{3.59}$$

where $\eta = [\nu, \lambda_0, \zeta]$, $\lambda_0 \geq 0$, and $\bar{K}_{H,\zeta}$ is the kernel in (3.57). The use of a further hyper-parameter, λ_0, will become clear later on. From the distribution (3.59) the kernel

$$\begin{aligned} \bar{K}_{SH,\eta} &= \left(\lambda_0 \bar{K}_{S,\nu}^{-1} + \bar{K}_{H,\zeta}^{-1}\right)^{-1} \\ &= \left[\lambda_0 \bar{K}_{S,\nu}^{-1} + P^\top (W_1^\top \widehat{Q}(\zeta) W_1 \otimes W_2 W_2^\top) P\right]^{-1} \end{aligned} \tag{3.60}$$

is derived, with hyper-parameters

$$\eta = [\nu, \lambda_0, \zeta] \tag{3.61}$$

and ζ as defined in (3.58). This kernel leads to both stable and low-complexity estimates, as will be demonstrated in Section 3.5.

Remark 3.4.4. As thoroughly discussed in Pillonetto et al. (2016), the kernel arising from the “Hankel” constraint alone would not necessarily lead to stable models. In fact, given an unstable system and its finite Hankel matrix $\mathbf{G}$, it is always possible to design a stable system whose finite Hankel matrix (with the same size of $\mathbf{G}$) has the same singular values of $\mathbf{G}$. In addition, the Hankel prior does not include information on the correlation among the impulse response coefficients (see Pillonetto et al. (2016)).

3.4.1.1 Variational Derivation of the Hankel-type prior

Adopting a regularization point of view, i.e. casting the Bayesian estimation problem under the framework of Section 2.4.1.3, the penalty induced by the kernel (3.57) can be also derived through a variational bound (Prando, Chiuso, and Pillonetto, 2014; Wipf, 2012).

Indeed, in order to force sparsity on the vector $s(\mathbf{g})$ of the singular values of $\widetilde{\mathbf{G}}$, one should penalize its ℓ_0-norm, $\|s(\mathbf{g})\|_0$, which is equal to the number of non-zero components of $s(\mathbf{g})$. However, since this norm can not be expressed as a quadratic form of $\mathbf{g}$, it can not fit the ℓ_2-penalty appearing in (2.150). Observing that

$$\sum_i \log |s_i(\mathbf{g})| \equiv \lim_{p\to 0} \frac{1}{p} \sum_i (|s_i(\mathbf{g})|^p - 1) \propto \|s(\mathbf{g})\|_0 \tag{3.62}$$

the ℓ_0-norm of $s(\mathbf{g})$ can be approximated by its Gaussian entropy measure $\sum_i \log |s_i(\mathbf{g})|$. These considerations suggest to adopt the penalty $\sum_i \log |s_i^2(\mathbf{g})| = \log |\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top|$ [2], which can be upper bounded by a quadratic form of $\mathbf{g}$. To this purpose, one should first observe that the concave function $\log |\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top|$ can be expressed as the minimum over a set of upper-bounding lines (Wipf, 2012):

$$\log |\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top| = \min_{\Psi \succ 0} \mathrm{Tr} \left[\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \Psi^{-1}\right] + \log |\Psi| - rp \tag{3.63}$$

with $\Psi \in \mathbb{R}^{rp \times rp}$ being a positive definite matrix of so-called variational parameters. By adopting $\log |\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top|$ as regularization function and by using its expression in (3.63), one has

$$\hat{\mathbf{g}} = \arg \min_{\mathbf{g} \in \mathbb{R}^{pmT}} (Y_N - \Phi_N \mathbf{g})^\top \widetilde{\Sigma}_N^{-1} (Y_N - \Phi_N \mathbf{g}) + \mathrm{Tr}\left[\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \Psi^{-1}\right] \tag{3.64}$$

where Y_N and Φ_N have respectively defined in (2.123) and (2.180), while $\widetilde{\Sigma}_N$ is the noise covariance matrix in equation (2.171). Exploiting the expression of the trace term found in (3.54), problem (3.64) can be rewritten as

$$\hat{\mathbf{g}} = \arg \min_{\mathbf{g} \in \mathbb{R}^{pmT}} (Y_N - \Phi_N \mathbf{g})^\top \widetilde{\Sigma}_N^{-1} (Y_N - \Phi_N \mathbf{g}) + \mathbf{g}^\top P^\top (W_1^\top \Psi^{-1} W_1 \otimes W_2 W_2^\top) P \mathbf{g} \tag{3.65}$$

In view of (3.63), all the variational parameters contained in Ψ should be treated as hyper-parameters, i.e.

$$\bar{K}_{H,\zeta} = \left[P^\top (W_1^\top \Psi^{-1} W_1 \otimes W_2 W_2^\top) P\right]^{-1}, \quad \zeta = \Psi \tag{3.66}$$

However, this choice generally leaves too many degrees of freedom in shaping of the kernel. In turn this fact has two detrimental effects: *first*, it could lead to overfitting in the final impulse response estimate and *second* it makes the solution of the marginal likelihood maximization problem (2.183) rather involved. These problems do not arise adopting the kernel (3.57) derived above by means of Maximum Entropy arguments, since the number of hyper-parameters is significantly reduced thanks to the specific structure postulated for the regularization matrix (3.52).

[2] The identity Trace[log(A)] = log(det(A)) has been used.

3.4.1.2 Connection with Nuclear Norm minimization approaches

Notice that, when $\zeta^* = [\lambda^*, \lambda^*, 0]$, the trace penalty (3.53) can be rewritten as

$$\mathrm{Tr}\left[\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \widehat{Q}(\zeta^*)\right] = \mathrm{Tr}\left[\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}^\top \lambda^* I_{rp}\right] = \lambda^* \sum_i s_i^2(\mathbf{g}) \tag{3.67}$$

where $s_i(\mathbf{g})$ are the singular values of $\widetilde{\mathbf{G}}$. Thus, the nuclear norm penalty on the (squared) Hankel matrix can be derived as a special case, i.e. for a special choice of the hyper-parameters.

The approach here proposed differs from those discussed in Sections 3.1.2 and 3.2.2 mainly for three reasons. First, a special weighting scheme, depending upon three hyper-parameters is proposed, which is robust against overfitting and reduces bias. Second, casting the nuclear norm minimization step into a Bayesian framework allows to use marginal likelihood approaches to estimate the hyper-parameters: while these techniques have been shown to be robust against noise (Pillonetto and Chiuso, 2015), they also allow to combine the weighted nuclear norm penalty with other penalties (as done in (3.60)). Third, while the works mentioned in Sections 3.1.2 and 3.2.1 adopt a nuclear norm penalty on the Hankel matrix, here the penalty is imposed on the squared Hankel matrix, thus leading to an ℓ_2 penalty on the Hankel singular values. This is essential in order to derive a Gaussian prior, implying that the marginal likelihood is available in closed form. Finally, notice that in the approach here considered, sparsity in the Hankel singular values is favoured by the weighting $\widehat{Q}(\zeta)$.

Remark 3.4.5. The algorithm here proposed, which uses the “squared” Hankel matrix, can be seen as an extension to the matrix case of so-called *reweighted-ℓ_2* algorithm (see e.g. Wipf and Nagarajan (2010)) for sparse estimation; see also Section 3.4.2.3 for more details).

3.4.2 Identification Algorithm

This section describes the iterative algorithm developed to estimate the impulse response $\mathbf{g}$ when the prior (3.59) is chosen. The algorithm alternates between the estimation of $\hat{\mathbf{g}}$ (according to equation (2.148)) for fixed hyper-parameters and marginal likelihood optimization (2.183).

The procedure is summarized in Algorithm 5. For ease of notation the vector $\lambda := [\lambda_0, \lambda_1, \lambda_2]$ has been defined. Consequently, the hyper-parameters vector η in (3.61) can be rewritten as

$$\eta = [\nu, \lambda_0, \zeta] = [\nu, \lambda_0, \lambda_1, \lambda_2, n] = [\nu, \lambda, n] \tag{3.68}$$

Furthermore, $\hat{\mathbf{g}}^{(k)}$, $\hat{\eta}^{(k)}$, $\hat{\lambda}^{(k)}$ and $\hat{n}^{(k)}$ denote estimators at the k-th iteration of the algorithm.

Remark 3.4.6. In Algorithm 5 the noise variance Σ is fixed to e.g. the sample variance of an estimated ARX or FIR model. Of course Σ could also be treated as a hyper-parameter, and estimated with the same procedure based on the marginal likelihood.

Algorithm 5 Identification Algorithm

```
1:  Set the resolution $\epsilon > 0$
2:  Estimate $\hat{\Sigma}$ as illustrated in Remark 3.4.6.
3:  $\hat{n}^{(0)} \leftarrow 0$
4:  $\widehat{U}_{\hat{n}^{(0)}} \equiv \widehat{U}_0 \leftarrow 0_{rp\times rp}$
5:  $\widehat{U}^{\perp}_{\hat{n}^{(0)}} \leftarrow I_{rp}$
6:  $\hat{\nu} \leftarrow \arg\max_{\nu\in\Omega}\ p_y(Y_N|\nu, [1,0,0], \hat{n}^{(0)}, \hat{\Sigma})$
7:  $\hat{\lambda}^{(0)} \leftarrow \arg\max_{\lambda\in\mathbb{R}^3_+}\ p_y(Y_N|\hat{\nu}, \lambda, \hat{n}^{(0)}, \hat{\Sigma})$
8:  $k \leftarrow 0$
9:  while $\hat{n}^{(k)} < pr$ do
10:     $\hat{\mathbf{g}}^{(k)} \leftarrow \mathbb{E}[\mathbf{g}|Y_N, \hat{\eta}^{(k)}, \hat{\Sigma}]$ (using (2.148))
11:     Compute the SVD: $\widetilde{\widetilde{\mathbf{G}}}^{(k)}\widetilde{\widetilde{\mathbf{G}}}^{(k)\top} = \widehat{U}\widehat{S}\widehat{U}^\top$
12:     $\hat{n}^{(k+1)} \leftarrow \hat{n}^{(k)}$
13:     Determine $\widehat{U}_{\hat{n}^{(k+1)}}$ and $\widehat{U}^{\perp}_{\hat{n}^{(k+1)}}$ from $\widehat{U}$
14:     $\hat{\lambda}^{(k+1)} \leftarrow \arg\max_{\lambda\in\mathbb{R}^3_+}\ p_y(Y_N|\hat{\nu}, \lambda, \hat{n}^{(k+1)}, \hat{\Sigma})$
15:     if $p_y(Y_N|\hat{\nu}, \hat{\lambda}^{(k+1)}, \hat{n}^{(k+1)}, \hat{\Sigma}) > (1+\epsilon)p_y(Y_N|\hat{\nu}, \hat{\lambda}^{(k)}, \hat{n}^{(k+1)}, \hat{\Sigma})$ then
16:         $k \leftarrow k+1$
17:     else
18:         $\hat{n}^{(k+1)} \leftarrow \hat{n}^{(k)} + 1$
19:         Perform steps 13 to 14.
20:         if $p_y(Y_N|\hat{\nu}, \hat{\lambda}^{(k+1)}, \hat{n}^{(k+1)}, \hat{\Sigma}) > (1+\epsilon)p_y(Y_N|\hat{\nu}, \hat{\lambda}^{(k)}, \hat{n}^{(k+1)}, \hat{\Sigma})$ then
21:             $k \leftarrow k+1$
22:         else
23:             break
24:         end if
25:     end if
26: end while
27: Return $\hat{\mathbf{g}} \leftarrow \hat{\mathbf{g}}^{(k)}$
```

Notice that the marginal likelihood maximization performed in steps 7 and 14 of Algorithm 5 boils down to the following optimization problem:

$$\hat{\lambda}^{(k)} = \arg\min_{\lambda\in\mathbb{R}^3_+} Y_N^\top \Lambda(\hat{\nu}, \lambda, \hat{n}^{(k)}, \hat{\sigma})^{-1} Y_N + \log|\Lambda(\hat{\nu}, \lambda, \hat{n}^{(k)}, \hat{\Sigma})| \tag{3.69}$$

where

$$\Lambda(\eta, \Sigma) := \widetilde{\Sigma}_N + \Phi_N \bar{K}_{SH,\eta} \Phi_N^\top \tag{3.70}$$

and $\widetilde{\Sigma}_N$ has been defined in equation (2.171). Section 3.4.3 will illustrate a Scaled Gradient Projection (SGP) method appropriately designed to solve (3.69). Issues related to initialisation and convergence of Algorithm 5 are now discussed.

3.4.2.1 Algorithm Initialization

The derivation of kernel $\bar{K}_{SH,\eta}$ in Section 3.4.1 has assumed that a preliminary estimate $\hat{\mathbf{g}}$ was available. Therefore the iterative algorithm outlined in this section has to be provided with an initial estimate $\hat{\mathbf{g}}^{(0)}$. Exploiting the structure of the kernel $\bar{K}_{SH,\eta}$ in (3.60), two straightforward choices are possible:

1. Initialize using only the stable-spline kernel (as the one in (3.27)), i.e.:

$$\begin{aligned} \hat{\mathbf{g}}^{(0)} &= \left(\Phi_N^\top \widetilde{\Sigma}_N^{-1} \Phi_N + \bar{K}_{S,\hat{\nu}^{(0)}}^{-1}\right)^{-1} \Phi_N^\top \widetilde{\Sigma}_N^{-1} Y_N \\ \hat{\eta}^{(0)} &= \left[\hat{\nu}^{(0)}, \hat{\lambda}^{(0)}, 0\right], \quad \hat{\lambda}^{(0)} = [1, 0, 0] \end{aligned} \tag{3.71}$$

   where only the hyper-parameters $\hat{\nu}^{(0)}$ are estimated through marginal-likelihood maximization (2.183).

2. Initialize using the stable-Hankel kernel with $\hat{n} = 0$, so that no preliminary estimate is needed to initialize $\hat{U}_n$ (which is empty) and thus $\hat{Q}(\hat{\zeta}^{(0)}) = \hat{\lambda}_2^{(0)} I$:

$$\begin{aligned} \hat{\mathbf{g}}^{(0)} &= \left(\Phi_N^\top \widetilde{\Sigma}_N^{-1} \Phi_N + \bar{K}_{SH,\hat{\eta}^{(0)}}^{-1}\right)^{-1} \Phi_N^\top \widetilde{\Sigma}_N^{-1} Y_N \\ \hat{\eta}^{(0)} &= \left[\hat{\nu}^{(0)}, \hat{\lambda}^{(0)}, 0\right], \qquad \hat{\lambda}^{(0)} = \left[1, \hat{\lambda}_2^{(0)}, \hat{\lambda}_2^{(0)}\right] \end{aligned} \tag{3.72}$$

   where $\hat{\nu}^{(0)}$ and $\hat{\lambda}_2^{(0)}$ are estimated through marginal likelihood maximization (2.183).

The procedure that is actually followed in Algorithm 5 combines the two strategies above. Namely, the first approach is adopted to fix the hyper-parameters $\hat{\nu}$ defining the stable-spline kernel (line 6). These are then kept fixed for the whole procedure. We then follow the second strategy to estimate $\hat{\lambda}^{(0)}$ (line 7). Note that in line 7 the hyper-parameters ν are fixed to $\hat{\nu}$ and not estimated as in (3.72). Analogously, $\hat{\lambda}_0^{(0)}$ is estimated by marginal-likelihood maximization and not set a-priori to 1 as in (3.72). Therefore, the estimate $\hat{\mathbf{g}}^{(0)}$ computed at line 10 is derived by adopting the kernel $\bar{K}_{SH,\hat{\eta}^{(0)}}$ with $\hat{\eta}^{(0)} = [\hat{\nu}, \hat{\lambda}^{(0)}, 0]$.

This sort of "hybrid" strategy has been chosen for two main reasons. *First*, it allows to fix the hyper-parameters ν by solving a simplified optimization problem (w.r.t. solving a problem involving all the hyper-parameters η). Notice that this also provides the user with a certain freedom on the choice of the kernel $\bar{K}_{S,\nu}$: using other kernel structures (see e.g. Chiuso et al. (2014)) additional properties (e.g. resonances, high-frequency components, etc.) of the impulse response can be accounted for. *Second*, it also allows to properly initialize the iterative procedure used to update the hyper-parameters λ_0 and ζ in (3.61), until a stopping condition is met (see next section for a discussion about convergence of Algorithm 5).

3.4.2.2 Convergence Analysis

Algorithm 5 is guaranteed to stop in a finite number of steps, returning a final estimate $\hat{\mathbf{g}}$. Indeed, at any iteration k four possible scenarios may arise:

1. Condition at line 15 is met and k is increased by one and the algorithm iterates.

2. Condition at line 15 is not met[3], so that $\hat{n}$ is increased by one, and condition 20 is not met, then the algorithm terminates returning $\hat{\mathbf{g}} := \hat{\mathbf{g}}^{(k)}$.

3. Condition at line 15 is not met [4], so that $\hat{n}$ is increased by one, while condition 20 is met, then k is increased by one and the algorithm iterates.

4. $\hat{n}^{(k)} = pr$, then the algorithm terminates returning $\hat{\mathbf{g}} := \widehat{\mathbf{g}}^{(k)}$.

Conditions (1) and (3) may only be satisfied a finite number of times, thus the algorithm terminates in a finite number of steps.

It should also be stressed that Algorithm 5 is only an ascent algorithm w.r.t. the marginal likelihood function without any guarantee of convergence to a local extrema. If $\hat{U}_{\hat{n}^{(k)}}$ was treated as a hyper-parameter and the marginal likelihood optimised over the Grassmann manifold, then convergence to a local maxima could be proven.[4] Notice indeed that a tailored Scaled Gradient Projection algorithm will be adopted to solve the marginal likelihood optimization problem at line 14 (see Algorithm 1 and Section 3.4.3): every accumulation point of the iterates generated by this algorithm is guaranteed to be a stationary point (Bonettini et al. (2015), Theorem 1); furthermore, for the specific problem here solved, the sequence of the iterates admits at least one limit point.

[3] This certainly happens after a finite number of iterations for any positive resolution ϵ and fixed $\hat{n}$.

[4] This variant has been tested, despite it is considerably more computationally expensive than Algorithm 1. Since no significant improvements have been observed, the simpler version is here presented.

Once the algorithm has converged, $\hat{n}$ is the optimal dimension of the "signal" subspace of $\widetilde{\mathbf{G}}$, respectively spanned by the columns of $\widehat{U}_{\hat{n}}$ and $\widehat{U}_{\hat{n}}^{\perp}$. Furthermore, the corresponding multipliers λ_1 and λ_2 in ζ are expected to tend, respectively, to 0 (meaning that no penalty is given on the signal component) and to ∞ (that is, a very large penalty is assigned to the noise subspace); if $\hat{\lambda}_2 = \infty$, $\hat{n}$ would actually be the McMillan degree of the estimated system.
In practice the estimated hyper-parameter $\hat{\lambda}_2$ is finite and, similarly, $\hat{\lambda}_1$ is strictly positive. As a result the McMillan degree of the estimated system is generically larger than, but possibly close to, $\hat{n}$. Therefore, estimation of the integer parameter n should not be interpreted as a hard decision on the complexity as instead happens for parametric model classes whose structure is estimated with AIC/BIC/Cross Validation. It could be said that Algorithm 5 performs a "soft" complexity selection, confirming that this Bayesian framework allows to describe model structures in a continuous manner: in fact, for any choice of $\hat{n}$, systems of different McMillan degrees are assigned non zero probability by the prior.

3.4.2.3 Connection with Iterative Reweighted Algorithms

Algorithm 5 shares key properties with the so-called iterative reweighted algorithms, proposed by Mohan and Fazel (2012) and Wipf and Nagarajan (2010). Considering a rank minimization problem, the algorithm introduced in Mohan and Fazel (2012) adopts a weighted trace heuristic as a surrogate to the rank function and iteratively updates the weighting matrix by means of a closed form expression depending on the current optimal point. The trace heuristic considered in Mohan and Fazel (2012) has a clear analogy to the penalty term (3.53), in which $\widehat{Q}(\zeta)$ plays the role of a weighting matrix. Also the structure of the matrix $\widehat{Q}(\zeta)$ in (3.52) resembles that of the weighting matrix in Mohan and Fazel (2012). Specifically, following the approach in Mohan and Fazel (2012), the weighting $\widehat{Q}^{(k)}$ at iteration k would be

$$\widehat{Q}^{(k)} = \left(\widehat{\widetilde{\mathbf{G}}}^{(k-1)} \left(\widehat{\widetilde{\mathbf{G}}}^{(k-1)} \right)^{\top} + \epsilon I_{pr} \right)^{-1} = \left(\widehat{U}\widehat{S}\widehat{U}^{\top} + \epsilon I_{pr} \right)^{-1} \tag{3.73}$$

where $\widehat{S}$ denotes the singular values matrix and ϵ is the regularization factor introduced in order to avoid numerical issues in the matrix inversion operation. Instead, Algorithm 5 adopts

$$\widehat{Q}^{(k)}(\hat{\zeta}) = \left(\frac{1}{\hat{\lambda}_1} \widehat{U}_{\hat{n}} \widehat{U}_{\hat{n}}^{\top} + \frac{1}{\hat{\lambda}_2} \widehat{U}_{\hat{n}}^{\perp} \left(\widehat{U}_{\hat{n}}^{\perp} \right)^{\top} \right)^{-1} = \left(\left(\frac{1}{\hat{\lambda}_1} - \frac{1}{\hat{\lambda}_2} \right) \widehat{U}_{\hat{n}} \widehat{U}_{\hat{n}}^{\top} + \frac{1}{\hat{\lambda}_2} I_{pr} \right)^{-1}$$

$$\bar{n} := \hat{n}^{(k-1)} \tag{3.74}$$

The similarity between (3.73) and (3.74) is apparent with $1/\hat{\lambda}_2$ playing the role of the regularization parameter ϵ and[5] $\left(\frac{1}{\hat{\lambda}_1} - \frac{1}{\hat{\lambda}_2}\right) \widehat{U}_{\bar{n}} \widehat{U}_{\bar{n}}^\top$ being a rescaled and truncated version of $\widehat{U}\widehat{S}\widehat{U}^\top$.

This peculiar structure of the weighting matrix, which arises from the Maximum Entropy derivation of the prior, acts as an hyper-regularizer which helps preventing overfitting; the hierarchical Bayesian model provides a natural framework based on which regularization can be tuned through the choice of $\hat{\lambda}_1$ and $\hat{\lambda}_2$ (see line 14 of Algorithm 5).

The Bayesian framework here adopted also connects Algorithm 5 to the non-separable reweighting scheme proposed in Wipf and Nagarajan (2010) for solving a Sparse Bayesian Learning (SBL) problem: the algorithm iteratively alternates the computation of the optimal estimate and the closed-form update of the hyper-parameters matrix, as the algorithm we propose. The main difference between the cited procedures and Algorithm 5 lies in the special structure of the weighting $\widehat{Q}(\zeta)$, which makes the weighting $\bar{K}_{SH,\eta}$ dependent on the hyper-parameter vector $\lambda = [\lambda_0, \lambda_1, \lambda_2]$ and n in a way such that closed form expressions for its update are not available.

3.4.3 SGP for Marginal Likelihood Optimization

A crucial step in Algorithm 5 is the marginal likelihood maximization (step 14) which is computationally expensive, especially when the number of inputs and outputs is large. To deal with this issue the Scaled Gradient Projection method (SGP), proposed in Bonettini et al. (2015) and illustrated in Algorithm 1 has been adapted to solve

$$\min_{\lambda \in \mathbb{R}^3_+} f_{ML}(\lambda) \tag{3.75}$$

$$f_{ML}(\lambda) = Y_N^\top \Lambda(\hat{\nu}, \lambda, \hat{n}, \hat{\Sigma})^{-1} Y_N + \log |\Lambda(\hat{\nu}, \lambda, \hat{n}, \hat{\Sigma})| \tag{3.76}$$

As observed in Section 2.4.5.2, the choice of the scaling matrix $D^{(k)}$ appearing in step 5 of Algorithm 1, is crucial. Indeed, its structure depends on both the objective function and the constraints of the optimization problem. The proposed implementation follows the choices made in Bonettini et al. (2015): $D^{(k)}$ is set to be diagonal and its update is based on the split gradient idea, shown in equation (2.201). Define

$$f_0(\lambda) := Y_N^\top \Lambda(\lambda)^{-1} Y_N, \qquad f_1(\lambda) := \log |\Lambda(\lambda)| \tag{3.77}$$

[5] Note that, even though no such constrained has been introduced, $\hat{\lambda}_1 \leq \hat{\lambda}_2$, so that $\left(\frac{1}{\hat{\lambda}_1} - \frac{1}{\hat{\lambda}_2}\right) > 0$.

where the simplified notation $\Lambda(\lambda) \equiv \Lambda(\hat{\nu}, \lambda, \hat{n}, \hat{\sigma})$ has been used. Moreover, let

$$\bar{K}_{SH,\lambda} := [\lambda_0\Gamma_0 + \lambda_1\Gamma_1 + \lambda_2\Gamma_2]^{-1} \tag{3.78}$$

where $\hat{\nu}$ and $\hat{n}$ are fixed and

$$\Gamma_0 = \bar{K}_{S,\hat{\nu}}^{-1} \tag{3.79}$$

$$\Gamma_1 = P^\top \left(W_1^\top \widehat{U}_{\hat{n}} \widehat{U}_{\hat{n}}^\top W_1 \otimes W_2 W_2^\top\right) P \tag{3.80}$$

$$\Gamma_2 = P^\top \left(W_1^\top \widehat{U}_{\hat{n}}^\perp \left(\widehat{U}_{\hat{n}}^\perp\right)^\top W_1 \otimes W_2 W_2^\top\right) P \tag{3.81}$$

Now, indicating with $[f'_{ML}(\lambda)]_i$ the gradient of f_{ML} w.r.t. to $\lambda_i,\ i = 0, 1, 2$, it follows:

$$[f'_0(\lambda)]_i = Y_N^\top \Lambda(\lambda)^{-1} \Upsilon(\lambda) \Lambda(\lambda)^{-1} Y_N \tag{3.82}$$

$$[f'_1(\lambda)]_i = -\text{Tr}\left\{\Lambda(\lambda)^{-1}\Upsilon(\lambda)\right\} \tag{3.83}$$

$$\Upsilon(\lambda) : = \Phi_N \bar{K}_{SH,\lambda} \Gamma_i \bar{K}_{SH,\lambda} \Phi_N^\top \tag{3.84}$$

From the positive definiteness of $\Lambda(\lambda)$ and the positive semidefiniteness of $\Upsilon(\lambda)$, it results that $[f'_0(\lambda)]_i \geq 0,\ \forall\lambda \in \mathbb{R}^3$. Furthermore, from Lemma II.1 in Lasserre (1995), it follows that $[f'_1(\lambda)]_i < 0,\ \forall\lambda \in \mathbb{R}^3$. This shows how the gradient of the objective function (3.76) admits the following decomposition:

$$f'_{ML}(\lambda) = f'_0(\lambda) + f'_1(\lambda) = U(\lambda) - V(\lambda) \tag{3.85}$$

with $U(\lambda) = f'_0(\lambda) \geq 0$ and $V(\lambda) = -f'_1(\lambda) > 0$ (here the inequalities have to be understood component wise). Following the derivation illustrated in Section 2.4.5.2, the scaling matrix $D^{(k)}$ is the defined as:

$$\left[D^{(k)}\right]_{ii} = \min\left(\max\left(L_{min}, \frac{\lambda_i^{(k)}}{V_i(\lambda^{(k)})}\right), L_{max}\right) \tag{3.86}$$

Further details on the setting of the parameters involved in Algorithm 1 and on the adopted stopping criterion will be given in Section 3.5.6.

3.5 Numerical Results

The identification procedure outlined in Algorithm 5 is here compared with off-the-shelf identification routines, as well as with recently proposed methods (see Section 3.1, 3.2

and 3.3).

3.5.1 Data

The numerical comparison is performed through some Monte-Carlo studies on three appropriately designed scenarios. The innovation process $e(t)$ appearing in all of them is a zero-mean Gaussian white noise with standard deviation randomly chosen in order to guarantee that the SNR on each output channel is a uniform random variable in the interval $[1, 4]$. For each scenario the identification procedures are tested on three different data lengths, which can be roughly classified as "few/average/many" data. Each Monte-Carlo study includes $N_{MC} = 200$ runs. A brief illustration of the three scenarios follows.

S1: A fixed fourth order system with transfer function $G(q) = C(qI - A)^{-1}B$ is considered, with

$$\begin{aligned} A &= \text{blockdiag}\left(\begin{bmatrix} 0.8 & 0.5 \\ -0.5 & 0.8 \end{bmatrix}, \begin{bmatrix} 0.2 & 0.9 \\ -0.9 & 0.2 \end{bmatrix}\right) \\ B &= [1\ 0\ 2\ 0]^\top \quad C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0.1 & 0 & 0.1 \\ 20 & 0 & 2.5 & 0 \end{bmatrix} \end{aligned} \tag{3.87}$$

The input is generated, for each Monte Carlo run, as a low pass filtered white Gaussian noise with normalized band $[0, \varrho]$ where ϱ is a uniform random variable in the interval $[0.8, 1]$. The identification of system (6.39) using data generated by a band-limited input appears particularly challenging because the system is characterized by two high-frequency resonances.
The three different data lengths that have been considered are: $N_{1,1} = 200$, $N_{1,2} = 500$, $N_{1,3} = 1000$.

S2: For each Monte Carlo run $G(q)$ is randomly generated using the MATLAB function `drmodel` with 5 outputs and 5 inputs while guaranteeing that all the poles of $G(q)$ are inside the disc of radius .85 of the complex plane. The system orders are randomly chosen from 1 to 10. The input $u(t)$ is zero-mean unit variance white Gaussian noise. The three different numbers of input-output data pairs that have been tested are: $N_{2,1} = 350$, $N_{2,2} = 500$, $N_{2,3} = 1000$.

S3: The systems have been randomly generated similarly to scenario S2, but with 10 inputs and 5 outputs. Moreover, the input $u(t)$ is a low-pass filtered Gaussian white noise with normalized band defined as in S1. The considered data lengths are: $N_{3,1} = 600,\ N_{3,2} = 800,\ N_{3,3} = 1000$.

Remark 3.5.1. The MATLAB routine `drmodel` produces a random stable model of the specified order and returns either its transfer function coefficients or its state-space matrices.

3.5.2 Identification Algorithms

The following algorithms have been tested:[6]

N4SID+Or: The subspace method, as implemented by the MATLAB routine `n4sid`. Different model complexities are tested; an Oracle chooses the order which maximises the impulse response fit (3.91).

N4SID(OE)+Or: As N4SID+Or but forcing the routine to return an Output-Error model.

N4SID: The MATLAB routine `n4sid`, equipped with default model order selection.

N4SID(OE): Same as N4SID but forcing an OE structure.

PEM+Or: PEM as implemented by the MATLAB routine `pem`. Different model complexities are tested: an Oracle chooses the order which maximises the impulse response fit (3.91).

PEM(OE)+Or: Same as PEM+Or but using the routine `oe`. For each of the tested complexities, the routine `oe` has been initialized with the model returned by `pem`.

PEM: The MATLAB routine `pem`, equipped with the default model order selection.

PEM(OE): The MATLAB routine `oe`, initialized with the model returned by `pem` (order as selected by the default choice in `pem`).

N2SID: The identification routine detailed in (3.18) and implemented through the code available from `http://users.isy.liu.se/en/rt/hansson/`. This routine returns a state-space model in innovation form. The estimation of Output-Error models through N2SID has not been tested, since the routine does not straightforwardly allow to force an OE model structure.

[6] Some methods appeal to an Oracle (Or) who knows the true system. Clearly these are not feasible in practice and are only reported for the sake of comparison.

SS: The estimator (2.148) where $\bar{K}_\eta$ is chosen to be the TC kernel (3.27) and the hyper-parameters η are estimated through marginal likelihood maximization. The estimator is computed through the MATLAB routine `arxRegul` (imposing a FIR model structure).

NN+CV: A FIR model of order T estimated solving

$$\hat{\mathbf{g}} = \arg\min_{\mathbf{g}\in\mathbb{R}^{pmT}} \|Y_N - \Phi_N \mathbf{g}\|^2 + \lambda^* \|\mathbf{G}\|_* \tag{3.88}$$

The optimization problem is solved through a tailored ADMM algorithm (as in Liu et al. (2013)), while λ^* is determined through Cross-Validation. This procedure has also been tested by replacing $\mathbf{G}$ in (6.40) with $\widetilde{\mathbf{G}}$ (see (3.40)).

RNN+CV: A FIR model of order T estimated by iteratively solving

$$\hat{\mathbf{g}} = \arg\min_{\mathbf{g}\in\mathbb{R}^{pmT}} \|Y_N - \Phi_N \mathbf{g}\|^2 + \lambda^* \|W_l \mathbf{G} W_r\|_* \tag{3.89}$$

The weight matrices W_l and W_r are updated at each iteration according to the procedure suggested in Mohan and Fazel (2010). λ^* is selected through Cross-Validation. The case in which $\mathbf{G}$ in (6.41) is replaced with $\widetilde{\mathbf{G}}$ has also been tested.

SH: The estimator returned by Algorithm 5 with $\bar{K}_{S,\nu}$ specified through the TC kernel.

Some implementation details follow. For SS, SH, NN+CV and RNN+CV, the length T of the estimated impulse response $\hat{\mathbf{g}}$ is set to 80 for scenario S1, to 50 for S2 and S3.
The regularization parameter λ in N2SID (Verhaegen and Hansson, 2014) is chosen within a set of 20 elements logarithmically spaced between 10^{-3} and 10^{-1} for S1 and 40 elements logarithmically spaced between 10^{-3} and 10^{5} for S2 and S3. The endpoints of these grids are selected so that the estimated value of λ is inside the interval.
The techniques directly based on the nuclear norm, i.e. NN+CV and RNN+CV, are only applied on the "average/large" data lengths scenarios, that is for $N_{i,2}$ and $N_{i,3}$, $i = 1, 2, 3$. In fact, since the regularization parameter in this case is estimated using cross-validation (which requires splitting the data in validation and identification subsets), the results are unreliable for the "few" data set scenarios $N_{i,1}$. In order to optimize the performance, in scenarios S2 and S3 two-thirds of the available data are used as training set and the remaining one third for the validation stage. Instead, in scenario S1, the available data are equally split into the training and the validation set. The regularization parameter λ^* is selected from the vector $\tilde{v} = \frac{v}{N_{tr}}$, where N_{tr} is the length of the training dataset,

while v is a vector of 25 elements logarithmically spaced between 10^2 and 10^7 for S1, between 10^3 and 10^7 for S2 and S3.

3.5.3 Impulse Response Estimate

To evaluate the estimators described above, the so-called coefficient of determination (COD) between time series a and b is introduced:

$$\operatorname{cod}(a^{N_c}, b^{N_c}) = 100 \left(1 - \sqrt{\frac{\sum_{k=1}^{N_c}(a(k)-b(k))^2}{\sum_{k=1}^{N_c}(a(k)-\bar{a})^2}} \right) \tag{3.90}$$

where $\bar{a} = \frac{1}{N_c}\sum_{k=1}^{N_c} a(k)$. The impulse response fit is measured using the average COD:

$$\mathcal{F}_{N_c}(\hat{\mathbf{g}}) := \frac{1}{pm}\sum_{i=1}^{p}\sum_{j=1}^{m} \operatorname{cod}\left(\left[g_0^{N_c}\right]_{ij}, \hat{g}_{ij}^{N_c} \right) \tag{3.91}$$

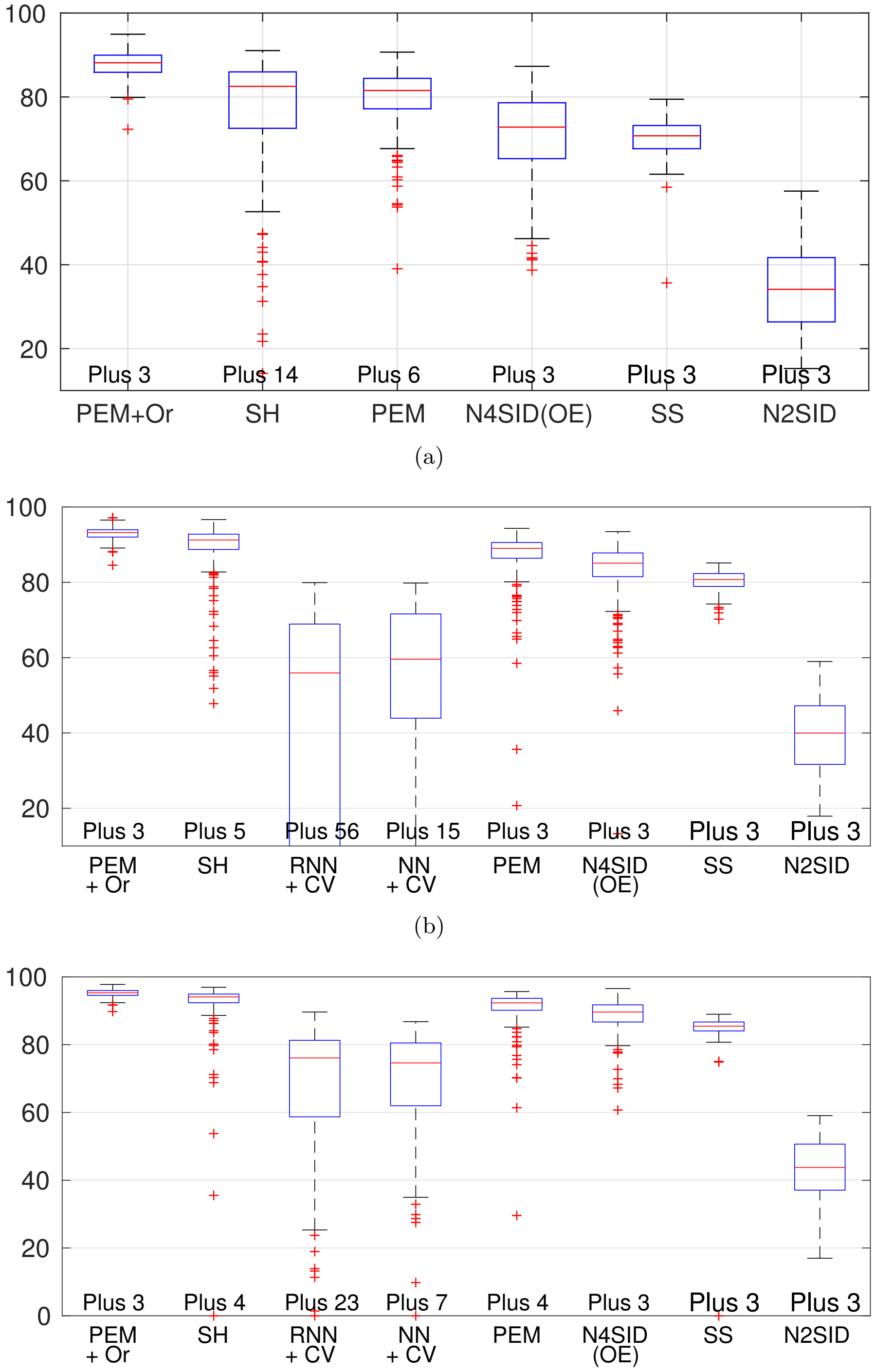
where $\left[g_0^{N_c}\right]_{ij}$ and $\hat{g}_{ij}^{N_c}$ denote the true and the estimated impulse responses from input j to output i, with $\hat{g}_{ij}(k) = 0,\ k = T+1, ..., N_c$. N_c is set to 1000.

Figures 3.2, 3.3 and 3.4 report the boxplots of (3.91) in the three scenarios detailed in Section 3.5.1 for some of the identification techniques listed in Section 3.5.2. In particular, among the methods equipped with the oracle for model complexity selection, only the results of PEM+Or are shown, since it gives the best performance. As far as the subspace techniques are concerned, only N4SID(OE) is reported, because it generally performs slightly better than N4SID; analogously, only the results achieved by the routine PEM are illustrated, since the performance of PEM(OE) is worse.

SH and RNN+CV achieve, among the procedures which can be practically implemented, the best performance in scenarios S2 and S3; instead, in scenario S1, RNN+CV has severe difficulties. It is also interesting to observe that the reweighted procedure in (6.41) (RNN+CV) improves the performance achieved by simple nuclear norm regularization (NN+CV) in all the scenarios except for S1. The results achieved imposing the nuclear norm penalty on the weighted Hankel matrix $\widetilde{\mathbf{G}}$ are not reported since they are in general slightly worse than those achieved by NN+CV and RNN+CV.

3.5.4 Predictive Performance

The predictive performance of the methods listed in Section 3.5.2 are here compared over a specifically designed scenario. Namely, system (6.39) is simulated with a unit variance white Gaussian noise input, while its output is corrupted by additive white Gaussian noise

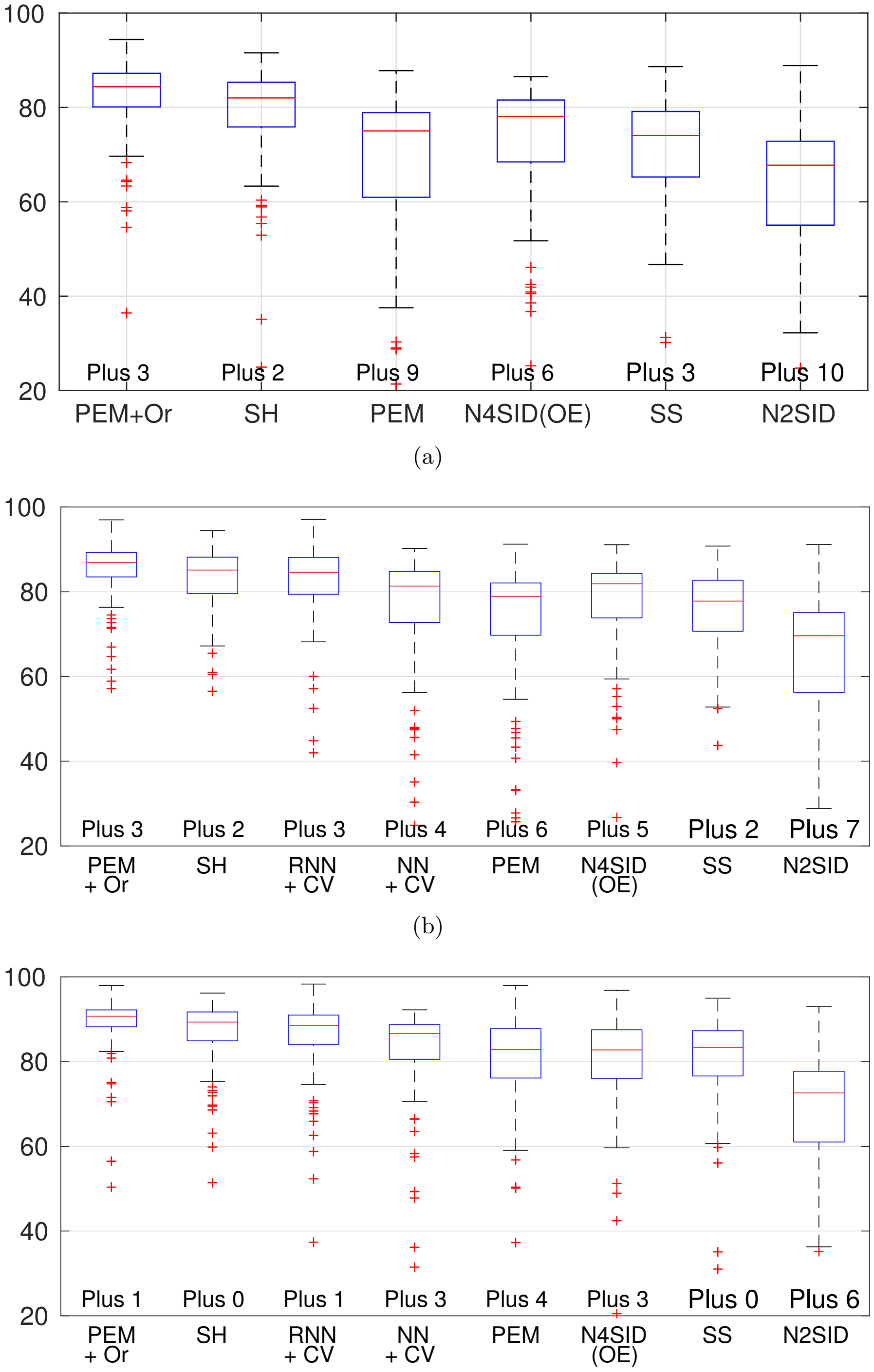


(a)

(b)

(c)

Figure 3.2: Scenario S1 - Impulse response fit (3.91) achieved by the identification algorithms listed in Section 3.5.2. Different data lengths are evaluated: $N_{1,1} = 200$ (a), $N_{1,2} = 500$ (b) and $N_{1,3} = 1000$ (c).

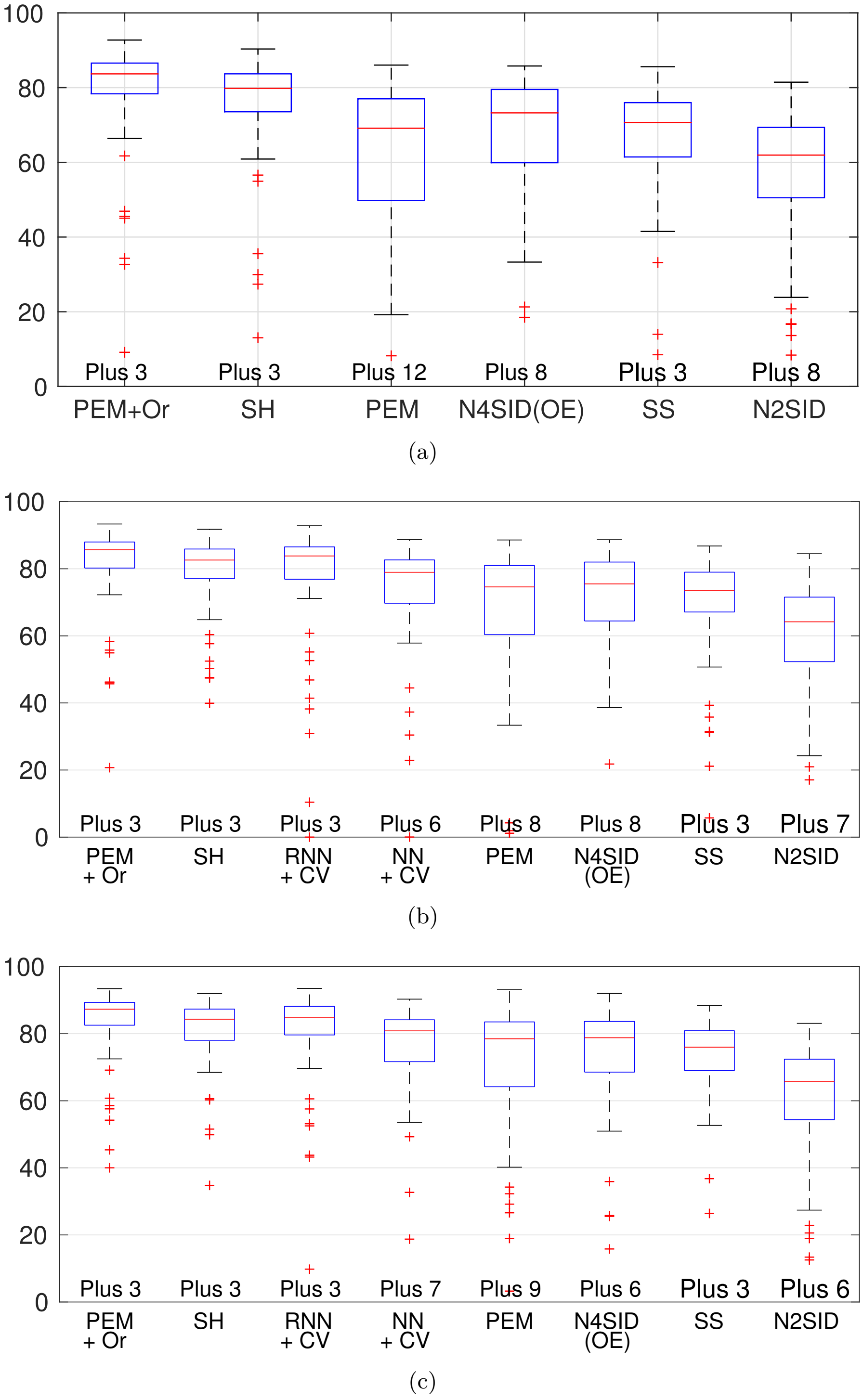


(a)

(b)

(c)

Figure 3.3: Scenario S2 - Impulse response fit (3.91) achieved by the identification algorithms listed in Section 3.5.2. Different data lengths are evaluated: $N_{2,1} = 350$ (a), $N_{2,2} = 500$ (b) and $N_{2,3} = 1000$ (c).



