KTH Electrical Engineering

On risk-coherent input design and Bayesian methods for nonlinear system identification

PATRICIO E. VALENZUELA PACHECO

Doctoral Thesis
Stockholm, Sweden 2016

KTH Royal Institute of Technology
School of Electrical Engineering
Department of Automatic Control
SE-100 44 Stockholm
SWEDEN

TRITA-EE 2016:193
ISSN 1653-5146
ISBN 978-91-7729-229-6

Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i elektro- och systemteknik fredagen den 10 februari 2017 klockan 10.00 i F3, Kungliga Tekniska högskolan, Lindstedtsvägen 26, Stockholm.

Tryck: Universitetsservice US AB

To Daniela and Fernanda,
the light in my life.

Abstract

System identification deals with the estimation of mathematical models from experimental data. As mathematical models are built for specific purposes, ensuring that the estimated model represents the system with sufficient accuracy is a relevant aspect in system identification. Factors affecting the accuracy of the estimated model include the experimental data, the manner in which the estimation method accounts for prior knowledge about the system, and the uncertainties arising when designing the experiment and initializing the search of the estimation method.

As the accuracy of the estimated model depends on factors that can be affected by the user, it is of importance to guarantee that the user decisions are optimal. Hence, it is of interest to explore how to optimally perform an experiment in the system, how to account for prior knowledge about the system and how to deal with uncertainties that can potentially degrade the model accuracy.

This thesis is divided into three topics. The first contribution concerns an input design framework for the identification of nonlinear dynamical models. The method designs an input as a realization of a stationary Markov process. As the true system description is uncertain, the resulting optimization problem takes the uncertainty on the true value of the parameters into account. The stationary distribution of the Markov process is designed over a prescribed set of marginal cumulative distribution functions associated with stationary processes. By restricting the input alphabet to be a finite set, the parametrization of the feasible set can be done using graph theoretical tools. Based on the graph theoretical framework, the problem formulation turns out to be convex in the decision variables. The method is then illustrated by an application to model estimation of systems with quantized measurements.

The second contribution of this thesis is on Bayesian techniques for input design and estimation of dynamical models. In regards of input design, we explore the application of Bayesian optimization methods to input design for identification of nonlinear dynamical models. By imposing a Gaussian process prior over the scalar cost function of the Fisher information matrix, the method iteratively computes the predictive posterior distribution based on samples of the feasible set. To drive the exploration of this set, a user defined acquisition function computes at every iteration the sample for updating the predictive posterior distribution. In this sense, the method tries to explore the feasible space only on those regions where an improvement in the cost function is expected. Regarding the estimation of dynamical models, this thesis discusses a Bayesian framework to account for prior information about the model parameters when estimating linear time-invariant dynamical models. Specifically, we discuss how to encode information about the model complexity by a prior distribution over the Hankel singular values of the model. Given the prior distribution and the likelihood function, the posterior distribution is approximated by the use of a Metropolis-Hastings sampler. Finally, the existence of the posterior distribution and the correctness of the Metropolis-Hastings sampler is analyzed and established.

As the last contribution of this thesis, we study the problem of uncertainty in system identification, with special focus in input design. By adopting a risk theoretical perspective, we show how the uncertainty can be handled in the problems arising in input design. In particular, we introduce the notion of coherent measure of risk and its use in the input design formulation to account for the uncertainty on the true system description. The discussion also introduces the conditional value at risk, which is a risk coherent measure accounting for the mean behavior of the cost function on the undesired cases. The use of risk coherent measures is also employed in application oriented input design, where the input is designed to achieve a prescribed performance in the intended model application.

Sammanfattning

Systemidentifiering handlar om uppskattningen av matematiska modeller från experimentell data. Eftersom de matematiska modeller är byggda för särskilda ändamål är en relevant aspekt inom systemidentifiering att den skattade modellen representerar systemet med tillräcklig noggrannhet. Nogrannheten i den skattade modellen påverkads av den experimentella data, hur beräkningsmetoden inkluderar förkunskaper om systemet, och den osäkerheten som uppstår vid experimentdesign och sökningens startvärdet i skattningsmetoden.

Eftersom noggrannheten i den skattade modellen kan påverkas av användaren är det viktigt att garantera optimala användarval. Därför är det viktigt att undersöka hur man kan genomföra ett optimalt experiment i systemet, hur man inkluderar förkunskaper om systemet och hur man handskas med osäkerhet som kan försämra modellen noggrannhet.

Denna avhandling är indelad i tre ämnen. Det första bidraget fokuserar på experimentdesign för identifiering av olinjära dynamiska modeller. Metoden designar en insignal som ett förverkligande av en stationär Markov process. Eftersom den sanna systembeskrivningen är osäker beaktar det resulterande optimeringsproblemet osäkerheten på det sanna värdet av parametrarna. Den stationära Markovprocessfördelningen är utformad över en föreskriven uppsättning marginella kumulativa fördelningsfunktioner i samband med stationära processer. Genom att begränsa insignalens alfabet till en ändlig uppsättningvärden kan insignalen parametriseras med hjälp av grafteoretiska verktyg. Baserat på grafteoretisk resultat visar det sig att problemformuleringen är konvex i beslutsvariablerna. Metoden illustreras sedan med ett program för att uppskatta system med kvantiserade mätningar.

Det andra bidraget med denna avhandling gäller bayesiska tekniker för experimentdesign och uppskattning av dynamiska modeller. I experimentdesign utforskar vi tillämpningen av bayesiska optimeringsmetoder för identifiering av olinjära dynamiska modeller. Genom att införa en Gaussianprocess prior över den skalära kostnadsfunktionen beräknar metoden iterativt den prediktiva aposteriorifördelningen baserad på sampel i den tillåtna mängden. För att uppdatera den prediktiva aposteriorifördelningen beräknas en användardefinierad anskaffningsfunktion på varje iteration sampel i uppsättningen. Metoden undersöker endast de områden där en förbättring av kostnadsfunktionen förväntas. I uppskattningen av dynamiska modeller diskuterar denna avhandling en bayesisk metod för att redogöra för förhandsinformation om modellparametrarna vid skattning av linjära tidsinvarianta dynamiska modeller. Här diskuterar vi hur man kan koda information om modellens komplexitet genom en tidigare fördelning över Hankel singulära värden av modellen. Med tanke på den tidigare fördelning och sannolikhetsfunktionen approximeras aposteriorifördelningen genom användning av en Metropolis-Hastings sampler. Dessutom etableras förekomsten av den aposteriorifördelningen och riktigheten av Metropolis-Hastings sampler.

I det sista bidraget med denna avhandling fokuserar vi på problemet med osäkerhet i systemidentifiering med särskilt fokus på experimentdesign. Genom att anta ett riskteoretiskt perspektiv visar vi hur osäkerheten kan hanteras

i de problem som uppstår i experimentdesign. I synnerhet introducerar vi begreppet sammanhängande riskmått och dess användning i experimentdesign formuleringen för att ta hänsyn till osäkerheten på den sanna systembeskrivningen. I diskussionen införs också det förväntad kortsiktig förlust, vilket är ett sammanhängande riskmått som står för det genomsnittliga utfallet av kostnadsfunktionen över de oönskade fallen. Med hjälp av sammanhängande riskmått analyseras även applikationsorienterad experimentdesign, där insignalen är utformad för att uppnå en föreskriven prestanda i den avsedda modellanvändningen.

Acknowledgements

It is very difficult for me to express my profound gratitude to my family, friends and colleagues in a couple of pages. This thesis comprises the love and support of many people I had the pleasure to meet along my work in electronics and engineering for more than fifteen years. I will do my best in these lines to acknowledge everyone, and please forgive me if someone is missing.

I would like to express my sincere gratitude to my Ph.D. advisors: Professor Cristian Rojas and Professor Håkan Hjalmarsson. I thank you deeply for the opportunity you gave me of continuing my studies towards a Ph.D. degree, as well as your advice and support during these years. I enjoyed very much these five years of joint work, and I am sure that this is the beginning of fruitful collaborations in the future. Thank you very much!

I also want to thank the people I had the pleasure to meet and collaborate with along my Ph.D. studies. My profound gratitude goes to Dr. Roland Hildebrand, Professor Thomas Schön, Professor Bo Wahlberg, Professor Brett Ninness, Professor Juan Carlos Agüero, Professor Ricardo Rojas, Professor Alessandra Parisio, Dr. Mariette Annergren, Dr. Boris Godoy, Dr. Johan Dahlin, Tekn. Lic. Afrooz Ebadat, Tekn. Lic. Niklas Everitt, Johan Bjurgert, Matías Müller and Rodrigo González. Thank you very much!

This thesis has received many valuable suggestions from people who volunteer to read early versions of this manuscript. Many thanks to Rodrigo González, Matías Müller, Robert Mattila, Afrooz Ebadat, Riccardo Sven Risuleo, Mohamed Rasheed Abdalmoaty, Demia Della Penda, Niklas Everitt and Håkan Terelius for the time you spent reading this thesis, and for the feedback you gave me to improve the document, I am very indebted to you!

I want to thank also the administrators Anneli Ström, Hanna Holmqvist, Kristina Gustafsson, Gerd Franzon, Silvia Cárdenas Svensson and Karin Karlsson for all the support I received from you along these years, and for all the good times (with a special mention to the episodes of *Lunch with Hanna*), thank you very much!

During my time in the Department I met many nice people, that contributes to make our working place even more enjoyable. I appreciate every conversation we have had, and all your support through these years. I would like to express my gratitude to all the former and present colleagues at the Department of Automatic Control, you make the Department an awesome place! I want to thank deeply to

Mohamed Rasheed Abdalmoaty, Mariette Annergren, Per Hägg, Christian Larsson, Niklas Everitt, Niclas Blomberg, Giulio Bottegal, Afrooz Ebadat, Matías Müller, Riccardo Sven Risuleo, and Miguel Ramos Galrinho for all the good times we have spent working together in the system identification group; Antonio Adaldo, Martin Jakobsson, Kuo-Yun Liang, Matías Müller, Marco Molinari, Olle Trollberg, André Teixeira, Sadegh Talebi, Pedro Lima, and Demia Della Penda for making our office a nice working place, and all the time we shared talking about non-research topics; Damiano Varagnolo, Themistoklis Charalambous, Winston García, Farhad Farokhi, Euhanna Ghadimi, Bart Besselink, P.G. di Marco, Martin Andreasson, Meng Guo, Stefan Magureanu, Arda Aytekin, Valerio Turri, and Burak Demirel for many interesting conversations; and Olle Trollberg and Håkan Terelius for being the best friends one can think of. Thanks a lot to all of you!

I cannot continue here without expressing my profound gratitude to all our friends here in Sweden. You have made this country a second home to Daniela, Fernanda and me, becaming our swedish family. I want to thank deeply to Awa and Mats Danielsson and their family, Fernando de Hoyos and Lupita Nava, Hilda "*tita*" Reyes, Hilda González and Roland Karlsson, Pilar Karlsson and Omid Rostam Khesal, Yasna Acevedo and her family, Camila Chávez and her family, and Matías Müller. Thank you for being there every time we need support and for all the good times we have shared. We love you all!

I also want to thank deeply the academics in the Department of Electronic Engineering in my home university, Professor Mario Salgado, Professor Eduardo Silva (rest in peace) and Professor Ricardo Rojas, for guiding me in my first steps in the research field, and for encouraging me to continue my studies towards a Ph.D. degree. This work is also yours for all the support I received from you. Thank you very much!

The step of doing a Ph.D. would not be possible without the support of many friends in Chile. I am really indebted with every one of you guys! My profound gratitude goes specially to Claudia Cortez, Marisol Vera, Alfred Rauch, Mauricio Moya, Cristian Carrasco, Felipe López, Sebastián "*el testigo*" Pulgar, Matías García, Gabriela Leal, Nelson López and Marcela Cubillos, Ramón Delgado, Rocío Guerra and their daughters, Diego Carrasco, Pedro Riffo, Valeria Araya and their daughter, Iván Velásquez, Verónica Contreras and their children, Francisco Arredondo and Lucrecia Aedo, Stjpe Halat, and Francisco Vargas. I really appreciate all the support I received from all of you over the years, thank you very much!

The love and support of our parents and relatives have been always present during these years. Your permanent company has contributed to reach this point. I am very indebted to our parents Germán, Patricio, Magda and Brígida, as well to our siblings Cristian and his family: Johana, Aiyana and Elías, Valentina and Gabriela, Alex and Claudia, Rosario and Claudia. My gratitude goes also to Raúl Díaz, Teresa Ojeda and Mauricio Díaz (rest in peace), who are beyond a friendship: they became part of my family, and their love is always present. We love you!

Finally, I would like to thank deeply the unconditional love and support of the most important persons in my life: my wife Daniela and our daughter Fernanda. Your love has made this journey enjoyable and your support has been crucial to overcome the difficulties along these years. This thesis is entirely yours. I love you!

Patricio E. Valenzuela
Stockholm, Sweden. December 2016.