(a)

(b)

(c)

Figure 3.4: Scenario S3 - Impulse response fit (3.91) achieved by the identification algorithms listed in Section 3.5.2. Different data lengths are evaluated: $N_{3,1} = 600$ (a), $N_{3,2} = 800$ (b) and $N_{3,3} = 1000$ (c).

Table 3.1: Modified Scenario S1 - Median, 5th and 95th percentiles over 200 Monte-Carlo runs of $\text{cod}(\tilde{y}_i^{N_{val}}, \hat{y}_i^{N_{val}})$, $N_{val} = 500$ (see (3.90)). Estimators are computed using 500 data (the best values among the realistic methods are highlighted in bold).

	$\text{cod}(\tilde{y}_1^{N_{val}}, \hat{y}_1^{N_{val}})$			$\text{cod}(\tilde{y}_2^{N_{val}}, \hat{y}_2^{N_{val}})$			$\text{cod}(\tilde{y}_3^{N_{val}}, \hat{y}_3^{N_{val}})$		
	md	5th pctl	95th pctl	md	5th pctl	95th pctl	md	5th pctl	95th pctl
PEM+Or	92.54	87.69	95.94	92.76	88.77	96.14	92.74	88.06	95.86
SH	**91.48**	**86.85**	**95.03**	**91.55**	**86.60**	**95.31**	**91.46**	**86.80**	**94.83**
RNN+CV	71.27	65.35	76.38	69.94	64.48	74.83	72.35	65.44	81.98
NN+CV	72.18	66.44	76.81	69.38	63.94	74.29	84.17	78.80	89.76
PEM	85.75	59.86	92.46	86.12	65.15	92.84	83.65	52.76	90.63
N4SID(OE)	82.42	70.05	89.71	81.85	66.69	90.22	88.36	80.80	92.39
SS	80.14	76.19	84.02	80.06	75.77	83.16	82.04	76.43	85.96
N2SID	34.78	11.85	51.04	26.59	7.57	43.34	58.95	49.26	65.79

with a variance chosen in order to have SNR= 2. 200 estimation datasets consisting of $N = 500$ data are generated in this way. A set of validation data $\widetilde{\mathcal{D}}^{N_{val}} = \{\tilde{u}^{N_{val}}, \tilde{y}^{N_{val}}\}$ is used to evaluate the COD for each system output, i.e. $\text{cod}(\tilde{y}_i^{N_{val}}, \hat{y}_i^{N_{val}})$, $i = 1, ..., p$, (see definition in (3.90)) with $\hat{y}_i(t)$ denoting the one-step ahead predictor for the i-th output channel. Table 3.1 compares the median, the 5th and the 95th percentiles of $\text{cod}(\tilde{y}_i^{N_v}, \hat{y}_i^{N_v})$ achieved by the considered identification methods.

3.5.5 Estimated Hankel Singular Values

Figures 3.5, 3.6 and 3.7 are concerned with the ability in estimating the Hankel singular values, which are grouped in the so called "signal singular values" (corresponding to the nonzero singular values of the true system) and "noise singular values" (corresponding to the zero singular vaues of the the true system). Indeed, the top plot in each figure shows the boxplots of the error on the "signal singular values":

$$\Delta_{signal}(\hat{\mathbf{g}}) := \sum_{i=1}^{\bar{n}} |\tilde{s}_i(\mathbf{g}_0) - \tilde{s}_i(\hat{\mathbf{g}})| \tag{3.92}$$

where $\mathbf{g}_0$ is the true impulse response vector, $\hat{\mathbf{g}}$ is the estimated one, $\tilde{s}_i(\mathbf{g})$ is the i-th normalized Hankel singular value and $\bar{n}$ here denotes the true system order. Similarly, the bottom plot contains the boxplots of the error on the "noise singular values":

$$\Delta_{noise}(\hat{\mathbf{g}}) := \sum_{i=\bar{n}+1}^{pr} |\tilde{s}_i(\mathbf{g}) - \tilde{s}_i(\hat{\mathbf{g}})| = \sum_{i=\bar{n}+1}^{pr} \tilde{s}_i(\hat{\mathbf{g}}) \tag{3.93}$$

Figure 3.5 shows that the poor performance observed in Figure 3.2 for NN+CV and RNN+CV is determined by the failure in detecting the "true" system complexity (as proven by the large error in the estimation of the "noise" singular values which can be

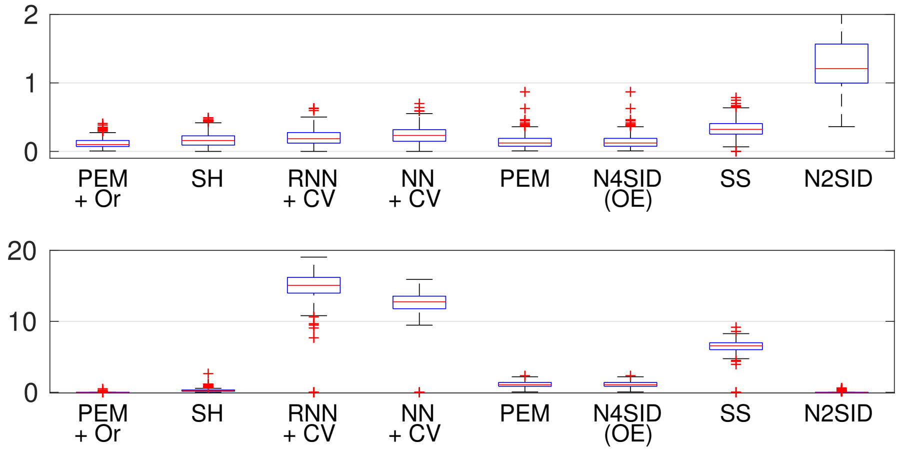


Figure 3.5: Scenario S1 - *Top*: Sum of absolute errors on the "signal" normalized Hankel singular values (see (3.92)). *Bottom*: Sum of absolute errors on the "noise" normalized Hankel singular values (see (3.93)). Considered data length: $N_{1,2} = 500$.

interpreted as overestimation of the system order). On the other hand, the unsatisfying performance of N2SID in Figure 3.2 is due to the under-estimation of the system complexity, which leads to a large bias in the estimation of the true Hankel singular values (top of Figure 3.5) and to the correct detection of the "noise" subspace. Among the feasible methods, SH seems to correctly estimate the system complexity in most cases.

With regards to scenarios S2 and S3, the joint analysis of Figures 3.3, 3.6 and 3.4, 3.7 reveals how the good performance in terms of impulse response fit achieved by PEM+Or and RNN+CV are mainly due to the correct reconstruction of the "noise" subspace; indeed, the performance of SH in terms of fit are slightly worse even if it better recovers the "signal" subspace. A deeper inspection reveals that the system complexity is underestimated by PEM+Or, RNN+CV and N2SID, thus explaining the almost perfect reconstruction of the "noise" subspace and the bias which affects the estimates of the "signal" subspace. This observation suggests that the good performance observed for RNN+CV in Figures 3.3 and 3.4 are favored by the nature of the systems in scenarios S2 and S3: indeed, underestimation of the system order does not have a detrimental effect in these scenarios where there are many "small" Hankel singular values.

Comparing the performance of NN+CV and RNN+CV in Figures 3.6 and 3.7, it is clear that the reweighted procedure significantly increases the degree of sparsity in the estimated Hankel singular values.

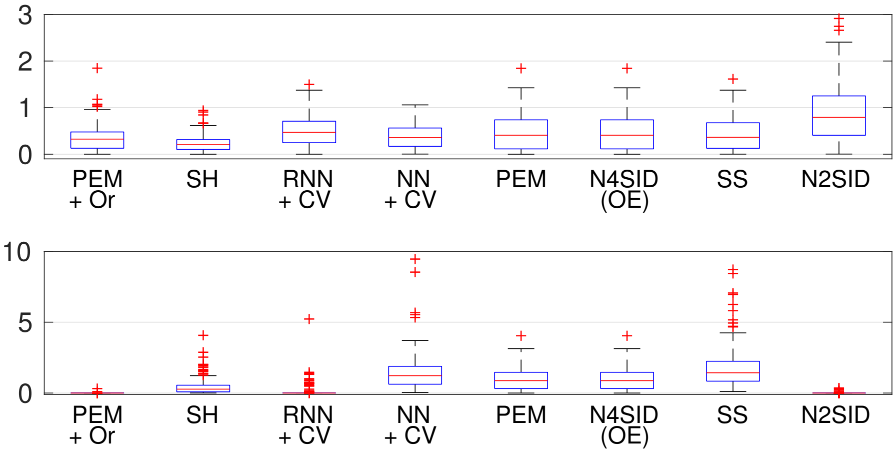


Figure 3.6: Scenario S2 - *Top*: Sum of absolute errors on the “signal” normalized Hankel singular values (see (3.92)). *Bottom*: Sum of absolute errors on the “noise” normalized Hankel singular values (see (3.93)). Considered data length: $N_{2,2} = 500$.

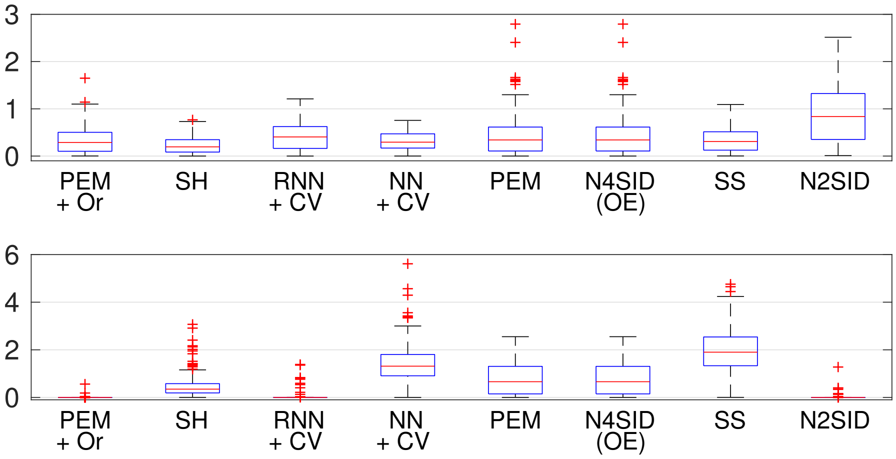


Figure 3.7: Scenario S3 - *Top*: Sum of absolute errors on the “signal” normalized Hankel singular values (see (3.92)). *Bottom*: Sum of absolute errors on the “noise” normalized Hankel singular values (see (3.93)). Considered data length: $N_{3,2} = 800$.

Table 3.2: Computational time (in sec) required to estimate a system: median, 5th and 95th percentiles over 200 Monte-Carlo runs. Estimators are computed using $N_{\cdot,3} = 1000$ data (best values among the realistic methods are highlighted in bold).

	S1			S2			S3		
	md	5th pctl	95th pctl	md	5th pctl	95th pctl	md	5th pctl	95th pctl
SH	84.89	43.70	175.24	67.22	37.49	548.89	276.62	129.07	775.38
RNN+CV	418.93	206.28	1287.55	95.87	68.84	584.58	285.70	196.60	615.72
NN+CV	63.72	58.29	69.50	49.51	39.46	206.27	132.90	110.32	193.97
PEM	3.12	2.44	4.72	1.60	**0.70**	12.89	11.47	**1.01**	**31.95**
N4SID(OE)	**1.54**	1.48	**1.67**	**1.46**	0.96	**8.61**	**7.99**	1.82	36.34
SS	1.64	**1.47**	1.84	10.51	8.86	13.23	31.33	25.58	44.09
N2SID	666.74	508.86	851.72	576.04	462.73	732.81	504.84	402.18	764.83

3.5.6 Computational Time

A comparison of the methods listed in Section 3.5.2 is now done in terms of computational time. All algorithms were run on a server with two quad core Intel Xeon E5450 processor at 3.00 GHz, 12 MB cache and 16 GB of RAM under MATLAB2014b.
Table 5.1 reports the median, the 5th and 95th percentiles of the computational time over the 200 systems of scenarios S1, S2 and S3, showing a clear gap in the performance of off-the-shelf methods (PEM, N4SID and SS) and non-off-the-shelf ones (SH, NN, RNN and N2SID); among the latter, the algorithm here proposed (SH) appears to be the least demanding one.

In Section 3.4.3 a tailored Scaled Gradient Projection (SGP) method has been illustrated to solve the Marginal Likelihood maximization problem at step 14 of Algorithm 5 (see also (3.69)). To assess the benefits of SGP, two implementations of Algorithm 5 are compared: both solve the optimization problem (3.69) using, respectively, the MATLAB routine `fmincon` and the SGP Algorithm 1. In Table 3.3 execution times are reported for the three scenarios described in Section 3.5.1.

The routine `fmincon` uses the interior-point algorithm and the default parameters setting (similar performance have been obtained through other algorithms, such as SQP or trust-region-reflective). The parameters involved in the SGP routine (Algorithm 1) are set as follows: $\kappa = 10^{-4}$, $\rho = 0.4$, $\alpha_{min} = 10^{-7}$, $\alpha_{max} = 10^{2}$, $L_{min} = 10^{-5}$, $L_{max} = 10^{10}$. The following stopping criterion is adopted:

$$f_{ML}(\lambda^{(k)}) - f_{ML}(\lambda^{(k+1)}) < 10^{-9}|f_{ML}(\lambda^{(k+1)})|$$

For both the algorithms the maximum number of iterations has been fixed to 5000.

Table 3.3: Computational time (in sec) required to estimate a system: median, 5th and 95th percentiles over 200 Monte-Carlo runs. Estimators are computed using $N_{\cdot,3} = 1000$ data.

	S1			S2			S3		
	md	5th pctl	95th pctl	md	5th pctl	95th pctl	md	5th pctl	95th pctl
`fmincon`	1358.30	853.80	1893.10	2545.10	1322.80	4816.80	6651.60	2951.60	12732.00
SGP	84.89	43.70	175.24	67.22	37.49	548.89	276.62	129.07	775.38

4
Estimators' Statistical Properties and Uncertainty

This chapter is devoted to the statistical characterization of the estimators illustrated in Chapter 2. In particular, Sections 4.1 and 4.2 provide an overview of the asymptotic properties of PEM and subspace algorithms: consistency, statistical efficiency and the asymptotic distribution of the estimated parameters will be investigated. Section 4.3 exploits the Bayesian perspective to derive the statistical properties of the estimators outlined in Section 2.4.

Characterizing the distribution of the estimates allows to define the so-called *confidence intervals*, i.e. random sets built around the estimate which should contain the true system with high probability. As such, confidence sets provide a measure of the reliability of the returned estimates. The contribution of this chapter is an experimental comparison between the confidence intervals returned by PEM and by non-parametric Bayesian methods. Due to the different nature of these two identification approaches, the comparison appears quite delicate: a significant difficulty is represented by the fact that the returned estimates live in different spaces which are related by a non-linear map. The comparative study outlined in Section 4.4 exploits sampling methods to define so-called "particle" confidence sets for both PEM and Bayesian estimates.

4.1 Statistical Properties of Prediction Error Estimates

The large diffusion of Prediction Error Methods into the system identification community is due to a large extent to its relationship with Maximum Likelihood estimation (pointed out in Section 2.2.3). In fact, this connection allows the direct extension of MLE properties to the PE estimates. The theory on MLE is mainly based on asymptotic arguments, which hold when the number of available data N tends to infinity ($N \to \infty$). Starting from the late Nineties, new interest arose in the study of finite sample properties ($N < \infty$) of system identification estimates. Some of the existing results will be briefly discussed in Section 4.1.2, while the following section will be focused on the classical asymptotic theory for PEM estimates.

Before proceeding, recall that the PE estimate is defined as (2.14)

$$\hat{\theta}_N = \arg\min_{\theta \in D_\theta} \; V_N(\theta, \mathcal{D}^N)$$

with the criterion function $V_N(\theta, \mathcal{D}^N)$ chosen according to the quadratic loss of equation (2.30) or to the general loss (2.31).

In the remainder of the section, it is assumed that the given data $\mathcal{D}^N$ are generated according to

$$\mathcal{S}: \qquad y(t) = G_0(q)u(t) + H_0(q)e_0(t), \qquad \mathbb{E}[e_0(t)] = 0_p,\ \mathbb{E}[e_0(t)e_0^\top(s)] = \Sigma_0\delta_{t,s} \quad (4.1)$$

4.1.1 Asymptotic Properties of PEM Estimates

The following (weak) assumptions are here considered.

A1: The data $\{u(t)\}$ and $\{y(t)\}$ are stationary processes.

A2: The input signal is persistently exciting.

A3: The Hessian $V_N''(\theta)$ is non-singular at least locally around the minimum points of $V_N(\theta)$.

A4: The filters $G(q,\theta)$ and $H(q,\theta)$ are differentiable functions of the parameter vector θ.

When mentioned, the following additional assumption is required:

A5: The set

$$D_T(\mathcal{S}, M) = \{\theta \in D_\theta |\ G_0(q) \equiv G(q,\theta),\ H_0(q) \equiv H(q,\theta),\ \Sigma_0 = \Sigma(\theta)\} \qquad (4.2)$$

consists only of the point θ_0.

4.1.1.1 Consistency

The estimate $\hat{\theta}_N$ is *consistent* if

$$\lim_{N\to\infty} \hat{\theta}_N = \theta_0 \qquad \text{w.p. } 1 \qquad (4.3)$$

Before proving the consistency of $\hat{\theta}_N$, its limiting value is derived under assumptions **A1-A4**. The true system is not required to belong to the chosen model class M, meaning that the set $D_T(\mathcal{S}, M)$ may be empty. On the other hand, the proof for consistency requires $D_T(\mathcal{S}, M)$ to be nonempty.

Quadratic Loss. The quadratic criterion function of equation (2.30) is first considered:

$$V_N(\theta, \mathcal{D}^N) = f_V(R_N(\theta, \mathcal{D}^N)), \qquad R_N(\theta, \mathcal{D}^N) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t,\theta)\varepsilon^\top(t,\theta)$$

The ergodicity theory for stationary signals guarantees that (Hannan, 2009):

$$\lim_{N\to\infty} R_N(\theta, \mathcal{D}^N) = \overline{\mathbb{E}}[\varepsilon(t,\theta)\varepsilon^\top(t,\theta)] =: R_\infty(\theta) \tag{4.4}$$

where the notation adopted by Ljung (1999) has been used:

$$\overline{\mathbb{E}}[x(t)] = \lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} \mathbb{E}[x(t)] \tag{4.5}$$

Since $f(\cdot)$ is assumed to be continuous, it follows

$$\lim_{N\to\infty} V_N(\theta, \mathcal{D}^N) = \lim_{N\to\infty} f_V(R_N(\theta, \mathcal{D}^N)) = f_V(R_\infty(\theta)) =: V_\infty(\theta) \qquad \text{w.p. } 1 \tag{4.6}$$

Thanks to the uniform convergence in equation (4.6) (Ljung, 1978), the following convergence holds true:

$$\lim_{N\to\infty} \hat{\theta}_N = \arg\min_{\theta\in D_\theta} V_\infty(\theta) =: D_c \qquad \text{w.p. } 1 \tag{4.7}$$

Assuming that the set $D_T(\mathcal{S}, M)$ in (4.2) is non empty and that the system operates in open loop, it can be proved that $D_c = D_T(\mathcal{S}, M)$, thus obtaining the consistency of the PE estimator. The proof can be found in Ljung (1999) (Theorem 8.3) or Söderström and Stoica (1989) (sec. 7.5).

General Loss. The general loss (2.31)

$$V_N(\theta, \mathcal{D}^N) = \frac{1}{N}\sum_{t=1}^{N} \ell(\theta, \varepsilon(t,\theta)), \qquad \ell : D_\theta \times \mathbb{R}^p \to \mathbb{R}$$

is here considered. The uniform convergence

$$\lim_{N\to\infty} V_N(\theta, \mathcal{D}^N) = \overline{\mathbb{E}}\left[\ell(\theta, \varepsilon(t,\theta))\right] =: V_\infty(\theta) \qquad \text{w.p. } 1 \tag{4.8}$$

holds true also in this case (Ljung, 1978), leading to

$$\lim_{N\to\infty} \hat{\theta}_N = \arg\min_{\theta\in D_\theta} V_\infty(\theta) =: D_c \qquad \text{w.p. } 1 \tag{4.9}$$

Hence, the PE estimate converges to the best possible approximation of the system which is available within the chosen model set M.

Consider now the case in which ℓ is independent of θ, i.e. $\ell(\theta, \varepsilon(t,\theta)) = \ell(\varepsilon(t,\theta))$. Assume that $D_T(\mathcal{S}, M)$ is nonempty and that the system operates in open loop. If $\ell''(\varepsilon) \in \mathbb{R}^{p\times p}$

is positive definite and the condition

$$\mathbb{E}[\ell'(e_0(t))] = 0 \tag{4.10}$$

is satisfied, then $D_c = D_T(\mathcal{S}, M)$. The interested reader is referred to Ljung (1999) (Theorem 8.5) for the proof.

Adopting the terminology of Söderström and Stoica (1989) (Sec. 6.4), it follows for both losses that the system is *system identifiable* under the assumptions **A1-A4**; if also **A5** holds, then the system is *parameter identifiable.*

4.1.1.2 Asymptotic Distribution of the Parameter Estimates

Assumption **A5** will be used in this section. The derivation of the asymptotic distribution of the estimates is based on the Taylor series expansion of $V_N'(\hat{\theta}_N)^\top$ around θ_0:

$$0 = V_N'(\hat{\theta}_N, \mathcal{D}^N)^\top \approx V_N'(\theta_0, \mathcal{D}^N)^\top + V_N''(\theta_0, \mathcal{D}^N)(\hat{\theta}_N - \theta_0) \tag{4.11}$$

$$\approx V_N'(\theta_0, \mathcal{D}^N)^\top + V_\infty''(\theta_0)(\hat{\theta}_N - \theta_0) \tag{4.12}$$

where V_N' and V_N'' respectively denote the gradient and the Hessian of V_N w.r.t. θ (analogously for V_∞). The approximation (4.12) arises from the convergence

$$\lim_{N\to\infty} V_N''(\theta_0, \mathcal{D}^N) = V_\infty''(\theta_0) \qquad \text{w.p. } 1$$

Provided that the matrix $V_\infty''(\theta_0)$ is invertible (as is the case if **A5** holds), for large N it is possible to write

$$\sqrt{N}(\hat{\theta}_N - \theta_0) \approx -[V_\infty''(\theta_0)]^{-1}[\sqrt{N} V_N'(\theta_0, \mathcal{D}^N)^\top] \tag{4.13}$$

While matrix $V_\infty''(\theta_0)$ is deterministic, the second term is a sum of dependent random variables with zero mean values; exploiting the fact that the dependence between distant terms in the sum decreases, the central limit theorem can be applied to obtain

$$\sqrt{N}(\hat{\theta}_N - \theta_0) \overset{\text{dist}}{\longrightarrow} \mathcal{N}(0_d, P_\theta) \tag{4.14}$$

$$P_\theta = \left\{V_\infty''(\theta_0)\right\}^{-1} P_0 \left\{V_\infty''(\theta_0)\right\}^{-1} \tag{4.15}$$

$$P_0 = \lim_{N\to\infty} N\, \mathbb{E}\left[V_N'(\theta_0, \mathcal{D}^N)^\top V_N'(\theta_0, \mathcal{D}^N)\right] \tag{4.16}$$

A rigorous proof of the previous statement can be found in Ljung (1999) (Theorem 9.1). The explicit expressions of the matrix P_θ for the quadratic loss defined in (2.30) and for the general loss (2.31) will be reported in the following.

Quadratic Loss. If the data are generated from a single-output system (i.e. $p = 1$), the asymptotic covariance matrix is given by

$$P_\theta = \sigma_0 \; \overline{\mathbb{E}}[\psi(t,\theta_0)\psi^\top(t,\theta_0)]^{-1} \tag{4.17}$$

where $\psi(t,\theta) \in \mathbb{R}^{d_\theta}$ is defined as

$$\psi(t,\theta) = -\left(\frac{\partial}{\partial\theta}\,\varepsilon(t,\theta)\right)^\top = \left(\frac{\partial}{\partial\theta}\,\hat{y}(t|\theta))\right)^\top \tag{4.18}$$

A complete derivation of expression (4.17) is provided in Ljung (1999) (Sec. 9.3) and Söderström and Stoica (1989) (Sec. 7.5). The presence of the gradient $\psi(t,\theta)$ of $\hat{y}(t|\theta)$ in formula (4.17) highlights how the asymptotic accuracy of a certain parameter depends on the sensitivity of the prediction $\hat{y}(t|\theta)$ w.r.t. that parameter.
In presence of a finite data sample $\mathcal{D}^N$, formula (4.17) can be approximated as

$$\widehat{P}_N = \widehat{\sigma}_N \left(\frac{1}{N}\sum_{t=1}^{N}\psi(t,\hat{\theta}_N)\psi^\top(t,\hat{\theta}_N)\right)^{-1} \tag{4.19}$$

$$\widehat{\sigma}_N = \frac{1}{N-1}\sum_{t=1}^{N}\varepsilon^2(t,\hat{\theta}_N) \tag{4.20}$$

The expression for the asymptotic variance P_θ when a multi-output system is considered ($p > 1$) is given by

$$P_\theta = \overline{\mathbb{E}}[\psi(t,\theta_0)F_V\psi^\top(t,\theta_0)]^{-1}\overline{\mathbb{E}}[\psi(t,\theta_0)F_V\Sigma_0F_V\psi^\top(t,\theta_0)]\overline{\mathbb{E}}[\psi(t,\theta_0)F_V\psi^\top(t,\theta_0)]^{-1} \tag{4.21}$$

where $\psi(t,\theta_0) \in \mathbb{R}^{d_\theta \times p}$ and $F_V \in \mathbb{R}^{p\times p}$, with its ij-th element defined as

$$[F_V]\,ij = \left.\frac{\partial f_V(Q)}{\partial Q_{ij}}\right|_{Q=\Sigma_0} \tag{4.22}$$

The complete computations which lead to formula (4.21) can be found in Söderström and Stoica (1989) (Appendix A7.1), where a lower bound for P_θ is also derived:

$$P_\theta \geq \overline{\mathbb{E}}[\psi(t,\theta_0)\Sigma_0^{-1}\psi^\top(t,\theta_0)]^{-1} \tag{4.23}$$

If $F_V = \Sigma_0^{-1}$, the equality is achieved, meaning that optimal accuracy is obtained. In particular, the equality condition holds when $f_V(Q) = \det Q$, i.e. when the PE estimate is equivalent to the MLE (if Gaussian innovations are present).

General Loss. The general criterion (2.31) is now considered in the single output case ($p = 1$). The explicit dependence on θ and t is here neglected, i.e. $\ell(t, \theta, \varepsilon) = \ell(\varepsilon)$. Assuming that

$$\mathbb{E}\left[\ell'(\varepsilon(t, \theta_0))\right] = 0, \qquad e_0(t) = \varepsilon(t, \theta_0) \tag{4.24}$$

it follows that

$$P_\theta = \kappa(\ell)\overline{\mathbb{E}}\left[\psi(t, \theta_0)\psi^\top(t, \theta_0)\right]^{-1} \tag{4.25}$$

$$\kappa(\ell) = \frac{\overline{\mathbb{E}}[\ell'(e_0(t))^2]}{\overline{\mathbb{E}}[\ell''(e_0(t))]^2} \tag{4.26}$$

where ℓ' and ℓ'' denote the first and the second derivatives of ℓ w.r.t. its argument, while $\psi(t, \theta_0)$ was defined in (4.18).

In the multi-variable case ($p > 1$) and under assumption (4.24), the expression for the asymptotic covariance becomes (Ljung, 1999)

$$P_\theta = \overline{\mathbb{E}}[\psi(t, \theta_0)\ \Xi\ \psi^\top(t, \theta_0)]^{-1}\ \overline{\mathbb{E}}[\psi(t, \theta_0)\ \Omega\ \psi^\top(t, \theta_0)]\overline{\mathbb{E}}[\psi(t, \theta_0)\ \Xi\ \psi^\top(t, \theta_0)]^{-1} \tag{4.27}$$

where $\Xi \in \mathbb{R}^{p \times p}$ and $\Omega \in \mathbb{R}^{p \times p}$ are defined as

$$\Xi = \overline{\mathbb{E}}[\ell''(e_0(t))] \tag{4.28}$$

$$\Omega = \overline{\mathbb{E}}\left[\left(\ell'(e_0(t))\right)^\top \ell'(e_0(t))\right] \tag{4.29}$$

4.1.1.3 Statistical Efficiency

An estimator is statistically efficient if its covariance matrix equals the so-called *Cramer-Rao lower bound*, which in turn is equal to the inverse of the *Fisher information matrix* for unbiased estimators. Specifically, consider the framework of Section 2.2.3 and let $p_y(y^N; \theta)$ denote the likelihood function for the data y^N (given u^N). Considering a single output system ($p = 1$), the Fisher information matrix is defined as

$$\mathbb{I}_N := \mathbb{E}\left[\left(\frac{\partial p_y(y^N; \theta)}{\partial \theta}\right)^\top \frac{\partial p_y(y^N; \theta)}{\partial \theta}\right] \tag{4.30}$$

$$= \frac{1}{\kappa_0} \sum_{t=1}^{N} \mathbb{E}\left[\psi(t,\theta_0)\psi^\top(t,\theta_0)\right] \tag{4.31}$$

where

$$\kappa_0 := \kappa(-\log p_e) \tag{4.32}$$

with $\kappa(\ell)$ as defined in (4.26). The complete computation of $\mathbb{I}_N$ can be found in Ljung (1999) (Sec. 7.4).

Thanks to the consistency of the PE estimate, the Cramer-Rao lower bound formula for unbiased estimators asymptotically holds for PE estimate; hence, it is possible to write

$$\operatorname{Cov}\left(\sqrt{N}(\hat{\theta}_N - \theta_0)\right) \geq \kappa(-\log p_e) \left(\sum_{t=1}^{N} \mathbb{E}\left[\psi(t,\theta_0)\psi^\top(t,\theta_0)\right]\right)^{-1} \tag{4.33}$$

It follows that the asymptotic covariance matrix P_θ in (4.25) equals the limit (as $N \to \infty$) of the Cramer-Rao bound if $\ell(\cdot) = \log p_e(\cdot)$. Therefore, through this choice of $\ell(\cdot)$ the PE estimate becomes asymptotically statistically efficient and equivalent to the MLE (as shown in Section (2.2.3)).

In particular, in presence of normally distributed disturbances with $p = 1$, the quadratic loss (2.30) satisfies the stated condition on ℓ; in the multi-variable case ($p > 1$), the PE estimate $\hat{\theta}_N$ is asymptotically statistically efficient if the function $f_V(\cdot)$ in the criterion (2.30) is selected so that $F_V = \Sigma_0^{-1}$.

4.1.1.4 Misspecification

Most of the previous results are derived assuming that the set $D_T(\mathcal{S}, M)$ is nonempty, i.e. that the true system $\mathcal{S}$ could be exactly described by at least a model within the chosen model class M. If this condition is not satisfied, e.g. if the true system is more complex than the models contained in M, then

$$\lim_{N\to\infty} \hat{\theta}_N = \theta^* := \arg\min_{\theta\in D_\theta} V_\infty(\theta) \tag{4.34}$$

that is, the PE estimate converge to a minimum point of the asymptotic loss function $V_\infty(\theta)$. Furthermore,

$$\sqrt{N}(\hat{\theta}_N - \theta^*) \overset{\text{dist}}{\longrightarrow} \mathcal{N}(0_p, P_\theta) \tag{4.35}$$

with

$$P_\theta = \left\{V''_\infty(\theta_0)\right\}^{-1} \left\{\lim_{N\to\infty} N\, \mathbb{E}\left[V'_N(\theta^*, \mathcal{D}^N)^\top V'_N(\theta^*, \mathcal{D}^N)\right]\right\} \left\{V''_\infty(\theta_0)\right\}^{-1} \tag{4.36}$$

The estimation of P_θ in case of undermodelling has been considered in Hjalmarsson and Ljung (1992).

4.1.1.5 Confidence Intervals

The derivation of the asymptotic distributions of the parameters estimate allows to define confidence intervals, which provide a measure of the estimate's uncertainty. Indeed, under the frequentist perspective, confidence sets are intervals (built using the given data), which include the true system with high probability if the estimation is repeated with new data. The probability of this event is determined by the so-called *confidence level* α.
In Section 4.1.1.2 it has been shown that

$$\sqrt{N}(\hat{\theta}_N - \theta_0) \overset{\text{dist}}{\longrightarrow} \mathcal{N}(0_p, P_\theta) \tag{4.37}$$

Therefore, it follows that

$$\zeta_N = N\ (\hat{\theta}_N - \theta_0)^\top P_\theta^{-1}(\hat{\theta}_N - \theta_0) \overset{\text{dist}}{\longrightarrow} \chi^2(d) \tag{4.38}$$

where $\chi^2(d)$ denotes the χ^2-distribution with d degrees of freedom. Indeed, if a random vector $\zeta \in \mathbb{R}^d$ is normally distributed, $\zeta \sim \mathcal{N}(0, P)$, then $\zeta^\top P^{-1}\zeta \sim \chi^2(d)$.
Let $\chi^2_d(\cdot)$ denote the quantile function of the χ^2-distribution with d degrees of freedom: $\chi^2_d(p)$ is equal to the value x for which $\Pr(\chi^2(d) \leq x) = p$.
If a confidence level α is fixed, the set

$$\mathcal{E}_\alpha^{PEM} = \left\{\theta \in \mathbb{R}^{d_\theta} \middle| N(\hat{\theta}_N - \theta)^\top P_\theta^{-1}(\hat{\theta}_N - \theta) \leq \chi^2_{d_\theta}(\alpha)\right\} \tag{4.39}$$

asymptotically (as $N \to \infty$) constitutes an ellipsoidal region in $\mathbb{R}^{d_\theta}$ and centred in $\hat{\theta}_N$.
As such, $\mathcal{E}_\alpha^{PEM}$ is the asymptotic α-level confidence set for the PE estimator $\hat{\theta}_N$.
In practice, since only a finite number N of data points is given, the above-illustrated properties are only approximatively valid. It should also be recalled that the asymptotic theory assumes that the chosen model class M is rich enough to contain the true system $\mathcal{S}$; in practical applications, the situation is very different, since the model class has to be selected using the finite data sample $\mathcal{D}^N$ (as widely discussed in Sections 2.2 and 2.5). Because of this requirement, PEM belongs to the class of *Post Model Selection Estimators (PMSE)*, which have been analysed by a number of authors. In particular, it has been shown that the finite-sample distribution of such estimators significantly differs from the results postulated by the asymptotic theory (Leeb and Potscher, 2005; Leeb and Pötscher, 2006). Nevertheless, the confidence set (4.39) is commonly adopted to assess

the reliability of the PE estimator by replacing the asymptotic covariance P_θ with its finite-sample counterpart $\widehat{P}_N$ (see e.g. (4.19)). The quality of such approximation has been studied e.g. by Garatti, Campi, and Bittanti (2004), who prove that the asymptotic theory is reliable even in presence of a high level of uncertainty in the estimated model (due e.g. to poor informative data), if an extra condition holds true for the chosen model class. They also provide a classification of model classes satisfying such condition.
A more detailed discussion on the so-called finite-sample properties of PEM estimates is postponed to the next section.

4.1.2 Finite-Sample Properties of PEM Estimates

The study of finite-sample properties of PEM estimates aims at assessing how many data points are needed to guarantee with probability $1-\epsilon$ that

$$\sup_\theta |V_N(\theta, \mathcal{D}^N) - V(\theta)| \leq C \tag{4.40}$$

for some value $C > 0$. As usual, $V_N(\theta)$ is the empirical quadratic criterion minimized by PEM (see equation (2.30) with $f_V(\cdot) = \mathrm{Tr}[\cdot]$), while $V(\theta)$ denotes its expected value,

$$V(\theta) := \mathbb{E}\left[\mathrm{Tr}[\varepsilon(t,\theta)\varepsilon^\top(t,\theta)]\right] \tag{4.41}$$

This problem has been studied by Weyer, Williamson, and Mareels (1999), who exploit risk minimization theory to derive uniform probabilistic bounds as (4.40) for FIR and ARX model classes. Their derivation does not assume that the true system belongs to the chosen model class and that the noise sequence is uniformly bounded. They also show that the number of samples N required to satisfy the derived bound is quadratic in the model order of FIR and ARX models. A similar study is due to Weyer (2000), whose derivation assume that the observed data are M-dependent and β-mixing. An extension of these results to general linear model structures has been provided by Campi and Weyer (2002), who resort to exponential inequalities for stochastic processes. They derive a bound for the difference

$$V(\hat{\theta}_N) - \frac{1}{N}\sum_{t=1}^{N} V(\bar{\theta}_N) \tag{4.42}$$

with $V(\theta)$ as defined in (4.41) and $\bar{\theta}_N$ given by

$$\bar{\theta}_N = \arg\min_{\theta \in D_\theta} \frac{1}{N}\sum_{t=1}^{N} V(\theta) \tag{4.43}$$

The bound depends on the model and the system order, on pole locations and on the noise variance.
Along this research line, the contribution of Vidyasagar and Karandikar (2006) should also be mentioned, where concepts of statistical learning theory (devoted to finite-sample estimates) are imported into the system identification field.

The bounds derived by the afore-mentioned contributions depend on the number of available data but not on the actually observed data. To overcome the conservatism that could arise from this property, the application of data-based methods has been investigated to assess the quality of the estimated models. These approaches essentially rely on bootstrap and subsampling techniques. Relevant contributions on this topic are due e.g. to Tjarnstrom and Ljung (2002) and Dunstan and Bitmead (2003). A criticism w.r.t. these data-based methods is the lack of rigorous finite-sample results.

The previous section has described how the asymptotic distribution of the PEM estimates is exploited to derive their uncertainty regions; however, it has been observed that the asymptotic theory may lead to a misleading quantification of the uncertainty, especially in presence of small datasets. During the first decade of the 2000s some authors have developed non-asymptotic confidence regions for PEM estimates. In particular, Campi and Weyer (2005) propose the approach called Leave-One-Out Sign-Dominant Correlation Regions (LSCR). Under minimal assumptions on the noise sequence affecting the given data, LSCR returns data-based confidence sets which contain the true parameter values θ_0 with an exact probability. Once empirical correlation functions are computed, LSCR requires to identify the regions in the parameter space where these functions assume positive or negative values too many times. These zones are not included in the confidence regions returned by LSCR. The intuition behind this procedure is the following: when evaluated for the true parameter value θ_0, the correlation functions are sums of zero mean random variables; hence, it is likely that they assume both negative and positive values an "equal" amount of times. The zones in which this event is verified (according to the empirical correlation functions) belong to the confidence region returned by LSCR. As expected, the shape and the size of this region are influenced by the noise level affecting the given data.
An overview of LSCR is given by Campi and Weyer (2006a), where its extension to the handling of non-linear systems is also presented. Later works from the same authors extend the application of LSCR to the case in which the true system $\mathcal{S}$ does not belong to the fixed model set M (Campi and Weyer, 2006b; Campi, Ko, and Weyer, 2009). Campi and Weyer (2010) relax the assumptions on the noise sequence: while the original LSCR requires it to be zero-mean and symmetrically distributed, the procedure proposed by

Campi and Weyer (2010) works with any noise sequence.
The confidence zones returned by LSCR hold with exact probability only for scalar parameters, while only probability bound can be guaranteed for the multidimensional case. In addition, the MLE is not guaranteed to belong to the returned regions. A more recent work (Csáji, Campi, and Weyer, 2015) introduces a new method, called Sign-Perturbed Sums (SPS), which overcomes these drawbacks in the case of Least-Squares estimates. Like LSCR, SPS assumes that the noise sequence affecting the given observations has zero-mean and symmetric distribution; further knowledge on its distribution is not needed. The confidence regions built by SPS contain the true parameter with exact (user-defined) probability; furthermore, they are star-convex with the LS estimate as a star center. The authors also describe an efficient computation of an ellipsoidal outer approximation of the confidence sets returned by SPS.
Den Dekker, Bombois, and Van den Hof (2008) focus on OE models and they use a test statistic based on a Fisher score to derive exact finite-sample confidence regions.

4.2 Statistical Properties of Subspace Estimates

Compared to Section 4.1, the discussion about the statistical properties of subspace estimates will skip several technical details, since only an overview of the main results will be here provided. The reason for it lies in the comparative study conducted in Section 4.4, which will take into account only PE and Bayesian estimates.

The statistical analysis of subspace methods appears more complicated than the one conducted for PEM, because no cost function is explicitly minimized. Basically, subspace algorithms consist of two Least-Squares stages, intermediated by the computation of an SVD. A complete understanding of the statistical properties of subspace estimates is still missing, even if some asymptotic characterization has been derived and has been summarized in the survey Bauer (2005). What complicates the asymptotic analysis of subspace estimates is the SVD stage, since the decomposition may not be unique. The results reported in Bauer (2005) highly rely on the asymptotic behaviour of compact self-adjoint operators (see Chatelin (1983), Prop 3.26 and Bauer (2005), Theorem 1).

From a statistical point of view, the subspace estimators are distinguished according to the way in which the system matrices are computed in the second stage of the algorithm described in Section 2.3.2. Specifically, subspace estimates computed through the so-called *shift-invariance approach* and those returned by the *state approach* enjoy different statistical properties. The algorithms proposed by Verhaegen (1993b, 1994) belong to the first class of above-mentioned methods, while the routine of Larimore (1983) falls into the second category. The two algorithms introduced by Van Overschee and De Moor (1994) for the estimations of system matrices can be considered as variants of the *state approach*. In the discussion which follows, such estimators will be referred to as *N4SID*.

An overview of the results regarding consistency and asymptotic distribution of the estimated system matrices is now provided; since technical details are omitted, several references to the original results are inserted.

4.2.1 Consistency

Assuming that the true system is described by the matrices (A_0, B_0, C_0, D_0), an estimator $(\widehat{A}_N, \widehat{B}_N, \widehat{C}_N, \widehat{D}_N)$ is *consistent* if there exists a deterministic matrix T (not depending on the number of data N) such that $\widehat{A}_N - TA_0T^{-1}$, $\widehat{B}_N - TB_0$, $\widehat{C}_N - C_0T^{-1}$, $\widehat{D}_N - D_0$ converge to zero in probability (as $N \to \infty$).

As for PE estimators, consistency is analysed assuming that the chosen system order is the true one.

All the variants of subspace algorithms have been proved to be consistent. Preliminary results on the consistency of the algorithms based on the *shift-invariance approach* are given in Verhaegen (1994) and Jansson and Wahlberg (1998); consistency for finite values of r and s (respectively, the future and past horizons) is proved by Bauer and Jansson (2000).

When no inputs are observed or if the measured inputs are white noise, Peternell, Scherrer, and Deistler (1996) derive the consistency of the *state approach*, letting the past horizon s tend to infinity ($s \to \infty$). Consistency of *N4SID* algorithms for a finite value of s arises from the results in Chiuso and Picci (2004b).

4.2.2 Misspecification

Misspecification arises when the selected system order differs from the true one. Two different situations may emerge: on the one hand, if the chosen complexity is larger than the true one, consistency is guaranteed for both *shift-invariance* and *state approaches*; one the other hand, if the chosen system order is smaller than the true one, both the estimators will be affected by some bias. When no inputs are observed, expressions for the asymptotic bias (due to under-modelling) exist for the *state approach* (Bauer (1998), Ch. 2 and Bauer, Deistler, and Scherrer (1998)). Despite some of them include results on the dependence of the bias distribution over frequency on the choice of the weighting matrices (see Section 2.3.3), their practical usefulness is limited. Analogously, the derived formulas for the asymptotic bias in the case of observed inputs do not appear so useful in practice.

On the other hand, the under-modelling bias affecting the estimates computed through the *shift-invariance approach* has not been suitably investigated in the literature.

4.2.3 Asymptotic Distribution of the Parameters Estimate

All the estimators returned by the different subspace algorithms are known to be asymptotically normally distributed. Despite several expressions for the asymptotic covariances have been derived, few results exist on the impact of the user's choices discussed in Section 2.3.3 on such covariances. As a consequence, guidelines on how to fix the user-defined parameters in order to achieve optimal asymptotic accuracy are mostly missing.

Asymptotic normality for the *shift-invariance approach* is stated by Bauer and Jansson (2000) and Jansson (2000). Jansson (1997) proves that the asymptotic distribution of the estimated system poles does not depend on W_1. Jansson (2000) derives explicit expressions for the asymptotic covariance of the estimated system matrices, proving their dependence on the horizons r and s and on the weighting matrix W_2. In particular, the formulas introduced by Jansson (2000) also show that the asymptotic variance of the estimates $\widehat{A}_N$ and $\widehat{C}_N$ does not depend on W_1. As above-mentioned, a clear understanding of the impact of the user's choices for s, r and W_2 is still missing. When r and s are fixed, the commonly used values for W_1 and W_2 seem to be suboptimal, since they do not lead to estimators achieving the Cramer-Rao bound.

The analysis for the *state approach* is almost complete for the case of no observed inputs or white noise inputs, while the understanding in the general case of coloured inputs is only partial, thus resembling the situation for the *shift-invariance approach*.

The asymptotic normality in the general case of coloured inputs is established by Bauer (1998), who also derives expressions for the corresponding covariance. However, such formulas do not provide significant insights on possible optimal choices of the user's parameters. Regarding *N4SID* (specifically Algorithm 1 in Van Overschee and De Moor (1994)), a later work of Chiuso and Picci (2004b) shows that under the assumptions of consistency, the *N4SID* estimators with finite r and s are asymptotically normal and provides the covariance of $\widehat{A}_N$ and $\widehat{C}_N$. Such results are obtained without assuming infinite persistence of the inputs. However, the use of such expressions for the comparison of different weighting matrices and different values of r appears difficult, due to the complexity of the derived formulas.

Passing to the case of no observed inputs (or white noise inputs), the analysis is almost complete. Asymptotic normality of the estimators is established by Bauer, Deistler, and

Scherrer (1999), while the independence of the asymptotic covariance on W_2 is stated by Bauer and Jansson (2000). Furthermore, Bauer and Ljung (2002) derive variance expressions which explicitly depend on the weighting W_1 and on the future horizon r. These have been exploited to infer optimal choices for the weighting matrices, as will be illustrated in Section 4.2.4.

4.2.4 Statistical Efficiency

Statistical efficiency has been proved only for the estimates returned by the *state approach* in the case of no observed inputs or white noise inputs. In such situation, by means of the simplified covariance expressions derived by Bauer and Ljung (2002) it has been shown that the *CVA* weightings of Larimore (1990) (see Section 2.3.3) are optimal for each fixed horizon r. Furthermore, the asymptotic accuracy of the corresponding estimates increases monotonically with r. Therefore, in presence of Gaussian innovations, if the system order is correctly selected and if $r \to \infty$ (at a rate which can be estimated from the given data), the Cramer-Rao lower bound is attained. The CVA subspace algorithm thus achieves optimal accuracy within the class of (asymptotically) unbiased estimators. In presence of non-Gaussian innovations, it can be proved that PEM and the CVA algorithm are asymptotically equivalent.

The statistical analysis of subspace algorithms reported in this section appears complete for the estimates returned by the so-called *state approach* in the case of no observed inputs (or white inputs): as above-stated, asymptotic optimality is guaranteed by the *CVA* weightings of Larimore (1990). On the other hand, the results regarding the *shift-invariance approach* and the *state approach* in presence of coloured inputs are partial: optimal choices of the weighting matrices and of the horizons r and s w.r.t. asymptotic accuracy have not been derived yet.

4.3 Statistical Properties of Non-Parametric Bayesian Estimates

Section 2.4 has shown how the non-parametric Bayesian methods can be equivalently treated as regularized techniques in Reproducing Kernel Hilbert Spaces (RKHS). This connection makes possible to study the properties of the returned estimators using different perspectives. On the one hand, the Bayesian interpretation allows to straightforwardly derive the finite-sample distribution of the obtained estimates, which can be exploited for the definition of confidence intervals; on the other hand, the asymptotic behaviour of non-parametric regularized regression has been investigated by several contributions in the statistical learning literature.

Section 4.3.1 provides a brief overview of the results concerning consistency in the context of GPR and regularized LS algorithms in RKHS. The discussion will be rather general, since the system identification literature has provided few results concerning the statistical properties of non-parametric regression approaches. A brief comment regarding misspecification is given in Section 4.3.2. Finally, finite-sample confidence sets for the impulse response estimate obtained through non-parametric Bayesian methods are derived in Section 4.3.3 under a Bayesian perspective.

4.3.1 Consistency

When dealing with non-parametric regression, the notion of consistency relies on the so-called *expected risk*, whose definition is now provided.

Definition 4.3.1 (Expected Risk)**.** Let $\mathcal{X}$ and $\mathcal{Y}$ respectively denote the input and output spaces $\mathcal{X}$ and $\mathcal{Y}$ on which the probability distribution $\mu(x, y)$ acts. If $\mathcal{Y}$ is a Hilbert space, given a function $f : \mathcal{X} \to \mathcal{Y}$, the ability of f to describe the distribution μ is measured by its *expected risk*

$$\mathcal{R}(f) = \int_{\mathcal{X}\times\mathcal{Y}} \|y - f(x)\|_2^2 \, d\mu(x, y) \tag{4.44}$$

The minimizer of the risk over the space of measurable functions on $\mathcal{X}$ taking value on $\mathcal{Y}$ is the so-called *regression function*, $f_\mu^*(x) = \mathbb{E}[y|x]$.

The definition of consistent learning algorithm can now be stated.

Definition 4.3.2 (Consistent Learning Algorithm)**.** A procedure which takes the training data $\mathcal{D}^N = \{(x_i, y_i)\}_{i=1}^N$ drawn i.i.d. from $\mu(x, y)$ and returns a function $\hat{f}_{\mathcal{D}^N}$ is *consistent* for the measure $\mu(x, y)$ if

$$\lim_{N\to\infty} \mathcal{R}(\hat{f}_{\mathcal{D}^N}) = \mathcal{R}(f_\mu^*) \qquad \text{w.p. } 1 \tag{4.45}$$

If $\hat{f}_{\mathcal{D}^N}$ is consistent for all Borel probability measures $\mu(x, y)$, then it is said to be *universally consistent.*

It follows that the asymptotic performance of a certain learning algorithm are typically evaluated in terms of the rate of convergence of its estimate to the regression function.

It should be observed that the general setting here presented differs from the standard setup considered in system identification, where the given data are not assumed to be i.i.d. from an unobserved distribution.

Assuming that the regression function is contained in the hypothesis space $\mathcal{H}$ within which the function $\hat{f}_{\mathcal{D}^N}$ lies, universal consistency for regularized LS algorithms has been proved. For such type of learning algorithms, several contributions in the literature of statistical learning theory have provided convergence rates to the regression function (Engl, Kunisch, and Neubauer, 1989; Smale and Zhou, 2007; Wu, Ying, and Zhou, 2006). Such rates are shown to depend on the capacity of the hypothesis space $\mathcal{H}$, measured in terms of metric entropy (or, equivalently, covering numbers) or Gaussian complexity. Optimal rates for the regression of vector-valued functions have been given by Caponnetto and De Vito (2007): the derived properties are then exploited to define a criterion for the choice of the regularization parameter as a function of the number of samples. Yuan et al. (2010) provide results on optimal convergence rates for functional linear regression using RKHS: such contribution fits into the framework illustrated in Sections 2.4.1.2 and 2.4.2.2, where the unknown function is observed through a linear functional.

Passing to the framework of Gaussian Process Regression, Choi and Schervish (2004) prove its consistency in the case of a one-dimensional input space $\mathcal{X}$ and under certain assumptions including smoothness of the mean and covariance function of the Gaussian Process. Furthermore, the measurement noise is required to have a normal or Laplacian distribution. Rates of convergence (or contraction) to the true posterior distribution have been more recently investigated by van der Vaart and van Zanten (2008).

Finally, in the context of system identification, consistency of the regularized LS estimator obtained through marginal likelihood maximization has been proved by Aravkin et al. (2014), assuming a kernel equal to a scaling of the identity, $\bar{K}_\eta = \eta I_{mpT}$. Under such assumption, desirable asymptotic properties in terms of the MSE are derived. Specifically, it is shown that the marginal likelihood estimate of η converges to the minimizer of the MSE in the case of white noise inputs; in the general case of coloured input convergence to a minimizer of a weighted MSE (with weights depending on N) is proved.

4.3.2 Misspecification

Considering the general framework introduced in Section 4.3.1, misspecification arises when the regression function f_μ^* does not belong to the chosen hypothesis space $\mathcal{H}$. In this case, besides the *estimation error*, an *approximation error* arises (see also the discussion in Section 2.5, where such error was referred to as *model bias*). The asymptotic behaviour of the approximation error is typically analyzed by means of the so-called *oracle inequalities* (Cavalier, Golubev, Picard, Tsybakov, et al., 2002). However, such situation appears less understood w.r.t. the case in which f_μ^* is assumed to belong to $\mathcal{H}$; relevant contributions have been provided by Steinwart, Hush, Scovel, et al. (2009) and Mendelson, Neeman, et al. (2010).

Moving the analysis to the system identification field, it should be recalled that Pillonetto and De Nicolao (2010) have proved that realizations from a zero-mean Gaussian process with stable-spline covariance (illustrated in Section 3.3.1) are almost surely the impulse response of a BIBO system. This in turn guarantees that the hypothesis space $\mathcal{H}$ induced by the so-called stable-spline kernels is rich enough to contain the impulse response of any BIBO stable LTI system. Consequently, under a suitable kernel choice, misspecification does not arise when regularized regression approaches are used in linear system identification. Furthermore, Pillonetto and Chiuso (2015) prove that the estimator computed through evidence maximization is robust even when undermodelling is present. Similar conclusions are drawn by Aravkin et al. (2014) in the context of Penalized Automatic Relevance Determination (PARD).

4.3.3 Confidence Intervals

The Bayesian interpretation of the system identification methods illustrated in Section 2.4 provides the user with finite-sample distributions of the computed estimators, expressed in terms of the derived posterior. Specifically, when the Empirical Bayes paradigm is followed, the posterior is a Gaussian distribution with mean and covariance given in equations (2.128) and (2.129), when the impulse response is treated as an infinite-dimensional object, or in (2.148) and (2.149), when a FIR model is estimated. On the other hand, when the Full Bayes approach is adopted, a sampled approximation of the posterior is derived: its mean and covariance could be inferred e.g. recurring to percentiles. In both cases, the estimated posterior can be used to define confidence intervals around the estimator, as will be detailed in the following discussion. In the Bayesian framework, such sets are typically called *credible intervals* (Jaynes and Kempthorne, 1976; Efron, 2005).

Empirical Bayes and Full Bayes will be separately treated in the following.

4.3.3.1 Empirical Bayes

When the impulse response is treated as an infinite-dimensional object, the α-level confidence interval for $\hat{g}(t)$ is readily derived from equations (2.128) and (2.129) as

$$\mathcal{C}_\alpha(t) = \left[\hat{g}(t) - z_\alpha \sqrt{P_{g_t}^{post}},\ \hat{g}(t) + z_\alpha \sqrt{P_{g_t}^{post}}\right] \tag{4.46}$$

where $P_{g_t}^{post}$ is the posterior covariance defined in equation (2.129), while z_p denotes the quantile function of the standard normal distribution. Exploiting the equivalence between the Bayesian estimations and regularized regression within RKHS, Wahba (1983, 1990) propose to build the confidence interval (4.46) around the function estimates computed using the latter technique. The properties of these confidence intervals are investigated by Nychka (1988), who introduces the so-called *Average Coverage Property (ACP)* for the α-level confidence intervals $\{\mathcal{C}_\alpha(t)\}_{t=1}^N$, built around the input locations:

$$ACP = \frac{1}{N}\sum_{t=1}^{N} \Pr[g_0(t) \in \mathcal{C}_\alpha(t)] \tag{4.47}$$

where $g_0(\cdot)$ denotes the true system impulse response. Nychka (1988) proves that the Bayesian confidence intervals (4.46) enjoy the ACP property, i.e. the ACP computed for them is close to the nominal level α. As an alternative to Bayesian confidence intervals, other authors (Härdle and Bowman, 1988; Hardle and Marron, 1991; Wahba, 1990) consider uncertainty sets derived from bootstrap procedures. Wang and Wahba (1995) compared these two approaches when Gaussian data are given, showing that they both enjoy the ACP property.

Considering the regularized LS framework treated in Sections 2.4.1.3 and 2.4.2.3, an α-level ellipsoidal confidence set lying in $\mathbb{R}^{pmT}$ is readily derived as

$$\mathcal{E}_\alpha^{EB} = \left\{ \mathbf{g} \in \mathbb{R}^{pmT} | (\mathbf{g} - \hat{\mathbf{g}})^\top \left(P_{\mathbf{g}}^{post}\right)^{-1} (\mathbf{g} - \hat{\mathbf{g}}) \le \chi^2_{pmT}(\alpha) \right\} \tag{4.48}$$

where $P_{\mathbf{g}}^{post}$ is the posterior covariance matrix defined in equation (2.149) and $\chi^2_d(\cdot)$ denotes the quantile function of the χ^2-distribution with d degrees of freedom.

4.3.3.2 Full Bayes

For simplicity, only the case in which the impulse response is treated as a finite dimensional vector is here considered, i.e. $\mathbf{g} \in \mathbb{R}^{pmT}$ (see Sections 2.4.1.3 and 2.4.2.3).
Recall the sampled approximated posterior in equation (2.185), here reported for convenience:

$$p_{\mathbf{g}}(\mathbf{g}|Y_N) = \int_{D_\eta} p_{\mathbf{g}}(\mathbf{g}|Y_N, \eta) p_\eta(\eta|Y_N) d\eta \approx \frac{1}{N_{sp}} \sum_{i=1}^{N_{sp}} p_{\mathbf{g}}(\mathbf{g}|Y_N, \eta^{(i)}) \tag{4.49}$$

An α-level confidence set around the estimated impulse response (e.g. the one reported in (2.186)) can be defined as

$$\mathcal{S}_\alpha^{FB} = \left\{ \mathbf{g}^{(i)} \in \mathbb{R}^{pmT} : \frac{1}{N_{sp}} \sum_{j=1}^{N_{sp}} p(\mathbf{g}^{(i)}|Y_N, \eta^{(j)}) \geq p_\alpha^{FB} \right\}, \tag{4.50}$$

where p_α^{FB} is the $(1-\alpha)$-percentile of the set

$$\left\{ \frac{1}{N_{sp}} \sum_{j=1}^{N} p(\mathbf{g}^{(i)}|Y_N, \eta^{(j)}),\ i = 1, ..., N_{sp} \right\}$$

That is, $\mathcal{S}_\alpha^{FB}$ contains the impulse response samples $\mathbf{g}^{(i)}$ associated with the α-fraction of the highest values of the approximated posterior (2.185).

Differently from the confidence sets previously defined for the estimators derived from the Empirical Bayes approach or from PEM, $\mathcal{S}_\alpha^{FB}$ is not a dense set, but rather a "particle" set, since it consists of sampled points.

4.4 PEM and Non-Parametric Bayesian Methods: a Comparison of the Estimators' Uncertainty

Previous sections have shown how the system identification algorithms described in Chapter 2 lead to the definition of different confidence sets around the returned estimator. The difference not only lies in the space in which such sets are defined, but also in their nature: while PEM, subspace methods and Bayesian procedures equipped with the Empirical Bayes paradigm give rise to dense confidence sets, Full Bayes approaches relies on sampling algorithms, thus building so-called "particle" sets. The contribution of this section is the introduction of a framework in which the listed confidence sets can be compared.

The comparative study will regard PEM estimators and Bayesian techniques (estimating a finite length impulse response), while subspace algorithms will not be taken into account; furthermore, the focus will be on SISO systems (i.e. $p = m = 1$).

To attempt a fair comparison, the confidence sets returned by the considered estimators are all translated into the impulse response space and converted into "particle" sets. The following discussion will detail how this is accomplished.

Remark 4.4.1. The reader could argue that the decision of performing the comparison in the impulse response space would favour the Bayesian approaches, whose estimators already lies in this space. However, the author considers this a fair choice, since the impulse response explicitly describes the input-output relation of the system to be identified. Furthermore, if the comparison had be done in the parameter space, this would have required a model reduction step on the Bayesian estimates: according to the author's opinion, this step is more delicate than the non-linear transformation that has to be applied on the parametric estimates in order to pass from the parameter space to the impulse response one.

4.4.1 PEM

The analysis here conducted regards the PE estimate $\hat{\theta}_N$ computed using the quadratic loss (2.30) with $f_V(x) = x$, that is

$$V_N(\theta, \mathcal{D}^N) = \frac{1}{N} \sum_{t=1}^{N} \varepsilon^2(t, \theta) \tag{4.51}$$

The asymptotic confidence set defined in Section 4.1.1.5 is here considered with the asymptotic covariance P_θ replaced by its finite-sample counterpart $\widehat{P}_N$ (4.19), namely

$$\widehat{\mathcal{E}}_\alpha^{PEM} = \left\{ \theta \in \mathbb{R}^{d_\theta} \middle| N(\hat{\theta}_N - \theta)^\top \widehat{P}_N^{-1} (\hat{\theta}_N - \theta) \leq \chi^2_{d_\theta}(\alpha) \right\} \tag{4.52}$$

Such set is converted into a "particle" set in the impulse response space by first drawing N_{sp} samples from the asymptotic distribution $\mathcal{N}(\hat{\theta}_N, \widehat{P}_N/N)$ and retaining only the ones which fall into the set $\widehat{\mathcal{E}}_\alpha^{PEM}$; the "particle" set is then defined by converting these parameter samples into the corresponding impulse responses through a suitable mapping $\mathfrak{M} : \mathbb{R}^{d_\theta} \to \mathbb{R}^T$. Formally, the derived "particle" set is defined as

$$\mathcal{S}_\alpha^{PEM+ASYMP} = \left\{ \mathbf{g}_{\theta^{(i)}} = \mathfrak{M}(\theta^{(i)}), \ \mathbf{g}_{\theta^{(i)}} \in \mathbb{R}^T | \theta^{(i)} \in \widehat{\mathcal{E}}_\alpha^{PEM}; \ i = 1, ..., N_{sp} \right\} \tag{4.53}$$

Section 4.1.1.5 has pointed out how the confidence sets derived from the asymptotic parameter distribution may provide misleading information in presence of few or poorly informative data. The subsequent Section 4.1.2 has discussed alternative definitions of confidence sets for PEM estimates, which hold exactly for datasets with finite size. It should be recalled that the comparative study described in this chapter considers confidence sets which are built by means of suitable sampling techniques. In line with this approach, a "non-asymptotic" confidence set for PEM estimates is here defined through an appropriate sampling of the likelihood function $p_y(y^N | \widehat{\Sigma}; \theta)$, with $\widehat{\Sigma}$ being a noise variance estimate (obtained e.g. through a Least-Squares model). In fact, assuming a flat prior distribution $p(\theta)$ for the parameters, the likelihood function is proportional to the posterior PDF:

$$p_\theta(\theta|y^N, \widehat{\Sigma}) \propto p_y(y^N|\widehat{\Sigma};\theta) = (2\pi\widehat{\Sigma})^{-N/2} \exp\left\{-\frac{N}{2\widehat{\Sigma}} V_N(\theta, \mathcal{D}^N)\right\} \tag{4.54}$$

An MCMC algorithm is designed to obtain N_{sp} samples $\theta^{(i)}$ from (4.54). From these the corresponding impulse responses $\mathbf{g}_{\theta^{(i)}} = \mathfrak{M}(\theta^{(i)})$ are computed and the set

$$\mathcal{S}_\alpha^{PEM+LIK} = \left\{\mathbf{g}_{\theta^{(i)}} : p_\theta(\theta^{(i)}|y^N, \widehat{\Sigma}) \geq p_\alpha^{PEM+LIK}, \theta^{(i)} \in D_\theta;\ i = 1, ..., N_{sp}\right\} \tag{4.55}$$

is defined, where $p_\alpha^{PEM+LIK}$ is the $(1-\alpha)$-percentile of the set $\{p_\theta(\theta^{(i)}|y^N, \widehat{\Sigma});\ i = 1, ..., N_{sp}\}$.

The set is denoted with $PEM + LIK$ in order to emphasize its strict connection with the likelihood function. Some readers could recognize in the definition of $\mathcal{S}_\alpha^{PEM+LIK}$ some analogies with the construction of confidence sets through bootstrap procedures. What mainly distinguishes $\mathcal{S}_\alpha^{PEM+LIK}$ from bootstrap confidence sets is its implementation. Specifically, parametric bootstrap methods build several datasets starting from a low-bias system estimate; from each of these datasets a new estimate is computed, which is later used to define a "particle" confidence set. Hence, roughly speaking, while bootstrap approaches sample datasets and then adopt search routines to compute an estimate, the procedure here proposed adopts an MCMC algorithm to directly sample parameter estimates. Furthermore, the construction of $\mathcal{S}_\alpha^{PEM+LIK}$ is based on an approximation of the parameters posterior distribution, thus resembling the derivation of the Bayesian confidence sets discussed in the following.

Remark 4.4.2. As observed in Section 4.1.2, sampling techniques allow to avoid approximations of asymptotic expressions. However, they are still approximations of the true uncertainty associated to the estimated parameter $\hat{\theta}_N$. Indeed, the definition of these confidence sets still relies on the assumption that the model class M and the model complexity are fixed, even if in practice model selection is performed using the available data. That is, $\hat{\theta}_{PEM}$ is a so-called post-model-selection estimator (PMSE): in order to define a more accurate confidence set, the uncertainty related to the model selection step should be taken into account. However, as reported in Section 4.1.1.5, Leeb and Potscher (2005) observe that the finite-sample distribution of a PMSE has generally a quite intricate shape.

Remark 4.4.3. The comparative study here conducted does not consider the finite-sample confidence regions returned by the LSCR method mentioned in Section 4.1.2 (Campi and Weyer, 2006a). The reason for this choice lies in the difficulty of assessing the shape and the size of the corresponding confidence sets when multidimensional parameters are estimated.

4.4.2 Empirical Bayes

Section 4.3.3 has shown how the confidence sets derived when resorting to the Empirical Bayes paradigm are ellipsoids centred in the minimum variance estimate $\hat{\mathbf{g}}$ and with shape defined by the posterior covariance $P_{\mathbf{g}}^{post}$. To adapt such sets to the proposed comparative setting, $\mathcal{E}_\alpha^{EB}$ is approximated by a point distribution obtained by drawing N_{sp} samples from the posterior $p(\mathbf{g}|Y_N, \hat{\eta}_{EB})$ and retaining only those belonging to (4.48), that is:

$$\mathcal{S}_\alpha^{EB} = \left\{ \mathbf{g}^{(i)} \in \mathbb{R}^T : \mathbf{g}^{(i)} \in \mathcal{E}_\alpha^{EB};\ i = 1, ..., N_{sp} \right\} \tag{4.56}$$

Here $\hat{\eta}_{EB}$ denotes the hyper-parameters estimate obtained in equation (2.183) through evidence maximization.

4.4.3 Full Bayes

The confidence set $\mathcal{S}_\alpha^{FB}$ defined in equation (4.50) for Bayesian estimators arising from the Full Bayes approach already belongs to the proposed comparative setting. Therefore, its quality will be numerically compared with that of the previously defined "particle" sets.

4.5 Numerical Results

The quality of the "particle" confidence sets derived in Section 4.4 is here evaluated through a Monte-Carlo study, composed of 200 experiments.

4.5.1 Data

The Monte-Carlo study here conducted exploits the datasets D2 and D4, which have been introduced and used in the paper Chen et al. (2014)). Both of them consist of 200 30th order random SISO dicrete-time systems having all the poles inside a circle of radius 0.95. The output data are affected by white Gaussian noise whose variance is equal to that of the noise-free output (i.e. the SNR on the output signal is equal to 1). What distinguishes the two data-banks is the input signal with which the systems are fed; namely:

D2: the input is unit variance white Gaussian noise;

D4: the input is a band-limited random Gaussian signal generated with the MATLAB routine `idinput`; its normalized band is set to $[0, 0.8]$.

The reader is referred to Chen et al. (2014) for further details on these datasets. Three different data lengths are here considered: $N_1 = 250$, $N_2 = 500$, $N_3 = 2500$.
In addition, the data bank S1D2 introduced in Chen et al. (2012) has been experimented. The obtained results are similar to the ones achieved on datasets D2 and D4 and outlined in the following; therefore, these will not be reported here.

4.5.2 Identification Algorithms

PEM: In the performed simulations, PEM is implemented through the MATLAB routine `oe`. Model selection is performed through the BIC criterion (2.218), since it generally outperforms AIC. This estimator will be denoted as PEM+BIC.
Moreover, as a reference an oracle estimator is also considered and denoted by PEM+OR. This has the (unrealistic) knowledge of the impulse response of the true system, $\{g(k)\}_{k=1}^{\infty}$: among the OE models with complexity ranging from 2 to 30, it selects the one giving the best fit to $\{g(k)\}_{k=1}^{\infty}$, according to the criterion (4.57).

EB, FB: The Bayesian methods here evaluated are implemented adopting a zero-mean Gaussian prior with a covariance matrix given by the DC kernel in equation (3.28) (Chen et al., 2012). The length T of the estimated impulse responses is set to 100, that is $\hat{\mathbf{g}} \in \mathbb{R}^{100}$.
The estimator computed using the Empirical Bayes approach will be referred to as EB; analogously, FB will denote the Bayesian estimate computed according to the Full Bayes paradigm. Such estimator is determined using an Adaptive Metropolis Hastings (AM) algorithm (Haario, Saksman, and Tamminen, 2001), i.e. an MCMC algorithm, whose proposal distribution is changed at each iteration according to the samples drawn at the previous steps.

For ease of notation, the apex (or the subscript) X will be used to denote a generic estimator among the ones previously illustrated, that is, PEM+BIC, PEM+OR, EB and FB.

4.5.3 Impulse Response Estimates

As a first comparison, the ability of the considered identification techniques on the reconstruction of the true impulse response is evaluated. For each estimated system and

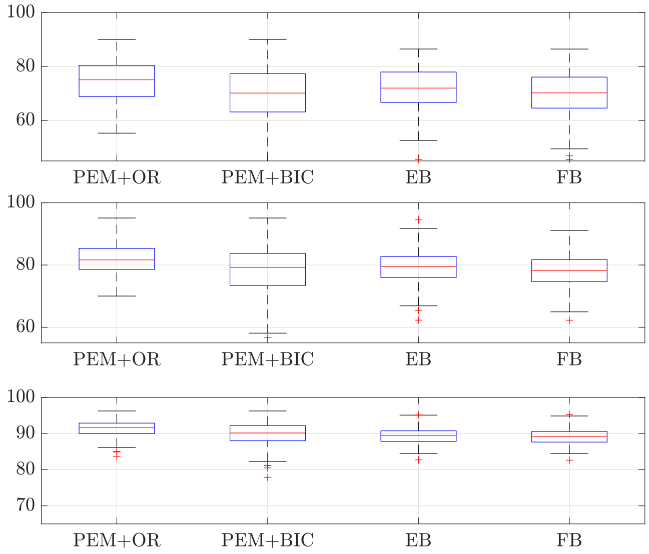


Figure 4.1: Dataset D2 - Impulse response fit (4.57) achieved by the identification algorithms listed in Section 4.5.2. Different data lengths are evaluated: $N_1 = 250$ (*Top*), $N_2 = 500$ (*Center*) and $N_3 = 2500$ (*Bottom*).

	PEM+OR	PEM+BIC	EB	FB
Average Fit ($N_1 = 250$)	71.97	67.52	71.39	70.49
Average Fit ($N_2 = 500$)	80.58	77.25	79.08	78.43
Average Fit ($N_3 = 2500$)	90.43	88.88	89.41	89.24

Table 4.1: Dataset D2 - Average impulse response fit (4.57) achieved by the identification algorithms listed in Section 4.5.2. Different data lengths are evaluated.

for each estimator X the so-called *impulse response fit* is computed:

$$\mathcal{F}_T(\hat{\mathbf{g}}_X) = 100 \cdot \left(1 - \frac{\|\mathbf{g}_0 - \hat{\mathbf{g}}_X\|_2}{\|\mathbf{g}_0 - \bar{\mathbf{g}}_0\|_2}\right), \qquad \bar{\mathbf{g}}_0 = \frac{1}{T}\sum_{k=1}^{T} [\mathbf{g}_0]_k \tag{4.57}$$

where $\mathbf{g}_0,\ \hat{\mathbf{g}} \in \mathbb{R}^T$ collect the first T true and estimated impulse response coefficients. Figure 4.1 and Table 4.1 display the boxplots and the average value of index (4.57) achieved by the four estimators on dataset D2. Figure 4.2 and Table report the results obtained on D4. The different data lengths are considered.

The four identification algorithms perform very similarly on the two datasets; the only exception is the behaviour of PEM+BIC, which leads to poor performance on D4. This is most likely due to the low pass characteristics of the input signal, which makes the order estimation step particularly delicate. Indeed, in D2 (and in S1D2), where the inputs are

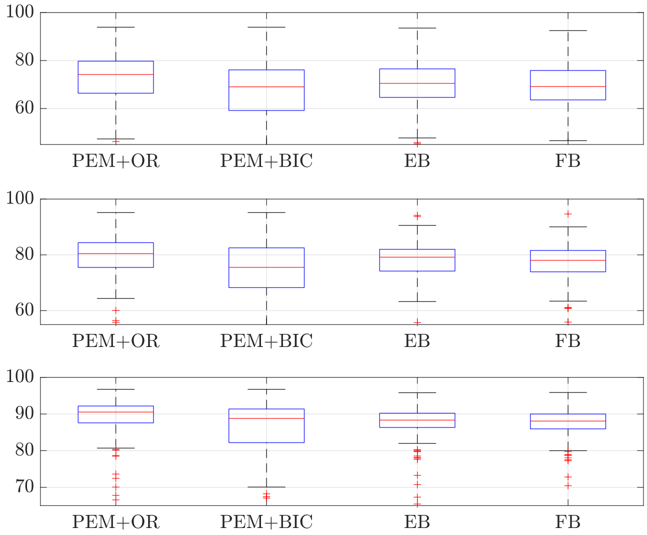


Figure 4.2: Dataset D4 - Impulse response fit (4.57) achieved by the identification algorithms listed in Section 4.5.2. Different data lengths are evaluated: $N_1 = 250$ (*Top*), $N_2 = 500$ (*Center*) and $N_3 = 2500$ (*Bottom*).

	PEM+OR	PEM+BIC	EB	FB
Average Fit ($N_1 = 250$)	71.43	56.30	69.93	68.26
Average Fit ($N_2 = 500$)	78.33	67.11	77.56	76.79
Average Fit ($N_3 = 2500$)	88.84	74.84	87.06	85.94

Table 4.2: Dataset D4 - Average impulse response fit (4.57) achieved by the identification algorithms listed in Section 4.5.2. Different data lengths are evaluated.

Gaussian white noises, PEM+BIC performs similarly to the Bayesian estimators.
The oracle estimator PEM+OR sets an upper bound on the achievable performance by parametric methods; compared to PEM+OR, EB performs remarkably well, with only a slightly inferior fit. The FB estimator performs similarly to EB, but it requires the implementation of an MCMC, which is highly computationally expensive. These results suggest that the marginal posterior $p_\eta(\eta|Y_N)$ is sufficiently well peaked to be approximated by a delta function (meaning that $p_{\mathbf{g}}(\mathbf{g}|Y_N) \simeq p_{\mathbf{g}}(\mathbf{g}|Y_N, \hat{\eta}_{EB})$).

4.5.4 Returned Confidence Sets

Section 4.4 has introduced two types of "particle" confidence sets for PEM estimators: $\mathcal{S}_\alpha^{PEM+ASYMP}$ in equation (4.53) and $\mathcal{S}_\alpha^{PEM+LIK}$ in (4.55). In the following, the first will be referred to as *asymptotic* confidence sets, while the denomination *likelihood sampling* will be used for the latter. For Bayesian estimators, $\mathcal{S}_\alpha^{EB}$ in (4.56) and $\mathcal{S}_\alpha^{FB}$ in (4.50) have been defined. As before, $\mathcal{S}_\alpha^X$ will generically denote one of them.
In the performed simulations, $\alpha = 0.95$, while the number N_{sp} of samples that are used to construct the aforementioned confidence sets takes different values for each of the considered Monte-Carlo runs. Specifically, it is set as the maximum chain length of the three implemented MCMC algorithms (i.e. those used for *likelihood sampling* for the two PEM estimators and the AM used to compute the Full Bayes estimator). For each of these routines, the chain length and the burn-in N_{bi} are set by applying twice the method proposed in Raftery and Lewis (1992).
Since the considered confidence sets are only approximations of a "true" α-level confidence set, the aim is to study how well they perform both in term of "coverage" (how often does the α-level confidence set contain the "true" value?) as well as of size (how big is an α-level confidence set?). Unfortunately, since the treated confidence sets simply consist of a set of points, it is not possible to define a notion of inclusion (does the true system belong to the set?). Hence, as a proxy to this, an index measuring the relative distance from the true system and the closest point within the confidence set is considered.
The evaluated indexes are listed below.

1. *Coverage Index*: For a fixed probability level α, it is given by

$$\mathcal{I}_1^X(\alpha) := \min_{x \in \mathcal{S}_\alpha^X} \frac{\|x - \mathbf{g}_0\|_2}{\|\mathbf{g}_0\|_2} \tag{4.58}$$

where $\mathbf{g}_0 \in \mathbb{R}^T$ denotes the true impulse response. For future analysis the concept of "coverage" will be meant as in definition (4.58).

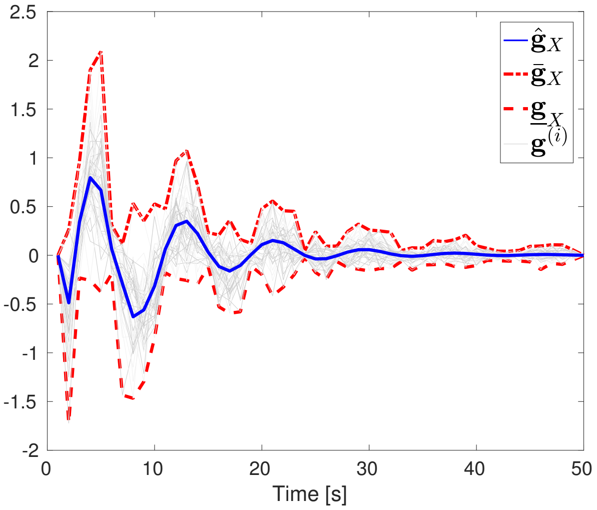


Figure 4.3: Illustration of the *Confidence set size* $\mathcal{I}_2^X(\alpha)$ index for a single system. The blue line denotes the estimated impulse response $\hat{\mathbf{g}}_X$; gray lines represent the impulse responses sampled from the confidence set $\mathcal{S}_\alpha^X$; dashed red line denotes $\underline{\mathbf{g}}_X$, while dashed-dotted line represents $\bar{\mathbf{g}}_X$. $\mathcal{I}_2^X(\alpha)$ is equal to the area between the two red lines.

2. *Confidence Set Size*: It evaluates the area of the interval which includes the whole slot of impulse responses contained in $\mathcal{S}_\alpha^X$. Specifically, define the vectors $\bar{\mathbf{g}}_X \in \mathbb{R}^T$ and $\underline{\mathbf{g}}_X \in \mathbb{R}^T$ whose j-entries are $[\bar{\mathbf{g}}_X]_j := \max_i[\mathbf{g}^{(i)}]_j$ and $[\underline{\mathbf{g}}_X]_j := \min_i[\mathbf{g}^{(i)}]_j$, respectively, with $\mathbf{g}^{(i)} \in \mathcal{S}_\alpha^X$. The evaluated index is defined as:

$$\mathcal{I}_2^X(\alpha) = \sum_{j=1}^{T} [\bar{\mathbf{g}}_X]_j - [\underline{\mathbf{g}}_X]_j \tag{4.59}$$

Referring to Figure 4.3, a large confidence set is more likely to contain the true impulse response, giving a low value of $\mathcal{I}_1^X(\alpha)$, but it will also denote a high uncertainty in the returned estimate, thus leading to a large value of $\mathcal{I}_2^X(\alpha)$.

Figures 4.4 and 4.5 illustrate the boxplots of index (4.58) when the compared identification algorithms are applied on data D2 and D4, respectively. As before, three sample sizes are considered. Again, the results observed in the two datasets are very similar. The Bayesian confidence sets have higher coverage performances then the parametric ones equipped with BIC. The unique exception is the *asymptotic* PEM+BIC confidence set when the data length is $N_3 = 2500$, that is, when the asymptotic theory is more reliable. Their accuracy is comparable with that achieved by the *likelihood sampling* PEM+OR confidence set, which is favoured by the knowledge of the true system. No substantial differences are detected between the two Bayesian approaches here compared.

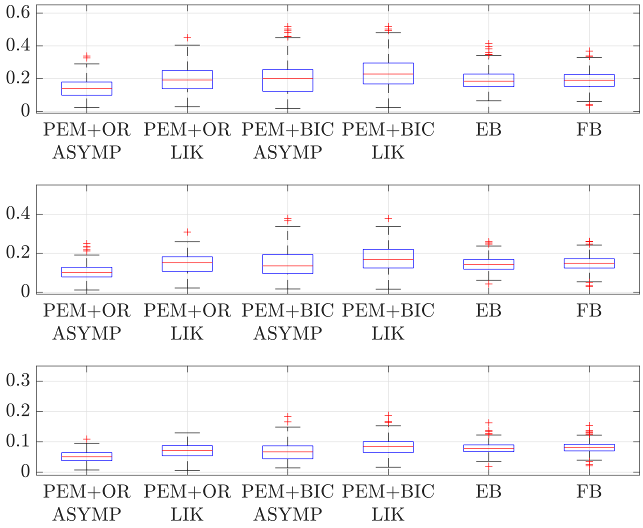


Figure 4.4: Dataset D2 - *Coverage Index* $\mathcal{I}_1^X(\alpha)$ (4.58) returned by the identification algorithms listed in Section 4.5.2. Different data lengths are evaluated: $N_1 = 250$ (*Top*), $N_2 = 500$ (*Center*) and $N_3 = 2500$ (*Bottom*).

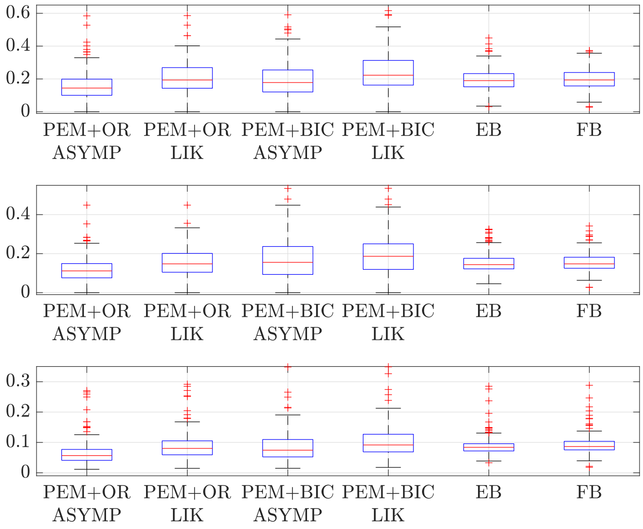


Figure 4.5: Dataset D4 - *Coverage Index* $\mathcal{I}_1^X(\alpha)$ (4.58) returned by the identification algorithms listed in Section 4.5.2. Different data lengths are evaluated: $N_1 = 250$ (*Top*), $N_2 = 500$ (*Center*) and $N_3 = 2500$ (*Bottom*).

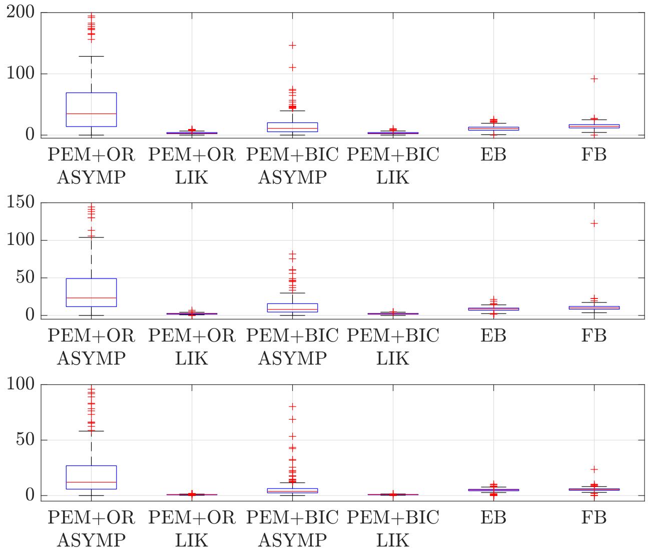


Figure 4.6: Dataset D2 - *Confidence Set Size* $\mathcal{I}_2^X(\alpha)$ (4.58) returned by the identification algorithms listed in Section 4.5.2. Different data lengths are evaluated: $N_1 = 250$ (*Top*), $N_2 = 500$ (*Center*) and $N_3 = 2500$ (*Bottom*).

Among the parametric confidence sets, as expected, PEM+OR outperforms PEM+BIC, whereas surprisingly, the *asymptotic* confidence sets outperform those built through *likelihood sampling*, which are constructed precisely for finite data lengths. This result can be explained analysing also index (4.59) displayed in Figures 4.6 and 4.7; the discussion is therefore postponed. Note that the *asymptotic* confidence sets show, correctly, a significant improvement for larger data lengths.

Figures 4.6 and 4.7 illustrate the boxplots of index (4.59) when the considered identification algorithms are respectively applied on datasets D2 and D4. No significant differences can be detected between the results achieved in the two datasets.
The EB confidence sets have a slightly smaller size than the FB ones: this follows from the fact that FB also accounts for the uncertainty related to the hyper-parameters estimation. The parametric approaches equipped with *likelihood sampling* return the smallest confidence sets, even smaller than the Bayesian ones. However, the coverage index in Figures 4.4 and 4.5 shows that they are less accurate than the Bayesian one.
Furthermore, notice that the two PEM+OR confidence sets are larger than those returned by the PEM+BIC estimator: this can be explained by the fact that PEM+OR tends to select higher-order models, thus bringing more uncertainty into the estimated systems.
Comparing the *asymptotic* and the *likelihood sampling* confidence sets it is clear that the latter is more precise than the former. Indeed, the *asymptotic* confidence set is an

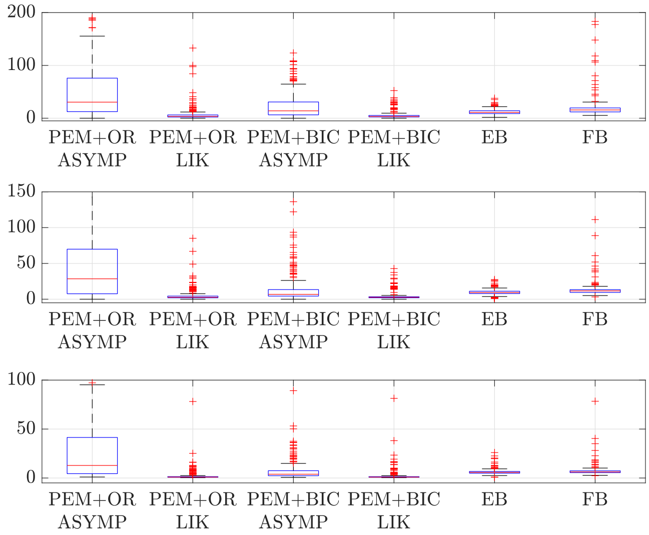


Figure 4.7: *Confidence Set Size* $\mathcal{I}_2^X(\alpha)$ (4.58) returned by the identification algorithms listed in Section 4.5.2. Different data lengths are evaluated: $N_1 = 250$ (*Top*), $N_2 = 500$ (*Center*) and $N_3 = 2500$ (*Bottom*).

approximation which holds for large datasets, while the *likelihood sampling* is correct for any finite sample size; however, this improvement comes at a rather high computational price needed to run the MCMC sampler. This explain why the *asymptotic* confidence sets outperform the *likelihood* ones in the metric (4.58): being much larger they have higher coverage performances. Analysing the size and coverage properties of the likelihood confidence sets they seems to be too much self confident, giving a small uncertainty to their estimate but with unsatisfactory performances in terms of coverage.

It is important to note that the asymptotic theory does not take into account stability issues: namely, the confidence set derived from the Gaussian asymptotic distribution (4.14) could contain unstable impulse responses. Therefore the sampling procedure described in Section 4.4.1 could yield to diverging confidence set size. In order to avoid this problem the asymptotic Gaussian distribution has been truncated within the stability region. Clearly, this fact shows an intrinsic problem of the asymptotic theory.

By comparing the results in Figures 4.4-4.6 and 4.5-4.7 the following conclusions could be drawn: among the feasible identification methods, EB and FB are preferable taking into account performances both in terms of coverage and size; in addition, according to the performed numerical tests, there seems to be no gain in using the more computationally expensive FB.

Remark 4.5.1. The reader could argue that the sets $\mathcal{S}_\alpha^X$ are only "sample" approximations of a confidence set, while one may be interested in having a bounded region as a confidence set. In the case of the EB estimator this region is directly defined since the posterior distribution is Gaussian, thus naturally leading to the ellipsoidal confidence set (4.48). For all the other estimators, it is in principle possible to build outer approximations of the confidence sets e.g. by building a minimum size set which includes all the points in $\mathcal{S}_\alpha^X$; examples are the convex hull or an ellipsoid. The convex hull can be computed with off-the-shelf algorithms (such as the MATLAB routine `convhulln.m`), while the smallest ellipsoid (in terms of sum of squared semi-axes length) can be found solving the following problem:

$$\begin{aligned} P_\alpha^{opt}, c_\alpha^{opt} := \quad & \arg \quad \min_{P,c} \operatorname{Tr}\, P \\ & s.t. \quad \begin{bmatrix} P & (\mathbf{g}^{(i)} - c) \\ (\mathbf{g}^{(i)} - c)^\top & 1 \end{bmatrix} \succ 0, \\ & \mathbf{g}^{(i)} \quad \in \mathcal{S}_\alpha^X \end{aligned} \tag{4.60}$$

See Calafiore (2002) for further details. The corresponding ellipsoid is given by

$$\mathcal{E}_\alpha^{opt} = \left\{ x \in \mathbb{R}^T : (x - c_\alpha^{opt})^\top (P_\alpha^{opt})^{-1} (x - c_\alpha^{opt}) \leq 1 \right\} \tag{4.61}$$

However, the computation of the convex hull as well as the resolution of the optimization problem (4.60) become computationally intractable for moderate ambient space and sample sizes. For instance, when the impulse response lives in $\mathbb{R}^T$, $T = 100$ and the set $\mathcal{S}_\alpha^X$ contains thousands of points (as in the situation we are facing), these computations are prohibitive with off-the-shelf methods. To overcome this issue, the optimal ellipsoid $\mathcal{E}_\alpha^{opt}$ has been tentatively approximated by the sample mean $\bar{\mathbf{g}}_\alpha^X$ and the sample covariance $\widehat{P}_\alpha^X$ of the elements in $\mathcal{S}_\alpha^X$; namely:

$$\widehat{\mathcal{E}}_\alpha^X \quad = \quad \left\{ x \in \mathbb{R}^T : (x - \bar{\mathbf{g}}_\alpha^X)^\top \left(\widehat{P}_\alpha^X \right)^{-1} (x - \bar{\mathbf{g}}_\alpha^X) \leq k_\alpha^X \right\} \tag{4.62}$$

where k_α^X is a constant appropriately chosen so that all the elements of $\mathcal{S}_\alpha^X$ fall within $\widehat{\mathcal{E}}_\alpha^X$. However, it can be observed that these ellipsoids are rather rough approximations of the sets $\mathcal{S}_\alpha^X$. Inspecting 2D sections of the T-dimensional ellipsoids, it can be seen that often the axis orientation is not correct, thus leading to sets which are much larger than needed. This fact has been mainly observed for the confidence sets related to PEM estimates.

These observations suggest that the quality of the confidence sets obtained through the ellipsoidal approximation (4.62) would have been highly dependent on the quality of the fitted ellipsoid. Therefore, a comparison among the different estimators, based on this kind of confidence set, would have led to unreliable results. Consequently, such results are not reported.