Contents

Notation

Symbol	Definition
$\delta(x)$	Dirac delta at $x = 0$.
$(\cdot)^\top$	Transpose operator.
$\#\mathcal{X}$	Cardinality of the set $\mathcal{X}$.
$\det(\cdot)$	Determinant operator.
$\exp\{x\}$	Exponential function: e^x.
$\lambda_{\min}(\cdot)$	Minimum eigenvalue.
$\mathbf{1}_X$	Indicator function: $\mathbf{1}_X = 1$ if X is true; 0 otherwise.
$\operatorname{erf}(x)$	Error function: $\frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du$.
$\operatorname{supp}(p)$	Support of the function p: $\overline{\{x \in \operatorname{dom}(p) \colon p(x) \neq 0\}}$.
$\operatorname{tr}\{\cdot\}$	Trace operator.
$\succeq$, $\preceq$	Matrix inequalities in the positive semidefinite sense.
$\succ$, $\prec$	Matrix inequalities in the positive definite sense.
q	Time shift operator, $q\, u_t = u_{t+1}$.
$x_{1:N}$	$(x_1, \ldots, x_N)$.
P	Cumulative distribution function (cdf).
p	Probability density function (pdf).
$\mathbf{E}\{\cdot\}$	Expectation operator.
$\operatorname{Cov}(x)$	Covariance matrix: $\mathbf{E}\left\{(x - \mathbf{E}\{x\})(x - \mathbf{E}\{x\})^\top\right\}$.
$\operatorname{Var}(x)$	Variance: $\mathbf{E}\left\{(x - \mathbf{E}\{x\})^2\right\}$ (x scalar random variable).
$\chi^2_\alpha(n)$	α-percentile of the χ^2-distribution with n degrees of freedom.
$\mathcal{N}(x, \Sigma)$	Gaussian distribution with mean x and covariance matrix $\Sigma \succeq 0$.
$\mathcal{U}[a, b]$	Uniform distribution over $[a, b]$, with $a < b$.
$\xrightarrow{d}$	Convergence in distribution.
$x \sim p(x)$	x is distributed according to $p(x)$.
$\mathbf{P}\{\cdot\|\cdot\}$	Conditional probability measure.
$\mathbf{P}\{\cdot\}$	Probability measure.
$\mathcal{P}$	Set of n_m-dimensional marginal cdfs of stationary processes.
$\mathcal{P}_\mathcal{C}$	Set of n_m-dimensional marginal probability mass functions of stationary processes.
$\mathcal{V}_{\mathcal{P}_\mathcal{C}}$	Set of all vertices of $\mathcal{P}_\mathcal{C}$.

$\mathcal{E}$	Set of edges of a graph.
$\mathcal{V}$	Set of nodes of a graph.
$\mathcal{G}_{\mathcal{V}}$	Graph with nodes in $\mathcal{V}$.
$\mathcal{G}_{\mathcal{C}^n}$	n-dimensional de Bruijn graph with symbols in $\mathcal{C}^n$.
$\mathcal{A}_r$	Set of ancestors of a node r.
$\mathcal{D}_r$	Set of descendants of a node r.
$\mathcal{M}$	Model set.
$\mathbf{Z}_T$	Data set composed of T samples.
Θ	Set of model parameters.
θ	Parameter to be estimated.
θ_0	True parameter in Θ.
$\hat{\theta}_T$	Estimated parameter based on the data set $\mathbf{Z}_T$.
$\hat{y}_{t\|t-1}(\theta)$	Mean square optimal one-step ahead predictor given $\mathbf{Z}_{t-1}$.
∇_x	Gradient operator with respect to x.
∇_x^2	Hessian operator with respect to x.
$\mathcal{S}_T(u_{1:T}, \theta_0)$	Score function: $\nabla_\theta \log p_\theta(y_{1:T}\|u_{1:T})\|_{\theta=\theta_0}$.
$\|z\|$	Magnitude of the complex number z.
$\lfloor x \rfloor$	Closest integer to the real number x from below.
$\mathbb{C}$	Complex set.
$\mathbb{C}^{r\times s}$	Set of $r \times s$ matrices with complex entries.
$\mathbb{D}$	$\{z \in \mathbb{C}\colon \|z\| < 1\}$.
$\mathbb{N}$	Set of natural numbers.
$\mathbb{R}$	Real set.
$\mathbb{R}^+$	$\{x \in \mathbb{R} : x > 0\}$.
$\mathbb{R}_0^+$	$\{x \in \mathbb{R} : x \geq 0\}$.
$\mathbb{R}^n$	Set of n-dimensional vectors with real entries.
$\mathbb{R}^{r\times s}$	Set of $r \times s$ matrices with real entries.
$\mathbb{Z}$	Set of integer numbers.
$\lambda_i(A)$	i-th eigenvalue of $A \in \mathbb{C}^{n\times n}$, where $\|\lambda_i(A)\| \geq \|\lambda_{i+1}(A)\|$.
$\|G\|_2$	2-norm of the transfer function G: $\sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi} \|G(e^{i\omega})\|^2 \, d\omega}$.

Abbreviations

Abbreviation	Definition
AOID	Application oriented input design.
APF	Auxiliary particle filter.
AR	Autoregressive.
ARMA	Autoregressive moving average.
ARMAX	Autoregressive moving average with exogenous input.
ARX	Autoregressive with exogenous input.
cdf	Cumulative distribution function.
CRLB	Cramér-Rao lower bound.
CVaR	Conditional value at risk.
EI	Expected improvement.
EM	Expectation maximization.
FIM	Fisher information matrix.
FIR	Finite impulse response.
FL	Fixed-lag.
GP	Gaussian process.
GPO	Gaussian process optimization.
IS	Importance sampling.
LGSS	Linear Gaussian state space.
LMI	Linear matrix inequality.
LTI	Linear time invariant.
MA	Moving average.
MAP	Maximum a posteriori.
MCMC	Markov chain Monte Carlo.
MH	Metropolis-Hastings.

MIMO	Multiple-input multiple-output.
ML	Maximum likelihood.
MPC	Model predictive control.
NL	Nonlinear.
NOE	Nonlinear output error.
NSSM	Nonlinear state space model.
OE	Output error.
pdf	Probability density function.
PEM	Prediction error method.
PF	Particle filter.
pmf	Probability mass function.
SIS	Sequential importance sampling.
SISO	Single input single output.
SMC	Sequential Monte Carlo.
SSM	State space model.
wss	wide sense stationary.

Chapter 1

Introduction

Most of the questions addressed by philosophers, scientists and engineers are related to understanding nature or modifying its properties. To properly address these questions, certain level of knowledge about the physical laws governing nature is needed. Depending on the purpose of the study, the level of knowledge is adjusted to the level of detail required to understand the phenomena. For example, in fluid mechanics, nature is described by its macroscopic properties (such as pressure and mass flow) [6], while in quantum mechanics an atomic description of nature is required [57].

The physical laws describing a phenomenon are usually condensed by a set of mathematical equations which govern the variables of interest. The mathematical equations are then an abstraction of the phenomenon under study. In the following, the mathematical abstraction of the phenomenon will be referred to as a *model* of the phenomenon.

A question raising here is how a model can be obtained in practice. There are two main approaches to address this question. The first approach, known as *physical modelling*, is to construct a model based on the physical laws governing the phenomenon. The second approach, referred to as *system identification*, is to build a model based on data collected from the phenomenon. The second approach is the focus of the present thesis.

In the following sections, we introduce the elements needed to understand the principles of system identification and the main contributions of this thesis. We start in Section 1.1 with a brief review of the main elements in system identification. We then continue in Section 1.2 by stating some asymptotic results in system identification. Based on the previous sections, we introduce in Section 1.3 the problem of input design for dynamical models. In Section 1.4, we consider Bayesian methods in system identification, and Section 1.5 extends the discussion to the role of uncertainty in system identification. Section 1.6 summarizes the contributions of this thesis, and Section 1.7 presents the outline of this thesis and the associated publications. Finally, Section 1.8 presents publications not covered in this thesis.

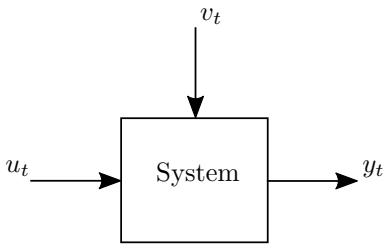

Figure 1.1: Block diagram of a system with input u_t, disturbance v_t, and output y_t.

The content of this chapter is partly based on [186].

1.1 System identification

System identification is concerned with building mathematical models based on data available from the phenomenon of interest, known as the *system*. A typical block diagram of a system is depicted in Figure 1.1. In this figure, $u_t \in \mathbb{R}^{n_u}$ represents the variables that can be affected to manipulate the system (referred to as *input*). The signal $v_t \in \mathbb{R}^{n_v}$ corresponds to variables affecting the system, but that cannot be manipulated (referred to as *disturbances*). Nevertheless, measurements of a subset of v_t can be available. Finally, $y_t \in \mathbb{R}^{n_y}$ denotes the variables we have access from the system (referred to as *output*). The index t in the tuple (y_t, v_t, u_t) refers to the set where the signals evolve, e.g., time or space. In the context of this thesis, the index t refers to time.

In addition to the system, we require three more elements to build a model in system identification [180, p. 9]:

- the data set,
- the model set, and
- the identification method.

These items are explained next.

Data set

In order to estimate a model, we require a set of input-output data from the system. This data set can also contain information regarding the measured disturbances. Following the notation in Figure 1.1, the data set is defined as $\mathbf{Z}_T := \{(y_t,\, v_t^m,\, u_t)\}_{t=1}^{T}$, where v_t^m is a variable containing the measured disturbances.

The model set

To build a model for the system, we also need to define the mathematical structure (parameterized by $\theta \in \Theta$) explaining the relations between the input, disturbance and output. Hence, the parameterized mathematical structure covers the possible behaviors of the system to be modelled. We refer to the set of all mathematical structures given by all θ in Θ as the *model set*, which will be denoted by $\mathcal{M}$. The choice of the model set is based on prior information available about the system, on information obtained from the data set using preprocessing techniques, or a combination of both approaches.

Remark 1.1 *(Model set and model structure) In the literature, the terms model set and model structure are introduced in different manners. For example, in [180, p. 9] both terms are used equivalently to refer to the class of models parameterized by $\theta \in \Theta$. On the other hand, in [122, pp. 107–108] the model set refers to the set of candidate models, while the model structure corresponds to the differentiable mapping parameterizing the model set by $\theta \in \Theta$ (cf. [122, Definition 4.3]).*

In the next examples we illustrate some common choices for the model set.

Example 1.1 *(Linear time invariant models) If the system is working locally around an equilibrium point, then it is reasonable to consider that it can be described by a model in the set*

$$\mathcal{M} = \{(G_\theta, H_\theta) \colon \theta \in \Theta\}\,, \tag{1.1}$$

where the model dynamics are given by

$$y_t = G_\theta(q)u_t + H_\theta(q)e_t \tag{1.2}$$

and $\Theta \subseteq \mathbb{R}^{n_\theta}$ is the set of parameters. $G_\theta(q)$ and $H_\theta(q)$ are real rational functions in the time shift operator q (i.e., $q\,u_t = u_{t+1}$), parameterized by θ. Here, $\{e_t\}$ ($e_t \in \mathbb{R}^{n_y}$) is white noise of zero mean and finite variance. In this case, the disturbance v_t in Figure 1.1 can be described as

$$v_t = H_{\theta_0}(q)e_t\,, \tag{1.3}$$

for some $\theta_0 \in \Theta$. In this example, the system in Figure 1.1 is assumed to have additive output disturbances.

Remark 1.2 *(Undermodeling) A common assumption about the model set $\mathcal{M}$ is that there exists at least one $\theta_0 \in \Theta$ such that the model evaluated at θ_0 describes the true system. If this condition is not fulfilled, then we say that the system is* undermodeled *by $\mathcal{M}$. In this thesis we assume that $\mathcal{M}$ contains the exact description of the system, i.e., there is no undermodeling.*

The set of linear time invariant (LTI) models in Example 1.1 covers a wide class of structures. By suitable assumptions on $G_\theta(q)$ and $H_\theta(q)$, the set $\mathcal{M}$ in

Example 1.1 can represent structures such as moving average (MA), autoregressive (AR), autoregressive with exogenous input (ARX), autoregressive moving average (ARMA), and autoregressive moving average with exogenous input (ARMAX), among others [122].

Example 1.2 *(Nonlinear state space models (NSSM)) Sometimes the linear assumption introduced in Example 1.1 is too restrictive. For example, the system can operate in regions where the nonlinearities cannot be neglected.*

There are several alternatives to model nonlinear systems. One of the most general model sets is given in terms of a nonlinear state space description [171]. A nonlinear state space model set is defined as

$$\mathcal{M} = \{(f_\theta, g_\theta, \mu_\theta) \colon \theta \in \Theta\} \ . \tag{1.4}$$

The model dynamics for the nonlinear state space model are given by

$$x_{t+1}|x_t \sim f_\theta(x_{t+1}|x_t, u_t)\,, \tag{1.5a}$$
$$y_t|x_t \sim g_\theta(y_t|x_t, u_t)\,, \tag{1.5b}$$
$$x_0 \sim \mu_\theta(x_0)\,, \tag{1.5c}$$

where $\Theta \subseteq \mathbb{R}^{n_\theta}$ is the set of parameters as in Example 1.1 and $\sim$ denotes distributed according to*. f_θ and g_θ are conditional probability density functions (pdf) parameterized by θ. In this example, we introduce the variable $x_t \in \mathbb{R}^{n_x}$ (referred to as* state*) accounting for internal information about the system.*

We note that the disturbance v_t in Figure 1.1 is considered via the pdfs f_θ and g_θ defining how the stochastic processes $\{x_t\}$ and $\{y_t\}$ are generated.

The model set in Example 1.2 includes also linear time invariant models with static nonlinearities, known as *Wiener-Hammerstein models* [122]. On the other hand, if we assume that the disturbance v_t acts on y_t as additive white noise, then the model set in Example 1.2 is reduced to the nonlinear output-error (NOE) class.

The model sets introduced in the previous examples are not the only ones existing in the literature. Indeed, it is possible to define model sets with time varying structure [122], linear parameter-varying [185], neural networks [176, 212], and models based on kernel estimators [135], among others.

The identification method

Once the data $\mathbf{Z}_T$ is obtained and the model set $\mathcal{M}$ is specified, the goal is then to determine the model in $\mathcal{M}$ that best explains the data $\mathbf{Z}_T$ in a prescribed sense. The technique employed to choose a model from $\mathcal{M}$ given $\mathbf{Z}_T$ is known as the *identification method*. There is a vast literature on identification methods, including least squares [77], instrumental variables [178, 179], subspace techniques [106, 192, 193], and kernel based methods [41, 146, 147], among others.

In this thesis, we work with three identification methods:

1. the maximum likelihood method,
2. the prediction error method, and
3. Bayesian methods.

The maximum likelihood method

The maximum likelihood (ML) method is one of the most attractive identification procedures due to its appealing statistical properties [67]. The ML method is based on the distribution of the measurements $y_{1:T} := (y_1, \ldots, y_T)$ given $u_{1:T} := (u_1, \ldots, u_T)$, which is parameterized by $\theta \in \Theta$. The objective is to find the estimated parameter $\hat{\theta}_T$ that best explains the measurements $y_{1:T}$. If we define $p_\theta(y_{1:T}|u_{1:T})$ as the pdf associated with $y_{1:T}$ given $u_{1:T}$, then the ML estimate $\hat{\theta}_T$ is

$$\hat{\theta}_T = \arg\max_{\theta\in\Theta} p_\theta(y_{1:T}|u_{1:T})\,. \tag{1.6}$$

The expression (1.6) has an intuitive interpretation: $\hat{\theta}_T \in \Theta$ is such that the observed event $y_{1:T}$ becomes "as likely as possible" [122].

For numerical reasons, the logarithm of the pdf $p_\theta(y_{1:T}|u_{1:T})$ is usually maximized instead of $p_\theta(y_{1:T}|u_{1:T})$. This quantity is called the *log-likelihood function*. Due to the monotonicity of the logarithm function, the solution

$$\hat{\theta}_T = \arg\max_{\theta\in\Theta} \log p_\theta(y_{1:T}|u_{1:T})\,, \tag{1.7}$$

is equal to the solution in (1.6). The expression (1.7) is usually prefered to (1.6) because products appearing in $p_\theta(y_{1:T}|u_{1:T})$ are converted into sums, and that the logarithm removes exponentials (when the density $p_\theta(y_{1:T}|u_{1:T})$ is in the exponential class [35]). Another reason is that the use of logarithms results in algorithms that are numerically more well-behaved [170].

To illustrate the computation of the log-likelihood function, we consider the following example:

Example 1.3 *(ML of a NSSM) Consider the NSSM described in Example 1.2. An important property associated with the stochastic models in* (1.4) *is the* Markov property*, i.e., the distribution of x_{t+1} and y_t given $\{x_k, u_k\}_{k=-\infty}^{t}$ equals their distribution given $\{x_t, u_t\}$.*

The likelihood function can be computed using the definition of conditional probability density functions as ([141])

$$p_\theta(y_{1:T}|u_{1:T}) = p_\theta(y_1|u_1)\prod_{t=2}^{T} p_\theta(y_t|y_{1:t-1}, u_{1:t})\,. \tag{1.8}$$

Taking the logarithm of (1.8)*, we obtain*

$$\log p_\theta(y_{1:T}|u_{1:T}) = \log p_\theta(y_1|u_1) + \sum_{t=2}^{T} \log p_\theta(y_t|y_{1:t-1}, u_{1:t})\,. \tag{1.9}$$

We use the Markov property of the model set (1.4) *to compute the pdfs in* (1.9) *as*

$$p_\theta(y_t|y_{1:t-1}, u_{1:t}) = \int_{\mathcal{X}_t} g_\theta(y_t|x_t, u_t) p_\theta(x_t|y_{1:t-1}, u_{1:t-1})\, dx_t\,, \tag{1.10}$$

$$p_\theta(x_t|y_{1:t}, u_{1:t}) = \frac{g_\theta(y_t|x_t, u_t) p_\theta(x_t|y_{1:t-1}, u_{1:t-1})}{p_\theta(y_t|y_{1:t-1}, u_{1:t})}\,, \tag{1.11}$$

$$p_\theta(x_{t+1}|y_{1:t}, u_{1:t}) = \int_{\mathcal{X}_t} f_\theta(x_{t+1}|x_t, u_t) p_\theta(x_t|y_{1:t}, u_{1:t})\, dx_t\,, \tag{1.12}$$

where $\mathcal{X}_t \subseteq \mathbb{R}^{n_x}$ *denotes the set of values for* x_t*. Equations* (1.10)-(1.11) *are known as the* measurement update*, and Equation* (1.12) *is known as the* time update*. Together, equations* (1.10)-(1.12) *can be employed to recursively compute the pdfs in the log-likelihood function* (1.9)*.*

Example 1.3 illustrates how the information available in the model set can be employed to compute the log-likelihood function. It is important to emphasize that, in general, equations (1.10)-(1.12) cannot be computed in closed form. An exception is when the system is linear and Gaussian, where we can recover the expressions for the Kalman filter [105] from (1.10)-(1.12). The reason is that the analytic solutions of the integrals (1.10) and (1.12) are only available for specific cases. When a closed-form expression is not available for the integrals (1.10) and (1.12), we need to rely on numerical methods for computing (1.10) and (1.12). This makes the optimization of (1.9) over Θ highly complex. An approach to solve this issue has been proposed in [171], where particle methods are employed to numerically compute the expectation-maximization algorithm [56, 131] to maximize (1.9). Other approaches to solve the identification problem for nonlinear models can be obtained by restricting to a specific class of models, such as block-oriented [83], and neural networks and fuzzy models [136, 177].

The prediction error method

Another technique to select a model in the set $\mathcal{M}$ given the data $\mathbf{Z}_T$ is the prediction error method (PEM) [122]. In this method, the estimated parameter $\hat{\theta}_T$ is obtained as

$$\hat{\theta}_T = \arg\min_{\theta\in\Theta} V_T(\theta)\,, \tag{1.13}$$

where

$$V_T(\theta) := \sum_{t=1}^{T} \ell(\varepsilon_t(\theta))\,, \tag{1.14}$$

$$\varepsilon_t(\theta) := y_t - \hat{y}_{t|t-1}(\theta)\,. \tag{1.15}$$

Here, $\hat{y}_{t|t-1}(\theta)$ denotes the mean square optimal one-step ahead predictor given $\{y_{1:t-1}, u_{1:t}\}$ and $\theta \in \Theta$, defined as

$$\hat{y}_{t|t-1}(\theta) := \mathbf{E}\{y_t|y_{1:t-1}, u_{1:t}\} = \int_{\mathcal{Y}_t} y_t \, p_\theta(y_t|y_{1:t-1}, u_{1:t}) \, dy_t \,, \tag{1.16}$$

where $\mathbf{E}\{\cdot\}$ is the expectation operator, $\mathcal{Y}_t \subseteq \mathbb{R}^{n_y}$ denotes the set of values for y_t, and the function $\ell \colon \mathbb{R}^{n_y} \to \mathbb{R}_0^+$ is an arbitrary positive function (typically defined as a quadratic function). We note that minimizing the prediction errors, $\varepsilon_t(\theta)$, is meaningful since the models are usually employed for prediction, as in control system synthesis. Normally the systems are stochastic, which means that the output of the system at time t cannot be exactly determined by the data up to time $t-1$. Therefore, it is valuable to know at time $t-1$ what the output y_t is likely to be in order to compute the appropriate control action [180].

The prediction error method has a number of benefits [123]:

- It can be applied to a wide spectrum of model parameterizations since only an expression for (1.16) is required.
- It gives models with excellent asymptotic properties, thanks to its kinship with the ML method (cf. Example 1.4 below).
- It can handle systems that operate in closed-loop (the input is partly determined via output feedback, when the data are collected) without additional modifications to the method [71]. This property is also enjoyed by the ML method.

To illustrate the connection between PEM and the ML method, we introduce the following example:

Example 1.4 *(PEM and ML method) Consider a single output system that can be written as*

$$y_t = \hat{y}_{t|t-1}(\theta_0) + e_t \,, \tag{1.17}$$

where $\{e_t\}$ is white noise, Gaussian distributed with zero mean and variance $\sigma^2(\theta_0)$, and $\hat{y}_{t|t-1}(\theta_0)$ is given in (1.16)*. Under the previous assumption, the log-likelihood function can be written as*

$$\log p_\theta(y_{1:T}|u_{1:T}) = -\frac{1}{2\,\sigma^2(\theta)} \sum_{t=1}^{T} \varepsilon_t^2(\theta) - \frac{T}{2} \log \sigma^2(\theta) + c \,, \tag{1.18}$$

where $\varepsilon_t(\theta)$ is defined in (1.15)*, and c is a term independent of θ. If we assume that $\varepsilon_t(\theta)$ and $\sigma^2(\theta)$ are independently parameterized in θ, then we can maximize over $\sigma^2(\theta)$ to obtain an expression in terms of $\varepsilon_t(\theta)$. This allows to rewrite* (1.18) *as*

$$\log p_\theta(y_{1:T}|u_{1:T}) = -\frac{1}{T} \sum_{t=1}^{T} \varepsilon_t^2(\theta) + c \,. \tag{1.19}$$

If we compare (1.19) *with* (1.14)*, we see that PEM is retrieved from ML when* $\{e_t\}$ *is Gaussian distributed white noise, and* $\ell(\varepsilon_t(\theta)) = T^{-1}\varepsilon_t^2(\theta)$*.*

In general, Equation (1.16) does not have a closed form expression. Except for linear and specific nonlinear models, expression (1.16) can only be computed numerically, e.g., using particle methods [171]. The next example shows the closed form expression for the optimal one-step ahead predictor in the linear case.

Example 1.5 *(Optimal one-step ahead predictor, linear case) Consider the model set* $\mathcal{M}$ *introduced in Example 1.1. We will compute its optimal one-step ahead predictor* $\hat{y}_{t|t-1}(\theta)$*, with* $\theta \in \Theta$*. To this end, we assume that* $\lim_{q\to\infty} H(q,\theta) = I$*. Under this assumption, we can rewrite any model in* $\mathcal{M}$ *as*

$$y_t = G_\theta(q)u_t + (H_\theta(q) - I)e_t + e_t\,. \tag{1.20}$$

The term $(H_\theta(q) - I)e_t$ *in* (1.20) *only contains information up to time* $t-1$*. Using* (1.2) *to compute* e_t *as a function of* y_t *and* u_t*, and inserting the result into* (1.20)*, we obtain*

$$y_t = H_\theta^{-1}(q)G_\theta(q)u_t + (I - H_\theta^{-1}(q))y_t + e_t\,. \tag{1.21}$$

The first two terms in the right-hand side of the equality in (1.21) *only depend on* $\{y_k,\, u_k\}_{k=1}^{t-1}$ *if* G_θ *is strictly proper (otherwise it is still known if* u_t *is deterministic). In addition, since* $\{e_t\}$ *is white noise, the best prediction of* e_t *given* $\mathbf{Z}_{t-1}$ *is* $\mathbf{E}\{e_t\} = 0$*. Therefore, the optimal one-step ahead predictor associated with the model set in Example 1.1 is*

$$\hat{y}_{t|t-1}(\theta) = H_\theta^{-1}(q)G_\theta(q)u_t + (I - H_\theta^{-1}(q))y_t\,. \tag{1.22}$$

The expression (1.22) *is a valid predictor if* $H_\theta^{-1}(q)G_\theta(q)$ *and* $H_\theta^{-1}(q)$ *are stable [122, 180].*

Remark 1.3 *When* $\{e_t\}$ *is white noise, the data set* $\mathbf{Z}_{t-1}$ *in Example 1.5 does not convey information about* e_t*. Due to this property,* e_t *is called the* innovation *of the process [122].*

A useful predictor is introduced in the following example:

Example 1.6 *(Optimal one-step ahead predictor, nonlinear state space models) Consider the nonlinear state space model introduced in Example 1.2. The optimal one-step ahead predictor associated with this model is*

$$\hat{y}_{t|t-1}(\theta) = \int_{\mathcal{Y}_t} y_t\, p_\theta(y_t|y_{1:t-1}, u_{1:t})dy_t\,, \tag{1.23}$$

where the pdf $p_\theta(y_t|y_{1:t-1}, u_{1:t})$ *is given by Equation* (1.10) *in Example 1.3.*

The limitations of PEM are that an expression for $\hat{y}_{t|t-1}(\theta)$ must be available, and that the optimal one-step ahead predictor $\hat{y}_{t|t-1}(\theta)$ must be stable. For general nonlinear models, $\hat{y}_{t|t-1}(\theta)$ can be difficult to compute. To circumvent this issue, it is possible to directly parameterize the model in terms of $\hat{y}_{t|t-1}(\theta)$ [122].

Remark 1.4 *Even though PEM has many advantages, it also has severe difficulties when numerically solving* (1.13)*. The main issue is that the model can be nonlinear in the parameters, leading frequently to nonconvex problems (e.g., when estimating a LTI Box-Jenkins model). When the problem* (1.13) *is nonconvex, there is a need for good initialization of the used numerical solver in order to ensure convergence to the global optimum.*

How to circumvent the nonconvex formulation (1.13) *has been of interest in recent years. In that line, the weighted null-space fitting method has recently been proposed [75], providing an alternative to PEM.*

Bayesian methods

Another formulation of the parameter estimation problem in system identification is given by a Bayesian framework [73, 98, 208]. In Bayesian methods, we do not only need the data set $\mathbf{Z}_T$ and the likelihood function $p_\theta(y_{1:T}|u_{1:T})$, but we also need to specify a prior pdf over the parameter space Θ. In this sense, the parameter θ is assumed to be a random variable, where the prior pdf (denoted by π_Θ) reflects the knowledge (or belief) we have prior to obtaining the data $\mathbf{Z}_T$.

The objective in Bayesian methods is to compute the posterior pdf $p(\theta|\mathbf{Z}_T)$, which is then used to construct an estimator $\hat{\theta}_T$. Based on Bayes' rule, it is possible to write the posterior pdf of θ as

$$p(\theta|\mathbf{Z}_T) = \frac{p_\theta(y_{1:T}|u_{1:T})\pi_\Theta(\theta)}{K_\Theta}\,, \tag{1.24}$$

where K_Θ is a normalization constant given by

$$K_\Theta = \int_\Theta p_\theta(y_{1:T}|u_{1:T})\pi_\Theta(\theta)d\theta\,. \tag{1.25}$$

We note from (1.24) that the posterior distribution of θ is proportional to the product of the likelihood function and the prior distribution π_Θ. Based on (1.24), it is possible to compute a point estimate $\hat{\theta}_T$ in different manners. Typical Bayesian estimators are

$$\hat{\theta}_T = \arg\max_{\theta\in\Theta} p(\theta|\mathbf{Z}_T)\,, \tag{1.26}$$

known as the *maximum a posteriori* (MAP) estimator, and

$$\hat{\theta}_T = \mathbf{E}\left\{\theta\,|\mathbf{Z}_T\right\}\,, \tag{1.27}$$

referred to as the *posterior mean* estimator.