5
On-line System Identification

The identification routines which have been described so far can be classified as *off-line* or *batch* methods, since all the given data are used simultaneously to find the system estimate. They are opposed to so-called *on-line* or *real-time* algorithms, which update a previous estimate as soon as new data become available. Distinctive traits of these methods are the limited requirements for both memory and computational time. Indeed, most of them do not need to store all the data that have been used until the present instant; furthermore, the computations performed to update the current estimate are modest, since the result should be returned before new data arrive.

On-line algorithms play a central role in adaptive control systems, where the controller is continuously re-designed according to the most recent system estimate. This type of routines also constitute the first step in a fault detection algorithm, where they are used to detect if some system properties have changed. Indeed, they are typically designed to track so-called time-varying systems, i.e. systems whose characteristics may vary with time.

Specifically, this chapter considers the following setup. At time k a certain estimate $\hat{\mathbf{x}}^{(i)}$ is available and has been computed using the data coming from a collection of i previous datasets $\boldsymbol{\mathcal{D}}_i = \bigcup_{l=1}^{i} \mathcal{D}_l^N = \{u(t), y(t)\}_{t=1}^{iN}$; at time $k+N$ new data $\mathcal{D}_{i+1}^N$ become available and a new estimate $\hat{\mathbf{x}}^{(i+1)}$ should be determined by exploiting them. Here, $\mathbf{x}$ could denote e.g., the system impulse response, the polynomials coefficients of a transfer function model, etc.

Real-time identification methods are typically based on recursive routines, which compute the estimate $\hat{\mathbf{x}}^{(i+1)}$ by simple modifications of $\hat{\mathbf{x}}^{(i)}$. Since most of the recursive identification algorithms are developed as approximations of off-line routines, there is always a trade-off between accuracy and computational parsimony to be paid.

Section 5.1 will briefly outline the so-called Recursive Prediction Error Methods, which represent a variation of classical PEM in order to satisfy the on-line requirements aforementioned. Section 5.2 will outline the recursive methods proposed in the context of subspace identification, while Section 5.3 will propose a way to adapt the non-parametric Bayesian methods described in Section 2.4 to the real-time setting here treated. The effectiveness of the approaches introduced in Section 5.3 will be evaluated through numerical experiments, whose results are reported in Section 5.4.

5.1 On-Line Identification with Prediction Error Methods

Recursive Prediction Error Methods (RPEM) represent a generalization of so-called Recursive Least-Squares (RLS) algorithms (see Söderström and Stoica (1989), Sec. 9.2

and Ljung and Söderström (1983)). Indeed, if the one-step ahead predictor of the selected model structure is linear w.r.t. the parameters vector, RPEM reduces to RLS (as happens for the off-line counterparts).

For simplicity, the following illustration of RPEM assumes $N = 1$, meaning that at each time instant i a new input-output data pair arrives. In addition, the following loss function is considered

$$V_i(\theta, \boldsymbol{\mathcal{D}}_i) = \frac{1}{2}\sum_{t=1}^{i} \varepsilon^\top(t,\theta) Q \varepsilon(t,\theta) \tag{5.1}$$

with Q being a positive definite weighting matrix (notice that the loss (5.1) coincides with that in equation (2.30) if $f_V(\cdot) = \mathrm{Tr}[\cdot]$).
Let $\hat{\theta}^{(i-1)}$ be the minimizer of $V_{i-1}(\theta, \boldsymbol{\mathcal{D}}_{i-1})$. Assuming that the minimum point of $V_i(\theta, \boldsymbol{\mathcal{D}}_i)$ is close to $\hat{\theta}^{(i-1)}$, it is possible to write the following second-order Taylor series expansion around $\hat{\theta}^{(i-1)}$:

$$\begin{aligned} V_i(\theta, \boldsymbol{\mathcal{D}}_i) \approx V_i(\hat{\theta}^{(i-1)}, \boldsymbol{\mathcal{D}}_i) &+ V_i'(\hat{\theta}^{(i-1)}, \boldsymbol{\mathcal{D}}_i)(\theta - \hat{\theta}^{(i-1)}) \\ &+ \frac{1}{2}(\theta - \hat{\theta}^{(i-1)})^\top V_i''(\hat{\theta}^{(i-1)}, \boldsymbol{\mathcal{D}}_i)(\theta - \hat{\theta}^{(i-1)}) \end{aligned} \tag{5.2}$$

The new estimate $\hat{\theta}^{(i)}$ can now be found by minimizing (5.2) w.r.t. θ:

$$\hat{\theta}^{(i)} = \hat{\theta}^{(i-1)} - \left[V_i''(\hat{\theta}^{(i-1)}, \boldsymbol{\mathcal{D}}_i)\right]^{-1} V_i'^\top(\hat{\theta}^{(i-1)}, \boldsymbol{\mathcal{D}}_i) \tag{5.3}$$

To make the procedure recursive, even the matrices involved in (5.3) should be recursively updated:

$$V_i(\theta, \boldsymbol{\mathcal{D}}_i) = V_{i-1}(\theta, \boldsymbol{\mathcal{D}}_{i-1}) + \frac{1}{2}\varepsilon^\top(i,\theta) Q \varepsilon(i,\theta) \tag{5.4}$$

$$V_i'(\theta, \boldsymbol{\mathcal{D}}_i) = V_{i-1}'(\theta, \boldsymbol{\mathcal{D}}_{i-1}) + \varepsilon^\top(i,\theta) Q \varepsilon'(i,\theta) \tag{5.5}$$

$$V_i''(\theta, \boldsymbol{\mathcal{D}}_i) = V_{i-1}''(\theta, \boldsymbol{\mathcal{D}}_{i-1}) + [\varepsilon'(i,\theta)]^\top Q \varepsilon'(i,\theta) + \varepsilon^\top(i,\theta) Q \varepsilon''(i,\theta) \tag{5.6}$$

where $\varepsilon^\top(i,\theta) Q \varepsilon''(i,\theta)$ is approximatively written, since $\varepsilon''(i,\theta)$ is a tensor for MIMO systems. Equations (5.5) and (5.6) can be further simplified by assuming

$$V_{i-1}'(\hat{\theta}^{(i-1)}, \boldsymbol{\mathcal{D}}_{i-1}) = 0 \tag{5.7}$$

$$V_{i-1}''(\hat{\theta}^{(i-1)}, \boldsymbol{\mathcal{D}}_{i-1}) = V_{i-2}''(\hat{\theta}^{(i-2)}, \boldsymbol{\mathcal{D}}_{i-1}) \tag{5.8}$$

$$\varepsilon^\top(i,\theta) Q \varepsilon''(i,\theta) \approx 0 \tag{5.9}$$

Approximation (5.7) arises from treating $\hat{\theta}^{(i-1)}$ as the minimizer of $V_{i-1}(\theta, \boldsymbol{\mathcal{D}}_{i-1})$, while

(5.8) assumes that $V''_{i-1}(\theta, \mathcal{D}_{i-1})$ varies slowly with θ. Finally, $\varepsilon^\top(i,\theta)Q\varepsilon''(i,\theta)$ could be neglected in $V''_i(\theta, \mathcal{D}_i)$ observing that $\varepsilon(i,\theta)|_{\theta=\theta_0}$ will be a white process and hence

$$\mathbb{E}[\varepsilon^\top(i,\theta)Q\varepsilon''(i,\theta)] = 0$$

It should be noticed that approximations (5.7)-(5.9) hold exactly for the LS case. By means of (5.7)-(5.9), the parameters update (5.3) can be rewritten as

$$\hat{\theta}^{(i)} = \hat{\theta}^{(i-1)} - \left[V''_i(\hat{\theta}^{(i-1)}, \mathcal{D}_i)\right]^{-1} [\varepsilon'(i, \hat{\theta}_{i-1})]^\top Q\varepsilon(i, \hat{\theta}_{i-1}) \tag{5.10}$$

$$V''_i(\hat{\theta}^{(i-1)}, \mathcal{D}_i) = V''_{i-1}(\hat{\theta}^{(i-2)}, \mathcal{D}_{i-1}) + [\varepsilon'(i, \hat{\theta}^{(i-1)})]^\top Q\varepsilon'(i, \hat{\theta}^{(i-1)}) \tag{5.11}$$

To further improve the recursive nature of the algorithm, the inverse of $V''_i(\hat{\theta}^{(i-1)}, \mathcal{D}_i)$ can be computed through the matrix inversion lemma; in addition $\varepsilon'(i, \hat{\theta}^{(i-1)})$ and $\varepsilon(i, \hat{\theta}^{(i-1)})$ should be approximated by quantities that can be computed on-line. Denote them as

$$\varepsilon^{(i)} \approx \varepsilon(i, \hat{\theta}^{(i-1)}), \qquad \psi^{(i)} \approx -[\varepsilon'(i, \hat{\theta}^{(i-1)})]^\top \tag{5.12}$$

The precise form of such approximations depend on the chosen model class. Introducing the notation $P^{(i)} := \left[V''_i(\hat{\theta}^{(i-1)}, \mathcal{D}_i)\right]^{-1}$, RPEM can finally be stated in its general form:

$$\hat{\theta}^{(i)} = \hat{\theta}^{(i-1)} + K^{(i)}\varepsilon^{(i)} \tag{5.13}$$

$$K^{(i)} = P^{(i)}\psi^{(i)}Q \tag{5.14}$$

$$P^{(i)} = P^{(i-1)} - P^{(i-1)}\psi^{(i)}[Q^{-1} + \left(\psi^{(i)}\right)^\top P^{(i-1)}\psi^{(i)}]^{-1} \left(\psi^{(i)}\right)^\top P^{(i-1)} \tag{5.15}$$

Many algorithms update $K^{(i)}$ through the following more efficient recursion

$$K^{(i)} = P^{(i-1)}\psi^{(i)}[Q^{-1} + \left(\psi^{(i)}\right)^\top P^{(i-1)}\psi^{(i)}]^{-1} \tag{5.16}$$

A faster implementation of the recursive algorithm in equations (5.13)-(5.15) is possible, admitting a change into the search direction in equation (5.15). This modification significantly reduces the computational effort of the algorithm, but at the expense of slowing down the estimates' convergence.

A convergence analysis of these recursive routines can be done by assuming that the true system belongs to the chosen model class. Specifically, it can be shown that RPEM converges globally to the set consisting of the stationary points of

$$V_\infty(\theta) = \overline{E}[\varepsilon^\top(i,\theta)Q\varepsilon(i,\theta)] \tag{5.17}$$

If the true parameter θ_0 is a unique stationary point, then RPEM returns consistent parameters estimates under weak assumptions. Furthermore, the RPEM estimates are asymptotically Gaussian distributed with the same distribution detailed in Section 4.1.1.2 for off-line procedures.

5.1.1 Dealing with Time-Varying Systems

On-line algorithms are typically designed to track the possible time-varying nature of the system to be identified. Two extreme modes of variation are typically conceived: in the first mode the system parameters are subject to sudden changes at isolated time instants, while the latter mode is characterized by slowly-varying parameters at a constant rate in time. In the following, these two variation modes will be respectively referred to as *jumping parameters* and *drifting parameters*.

To equip RPEM with the ability to track the afore-mentioned parameters variations, some modifications to algorithm (5.13)-(5.15) have to be done. Three approaches are commonly adopted and will be here briefly illustrated.

A classical technique applies a rectangular sliding window on the given data. If N_w is the length of the chosen window, only the last N_w data are used to compute the current estimate. To account for abrupt changes in the true parameters values, N_w should be varied with time: however, this solution is rarely applied, since its computational effort is significant.

A second approach modifies the loss function (5.1) in order to exponentially weight the input-output data

$$V_i(\theta, \boldsymbol{\mathcal{D}}_i) = \frac{1}{2} \sum_{t=1}^{i} \gamma^{i-t} \varepsilon^\top(t,\theta) Q \varepsilon(t,\theta) \tag{5.18}$$

where γ, $0 < \gamma \leq 1$, is the so-called *forgetting factor*, typically set very close to 1. Consequently, recent measurements count more than older ones in the estimation criterion. The smaller the value of γ, the faster the information contained in the data is forgotten. To account for the presence of γ, the RPEM algorithm in equations (5.13)-(5.15) is modified as

$$\hat{\theta}^{(i)} = \hat{\theta}^{(i-1)} + K^{(i)} \varepsilon^{(i)} \tag{5.19}$$

$$K^{(i)} = P^{(i-1)} \psi^{(i)} [\gamma Q^{-1} + \left(\psi^{(i)}\right)^\top P^{(i-1)} \psi^{(i)}]^{-1} \tag{5.20}$$

$$P^{(i)} = \frac{1}{\gamma} P^{(i-1)} - \frac{1}{\gamma} P^{(i-1)} \psi^{(i)} [\gamma Q^{-1} + \left(\psi^{(i)}\right)^\top P^{(i-1)} \psi^{(i)}]^{-1} \left(\psi^{(i)}\right)^\top P^{(i-1)} \tag{5.21}$$

In several applications, the forgetting factor is varied with time. For the case of *jumping*

parameters, γ should be equal to 1, when no changes are detected, while γ should temporarily decrease below 1 at the jumping instants. Some authors have considered this way of setting γ as a soft-prewindowing, with the usual prewindowing arising when γ is varied according to a step function. On the other hand, when the parameters slowly vary, there exists an optimum value for γ, which is constant or very slowly varying.
A possible choice lets $\gamma(i)$ tend exponentially to one, according to

$$\gamma(i) = 1 - \gamma_0^i(1 - \gamma(0)) \tag{5.22}$$

Typically, γ_0 is set to 0.99, while $\gamma(0)$ is set to 0.95 (Ljung, 1999).
Several schemes for the on-line update of $\gamma(i)$ have been proposed in the literature of Recursive Least Squares (RLS). Essentially, at the i-th step, the tuning of $\gamma(i)$ is based on the current prediction error:

$$\varepsilon(i, \hat{\theta}^{(i-1)}) = y(i) - \hat{y}(i|\hat{\theta}^{(i-1)}) \tag{5.23}$$

The method introduced by Slock and Kailath (1989) obtains the variable forgetting factor by minimizing the Excess Mean Squared Error (EMSE) which varies proportionally with the inverse of the autocorrelation of the error signal $\{\varepsilon(i, \hat{\theta}^{(i-1)})\}$. Similarly, in Toplis and Pasupathy (1988), $\gamma(i)$ varies in proportion to the inverse of the squared error; the risk of getting a negative forgetting factor is prevented by using a pre-specified threshold. Other methods which tune $\gamma(i)$ according to the squared error are due to Fortescue, Kershenbaum, and Ydstie (1981); Park, Jun, and Kim (1991); Song, Lim, Baek, and Sung (2000). However, it has been shown that such approaches are particularly sensitive to the measurement noise. An average of M previous values of the squared error is exploited by Cho, Kim, and Powers (1991), whose solution updates $\gamma(i)$ according to

$$\gamma(i) = 1 - \frac{Q(i)}{\hat{\sigma} N_{max}}, \qquad Q(i) = \frac{1}{M}\sum_{t=0}^{M-1} \varepsilon^2(i-t, \hat{\theta}^{(i-t)}) \tag{5.24}$$

with N_{max} being the maximum memory length and $\hat{\sigma}$ a noise variance estimate. To simplify the exposition, a scalar output signal is here considered (i.e. $p = 1$).
The approach proposed by Jiang and Cook (1992) directly perturbs the covariance matrix $P^{(i)}$ whenever a change is detected.
A gradient-like update is proposed by Song et al. (2000):

$$\gamma(i) = \gamma(i-1) + \alpha \nabla_\lambda J(i), \qquad J(i) = \frac{1}{2}\mathbb{E}[\varepsilon^2(i, \hat{\theta}^{(i-1)})] \tag{5.25}$$

with α being an appropriate step-size. However, this algorithm works well only in the slowly time-varying case. To increase the speed of tracking, the second derivatives of the cost function $J(i)$ could be incorporated, as in the Gauss-Newton algorithm. Leung and So (2005) propose a similar approach, where the above step-size α is replaced by $\frac{\alpha}{1-\gamma(i-1)}$. As a result, the evolution of the forgetting factor is constrained to be bounded by two levels.

The solution introduced by Paleologu, Benesty, and Ciochina (2008) is based on the prediction error $\{\varepsilon(i,\hat{\theta}^{(i-1)})\}$ and on the signal $q(i) = \varphi^\top(i)P^{(i-1)}\varphi(i)$, with $\varphi(i)$ being the regressors vector at time i. Specifically, $\gamma(i)$ is updated as

$$\gamma(i) = \min\left\{\frac{\sqrt{\hat{\sigma}_q(i)\hat{\sigma}(i)}}{\xi + |\sqrt{\hat{\sigma}_e(i)} - \sqrt{\hat{\sigma}(i)}|}, \gamma_{max}\right\} \tag{5.26}$$

where $\hat{\sigma}_e$ and $\hat{\sigma}_q$ are the estimated variances of $\varepsilon(t,\hat{\theta}^{(i-1)})$ and of $q(i)$, while $\hat{\sigma}(i)$ is the current noise variance estimate. These quantities are recursively computed as

$$\hat{\sigma}_e(i) = \alpha\hat{\sigma}_e(i-1) + (1-\alpha)e^2(i) \tag{5.27}$$

$$\hat{\sigma}_q(i) = \alpha\hat{\sigma}_q(i-1) + (1-\alpha)q^2(i) \tag{5.28}$$

$$\hat{\sigma}(i) = \beta\hat{\sigma}(i-1) + (1-\beta)e^2(i) \tag{5.29}$$

with α and β being suitable step-sizes. It turns out that before an abrupt change of the system, $\hat{\sigma}_e(i)$ is large compared to $\hat{\sigma}(i)$; thus, $\gamma(i)$ takes low values, guaranteeing fast tracking. When a steady-state situation is detected, $\hat{\sigma}_e(i) \approx \hat{\sigma}(i)$ and $\gamma(i)$ tends to γ_{max}, thus slowing down the rate at which data are forgotten.

A more recent and involved approach for the update of $\gamma(i)$ is due to Bhotto and Antoniou (2013).

A third alternative postulates that the system parameters vary according to a stationary first-order Markov process, namely

$$\theta^{(i+1)} = \theta^{(i)} + w(i), \qquad \mathbb{E}[w(i)w^\top(i)] = R_1(i) \tag{5.30}$$

This model is generally adopted to describe the case of drifting parameters. For further details on this methodology, the interest reader is referred to Ljung (1999) (Sec. 11.2) ot Söderström and Stoica (1989) (Sec. 9.3).

It should be recalled that parametric methods require to a-priori specify a model class within which the model is searched for. If the properties of the underlying system vary significantly, it may happen that the selected model class is no more suitable to capture the whole system dynamics. In turn, a new choice should be made. As widely discussed in Section 2.5, such decision is typically taken by estimating models with different complexities and by applying tools such as cross-validation or information criteria to select the most appropriate one. Since the estimation of multiple models may be computationally expensive, such procedure could not be suited for the real-time identification of time-varying systems. On the other hand, the non-parametric Bayesian methods detailed in Section 2.4 overpass the aforementioned issue by jointly performing estimation and order selection, thus representing a sound alternative to parametric techniques. Section 5.3 will illustrate how the batch procedure detailed in Section 2.4 can be tailored to the real-time setup.

5.2 On-Line Identification with Subspace Methods

This section will briefly overview how subspace algorithms have been adapted to the on-line scenario illustrated in the chapter introduction. Technical details will be omitted, since subspace methods will not be taken into account in the experimental analysis performed in Section 5.4.

Before proceeding, it should be recalled that the core of any subspace algorithm is the SVD of data-depending matrices from which the extended observability matrix is derived. Such step also constitutes the major bottleneck in a possible real-time implementation of a subspace algorithm, because of its significant computational complexity. Hence, the attention of researchers has mainly focused on the development of routines which either recursively perform this stage or avoid it.

The literature on real-time implementations of subspace algorithms is not so vast, even if this topic has been treated since the beginning of the 1990s, when subspace algorithms became one the main research subjects for the system identification community. Indeed the first contributions date back to 1991 and 1994, with the works of Verhaegen and Deprettere (1991) and Cho, Xu, and Kailath (1994). The authors mainly focus on both the recursive update of the data matrices and on efficient ways of updating the SVD step. For instance, concerning the latter problem, Verhaegen and Deprettere (1991) propose to split the SVD stage into a partial update of an LQ factorization and a subsequent rank-one update of a previous SVD. A major drawback associated with these algorithms is the requirement for the output measurement noise to be spatially and temporally white.

A possible alternative to aforementioned approaches, which is not considered by these first works, is the possibility to directly update the estimate of the extended observability matrix. This way has been first explored by Gustafsson (1997) and Gustafsson, Lovera, and Verhaegen (1998). Specifically, they have extended the PAST (Projection Approximation Subspace Tracking) algorithm developed by Yang (1995) to the setting of subspace system identification. Such routine was introduced few years before into the signal processing community. As the name reflects, the algorithm is designed to recursively track a signal subspace from measurements affected by temporally and spatially white noise. With regard to subspace identification, the signal subspace is the column space of the extended observability matrix. PAST exploits RLS to solve a projection problem through which the signal subspace is retrieved. The computational complexity of the method proposed by Yang (1995) is $O(mn)$, where m is the size of the input vector, while n is the number of desired eigen-components, i.e. the desired dimension of the signal subspace. Yang (1995) proves that his algorithm represents a robust alternative to classical SVD approaches. However, because of the used approximations, the estimate returned by PAST converges to a slightly different subspace from the one obtained through the eigen-decomposition. Gustafsson et al. (1998) have developed the so-called IV-PAST (Instrumental Variables Projection Approximation Subspace Tracking), which extends PAST by introducing the instrumental variables in order to deal with the case in which the noise is not spatially white. It should be stressed that the proposed procedure assumes that the order of the system is a-priori known. This approach is extended by Oku and Kimura (2002), who adopt gradient type subspace tracking to search for the global minimizer of the projection problem above-mentioned. They also prove the convergence of the proposed algorithm under the assumption that the stepsize for the gradient update is within $[0, 1]$. Lovera, Gustafsson, and Verhaegen (2000) provides an overview of these approaches.

The work of Utschick (2002) lies at the basis of the algorithms proposed by Mercere, Lecoeuche, and Lovera (2004) and Mercère, Bako, and Lecœuche (2008). Compared to PAST, these methods do not introduce an approximation in the formulation of the tracking problem. The convergence properties of these propagator-based subspace identification methods are studied by Mercère and Lovera (2007), who show that under suitable conditions on the input signal and the system, these techniques return a consistent estimate of the state-space system matrices.

5.3 On-Line Identification with Non-Parametric Bayesian Methods

The batch technique described in Section 2.4 is here adapted to the on-line setup illustrated in the introduction to the chapter. To highlight the practical nature of this section, the estimation of a finite-length impulse response will be considered; hence, the perspective taken in Sections 2.4.1.3 and 2.4.2.3 will be adopted.

The Bayesian procedures of Section 2.4 mainly consist of two steps: hyper-parameters tuning and computation of the impulse response estimate. From a computational point of view, the first step is the most committing one: indeed, if the Empirical Bayes approach is adopted, once the hyper-parameters are fixed, the impulse response estimate can be efficiently computed through equation (2.193):

$$\hat{\mathbf{g}} = (\Phi_N^\top \widetilde{\Sigma}_N^{-1}\Phi_N + \bar{K}_\eta^{-1})^{-1}\Phi_N^\top\widetilde{\Sigma}_N^{-1}Y_N \tag{5.31}$$

However, this formulation is not suited for a real-time implementation, since a recursive update of the matrices appearing in the latter formula should be first derived. Specifically, at time $k+N$, when data $\mathcal{D}_{i+1}^N = \{u(t), y(t)\}_{t=iN+1}^{(i+1)N}$ arrive, the products of the data matrices appearing in equation (5.31) are updated through the following recursions

$$R^{(i+1)} := \Phi_{(i+1)N}^\top \widetilde{\Sigma}_{(i+1)N}^{-1}\Phi_{(i+1)N} = R^{(i)} + \left(\Phi_{iN+1}^{(i+1)N}\right)^\top \widetilde{\Sigma}_N^{-1}\Phi_{iN+1}^{(i+1)N} \tag{5.32}$$

$$\widetilde{Y}^{(i+1)} := \Phi_{(i+1)N}^\top \widetilde{\Sigma}_{(i+1)N}^{-1}Y_{(i+1)N} = \widetilde{Y}^{(i)} + \left(\Phi_{iN+1}^{(i+1)N}\right)^\top \widetilde{\Sigma}_N^{-1}Y_{iN+1}^{(i+1)N} \tag{5.33}$$

$$\bar{Y}^{(i+1)} := \bar{Y}_{(i+1)N}^\top \widetilde{\Sigma}_{(i+1)N}^{-1}\bar{Y}_{(i+1)N} = \bar{Y}^{(i)} + \left(\bar{Y}_{iN+1}^{(i+1)N}\right)^\top \widetilde{\Sigma}_N^{-1}\bar{Y}_{iN+1}^{(i+1)N} \tag{5.34}$$

where

$$\begin{aligned}
\Phi_{iN+1}^{(i+1)N} &:= \begin{bmatrix}\varphi(iN+1) & \cdots & \varphi(iN+N)\end{bmatrix}^\top, & \Phi_{iN+1}^{(i+1)N} &\in \mathbb{R}^{Np\times pmT}\\
Y_{iN+1}^{(i+1)N} &:= \begin{bmatrix}y^\top(iN+1) & \cdots & y^\top(iN+N)\end{bmatrix}, & Y_{iN+1}^{(i+1)N} &\in \mathbb{R}^{pN}
\end{aligned}$$

with $\varphi(t)$ as stated in equation (2.181). The definition of $\Phi_{(i+1)N}$ and $Y_{(i+1)N}$ is respectively given in equations (2.180) and (2.123) (with $(i+1)N$ replaced by N). Analogously, $\widetilde{\Sigma}_{(i+1)N}$ is specified in equation (2.171) with N in place of $(i+1)N$.

Recalling that T denotes the length of the estimated impulse response, the computational cost of the updates (5.32)-(5.34) is $O((pmT)^2(Np))$, $O((pmT)(Np))$ and $O((Np)^2)$, respectively.

The hyper-parameters tuning is here accomplished through Marginal Likelihood maximization (2.183). Denoting with $f_{ML}^N(\eta)$ the evidence function (2.184) computed with N data under Gaussian assumptions, it follows that the new hyper-parameters $\hat{\eta}^{(i+1)}$ have to be computed by minimizing $f_{ML}^{(i+1)N}(\eta) \equiv f_{ML}^{k+N}(\eta)$. The recursive updates (5.32)-(5.34) also allow to efficiently evaluate $f_{ML}^{(i+1)N}(\eta)$; namely, recalling equation (2.195):

$$f_{ML}^{(i+1)N}(\eta) = \bar{Y}^{(i+1)} - \left(\widetilde{Y}^{(i+1)}\right)^\top L(I_{Tmp} + L^\top R^{(i+1)}L)^{-1}L^\top \widetilde{Y}^{(i+1)} \tag{5.35}$$

$$+ (i+1)N(\sum_{j=1}^{p} \ln \sigma_j) + \ln \det(I_{Tmp} + L^\top R^{(i+1)} L) \tag{5.36}$$

where $LL^\top := \bar{K}_\eta$. As illustrated in Section 2.4.5.2, the Marginal Likelihood maximization could be performed through iterative routines, such as 1st or 2nd order optimization algorithms (Bonettini et al., 2015), or through the EM algorithm (Dempster et al., 1977; Bottegal et al., 2016). Since these methods may require a large number of iterations before reaching convergence, they may be unsuited for on-line applications. To overcome this issue and hence to tackle the real-time constraints, the procedure detailed in Algorithm 6 is here proposed. Its main feature is the computation of $\hat{\eta}^{(i+1)}$ by means of only one iteration of the aforementioned iterative algorithms. In particular, whenever new data arrive, such routines are initialized with the previous estimate $\hat{\eta}^{(i)}$, obtained using the data $\mathcal{D}_i \bigcup_{l=1}^{i} \mathcal{D}_l^N$, which is likely to be close to a local optimum of the old objective function $f_{ML}^{iN}(\eta)$. If the number of new data N is small, it is reasonable to suppose that $\arg\min_{\eta \in D_\eta} f_{ML}^{iN}(\eta) \approx \arg\min_{\eta \in D_\eta} f_{ML}^{(i+1)N}(\eta)$. Therefore, by just performing one iteration of the EM algorithm or of a gradient method, $\hat{\eta}^{(i+1)}$ will be sufficiently close to a local optimum of $f_{ML}^{(i+1)N}(\eta)$.

In the following such approach will be referred to as the *1-step Marginal Likelihood (ML)* method.

Algorithm 6 On-Line Bayesian System Identification

```
Inputs: previous estimates $\{\hat{\eta}^{(i)}, \hat{\eta}^{(i-1)}\}$, previous data matrices $\{R^{(i)}, \widetilde{Y}^{(i)}, \bar{Y}^{(i)}\}$,
    new data $\mathcal{D}_{i+1}^N = \{u(t), y(t)\}_{t=iN+1}^{(i+1)N}$
Use Recursive Least Squares to compute $\hat{\mathbf{g}}_{LS}^{(i+1)}$
Estimate $\widehat{\Sigma}$ using $\hat{\mathbf{g}}_{LS}^{(i+1)}$
Compute $R^{(i+1)}$ as in equation (5.32)
Compute $\widetilde{Y}^{(i+1)}$ as in equation (5.33)
Compute $\bar{Y}^{(i+1)}$ as in equation (5.34)
Compute $\hat{\eta}^{(i+1)}$ through 1-step Marginal Likelihood maximization initialized with
    $\hat{\eta}^{(i)}$ and $\hat{\eta}^{(i-1)}$
$\hat{\mathbf{g}}^{(i+1)} \leftarrow \left(R^{(i+1)} + \bar{K}_{\hat{\eta}^{(i+1)}}^{-1}\right)^{-1} \widetilde{Y}^{(i+1)}$
Output: $\hat{\mathbf{g}}^{(i+1)}$, $\hat{\eta}^{(i+1)}$
```

5.3.1 Dealing with Time-Varying Systems

As said in the introduction to the section, on-line algorithms find a natural application in the context of time-varying systems, where the data that progressively arrive are generated by changing systems. In order to tackle this kind of application, the estimators have to be equipped with tools through which past data are disregarded or become less relevant for the current estimation, since old information may be outdated. In the following, two routines which combine the “on-line Bayesian estimation” above sketched with the ability to “forget” past data are proposed.

5.3.1.1 Fixed Forgetting Factor

Following a classical practice in parametric system identification (see Section 5.1), a forgetting factor $\gamma \in (0,1]$ is introduced into the regularized estimation criterion (2.150). Specifically, at time k the estimate is determined as:

$$\begin{align}
\hat{\mathbf{g}} &:= \arg\min_{\mathbf{g}\in\mathbb{R}^{pmT}} \sum_{t=1}^{k} \gamma^{k-t}(y(t)-\varphi^\top(t)\mathbf{g})^\top \Sigma^{-1}(y(t)-\varphi^\top(t)\mathbf{g}) + \mathbf{g}^\top \bar{K}_{\hat{\eta}}^{-1}\mathbf{g} \tag{5.37}\\
&= \arg\min_{\mathbf{g}\in\mathbb{R}^{pmT}} (Y_k-\Phi_k\mathbf{g})^\top\, \Psi_k\widetilde{\Sigma}_k^{-1}\Psi_k\,(Y_k-\Phi_k\mathbf{g}) + \mathbf{g}^\top \bar{K}_{\hat{\eta}}^{-1}\mathbf{g} \nonumber\\
&= \left(\Phi_k^\top\Psi_k\widetilde{\Sigma}_k^{-1}\Psi_k\Phi_k + \bar{K}_{\hat{\eta}_\gamma}^{-1}\right)^{-1}\Phi_k^\top\Psi_k\widetilde{\Sigma}_k^{-1}\Psi_k Y_k \tag{5.38}
\end{align}$$

where

$$\Psi_k\Psi_k := \Gamma_k := \operatorname{diag}\left(\gamma^{k-1},\gamma^{k-2},...,\gamma^{0}\right)\otimes I_p \tag{5.39}$$

and $\varphi(t)$ has been defined in (2.181). Notice that, for simplicity, the same forgetting factor is applied on all the output channels.

It should be noticed that the introduction of the forgetting factor in the loss function (5.37) coincides with postulating a model of the type

$$\Psi_k Y_k = \Psi_k\Phi_k\mathbf{g} + E, \qquad E\sim\mathcal{N}(0_{kp},\ \widetilde{\Sigma}_k) \tag{5.40}$$

which, in turn, is equivalent to

$$Y_k = \Phi_k\mathbf{g} + E_\gamma, \qquad E_\gamma = \left[e_\gamma^\top(1),...,e_\gamma^\top(k)\right]^\top, \quad E_\gamma\sim\mathcal{N}(0_{pk},\Psi_k^{-1}\widetilde{\Sigma}_k\Psi_k^{-1}) \tag{5.41}$$

Therefore, the use of the forgetting factor as a hyper-parameter is equivalent to modelling the noise with a non-constant variance and to give to the diagonal entries of the covariance matrix an exponential decaying structure.

Correspondingly, the hyper-parameters should be estimated solving:

$$\begin{aligned}\hat{\eta} &= \underset{\eta \in D_\eta}{\arg\min}\ Y_k^\top (\Phi_k \bar{K}_\eta \Phi_k^\top + \Psi_k^{-1} \widetilde{\Sigma}_k \Psi_k^{-1})^{-1} Y_k + \ln\det(\Phi_k \bar{K}_\eta \Phi_k^\top + \Psi_k^{-1} \widetilde{\Sigma}_k \Psi_k^{-1}) \\ &= \underset{\eta \in D_\eta}{\arg\min}\ \Big\{ Y_k^\top \Psi_k\ (\Psi_k \Phi_k \bar{K}_\eta \Phi_k^\top \Psi_k + \widetilde{\Sigma}_k)^{-1} \Psi_k Y_k + \ln\det(\Psi_k \Phi_k \bar{K}_\eta \Phi_k^\top \Psi_k + \widetilde{\Sigma}_k) \\ &\qquad\qquad - \ln\det(\Gamma_k) \Big\} \end{aligned} \tag{5.42}$$

Algorithm 7 illustrates the on-line implementation of the identification procedure based on equations (5.38) and (5.42). In particular, it assumes that at time k the estimates $\hat{\mathbf{g}}^{(i)}$ and $\hat{\eta}^{(i)}$ are available and they have been computed by solving, respectively, (5.37) and (5.42); these estimates are then "on-line" updated once the new data $\mathcal{D}_{i+1}^N$ are provided. Notice that the forgetting factor γ explicitly appears in the updated of the data matrices (see steps 3-5 of Algorithm 7).

Algorithm 7 On-Line Bayesian System Identification - Fixed Forgetting Factor

```
   Inputs: forgetting factor $\gamma$, previous estimates $\left\{\hat{\eta}^{(i)}, \hat{\eta}^{(i-1)}\right\}$, previous data matrices
   $\left\{R_\gamma^{(i)}, \widetilde{Y}_\gamma^{(i)}, \bar{Y}_\gamma^{(i)}\right\}$, new data $\mathcal{D}_{i+1}^N = \{u(t), y(t)\}_{t=iN+1}^{(i+1)N}$
1: Use Recursive Least Squares to compute $\hat{\mathbf{g}}_{LS}^{(i+1)}$
2: Estimate $\widehat{\Sigma}$ using $\hat{\mathbf{g}}_{LS}^{(i+1)}$
3: $R_\gamma^{(i+1)} \leftarrow \gamma^N R_\gamma^{(i)} + \left(\Phi_{iN+1}^{(i+1)N}\right)^\top \Psi_N \widetilde{\Sigma}_N^{-1} \Psi_N\ \Phi_{iN+1}^{(i+1)N}$
4: $\widetilde{Y}_\gamma^{(i+1)} \leftarrow \gamma^N \widetilde{Y}_\gamma^{(i)} + \left(\Phi_{iN+1}^{(i+1)N}\right)^\top \Psi_N \widetilde{\Sigma}_N^{-1} \Psi_N\ \ Y_{iN+1}^{(i+1)N}$
5: $\bar{Y}_\gamma^{(i+1)} \leftarrow \gamma^N \bar{Y}_\gamma^{(i)} + \left(Y_{iN+1}^{(i+1)N}\right)^\top \Psi_N \widetilde{\Sigma}_N^{-1} \Psi_N\ Y_{iN+1}^{(i+1)N}$
6: $\hat{\eta}^{(i+1)} \leftarrow \arg\min_{\eta \in D_\eta}\ f_{ML}^{(i+1)N}(\eta)$
   (performing 1-step Marginal Likelihood maximization initialized with $\hat{\eta}^{(i)}, \hat{\eta}^{(i-1)}$)
7: $\hat{\mathbf{g}}^{(i+1)} \leftarrow \left(R_\gamma^{(i+1)} + \bar{K}_{\hat{\eta}^{(i+1)}}^{-1}\right)^{-1} \widetilde{Y}_\gamma^{(i+1)}$
   Output: $\hat{\mathbf{g}}^{(i+1)}$, $\hat{\eta}^{(i+1)}$
```

5.3.1.2 Treating the Forgetting Factor as a Hyper-parameter

The Bayesian framework provides the user with the possibility to treat the forgetting factor as a hyper-parameter and to estimate it through evidence maximization. Specifically, at time k (that is, at the i-th iteration of an online identification algorithm), the forgetting factor is estimated together with the usual hyper-parameters η by solving

$$\left(\hat{\eta}^{(i)}, \hat{\gamma}^{(i)}\right) = \underset{\eta \in D_\eta, \gamma \in (0,1]}{\arg\min}\ f_{ML}^k(\eta, \gamma) \tag{5.43}$$

with

$$f_{ML}^{k}(\eta,\gamma) = Y_k^\top(\Phi_k\bar{K}_\eta\Phi_k^\top + \Psi_k^{-1}(\gamma)\widetilde{\Sigma}_k\Psi_k^{-1}(\gamma))^{-1}Y_k \\ + \ln\det(\Phi_k\bar{K}_\eta\Phi_k^\top + \Psi_k^{-1}(\gamma)\widetilde{\Sigma}_k\Psi_k^{-1}(\gamma)) \tag{5.44}$$

Notice that the dependence of $\Psi_k(\gamma)$ on the unknown γ has been made explicit. To allow a recursive implementation of the corresponding identification algorithm, $\Psi_k(\gamma)$ has to be defined as:

$$\Psi_k(\gamma)\Psi_k(\gamma) := \Gamma_k(\gamma) := \text{blockdiag}(\gamma^N\widehat{\Gamma}^{(i-1)}, \boldsymbol{\gamma}_N(\gamma)) \tag{5.45}$$

where

$$\boldsymbol{\gamma}_N(\gamma) = \text{diag}\left(\begin{bmatrix}\gamma^{N-1} & \cdots & \gamma & 1\end{bmatrix}\right) \otimes I_p, \qquad \boldsymbol{\gamma}_N(\gamma) \in \mathbb{R}^{Np\times Np} \tag{5.46}$$

$$\widehat{\Gamma}^{(i-1)} = \text{blockdiag}\left(\prod_{l=1}^{i-2}\boldsymbol{\gamma}_N(\hat{\gamma}^{(i-l)}),\ \cdots,\ \boldsymbol{\gamma}_N(\hat{\gamma}^{(i-1)})\right), \qquad \widehat{\Gamma}^{(i-1)} \in \mathbb{R}^{Np(i-1)\times Np(i-1)} \tag{5.47}$$

For future use, define also $\boldsymbol{\psi}_N(\gamma)\boldsymbol{\psi}_N(\gamma) := \boldsymbol{\gamma}_N(\gamma)$.
Correspondingly, the products of the data matrices are updated as

$$R^{(i+1)}(\gamma) = \gamma^N\Phi_{iN}^\top\widehat{\Gamma}^{(i)}\Phi_{iN}^\top + \left(\Phi_{iN+1}^{(i+1)N}\right)^\top \boldsymbol{\psi}_N(\gamma)\widetilde{\Sigma}_N^{-1}\boldsymbol{\psi}_N(\gamma)\ \Phi_{iN+1}^{(i+1)N} \\ =: \gamma^N\widehat{R}^{(i)} + \left(\Phi_{iN+1}^{(i+1)N}\right)^\top \boldsymbol{\psi}_N(\gamma)\widetilde{\Sigma}_N^{-1}\boldsymbol{\psi}_N(\gamma)\ \Phi_{iN+1}^{(i+1)N} \tag{5.48}$$

Analogous recursions hold true for $\widetilde{Y}^{(i+1)}(\gamma)$ and $\bar{Y}^{(i+1)}(\gamma)$.
The on-line implementation of this approach is detailed in Algorithm 8. Differently from the previous algorithms, the marginal likelihood maximization at step (6) of Algorithm 8 also requires to compute the derivative $\frac{\partial f_{ML}^{k}(\eta,\gamma)}{\partial\gamma}$. An efficient computation of this quantity exploits the recursive updates performed at steps 3-5 of Algorithm 8.

Algorithm 8 Online Bayesian SysId: Forgetting Factor as a hyper-parameter

```
Inputs: previous estimates $\{\hat{\eta}^{(i)}, \hat{\eta}^{(i-1)}, \hat{\gamma}^{(i)}, \hat{\gamma}^{(i-1)}\}$, previous data matrices $\{\widehat{R}^{(i)}, \widehat{\widetilde{Y}}^{(i)}, \widehat{\bar{Y}}^{(i)}\}$, new data $\mathcal{D}_{i+1} = \{u(t), y(t)\}_{t=iN+1}^{(i+1)N}$
Use Recursive Least Squares to compute $\hat{\mathbf{g}}_{LS}^{(i+1)}$
Estimate $\widehat{\Sigma}$ using $\hat{\mathbf{g}}_{LS}^{(i+1)}$
$R^{(i+1)}(\gamma) \leftarrow \gamma^N \widehat{R}^{(i)} + \left(\Phi_{iN+1}^{(i+1)N}\right)^\top \boldsymbol{\psi}_N(\gamma)\widetilde{\Sigma}_N^{-1}\boldsymbol{\psi}_N(\gamma)\ \Phi_{iN+1}^{(i+1)N}$
$\widetilde{Y}^{(i+1)}(\gamma) \leftarrow \gamma^N \widehat{\widetilde{Y}}^{(i)} + \left(\Phi_{iN+1}^{(i+1)N}\right)^\top \boldsymbol{\psi}_N(\gamma)\widetilde{\Sigma}_N^{-1}\boldsymbol{\psi}_N(\gamma)\ Y_{iN+1}^{(i+1)N}$
$\bar{Y}^{(i+1)}(\gamma) \leftarrow \gamma^N \widehat{\bar{Y}}^{(i)} + \left(Y_{iN+1}^{(i+1)N}\right)^\top \boldsymbol{\psi}_N(\gamma)\widetilde{\Sigma}_N^{-1}\boldsymbol{\psi}_N(\gamma)\ Y_{iN+1}^{(i+1)N}$
$\hat{\eta}^{(i+1)}, \hat{\gamma}^{(i+1)} \leftarrow \arg\min_{\eta \in D_\eta, \gamma \in (0,1]}\ f_{ML}^{(i+1)N}(\eta, \gamma)$
(performing 1-step Marginal Likelihood maximization initialized with $\hat{\eta}^{(i)}, \hat{\eta}^{(i-1)}, \hat{\gamma}^{(i)},\ \hat{\gamma}^{(i-1)}$)
$\widehat{R}^{(i+1)} \leftarrow \left(\hat{\gamma}^{(i+1)}\right)^N \widehat{R}^{(i)} + \left(\Phi_{iN+1}^{(i+1)N}\right)^\top \boldsymbol{\psi}_N(\hat{\gamma}^{(i+1)})\widetilde{\Sigma}_N^{-1}\boldsymbol{\psi}_N(\hat{\gamma}^{(i+1)})\ \Phi_{iN+1}^{(i+1)N}$
$\widehat{\widetilde{Y}}^{(i+1)} \leftarrow \left(\hat{\gamma}^{(i+1)}\right)^N \widehat{\widetilde{Y}}^{(i)} + \left(\Phi_{iN+1}^{(i+1)N}\right)^\top \boldsymbol{\psi}_N(\hat{\gamma}^{(i+1)})\widetilde{\Sigma}_N^{-1}\boldsymbol{\psi}_N(\hat{\gamma}^{(i+1)})\ Y_{iN+1}^{(i+1)N}$
$\widehat{\bar{Y}}^{(i+1)} \leftarrow \left(\hat{\gamma}^{(i+1)}\right)^N \widehat{\bar{Y}}^{(i)} + \left(Y_{iN+1}^{(i+1)N}\right)^\top \boldsymbol{\psi}_N(\hat{\gamma}^{(i+1)})\widetilde{\Sigma}_N^{-1}\boldsymbol{\psi}_N(\hat{\gamma}^{(i+1)})\ Y_{iN+1}^{(i+1)N}$
$\hat{\mathbf{g}}^{(i+1)} \leftarrow \left(\widehat{R}^{(i+1)} +\ \bar{K}_{\hat{\eta}^{(i+1)}}^{-1}\right)^{-1} \widehat{\widetilde{Y}}^{(i+1)}$
Output: $\hat{\mathbf{g}}^{(i+1)}$, $\hat{\eta}^{(i+1)}$, $\hat{\gamma}^{(i+1)}$
```

5.4 Numerical Results

The proposed adaptation of non-parametric Bayesian methods to the real-time setup is here evaluated through two Monte-Carlo studies. Section 5.4.1 will compare several iterative algorithms for the optimization of the marginal likelihood: in particular, the aim is to evaluate whether the proposed *1-step Marginal Likelihood* approach is effective in terms of quality of the returned estimates and of computational savings. 5.4.2 will compare RPEM with the algorithm introduced in Section 5.3 on a Monte-Carlo scenario composed of time-varying systems.

5.4.1 Time-Invariant Systems

5.4.1.1 Data

200 Monte-Carlo runs are here considered: for each of them a random SISO discrete-time system is generated through the MATLAB routine `drmodel.m` (see Remark 3.5.1 for a detailed description of the function). The system orders have been randomly chosen in the range $[5, 10]$, while the systems poles are all inside a circle of radius 0.95. The input signal is a unit variance band-limited Gaussian signal with normalized band $[0, 0.8]$. A zero mean white Gaussian noise, with variance adjusted so that the Signal to Noise Ratio (SNR) is always equal to 5, isw added to the output data. For each Monte-Carlo run 5000 input-output data pairs have been generated, while the length N of the on-line upcoming datasets $\mathcal{D}_i^N$ is set to 10.

5.4.1.2 Identification Algorithms

The on-line version of the Bayesian approaches illustrated in Section 2.4 is here evaluated. Specifically, the procedure which estimates the hyper-parameters by means of an iterative algorithm which run until convergence (such as a gradient methods or the EM) is compared with that which performs only one iteration of the aforementioned methods (as illustrated in Algorithm 6). In the following the first procedure will be referred to as OPT, while the notation 1-STEP ML will be adopted for the latter one. While OPT exploits the SGP routine (Algorithm 1) to solve the Marginal Likelihood maximization problem, multiple algorithms are compared when applied to accomplish such step in the 1-STEP ML procedure. Specifically, the following routines are evaluated:

SGP: Algorithm 1.