In the light of (1.26) and (1.27), we see that the key ingredient for Bayesian estimators is the posterior pdf $p(\theta|\mathbf{Z}_T)$. However, the posterior distribution is often unavailable in closed form or even infeasible to sample from. The issue of approximating $p(\theta|\mathbf{Z}_T)$ has been addressed in the literature by the use of Markov chain Monte Carlo (MCMC) methods [160]. We refer to Chapter 7 for more details about the use of MCMC methods in Bayesian estimation.

1.2 Asymptotic analysis

Given an estimator associated with an identification method, we would like to understand its properties. In system identification, unbiasedness and consistency are two important properties expected from an estimator. An estimator $\hat{\theta}_T$ is said to be *unbiased* if $\mathbf{E}\{\hat{\theta}_T\} = \theta_0$. When the estimator satisfies $\lim_{T\to\infty} \hat{\theta}_T = \theta_0$ with probability 1, then the estimator is said to be *consistent*. We note that unbiasedness is a property defined for all T, while consistency requires $T \to \infty$.

Another element analyzed in system identification is the *accuracy* associated with an identification method, i.e., the size of the variation of the identified model around its true value. In this subsection we briefly discuss the consistency and accuracy of the parameter estimates $\hat{\theta}_T$ as $T \to \infty$. To begin, we require the following result [47, 122]:

Lemma 1.1 *(Cramér-Rao bound) Let $\hat{\theta}_T$ be an unbiased estimator of θ. Assume that $p_{\theta_0}(y_{1:T}|u_{1:T})$ (the pdf of $y_{1:T}$) is defined for all $\theta_0 \in \Theta$, that for all values of $y_{1:T}$ where $p_\theta(y_{1:T}|u_{1:T}) > 0$ the expression*

$$\nabla_\theta \log p_\theta(y_{1:T}|u_{1:T}) \tag{1.28}$$

exists, and that

$$\left| \frac{\partial}{\partial \theta_i} \log p_\theta(y_{1:T}|u_{1:T}) \right| \tag{1.29}$$

is upper bounded by an integrable function over the set defined for $y_{1:T}$, for all $i \in \{1, \ldots, n_\theta\}$. In addition, suppose that $y_{1:T}$ can take values in a set whose boundaries do not depend on θ. Then

$$\mathbf{E}\left\{ \left[\hat{\theta}_T - \theta_0\right] \left[\hat{\theta}_T - \theta_0\right]^\top \right\} \succeq \{\mathcal{I}_F^T(u_{1:T}, \theta_0)\}^{-1}, \tag{1.30}$$

where

$$\begin{aligned} \mathcal{I}_F^T(u_{1:T}, \theta_0) &:= \mathbf{E}\left\{ \nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta_0} \nabla_\theta^\top \log p_\theta(y_{1:T}|u_{1:T})\big|_{\theta=\theta_0} \Big| u_{1:T} \right\} \\ &= -\mathbf{E}\left\{ \nabla_\theta^2 \log p_\theta(y_{1:T}|u_{1:T})\big|_{\theta=\theta_0} \Big| u_{1:T} \right\}. \end{aligned} \tag{1.31}$$

Lemma 1.1 states that the covariance of any unbiased estimator $\hat{\theta}_T$ cannot be smaller (in a positive semidefinite sense) than the inverse of $\mathcal{I}_F^T$, known as the *Fisher*

information matrix. We notice that the computation of $\mathcal{I}_F^T$ requires the knowledge of $\theta_0 \in \Theta$, which implies that the exact value of $\mathcal{I}_F^T$ might not be available. How to properly account for the lack of knowledge on θ_0 to compute or even maximize $\mathcal{I}_F^T$ is one question addressed in this thesis. We refer to Section 1.5, where a more detailed discussion about this aspect is presented.

Remark 1.5 *The quantity*

$$\mathcal{S}_T(u_{1:T}, \theta) := \nabla_\theta \log p_\theta(y_{1:T}|u_{1:T}) \tag{1.32}$$

is known as the score function. *The score function will be employed in Chapter 4 to compute the Fisher information matrix for nonlinear state space models.*

To continue, we introduce the following definition:

Definition 1.1 *(Efficient and asymptotically efficient estimators) An unbiased estimator $\hat{\theta}_T$ is said to be* efficient *if expression* (1.30) *holds with equality for all T. If*

$$\lim_{T\to\infty} \mathbf{E}\left\{T\left[\hat{\theta}_T - \theta_0\right]\left[\hat{\theta}_T - \theta_0\right]^\top\right\} = \{\overline{\mathcal{I}}_F(\theta_0)\}^{-1}, \tag{1.33}$$

holds and $\hat{\theta}_T$ is asymptotically unbiased, where

$$\overline{\mathcal{I}}_F(\theta_0) = \lim_{T\to\infty} \frac{1}{T}\mathcal{I}_F^T(u_{1:T}, \theta_0), \tag{1.34}$$

and $\mathcal{I}_F^T$ is given in Lemma 1.1, then $\hat{\theta}_T$ is said to be asymptotically efficient.

The estimators obtained by the ML method and PEM (for a particular ℓ and Gaussian innovations) are asymptotically efficient. From this perspective, it is interesting to analyze how the estimator $\hat{\theta}_T$ behaves as $T \to \infty$. The key result in this area was introduced in [47, 198] for the asymptotic distribution of maximum likelihood estimators obtained from independent observations:

Lemma 1.2 *(Consistency and asymptotic distribution of ML estimators) Suppose that the random variables $z_{1:T}$ are independent and identically distributed. Suppose also that the distribution of $z_{1:T}$ is given by p_{θ_0} for some value $\theta_0 \in \Theta$. Then, as $T \to \infty$, the ML estimator $\hat{\theta}_T$ tends to θ_0 with probability one, and $\sqrt{T}\left(\hat{\theta}_T - \theta_0\right)$ satisfies*

$$\sqrt{T}\left(\hat{\theta}_T - \theta_0\right) \xrightarrow{d} \mathcal{N}(0, \{\overline{\mathcal{I}}_F(\theta_0)\}^{-1}), \tag{1.35}$$

where $\overline{\mathcal{I}}_F(\theta_0)$ is given by (1.34).

The result in Lemma 1.2 shows that, as $T \to \infty$, the distribution of the random variable $\sqrt{T}(\hat{\theta}_T - \theta_0)$ tends to be normal with zero mean and covariance matrix given by the Cramér-Rao bound. We notice that the result in Lemma 1.2 is also true

when the ML method is applied to dynamical systems under some mild conditions. Moreover, under technical conditions, the result in Lemma 1.2 still holds for the estimator given by PEM when the white noise $\{e_t\}$ is Gaussian and ℓ is a quadratic function [122].

Remark 1.6 *The asymptotic distribution given in Lemma 1.2 does not necessarily imply that*

$$\mathrm{Cov}(\sqrt{T}\hat{\theta}_T) := T\,\mathbf{E}\left\{(\hat{\theta}_T - \mathbf{E}\{\hat{\theta}_T\})(\hat{\theta}_T - \mathbf{E}\{\hat{\theta}_T\})^\top\right\} \to \{\overline{\mathcal{I}}_F(\theta_0)\}^{-1} \text{ as } T \to \infty\,. \tag{1.36}$$

The result (1.36) *requires more technical conditions on the stochastic processes acting on the system [122, Appendix 9B]. In this thesis we assume that those conditions are fulfilled and hence*

$$\mathrm{Cov}(\hat{\theta}_T) \approx \frac{1}{T}\{\overline{\mathcal{I}}_F(\theta_0)\}^{-1}\,, \tag{1.37}$$

for sufficiently large T.

As we can see in (1.37), the covariance matrix of an asymptotically efficient estimator can be expressed in terms of the Cramér-Rao bound. At this point we can consider adjusting the covariance matrix of $\hat{\theta}_T$ to improve the accuracy of the parameter estimates. Indeed, since the Fisher information matrix (1.31) is conditioned on the input sequence $u_{1:T}$, it is possible to design the covariance matrix of $\hat{\theta}_T$ by designing $u_{1:T}$. This is the main objective of input design, described in the next section.

1.3 Input design

Optimal input design concerns the design of an excitation that maximizes the information in the data set $\mathbf{Z}_T$ about the system [1, 23, 45, 66, 70, 72, 82, 87, 132, 158, 209, 211]. The maximization is usually performed by optimizing a cost function related to the intended model application. Another standard choice is a scalar function of the Fisher information matrix $\mathcal{I}_F^T(u_{1:T}, \theta_0)$ [87, 102, 122], as it imposes a bound on the asymptotic covariance matrix of estimators. We denote this cost function by $h\colon \mathbb{R}^{n_\theta \times n_\theta} \times \Theta \to \mathbb{R}$. The cost function h is usually required to fulfill the following [21]:

Definition 1.2 *(Matrix convex function) A function $f : \mathbb{R}^{r\times r} \to \mathbb{R}$ is called* matrix convex *if and only if, for every two positive semidefinite matrices X, $Y \in \mathbb{R}^{r\times r}$, and for all $\lambda \in [0, 1]$,*

$$f(\lambda X + (1-\lambda)Y) \le \lambda f(X) + (1-\lambda) f(Y)\,. \tag{1.38}$$

More precisely, the function h is required to be a matrix convex function in its first argument for every $\theta \in \Theta$.

Table 1.1: Typical choices of h.

Optimality criterion	h
A-optimality	$\operatorname{tr}\left\{(\cdot)^{-1}\right\}$
D-optimality	$-\log\det(\cdot)$
E-optimality	$-\lambda_{\min}(\cdot)$
L-optimality	$\operatorname{tr}\left\{W(\cdot)^{-1}\right\}$

According to Definition 1.2, some suitable choices for h are $h(X) = -\log\det(X)$ (D-criterion), $h(X) = \operatorname{tr}\left\{X^{-1}\right\}$ (A-criterion), and $h(X) = -\lambda_{\min}(X)$ (E-criterion), among others. Table 1.1 summarizes the definitions commonly used for the cost function h [107].

By designing an optimal input sequence for identification we mean that, for a given data length T, we optimize the accuracy for the parameter estimates in a prescribed sense. Since the cost function is associated with $\mathcal{I}_F^T(u_{1:T}, \theta_0)$, by Remark 1.6 we conclude that the maximization of $\mathcal{I}_F^T(u_{1:T}, \theta_0)$ implies a reduction in the covariance matrix of $\hat{\theta}_T$. In practical applications, input design allows to reduce the time associated with the experiment to obtain a prescribed accuracy. To see this, we note that equation (1.37) implies that the covariance matrix of the parameter estimates decays as T^{-1}. Furthermore, if $\mathcal{I}_F^T(u_{1:T}, \theta_0)$ is optimized, then from equation (1.37) we conclude that we can reduce the number of samples required to achieve the desired accuracy for $\hat{\theta}_T$.

To illustrate the importance of input design, we consider a simple example:

Example 1.7 *(Fisher information matrix for an FIR model) Consider the finite impulse response (FIR) model*

$$y_t = \theta_1 u_t + \theta_2 u_{t-1} + e_t\,, \tag{1.39}$$

where $\theta = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}^\top \in \mathbb{R}^2$*, and* $\{e_t\}$ *is Gaussian white noise of zero mean and variance* σ^2*. We are interested in identifying* $\theta \in \mathbb{R}^2$ *by performing an experiment with* T *samples. The Fisher information matrix* $\mathcal{I}_F^T(u_{1:T}, \theta_0)$ *for the model* (1.39) *is*

$$\mathcal{I}_F^T(u_{1:T}, \theta_0) = \frac{1}{\sigma^2}\sum_{t=1}^{T-1}\begin{bmatrix} u_t^2 & u_t\,u_{t+1} \\ u_t\,u_{t+1} & u_{t+1}^2 \end{bmatrix}. \tag{1.40}$$

From (1.40) *we see that* $\mathcal{I}_F^T$ *depends on the input samples* $u_{1:T}$*. Therefore, the choice of the input for the experiment is crucial to identify* θ*. For example, if* $u_t = c$ *for a given* $c \in \mathbb{R}$*,* $c \neq 0$*, and all* t*, the matrix* $\mathcal{I}_F^T$ *becomes*

$$\mathcal{I}_F^T(u_{1:T}, \theta_0) = \frac{c(T-1)}{\sigma^2}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}, \tag{1.41}$$

which is singular. This implies that the elements in the parameter θ cannot be independently identified. Nevertheless, by using the constant input in (1.39)*, we have*

$$y_t = (\theta_1 + \theta_2)\, c + e_t\,. \tag{1.42}$$

Hence, a constant input signal can be employed to identify the DC gain of the model (1.39)*, which is the sum of the parameters.*

On the other hand, if we choose u_t as

$$u_t = \begin{cases} c, & t \text{ even}, \\ 0, & \text{otherwise}, \end{cases} \tag{1.43}$$

then $\mathcal{I}_F^T(u_{1:T},\, \theta_0)$ becomes

$$\mathcal{I}_F^T(u_{1:T},\, \theta_0) = \frac{c}{\sigma^2} \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix}, \tag{1.44}$$

where we assume that $T = 2K + 1$, for some positive integer K. For the periodic input (1.43)*, the Fisher information matrix* (1.44) *is nonsingular. This implies that the elements in the parameter vector θ can now be independently estimated.*

The optimal input design problem has been widely analyzed in the literature, with numerous results available. In the next subsection we provide a literature review of the main results in input design.

Literature review on input design

Most results on input design for dynamical systems have been developed for linear models [122]. The assumption of a linear model allows us to use convex optimization tools to solve the input design problem [86, 102, 117, 122, 167].