BB: Algorithm 1 with scaling matrix set equal to the identity: $D^{(i)} = I_{d_\eta}$; the name BB refers to the Barzilai-Borwein rules adopted to fix the stepsize $\alpha^{(i)}$.

BFGS: Algorithm 1 where the product $\alpha^{(i)}D^{(i)}$ at step 5 is replaced by the BFGS inverse Hessian approximation (Nocedal and Wright, 2006):

$$B^{(i)} := (I - \rho r^{(i-1)} w^{(i-1)^\top}) B^{(i-1)} (I - \rho w^{(i-1)} r^{(i-1)^\top}) + \rho r^{(i-1)} r^{(i-1)^\top} \tag{5.49}$$

where

$$\rho : = 1/(w^{(i-1)^\top} r^{(i-1)}) \tag{5.50}$$

$$r^{(i-1)} : = \eta^{(i)} - \eta^{(i-1)} \tag{5.51}$$

$$w^{(i-1)} : = [f'_{ML}(\eta^{(i)}) - f'_{ML}(\eta^{(i-1)})]^\top \tag{5.52}$$

EM: Algorithm 3.

Finally, it should be stressed that the on-line Algorithm 6 is initialized by computing the batch procedure on the first 100 data.

In the following experiments, the length T of the estimated impulse responses is set to 80, while the adopted kernel is the TC one (3.27):

$$\left[\bar{K}_\eta^{TC}\right]_{kj} = \lambda \min(\beta^k, \beta^j), \qquad \eta = [\lambda,\ \beta],\ \lambda \geq 0,\ 0 \leq \beta \leq 1 \tag{5.53}$$

Notice that such kernel is defined by means of two hyper-parameters: the scaling factor λ and the decay rate β. In the interest of reducing the computational time of the on-line updates two versions of BFGS, SGP, BB, EM are proposed: the first one updates both the hyper-parameters in η whenever a new dataset $\mathcal{D}_i^N$ becomes available, while the second one updates only the scaling factor λ, retaining β fixed to its initial value. It is clear that the latter case allows a faster computation, at the expense of a less precise impulse response estimate. In addition, two cases of the EM version which only updates λ are considered:

EM2: The correct formula for the update of λ is adopted, that is

$$\widehat{\lambda}^{(i+1)} = \frac{1}{pmT} \left(\hat{\mathbf{g}}^{(i)^\top} \bar{K}_{\hat{\beta}}^{-1} \hat{\mathbf{g}}^{(i)} + \operatorname{Tr}\left\{ \bar{K}_{\hat{\beta}}^{-1} (R^{(i+1)} + \bar{K}_{\hat{\beta}}^{-1})^{-1} \right\} \right) \tag{5.54}$$

EM1: the following approximated update is used:

$$\widehat{\lambda}^{(i+1)} = \frac{1}{pmT} \hat{\mathbf{g}}^{(i)^\top} \bar{K}_{\hat{\beta}}^{-1} \hat{\mathbf{g}}^{(i)} \tag{5.55}$$

Equation (5.55) represents the current approximation of the asymptotically optimal value for λ. The aim is to show a comparison between the asymptotic theory and the EM update (Aravkin et al., 2014).

5.4.1.3 Impulse Response Estimates

The adherence of the impulse response estimate to the true one is here evaluated. For each estimated system and for each procedure the following impulse response fit is computed:

$$\mathcal{F}_T(\hat{\mathbf{g}}) = 100 \cdot \left(1 - \frac{\|\mathbf{g}_0 - \hat{\mathbf{g}}\|_2}{\|\mathbf{g}_0 - \bar{\mathbf{g}}_0\|_2}\right), \qquad \bar{\mathbf{g}}_0 = \frac{1}{T}\sum_{j=1}^{T}[\mathbf{g}_0]_j \tag{5.56}$$

where $\mathbf{g}_0$, $\hat{\mathbf{g}} \in \mathbb{R}^T$ respectively contain the true and the estimated truncated impulse coefficients of the considered system.

Figure 5.1 shows the impulse response fits (5.56) achieved in the Monte-Carlo simulations along with the increase of the number of observed data. Specifically, k on the top of each plot denotes the number of data that have been so far processed by a certain algorithm. OPT procedure is compared with 1-STEP ML when implemented with the algorithms SGP, BB, BFGS and EM (that is, the single step of marginal likelihood optimization is performed by computing single iteration of one of these routines). On the left hand side the obtained results optimizing both hyper-parameters of kernel TC (5.53) are reported, while the results on the right hand side are obtained by updating only λ.

All the 1-STEP ML procedures which update both hyper-parameters perform remarkably well, with the fit index being almost equivalent to the one obtained with the OPT procedure. This suggests that the complete optimization of the Marginal Likelihood does not bring any particular advantage in terms of fit in the on-line setting. Notice that a sort of worst case approximation is taken, since the optimization algorithm is stopped after only one step: some more advanced techniques could be considered (e.g. an early stopping criterion (Yao, Rosasco, and Caponnetto, 2007)). The 1-STEP ML updates optimizing only λ , after a transient period, perform comparably (but slightly worse) to the other techniques; the only exception is represented by EM1 which achieves inferior fits.

5.4.1.4 Computational Time

The cumulative computational time of the algorithms detailed in Section 5.4.1.2 is here evaluated. The term “cumulative time” here denotes the time spent by a certain algorithm to process k data. Figure 5.2 contains the relative boxplots, while Table 5.1 reports the average values of the computed cumulative time, together with their standard deviation. The OPT procedure, as expected, is much slower than the 1-STEP ML procedures.

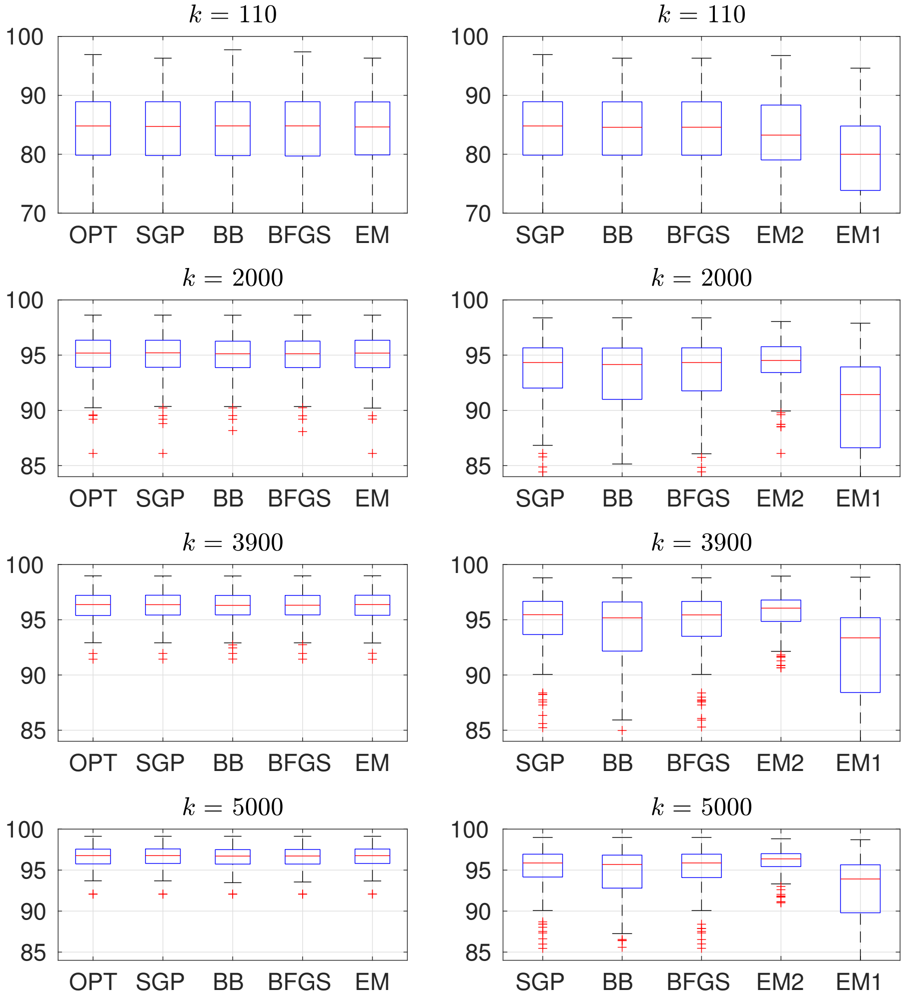


Figure 5.1: Monte Carlo results over 200 runs - Boxplots of the impulse response fit (5.56) achieved by the identification algorithms listed in Section 5.4.1.2. *Left:* Both the hyper-parameters of kernel $\bar{K}^{TC}_{\eta}$ (5.53) are updated. *Right*: Only hyper-parameter λ of kernel $\bar{K}^{TC}_{\eta}$ (5.53) is updated.

	Update λ and β					Update only λ				
	OPT	SGP	BB	BFGS	EM	SGP	BB	BFGS	EM2	EM1
mean	163.1	0.56	0.93	1.19	0.57	0.31	0.60	0.45	0.18	0.30
std	18.45	0.13	0.16	0.36	0.11	0.06	0.13	0.25	0.06	0.92

Table 5.1: Monte-Carlo results over 200 runs - Mean and standard deviation (std) of the cumulative computational time required by the algorithms listed in Section 5.4.1.2 to process 5000 data. *Left columns:* Both the hyper-parameters of kernel $\bar{K}^{TC}_{\eta}$ (5.53) are updated. *Right columns*: Only hyper-parameter λ of kernel $\bar{K}^{TC}_{\eta}$ (5.53) is updated.

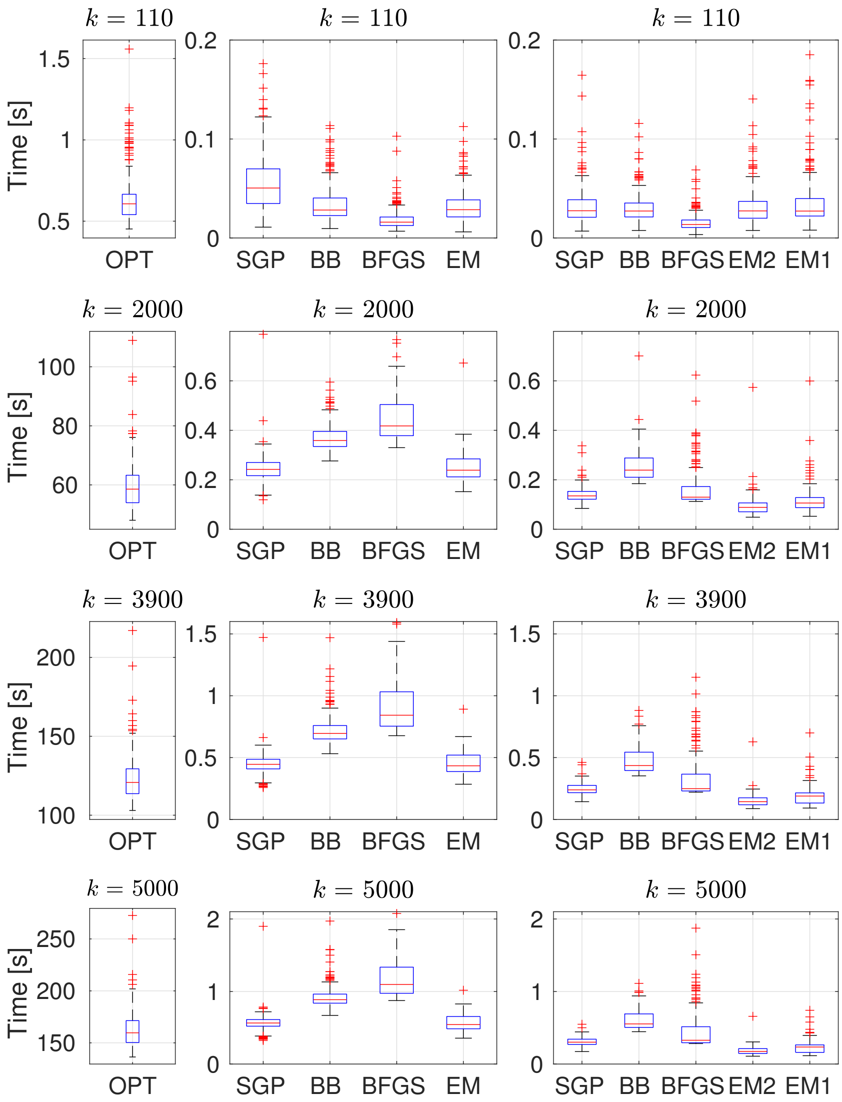


Figure 5.2: Monte Carlo results over 200 runs - Boxplots of the cumulative computational time required by the identification algorithms listed in Section 5.4.1.2. Each row of plots reports the time required to process k data. *Left:* OPT procedure (which updates both the hyper-parameters of kernel $\bar{K}_\eta^{TC}$ (5.53)). *Mid:* Both the hyper-parameters of kernel $\bar{K}_\eta^{TC}$ (5.53) are updated through 1-STEP ML optimization. *Right*: Only hyper-parameter λ of kernel $\bar{K}_\eta^{TC}$ (5.53) is updated through 1-STEP ML optimization.

This could suggest that the 1-STEP ML routines appear to be excellent candidates for real-time applications. Indeed, these techniques perform comparably in terms of fit w.r.t. the OPT procedure, but demanding a computational time which is two or three order of magnitude faster. Furthermore, the difference in terms of computational time diverges in favour of the 1-STEP ML procedure with the increase of the number of processed data.

Among the 1-STEP ML procedures SGP and EM provide the fastest updates: this is surprisingly positive for the EM update since only λ has a closed form update, while β is the solution of a maximization problem; indeed, in the right hand side of Figure 5.2, where only λ is updated, EM1 and EM2 outperform SGP. The update BB is a particular case of SGP, where $D^{(i)} = I_{d_\eta}$, but it is significantly slower: this is probably due to the backtracking loop at steps 8-12 in Algorithm 1.
As a final remark, the right hand side of Figure 5.2 shows the advantage of updating only λ: the cumulative computational time is significantly lower than that appearing in the mid-column of the figure.

5.4.2 Time-Varying Systems

In this section RPEM and the on-line version of the Bayesian methods of Section 2.4 are experimentally evaluated on a Monte-Carlo study composed of 200 time-varying systems.

5.4.2.1 Data

200 datasets consisting of 4000 input-output measurement pairs are generated. Each of them is created as follows: the first 1000 data are produced by a system contained in the data-bank D4 (used in Chen et al. (2014)), while the remaining 3000 data are generated by perturbing the D4-system with two additional poles and zeros. These are chosen such that the order of the D4-system changes, thus creating a switch on the data generating system at time $k = 1001$.
The data-bank D4 consists of 30th order random SISO dicrete-time systems having all the poles inside a circle of radius 0.95. These systems are simulated with a unit variance band-limited Gaussian signal with normalized band $[0, 0.8]$. A zero mean white Gaussian noise, with variance adjusted so that the Signal to Noise Ratio (SNR) is always equal to 1, is then added to the output data.

5.4.2.2 Identification Algorithms

RPEM: The parametric estimators are computed with the `roe` MATLAB routine, using the BIC criterion for model class selection (see (2.218)). In the following this estimator will be denoted as RPEM+BIC. Furthermore, the parametric oracle estimator is introduced as a benchmark (and called RPEM+OR): it selects the model complexity by choosing the model that gives the best fit to the impulse response of the true system. The order selection is performed every time a new dataset becomes available: multiple models with orders ranging from 1 to 20 are

estimated and the order selection is performed according to the two above-described criteria.
Both methods adopts a forgetting factor γ equal to 0.998.

Non-Parametric Bayesian Methods: The TC kernel (3.27) is adopted also in this case, while the length T of the estimated impulse responses is set to 100. In the following, the acronym TC will denote non-parametric methods. As before, the notation OPT will refer to the standard Bayesian procedure, in which the SGP algorithm adopted to optimize the marginal likelihood $f_{ML}^k(\eta)$ is run until convergence, i.e until the relative change in $f_{ML}^k(\eta)$ is less than 10^{-9}. The acronyms TC FF and TCestFF refer to the 1-STEP ML procedure: TC FF denotes the use of a fixed forgetting factor (Algorithm 7), while TCestFF is related to the treatment of the forgetting factor as a hyper-parameter (Algorithm 8).
The forgetting factor in TC FF is set to 0.998, while its estimation in TCestFF is initialized with 0.995.

For each Monte-Carlo run, the identification algorithms are initialized using the first 300 data. After this initial step, the estimators are updated every $N = 10$ time steps, when new data $\mathcal{D}_{i+1}^N = \{u(t), y(t)\}_{t=iN}^{(i+1)N}$ are provided.

5.4.2.3 Impulse Response Estimates

The adherence of the estimated impulse response $\hat{\mathbf{g}}$ to the true one $\mathbf{g}_0$ is first evaluated through criterion (5.56).
Figure 5.4 shows the average fit (over the 200 Monte-Carlo runs) achieved at each time instant by the identification algorithms listed in Section 5.4.2.2. The results observed with time-invariant systems are here confirmed, since the methods TC OPT FF and TC FF performs identically (indeed, the line corresponding to the method TC OPT FF is not visible, because it coincides with that of TC FF).
It is interesting to note that immediately before the change in the data generating system ($k = 1000$) the TC methods slightly outperform the ideal parametric estimator RPEM+OR. After the switch (occurring at $k = 1001$), among the regularization/Bayesian routines TCestFF recovers the fit performance a bit faster than TC FF; moreover, even at regime it outperforms the latter because it can choose forgetting factor values that retain a larger amount of data.

The unrealistic RPEM+OR represents the reference on the achievable performance of the RPEM estimators; it outperforms TC methods in the transient after the switch, while it has comparable performance at regime. On the other hand, RPEM+BIC estimator performs very poorly.

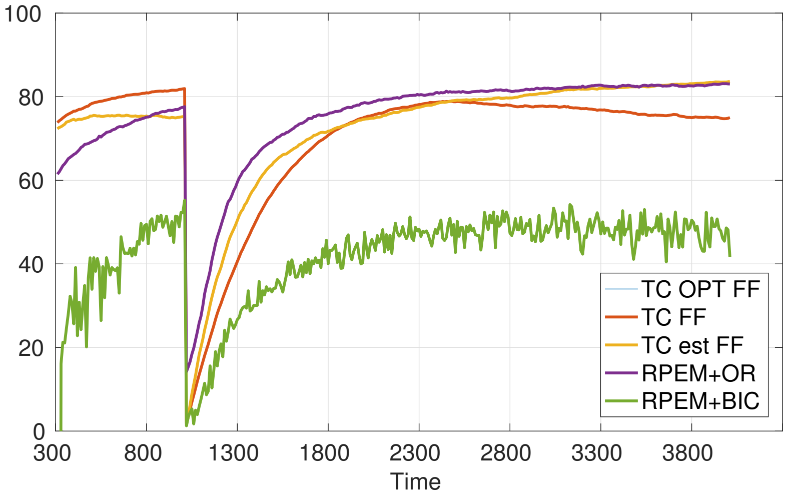


Figure 5.3: Monte-Carlo Results over 200 runs - Average impulse response fit (5.56) achieved at each time instant by the identification algorithms listed in Section 5.4.2.2.

Figure 5.3 reports the boxplots of the average fit (5.56) achieved by the tested identification algorithms over the 4000 available data. The observed results confirm the effectiveness of the on-line implementation of Bayesian methods, which perform almost comparably with the RPEM unrealistically equipped with an oracle.

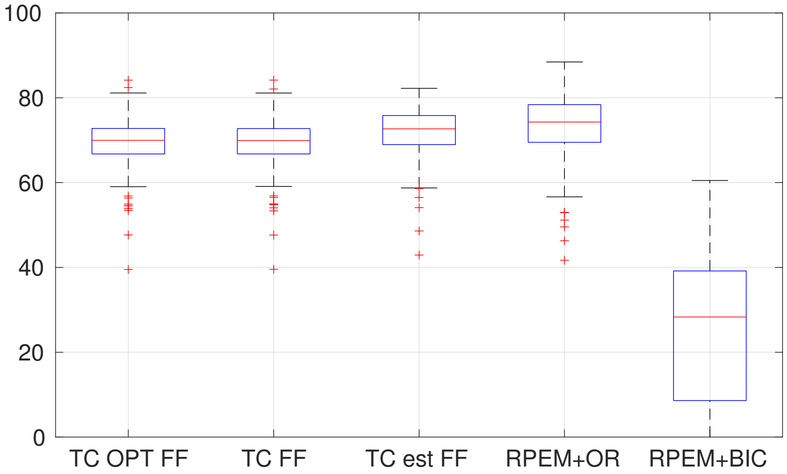


Figure 5.4: Monte-Carlo results over 200 runs - Boxplots of the average over time of the impulse response fit (5.56) achieved by the identification algorithms listed in Section 5.4.2.2.

	TC			RPEM	
	TC OPT FF	TC FF	TCestFF	RPEM+OR	RPEM+BIC
mean	6.70	0.44	0.90	18.44	18.44
std	1.28	0.03	0.37	0.69	0.69

Table 5.2: Monte-Carlo results over 200 runs - Computational cumulative time after data 4000 have been processed: mean and standard deviation (std) over 200 datasets.

5.4.2.4 Computational Time

Table 5.2 analyses the cumulative computational time of the evaluated identification algorithms: specifically,the reported values are its mean and standard deviation computed after the estimators are fed with all the 4000 data contained in the designed datasets. The 1-STEP ML methods are one order of magnitude faster than the corresponding OPT ones. The TCestFF estimator appears a bit slower slower than TC FF, since three hyper-parameters have to be estimated at each iteration. On the other hand the RPEM estimators are three times slower than OPT ones, thus appearing not particularly appealing for on-line applications. The large computational effort detected for RPEM is due to the necessity of selecting a new model complexity, whenever new data arrive.

6
Model Reduction

The estimators $\hat{\mathbf{g}}$ produced by the Bayesian methods described in Section 2.4 are FIR models of length T. As previously remarked, the value of T is not related to the complexity of the estimated model, but it only depends on the dominant time constant of the system. It should be recalled that within the regularization framework model selection is implicitly performed through the choice of the regularization parameters (or hyper-parameters in a Bayesian setting) appearing in the penalty terms; model complexity can be measured in terms of degrees of freedom (Hastie et al., 2009; Pillonetto and Chiuso, 2015). However this quantity does not directly relate to the McMillan degree of the system, which instead measures the complexity of a minimal state space realization.
Once the high-order FIR estimate (2.148) has been obtained, it would be desirable to approximate it with a lower order state-space model, more suited for filtering and control purposes. Indeed, high-order models lead to complex controllers and prediction filters, whose implementation may be critical. The approximation can be achieved either by computing a high-order state-space realization of the FIR model and subsequently reducing it to the desired low order, or by directly building a state-space realization from the impulse response data contained in the FIR model, according to one of the algorithms proposed e.g. by Ho and Kalman (1966) and Kung (1978) (see Section 6.1). The former approach involves the adoption of a model reduction procedure, which computes a reduced-order approximation of the original system, while preserving its main dynamical properties. The model reduction problem has been intensively studied by the control systems community, as proved by the surveys Antoulas, Sorensen, and Gugercin (2001); Gugercin and Antoulas (2004) and the books Antoulas (2005a); Obinata and Anderson (2012). Most of the existing techniques approximate the original large-scale system by means of a projection onto a lower dimensional space. A brief overview of these methodologies will be provided in Section 6.1.

The role played by model reduction in system identification will be briefly discussed in Section 6.2. While model reduction is implicitly performed by subspace algorithms, some contributions (Wahlberg, 1989b; Söderström et al., 1991; Galrinho, Rojas, and Hjalmarsson, 2014) have connected it to PEM by developing two-stage procedures, where an initial high-order model is estimated through PEM and then reduced according to some “optimal” criteria. On the other hand, little attention has been devoted by the literature of Bayesian system identification on a possible post-processing stage, where the high-order estimated FIR model is converted into a more manageable low-order one. The work presented in this chapter aims at investigating some procedures (detailed in Section 6.3) which could robustly perform such reduction stage. The transition to the parametric framework requires to choose the complexity of the reduced model, which

turns out to be a crucial ingredient. For this reason, the numerical analysis in Section 6.4 is largely focused on the comparison of several order selection techniques.
As a final remark, it should be mentioned that goal-oriented model reduction is also typically applied in the control system field: the intended use of the low order model is explicitly taken into account in the reduction criterion (Hovland, Willcox, and Gravdahl, 2006; Bui-Thanh, Willcox, Ghattas, and van Bloemen Waanders, 2007; Carlberg and Farhat, 2011). However, the approach taken in the work here illustrated does not consider a specific goal, but simply intends to transform a high-order estimated FIR model into a low-order system, suitable for general use. Extensions to goal-oriented reduction could be developed, but are out of the scope of the present contribution.

6.1 Model Reduction in Control System Theory

The theory of model reduction for LTI systems is generally formulated in terms of state-space models. For ease of notation, in the remainder of the chapter, a state-space system described by the matrices A, B, C and D will be compactly denoted as $\mathscr{G} = (A, B, C, D)$. Moreover, the terms "order" and "McMillan degree" of a system will be interchangeably used in the rest of the chapter.
The classical model reduction problem solved in the linear system theory can be stated as follows.

Given the state-space system of order n, $\mathscr{G} = (A, B, C, D)$, find a system $\widehat{\mathscr{G}} = (\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D})$ of order (equivalently, McMillan degree) $\rho < n$ such that:

1. *Basic properties, like stability and passivity, are preserved.*
2. *$\mathscr{G}$ and $\widehat{\mathscr{G}}$ are close in terms of the $\mathcal{H}_\infty$ or the $\mathcal{H}_2$ norms. It should be recalled that*

$$\|G - \widehat{G}\|_{\mathcal{H}_2} := \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi} \mathrm{Tr}\left[\left(G(e^{j\omega}) - \widehat{G}(e^{j\omega})\right)^\top \left(G(e^{j\omega}) - \widehat{G}(e^{j\omega})\right)\right] d\omega} \tag{6.1}$$

$$\|G - \widehat{G}\|_{\mathcal{H}_\infty} := \sup_{\omega\in[0,\pi]} s_{\max}\left(G(e^{j\omega}) - \widehat{G}(e^{j\omega})\right) \tag{6.2}$$

with $G(e^{j\omega})$ denoting the frequency response of system $\mathscr{G}$ and $s_{\max}(A)$ denoting the largest singular value of matrix A.

Additionally, the computational and storage requirements of the reduction procedure should be restrained.

Basically, model reduction procedures derive the low order system $\widehat{\mathscr{G}}$ by means of an

appropriate projection. Specifically, the projection is defined as $\Pi = VW^\top$, where $V, W \in \mathbb{R}^{n\times\rho}$ are such that $W^\top V = I_\rho$. The approximating state $\hat{x}$ is then computed as $\Pi x = V\hat{x}$. The corresponding reduced system is described by the matrices $\widehat{\mathscr{G}} = (W^\top AV, W^\top B, CV, D)$ (Antoulas, 2005b; Benner, Gugercin, and Willcox, 2015).
The book Antoulas (2005a) divides the various model reduction algorithms into three main categories:

1. SVD-based methods
2. Krylov-based methods
3. SVD- and Krylov-based methods

These families are briefly discussed in the next sections.

6.1.1 SVD-based Methods

For linear systems, SVD-based methods include the balanced truncation and the Hankel approximation. While the former technique will be briefly outlined in the remainder of the section, the latter will not be treated in this manuscript. It is worth mentioning that the Hankel approximation is optimal w.r.t. the 2-induced norm of the Hankel operator; explicit formulas for optimal and suboptimal approximations exist and an error bound in the $\mathcal{H}_\infty$-norm has been derived. For further details on this method, the interested reader is referred to Glover (1984), Latham and Anderson (1985) and Antoulas (2005a) (Ch. 8).

Balanced model reduction is a sound and widely adopted procedure, which was introduced by Moore (1981). The basic technique is the so-called *Lyapunov balancing method*, which first requires to transform the large-scale system $\mathscr{G}$ into its balanced realization. To determine the so-called "balanced" basis, the two Lyapunov equations

$$A\mathcal{P}A^\top - \mathcal{P} = -BB^\top, \qquad \mathcal{P} > 0 \tag{6.3}$$

$$A^\top \mathcal{Q}A - \mathcal{Q} = -C^\top C, \qquad \mathcal{Q} > 0 \tag{6.4}$$

need to be solved, thus obtaining the reachability and the observability gramians:

$$\mathcal{P} := \sum_{i=1}^{\infty} A^i BB^\top (A^i)^\top, \qquad \mathcal{Q} := \sum_{i=1}^{\infty} (A^i)^\top C^\top C A^i \tag{6.5}$$

In the balanced realization, the two gramians are simultaneously diagonalized, namely:

$$\mathcal{P} = \mathcal{Q} = \operatorname{diag}(s_1, \cdots, s_n) \tag{6.6}$$

where s_i, $i = 1, ..., n$ are the Hankel singular values of the system $\mathscr{G}$, which equal the square roots of the eigenvalues of the product $\mathcal{P}\mathcal{Q}$, $s_i = \sqrt{\lambda_i(\mathcal{P}\mathcal{Q})}$. As a consequence of condition 6.6, in a balanced realization every state is as controllable as it is observable. Hence, the states can be ordered in terms of their contribution to the input-output properties of the system and the states providing the least contribution can be removed in order to obtain a reduced model.

Once the system $\mathscr{G}$ is transformed into its balanced realization, $\mathscr{G} = (A_b, B_b, C_b, D)$, the system matrices are partitioned as

$$A_b = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right], \qquad B_b = \left[\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right], \qquad C_b = \left[\begin{array}{c|c} C_1 & C_2 \end{array}\right] \tag{6.7}$$

where $A_{11} \in \mathbb{R}^{\rho\times\rho}$, $B_1 \in \mathbb{R}^{\rho\times m}$, $C_1 \in \mathbb{R}^{p\times\rho}$. The corresponding reduced order model is $\widehat{\mathscr{G}} = (A_{11}, B_1, C_1, D)$ and the associated projector is $\Pi = VW^\top$ with $V = W = [I_\rho \;\; 0_{n-\rho}]^\top$. Among the advantages of this reduction procedure, there are the preservation of the system stability (i.e. $\widehat{\mathscr{G}}$ is stable if $\mathscr{G}$ is stable) and the existence of a global error bound, namely

$$s_{\rho+1} \leq \|G - \widehat{G}\|_{\mathcal{H}_\infty} \leq 2(s_{\rho+1} + \cdots + s_n) \tag{6.8}$$

However, the algorithm requires matrix factorizations and inversions, making its computational effort of order $O(n^3)$; in addition, since no iterative way of computing the reduced order exists, the whole original system has to be stored, making the storage requirement of order $O(n^2)$. As a consequence, approximate and efficient versions of the above-detailed Lyapunov balanced truncation have been developed; see the survey Gugercin and Antoulas (2004) and the book Antoulas (2005a), where also other types of balancing are reviewed, such as stochastic balancing, bounded real balancing, positive real balancing and frequency weighted balancing.

A balanced state-space realization of a desired order can also be computed starting from impulse response data by means of the algorithm initially proposed by Ho and Kalman (Ho and Kalman, 1966). This routine was developed to solve the so-called "minimal state-space realization problem for LTI systems", which can be stated as follows:

Given some data about an LTI system, find a state-space description of minimal size that explains the given data.

The data could be e.g. the impulse response of the system, its step response, some input-output measurements or frequency response data.

The minimal state-space realization problem has been studied since the early 1960s, when was first faced by Gilbert (Gilbert, 1963) and Kalman (Kalman, 1965). In 1966, Ho and Kalman formulated the procedure reported in Algorithm 9 which returns the minimal state-space realization starting from the entire sequence of Markov parameters of the system (Ho and Kalman, 1966). Kalman (1971) and Tether (1970) extended the original version to handle partial sequences of Markov parameters, while the routine proposed by Kung (Algorithm 10) can be applied when a finite number of noisy Markov parameters of an LTI system is available (Kung, 1978). An overview of the several algorithms proposed in the literature to solve the minimal state-space realization problem can be found in the survey by De Schutter (2000).

Some comments on the pseudo-code in Algorithm 9 and 10 are needed. At step 4 of the two algorithms, the shifted Hankel matrix $\bar{\mathbf{G}}$ is built; for generic numbers r and c of block rows and columns, it is defined as:

$$\bar{\mathbf{G}} = \begin{bmatrix} g(2) & g(3) & \cdots & g(c+1) \\ g(3) & g(4) & \ddots & g(c+2) \\ \vdots & \vdots & \ddots & \vdots \\ g(r+1) & g(r+2) & \cdots & g(r+c) \end{bmatrix} \tag{6.9}$$

At step 5 of Algorithm 9, the full rank decomposition of $\mathbf{G}$ can be reliably determined computing its SVD, $\mathbf{G} = USV^\top$, $U \in \mathbb{R}^{pr \times pr}$, $V \in \mathbb{R}^{mr \times mr}$ and $S \in \mathbb{R}^{pr \times mr}$. Accordingly, $\mathbf{G}_o = US^{1/2}$ and $\mathbf{G}_c = S^{1/2}V^\top$. As a final comment, following the Matlab convention, the notation $A(1:m, 1:n)$ denotes the block of the first m rows and n columns extracted from matrix A.

Algorithm 9 Ho and Kalman Algorithm

```
   Inputs: Entire sequence of impulse response coefficients ${g(k)}_{k=0}^{\infty}$
1: $\widehat{D} \leftarrow g(0)$
2: Choose $r$ (large enough), the number of Hankel block rows and columns
3: Build the Hankel matrix $\mathbf{G} \in \mathbb{R}^{pr \times mr}$
4: Build the shifted Hankel matrix $\bar{\mathbf{G}} \in \mathbb{R}^{pr \times mr}$
5: Compute the full-rank factorization: $\mathbf{G} \leftarrow \mathbf{G}_o\mathbf{G}_c$, $\mathbf{G}_o \in \mathbb{R}^{pr \times \rho}$, $\mathbf{G}_c \in \mathbb{R}^{\rho \times rm}$
6: $\widehat{A} \leftarrow \mathbf{G}_o^\dagger \bar{\mathbf{G}} \mathbf{G}_c^\dagger$
7: $\widehat{B} \leftarrow \mathbf{G}_c(:, 1:m)$
8: $\widehat{C} \leftarrow \mathbf{G}_o(1:p, :)$
   Output: $\widehat{\mathscr{G}} = (\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D})$, balanced state-space realization of order $\rho$
```

Algorithms 9 and 10 return state-space realizations whose system matrices have all non-zero entries, meaning that in general $\rho(\rho+p+m)+pm$ entries have to be computed.

Algorithm 10 Kung's Algorithm

```
Inputs: Partial sequence of impulse response coefficients $\{g(k)\}_{k=0}^{N}$
$\widehat{D} \leftarrow g(0)$
Choose the number of Hankel block rows $r$ and columns $c$ such that $r + c = N$
Build the Hankel matrix $\mathbf{G} \in \mathbb{R}^{pr\times mc}$
Build the shifted Hankel matrix $\bar{\mathbf{G}} \in \mathbb{R}^{pr\times mc}$
Compute the SVD of $\mathbf{G}$: $\mathbf{G} \leftarrow USV^\top$
Choose the number $\rho$ of singular values $S_{ii}$ to be retained
$U_\rho \leftarrow U(:, 1:\rho)$
$V_\rho \leftarrow V(:, 1:\rho)$
$S_\rho \leftarrow S(1:\rho, 1:\rho)$
$\mathbf{G}_o \leftarrow U_\rho S_\rho^{1/2}$
$\mathbf{G}_c \leftarrow S_\rho^{1/2} V_\rho^\top$
$\widehat{A} \leftarrow \mathbf{G}_o^\dagger \bar{\mathbf{G}} \mathbf{G}_c^\dagger$
$\widehat{B} \leftarrow \mathbf{G}_c(:, 1:m)$
$\widehat{C} \leftarrow \mathbf{G}_o(1:p, :)$
Output: $\widehat{\mathscr{G}} = (\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D})$, balanced state-space realization of order $\rho$
```

Several authors have developed extensions of these procedures, which allow to derive state-space models with specific canonical structures.

6.1.2 Krylov-based Methods

Krylov-based methods rely on *moment matching.* The moment of a system $\mathscr{G}$ at $q_0 \in \mathbb{C}$ are the coefficients of the Laurent series expansion of the transfer function $G(q)$ around q_0:

$$G(q) = G(q_0) + G^{(1)}(q)\frac{(q-q_0)}{1!} + G^{(2)}(q)\frac{(q-q_0)^2}{2!} + \cdots + G^{(k)}(q)\frac{(q-q_0)^k}{k!} + \cdots \tag{6.10}$$

$$= \eta_0(q_0) + \eta_1(q_0)\frac{(q-q_0)}{1!} + \eta_2(q_0)\frac{(q-q_0)^2}{2!} + \cdots + \eta_k(q_0)\frac{(q-q_0)^k}{k!} + \cdots \tag{6.11}$$

It should be observed that the moments are the Markov coefficients of $\mathscr{G}$ if the expansion is computed around infinity, i.e. $\eta_0(\infty) = D,\ \eta_k(\infty) = CA^{k-1}B,\ k > 0$.
Moment matching approximates the original system $\mathscr{G}$ by finding $\widehat{\mathscr{G}} = (\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D})$, such that its transfer function can be expanded as

$$\widehat{G}(q) = \hat{\eta}_0(q_0) + \hat{\eta}_1(q_0)\frac{(q-q_0)}{1!} + \hat{\eta}_2(q_0)\frac{(q-q_0)^2}{2!} + \hat{\eta}_k(q_0)\frac{(q-q_0)^3}{3!} + \cdots \tag{6.12}$$

and

$$\eta_j(q_0) = \hat{\eta}_j(q_0), \qquad j = 1, 2, ..., k \tag{6.13}$$

for an appropriate k. The problem of finding $\widehat{\mathscr{G}}$ through moment matching is also known as *rational interpolation*. This problem can be solved by means of iterative procedures, which avoid the direct computation of the moments. Hence, w.r.t. SVD-based methods, Krylov-based approaches admit numerically efficient implementations, such as the well-known *Lanczos* and *Arnoldi* algorithms. The number of numerical operations they require is $O(\rho n^2)$ or $O(\rho^2 n)$, which needs to be compared to the complexity of $O(n^3)$ characterizing SVD-based methods.

At the ρ-th iteration, the Arnoldi algorithm builds an orthogonal matrix V_ρ and the application of the projection $\Pi = V_\rho V_\rho^\top$ leads to the reduced-order model $\widehat{\mathscr{G}} = (\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}) = (V_\rho^\top A V_\rho, V_\rho^\top B, C V_\rho, D)$. The Arnoldi procedure guarantees that the first ρ Markov parameters are matched, namely

$$\hat{g}(j) = \widehat{C}\widehat{A}^{j-1}\widehat{B} = CA^{j-1}B = g(j), \qquad j = 1, ..., \rho \tag{6.14}$$

The two-sided Lanczos algorithm is an alternative routine, which iteratively constructs two biorthogonal matrices V_ρ and W_ρ (i.e., such that $W_\rho^\top V_\rho = I_\rho$), starting from the matrix A and the vectors B and $C^\top$. The reduced-order system is obtained by means of the projection $\Pi = V_\rho W_\rho^\top$, leading to $\widehat{\mathscr{G}} = (\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}) = (W_\rho^\top A V_\rho, W_\rho^\top B, C V_\rho, D)$. In this case the first 2ρ moments are matched:

$$\hat{g}(j) = \widehat{C}\widehat{A}^{j-1}\widehat{B} = CA^{j-1}B = g(j), \qquad j = 1, ..., 2\rho \tag{6.15}$$

In practice, at the ρ-th iteration of both the Arnoldi and the Lanczos method, a certain canonical form of $\mathscr{G} = (A, B, C, D)$ is derived and the reduced-order system is obtained by truncating the state. However, thanks to the iterative implementation, the reduced matrices $\widehat{A}$, $\widehat{B}$ and $\widehat{C}$ are directly computed, thus avoiding the explicit computation of the canonical forms as well as the state truncation.

6.1.3 SVD and Krylov-based methods

The discussion in Sections 6.1.1 and 6.1.2 highlighted how SVD-based and Krylov-based methods are characterized by some advantages but also by some drawbacks.

SVD-based approaches preserve the stability of the original system and enjoy a global error bound between the large-scale and the reduced-order system; however, the computational and storage requirements are significant, since matrix inversions and factorization have to be performed and no iterative routine can be applied.

On the other hand, Krylov-based methods admit iterative implementations, which simply require matrix-vector multiplications, thus making these approaches particularly efficient from the numerical point of view. However, no a-priori error bound can be derived for the reduced-order system and the preservation of stability is not guaranteed. In addition, Krylov methods tend to approximate better the high frequency components of the original system, leading sometimes to relevant steady-state errors. To overcome this issue, *rational Krylov methods* can be adopted, where the matching is done on the coefficients of the Laurent series expansion around frequencies different from infinity.

Recent research on the field has tried to combine the benefits of the two families of methods in order to overcome the aforementioned drawbacks characterizing them. For instance, iterative methods have been developed to approximatively solve the Lyapunov equations (6.3) and (6.4), which represent the computational bottleneck of balanced truncation (Sorensen and Antoulas, 2002; Gugercin, Sorensen, and Antoulas, 2003; Penzl, 2006). Along this research line, Gugercin (2008) proposes an iterative method returning a reduced-order system, which is stable, matches certain moments and solves an $\mathcal{H}_2$ minimization problem. Similar guarantees are also achieved by the least-squares approximation proposed in Gugercin and Antoulas (2006).

Gugercin, Antoulas, and Beattie (2008) develop a new set of local optimality conditions for the $\mathcal{H}_2$ model reduction problem, proving that the existing SVD- and Krylov-based optimality conditions are equivalent to each other.

More details on these combined approaches can be found in the book Antoulas (2005a) (Ch. 12).

6.2 Model Reduction in System Identification

From a certain perspective, system identification can be viewed as a model reduction problem. Indeed, the given dataset $\mathcal{D}^N$ of N input-output data can be interpreted as a non-parametric model of the unknown system. Consequently, system identification turns out to be a model reduction procedure which converts such N-th order model into a lower order one within a specified model set (Ljung, 1985). Differently from the approaches discussed in Section 6.1, system identification operates on noisy data, meaning that the high-order system is just an approximation of the underlying unknown system. This introduces new considerations when it comes to choose the order of the reduced model. While the reduction procedures discussed in Section 6.1.1 admit precise error bounds between the high- and low-order models, such results should be carefully exploited in system identification. Indeed, a small error between the high- and low-dimensional models does not directly imply a good fit with the underlying unknown system. Stated in other words, the risk of overfitting should always be considered. These observations simply restate how the model class selection is a crucial stage in any identification procedure, as already remarked throughout this manuscript.

The next sections intend to provide a brief overview of the interplay between model reduction procedures and the three types of system identification methods which have been illustrated in Chapter 2.

6.2.1 Model Reduction and Prediction Error Methods

PEM offer two main routes to estimate low-order models. The first and standard procedure is to directly apply PEM on the given data $\mathcal{D}^N$ to search for an approximation of the unknown system within a pre-specified model set. Indeed, PEM return estimates which are L_2 approximations of the true system in a frequency-weighted norm defined by the input spectrum $S_u(\omega)$: denoting with $\hat{\theta}_N$ the PE estimate and with θ^* the best model within the chosen model class, it holds

$$\hat{\theta}_N \overset{N\to\infty}{\longrightarrow} \theta^* = \arg\min_{\theta\in D_\theta} \int_{-\pi}^{\pi} \mathrm{Tr}\left[\Big(G_0(e^{j\omega}) - G(e^{j\omega},\theta)\Big)^\top S_u(\omega) \Big(G_0(e^{j\omega}) - G(e^{j\omega},\theta)\Big)\right] d\omega \tag{6.16}$$

A second route prescribes to adopt PEM to first estimate a high-order model and in a second stage to reduce it to the desired order by minimizing a specific criterion or by applying one of the techniques illustrated in Section 6.1.

The remainder of this section will focus on this second procedure and will provide a short overview of some contributions which have considered such estimation approach.

The first works treating the reduction of high-order models returned by an estimation algorithm appeared in the time-series literature. Durbin (1960) proposed the idea of first estimating a high-order AR model and subsequently using it to form a low-order ARMA estimate. Other contributions in the time series literature are due to Mayne and Firoozan (1982) and to Wahlberg (1989a), who followed Durbin's approach but transformed it into the frequency domain, thus fortmulating the reduction step as an L_2-norm approximation problem. In the control field, the early work of Genesio and Pomé (1975) was followed by the algorithms proposed by Wahlberg (1989b), Söderström et al. (1991) and Zhu and Backx (2012).

Wahlberg (1989b) proposes a two-steps procedure, where a high-order model is first estimated and then reduced according to Maximum Likelihood criterion, based on the asymptotic distribution of the high-order estimate. Such criterion turns out to be a frequency weighted L_2-norm model reduction, with the weighting function given by the inverse variance of the high-order estimate. The author also mentions the possibility of modifying such weighting function in order to account for the intended use of the low-order model.

To briefly illustrate the method, the estimation of a FIR model of order n in the first stage is here considered (even if other model structures are admitted, as outlined in Wahlberg (1989b)):

$$\hat{\theta}_N = R_N^{-1}(n) \sum_{t=1}^{N} \varphi(t) y(t) \tag{6.17}$$

with

$$R_N(n) : = \sum_{t=1}^{N} \varphi(t) \varphi^\top(t), \qquad \varphi(t) := [u(t-1) \ \cdots \ u(t-n)]^\top \tag{6.18}$$

The model is then reduced according to the criterion

$$\hat{\nu}_N = \arg\min_{\nu \in D_\nu} (F_1(\nu) - \hat{\theta}_N)^\top R_N(n) (F_1(\nu) - \hat{\theta}_N) \tag{6.19}$$

$$F_1(\nu) = R_N(n)^{-1} \sum_{t=1}^{N} G(q,\nu) u(t) \varphi(t) \tag{6.20}$$

Replacing $R_N(n)$ by its limit $R(n)$,

$$\lim_{N\to\infty} \frac{1}{N} R_N(n) = \begin{bmatrix} R_u(0) & R_u(1) & \cdots & R_u(n-1) \\ R_u(1) & R_u(0) & \cdots & R_u(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_u(n-1) & R_u(n-2) & \cdots & R_u(0) \end{bmatrix} =: R(n) \tag{6.21}$$

with

$$R_u(\tau) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_u(e^{j\omega}) e^{j\tau\omega} d\omega \tag{6.22}$$

criterion (6.19) can be expressed in the frequency domain as

$$\begin{aligned} \hat{\nu}_N = \arg\min_{\nu \in D_\nu} \Bigg\{ \int_{-\pi}^{\pi} \mathrm{Tr} \left[\left(\widehat{G}(e^{j\omega}, \hat{\theta}_N) - G(e^{j\omega}, \nu) \right)^\top S_u(e^{j\omega}) \left(\widehat{G}(e^{j\omega}, \hat{\theta}_N) - G(e^{j\omega}, \nu) \right) \right] d\omega \\ + \Delta_\nu(n,N) \Bigg\} \end{aligned} \tag{6.23}$$

with $\lim_{N\to\infty} \|\Delta_\nu(n,N)\|_2$ going exponentially to zero as $n \to \infty$.

Under mild conditions (exponentially stable system and persistence of excitation), Wahlberg (1989b) proves that $\hat{\nu}_N$ is an asymptotically efficient estimate, meaning that the Cramér-Rao bound is met as the order n of the large FIR model and the number of data N tend to infinity.

The Indirect Prediction Error Method (IPEM) proposed by Söderström et al. (1991) uses PEM (specifically, Least-Squares) to estimate a high-order model $\hat{\theta}_N$ within a model structure $\mathcal{M}_2$ and subsequently reduces it to a simpler model belonging to $\mathcal{M}_1$, such that $\mathcal{M}_1 \subset \mathcal{M}_2$. Since $\mathcal{M}_1$ and $\mathcal{M}_2$ are nested, there exists a non-linear map $F_2(\theta)$ such that $\nu = F_2^{-1}(\theta)$. The proposed reduction criterion is

$$\bar{\nu} = \arg\min_{\nu \in D_\nu} (F_2(\nu) - \hat{\theta}_N)^\top \widehat{P}_N^{-1} (F_2(\nu) - \hat{\theta}_N) \tag{6.24}$$

where $\widehat{P}_N$ is a consistent estimate of

$$P_\theta = \left\{ \mathbb{E}\left[\left(\frac{\partial \varepsilon(t,\theta)}{\partial \theta}\bigg|_{\theta=\theta_0} \right)^\top \left(\frac{\partial \varepsilon(t,\theta)}{\partial \theta}\bigg|_{\theta=\theta_0} \right) \right] \right\}^{-1} \tag{6.25}$$

and $\varepsilon(t,\theta)$ in equation (6.25) denotes the prediction error achieved using the parameter θ, while θ_0 is the true parameter vector. Notice that equation (6.25) is the asymptotic covariance of the normalized estimation errors; hence, recalling the discussion of Section 4.1.1.2, a natural estimate is

$$\widehat{P}_N = \left\{ \frac{1}{N} \sum_{t=1}^{N} \left(\frac{\partial \varepsilon(t,\theta)}{\partial \theta}\bigg|_{\theta=\hat{\theta}_N} \right)^\top \left(\frac{\partial \varepsilon(t,\theta)}{\partial \theta}\bigg|_{\theta=\hat{\theta}_N} \right) \right\}^{-1} \tag{6.26}$$

The authors prove that the estimate $\bar{\nu}$ in (6.24) has the same asymptotic distribution of $\hat{\nu}_N$, the ML estimate (that is, the one returned by standard PEM). However, IPEM results to be more computationally efficient than classical PEM, thanks to the LS problem solved at the first stage and to the tailored Gauss-Newton algorithm developed by the authors to solve the reduction problem (6.24).

An alternative approach is the one proposed by Zhu and Backx (2012), whose starting point is the estimation of a ARX model of (large) order n through LS, thus obtaining the polynomials $A(q,\hat{\theta}_N)$ and $B(q,\hat{\theta}_N)$. These are used in an intermediate step to filter the original input-output data:

$$u_f(t) = A(q,\hat{\theta}_N)u(t), \qquad y_f(t) = \frac{B(q,\hat{\theta}_N)}{A(q,\hat{\theta}_N)} u_f(t) \tag{6.27}$$

As a final stage, a low order OE model is estimated from the data $\{y_f(t), u_f(t)\}_{t=1}^N$. The authors prove that this approach is asymptotically efficient in the model order n and in the number of data N.

A statistical analysis of these two-steps procedures is provided by Tjärnström and Ljung (2002) and Tjärnström (2003). Tjärnström and Ljung (2002) prove that directly estimating a low-order FIR model from the data results into a larger variance w.r.t. first estimating a high-order FIR and then reducing it to the desired low order. In the case of OE models, the two procedures are equivalent in terms of the variance of the estimates, if the reduced model class contains the true system. If this is not the case (i.e. in presence of undermodelling), Tjärnström (2003) proves that the low-order OE model obtained through L_2 reduction of a higher-order estimate has a smaller variance than a low-order OE model directly inferred from the given data.

Finally, a more recent contribution (Galrinho et al., 2014) introduces an iterative procedure consisting of three LS problems:

1. A high-order FIR model $\hat{\theta}_N$ is estimated trough LS from the given data $\mathcal{D}^N$.
2. $\hat{\theta}_N$ is reduced to a structured model, $\bar{\nu} \in D_\nu$, by solving a second LS problem, that is, by projecting $\hat{\theta}_N$ onto the low dimensional space D_ν.
3. The final estimate $\hat{\nu}_N$ is fitted through weighted LS to $\hat{\theta}_N$, using the weights obtained from $\bar{\nu}$.

The authors claim that their method is asymptotically efficient under mild assumptions.

6.2.2 Model Reduction and Subspace Methods

Among the identification techniques illustrated in Chapter 2, subspace methods probably show the strongest interplay with the concept of model reduction. Indeed, the procedure detailed in Section 2.3 could be viewed as a model reduction algorithm, where the N-dimensional subspace directly derived from the given data $\mathcal{D}^N$ is reduced to the so-called signal subspace of size $n \leq N$. In practice, this reduction is performed by resorting to an SVD-based approach, as clarified by equations (2.78) and (2.89).

6.2.3 Model Reduction and Non-parametric Bayesian Methods

As mentioned in introduction of this chapter, the high-order FIR model returned by a non-parametric Bayesian identification procedure may not be suited for the intended use of the model. For instance, if the estimation stage is just the preliminary step for the subsequent controller design, the order of the resulting controller will be large, thus complicating its analysis and its implementation.

The possibility of transforming the non-parametric estimate into a model belonging to a desired model set M is investigated in the seminal paper of Pillonetto and De Nicolao (2010), where a mean-square optimal approximation is suggested. Specifically, if $\hat{g}$ denotes the Bayesian estimate defined in equation (2.128), the approximation $\hat{g}^M \in M$ returned by the proposed criterion

$$\hat{g}^M = \arg\min_{g \in M} \sum_{t=0}^{\infty} \mathrm{Tr}\left[(\hat{g}(t) - g(t))^\top W(t)(\hat{g}(t) - g(t))\right] \tag{6.28}$$

also minimizes the weighted MSE, that is

$$\hat{g}^M = \arg\min_{g \in M} \sum_{t=0}^{\infty} \mathbb{E}\left[\mathrm{Tr}[(g_0(t) - g(t|y^N))^\top W(t)(g_0(t) - g(t|y^N))] \;\Big|\; y^N\right] \tag{6.29}$$

where $g(t|y^N)$ denotes the impulse response computed starting from the observations y^N and $W(\cdot)$ a suitably designed weighting function. It should be clarified that the notation $g \in M$ in equations (6.28) and (6.29) indicates that g is the impulse response of a model included in the class M.

In practice, criterion (6.28) suggests a two-stage procedure, where a non-parametric Bayesian estimate is first computed and then approximated through a projection onto the set M. Furthermore, by Parseval's theorem, criterion (6.28) is easily translated into the frequency domain, thus becoming a classical L_2 approximation problem, i.e.

$$\widehat{G}^M = \arg\min_{G \in M} \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathrm{Tr}\left[(\widehat{G}(e^{j\omega}) - G(e^{j\omega}))^\top W(\omega)(\widehat{G}(e^{j\omega}) - G(e^{j\omega}))\right] d\omega \tag{6.30}$$

As before, the notation $G \in M$ means that $G(e^{j\omega})$ is the frequency response of a model belonging to the set M.

No mention is given by the authors about possible ways of selecting the model class M in order to achieve the best fit with the true unknown system. This step turns out to be crucial in determining the goodness of the reduced model: if M is not properly chosen, the gap between the true system and the reduced model could be significantly larger than the original error produced by the Bayesian estimate. For this reason, the investigation conducted in the next two sections is particularly focused on the problem of model class selection: besides proposing two reduction procedures, several techniques for complexity selection are compared. The experimental results reported in Section 6.4 will show how the problem is not trivial, suggesting the need for further investigations.

6.3 From Non-parametric to Parametric Models: Model Reduction Meets Order Selection

This section introduces two routines which transform the FIR model estimated through a non-parametric Bayesian algorithm into a low-order OE model. Several methods for the choice of its complexity are experimentally compared (these are listed in Section 6.3.1). The final goal of this investigation is the development of a completely automatic procedure, which takes as input a high-order unstructured estimate (returned by a Bayesian identification algorithm) and reduces it to a structured model with low McMillan degree.

Algorithm 11 Reduction of non-parametric Bayesian estimates

```
   Input: $\widehat{\mathcal{M}}$, a realization of order $T$ of the FIR impulse response estimate $\hat{\mathbf{g}}$ (2.148)
1: for $\rho = 1$ to $T - 1$ do
2:     Use one of the techniques illustrated in Section 6.1 to approximate $\widehat{\mathcal{M}}$ with the
       model $\widehat{\mathcal{M}}_\rho$ of order $\rho$.
3:     Using the original model $\widehat{\mathcal{M}}$, compute the prediction $\hat{y}^N$ on the estimation data
       $\mathcal{D}^N$.
4:     Estimate an OE model $\widehat{\mathcal{M}}_\rho^{PEM}$ (by means of the MATLAB routines pem or oe)
       using the data $\widehat{\mathcal{D}}^N = \{\hat{y}(t), u(t)\}_{t=1}^N$ and the model $\widehat{\mathcal{M}}_\rho$ as initialization for the
       routine.
5:     Using the model $\widehat{\mathcal{M}}_\rho^{PEM}$, compute the prediction $\hat{y}_\rho^N$ on the estimation data $\mathcal{D}^N$.
6: end for
7: Use one of the criteria listed in Section 6.3.1 to select the final model $\widehat{\mathcal{M}}_*^{PEM}$ within
   the set $\left\{\widehat{\mathcal{M}}_{\rho^*}^{PEM};\ \rho = 1, ..., T-1\right\}$.
   Output: $\widehat{\mathcal{M}}_{\rho^*}^{PEM}$, OE model of order $\rho^* << T$.
```

Algorithm 11 summarizes the proposed procedure for the reduction of the high-order FIR model $\hat{\mathbf{g}}$ (2.148) into a low-order OE model.

The numerical experiments in Section 6.4 will compare Algorithm 11 with an alternative approach, where the reduced non-parametric estimate is used to initialize a classical PEM routine applied on the original data $\mathcal{D}^N$. For the sake of clarity, this procedure is outlined in Algorithm 12. A similar approach has already be considered in the frequency domain by Geerardyn, Lumori, and Lataire (2015).

It should be noticed that Algorithms 11 and 12 consist of two model reduction stages: one computed at step 2 by means of an SVD- or a Krylov-based method (see Section 6.1) and an L_2-norm approximation performed at step 4 of Algorithm 11 and at step 3 of Algorithm 12.

Algorithm 12 Reduction of non-parametric Bayesian estimates

```
   Input: $\widehat{\mathcal{M}}$, a realization of order $T$ of the FIR impulse response estimate $\hat{\mathbf{g}}$ (2.148)
1: for $\rho = 1$ to $T-1$ do
2:     Use one of the techniques illustrated in Section 6.1 to approximate $\widehat{\mathcal{M}}$ with the
       model $\widehat{\mathcal{M}}_\rho$ of order $\rho$.
3:     Estimate an OE model $\widehat{\mathcal{M}}_\rho^{PEM}$ (by means of the MATLAB routines pem or oe)
       using the estimation data $\mathcal{D}^N$ and the model $\widehat{\mathcal{M}}_\rho$ as initialization for the routine.
4:     Using the model $\widehat{\mathcal{M}}_\rho^{PEM}$, compute the prediction $\hat{y}_\rho^N$ on the estimation data $\mathcal{D}^N$.
5: end for
6: Use one of the criteria listed in Section 6.3.1 to select the final model $\widehat{\mathcal{M}}_{\rho^*}^{PEM}$ within
   the set $\left\{\widehat{\mathcal{M}}_\rho^{PEM};\ \rho = 1, ..., T-1\right\}$.
   Output: $\widehat{\mathcal{M}}_{\rho^*}^{PEM}$, OE model of order $\rho^* << T$.
```

The procedure detailed in Algorithm 11 deserves some additional comments.
If the FIR model $\widehat{\mathcal{M}}$ was estimated by classical LS, steps 3 and 4 would coincide with the asymptotically efficient procedure proposed by Wahlberg (1989b). To clarify the connection, the one-step ahead predictor corresponding to the initial FIR estimate $\hat{\mathbf{g}}$ is rewritten as

$$\hat{y}(t) = \sum_{k=1}^{T} \hat{g}(k)u(t-k) = [\hat{g}(1) \ \cdots \ \hat{g}(T)] \begin{bmatrix} u(t-1) \\ \vdots \\ u(t-T) \end{bmatrix} =: \underline{\hat{g}}^\top \varphi(t) \tag{6.31}$$

In addition, the corresponding frequency response is expressed as

$$\widehat{G}(e^{j\omega}) = \underline{\hat{g}}^\top \begin{bmatrix} e^{-j\omega} I_m \\ \vdots \\ e^{-jT\omega} I_m \end{bmatrix} =: \underline{\hat{g}}^\top W_T(\omega) \tag{6.32}$$

Analogously, the one-step ahead predictor for the OE model parametrized by $\nu \in D_\nu$ is given by

$$\begin{aligned} \hat{y}(t|\nu) = G(q,\nu)u(t) = \sum_{k=1}^{\infty} g_\nu(k)u(t-k) &= \sum_{k=1}^{T} g_\nu(k)u(t-k) + \sum_{k=T+1}^{\infty} g_\nu(k)u(t-k) \\ &= \left(\underline{g}_{\nu,T}\right)^\top \varphi(t) + \Delta_1(T,N) \end{aligned} \tag{6.33}$$

Assuming a bounded input signal, that is $|u(t)| < C$ for some $C > 0$, then $\lim_{N\to\infty} \|\Delta_1(T,N)\|_2$ goes exponentially to zero as T tends to infinity. Correspondingly, the frequency response can be written as

$$G(e^{j\omega},\nu) = \left(\underline{g}_{\nu,T}\right)^\top \begin{bmatrix} e^{-j\omega} I_m \\ \vdots \\ e^{-jT\omega} I_m \end{bmatrix} + \Delta_2(T,N) =: \left(\underline{g}_{\nu,T}\right)^\top \; W_T(\omega) + \Delta_2(T,N) \tag{6.34}$$

where $\lim_{N\to\infty} \|\Delta_2(T,N)\|_2$ decreases exponentially to zero in T.
Having defined the above quantities, the optimization problem solved by the PEM routine applied at step 4 of Algorithm 11 can be stated as follows

$$\begin{aligned}
\hat{\nu}_N &= \arg\min_{\nu\in D_\nu} \sum_{t=1}^N \mathrm{Tr}\left[\left(\hat{y}(t) - \hat{y}(t|\nu)\right)\left(\hat{y}(t) - \hat{y}(t|\nu)\right)^\top\right] \\
&= \arg\min_{\nu\in D_\nu} \sum_{t=1}^N \mathrm{Tr}\left[\left(\left(\underline{\hat{g}} - \underline{g}_{\nu,T}\right)^\top \varphi(t) - \Delta_1(T,N)\right) \cdot \left(\left(\underline{\hat{g}} - \underline{g}_{\nu,T}\right)^\top \varphi(t) - \Delta_1(T,N)\right)^\top\right] \\
&= \arg\min_{\nu\in D_\nu} \sum_{t=1}^N \mathrm{Tr}\left[\left(\underline{\hat{g}} - \underline{g}_{\nu,T}\right)^\top \varphi(t)\varphi(t)^\top\left(\underline{\hat{g}} - \underline{g}_{\nu,T}\right) + \Delta_3(T,N)\right] \\
&= \arg\min_{\nu\in D_\nu} \mathrm{Tr}\left[\left(\underline{\hat{g}} - \underline{g}_{\nu,T}\right)^\top R_N(T)\left(\underline{\hat{g}} - \underline{g}_{\nu,T}\right)\right] + \sum_{t=1}^N \mathrm{Tr}\left[\Delta_3(T,N)\right] \\
&\overset{N,T\to\infty}{\approx} \arg\min_{\nu\in D_\nu} \int_{-\pi}^{\pi} \mathrm{Tr}\left[\left(\widehat{G}(e^{j\omega}) - G(e^{j\omega},\nu)\right)^\top S_u(e^{j\omega})\left(\widehat{G}(e^{j\omega}) - G(e^{j\omega},\nu)\right)\right] d\omega
\end{aligned} \tag{6.35}$$

where the last expression exploits the asymptotic value of $R_N(T)$ (see equations (6.21) and (6.22)) and derives from the fact that $\lim_{N\to\infty} \|\Delta_3(T,N)\|_2$ goes exponentially to zero as $T \to \infty$. Comparing equations (6.23) and (6.35), the analogy between the two criteria is clear.

A possible extension of the routine reported in Algorithm 11 exploits the model $\widehat{\mathcal{M}}_\rho^{PEM}$ computed at step 4 as initialization of a further application of PEM on the original data $\mathcal{D}^N$. The final low-order model is chosen among the ones returned by this additional stage. This option has been numerically evaluated but its performances are comparable to the ones achieved through Algorithms 11 and 12. Because of the additional computational effort required by this option, the other two approaches are preferred and analysed in details in Section 6.4.

It is important to observe that the quality of the final low-order system will not only depend on the chosen reduction procedure but also on the initial non-parametric estimate.

For instance, it is worth recalling that the Hankel kernel (3.57) has been designed in order to encourage a small number of Hankel singular values. It is thus to be expected that it will be easier to provide a low McMillan degree approximation of $\hat{\mathbf{g}}$, when this has been estimated using the kernel (3.57). Analogous considerations should hold when estimation is performed using nuclear-norm type penalties on the Hankel singular values (e.g. by using the penalty $\|\mathbf{G}\|_*$). On the other hand, when using only the Stable-Spline kernel illustrated in Section 3.3.1, the Hankel singular values of the estimated systems show a much slower decaying profile. Hence, performing model reduction on these systems possibly leads to neglect some components of the system dynamics.

6.3.1 Choice of the Reduced Order

The preceding sections have remarked several times the criticality represented by the selection of the reduced order, when the knowledge of the high-order system is uncertain (e.g. when it has been estimated from noisy data). This section lists several criteria which could be used to accomplish this task. These techniques will be experimentally compared and analysed in the simulations of Section 6.4.

Statistical Test on Residuals Size. This method evaluates the prediction abilities of the models $\left\{\widehat{\mathcal{M}}_\rho^{PEM}; \rho = 1, ..., T-1\right\}$. Starting from $\rho = 1$, the following steps are repeated:

1. For each output channel $i \in [1, p]$, compute

$$x_{i,\rho} = \frac{1}{\hat{\sigma}_i}\left(\sum_{t=1}^{N}(y_i(t) - \hat{y}_{\rho_i}(t))^2\right) \tag{6.36}$$

   where $\hat{\sigma}_i$ is the estimate of the noise variance on the i-th output channel (obtained through the original model $\widehat{\mathcal{M}}$), while y_i denotes the data related to the i-th output channel (analogously for $\hat{y}_{\rho_i}$).

2. Fix the significance level α and let $F(\mu, \varsigma)$ denote the χ^2 cumulative distribution for a given probability μ and degrees of freedom ς. If

$$x_{i,\rho} \leq F^{-1}(1-\alpha, N-1), \quad \forall i \in [1, p] \tag{6.37}$$

   then choose $\rho^* = \rho$ as the optimal reduced order, otherwise continue to iterate.

If the condition (6.37) is not satisfied by any reduced order $\rho \in [1, T-1]$, ρ^* is set to T. Notice that the χ^2 test (6.37) is based on the assumption that, if $e(t)$ is white Gaussian

noise, the quantities $y_i(t) - \hat{y}_{\rho_i}(t)$ are i.i.d. zero-mean Gaussian random variables with variance $\hat{\sigma}_i$, so that $x_{i,\rho} \sim \hat{\sigma}_i \chi^2(N-1)$. In addition, the significance level α has to be fixed. The result of the previous procedure actually depends on its value, since small values of α tend to favour the selection of lower model orders. However, experimental evidence has shown that the sensitivity of the test to the value of α is low for a quite large range of its values. Further comments on this topic will be given in Section 6.4.
As a final remark, the test in equation (6.37) relying on the statistic $x_{i,\rho}$ corresponds to accepting the smallest model $\widehat{\mathcal{M}}_\rho^{PEM}$ which is not falsified by the observed data under the assumption that noise is Gaussian.
This test will be referred to as χ^2_ε in the plots of Section 6.4.

Statistical Test on Residuals Whiteness. The test (2.227) is applied on $\widehat{\mathcal{M}}_\rho^{PEM}$, $\rho = 1, ..., T-1$ using the original data $\mathcal{D}^N$.
This approach will be referred to as $\chi^2_{\varepsilon\varepsilon}$ in Section 6.4.

Statistical Test on Independence Between Residuals and Past Inputs. The test (2.230) is applied on $\widehat{\mathcal{M}}_\rho^{PEM}$, $\rho = 1, ..., T-1$ using the original data $\mathcal{D}^N$.
Such approach will be denoted as $\chi^2_{\varepsilon u}$ in Section 6.4.

Combination of the Statistical Tests. The above-detailed statistical tests are simultaneously applied. The selected model is either the simplest one which passes all the tests or the simplest one which passes at least one of the tests. In the simulations of Section 6.4, these two criteria will be respectively denoted with the symbols "$\wedge$" and "$\vee$" in between the symbols representing the statistical tests.

AIC. The models $\widehat{\mathcal{M}}_\rho^{PEM}$, $\rho = 1, ..., T-1$ are compared through the AIC criterion (2.216).

BIC. The models $\widehat{\mathcal{M}}_\rho^{PEM}$, $\rho = 1, ..., T-1$ are compared through the BIC criterion (2.218).

Bootstrap. A FIR model of length T is estimated through the original data $\mathcal{D}^N$. This is used to generate $B = 20$ bootstrap datasets, as detailed in equations (2.219). The procedure detailed in Algorithm 11 is repeated for each of these datasets, obtaining the models $\widehat{\mathcal{M}}_{\rho,b}^{PEM}$, $\rho = 1, ..., T-1$, $b = 1, ..., B$. For each order ρ, these models are used to compute the covariance penalty criterion as detailed in equation (2.220). The model order giving the lowest value of this criterion is finally chosen. It should be stressed

that this procedure is particularly involved from a computational point of view, since it requires the estimation of several models.
In the plots of Section 6.4 such method will be denoted as BT.

Bayesian Posterior. The posterior distribution returned by the Bayesian identification procedure is evaluated on the impulse response of the models $\left\{\widehat{\mathcal{M}}_{\rho}^{PEM};\ \rho = 1, ..., T-1\right\}$. The one giving the largest posterior value is selected. It should be observed that this criterion strongly depends on the kernel adopted in the Bayesian identification, as will be confirmed by the numerical results of Section 6.4, where this technique will be referred to as POS.

Hankel Marginal Likelihood. The Marginal Likelihood corresponding to kernel $\bar{K}_{H,\zeta}$ in equation (3.57) is computed for each model in the set $\left\{\widehat{\mathcal{M}}_{\rho}^{PEM};\ \rho = 1, ..., T-1\right\}$. Specifically, for each ρ, the matrix $\widehat{Q}(\zeta)$ appearing in the kernel is given by

$$\widehat{Q}(\zeta) := \widehat{U}\ \text{blockdiag}(\hat{\lambda}_1 I_{\rho},\ \hat{\lambda}_2 I_{pr-\rho})\ \widehat{U}^{\top} \tag{6.38}$$

where $\widehat{U}$ contains the Hankel singular values of $\widehat{\mathcal{M}}_{\rho}^{PEM}$ and the scaling factors $\hat{\lambda}_1$ and $\hat{\lambda}_2$ are estimated by solving the optimization problem (3.69). The model $\widehat{\mathcal{M}}_{\rho^*}^{PEM}$ returning the largest marginal likelihood value is finally chosen.
This model selection technique (referred to as HANK ML in the following experimental section) has also been tested by replacing the kernel $\bar{K}_{H,\zeta}$ with $\bar{K}_{SH,\eta}$, defined in equation (3.60); since the observed performance are slightly worse than those obtained with $\bar{K}_{H,\zeta}$, they are omitted in Section 6.4.

6.4 Numerical Results

The two model reduction routines detailed in Algorithms 11 and 12 equipped with the order selection criteria listed in Section 6.3.1 are here evaluated by means of some Monte-Carlo studies.

6.4.1 Data

The Monte-Carlo simulations here reported are conducted on four scenarios, each of them consisting of $N_{MC} = 200$ runs. The data belonging to the four scenarios are affected by a zero-mean white Gaussian noise $e(t)$ with a standard deviation chosen in order to obtain different values for the SNR, according to the specific scenario. These choices will be clarified in the brief illustration which follows.

S0: For each Monte-Carlo run the transfer function $G(q)$ is generated as

$$G(q) := \frac{q+0.99}{q} \sum_{i=1}^{N_r} K_i \frac{(q+0.9)}{(q-p_i)(q-p_i^*)} + G_{N_r+1}(q)$$

where $G_{N_r+1}(q)$ is a random 4-th order transfer function generated by the MATLAB routine `drmodel` (see remark 3.5.1 for further details on this routine), with the constraint that its poles are inside the disk of radius 0.95. The parameters N_r, p_i, K_i for each independent Monte-Carlo run are generated as follows: $N_r \sim \mathcal{U}[3,5]$, $K_i \sim \mathcal{U}[2,10]$, $p_i = \varsigma_i e^{j[\phi_0 + \frac{\pi-\phi_0}{N_r}(i-1)]}$, $\varsigma_i \sim \mathcal{U}[0.9, 0.99]$, $\phi_0 \sim \mathcal{U}[0, \pi/2]$. The Gaussian input $u(t)$ is generated (independently for each run) by the MATLAB function `idinput` with normalized band 0.9.
The SNR on the output channel is a uniform random variable in the interval $[1,4]$ and $N = 500$ input-output data pairs are available for each system.

S1: This scenario was already considered in Section 3.5. To help the reader, its description is also reported here. A fixed fourth order system with transfer function $G(q) = C(qI - A)^{-1}B$ is considered, where

$$\begin{aligned} A &= \text{blockdiag}\left(\begin{bmatrix} 0.8 & 0.5 \\ -0.5 & 0.8 \end{bmatrix}, \begin{bmatrix} 0.2 & 0.9 \\ -0.9 & 0.2 \end{bmatrix} \right) \\ B &= [1\ 0\ 2\ 0]^\top \quad C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0.1 & 0 & 0.1 \\ 20 & 0 & 2.5 & 0 \end{bmatrix} \end{aligned} \tag{6.39}$$

The input is generated, for each Monte Carlo run, as a low-pass filtered white Gaussian noise with normalized band $[0, \varrho]$ where ϱ is a uniform random variable in the interval $[0.8, 1]$. The SNR on the output signal is a uniform random variable in the interval $[1,4]$. For each system $N = 500$ input-output data pairs are available.

D2: This data-bank is exploited in the paper Chen et al. (2014) and it consists of 30-th order random SISO discrete-time systems having all poles inside a circle of radius 0.95. The systems are simulated with a unit variance white Gaussian noise. The SNR on the output signal is equal to 1, while the number N of available input-output data pairs is 210.

D4: This data-bank is also used in the paper Chen et al. (2014) and was previously exploited for the numerical experiments conducted in Chapter 4. This scenario contains the same systems appearing in D2 but they are simulated with unit variance band-limited Gaussian signal with normalized band $[0, 0.8]$. Furthermore, each dataset contains $N = 500$ data.

6.4.2 Identification Algorithms

The model reduction procedures outlined in Algorithms 11 and 12 are applied on the estimates returned by the following identification algorithms:

SS: The estimator (2.148) where $\bar{K}_\eta$ is chosen to be the TC kernel (3.27) and the hyper-parameters η are estimated through marginal likelihood maximization. The estimator is computed through the MATLAB routine `arxRegul` (imposing a FIR model structure).

SH: The estimator returned by Algorithm 5 in Chapter 3 with $\bar{K}_{S,\nu}$ specified through the TC kernel.

NN: A FIR model of order T estimated solving

$$\hat{\mathbf{g}} = \underset{\mathbf{g} \in \mathbb{R}^{pmT}}{\arg\min} \|Y_N - \Phi_N \mathbf{g}\|^2 + \lambda^* \|\mathbf{G}\|_* \tag{6.40}$$

The optimization problem is solved through a tailored ADMM algorithm (as in Liu et al. (2013)), while λ^* is determined through Cross-Validation. This procedure has also been tested by replacing $\mathbf{G}$ in (6.40) with its weighted version $\widetilde{\mathbf{G}}$ (see (3.40)).

RNN: A FIR model of order T estimated by iteratively solving

$$\hat{\mathbf{g}} = \underset{\mathbf{g} \in \mathbb{R}^{pmT}}{\arg\min} \|Y_N - \Phi_N \mathbf{g}\|^2 + \lambda^* \|W_l \mathbf{G} W_r\|_* \tag{6.41}$$

The weight matrices W_l and W_r are updated at each iteration according to the procedure suggested by Mohan and Fazel (2010). λ^* is selected through Cross-Validation. The case in which $\mathbf{G}$ in (6.41) is replaced with $\widetilde{\mathbf{G}}$ has also been tested.

The two identification techniques relying on nuclear norm regularization could also benefit from the model reduction procedures detailed in Section 6.2.3. Indeed, the nuclear norm penalty is used to enforce a low McMillan degree on the unstructured estimate $\hat{\mathbf{g}}$, thus facilitating the recovery of a low-order structured model in an eventual post-processing stage. The practical validity of these considerations is therefore evaluated in the simulations which follow.

The above-listed algorithms are implemented setting the length T of the estimated impulse response $\hat{\mathbf{g}}$ equal to 80 for scenarios S1 and D2, to 200 for S2 and to 100 for D4.

As a comparison with parametric techniques, which return a model with a well-defined order, PEM equipped with an oracle is considered (denoted as PEM+OR in the following). Specifically, PEM+OR represents PEM as implemented by the MATLAB routine `pem` with an oracle which selects the order giving the highest fit to the true impulse response. The fit is measured according to formula (3.91) defined in Chapter 3, that is

$$\mathcal{F}_{N_c}(\hat{\mathbf{g}}) := \frac{1}{pm} \sum_{i=1}^{p} \sum_{j=1}^{m} \text{cod}\left(\left[g_0^{N_c} \right]_{ij}, \hat{g}_{ij}^{N_c} \right) \tag{6.42}$$

with $N_c = 1000$.
The results achieved by PEM equipped with BIC criterion for the complexity selection will be also reported; this method will be referred to as PEM+BIC.

6.4.2.1 Details on the implementation of Algorithms 11 and 12

The model reduction required by step 2 of Algorithms 11 and 12 is performed either by the balanced truncation detailed in Section 6.1.1 or by Algorithm 10. Since the latter methodology leads to slightly better performances, the results achieved by means of the balanced approximation are omitted.

The re-estimation at step 4 of Algorithm 11 (and step 3 of Algorithm 12) is performed using the MATLAB routine `pem`.

Several combinations of the statistical tests listed in Section 6.3.1 have been tested, observing more robust results w.r.t. the application of a single test. According to the performed simulations, the best results are achieved combining the test on residuals size and on that the independence between residuals and past inputs. Consequently, only their performance will be reported in the following plots.
The application of the statistical tests requires to the user to fix a certain significance level. According to the performed numerical tests, this choice does not appear straightforward and it should depend on the unknown system: if this is known to be particularly complex a high significance level is suggested (e.g. $\alpha = 0.4$), in order to avoid possible undermodelling issues; on the other hand, in presence of simple systems, a low value for α is more suited (e.g. $\alpha = 0.05$), thus preventing the risk of overfitting.
Finally, the lag $\bar{\tau}$ used in the tests on the whiteness of the residuals and on their independence from past inputs is set to 25.

6.4.3 Impulse Response Estimates and Selected Low-Orders

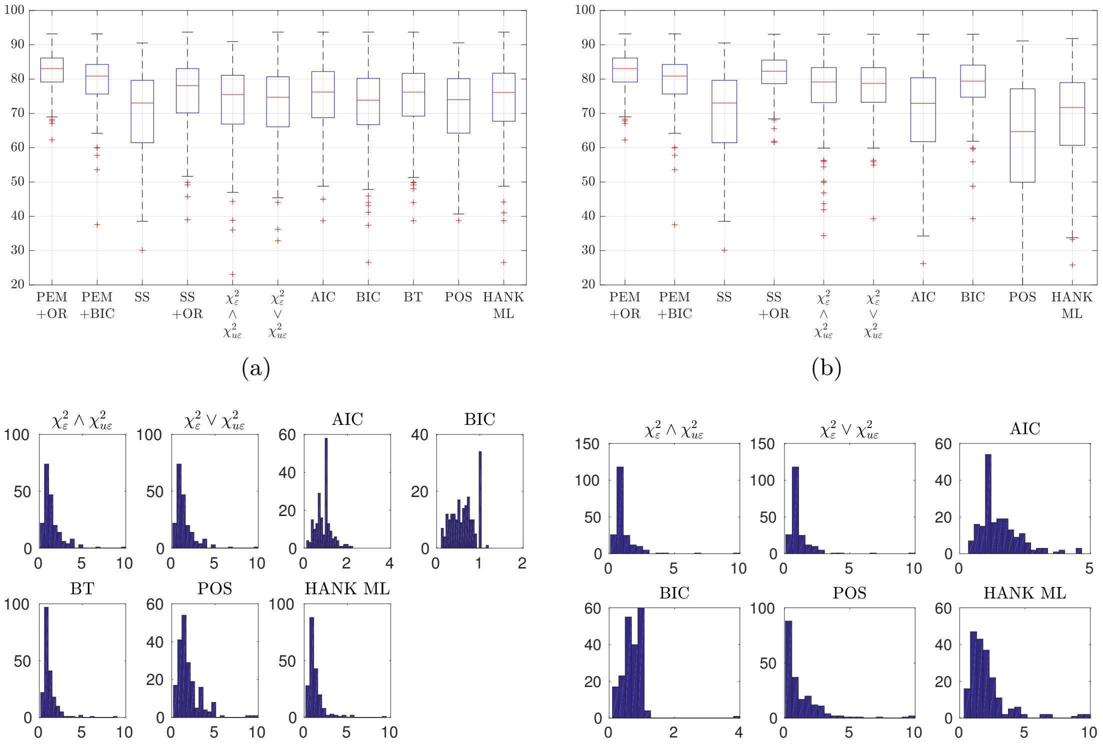
The quality of the impulse response estimates is measured according to the criterion (6.42). The corresponding boxplots obtained in the Monte-Carlo scenarios of Section 6.4.1 are reported in the following pages.

The first two columns in each plot contain the performance of PEM, respectively equipped with an oracle for the order selection (this is an unrealistic estimator which represents the upper bound achievable by PEM) and with the BIC criterion. The third column of the plots reports the results obtained by one of the estimators listed in Section 6.4.2, while the fourth column shows the largest fit achievable after the reduction of the unstructured estimate to a low-order model (again, such estimator is not realizable in practice, but it serves as an upper bound for the considered performance). Finally, the right-most columns of the plots show the fit obtained after performing the model reduction procedure with the reduced order chosen according to one of the criteria in Section 6.3.1.

The second row of the following figures reports the histograms of the ratios between the reduced order selected by the Oracle routines (i.e. SS+OR, SH+OR, NN+OR and RNN+OR) and the one chosen by the realistic model selection criteria.

Figures 6.1 and 6.2 show the results achieved by the non-parametric Bayesian estimator respectively equipped with the TC kernel (reported in equation (3.27)) and with the so-called “stable-Hankel” kernel defined in equation (3.60). It can be noticed that the performance of the non-parametric estimates are improved by means of both the reduction procedures detailed in Algorithms 11 and 12. While the latter could ideally achieve better performance (according to oracle’s results), the order selection procedure appears more robust when the first reduction procedure is adopted. The comparison of the different criteria listed in Section 6.3.1 shows that the BIC criterion performs better when jointly applied with Algorithm 12, that is, when the PEM estimation is performed on the original data $\mathcal{D}^N$. Inspecting Figure 6.2(b) and in particular the columns PEM+BIC and BIC, it should be noticed how the initialization of PEM with the Bayesian estimator improves the results achieved by the standard MATLAB routine `pem` (which is initialized by means of a subspace estimate).

W.r.t. BIC, opposite performance is observed for the AIC criterion, which tends to select more complex models w.r.t. BIC, thus being penalized when the noisy data $\mathcal{D}^N$ are used for estimation. Bootstrap achieves very robust results but its use is penalized by the significant computational effort it requires. As it could be expected, the criterion relying on the posterior distribution returned by the Bayesian estimator appears strongly influenced by its performance. Differently, the criterion based on the marginal likelihood of kernel (3.57) leads to good performance when jointly applied with Algorithm 11, while



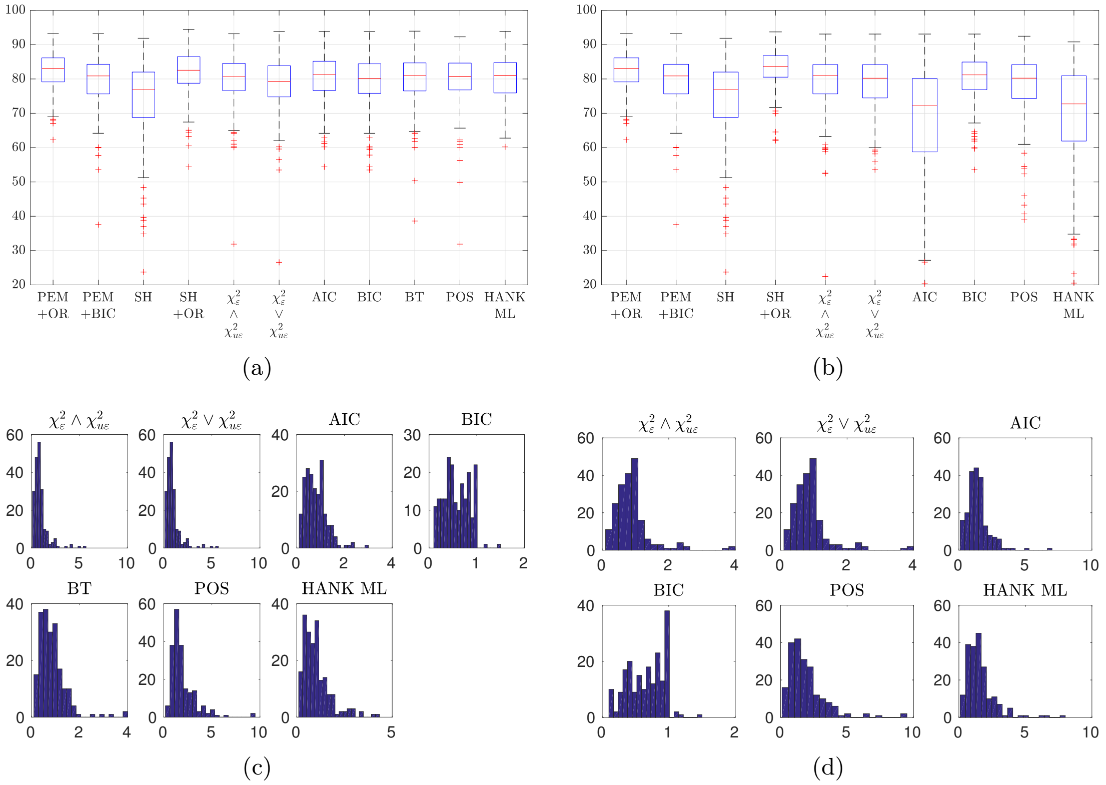
(a)

(b)

(c)

(d)

Figure 6.1: Scenario S0 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator SS (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator SS, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by SS+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.



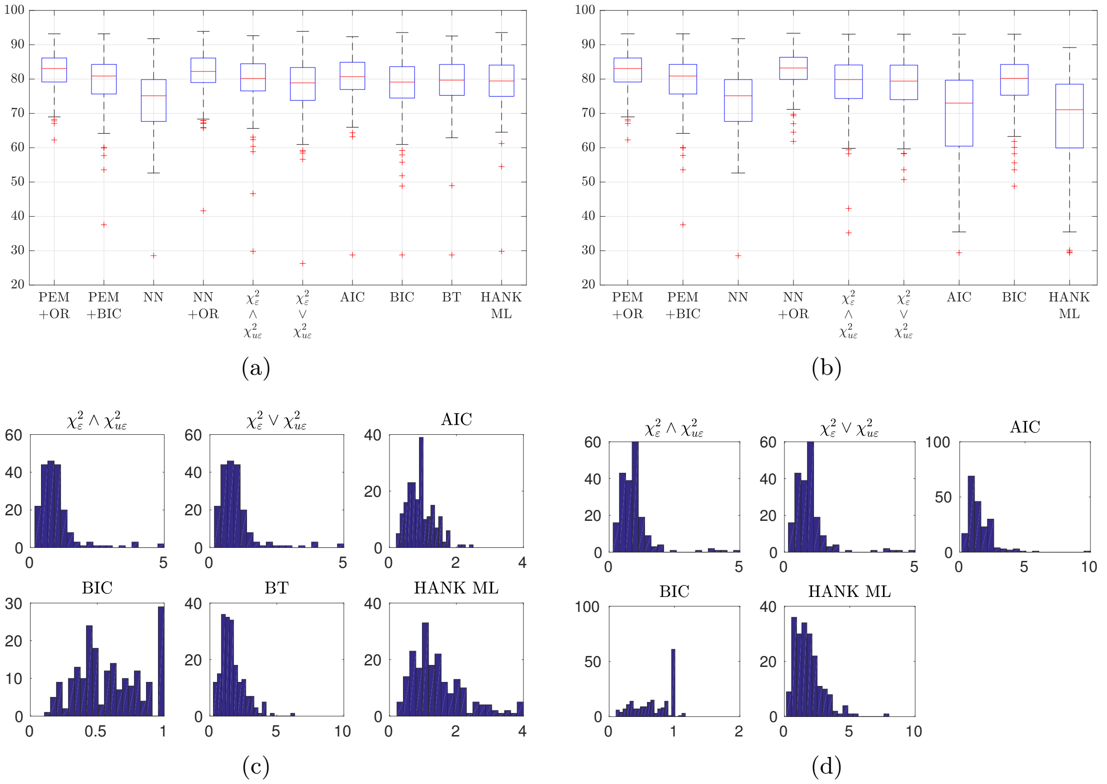
(a)

(b)

(c)

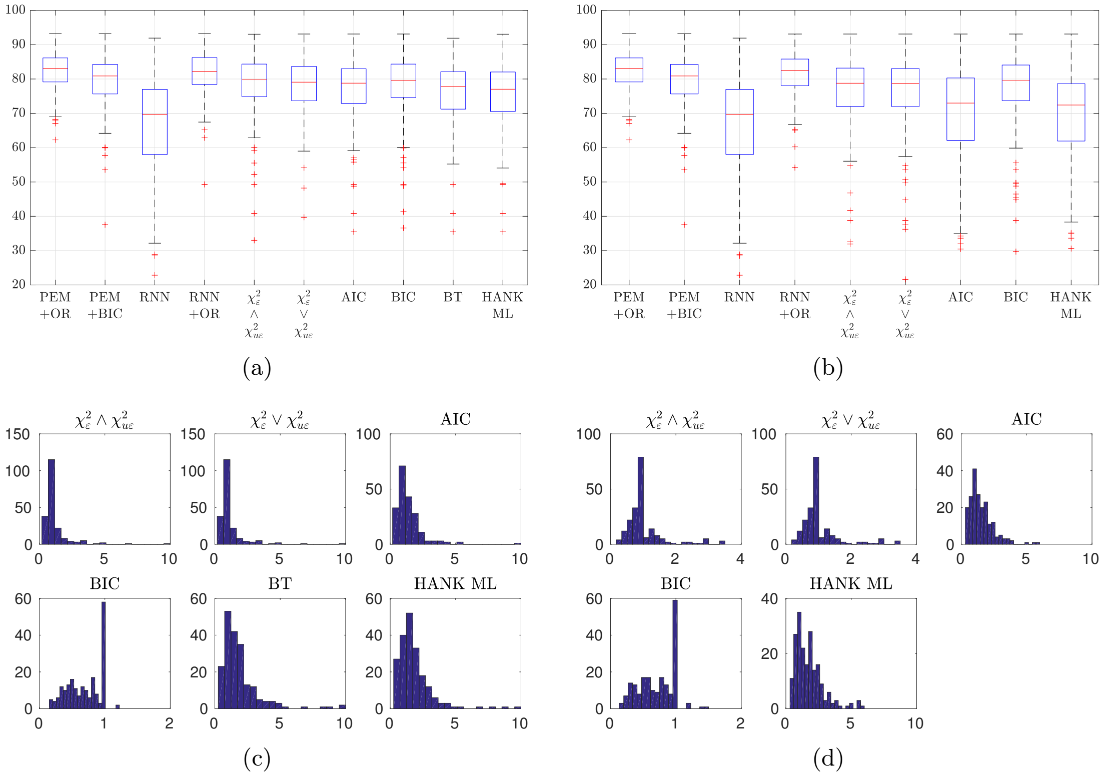
(d)

Figure 6.2: Scenario S0 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator SH (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator SH, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by SH+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.



(a) (b) (c) (d)

Figure 6.3: Scenario S0 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator NN (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator NN, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by NN+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.