One possibility to solve the optimal input design problem for linear systems is by employing linear matrix inequalities (LMI) to characterize autocovariance functions associated with a feasible input spectrum [102, 117, 169, 197]. In [102] the input design problem is solved in the frequency domain, where the input spectrum is designed. To this end, [102] parameterizes the input spectrum using rational basis functions. This allows to obtain a convex problem in the decision variables, where quality constraints on the identified model and power constraints on the input signal can be included. A D-optimal multisine excitation is designed in [169], where the signal is employed to model physiological or electrochemical phenomena from spectroscopy measurements. In [197] the optimal input design is presented for finite impulse response (FIR) models, minimizing the uncertainty of the identified model while the variance of the input is kept as small as possible. A Markov chain approach is presented in [24, 25] to design inputs with amplitude constraints. The input sequence is assumed to be the output of a Markov chain, where the transition probability matrix is designed to maximize the information retrieved

from the experiment. However, the resulting problem is non-convex. We find the same problem for time domain gradient-based schemes in [86, 182], where only the convergence to local optima can be guaranteed.

In recent years, the interest on input design has been extended from linear to nonlinear systems. The main issue here is that most of the tools used for input design for linear systems based on frequency domain techniques are no longer valid for the nonlinear case. One approach to input design for nonlinear systems is introduced in [99], where a linear systems perspective is considered. Based on a particular nonlinear system, [99] raises the issue of obtaining a parametrization for the input sequence resulting in tractable problems. The main message in [99] is that it is possible to reuse some of the parameterizations of input sequences for linear models in the nonlinear case. Thus, it is possible to parameterize the input sequence in terms of the marginal pdfs describing a stationary stochastic process, but it is not straightforward how to parameterize the input sequence in terms of its autocovariance function. In addition, [99] shows that it is possible to use the sum-of-squares method to relax the input design problem for nonlinear models.

The initial work in [99] has been extended to nonlinear FIR systems in [115], where a characterization of probability density functions of the input is employed. The assumption in [115] is that the input sequence is a realization of a stationary process. Taking into account the structure of nonlinear FIR models, in [115] it is shown that the requirement of stationarity for the input can be combined with the dynamical model to obtain a convex optimization problem. Input design for the identification of structured nonlinear models is introduced in [194, 195], where the system is assumed to be an interconnection of linear systems and static nonlinearities. The objective in [194, 195] is to minimize the variance of the experiment, while achieving the desired accuracy in the parameter estimates. It is shown in [194, 195] that the optimization problem can be expressed in terms of the probability mass function characterizing the input sequence. Moreover, [194, 195] show that the resulting optimization problem is convex in the decision variables. Once the optimal probability mass function is obtained, [194, 195] generate an input realization using elements from graph theory.

Input design for the identification of nonlinear state space models is presented in [88], where a particle filter is used to approximate the cost function associated with the input design problem. The estimated cost function is then optimized over a particular class of input vectors using stochastic approximation. In [88] it is assumed that the input sequence is an autoregressive process, and the parameters of this process are optimized numerically.

The class of nonlinear model structures is also generalized in [69], where the input signal is optimized over an alphabet of finite cardinality. The multilevel excitation design is also considered in [53] for the identification of Wiener models, and employed in [54] for the identification of nonlinear FIR models. Nevertheless, the results in [53, 54, 69] restrict the discussion to specific dynamical models and non-stationary inputs.

The restriction of the problem setup to specific nonlinear models and inputs can

be employed to derive closed form expressions for a scalar function of the Fisher information matrix. In [125, 126], the restriction of the input to be a realization of a Gaussian mixture leads to a closed form expression for the determinant of the Fisher information matrix when a Wiener model is considered. The expression derived in [126] is then used to develop a method to optimize the Fisher information matrix using the D-criterion. Even though the results in [126] provide insight on the effects of the input in the Fisher information matrix, the restriction to Wiener models and D-criterion limits the extension of these results to a more general models and other optimality criteria.

The methods previously mentioned [88, 99, 115, 194, 195] are in general highly complex (usually leading to non-convex problems, e.g., [88]) and are restricted to particular model structures (e.g., [99, 115, 194, 195]) or particular classes of input signals (e.g., AR processes [88]). Moreover, except for the results in [24, 25, 115], the methods introduced cannot handle input design with amplitude constraints. Amplitude constraints can arise due to safety reasons or physical limitations in the system. Therefore, input design with amplitude constraints even for linear systems also requires further study.

An additional limitation of the input design methods previously mentioned is the assumption of a prior estimate of the model parameters. The requirement of such knowledge is a common issue in input design. In the literature of input design, two approaches are mainly employed to overcome this difficulty. The first technique relies on adaptive design, where the input is designed iteratively as information is collected from the system [80, 151, 165]. The second method is based on robust methods, where the uncertainty on the model parameter is considered when designing the input signal [114, 164, 167, 204]. In regards of robust input design, the work in [167] designs the input sequence by optimizing the experiment for the worst case scenario defined by the model parameters, which are assumed to lie in a given compact set. The optimal input design in [167] also considers energy (or power) constraints for the input sequence. Moreover, [167] formulates a convex optimization problem that can be employed to solve a discretized approximation to the design problem.

1.4 Bayesian methods in system identification

As we have seen in Section 1.1, Bayesian techniques can be used to encode prior information about the model parameters when estimating a dynamical model. This principle can be extended to other problems as well, where a Bayesian approach can provide an alternative to not only encode prior information, but also to reduce the computational complexity of the resulting methods. In this section, we consider the use of Bayesian methods in two problems: (i) input design for nonlinear dynamical models, and (ii) parameter estimation of dynamical models. For Item (i), we discuss how Bayesian methods can provide an alternative for solving the input design problem. In Item (ii), we address the problem of encoding information about the system properties in the prior pdf, in particular about the order of the dynamical

system to be identified.

Bayesian methods for input design

In general, the problem of input design for nonlinear dynamical models is difficult to solve. The main issue for designing an input is the computation of the Fisher information matrix. Unless restrictions over the model are imposed to obtain the Fisher information matrix in closed form, the problem relies on estimates of this matrix. To illustrate this point, we consider the following example:

Example 1.8 *(Computation of the score function) Consider the expression for the Fisher information matrix $\mathcal{I}_F^T$ in Lemma 1.1. This expression depends on the score function $\mathcal{S}_T(u_{1:T}, \theta)$, which can be computed using the Fisher identity in Theorem B.1 to obtain*

$$\mathcal{S}_T(u_{1:T}, \theta_0) = \int_{\mathcal{X}_{1:T}} \nabla_\theta \log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})|_{\theta=\theta_0}\, p_{\theta_0}(x_{1:T}|y_{1:T}, u_{1:T}) dx_{1:T}\,. \tag{1.45}$$

In expression (1.45)*, $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$ denotes the complete data log-likelihood, and $p_{\theta_0}(x_{1:T}|y_{1:T}, u_{1:T})$ the conditional pdf of $x_{1:T}$ given the measurements $y_{1:T}$ and the input $u_{1:T}$. Hence, the computation of the score function $\mathcal{S}_T(u_{1:T}, \theta)$ requires the knowledge of the complete data log-likelihood and the conditional pdf appearing in the right hand side of the equality in* (1.45)*.*

With the exception of linear Gaussian state-space models, the expressions for $p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$ and $p_{\theta_0}(x_{1:T}|y_{1:T}, u_{1:T})$ in Example 1.8 are not available in closed form. Hence, if the evaluation of the score function is required (as it is the case for input design), an estimate of these quantities is needed.

Even though an estimate of the score function may be available, the resulting estimate is a noisy measurement. In consequence, we require an optimization method that accounts for this uncertainty as it searches for the optimal input.

Another difficulty associated with the input design problem is the optimization of the cost function over the feasible set. This requires the evaluation of the cost function over a potentially large number of elements in the set. Therefore, the optimization technique should guarantee that the samples over the feasible set are made in such a way that the method only explores those regions where an improvement on the cost function is expected, or where there are large uncertainties in the value of the cost function.

To address the previous points, this thesis explores the use of a Bayesian optimization framework to design inputs for the identification of nonlinear dynamical models. By specifying that the scalar function of the Fisher information matrix is a realization from a Gaussian process (GP), we can compute its predictive posterior distribution given a set of samples over the feasible set. The predictive posterior distribution acts as a surrogate of the intractable objective function, and is employed to compute the next sample over the feasible set by using an acquisition rule. This

technique recursively explores the feasible set to determine the element maximizing a surrogate function. The advantage of this approach when compared with existing techniques is that it can handle uncertainty in the estimates of the objective function, and it drives the exploration of the input space towards those regions where an improvement of the objective function is expected.

The Bayesian optimization method for input design is analyzed in details in Chapter 6, where we provide the implementation of an algorithm for designing inputs to estimate nonlinear dynamical models.

Bayesian identification of dynamical models

As discussed in Section 1.1, one alternative to maximum likelihood estimation is given by Bayesian identification. In the Bayesian framework, we compute the posterior pdf of the model parameters given the likelihood function and a prior pdf specified over the parameter set. In the case of the prior pdf, we have the freedom to impose the distribution to account for knowledge or belief prior to obtain $\mathbf{Z}_T$. Therefore, different posterior estimates can be derived from the same data set $\mathbf{Z}_T$ (for finite T), which will depend on the specified prior pdf.

A common rule when specifying a prior pdf over Θ is that it should reflect the knowledge on θ prior to performing the experiment. Even though the physical interpretation of the prior pdf over Θ is relevant, it is also equally desirable that the posterior pdf should be available in closed form. Closed form expressions for the posterior distribution of the model parameters allow to easily compute posterior estimates and to draw samples from it.

Closed form expressions for the posterior distribution $p(\theta|\mathbf{Z}_T)$ are usually enforced by restricting the prior π_Θ to belong to a specific family of distributions, referred to as conjugate priors:

Definition 1.3 *(Conjugate priors [159, Section 3.3]) A family $\mathcal{T}$ of probability distributions on Θ is said to be* conjugate *for a likelihood function $p_\theta(y_{1:T}|u_{1:T})$ if, for every $\pi_\Theta \in \mathcal{T}$, the posterior distribution $p(\theta|\mathbf{Z}_T)$ also belongs to $\mathcal{T}$.*

The convenience of imposing a prior pdf π_Θ that is conjugate to the likelihood function $p_\theta(y_{1:T}|u_{1:T})$ follows from that finding the posterior only requires an update of the distribution of the parameters [170]. Table 1.2 presents a list of some likelihood functions together with their conjugate priors [170].

As we can see from Table 1.2, if closed form expressions for the posterior distribution are of interest, then the prior must be chosen as function of the structure of the likelihood. Hence, we are limited to only imposing a particular class of prior distributions, which are not necessarily aligned with the prior knowledge (or belief) we have about the model parameters before performing the experiment.

Given the computational resources available nowadays, it is possible to define prior distributions accounting more properly for the prior knowledge about the model parameters than what the conjugate priors can. In particular, the prior pdf can be used to account for prior knowledge about the model complexity. How

Table 1.2: Examples of conjugate prior distributions for some likelihood functions (Source: [170]).

Likelihood function	Model parameters	Conjugate prior
Multivariate normal (with known covariance)	Mean	Multivariate Normal
Multivariate normal (with known mean)	Covariance	Inverse-Wishart
Multivariate normal	Mean and covariance	Normal-inverse-Wishart
Exponential	Rate	Gamma
Gamma (known shape)	Inverse scale	Gamma

to approximate the posterior distribution and encode prior information about the model complexity in the prior pdf is a question to be addressed in this thesis.

Literature review on Bayesian methods

The review of the state of the art in this section is divided into two parts: (i) Bayesian optimization and (ii) Bayesian system identification.

Bayesian optimization

Bayesian optimization techniques have gained interest in the Machine learning community [174]. Applications of this method can be found in robotics [121, 129], environmental monitoring [127], information extraction [202], combinatorial optimization [101, 203], sensor networks [76] and reinforcement learning [26], among others. In such cases, Bayesian optimization provides an alternative solution to the problem, where a surrogate model of the cost function is computed and optimized instead of the true cost function.

An advantage of using Bayesian optimization over some of the standard optimization techniques is that it accounts for the uncertainty associated with the estimation of the cost function. In this sense, Bayesian optimization is similar to stochastic optimization methods [181], where the optimization is solved by considering noisy measurements of the cost function.

The key ingredient to implement a Bayesian optimization method is the statistical model characterizing the cost function. In this regard, one choice is the Gaussian process (GP) as prior distribution for the cost function. Gaussian processes extend the notion of Gaussian distributions over random vectors to functions [155], and it

has been shown to be a useful tool in fields such as sensor networks [140], power systems [40], as well as in system identification [157]. Imposing a GP prior over the cost function is attractive as it can encode properties known a priori about the function, such as smoothness.

An application of Bayesian optimization to system identification has been made in [48], where Bayesian optimization is used to estimate nonlinear state-space models. The work in [48] imposes a Gaussian process prior over the likelihood function and computes its predictive posterior distribution based on estimations of the likelihood function over the feasible set.

Bayesian system identification

The use of Bayesian techniques for identification of dynamical models has gained attention in the recent years. The existing Bayesian methods for system identification are mainly implemented using the Gibbs sampler [34, 73, 78, 119, 208]. The Gibbs sampler is a particular implementation of the Metropolis Hastings algorithm [160], and hence it computes samples that are approximately distributed according to the posterior pdf. However, the implementation of the Gibbs sampler requires that the conditional marginal distributions of the posterior pdf are known in closed form, and that they can be used to generate samples. This property is commonly ensured by restricting the prior to the set of conjugate distributions of the likelihood (cf. Definition 1.3). Prior distributions outside this set are often intractable for the Gibbs sampler, and other methods are needed. For example, this is the case if we specify a prior distribution for the order of a dynamical system [160, Chapter 11].

In the case of model order selection, a Bayesian framework has also been developed [7, 32, 44]. The main method employed is the reversible jump Markov chain Monte Carlo (RJMCMC) sampler [91, 93]. The implementation of this algorithm requires the definition of a differentiable bijective transformation between model sets of different order, which is achieved by introducing auxiliary variables. While this approach is relatively easy to implement for small models, the complexity increases rapidly with the model order.

We remark that the Bayesian approach to system identification is not a recent development. In this line, [144] argued already in 1981 that Bayesian techniques can provide an attractive solution to the system identification problem. More recently, [137] presented a Markov chain Monte Carlo method to implement the Bayesian framework in system identification, where the Metropolis Hastings sampler is employed to compute posterior distributions. On the other hand, the role of sparse estimation in Bayesian system identification has gained interest in recent years. Indeed, in [150] the role of rank penalties in Bayesian system identification is discussed. The model order of the estimated system is implicitly taken into account, by including a regularization term associated with a weighted quadratic function of the Hankel matrix of the estimated impulse response coefficients. Even though the model complexity is considered in the optimization problem proposed in [150], adding a quadratic function of the Hankel matrix as a regularization term does not

guarantee that the optimal solution is of low order; this is because the regularization term is a generically differentiable function of the Hankel singular values, hence it cannot enforce sparseness on them.

1.5 On the role of the uncertainty in system identification

Some of the problems previously discussed assume that the information required for solving them is fully available. One example is the formulation of the input design problem based on a nominal model parameter. However, it is common the case where the information to solve the problem is not completely available. This lack of information is the main difficulty encountered in input design. Indeed, as can be noted from Section 1.3, the input design problem suffers from a crucial difficulty: the cost function may depend on the model parameter θ_0 describing the true system, as the Fisher information matrix $\mathcal{I}_F^T$ can depend on θ_0 (except, e.g., when the model is linearly parameterized). Therefore, the optimized input depends on the parameter to be estimated (cf. the literature review on input design on page 14). To illustrate this point, we consider the following example:

Example 1.9 *(Fisher information matrix for an OE model) Consider the OE model*

$$y_t = G_\theta(q)u_t + e_t\,, \tag{1.46}$$

where $G_\theta(q)$ is a real rational function in q parameterized by $\theta \in \Theta$, and $\{e_t\}$ is a Gaussian white noise of zero mean and variance σ^2. Under the assumption that there exists a $\theta_0 \in \Theta$ such that (1.46) *describes the true input-output relation when $\theta = \theta_0$, the Fisher information matrix becomes*

$$\mathcal{I}_F^T(u_{1:T},\,\theta_0) = \frac{1}{\sigma^2}\sum_{t=1}^{T} \nabla_\theta \hat{y}_{t|t-1}(\theta)\big|_{\theta=\theta_0}\, \nabla_\theta^\top \hat{y}_{t|t-1}(\theta)\big|_{\theta=\theta_0}\,, \tag{1.47}$$

with

$$\nabla_\theta \hat{y}_{t|t-1}(\theta) = \left[\nabla_\theta G_\theta(q)\right] u_t\,. \tag{1.48}$$

Based on Equations (1.47)-(1.48)*, we conclude that, in general, the Fisher information matrix for the OE model* (1.47) *depends on the value of the parameter describing the true input-output relation, as $\nabla_\theta \hat{y}_{t|t-1}(\theta)$ is often a function of θ.*

The problem of accounting for the uncertainty in input design has been addressed in the literature in different manners. On page 22 we provide a discussion on the existing approaches to cope with this uncertainty (cf. the literature review on input design on page 14). In this thesis, we motivate the discussion by means of robust input design [167].

In robust input design, the goal is to design an excitation maximizing a scalar cost of the Fisher information matrix, which also takes the uncertainty on the model parameters into account. The standard formulation for robust input design is

$$u_{1:T}^{\text{opt}} := \arg \min_{u_{1:T}\in\mathbb{U}^T} \mathcal{R}\left\{h(\mathcal{I}_F^T(u_{1:T},\theta),\,\theta)\right\}\,, \tag{1.49}$$

where $\mathcal{R}\colon \mathcal{L}^2 \to \mathbb{R}$ is a predefined functional accounting for the uncertainty on θ_0 (characterized by a probability measure $\mathbf{P}_\Theta$), and $\mathbb{U} \subseteq \mathbb{R}$ is the feasible set of u_t. The set $\mathcal{L}^2$ contains those random variables on Θ whose mean and variance (with respect to the probability measure defined on Θ) are finite.

Remark 1.7 *In the literature of control theory and input design, the term robust usually refers to the worst case design [167, 215]. In this thesis, the term robust will refer to any functional $\mathcal{R}\colon \mathcal{L}^2 \to \mathbb{R}$ quantifying the cost incurred by the lack of knowledge on the value of a random variable in $\mathcal{L}^2$.*

We note that the function $\mathcal{R}$ covers a wide range of possibilities to account for the uncertainty on θ_0. Examples for $\mathcal{R}$ are $\mathcal{R} = \mathbf{E}_\Theta\{\cdot\}$ (where it is assumed that Θ is a probability space with known cumulative distribution function P_Θ), and $\mathcal{R} = \max_{\theta\in\Theta}\{\cdot\}$.

Due to the generality of $\mathcal{R}$, we have a new issue for accounting for the uncertainty. Now, the difficulty is in how to properly define $\mathcal{R}$ such that the optimized input is less sensitive to the value of θ_0 describing the system. Even though the expected value and the worst case design are standard tools in input design, an approach to properly define $\mathcal{R}$ is missing.

A new perspective on the problem of properly defining $\mathcal{R}$ can be gained if we consider a risk theoretical framework. An approach to properly choose $\mathcal{R}$ is one of the contributions of this thesis. In the next subsection we provide a review of the literature of risk theory.

Literature review on uncertainty and risk theory

Uncertainty is a frequent issue in many fields. By uncertainty we mean the lack of knowledge about a phenomenon to fully understand its behavior. Depending on the subject, the uncertainty can be interpreted in various forms. For example, in robust control the uncertainty arises as the lack of knowledge about the plant dynamics [215], while in portfolio optimization the uncertainty describes the lack of knowledge regarding the future returns of an asset [74, 143].

In system identification, the uncertainty corresponds to the lack of knowledge about the location of the parameter defining the true system. Under suitable assumptions on the model and the parameter set, it is possible to approximate such uncertainty by the use of the central limit theorem [122]. This information can be then used to provide confidence bounds on the location of the true parameter in the parameter set, which has been the main approach used in system identification to account for this uncertainty [18, 96].

A topic in system identification where the uncertainty plays a crucial role is input design. As mentioned in Section 1.3, the goal in input design is to obtain an excitation maximizing the amount of information about the system, quantified as a scalar cost function of the intended model application. The main issue in input design is that it usually relies on prior information about the model parameters. Solutions to this issue have been previously discussed in the literature review on

input design on page 14, where adaptive and robust schemes are proposed. This thesis focuses on robust techniques.

In regards of robust approaches to input design, we can quantify the uncertainty in different manners. For example, we can just consider a nominal parameter to design the experiment, which is referred to as nominal input design [186], use chance constraints to guarantee a prescribed accuracy on the estimated parameter [166], or a worst case scenario, where the input is designed by considering the parameter in the parameter set delivering the worst performance [164]. Even though many approaches have been considered to account for the uncertainty on the model parameter, there is no clear approach to systematically choose the robust measure to include the uncertainty in the problem formulation.

On the other hand, the issue of systematically accounting for the uncertainty in optimization problems has been well studied in the risk theory literature [9, 17]. The focus on risk theory is on minimizing the losses due to uncertainty in the returns of a portfolio of assets. To this end, the notion of coherent measures of risk has been introduced [10, 161], which provides a framework to consider the uncertainty in the returns of the assets. A portfolio optimization based on coherent measures of risk leads to desired property of diversification: it is always better to invest on several assets rather in a single one, since it leads to a reduction in the risk of losing with a portfolio.

1.6 Thesis contributions

This thesis makes contributions to three topics: (i) input design for identification of nonlinear models, (ii) Bayesian techniques in system identification, and (iii) Risk coherent formulations in system identification.

Input design

In an effort to extend the methods in input design to nonlinear dynamical models, this thesis introduces a robust input design method for the identification of nonlinear state-space models. The input is constrained to be a realization of a stationary Markov process with finite alphabet, whose stationary probability mass function (pmf) is defined over vectors of consecutive inputs. Therefore, the problem is to find the stationary pmf maximizing the information obtained from the experiment, quantified as a scalar function of the Fisher information matrix. As the true system is unknown, the method takes the uncertainty on the true value of the parameters into account by formulating a robust problem.

By using notions of graph theory, we can express the feasible set of pmfs as a convex combination of the pmfs describing the vertices of the set. Since the vertices of the set can be explicitly computed by known graph theoretical algorithms [103, 210], the optimization problem becomes easy to pose, as it needs the computation of the Fisher information matrix only for the pmfs defining the vertices of the feasible set. The required Fisher information matrices are then computed by particle methods.

For standard choices of the cost function, the proposed formulation for input design is convex even for nonlinear systems, which reduces the computational complexity compared with the Markov chain approach in [24, 25]. Finally, since the input is restricted to a finite alphabet, the method naturally incorporates amplitude constraints.

The proposed input design method has proven to be an attractive solution to practical problems arising in the identification and estimation literature. As an illustration of the relevance of the proposed method, this thesis presents the use of the robust input design technique for model estimation with quantized output data. In this problem the quantized output data introduces nonlinear behavior in the system, which restricts the techniques that can be employed to design the input sequence.

Bayesian methods

This thesis also provides a framework to implement Bayesian techniques for both experiment design and parameter estimation for dynamical models. The first contribution of this part is on addressing the computational complexity when calculating the objective function used in input design. To this end, a Gaussian process optimization (GPO) method to input design for identification of nonlinear dynamical models is introduced [174]. By imposing that the scalar function of the Fisher information matrix is a realization from a Gaussian process (GP) [155], the technique computes the predictive posterior distribution of the objective function given a set of samples over the feasible set. The predictive posterior distribution acts as a surrogate of the unavailable objective function, and is employed to compute the next sample over the feasible set by using an acquisition rule. This technique iteratively explores the feasible set to determine the element optimizing the surrogate function. The advantage of this approach when compared to existing techniques is that it can handle uncertainty in the estimates of the objective function, and it drives the exploration of the input space towards those regions where an improvement of the objective function is expected.

The second contribution of this part is the development of a Bayesian method for the identification of SISO linear time-invariant dynamical models. The method accounts for prior knowledge on the model complexity, which is encoded by imposing a prior on the Hankel singular values of the transfer function of the system. As the posterior distribution of the model parameters is often unavailable in closed form, a Metropolis-Hastings (MH) sampler is implemented [94, 160]. The MH sampler approximates the posterior distribution of the model parameters by samples that are approximately given by this distribution. Then, a posterior estimate of the model parameters is provided by, e.g, the maximum a posteriori (MAP) or the posterior mean. The existence of the posterior distribution of the model parameters, as well as the convergence of the output of the MH sampler to the posterior distribution is studied and proven.

Risk theory in system identification

The third part of this thesis focuses on how risk theoretical tools can be used to define functionals properly accounting for the uncertainty in system identification. By introducing the notion of coherent measures of risk [10], the thesis explores two applications of such measures in input design. The first application considers accounting for the uncertainty in the classical formulation of input design, where a scalar cost function of the Fisher information matrix is optimized. As the problem formulation depends on the unknown model parameter, we discuss the use of risk coherent measures to account for the uncertainty on the model parameter. In particular, we employ the conditional value at risk (CVaR) to include the uncertainty on the model parameter into the problem formulation [161].

The second use of risk coherent measures is in application oriented input design. In this problem, we are interested in minimizing the cost of the experiment to be applied to the system, while guaranteeing a prescribed accuracy on the estimated parameter based on the data provided by the experiment. The main difficulty here is that the accuracy on the model parameter depends on both the true system description and the estimated parameter, which are subject to uncertainty. To account for the uncertainty on both parameters, a risk coherent framework for application oriented input design is proposed. As the uncertainty on the estimated parameter depends on the experiment to be designed in the optimization, we also provide a stochastic approximation method to find a suboptimal excitation satisfying the quality constraint on the estimated parameter. The convergence of the proposed technique is analyzed and proven.

1.7 Outline of the thesis and publications

The thesis is structured in ten chapters (including the present one). The chapters composing the main body of this work are grouped into three parts:

Part I: Graph theory in input design. This part of the thesis covers the problem of input design for nonlinear dynamical models. The focus is on the use of graph theoretical tools to pose a convex optimization problem, where the solution defines the optimal stationary distribution of a Markov process. Four chapters are considered in this part:

Chapter 2. Graph theory and Markov processes: This chapter describes how graph theory can be used to parameterize the set of stationary Markov processes with finite alphabet. The contents of this chapter are based on

(J1) P.E. Valenzuela, C.R. Rojas, H. Hjalmarsson, and R. Hildebrand. On accuracy analysis of a graph theoretical approach to input design for identification of nonlinear dynamical models. In preparation.

(J2) P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. A graph theoretical approach to input design for identification of nonlinear dynamical models. Automatica, volume 51, January 2015, pp. 233–242.

Chapter 3. A review on sequential Monte Carlo methods: This chapter is focused on sequential Monte Carlo techniques, which are needed to implement the input design methods introduced in this thesis.

Chapter 4. Robust input design for nonlinear models: This chapter is concerned with the design of inputs for the identification of nonlinear state-space models. To account for the uncertainty in the model description, a robust measure is considered for the cost function. The content of this chapter is based on

(J3) P.E. Valenzuela, J. Dahlin, C.R. Rojas, and T.B. Schön. On robust input design for nonlinear dynamical models. Accepted for publication. Automatica, 2016.

(J2) P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. A graph theoretical approach to input design for identification of nonlinear dynamical models. Automatica, volume 51, January 2015, pp. 233–242.

(C1) P.E. Valenzuela, J. Dahlin, C.R. Rojas, and T.B. Schön. A graph/particle-based method for experiment design in nonlinear systems. In *Proceedings of the* 19^{th} *IFAC World Congress*, Cape Town, South Africa, 2014.

(C2) P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. Optimal input design for non-linear dynamic systems: a graph-theory approach. In *Proceedings of the* 52^{nd} *Conference on Decision and Control (CDC)*, Florence, Italy, 2013.