(a)

(b)

(c)

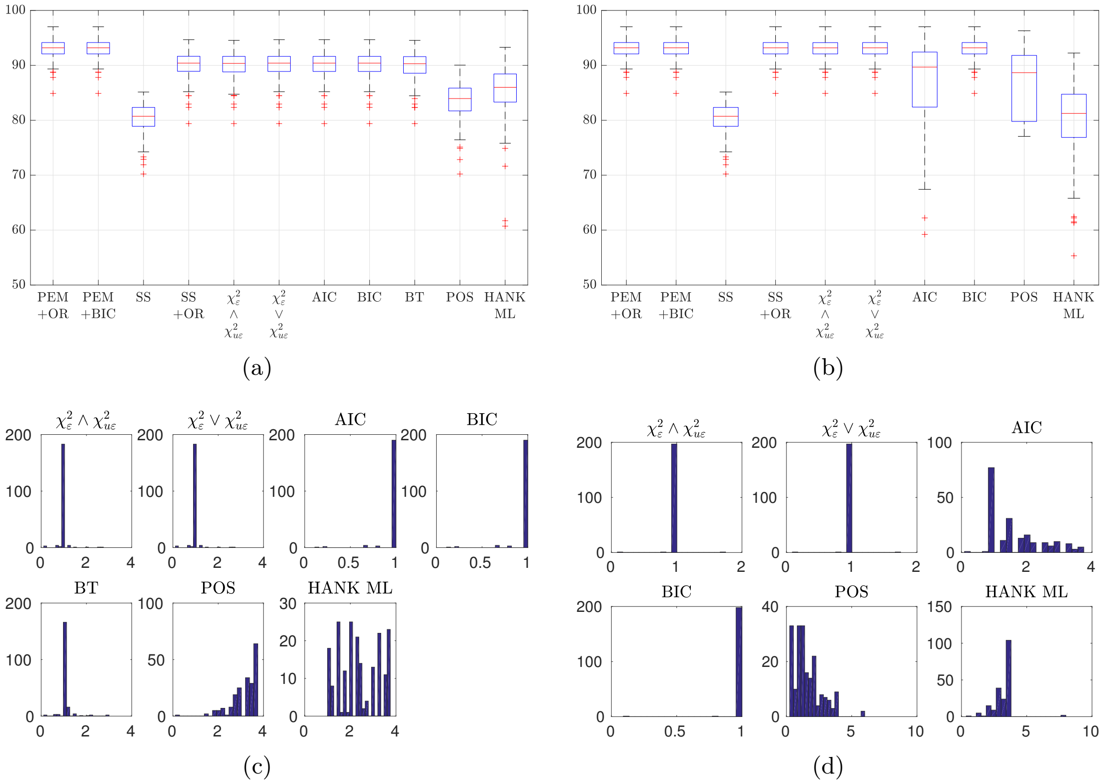
(d)

Figure 6.4: Scenario S0 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator RNN (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator RNN, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by RNN+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.

is less effective when Algorithm 12 is used for model reduction. The reasons of this behaviour are analogous to the ones above-mentioned in relation to AIC: such criterion tends to select complex models, which are suited when the estimation data are non-noisy (as is the case of Algorithm 11) but they are not advised in presence of noisy data. Finally, the statistical tests on the residuals seem a robust complexity selection method but, as observed in Section 6.4.2.1, their effectiveness is highly influenced by the value of the significance level α.

Figures 6.3 and 6.4 report the results observed when Algorithms 11 (left plot) and 12 (right plot) are applied on the two identification methods exploiting nuclear norm regularization. What observed for Bayesian methods is here confirmed: Algorithm 11 seems to make the order selection stage easier, since the compared criteria lead to comparable performance. Nonetheless, a gap between their fit and the optimal one achieved by the oracle estimators after reduction (fourth column in each plot) is still noticeable.
In this case, the performance of the reduced models is little influenced by those of the original unstructured model. The reader could pose the attention e.g. on Figure 6.4: the fits achieved after model reduction appear comparable to those observed in Figure 6.3, despite the very poor performance of the estimates returned by RNN. This behaviour contrasts with that previously observed with the Bayesian estimators: the unsatisfying performance of SS also impact the effectiveness of the subsequent reduction procedure. The reason for this phenomenon probably lies in the profile of the estimated system Hankel singular values: while those obtained through the use of the stable-spline kernel (SS) show a slowly decaying profile, the singular values returned by nuclear norm regularization methods typically present a clear gap between those associated with the system dynamics and those related to the noise realization in the data. This type of profile makes easier the subsequent detection of a low-order approximation to the estimated high-order FIR model.

Figures 6.5 and 6.6 refer to scenario S1 and report the results achieved after the application of model reduction on the Bayesian estimates denoted with SS and SH. Since the four Hankel singular values give equal contribution to the system dynamics, the detection of the right system complexity appears easier. This observation is confirmed by the results observed in Figures 6.5, 6.6, where the true order is detected by almost all the tested criteria. The only exceptions are AIC and HANK ML when applied together with Algorithm 12. This behaviour confirms what already observed in scenario S0. The strong dependence on the original Bayes estimator of the criterion based on its posterior distribution is detrimental when the performance of the Bayesian estimator are not satisfying, as clearly noticeable in Figure 6.5.



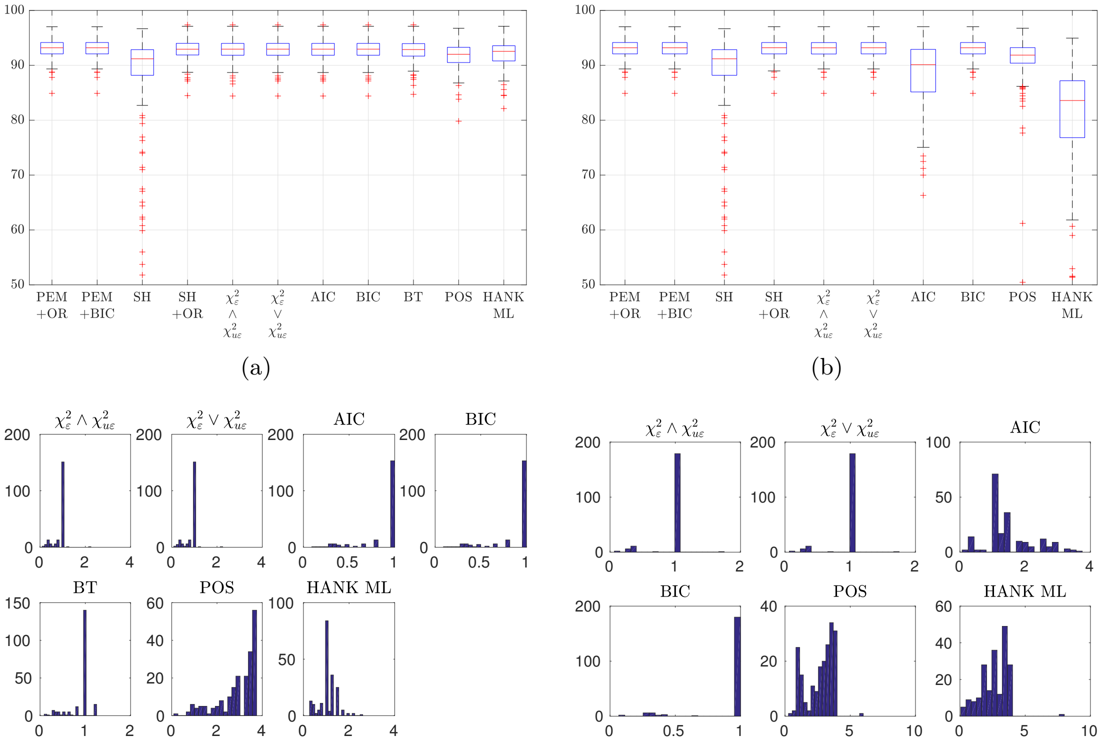
(a)

(b)

(c)

(d)

Figure 6.5: Scenario S1 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator SS (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator SS, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by SS+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.



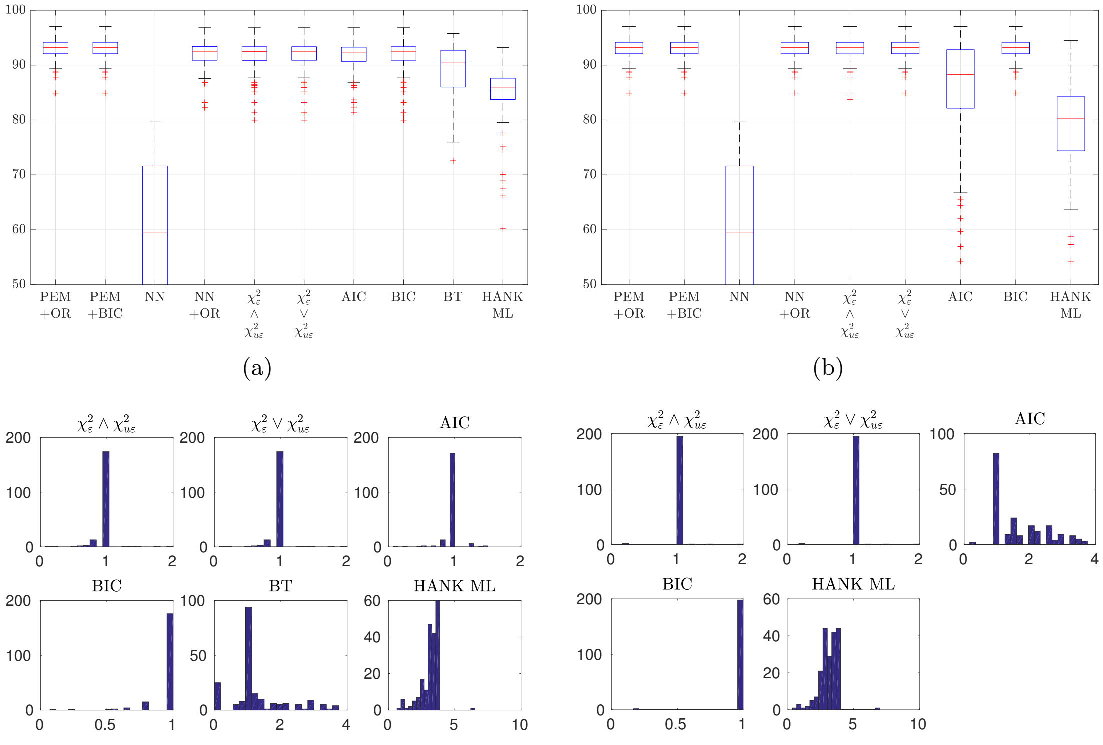
(a)

(b)

(c)

(d)

Figure 6.6: Scenario S1 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator SH (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator SH, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by SH+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.



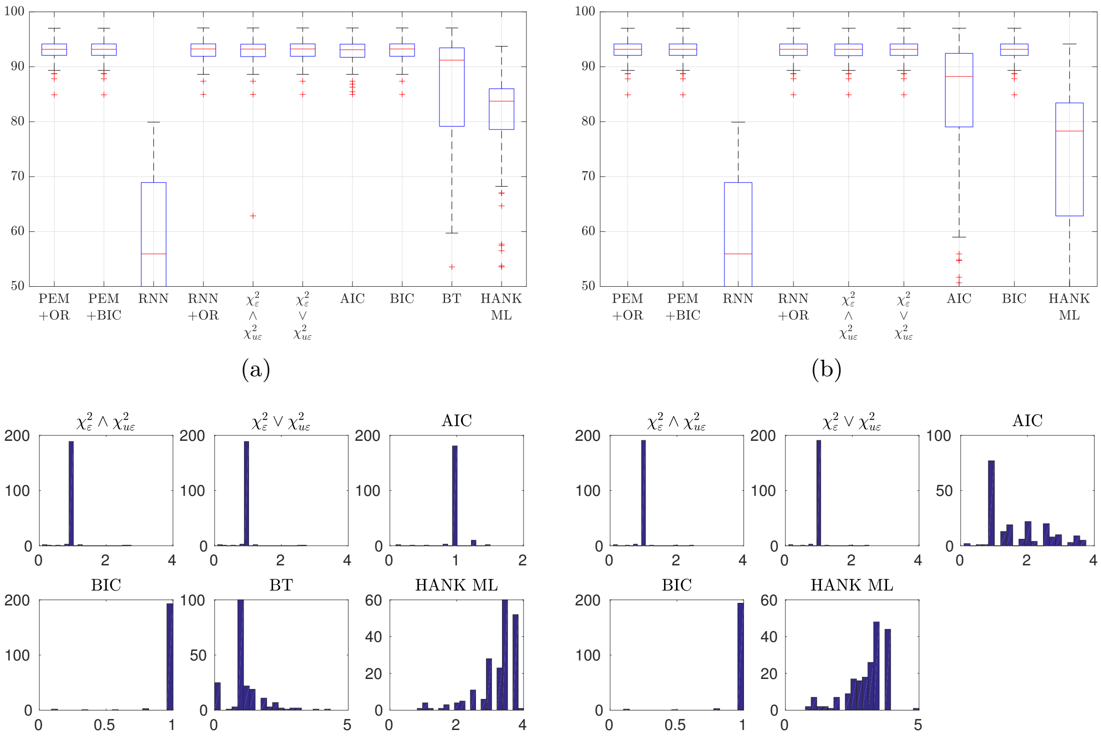
(a)

(b)

(c)

(d)

Figure 6.7: Scenario S1 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator NN (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator NN, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by NN+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.



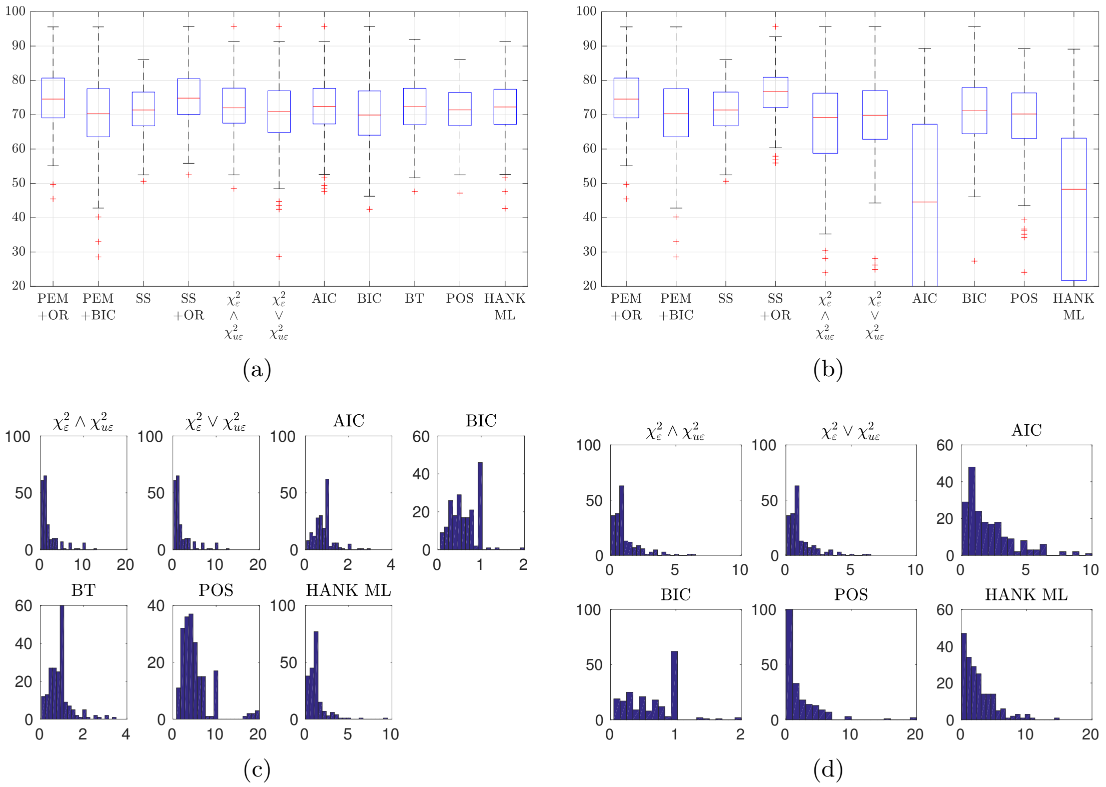
(a)

(b)

(c)

(d)

Figure 6.8: Scenario S1 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator RNN (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator RNN, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by RNN+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.



(a) (b)

(c) (d)

Figure 6.9: Scenario D2 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator SS (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator SS, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by SS+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.

Figures 6.7 and 6.8 refer to the application of Algorithms 11 and 12 on the high-order FIR models estimated by means of PEM equipped with nuclear norm regularization (see Section 6.4.2). Despite the unsatisfying performance of these estimators in scenario S1, the application of a model reduction procedure allows to recover a good adherence to the true unknown impulse response. With regard to the various complexity selection criteria, the comments written in relation to the previous plots still hold.

Figures 6.9 and 6.10 show the performance obtained in scenario D2, starting from the Bayesian estimates SS and SH, respectively. Differently from the previous two scenarios, here the application of a reduction procedure on the non-parametric estimate does not improve its performance and could sometimes worsen them. In particular, this event happens more often when Algorithm 12 is applied (see Figures 6.9(b) and 6.10(b)). The

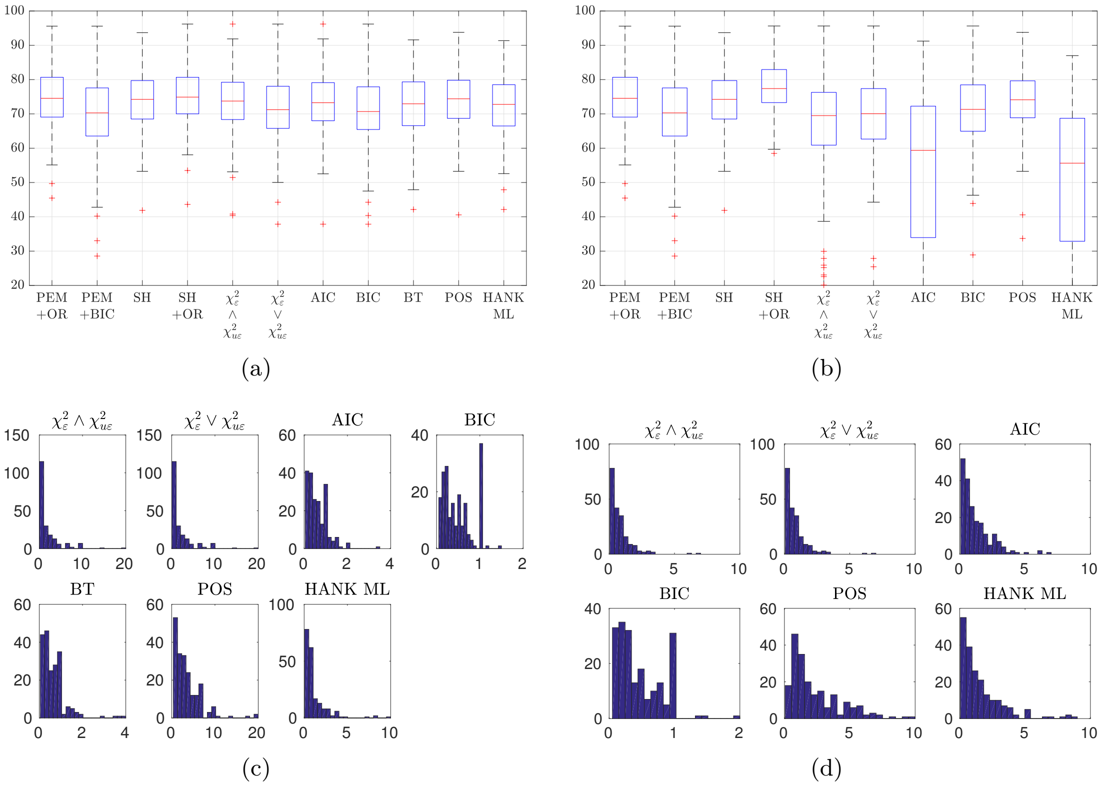
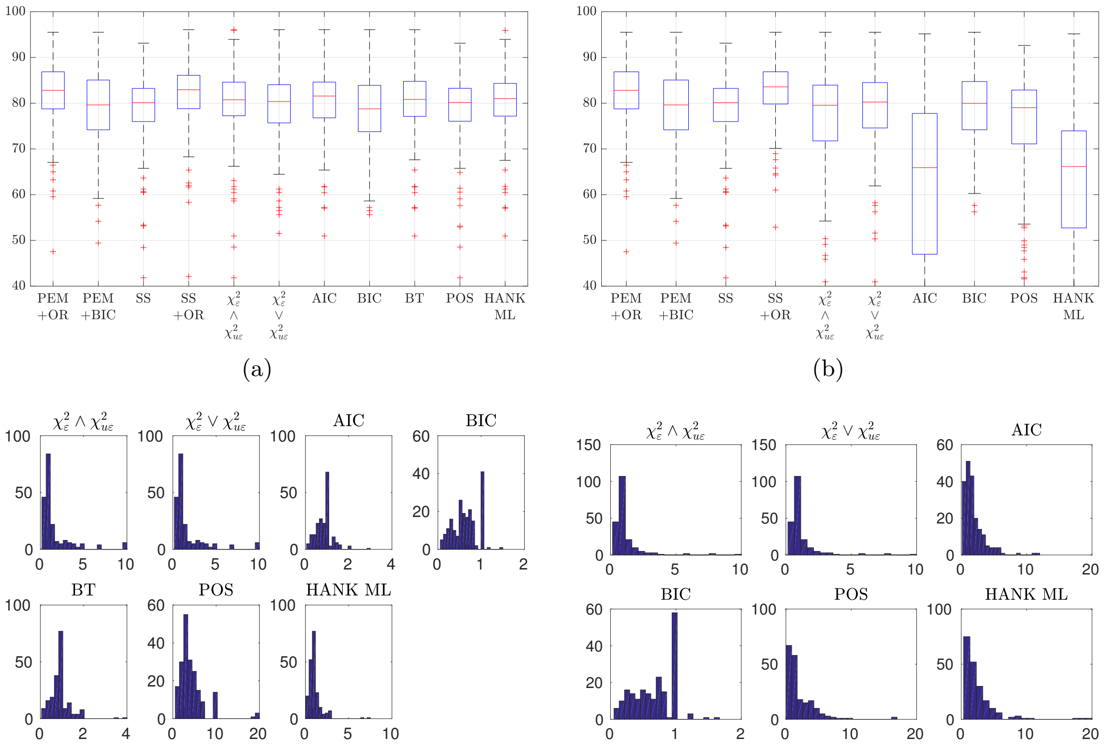


Figure 6.10: Scenario D2 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator SH (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator SH, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by SH+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.



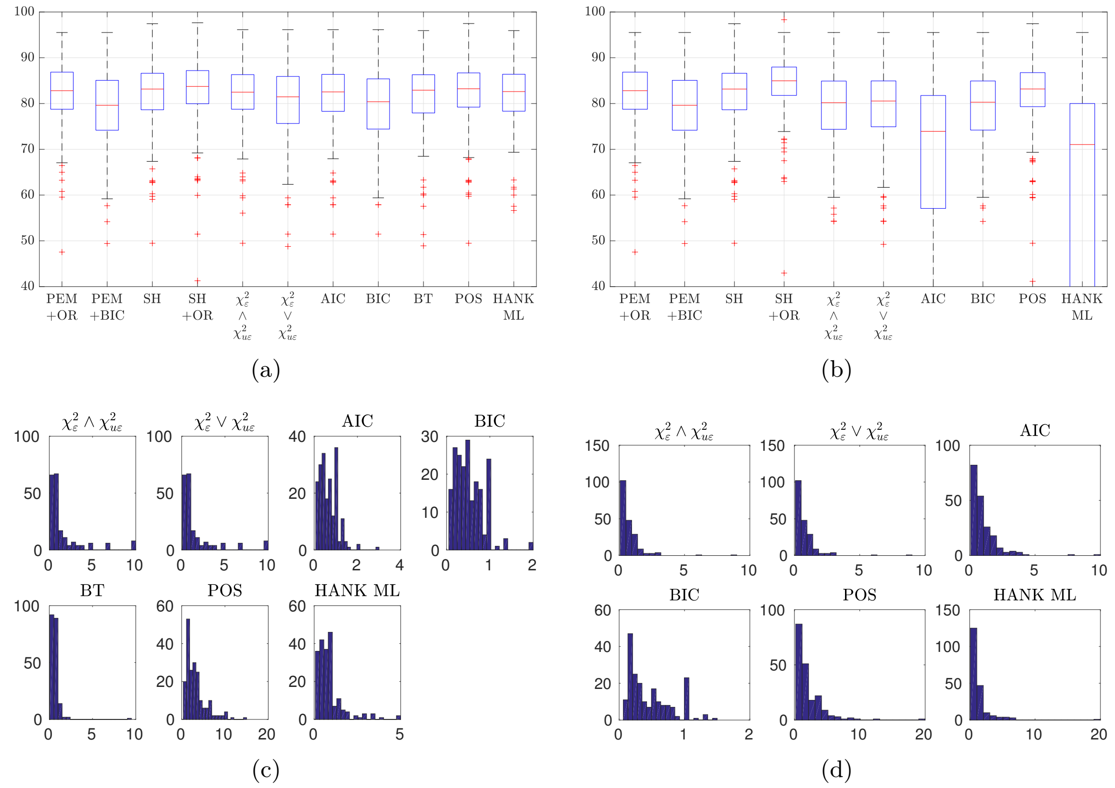
(a) (b) (c) (d)

Figure 6.11: Scenario D4 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator SS (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator SS, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by SS+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.

crucial step seems to be the order selection, since the performance achieved by the "oracle" (fourth column of the boxplots) are satisfying, but they are not approached by the realistic criteria here evaluated for complexity selection. Inspecting the results in Figures 6.9(a) and 6.10(a), the AIC criterion, the bootstrap technique and the criterion relying on the marginal likelihood arising from kernel $\bar{K}_{H,\eta}$ appear to be the most robust ones.

Similar considerations hold for the results observed in Scenario D4 (reported in Figures 6.11 and 6.12).

The numerical experiments previously reported have highlighted how the problem investigated in this chapter is not trivial. In particular, the classical model selection criteria adopted in the context of parametric methods do not seem to be robust enough to



(a)

(b)

(c)

(d)

Figure 6.12: Scenario D4 - *Top:* Impulse response fit (6.42) achieved by reducing the FIR model returned by estimator SH (see Section 6.4.2). In each boxplot, the third column reports the fit achieved by estimator SH, the fourth column contains the optimal fit achieved after reduction (i.e. using the optimal choice of the reduced order) and the right-most columns show the fit obtained using the order selection criteria listed in Section 6.3.1. *Bottom:* Histograms of the ratio between the reduced orders selected by SH+OR and by the other realistic criteria. (a),(c): Reduction is performed by means of Algorithm 11; (b),(d): Reduction is performed using Algorithm 12.

guarantee (or to improve) the performance of the original Bayesian estimate. The author believes that further investigations on this topic should be conducted; a future research direction could also include so-called goal-oriented model reduction techniques.

7
Conclusions and Future Work

The thesis has presented some extensions of a non-parametric Bayesian method which has been recently introduced to tackle the system identification problem. Such techniques have had the merit of importing classical machine learning tools into the system identification community. Specifically, these new identification procedures mainly resort on ideas coming from Gaussian Processes Regression and from the theory of Reproducing Kernel Hilbert Spaces, which provides a regularization framework where non-parametric regression is possible. Compared to the standard machine learning setup, where the given data are assumed to i.i.d. according to unknown distribution, the data used by system identification routines are temporally correlated. Consequently, the importation of standard machine learning tools into the system identification framework has required to account for such correlation. This has been done mainly through the development of suitable prior distributions or, equivalently, regularization function.

While providing an overview of these new approaches, the thesis has attempted to draw an extensive picture of the system identification field. Classical techniques, such as Prediction Error Methods and subspace algorithms have been extensively reviewed, while highlighting several connections between them and the recently introduced non-parametric approaches. Chapter 2 has described these three main families of routines appearing in the system identification literature. Theoretical properties, as well as implementation details have been described: particular attention has been devoted to the choices that the user has to take when applying them and to specific computational aspects. Model selection and model validation have also been discussed, trying to provide an overview on the way in which the different identification procedures deal with them.

The remaining chapters of the manuscript have presented the innovative results achieved during the author's research activity on non-parametric Bayesian methods for system identification. The illustration has intended to connect these new contributions to already existing results regarding parametric Prediction Error Methods and subspace algorithms. To this purpose, the initial part of each chapter has been devoted to a summary of already derived theoretical properties or methodologies.

Chapter 3 has dealt with the problem of prior design or equivalently, of the shaping of a suitable regularization function. Exploiting the well-known connection between regularization and Bayesian inference under Gaussian assumptions, the role played by regularization in system identification has been investigated. The main examples of the application of ℓ_2- and ℓ_1-type penalties in identification procedures have been illustrated. Specifically, the attention has been devoted to regularization inducing stability and low-complexity of the estimated system. Drawing inspiration from recently proposed regularization techniques, a new prior for non-parametric Bayesian system identification

has been derived using Maximum Entropy arguments under stability and complexity constraints. The new prior combines the classical stable-spline kernel with a term controlling the rank of the block Hankel matrix built with the Markov coefficients. This specific structure allows to enforce both stability and low complexity (measured in terms of McMillan degree) of the estimated system. A specific algorithm has been designed to solve the identification problem. It iteratively refines the impulse response estimate by updating the hyper-parameters defining the prior and in turn by refining the estimated signal subspace, i.e. the subspace spanned by the non-zero Hankel singular values. A tailored Scaled Gradient Projection algorithm has been designed in order to reduce the computational effort required by the algorithm: numerical simulations have proved the significant computational time savings brought by the proposed gradient method w.r.t. off-the-shelf algorithms. The newly proposed identification procedure has been compared with already existing ones through an extensive numerical study. The reported results clearly prove the effectiveness of the new approach. In particular, when MIMO systems have to be identified, the Hankel-based method appears more effective than the original regularization/Bayesian technique relying only on the sole stable-spline kernel. When compared with other methods which include a Hankel-type penalty, it provides comparable performance on randomly generated "large" MIMO systems, while it appears preferable on a fourth order "mildly-resonant" system. Finally, compared to traditional methods, such as PEM an subspace algorithms, the new routine provides more accurate estimates, especially in presence of a small identification dataset.
Future work will include the design of a more efficient numerical implementation, as well the extension to the identification of ARMAX models. Furthermore, a deeper statistical analysis of this approach deserves to be conducted.

Chapter 4 has been focused on the statistical properties of the estimators returned by the three main algorithms considered in the thesis. In particular, the consistency, as well as the (asymptotic) distribution of the returned estimates have been analysed. Specific attention has been reserved to the so-called confidence intervals, i.e. to the uncertainty sets that are built around the estimates. The novel contribution presented in the chapter is the development of a framework through which the confidence sets returned by parametric PEM and by non-parametric Bayesian techniques are compared. The different nature of these two sets has been highlighted: first, the confidence sets returned by PEM are finite-sample approximations of their asymptotic counterpart, while Bayesian "credible" sets are precise even in finite-sample cases; second, PEM's uncertainty sets lie in the parameter space, while non-parametric ones lie in the impulse response space; third, when adopting the Full Bayes approach and hence resorting to sampling-based

techniques, the confidence sets returned by Bayesian methods consist of sampled points, which need to be somehow compared with the dense sets provided by PEM. The proposed comparative framework converts the parametric confidence sets into "particle" sets lying in the impulse response space; analogously, the ellipsoidal set returned by non-parametric Bayesian methods relying on the Empirical Bayes approach is converted into a "particle" set. The numerical comparative study has shown that the Bayesian estimators and their confidence sets are competitive even with the parametric methods equipped with an oracle which has the unrealistic knowledge of the true impulse response.
A further contribution reported in Chapter 4 is the numerical comparison between Empirical Bayes and Full Bayes approaches, which provide two different approximations to the analytical intractability of the stated Bayesian inference problem. The preliminary results here reported do not show a significant performance gap between the estimators returned by the two techniques; however, Empirical Bayes approaches have a clear advantage in terms of computational complexity. A deeper comparison of these two methodologies will be subject of further research.

Chapter 5 has considered the problem of real-time identification, where a current estimate needs to be updated as soon as new data become available. As observed in the chapter, these techniques play an important role in practical contexts, since they constitute the basis for the design of adaptive controllers or for fault detection. In addition, these methods allow to track (slowly) time-varying systems. A brief overview of the existing real-time parametric identification procedures has been given: they all rely on recursive formulations of the original batch algorithm. Specifically, key ingredients for these methods are the modest amount of computations and of memory storage that they require. The innovative contribution of the chapter is the reformulation of the non-parametric Bayesian routines as real-time algorithms. The key ingredients in this case are the recursive updates of the data-dependent matrices and the approximative resolution of the marginal likelihood optimization problem which arise when the Empirical Bayes approach is followed. Specifically, only one iteration of a chosen iterative routine is performed. Both gradient methods and the Expectation Maximization (EM) algorithm have been compared. The numerical study has shown the effectiveness of this real-time implementation, when applied for the identification of both time-invariant and time-varying systems. In addition, the computational advantages of this procedure w.r.t. the batch counterpart have been proved. The author believes that the preliminary investigation here performed may pave the way for further research in this topic. For instance, a future research direction could consider the recursive update of the Bayesian estimate, resembling the one which is available for parametric techniques.

Chapter 6 has considered a post-processing model reduction stage, which is required when the estimate returned by a non-parametric Bayesian method has to be adopted for practical purposes, such as filtering or controller design. As thorough discussed in the thesis, non-parametric Bayesian techniques return a high-order FIR model, which is not suitable for practical uses. This issue has not been properly investigated in the recent system identification literature. Consequently, a model reduction procedure has been here proposed, which is fed with a high-order FIR model estimated through a non-parametric algorithm and returns a lower-order model. A crucial step of this procedure is the choice of the order of this reduced model: classical and tailored complexity selection techniques have been experimentally compared. The achieved results are dependent on the quality of the estimated non-parametric model. Overall, from the conducted numerical study, it seems difficult to extrapolate a sound procedure which guarantees robust results in a wide range of scenarios. However, it is fair to say that when the proposed model reduction routine is equipped with a suitable model selection criterion, it returns performances which are comparable with (or even better than) those of the original non-parametric Bayesian estimators. According to the author's opinion, this topic should deserve further research in the future, starting e.g. from the investigation of goal-oriented reduction procedures.

A
Reproducing Kernel Hilbert Spaces

This Appendix intends to provide the reader with the basic concepts concerning the theory of Reproducing Kernel Hilbert Spaces (RKHS).
Some definitions are first provided.

Definition A.0.1 (Hilbert Space)**.** A Hilbert space $\mathcal{H}$ is a space endowed with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$, which is complete w.r.t. the induced norm $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f\rangle_{\mathcal{H}}}$ (i.e. all Cauchy sequences converge).

Definition A.0.2 (RKHS)**.** A Reproducing Kernel Hilbert Space over a non-empty set $\mathcal{X}$ is a Hilbert space $\mathcal{H}$ of functions $g : \mathcal{X} \to \mathbb{R}$ such that point-wise evaluations are continuous linear functionals on $\mathcal{H}$, i.e.

$$\forall x \in \mathcal{X}, \quad \exists C_x < \infty : |f(x)| \leq C_x \|g\|_{\mathcal{H}}, \quad \forall f \in \mathcal{H}$$

Definition A.0.3 (Positive Semidefinite Kernel)**.** Let $\mathcal{X}$ be a nonempty set. A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive semidefinite kernel if, for any finite $p \in \mathbb{N}$, it holds

$$\sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j K(x_i, x_j) \geq 0, \qquad \forall (x_k, a_k) \in (\mathcal{X}, \mathbb{R}), \quad k = 1, ..., p$$

Definition A.0.4 (Kernel Section)**.** Given a kernel K, the kernel section $K_x \in \mathcal{H}$ centred at x is defined as

$$K_x(a) = K(x, a), \qquad \forall a \in \mathcal{X}$$

The following theorem, due to Aronszajn (1950), establishes a one-to-one correspondence between RKHS and positive semidefinite kernels.

Theorem A.0.5. *[Moore-Aronszajn]Given a RKHS $\mathcal{H}$, there exists a unique positive semidefinite kernel, called the reproducing kernel, such that the reproducing property holds*

$$f(x) = \langle g, K_x\rangle_{\mathcal{H}}, \qquad \forall (x, f) \in (\mathcal{X}, \mathcal{H})$$

Conversely, given a positive semidefinite kernel, there exists a unique RKHS of real valued functions defined over $\mathcal{X}$ with reproducing kernel K.

The proof of the theorem shows how each RKHS is completely characterized by its associated kernel. Namely, each function $f \in \mathcal{H}$ can be represented as

$$f(\cdot) = \sum_{i=1}^{p} a_i K_{x_i}(\cdot) \tag{A.1}$$

for any choice of $p,\ a_i,\ x_i$. It turns out that every function belonging to the RKHS enjoys the properties which are encoded into the kernel.
Moreover, given the functions $f(\cdot) = \sum_{i=1}^{p} a_i K_{x_i}(\cdot)$ and $g(\cdot) = \sum_{i=1}^{m} b_i K_{s_i}(\cdot)$, the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is defined as

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{p} \sum_{j=1}^{m} a_i b_j K(x_i, s_j) \tag{A.2}$$

The key result for the theory of inverse problems is due to Kimeldorf and Wahba (1971), who showed that the solution of the variational problem

$$\underset{f \in \mathcal{H}}{\arg\min} \quad \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2 \tag{A.3}$$

can be expressed as the linear combination of a finite number of basis functions. In particular, such number equals the number of given data points N.

Theorem A.0.6. *[Representer Theorem]If $\mathcal{H}$ is a RKHS, the solution of* (A.3) *is*

$$\hat{f}(x) = \sum_{i=1}^{N} \hat{c}_i K_{x_i}(x) \tag{A.4}$$

where

$$\begin{aligned} \hat{c} = [\hat{c}_1\ \hat{c}_2\ \cdots\ \hat{c}_N]^\top &= (\bar{K} + \gamma I_N)^{-1} Y_N \\ \bar{K}_{ij} := K(i, j), \qquad & \bar{K} \in \mathbb{R}^{N \times N} \end{aligned}$$

In the literature the estimators (A.4) is also known as *regularization network* (Poggio and Girosi, 1990) or *least squares support vector machine* (Suykens and Vandewalle, 1999).
A generalization of the previous theorem has been derived by Schölkopf, Herbrich, and Smola (2001)s: the theorem still holds if the quadratic loss is replaced by other convex losses, such as the Huber (Huber, 2011) or the Vapnik loss (Vapnik, 1998).
An extensive treatment of the theory of RKHS and of inverse problems is provided in Cucker and Smale (2002).

B
Connections between EM and other algorithms

This appendix derives connections between the EM routine (Algorithm 3) and the gradient methods, such as the SGP (Algorithm 1), when they are adopted for Marginal Likelihood maximization. Such relations arise when the kernel $\bar{K}_\eta$ is assumed to have a simplified structure, i.e. it can be expressed as $\bar{K}_\eta = \eta \bar{K}$, where only $\eta \in \mathbb{R}_+$ has to be optimized. Under this assumption it is shown that the EM update rule coincides with a gradient-based update if a specific step-size $\alpha^{(k)}$ is chosen (see equation (2.196)). In addition, a connection between the EM algorithm and the iterative reweighted methods is highlighted: these approaches have been recently introduced in the compressive sensing literature and they have found wide application during the last years (Candes et al., 2008; Chartrand and Yin, 2008).

B.1 Connection between EM and Gradient Methods

In Section 2.4.5.2, the EM algorithm has been presented as an iterative method, where each iteration consists of two steps. At a generic iteration k, the first step requires to compute the lower bound $\mathfrak{L}(p_{\mathbf{g}}(\mathbf{g}|Y_N, \eta^{(k)}), \eta)$, while the second one determines the value of η which optimizes it.

Assuming $\bar{K}_\eta = \eta \bar{K}$, it follows that

$$\mathfrak{L}\left(p_{\mathbf{g}}(\mathbf{g}|Y_N, \eta^{(k)}), \eta\right) = -\frac{1}{2}\ln\det(\eta\bar{K}) - \frac{1}{2}\mathrm{Tr}\left[\frac{\bar{K}^{-1}}{\eta}\left(\Phi_N^\top \widetilde{\Sigma}_N^{-1}\Phi_N + \frac{\bar{K}^{-1}}{\hat{\eta}^{(k)}}\right)^{-1}\right] - \frac{1}{2\eta}\hat{\mathbf{g}}^{(k)\top}\bar{K}^{-1}\hat{\mathbf{g}}^{(k)} + \text{cost} \tag{B.1}$$

where terms not depending on η have been omitted, while $\hat{\mathbf{g}}^{(k)}$ denotes the impulse response estimate computed with the hyper-parameter η fixed to $\hat{\eta}^{(k)}$. The M-step is then performed by computing the derivative of the previous equation w.r.t. η and setting it to zero, leading to:

$$\hat{\eta}_{EM}^{(k+1)} = \frac{1}{pmT}\left\{\hat{\mathbf{g}}^{(k)\top}\bar{K}^{-1}\hat{\mathbf{g}}^{(k)} + \mathrm{Tr}\left[\bar{K}^{-1}\left(\Phi_N^\top \widetilde{\Sigma}_N^{-1}\Phi_N + \frac{\bar{K}^{-1}}{\hat{\eta}^{(k)}}\right)^{-1}\right]\right\} \tag{B.2}$$

Hence, $\hat{\eta}_{EM}^{(k+1)}$ is the hyper-parameter update computed by the EM algorithm.

Consider now the gradient update rule (2.196) for $\hat{\eta}^{(k+1)}$ (based on the minimization of the function $f_{ML}(\eta)$ defined in (2.184)):

$$\hat{\eta}_{GR}^{(k+1)} = \hat{\eta}^{(k)} - \alpha^{(k)} f'_{ML}(\eta^{(k)}) \tag{B.3}$$

where it has been set $H_N^{(k)} = 1$. The following result is derived.

Lemma B.1.1. *If* $\alpha^{(k)} = \frac{(\hat{\eta}^{(k)})^2}{pmT}$ *in* (B.3)*, then* $\hat{\eta}_{GR}^{(k+1)} = \hat{\eta}_{EM}^{(k+1)}$.

Proof: From (2.184), it follows:

$$f'_{ML}(\eta^{(k)}) = \frac{pmT}{\hat{\eta}^{(k)}} - \frac{1}{(\hat{\eta}^{(k)})^2}\mathrm{Tr}\left[\bar{K}^{-1}\left(\Phi_N^\top \widetilde{\Sigma}_N^{-1}\Phi_N + \frac{\bar{K}^{-1}}{\hat{\eta}^{(k)}}\right)^{-1}\right] - \frac{1}{(\hat{\eta}^{(k)})^2}\hat{\mathbf{g}}^{(k)^\top}\bar{K}^{-1}\hat{\mathbf{g}}^{(k)}$$

Now, introducing this value into (B.3) gives the result.

B.2 Connection between EM and Iterative Reweighted Methods

Iterative reweighted methods have been quite recently introduced in the compressive sensing field in order to improve the recovery of sparse solutions. Here the focus is on the ℓ_2-reweighted scheme that has been proposed by Wipf and Nagarajan (2010) for Sparse Bayesian Learning (SBL) (Tipping, 2001).

Recall the optimization problem (2.183) which has to be solved to determine $\hat{\eta}$. Under Gaussian assumptions, the following function has to be minimized:

$$\min_{\eta \geq 0} -\ln p_y(Y_N|\eta) = \min_{\eta \geq 0} Y_N^\top \Lambda(\eta)^{-1} Y_N + \ln\det\Lambda(\eta) \tag{B.4}$$

where $\Lambda(\eta) := \eta\Phi_N\bar{K}\Phi_N^\top + \widetilde{\Sigma}_N$. Notice that (Tipping (2001), Appendix A)

$$Y_N^\top\Lambda(\eta)^{-1}Y_N = \min_{\mathbf{g}\in\mathbb{R}^{pmT}} \|Y_N - \Phi_N\mathbf{g}\|^2_{\widetilde{\Sigma}_N^{-1}} + \mathbf{g}^\top(\eta\bar{K})^{-1}\mathbf{g}$$

Hence

$$\begin{aligned}\min_{\eta\geq 0} -\ln p_y(Y_N|\eta) &= \min_{\eta\geq 0, \mathbf{g}\in\mathbb{R}^{pmT}} \|Y_N - \Phi_N\mathbf{g}\|^2_{\widetilde{\Sigma}_N^{-1}} + \mathbf{g}^\top(\eta\bar{K})^{-1}\mathbf{g} + \ln\det\Lambda(\eta)\\ &= \min_{\mathbf{g}\in\mathbb{R}^{pmT}} \|Y_N - \Phi_N\mathbf{g}\|^2_{\widetilde{\Sigma}_N^{-1}} + b(\mathbf{g})\end{aligned}$$

where $b(\mathbf{g}) = \min_{\eta\geq 0}\mathbf{g}^\top(\eta\bar{K})^{-1}\mathbf{g} + \ln\det\Lambda(\eta)$, is a non-separable penalty function, since it can not be expressed as a summation over functions of the individual entries in $\mathbf{g}$. Furthermore, it is a non-decreasing concave function of $\mathbf{g}^2 := [\mathrm{vec}^\top(g(1)^2) \;\cdots\; \mathrm{vec}^\top(g(T)^2)]^\top$, thus allowing to employ iterative reweighted ℓ_2 schemes to minimize the function above.

Namely,

$$\begin{aligned} b(\mathbf{g}) &\leq \mathbf{g}^\top(\eta\bar{K})^{-1}\mathbf{g} + \ln\det\Lambda(\eta) \\ &= \mathbf{g}^\top(\eta\bar{K})^{-1}\mathbf{g} + \ln\det(\eta\bar{K}) + \ln\det\left(\Phi_N^\top\widetilde{\Sigma}_N^{-1}\Phi_N + (\eta\bar{K})^{-1}\right) + \text{cost} \qquad (B.5) \\ &\leq \mathbf{g}^\top(\eta\bar{K})^{-1}\mathbf{g} + \ln\det(\eta\bar{K}) + z\eta^{-1} - v^*(z) + \text{cost} \qquad (B.6) \end{aligned}$$

where $v^*(z)$ denotes the concave conjugate of $v(a) := \ln\det\left(\Phi_N^\top\widetilde{\Sigma}_N^{-1}\Phi_N + a\bar{K}^{-1}\right)$, $a = \eta^{-1}$, given by:

$$v^*(z) = \min_a za - \ln\det\left(\Phi_N^\top\widetilde{\Sigma}_N^{-1}\Phi_N + a\bar{K}^{-1}\right), \qquad a = \eta^{-1}$$

Notice that in (B.5) the Silvester's determinant identity is used and the bound (B.6) holds for all $z, \eta \geq 0$. Hence, we have

$$\begin{aligned} \min_{\eta\geq 0} -\ln p_y(Y_N|\eta) = \min_{\eta\geq 0, z\geq 0, \mathbf{g}\in\mathbb{R}^{pmT}} &\|Y_N - \Phi_N\mathbf{g}\|^2_{\widetilde{\Sigma}_N^{-1}} + \mathbf{g}^\top(\eta\bar{K})^{-1}\mathbf{g} \\ &+ \ln\det(\eta\bar{K}) + z\eta^{-1} - v^*(z) \qquad (B.7) \end{aligned}$$

where the terms that are not relevant to the optimization problem have been omitted. The analogies with the two steps of the EM algorithm can now be stated. Specifically, recall that the E-step in the EM is equivalent to solving problem (2.209), here reported for convenience:

$$\mathfrak{L}\left(p_{\mathbf{g}}(\mathbf{g}|Y_N, \eta^{(k)}), \eta\right) = \max_{q(\mathbf{g})} \mathfrak{L}(q(\mathbf{g}), \eta^{(k)}) \qquad (B.8)$$

The solution is given by the posterior distribution of $\mathbf{g}$ given $\hat{\eta}^{(k)}$, i.e. $p_{\mathbf{g}}(\mathbf{g}|Y_N, \hat{\eta}^{(k)})$. Analogously, solving (B.7) w.r.t. $\mathbf{g}$ for fixed $\hat{\eta}^{(k)}$ leads to an a-posteriori estimate, namely the Empirical Bayes estimator $\hat{\mathbf{g}}^{(k+1)} = \mathbb{E}[\mathbf{g}|Y_N, \hat{\eta}^{(k)}]$, which coincides with the Maximum a Posteriori estimator of $\mathbf{g}$.
On the other hand, solving (B.7) for fixed $\hat{\mathbf{g}}^{(k)}$ leads to

$$\hat{\eta}^{(k+1)} = \frac{1}{pmT}\left(\hat{\mathbf{g}}^{(k)\top}\bar{K}^{-1}\hat{\mathbf{g}}^{(k)} + z^*\right) \qquad (B.9)$$

where (Wipf and Nagarajan, 2010)

$$z^* = \frac{\partial}{\partial a}\ln\det\left(\Phi_N^\top\widetilde{\Sigma}_N^{-1}\Phi_N + a\bar{K}^{-1}\right) = \operatorname{Tr}\left[\bar{K}^{-1}\left(\Phi_N^\top\widetilde{\Sigma}_N^{-1}\Phi_N + \frac{\bar{K}^{-1}}{\hat{\eta}^{(k)}}\right)^{-1}\right]$$

Thus, the update (B.9) coincides with the M-step in (2.210).