Chapter 5. Robust input design for systems with quantized inputs and measurements: As an application of the graph theoretical approach to input design, we consider the problem of input design when the output of the system is subject to quantized measurements. This chapter is based on

(J4) P.E. Valenzuela, B.I. Godoy, and C.R. Rojas. On robust input design for identification of FIR systems with quantized measurements. Submitted to the IEEE Signal Processing Letters, 2016.

(C3) B.I. Godoy, P.E. Valenzuela, C.R. Rojas, J.C. Agüero, and B. Ninness. A novel input design approach for systems with quantized output data. In *Proceedings of the* 13^{th} *European Control Conference (ECC)*, Strasbourg, France, 2014.

Part II: Bayesian techniques in system identification. This part of the thesis explores how existing techniques in Bayesian inference can be used to solved the classical problems in system identification. The discussion in this part is divided into two chapters:

Chapter 6. A Gaussian process optimization method to input design for the identification of nonlinear models: Motivated by the difficulty of solving the input design problem for the identification of nonlinear

models, this chapter explores the use of a Gaussian process optimization framework in input design. By imposing a prior on the objective function, the technique iteratively explores the feasible set to find the optimal input. The discussion in this chapter is based on

(C4) P.E. Valenzuela, J. Dahlin, C.R. Rojas, and T.B. Schön. Particle-based Gaussian process optimization for input design in nonlinear dynamical models. Accepted for publication in *the* 55^{th} *Conference on Decision and Control (CDC)*, Las Vegas, United States, 2016.

Chapter 7. On model order priors for Bayesian identification of linear models: This chapter discusses how to estimate the model complexity in a Bayesian setting for the identification of SISO LTI models. By imposing priors over the Hankel singular values of the rational transfer function, the posterior distribution of the model parameters is computed using a Metropolis-Hastings sampler. The contents of this chapter are based on

(J5) P.E. Valenzuela, T.B. Schön, and C.R. Rojas. On model order priors for Bayesian identification of SISO linear systems. Provisionally accepted. International Journal of Control, 2016.

Part III: Risk coherent framework for system identification. As highlighted in this chapter, model uncertainties play a crucial role in many problems in system identification. This chapter focuses on how to properly measure the uncertainty and include it into the problem formulation. To this end, a risk coherent formulation is presented. This part is divided into two chapters:

Chapter 8. On risk theory in system identification: As a first approach to the problem of uncertainty, this chapter discusses how the notion of risk coherent measures can be useful to account for the uncertainty in system identification. The focus in this chapter is on input design, but the same principles can be extended to other areas in the field. The contents of this chapter are based on

(C5) P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. Uncertainty in system identification: learning from the theory of risk. In *Proceedings of the* 17^{th} *IFAC Symposium on System Identification (SYSID)*, Beijing, China, 2015.

Chapter 9. Risk coherent application oriented input design: To account for the uncertainty in application oriented input design, a risk coherent framework is discussed in this chapter. The chapter focuses on how to include risk coherent measures into the problem formulation and the difficulties in solving the resulting optimization problem. To address the difficulty of solving the optimization, a stochastic approximation method is proposed and its convergence is analyzed. The contents of the present chapter are based on

(J6) P.E. Valenzuela, A. Ebadat, and M. Annergren. On risk coherent application oriented input design. In preparation.

Finally, the last chapter of this thesis presents concluding remarks and future research paths.

1.8 Other publications

Publications not covered in the present thesis where the author has contributed are:

(J7) R.A. González, P.E. Valenzuela, C.R. Rojas, and R.A. Rojas. Optimal enforcement of causality in non-parametric transfer function estimation. Submitted to Signal Processing, 2016.

(C6) M.I. Müller, P.E. Valenzuela, and C.R. Rojas. Risk-coherent $\mathcal{H}_2$ optimal disturbance rejection under model uncertainty. Submitted to the 20^{th} IFAC World Congress, Toulouse, France, 2017.

(J8) P.E. Valenzuela, A. Ebadat, N. Everitt, and A. Parisio. Closed-loop identification for model predictive control of HVAC systems: a guideline from input design to controller design. Submitted to the IEEE Transactions on Automation Science and Engineering, 2016.

(J9) J. Bjurgert, P.E. Valenzuela, and C.R. Rojas. On adaptive boosting for system identification. Submitted to the IEEE Transactions on Neural Networks and Learning Systems, 2016.

(J10) A. Ebadat, P.E. Valenzuela, C.R. Rojas, and B. Wahlberg. Model predictive control oriented experiment design for system identification: A graph theoretical approach. Submitted to the Journal of Process Control, 2015.

(C7) C.R. Rojas, P.E. Valenzuela, and R.A. Rojas. A critical view on benchmarks based on randomly generated systems. In *Proceedings of the* 17^{th} *IFAC Symposium on System Identification (SYSID)*, Beijing, China, 2015.

(C8) A. Ebadat, P.E. Valenzuela, C.R. Rojas, H. Hjalmarsson, and B. Wahlberg. Applications oriented input design for closed-loop system identification: a graph-theory approach. In *Proceedings of the* 53^{rd} *Conference on Decision and Control (CDC)*, Los Angeles, United States, 2014.

Part I

Graph theory in input design

Chapter 2

Graph theory and Markov processes

The first part of this thesis relies on a specific link between graph theory and stationary processes. The link is established between a specific set of cycles in a de Bruijn graph and a set of marginal distributions of stationary processes. Specifically, by making use of the results in [210], we can link a set of marginal probability mass functions (pmf) of stationary processes with finite alphabet with a set of cycles in the equivalent de Bruijn graph (cf. Definition 2.15 and Theorem 2.1).

The relationship between de Bruijn graphs and marginal pmfs of stationary processes with finite alphabet is then used to parameterize stationary Markov processes in terms of their stationary pmfs. The resulting parametrization of the stationary pmf is given as the convex combination of the measures associated with a specific set of cycles of the equivalent de Bruijn graph (cf. Definition 2.15). This property provides a convex formulation for the input design problem discussed in Chapters 4 and 5.

Given an element in the set of stationary distributions of the Markov process, we are interested in obtaining a realization of the associated stationary Markov process. How to obtain such realization from the stationary Markov process is also discussed in this chapter. The contents of this chapter are partly based on [186].

2.1 Graph theory: basic concepts

In this section we provide a number of definitions of graph theoretical concepts employed in Chapters 4 and 5. This section is partly based on [103].

Definition 2.1 *(Directed graph) A* directed graph $\mathcal{G}_{\mathcal{V}} = (\mathcal{V}, \mathcal{E})$ *is a pair consisting of a finite nonempty set of points (called* nodes *or* vertices*)* $\mathcal{V}$ *and a set* $\mathcal{E}$ *of ordered pairs* (z_i, z_j) *of vertices* $z_i, z_j \in \mathcal{V}$ *called* edges.

Definition 2.2 *(Path) A* path *in a directed graph $\mathcal{G}_{\mathcal{V}}$ from node z to node u is a tuple of vertices*

$$\boldsymbol{\omega}_{zu} = (z_1, z_2, \ldots, z_k),$$

such that $(z_i, z_{i+1}) \in \mathcal{E}$ for all $i \in \{1, \ldots, k-1\}$, $z_1 = z$ and $z_k = u$.

Definition 2.3 *(Cycle and elementary cycle) A* cycle *is a path $\boldsymbol{\omega}_{zu}$ in which z and u coincide. A cycle is* elementary *if no vertex except the first and last appears twice.*

Definition 2.4 *(Cyclic permutation) Consider two cycles $\boldsymbol{\omega}_{uu}$ and $\boldsymbol{\omega}_{zz}$ in a directed graph $\mathcal{G}_{\mathcal{V}}$. We say that $\boldsymbol{\omega}_{zz}$ is a* cyclic permutation *of $\boldsymbol{\omega}_{uu}$ if and only if there exist paths $\boldsymbol{\omega}_{uz}$, and $\boldsymbol{\omega}_{zu}$ such that $\boldsymbol{\omega}_{zz} = (\boldsymbol{\omega}_{zu}, \boldsymbol{\omega}_{uz})$, where the first element in $\boldsymbol{\omega}_{uz}$ is removed, and $\boldsymbol{\omega}_{uu} = (\boldsymbol{\omega}_{uz}, \boldsymbol{\omega}_{zu})$, where the first element in $\boldsymbol{\omega}_{zu}$ is removed.*

Definition 2.5 *(Distinct elementary cycles) Two elementary cycles are* distinct *if and only if one is not a cyclic permutation of the other.*

De Bruijn graphs

The results in this thesis are based on a specific class of directed graphs, called *de Bruijn* graphs [52]. Their definition is given below.

Definition 2.6 *(de Bruijn graph) The n-dimensional* de Bruijn graph *of m symbols in $\mathcal{C} = \{s_1, \ldots, s_m\}$ denoted as $\mathcal{G}_{\mathcal{C}^n}$ is a directed graph whose set of vertices $\mathcal{V}$ is given by*

$$\begin{aligned}\mathcal{V} = \mathcal{C}^n := \{&(s_1, \ldots, s_1, s_1), (s_1, \ldots, s_1, s_2), \ldots,\\ &(s_1, \ldots, s_1, s_m), (s_1, \ldots, s_2, s_1), \ldots,\\ &(s_m, \ldots, s_m, s_1), \ldots, (s_m, \ldots, s_m, s_m)\},\end{aligned} \tag{2.1}$$

and whose set of directed edges $\mathcal{E}$ is

$$\mathcal{E} = \{((r_1, \ldots, r_n), (r_2, \ldots, r_n, v)) : v, r_1, \ldots, r_n \in \mathcal{C}\}. \tag{2.2}$$

Remark 2.1 *According to Definition 2.6, an n-dimensional de Bruijn graph of m symbols is a directed graph representing overlaps between pairs of n-dimensional vectors. It has m^n vertices, consisting of all possible n-dimensional vectors built from a set with m symbols. We note that the same symbol can appear multiple times in the vector. Moreover, if one of the vertices can be expressed as another vertex by shifting all its elements one place to the left and adding a new symbol at the end, then the latter has a directed edge to the former vertex.*

To illustrate Definition 2.6 we consider the following example:

Table 2.1: Transitions from $(u_{t-1},\, u_t)$ to $(u_t,\, u_{t+1})$, Example 2.1.

$(u_{t-1},\, u_t)$	$(u_t,\, u_{t+1})$
(0, 0)	{(0, 0), (0, 1)}
(0, 1)	{(1, 0), (1, 1)}
(1, 0)	{(0, 0), (0, 1)}
(1, 1)	{(1, 0), (1, 1)}

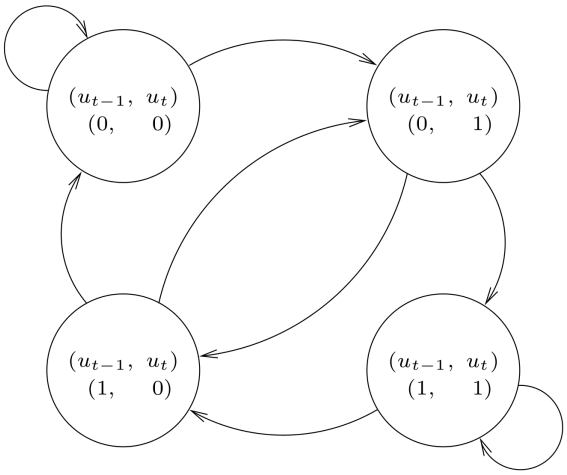

Figure 2.1: A 2-dimensional de Bruijn graph with symbols in $\mathcal{C} = \{0, 1\}$.

Example 2.1 *Consider a sequence $\{u_t\}$ over the alphabet $\mathcal{C} = \{0, 1\}$. We are interested in deriving the de Bruijn graph associated with the possible transitions of $(u_{t-1},\, u_t)$ among its states in $\mathcal{C}^2$. In other words, we want to derive the possible transitions from $(u_{k-1},\, u_k)$ to $(u_k,\, u_{k+1})$. Table 2.1 contains all the possible transitions between the elements in $\mathcal{C}^2$. We note that the number of possible values of $(u_k,\, u_{k+1})$ given $(u_{k-1},\, u_k)$ is two, which is the number of elements in $\mathcal{C}$. This is due to the fact that the new element u_{k+1} belongs to $\mathcal{C}$.*

The resulting de Bruijn graph is depicted in Figure 2.1. We observe that the nodes are given by the elements of $\mathcal{C}^2$, and the edges correspond to the transitions presented in Table 2.1.

2.2 De Bruijn graphs and stationary processes

In this section, we establish the connection between the n-dimensional de Bruijn graph with symbols in $\mathcal{C}$ and the n-dimensional marginal distributions associated

with stationary processes over the alphabet $\mathcal{C}$. Before proceeding, we introduce the following definitions:

Definition 2.7 *(Cumulative distribution function) Let n_m be a positive integer, and X be an n_m-dimensional random vector with associated probability measure $\mathbf{P}$. The function $P\colon \mathcal{C}^{n_m} \to \mathbb{R}$, defined as*

$$P(x_1, \ldots, x_{n_m}) := \mathbf{P}\{X_1 \leq x_1, \ldots, X_{n_m} \leq x_{n_m}\}, \tag{2.3}$$

is the cumulative distribution function *(cdf) of the random vector X.*

Definition 2.8 *(Stationary process) A stochastic process $\{u_t\}$ is* stationary *if and only if, for every tuple of integers (i, j, k) such that $k \geq 1$, the k-dimensional marginal cumulative distribution functions $\{P_t^{(k)}\}$ ($P_t^{(k)}\colon \mathcal{C}^k \to \mathbb{R}$) associated with k consecutive time instants of $\{u_t\}$ satisfy $P_i^{(k)}(u_{i-k+1:i}) = P_j^{(k)}(u_{j-k+1:j})$.*

As we can see from Definition 2.8, a stationary process requires that all its consecutive marginal cumulative distribution functions (cdfs) are time invariant.

Before concluding this part, we introduce the following definition:

Definition 2.9 *(Probability mass function) A* probability mass function *(pmf) p is a probability measure whose support is a countable set.*

A relevant question for Chapters 4 and 5 is how to characterize a stochastic process in a tractable manner, as the stochastic process is the basis for the input design method discussed there. This point will be considered in the following section.