References

Akaike H. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer, 1998.

Akaike H. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

Allen D. M. The relationship between variable selection and data agumentation and a method for prediction. *Technometrics*, 16(1):125–127, 1974.

Andrieu C., Doucet A., and Holenstein R. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

Antoulas A. C. *Approximation of large-scale dynamical systems*. SIAM, 2005a.

Antoulas A. C. An overview of approximation methods for large-scale dynamical systems. *Annual reviews in Control*, 29(2):181–190, 2005b.

Antoulas A. C., Sorensen D. C., and Gugercin S. A survey of model reduction methods for large-scale systems. *Contemporary mathematics*, 280:193–220, 2001.

Aoki M. *State space modeling of time series*. Springer Science & Business Media, 1990.

Aravkin A., Burke J. V., Chiuso A., and Pillonetto G. On the mse properties of empirical bayes methods for sparse estimation. *IFAC Proceedings Volumes*, 45(16): 965–970, 2012.

Aravkin A. Y., Burke J. V., Chiuso A., and Pillonetto G. Convex vs non-convex estimators for regression and sparse estimation: the mean squared error properties of ard and glasso. *Journal of Machine Learning Research*, 15(1):217–252, 2014.

Aronszajn N. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

Aström K. J. *Lectures on the identification problem the least squares method*. Institute of Technology, Division of Automatic Control, 1968.

Åström K.-J. and Bohlin T. Numerical identification of linear dynamic systems from normal operating records. In *Theory of self-adaptive control systems*, pages 96–111. Springer, 1966.

Åström K. J. and Eykhoff P. System identification?a survey. *Automatica*, 7(2): 123–162, 1971.

Baraniuk R. G. Compressive sensing. *IEEE signal processing magazine*, 24(4), 2007.

Barzilai J. and Borwein J. M. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

Bauer D. Some asymptotic theory for the estimation of linear systems using maximum likelihood methods or subspace algorithms. 1998.

Bauer D. Order estimation for subspace methods. *Automatica*, 37(10):1561–1573, 2001.

Bauer D. Asymptotic properties of subspace estimators. *Automatica*, 41(3):359 – 376, 2005. ISSN 0005-1098. URL http://www.sciencedirect.com/science/article/pii/S0005109804003292. Data-Based Modelling and System Identification.

Bauer D., Deistler M., and Scherrer W. Asymptotic distributions of subspace estimates under misspecification of the order. In *Proceedings of Mathematical Theory of Networks and Systems*, 1998.

Bauer D., Deistler M., and Scherrer W. Consistency and asymptotic normality of some subspace algorithms for systems without observed inputs. *Automatica*, 35(7): 1243–1254, 1999.

Bauer D. and Jansson M. Analysis of the asymptotic properties of the moesp type of subspace algorithms. *Automatica*, 36(4):497–509, 2000.

Bauer D. and Ljung L. Some facts about the choice of the weighting matrices in larimore type of subspace algorithms. *Automatica*, 38(5):763–773, 2002.

Bekiroglu K., Yilmaz B., Lagoa C., and Sznaier M. Parsimonious model identification via atomic norm minimization. In *Proc. of European Control Conference*, 2014.

Bellman R. and Åström K. On structural identifiability. *Mathematical Biosciences*, 7(3):329 – 339, 1970.

Benner P., Gugercin S., and Willcox K. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.

Berger J. O. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

Bertsekas D. P. *Constrained optimization and Lagrange multiplier methods.* Academic press, 2014.

Bhotto M. Z. A. and Antoniou A. New improved recursive least-squares adaptive-filtering algorithms. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60 (6):1548–1558, 2013.

Bishop C. M. *Pattern Recognition and Machine Learning.* Springer, 2006. URL http://research.microsoft.com/en-us/um/people/cmbishop/prml/.

Bonettini S., Chiuso A., and Prato M. A scaled gradient projection methods for Bayesian learning in dynamical systems. *SIAM Journal on Scientific Computing*, 37 (3):1297–1318, 2015.

Bottegal G., Aravkin A. Y., Hjalmarsson H., and Pillonetto G. Robust em kernel-based methods for linear system identification. *Automatica*, 67:114–126, 2016.

Bottegal G. and Pillonetto G. Regularized spectrum estimation using stable spline kernels. *Automatica*, 49(11):3199–3209, 2013.

Box G. E. and Jenkins G. M. Time series analysis forecasting and control. Technical report, DTIC Document, 1970.

Brockett R. *Finite dimensional linear systems.* Series in decision and control. Wiley, 1970.

Brockwell P. J. and Davis R. A. *Time series: theory and methods.* Springer Science & Business Media, 2013.

Bui-Thanh T., Willcox K., Ghattas O., and van Bloemen Waanders B. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224(2):880–896, 2007.

Burnham K. P., Anderson D. R., and Burnham K. P. *Model selection and multimodel inference: A practical information-theoretic approach.* Springer, 2nd edition, 2002.

Caines P. Stationary linear and nonlinear system identification and predictor set completeness. *IEEE Transactions on Automatic Control*, 23(4):583–594, 1978.

Calafiore G. Approximation of n-dimensional data using spherical and ellipsoidal primitives. *IEEE Transaction on System, Mand, And Cybernetics*, 32, March 2002.

Camba-Mendez G. and Kapetanios G. Testing the rank of the hankel covariance matrix: A statistical approach. *IEEE Transactions on Automatic Control*, 46(2): 331–336, 2001.

Campi M. C., Ko S., and Weyer E. Non-asymptotic confidence regions for model parameters in the presence of unmodelled dynamics. *Automatica*, 45(10):2175–2186, 2009.

Campi M. C. and Weyer E. Finite sample properties of system identification methods. *IEEE Transactions on Automatic Control*, 47(8):1329–1334, 2002.

Campi M. C. and Weyer E. Guaranteed non-asymptotic confidence regions in system identification. *Automatica*, 41(10):1751–1764, 2005.

Campi M. C. and Weyer E. Identification with finitely many data points: The lscr approach. *IFAC Proceedings Volumes*, 39(1):46–64, 2006a.

Campi M. C. and Weyer E. Non-asymptotic confidence sets for input-output transfer functions. In *Decision and Control, 2006 45th IEEE Conference on*, pages 157–162. IEEE, 2006b.

Campi M. C. and Weyer E. Non-asymptotic confidence sets for the parameters of linear transfer functions. *IEEE Transactions on Automatic Control*, 55(12):2708–2720, 2010.

Candès E. J. and Recht B. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

Candes E., Wakin M., and Boyd S. Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

Caponnetto A. and De Vito E. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Carlberg K. and Farhat C. A low-cost, goal-oriented ?compact proper orthogonal decomposition?basis for model reduction of static systems. *International Journal for Numerical Methods in Engineering*, 86(3):381–402, 2011.

Carli F., Chen T., and Ljung L. Maximum entropy kernels of system identification. *ArXiv*, 2014.

Carli F. P., Chiuso A., and Pillonetto G. Efficient algorithms for large scale linear system identification using stable spline estimators. *IFAC Proceedings Volumes*, 45(16): 119–124, 2012.

Caruana R. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

Cavalier L., Golubev G., Picard D., Tsybakov A., and others . Oracle inequalities for inverse problems. *The Annals of Statistics*, 30(3):843–874, 2002.

Chartrand R. and Yin W. Iteratively reweighted algorithms for compressive sensing. In *Acoustics, speech and signal processing, 2008. ICASSP 2008. IEEE international conference on*, pages 3869–3872. IEEE, 2008.

Chatelin F. Spectral approximation of linear operators' a c. *Press. New York*, 1983.

Chen T., Ardeshiri T., Carli F. P., Chiuso A., Ljung L., and Pillonetto G. Maximum entropy properties of discrete-time first-order stable spline kernel. *Automatica*, 2016.

Chen T., Ohlsson H., and Ljung L. On the estimation of transfer functions, regularizations and Gaussian processes - revisited. *Automatica*, 48(8):1525–1535, 2012.

Chen T., Andersen M. S., Ljung L., Chiuso A., and Pillonetto G. System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques. *IEEE Transactions on Automatic Control*, 59(11):2933–2945, 2014.

Chen T. and Ljung L. Implementation of algorithms for tuning parameters in regularized least squares problems in system identification. *Automatica*, 49(7):2213–2220, 2013.

Chen T. and Ljung L. Constructive state space model induced kernels for regularized system identification. *IFAC Proceedings Volumes*, 47(3):1047–1052, 2014.

Chiuso A. On the asymptotic properties of closed-loop CCA-type subspace algorithms: Equivalence results and role of the future horizon. *IEEE Transactions on Automatic Control*, 55(3):634–649, March 2010. ISSN 0018-9286.

Chiuso A. Regularization and bayesian learning in dynamical systems: Past, present and future. *Annual Reviews in Control*, 41:24–38, 2016.

Chiuso A., Chen T., Ljung L., and Pillonetto G. Regularization strategies for nonparametric system identification. In *Proc. of IEEE Conf. on Dec. and Control (CDC2013)*, 2013.

Chiuso A. and Pillonetto G. A Bayesian approach to sparse dynamic network identification. *Automatica*, 48(8):1553 – 1565, 2012. ISSN 0005-1098.

Chiuso A., Chen T., Ljung L., and Pillonetto G. On the design of multiple kernels for nonparametric linear system identification. In *53rd IEEE Conference on Decision and Control*, pages 3346–3351. IEEE, 2014.

Chiuso A. and Picci G. The asymptotic variance of subspace estimates. *Journal of Econometrics*, 118(1):257–291, 2004a.

Chiuso A. and Picci G. Numerical conditioning and asymptotic variance of subspace estimates. *Automatica*, 40(4):677–683, 2004b.

Chiuso A. and Picci G. Consistency analysis of some closed-loop subspace identification methods. *Automatica*, 41(3):377–391, 2005.

Chiuso A. and Pillonetto G. Learning sparse dynamic linear systems using stable spline kernels and exponential hyperpriors. In *Advances in Neural Information Processing Systems*, pages 397–405, 2010.

Cho Y. S., Kim S. B., and Powers E. J. Time-varying spectral estimation using ar models with variable forgetting factors. *IEEE Transactions on Signal Processing*, 39 (6):1422–1426, 1991.

Cho Y. M., Xu G., and Kailath T. Fast recursive identification of state space models via exploitation of displacement structure. *Automatica*, 30(1):45–59, 1994.

Choi T. and Schervish M. J. Posterior consistency in nonparametric regression problems under gaussian process priors. 2004.

Clarke D. Generalized least squares estimation of the parameters of a dynamic model. In *First IFAC Symposium on Identification in Automatic Control Systems, Prague*, 1967.

Cover T. and Thomas J. *Elements of Information Theory.* Series in Telecommunications and Signal Processing. Wiley, 1991.

Cramér H. *Mathematical Methods of Statistics*, volume 9. Princeton University Press, 1945.

Csáji B. C., Campi M. C., and Weyer E. Sign-perturbed sums: A new system identification approach for constructing exact non-asymptotic confidence regions in linear regression models. *IEEE Transactions on Signal Processing*, 63(1):169–181, 2015.

Cucker F. and Smale S. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49, 2002.

Darwish M., Tóth R., and van den Hof P. Bayesian system identification based on generalized orthonormal basis functions. 2014.

Daubechies I., DeVore R., Fornasier M., and Güntürk C. S. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.

De Mol C., Giannone D., and Reichlin L. Forecasting using a large number of predictors: Is bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics*, 146(2):318–328, 2008.

De Moor B., Van Overschee P., and Suykens J. Subspace algorithms system identification and stochastic realization. *Proceedings MTNS, Kobe, Japan*, 1991.

De Moor B., Vandewalle J., Moonen M., Van Mieghem P., and Vandenberghe L. A geometrical strategy for the identification of state space models of linear multivariable systems with singular value decomposition. In *Proc. 8th IFAC/IFORS Symposium on identification and system parameter estimation*, pages 700–704, 1988.

De Schutter B. Minimal state-space realization in linear system theory: an overview. *Journal of Computational and Applied Mathematics*, 121(1):331–354, 2000.

DeGroot M. H. *Optimal statistical decisions*, volume 82. John Wiley & Sons, 2005.

Deistler M., Peternell K., and Scherrer W. Consistency and relative efficiency of subspace methods. *Automatica*, 31(12):1865–1875, 1995.

Dempster A. P., Laird N. M., and Rubin D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Den Dekker A., Bombois X., and Van den Hof P. Finite sample confidence regions for parameters in prediction error identification using output error models. In *Proceedings of the 17th IFAC World Congress*, 2008.

Dennis Jr J. E. and Schnabel R. B. *Numerical methods for unconstrained optimization and nonlinear equations*, volume 16. Siam, 1996.

Dinuzzo F. Kernels for linear time invariant system identification. *SIAM Journal on Control and Optimization*, 53(5):3299–3317, 2015.

Doan T., Litterman R., and Sims C. Forecasting and conditional projection using realistic prior distributions. *Econometric Reviews*, 3:1–100, 1984.

Draper N. R. Applied regression analysis bibliography update 1994-97. *Communications in Statistics-Theory and Methods*, 27(10):2581–2623, 1998.

Dunstan W. J. and Bitmead R. R. Empirical estimation of parameter distributions in system identification. In *Proceedings of the 13th IFAC Symposium on System Identificatin, The Netherlands.* Citeseer, 2003.

Durbin J. The fitting of time-series models. *Revue de l'Institut International de Statistique*, pages 233–244, 1960.

Efron B. Bayesians, frequentists, and scientists. *Journal of the American Statistical Association*, 100(469):1–5, 2005.

Efron B. The estimation of prediction error. *Journal of the American Statistical Association*, 2012.

Engl H. W., Kunisch K., and Neubauer A. Convergence rates for tikhonov regularisation of non-linear ill-posed problems. *Inverse problems*, 5(4):523, 1989.

Enns D. F. Model reduction for control system design. 1985.

Evgeniou T., Micchelli C. A., and Pontil M. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(Apr):615–637, 2005.

Evgeniou T. and Pontil M. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

Evgeniou T., Pontil M., and Poggio T. Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1–50, 2000.

Fazel M., Hindi H., and Boyd S. P. A rank minimization heuristic with application to minimum order system approximation. In *In Proceedings of the 2001 American Control Conference*, pages 4734–4739, 2001.

Fazel M., Kei P. T., Sun D., and Tseng P. Hankel matrix rank minimization with applications to system identification and realization. *SIAM Journal on Matrix Analysis and Applications*, 34(3):946–977, 2013.

Fornasier M., Rauhut H., and Ward R. Low-rank matrix recovery via iteratively reweighted least squares minimization. *SIAM Journal on Optimization*, 21(4):1614–1640, 2011.

Fortescue T., Kershenbaum L., and Ydstie B. Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, 17(6):831–835, 1981.

Galrinho M., Rojas C., and Hjalmarsson H. A weighted least-squares method for parameter estimation in structured models. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 3322–3327. IEEE, 2014.

Garatti S., Campi M. C., and Bittanti S. Assessing the quality of identified models through the asymptotic theory?when is the result reliable? *Automatica*, 40(8): 1319–1332, 2004.

Geerardyn E., Lumori M. L., and Lataire J. Frf smoothing to improve initial estimates for transfer function identification. *IEEE Transactions on Instrumentation and Measurement*, 64(10):2838–2847, 2015.

Genesio R. and Pomé R. Identification of reduced models from noisy data. *International Journal of Control*, 21(2):203–211, 1975.

Gevers M. and Wertz V. Paper: Uniquely identifiable state-space and arma parametrizations for multivariable linear systems. *Automatica*, 20(3):333–347, May 1984.

Giannone D., Lenza M., and Primiceri G. E. Prior selection for vector autoregressions. *Review of Economics and Statistics*, 97(2):436–451, 2015.

Gilbert E. G. Controllability and observability in multivariable control systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 1(2):128–151, 1963.

Gilks W. R. *Markov chain monte carlo.* Wiley Online Library, 2005.

Girosi F., Jones M., and Poggio T. Regularization theory and neural networks architectures. *Neural computation*, 7(2):219–269, 1995.

Glover K. All optimal hankel-norm approximations of linear multivariable systems and their l,?-error bounds. *International journal of control*, 39(6):1115–1193, 1984.

Golub G. H., Heath M., and Wahba G. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

Goodwin G., Gevers M., and Ninness B. Quantifying the error in estimated transfer functions with application to model order selection. *IEEE Trans. on Automatic Control*, 37(7):913–928, 1992.

Goodwin G. C. and Payne R. L. Dynamic system identification: experiment design and data analysis. 1977.

Grenander U. Stochastic processes and statistical inference. *Arkiv för matematik*, 1 (3):195–277, 1950.

Grossmann C., Jones C. N., and Morari M. System identification via nuclear norm regularization for simulated moving bed processes from incomplete data sets. In *CDC*, pages 4692–4697. IEEE, 2009.

Gugercin S. An iterative svd-krylov based method for model reduction of large-scale dynamical systems. *Linear Algebra and its Applications*, 428(8-9):1964–1986, 2008.

Gugercin S. and Antoulas A. C. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8):748–766, 2004.

Gugercin S. and Antoulas A. C. Model reduction of large-scale systems by least squares. *Linear algebra and its applications*, 415(2-3):290–321, 2006.

Gugercin S., Antoulas A. C., and Beattie C. H_2 model reduction for large-scale linear dynamical systems. *SIAM journal on matrix analysis and applications*, 30(2): 609–638, 2008.

Gugercin S., Sorensen D. C., and Antoulas A. C. A modified low-rank smith method for large-scale lyapunov equations. *Numerical Algorithms*, 32(1):27–55, 2003.

Gustafsson T. Recursive system identification using instrumental variable subspace tracking. In *Proceedings of the 11th IFAC Symposium on System Identification (SYSID 1997), Fukuoka, Japan.* Citeseer, 1997.

Gustafsson T., Lovera M., and Verhaegen M. A novel algorithm for recursive instrumental variable based subspace identification. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 4, pages 3920–3925. IEEE, 1998.

Ha H., Welsh J. S., Blomberg N., Rojas C. R., and Wahlberg B. Reweighted nuclear norm regularization: A sparseva approach. *IFAC-PapersOnLine*, 48(28): 1172–1177, 2015.

Haario H., Saksman E., and Tamminen J. An adaptive metropolis algorithm. *Bernoulli*, pages 223–242, 2001.

Hannan E. J. *Multiple time series*, volume 38. John Wiley & Sons, 2009.

Hannan E. J. and Deistler M. *The statistical theory of linear systems*, volume 70. SIAM, 1988.

Hansson A., Liu Z., and Vandenberghe L. Subspace system identification via weighted nuclear norm optimization. In *Proc. of CDC*, pages 3439–3444. IEEE, 2012. ISBN 978-1-4673-2065-8.

Härdle W. and Bowman A. W. Bootstrapping in nonparametric regression: local adaptive smoothing and confidence bands. *Journal of the American Statistical Association*, 83(401):102–110, 1988.

Hardle W. and Marron J. Bootstrap simultaneous error bars for nonparametric regression. *The Annals of Statistics*, pages 778–796, 1991.

Harville D. A. Matrix algebra from a statistician's perspective. *Technometrics*, 40(2): 164–164, 1998.

Hastie T., Tibshirani R., and Friedman J. *The elements of statistical learning: data mining, inference and prediction.* Springer, 2009. URL http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

Hill S. D. Reduced gradient computation in prediction error identification. Technical report, DTIC Document, 1985.

Hjalmarsson H., Welsh J., and Rojas C. Identification of Box-Jenkins models using structured ARX models and nuclear norm relaxation. 2012.

Hjalmarsson H. and Ljung L. Estimating model variance in the case of undermodeling. *IEEE Transactions on Automatic Control*, 37(7):1004–1008, 1992.

Ho B. and Kalman R. E. Editorial: Effective construction of linear state-variable models from input/output functions. *at-Automatisierungstechnik*, 14(1-12):545–548, 1966.

Hoerl A. E. and Kennard R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Hotelling H. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

Hovland S., Willcox K., and Gravdahl J. Mpc for large-scale systems via model reduction and multiparametric quadratic programming. In *Decision and Control, 2006 45th IEEE Conference on*, pages 3418–3423. IEEE, 2006.

Hsia T. C. *System identification: Least-squares methods.* Lexington, Mass., D. C. Heath and Co., 1977. 177 p, 1977.

Huber P. J. *Robust statistics.* Springer, 2011.

Jansson M. *On subspace methods in system identification and sensor array signal processing.* PhD thesis, School of Electrical Engineering, Royal Institute of Technology, 1997.

Jansson M. Asymptotic variance analysis of subspace identification methods. In *IN PROCEEDINGS OF SYSID2000, S. BARBARA CA.* Citeseer, 2000.

Jansson M. and Wahlberg B. On weighting in state-space subspace system identification. In *Proc. European Control Conference, ECC*, volume 95, pages 435–440, 1995.

Jansson M. and Wahlberg B. On consistency of subspace methods for system identification. *Automatica*, 34(12):1507–1519, 1998.

Jaynes E. T. and Kempthorne O. Confidence intervals vs bayesian intervals. In *Foundations of probability theory, statistical inference, and statistical theories of science*, pages 175–257. Springer, 1976.

Jiang J. and Cook R. Fast parameter tracking rls algorithm with high noise immunity. *Electronics Letters*, 28(22):2043–2045, 1992.

Kailath T. *Linear systems*, volume 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

Kalman R. Irreducible realizations and the degree of a rational matrix. *Journal of the Society for Industrial and Applied Mathematics*, 13(2):520–544, 1965.

Kalman R. E. On minimal partial realizations of a linear input/output map. 1971.

Kay S. M. Modern spectral estimation: theory and application. *Signal Processing Series. 4th Edition, Prentice Hall, Englewood Cliffs*, 1988.

Kimeldorf G. and Wahba G. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95, 1971.

Kimeldorf G. S. and Wahba G. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502, 1970.

Knox T., Stock J. H., and Watson M. W. Empirical bayes forecasts of one time series using many predictors, 2001.

Kung S.-Y. A new identification and model reduction algorithm via singular value decomposition. In *Proceedings of the 12th Asilomar conference on circuits, systems and computers*, pages 6–8, 1978.

Lacy S. L. and Bernstein D. S. Subspace identification with guaranteed stability using constrained optimization. *IEEE Transactions on automatic control*, 48(7):1259–1263, 2003.

Larimore W. E. System identification, reduced-order filtering and modeling via canonical variate analysis. In *American Control Conference, 1983*, pages 445–451. IEEE, 1983.

Larimore W. E. Canonical variate analysis in identification, filtering, and adaptive control. In *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, pages 596–604. IEEE, 1990.

Larimore W. E. The optimality of canonical variate identification by example. In *Proc. of SYSID*, volume 94, pages 151–156, 1994.

Lasserre J. B. A trace inequality for matrix product. *IEEE Transactions on Automatic Control*, 40(8):1500–1501, 1995.

Lataire J. and Chen T. Transfer function and transient estimation by gaussian process regression in the frequency domain. *Automatica*, 72:217–229, 2016.

Latham G. A. and Anderson B. D. Frequency-weighted optimal hankel-norm approximation of stable transfer functions. *Systems & Control Letters*, 5(4):229–236, 1985.

Lawson C. L. and Hanson R. J. *Solving least squares problems*, volume 15. SIAM, 1995.

Leeb H. and Potscher B. M. Model selection and inference: facts and fiction. *Econometric Theory*, 21(01):21–59, 2005.

Leeb H. and Pötscher B. M. Can one estimate the conditional distribution of post-model-selection estimators? *The Annals of Statistics*, pages 2554–2591, 2006.

Leung S.-H. and So C. Gradient-based variable forgetting factor rls algorithm in time-varying environments. *IEEE Transactions on Signal Processing*, 53(8):3141–3150, 2005.

Liu Z., Hansson A., and Vandenberghe L. Nuclear norm system identification with missing inputs and outputs. *Systems and Control Letters*, 62(8):605–612, 2013.

Liu Z. and Vandenberghe L. Interior-Point Method for Nuclear Norm Approximation with Application to System Identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2009.

Ljung L. *System Identification - Theory for the User.* Prentice-Hall, Upper Saddle River, N.J., 2nd edition, 1999.

Ljung L. and Rissanen J. On canonical forms, parameter identifiability and the concept of complexity. *In Proc. 4th IFAC Symposium on Identification and System Parameter Estimation*, pages 58–69, 1976.

Ljung L. On the consistency of prediction error identification methods. *Mathematics in Science and Engineering*, 126:121–164, 1976.

Ljung L. Convergence analysis of parametric identification methods. *IEEE transactions on automatic control*, 23(5):770–783, 1978.

Ljung L. On the estimation of transfer functions. *Automatica*, 21(6):677–696, 1985.

Ljung L. Aspects and experiences of user choices in subspace identification. 2003.

Ljung L. Frequency domain versus time domain methods in system identification–revisited. In *Control of Uncertain Systems: Modelling, Approximation, and Design*, pages 277–291. Springer, 2006.

Ljung L. System identification toolbox for use with {MATLAB}. 2007.

Ljung L. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.

Ljung L. and Glover K. Frequency domain versus time domain methods in system identification. *Automatica*, 17(1):71–86, 1981.

Ljung L., Goodwin G. C., and Agüero J. C. Stochastic embedding revisited: A modern interpretation. *nation*, 15(24):35, 2014.

Ljung L., Hjalmarsson H., and Ohlsson H. Four encounters with system identification. *European Journal of Control*, 17(5):449–471, 2011.

Ljung L. and McKelvey T. Subspace identification from closed loop data. *Signal processing*, 52(2):209–215, 1996.

Ljung L. and Söderström T. Theory and practice of recursive identification. 1983.

Lovera M., Gustafsson T., and Verhaegen M. Recursive subspace identification of linear and non-linear wiener state-space models. *Automatica*, 36(11):1639–1650, 2000.

MacKay D. J. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

MacKay D. J. and Neal R. M. Automatic relevance determination for neural networks. In *Technical Report in preparation*. Cambridge University, 1994.

Magni P., Bellazzi R., and De Nicolao G. Bayesian function learning using mcmc methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12): 1319–1331, 1998.

Mallows C. L. Some comments on c p. *Technometrics*, 15(4):661–675, 1973.

Maritz J. and Lwin T. Empirical bayes methods. Technical report, 1989.

Mayne D. Q. and Firoozan F. Linear identification of arma processes. *Automatica*, 18(4):461–466, 1982.

McKelvey T. *Identification of state-space models from time and frequency data.* Department of Electrical Engineering, Linköping University, 1995.

McKelvey T., Akçay H., and Ljung L. Subspace-based multivariable system identification from frequency response data. *IEEE Transactions on Automatic Control*, 41(7):960–979, 1996.

McLachlan G. and Krishnan T. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

Mendel J. *Discrete techniques of parameter estimation: the equation error formulation.* Control and Systems Theory Series. M. Dekker, 1973.

Mendelson S., Neeman J., and others . Regularization in kernel learning. *The Annals of Statistics*, 38(1):526–565, 2010.

Mercère G., Bako L., and Lecœuche S. Propagator-based methods for recursive subspace model identification. *Signal Processing*, 88(3):468–491, 2008.

Mercere G., Lecoeuche S., and Lovera M. Recursive subspace identification based on instrumental variable unconstrained quadratic optimization. *International Journal of Adaptive Control and Signal Processing*, 18(9-10):771–797, 2004.

Mercère G. and Lovera M. Convergence analysis of instrumental variable recursive subspace identification algorithms. *Automatica*, 43(8):1377–1386, 2007.

Micchelli C. and Pontil M. On learning vector-valued functions. *Neural Comput.*, 17 (1):177–204, 2005a.

Micchelli C. A. and Pontil M. On learning vector-valued functions. *Neural computation*, 17(1):177–204, 2005b.

Mohan K. and Fazel M. Reweighted nuclear norm minimization with application to system identification. In *American Control Conference (ACC), 2010*, pages 2953–2959. IEEE, 2010.

Mohan K. and Fazel M. Iterative reweighted algorithms for matrix rank minimization. *Journal of Machine Learning Research*, 13:3441–3473, 2012.

Moore B. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*, 26(1):17–32, 1981.

Nguyen V. and Wood E. Review and unification of linear identifiability concepts. *SIAM review*, 24(1):34–51, 1982.

Ninness B. and Henriksen S. Bayesian system identification via markov chain monte carlo techniques. *Automatica*, 46(1):40–51, 2010.

Nocedal J. and Wright S. *Numerical optimization.* Springer Science & Business Media, 2006.

Nychka D. Bayesian confidence intervals for smoothing splines. *Journal of the American Statistical Association*, 83(404):1134–1143, 1988.

Obinata G. and Anderson B. D. *Model reduction for control system design.* Springer Science & Business Media, 2012.

Ohlsson H., Ljung L., and Boyd S. Segmentation of arx-models using sum-of-norms regularization. *Automatica*, 46(6):1107–1111, 2010.

Oku H. and Kimura H. Recursive 4sid algorithms using gradient type subspace tracking. *Automatica*, 38(6):1035–1043, 2002.

Ottersten B., Sensorer S., Ottersten B. O., Viberg M., and others . A subspace based instrumental variable method for state-space system identification. In *In: Proc. of SYSID'94.* Citeseer, 1994.

Paleologu C., Benesty J., and Ciochina S. A robust variable forgetting factor recursive least-squares algorithm for system identification. *IEEE Signal Processing Letters*, 15:597–600, 2008.

Papoulis A. and Pillai S. U. *Probability, random variables, and stochastic processes.* Tata McGraw-Hill Education, 2002.

Park D., Jun B., and Kim J. Fast tracking rls algorithm using novel variable forgetting factor with unity zone. *Electronics Letters*, 27(23):2150–2151, 1991.

Parzen E. An approach to time series analysis. *The Annals of Mathematical Statistics*, pages 951–989, 1961.

Parzen E. Statistical inference on time series by rkhs methods. Technical report, DTIC Document, 1970.

Penzl T. Algorithms for model reduction of large dynamical systems. *Linear Algebra and its Applications*, 415(2-3):322–343, 2006.

Peternell K. *Identification of linear dynamic systems by subspace and realization-based algorithms.* na, 1995.

Peternell K., Scherrer W., and Deistler M. Statistical analysis of novel subspace identification methods. *Signal Processing*, 52(2):161–177, 1996.

Pillonetto G., Chen T., Chiuso A., Ljung L., and Nicolao G. D. Regularized linear system identification using atomic, nuclear and kernel-based norms: The role of the stability constraint. *Automatica*, 59:–, 2016.

Pillonetto G. and Chiuso A. Tuning complexity in kernel-based linear system identification: the robustness of the marginal likelihood estimator. *Automatica*, 58: 106–117, 2015.

Pillonetto G., Chiuso A., and De Nicolao G. Prediction error identification of linear systems: a nonparametric Gaussian regression approach. *Automatica*, 47(2): 291–305, 2011a.

Pillonetto G. and De Nicolao G. A new kernel-based approach for linear system identification. *Automatica*, 46(1):81–93, 2010.

Pillonetto G., Dinuzzo F., Chen T., Nicolao G. D., and Ljung L. Kernel methods in system identification, machine learning and function estimation: a survey. *Automatica*, 2014.

Pillonetto G. and De Nicolao G. Pitfalls of the parametric approaches exploiting cross-validation for model order selection. *IFAC Proceedings Volumes*, 45(16):215–220, 2012.

Pillonetto G., Quang M. H., and Chiuso A. A new kernel-based approach for nonlinearsystem identification. *IEEE Transactions on Automatic Control*, 56(12): 2825–2840, 2011b.

Pintelon R., Guillaume P., Rolain Y., Schoukens J., Van Hamme H., and others . Parametric identification of transfer functions in the frequency domain-a survey. *IEEE transactions on automatic control*, 39(11):2245–2260, 1994.

Pintelon R. and Schoukens J. *System identification: a frequency domain approach.* John Wiley & Sons, 2012.

Poggio T. and Girosi F. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

Prando G. and Chiuso A. Model reduction for linear bayesian system identification. In *Proc. of IEEE CDC*, 2015.

Prando G., Chiuso A., and Pillonetto G. Bayesian and regularization approaches to multivariable linear system identification: the role of rank penalties. In *Proc. of IEEE CDC*, 2014.

Prando G., Pillonetto G., and Chiuso A. The role of rank penalties in linear system identification. In *Proc. of 17th IFAC Symposium on System Identification, SYSID, Beijing*, 2015.

Prando G., Romeres D., Pillonetto G., and Chiuso A. Classical vs. bayesian methods for linear system identification: point estimators and confidence sets. In *Proc. of ECC*, 2016a.

Prando G., Chiuso A., and Pillonetto G. Maximum entropy vector kernels for mimo system identification. *arXiv preprint arXiv:1508.02865, Automatica (accepted as regular paper)*, 2017.

Prando G., Romeres D., and Chiuso A. Online identification of time-varying systems: a bayesian approach. In *Proc. of IEEE CDC*, 2016b.

Qin S. J. and Ljung L. Closed-loop subspace identification with innovation estimation. 2003.

Raftery A. and Lewis S. One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. *Statistical Science*, 7:493–497, 1992.

Raftery A. E. and Lewis S. M. Implementing mcmc. *Markov chain Monte Carlo in practice*, pages 115–130, 1996.

Rahimi A. and Recht B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

Rake H. Step response and frequency response methods. *Automatica*, 514:519–526, 1980.

Rake H. Identification: transient and frequency response methods. *Systems and control encyclopedia; theory, technology, applications. Pergamon Press, Oxford*, 1987.

Rasmussen C. and Williams C. *Gaussian Processes for Machine Learning.* The MIT Press, 2006.

Recht B., Fazel M., and Parrilo P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

Rissanen J. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Risuleo R. S., Bottegal G., and Hjalmarsson H. A kernel-based approach to hammerstein system identication. *IFAC-PapersOnLine*, 48(28):1011–1016, 2015.

Rojas C. R. and Hjalmarsson H. Sparse estimation based on a validation criterion. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2825–2830. IEEE, 2011.

Romeres D., Pillonetto G., and Chiuso A. Identification of stable models via nonparametric prediction error methods. In *Control Conference (ECC), 2015 European*, pages 2044–2049. IEEE, 2015.

Romeres D., Prando G., Pillonetto G., and Chiuso A. On-line bayesian system identification. In *Proc. of ECC*, 2016.

Sadigh D., Ohlsson H., Sastry S. S., and Seshia S. A. Robust subspace system identification via weighted nuclear norm optimization. *CoRR*, abs/1312.2132, 2013.

Schmidt R. O. A signal subspace approach to multiple emitter location spectral estimation. *Ph. D. Thesis, Stanford University*, 1981.

Schölkopf B., Herbrich R., and Smola A. J. A generalized representer theorem. In *International Conference on Computational Learning Theory*, pages 416–426. Springer, 2001.

Schölkopf B. and Smola A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

Shah P., Narayan Bhaskar B., Tang G., and Recht B. Linear System Identification via Atomic Norm Regularization. In *Proc. of Conference on Decision and Control*, pages 6265–6270, 2012.

Sheskin D. J. *Handbook of parametric and nonparametric statistical procedures.* crc Press, 2003.

Sjöberg J., McKelvey T., and Ljung L. On the use of regularization in system identification. In *Proceedings of the 12th IFAC World Congress, Sydney, Australia*, volume 7, pages 381–386, 1993.

Slock D. T. and Kailath T. Fast transversal filters with data sequence weighting. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):346–359, 1989.

Smale S. and Zhou D.-X. Learning theory estimates via integral operators and their approximations. *Constructive approximation*, 26(2):153–172, 2007.

Smith R. Frequency Domain Subspace Identification using Nuclear Norm Minimization and Hankel Matrix Realizations. *IEEE Transactions on Automatic Control*, 59(11): 2886–2896, November 2014.

Smola A. J. and Schölkopf B. Sparse greedy matrix approximation for machine learning. 2000.

Söderström T. Test of pole-zero cancellation in estimated models. *Automatica*, 11(5): 537–539, 1975.

Söderström T. and Stoica P. *System Identification.* Prentice-Hall, 1989.

Söderström T., Stoica P., and Friedlander B. An indirect prediction error method for system identification. *Automatica*, 27(1):183–188, 1991.

Söderström T. D. and Stoica P. G. *Instrumental variable methods for system identification*, volume 57. Springer, 1983.

Solbrand G., Ahlén A., and Ljung L. Recursive methods for off-line identification. *International Journal of Control*, 41(1):177–191, 1985.

Song S., Lim J.-S., Baek S., and Sung K.-M. Gauss newton variable forgetting factor recursive least squares for time varying parameter tracking. *Electronics letters*, 36(11):988–990, 2000.

Sorelius J. *Subspace-based parameter estimation problems in signal processing.* Department of Information Technology, Uppsala University,, 1999.

Sorensen D. C. and Antoulas A. The sylvester equation and approximate balanced reduction. *Linear algebra and its applications*, 351:671–700, 2002.

Stein C. M. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.

Steinwart I., Hush D. R., Scovel C., and others . Optimal rates for regularized least squares regression. In *COLT*, 2009.

Stoica P. and Moses R. L. *Introduction to spectral analysis*, volume 1. Prentice hall Upper Saddle River, 1997.

Suykens J. A. and Vandewalle J. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

Swindlehust A., Roy R., Ottersten B., and Kailath T. A subspace fitting method for identification of linear state-space models. *IEEE transactions on automatic control*, 40(2):311–316, 1995.

Sznaier M. and Camps O. A rank minimization approach to trajectory (in) validation. In *American Control Conference (ACC), 2011*, pages 675–680. IEEE, 2011.

Tether A. Construction of minimal linear state-variable models from finite input-output data. *IEEE Transactions on Automatic Control*, 15(4):427–436, 1970.

Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

Tibshirani R. J. Degrees of freedom and model search. *arXiv preprint arXiv:1402.1920*, 2014.

Tikhonov A. N. and Arsenin V. Y. Solutions of ill-posed problems. 1977.

Tipping M. E. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

Tjarnstrom F. and Ljung L. Using the bootstrap to estimate the variance in the case of undermodeling. *IEEE transactions on automatic control*, 47(2):395–398, 2002.

Tjärnström F. Variance analysis of l 2 model reduction when undermodeling?the output error case. *Automatica*, 39(10):1809–1815, 2003.

Tjärnström F. and Ljung L. L 2 model reduction and variance reduction. *Automatica*, 38(9):1517–1530, 2002.

Toplis B. and Pasupathy S. Tracking improvements in fast rls algorithms using a variable forgetting factor. *IEEE Transactions on acoustics, speech, and signal processing*, 36(2):206–227, 1988.

Tóth R., Sanandaji B. M., Poolla K., and Vincent T. L. Compressive system identification in the linear time-invariant framework. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 783–790. IEEE, 2011.

Tse E. and Weinert H. Structure determination and parameter identification for multivariable stochastic linear systems. *IEEE Transactions on Automatic Control*, 20 (5):603–613, 1975.

Twomey S. Introduction to the mathematics of inversion in remote sensing and indirect measurements. 1977.

Utschick W. Tracking of signal subspace projectors. *IEEE Transactions on Signal Processing*, 50(4):769–778, 2002.

van der Vaart A. W. and van Zanten J. H. Rates of contraction of posterior distributions based on gaussian process priors. *The Annals of Statistics*, pages 1435–1463, 2008.

Van Gestel T., Suykens J. A., Van Dooren P., and De Moor B. Identification of stable models in subspace identification by using regularization. *IEEE Transactions on Automatic Control*, 46(9):1416–1420, 2001.

Van Overschee P. Subspace identification: theory, implementation, application. 1995.

Van Overschee P. and De Moor B. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, 1993.

Van Overschee P. and De Moor B. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

Van Overschee P. and De Moor B. Choice of state-space basis in combined deterministic-stochastic subspace identification. *Automatica*, 31(12):1877–1883, 1995a.

Van Overschee P. and De Moor B. A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864, 1995b.

Van Overschee P. and De Moor B. Continuous-time frequency domain subspace system identification. *Signal Processing*, 52(2):179–194, 1996.

Van Overschee P. and De Moor B. *Subspace identification for linear systems: Theory?Implementation?Applications.* Springer Science & Business Media, 2012.

Van Zee G. and Bosgra O. Gradient computation in prediction error identification of linear discrete-time systems. *IEEE Transactions on Automatic Control*, 27(3):738–739, 1982.

Vapnik V. *Statistical learning theory*, volume 1. Wiley New York, 1998.

Verdult V. *Non linear system identification: a state-space approach.* Twente University Press, 2002.

Verdult V. and Verhaegen M. Subspace identification of multivariable linear parameter-varying systems. *Automatica*, 38(5):805–814, 2002.

Verhaegen M. A novel non-iterative mimo state space model identification technique. In *Preprints 9th IFAC/IFORS symposium on Identification and System parameter estimation, Budapest, Hungary*, pages 1453–1458, 1991.

Verhaegen M. Application of a subspace model identification technique to identify lti systems operating in closed-loop. *Automatica*, 29(4):1027–1040, 1993a.

Verhaegen M. Subspace model identification part 3. analysis of the ordinary output-error state-space model identification algorithm. *International Journal of control*, 58 (3):555–586, 1993b.

Verhaegen M. Identification of the deterministic part of mimo state space models given in innovations form from input-output data. *Automatica*, 30(1):61–74, 1994.

Verhaegen M. and Deprettere E. A fast, recursive mimo state space model identification algorithm. In *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, pages 1349–1354. IEEE, 1991.

Verhaegen M. and Hansson A. Nuclear norm subspace identification (n2sid) for short data batches. *IFAC Proceedings Volumes*, 47(3):9528–9533, 2014.

Verhaegen M. and Verdult V. *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.

Viberg M. Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31(12):1835–1851, 1995.

Viberg M. and Ottersten B. Sensor array processing based on subspace fitting. *IEEE Transactions on signal processing*, 39(5):1110–1121, 1991.

Viberg M., Ottersten B., Wahlberg B., and Ljung L. A statistical perspective on state-space modeling using subspace methods. In *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, pages 1337–1342. IEEE, 1991.

Viberg M., Wahlberg B., and Ottersten B. Analysis of state space system identification methods based on instrumental variables and subspace fitting. *Automatica*, 33 (9):1603–1616, 1997.

Vidyasagar M. and Karandikar R. L. A learning theory approach to system identification and stochastic adaptive control. In *Probabilistic and Randomized Methods for Design under Uncertainty*, pages 265–302. Springer, 2006.

Wahba G. *Spline models for observational data*. CBMS-NSF regional conference series in applied mathematics. Society for industrial and applied mathematics, Philadelphia, 1990. ISBN 0-89871-244-0. URL http://opac.inria.fr/record=b1080403. Based on a series of 10 lectures at Ohio State University at Columbus, Mar. 23-27, 1987.

Wahba G. Bayesian confidence intervals for the cross-validated smoothing spline. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 133–150, 1983.

Wahlberg B. Estimation of autoregressive moving-average models via high-order autoregressive approximations. *Journal of Time Series Analysis*, 10(3):283–299, 1989a.

Wahlberg B. Model reductions of high-order estimated models: the asymptotic ml approach. *International Journal of Control*, 49(1):169–192, 1989b.

Wahlberg B. System identification using laguerre models. *IEEE Transactions on Automatic Control*, 36(5):551–562, 1991.

Wang Y. and Wahba G. Bootstrap confidence intervals for smoothing splines and their comparison to bayesian confidence intervals. *Journal of Statistical Computation and Simulation*, 51(2-4):263–279, 1995.

Wellstead P. and Rojas R. Instrumental product moment model-order testing: extensions and application. *International Journal of Control*, 35(6):1013–1027, 1982.

Wellstead P. E. Non-parametric methods of system identification. *Automatica*, 17(1): 55–69, 1981.

Weyer E. Finite sample properties of system identification of arx models under mixing conditions. *Automatica*, 36(9):1291–1299, 2000.

Weyer E., Williamson R. C., and Mareels I. M. Finite sample properties of linear model identification. *IEEE Transactions on Automatic Control*, 44(7):1370–1383, 1999.

Whittle P. Estimation and information in stationary time series. *Arkiv för matematik*, 2(5):423–434, 1953.

Wipf D. P. Non-convex rank minimization via an empirical Bayesian approach. pages 914–923. AUAI Press, 2012.

Wipf D. P. and Nagarajan S. S. Iterative reweighted l1 and l2 methods for finding sparse solutions. *J. Sel. Topics Signal Processing*, 4(2):317–329, 2010.

Wipf D. P. and Nagarajan S. S. A new view of automatic relevance determination. In *Advances in neural information processing systems*, pages 1625–1632, 2008.

Wipf D. P., Rao B. D., and Nagarajan S. Latent variable bayesian models for promoting sparsity. *IEEE Transactions on Information Theory*, 57(9):6236–6255, 2011.

Woodside C. Estimation of the order of linear systems. *Automatica*, 7(6):727–733, 1971.

Wu Q., Ying Y., and Zhou D.-X. Learning rates of least-square regularized regression. *Foundations of Computational Mathematics*, 6(2):171–192, 2006.

Yang B. Projection approximation subspace tracking. *IEEE Transactions on Signal processing*, 43(1):95–107, 1995.

Yao Y., Rosasco L., and Caponnetto A. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Ye J. On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93(441):120–131, 1998.

Yuan M., Cai T. T., and others . A reproducing kernel hilbert space approach to functional linear regression. *The Annals of Statistics*, 38(6):3412–3444, 2010.

Zadeh L. On the identification problem. *IRE Transactions on Circuit Theory*, 3(4): 277–281, 1956.

Zadeh L. A. From circuit theory to system theory. *Proceedings of the IRE*, 50(5): 856–865, 1962.

Zeiger H. p. and McEwen A. Approximate linear realizations of given dimension via ho's algorithm. *IEEE Transactions on Automatic Control*, 19(2):153–153, 1974.

Zhang K. and Kwok J. T. Clustered nyström method for large scale manifold learning and dimension reduction. *IEEE Transactions on Neural Networks*, 21(10):1576–1587, 2010.

Zhao Z. and others . Parametric and nonparametric models and methods in financial econometrics. *Statistics Surveys*, 2:1–42, 2008.

Zhu H., Williams C. K., Rohwer R., and Morciniec M. Gaussian regression and optimal finite dimensional linear models. 1997.

Zhu Y. and Backx T. *Identification of multivariable industrial processes: for simulation, diagnosis and control.* Springer Science & Business Media, 2012.

Zorzi M. and Chiuso A. A Bayesian approach to sparse plus low rank network identification. In *54th IEEE Conference on Decision and Control*, pages 7386–7391, Dec 2015.

Zou H. and Hastie T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.