Characterizing a stochastic process with finite alphabet

We would like to characterize a stationary stochastic process u_t having a finite alphabet, i.e. $u_t \in \mathcal{C}$. Such characterization requires the definition of all its k-dimensional marginal cdfs $\{P^{(k)}\}_{k\geq 1}$ ($P^{(k)}\colon \mathcal{C}^k \to \mathbb{R}$). However, such description of the stationary process is intractable since we need to define an infinite number of cumulative distribution functions (cf. Definition 2.8).

Instead, we will further restrict the process to be described by a stationary Markov process. To this end, we will need the following definitions:

Definition 2.10 *(Markov process, [58, p. 80]) A* Markov process *is a stochastic process $\{x_t\}$ ($x_t \in \mathbb{R}^{n_x}$) satisfying the following condition: for any positive integer n, if $\{t_i\}_{i=1}^n$ are parameter values such that $t_i < t_j$ for all $i < j$, $i, j \in \{1, \ldots, n\}$, the probabilities of x_{t_n} conditioned on $\{x_{t_i}\}_{i=1}^{n-1}$ are the same as those conditioned on $x_{t_{n-1}}$ in the sense that, for each $\lambda \in \mathbb{R}^{n_x}$,*

$$\mathbf{P}\left\{x_{t_n} \leq \lambda \,\middle|\, x_{t_1}, \ldots, x_{t_{n-1}}\right\} = \mathbf{P}\left\{x_{t_n} \leq \lambda \,\middle|\, x_{t_{n-1}}\right\}, \tag{2.4}$$

with probability 1.

In the following, we consider a stationary Markov process defined for all $t \geq 0$ as

$$x_{t+1}|x_t \sim p(x_{t+1}|x_t)\,, \tag{2.5}$$

where p is a conditional probability mass function, $x_t \in \mathcal{X}$ for all $t \geq 0$, and $\mathcal{X}$ is a finite set. Based on (2.5), we recursively define the transition probability mass functions of the stationary Markov process $\{x_t\}_{t\geq 0}$ as follows:

$$p^{(1)}\left(x_1\,|x_0\,\right) = p\left(x_1\,|x_0\,\right)\,, \tag{2.6a}$$

$$p^{(n)}\left(x_n\,|x_0\,\right) = \sum_{x_{n-1}\in\mathcal{X}} p\left(x_n\,|x_{n-1}\,\right)\, p^{(n-1)}\left(x_{n-1}\,|x_0\,\right)\,,\ n>1\,. \tag{2.6b}$$

Before continuing, we need the following statements [58]:

Definition 2.11 *(Irreducible Markov process) A Markov process $\{x_t\}_{t\geq 0}$ in $\mathcal{X}$ is* irreducible *if and only if for all $x_0, x \in \mathcal{X}$, there exists some $n > 0$, possibly depending on both x and x_0, such that $p^{(n)}(x|x_0) > 0$.*

Definition 2.12 *(Periodic Markov process) A Markov process $\{x_t\}_{t\geq 0}$ in $\mathcal{X}$ is* periodic *if and only if there exists a partition $\{\mathcal{X}_i\}_{i=1}^{d}$ of $\mathcal{X}$ for some $d \geq 2$ (called a* cyclic partition *of $\mathcal{X}$) such that $\sum_{z\in\mathcal{X}_{i+1}} p^{(1)}(z|x) = 1$ for all $x \in \mathcal{X}_i$ ($1 \leq i \leq d-1$), and $\sum_{z\in\mathcal{X}_1} p^{(1)}(z|x) = 1$ for all $x \in \mathcal{X}_d$.*

Definition 2.13 *(Aperiodic Markov process) A Markov process $\{x_t\}_{t\geq 0}$ in $\mathcal{X}$ is* aperiodic *if and only if it is not periodic.*

Lemma 2.1 *[58, p. 182] Consider an irreducible, aperiodic, Markov process $\{x_t\}_{t\geq 0}$ in $\mathcal{X}$ (of finite cardinality). Then there exists a unique function $p^{\mathrm{st}}\colon \mathcal{X} \to \mathbb{R}$, called the* stationary probability mass function*, satisfying*

$$p^{\mathrm{st}}(x) \geq 0\,,\ \textit{for all } x \in \mathcal{X}\,, \tag{2.7a}$$

$$\sum_{x\in\mathcal{X}} p^{\mathrm{st}}(x) = 1\,, \tag{2.7b}$$

$$p^{\mathrm{st}}(z) = \sum_{x\in\mathcal{X}} p\left(z\,|x\,\right) p^{\mathrm{st}}(x)\,,\ \textit{for all } z \in \mathcal{X}\,. \tag{2.7c}$$

In addition, for all z, $x \in \mathcal{X}$,

$$\lim_{n\to\infty} \left| p^{(n)}(z|x) - p^{\mathrm{st}}(z) \right| = 0\,. \tag{2.8}$$

Based on the previous statements, we introduce a suitable assumption to characterize the stochastic process $\{u_t\}_{t\geq -n_m+1}$ for positive integers n_m:

Assumption 2.1 *The stationary process* $\{u_t\}_{t\geq -n_m+1}$, $u_t \in \mathcal{C} \subset \mathbb{R}$, *where* $\mathcal{C}$ *is a finite set of cardinality* $n_{\mathcal{C}}$, *and* n_m *is a positive integer, is associated with an irreducible and aperiodic Markov process given by* (2.5), *with* $x_t := u_{t-n_m+1:t} \in \mathcal{C}^{n_m}$, *where* $u_{t-n_m+1:t} = (u_{t-n_m+1}, \ldots, u_t)$. *We will say that* $\{u_t\}$ *is a* (stationary) Markov process of memory n_m. *In addition, the stationary probability mass function of the Markov process* $\{x_t\}_{t\geq 0}$ *is given by* p^{st} *(cf. Lemma 2.1), where* p^{st} *belongs to the set of* n_m*-dimensional marginal pmfs of stationary processes.*

Given the restriction of the process $\{u_t\}$ to satisfy Assumption 2.1, the problem is then to characterize the set of n_m-dimensional marginal pmfs of stationary processes. Before continuing, we require the following definition [210, p. 679]):

Definition 2.14 *(*n*-stationary cdf and pmf) The cdf* $P\colon \mathbb{R}^n \to \mathbb{R}$ *(equivalently, the pmf* $p\colon \mathbb{R}^n \to \mathbb{R}$*) is* n-stationary *if and only if for all positive integers* i, j, k *satisfying* $1 \leq i \leq j \leq n-k$, *the relation* $u_{i:i+k} \overset{d}{\sim} u_{j:j+k}$ *is satisfied, where* $\overset{d}{\sim}$ *denotes equal in distribution.*

Based on Definition 2.14, we introduce the following result in terms of cdfs (whose proof comes from [210, Theorem 2]):

Lemma 2.2 *(Shift invariant property) A cdf* $P\colon \mathbb{R}^{n_m} \to \mathbb{R}$ *is* n_m*-stationary if and only if, for all* $\mathbf{z} \in \mathbb{R}^{n_m-1}$,

$$\int_{v\in\mathbb{R}} dP\left([v, \mathbf{z}]\right) = \int_{v\in\mathbb{R}} dP\left([\mathbf{z}, v]\right) . \tag{2.9}$$

Proof *First we assume that* P *is* n_m*-stationary. This implies that* $u_{1:n_m-1} \overset{d}{\sim} u_{2:n_m}$, *which is equivalent to the condition written in* (2.9). *To prove the converse, we assume that* (2.9) *is true. Therefore,* $u_{1:n_m-1} \overset{d}{\sim} u_{2:n_m}$ *is satisfied. Then for any* $k < n_m$, *and by successive shifts, the marginal cdf* $P(u_{1:k})$ *obtained from* $P(u_{1:n_m})$ *satisfies*

$$u_{1:k} \overset{d}{\sim} u_{2:k+1} \overset{d}{\sim} u_{3:k+2} \ldots \overset{d}{\sim} u_{n_m-k+1:n_m} , \tag{2.10}$$

which implies that $u_{1:n_m}$ *has an* n_m*-stationary cdf* P. ■

Lemma 2.2 states that an n_m-dimensional cdf P is n_m-stationary if and only if the cdf obtained by marginalizing over u_{n_m} is equivalent to the one obtained by marginalizing over u_1. If a cdf P satisfies this property, then we say that P is *shift invariant*.

The result introduced in Lemma 2.2 allows us to characterize the set of n_m-stationary cdfs as

$$\mathcal{P} := \Big\{ P\colon \mathbb{R}^{n_m} \to \mathbb{R} \Big| P(\mathbf{x}) \geq 0,\ \forall \mathbf{x} \in \mathbb{R}^{n_m};$$
$$P \text{ is monotone nondecreasing ;}$$

$$\lim_{\substack{x_i \to \infty \\ i \in \{1, \ldots, n_m\}}} P(x_1, \ldots, x_{n_m}) = 1;$$

$$\int_{v \in \mathbb{R}} dP\left([v, \mathbf{z}]\right) = \int_{v \in \mathbb{R}} dP\left([\mathbf{z}, v]\right), \forall \mathbf{z} \in \mathbb{R}^{n_m - 1} \Bigg\}. \quad (2.11)$$

We notice in (2.11) that the first three properties define a valid cdf in $\mathbb{R}^{n_m}$. Introducing shift invariance as the fourth property in the set $\mathcal{P}$, we restrict the cdfs to be n_m-stationary [210].

To establish a relation between de Bruijn graphs and n_m-stationary cdfs, we use the finite cardinality of $\mathcal{C}$ required in Assumption 2.1. In the following, we employ the set of n_m-stationary pmfs defined as

$$\mathcal{P}_{\mathcal{C}} := \Bigg\{ p \colon \mathcal{C}^{n_m} \to \mathbb{R} \Big| p(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathcal{C}^{n_m};$$

$$\sum_{\mathbf{x} \in \mathcal{C}^{n_m}} p(\mathbf{x}) = 1;$$

$$\sum_{v \in \mathcal{C}} p\left([v, \mathbf{z}]\right) = \sum_{v \in \mathcal{C}} p\left([\mathbf{z}, v]\right), \forall \mathbf{z} \in \mathcal{C}^{n_m - 1} \Bigg\}. \quad (2.12)$$

To make use of the set $\mathcal{P}_{\mathcal{C}}$, we need to parameterize the elements of $\mathcal{P}_{\mathcal{C}}$ in a tractable manner. This issue is relevant here, as in the next chapters we will optimize cost functions defined in terms of $u_{1:T}$, which is a realization of the Markov process satisfying Assumption 2.1. A natural approach is to parameterize $p(u_{1:n_m}) \in \mathcal{P}_{\mathcal{C}}$ (the stationary distribution of the Markov process) in terms of a finite number of elements of $\mathcal{P}_{\mathcal{C}}$. Such a parameterization is described in the next subsection.

Remark 2.2 *Lemma 2.2 does not necessarily imply that every $P \in \mathcal{P}$ is the n_m-dimensional marginal cdf of a stationary process $\{u_t\}$. However, the existence of a stationary process $\{u_t\}$ will be proven for the case of the set $\mathcal{P}_{\mathcal{C}}$, which is stated in Theorem 2.2.*

Parameterization of $\mathcal{P}_{\mathcal{C}}$

To parameterize the elements in $\mathcal{P}_{\mathcal{C}}$, we first notice that $\mathcal{P}_{\mathcal{C}}$ is a convex set, and, in particular, a polyhedron [163, pp. 170], since it is described by a finite number of linear equalities and inequalities. Hence, any element of $\mathcal{P}_{\mathcal{C}}$ can be described as a convex combination of its extreme points, which are themselves pmfs [163, Corollaries 18.3.1 and 19.1.1]. In other words, if we define $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}} := \{v_i\}_{i=1}^{n_{\mathcal{V}}}$ as the set of all the vertices of $\mathcal{P}_{\mathcal{C}}$, then for every $f \in \mathcal{P}_{\mathcal{C}}$ we have

$$f = \sum_{i=1}^{n_{\mathcal{V}}} \alpha_i \, v_i \,, \quad (2.13)$$

where $\alpha_i \geq 0$, $i \in \{1, \ldots, n_{\mathcal{V}}\}$, and

$$\sum_{i=1}^{n_{\mathcal{V}}} \alpha_i = 1\,. \tag{2.14}$$

The set $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$ can be characterized in a graph-theoretical manner. Since we have restricted u_t to belong to a finite alphabet $\mathcal{C}$ with $n_\mathcal{C}$ elements, we note that the set of possible values for $u_{t-n_m+1:t}$, $\mathcal{C}^{n_m}$, is composed of $n_\mathcal{C}^{n_m}$ elements, which can be viewed as nodes in a graph. In addition, the transitions between the elements in $\mathcal{C}^{n_m}$, as described by a n_m-stationary pmf, are given by all the possible values of u_{t+1} when moving from $u_{t-n_m+1:t}$ to $u_{t-n_m+2:t+1}$, for all integers $t \geq 0$. The edges between the elements in $\mathcal{C}^{n_m}$ denote the possible transitions between the states, represented by the nodes of the graph. The resulting graph corresponds to an n_m-dimensional de Bruijn graph with symbols in $\mathcal{C}$. To illustrate the transition among the nodes in the equivalent de Bruijn graph, we present the following example:

Example 2.2 *Consider the de Bruijn graph depicted in Figure 2.1 on page 33, where $n_m = 2$, and $\mathcal{C} = \{0, 1\}$. From this figure we can see that, if we are at node $(0, 1)$ at time t, then we can only transit to node $(1, 0)$ or $(1, 1)$ at time $t + 1$ (cf. Table 2.1 on page 33).*

In order to describe the elements of $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$, the set of vertices of $\mathcal{P}_\mathcal{C}$, we need the concept of prime cycles, whose definition is introduced below [210, p. 678]:

Definition 2.15 *(Prime cycle) A* prime cycle *in a directed graph $\mathcal{G}_\mathcal{V}$ is an elementary cycle whose set of nodes do not have a proper subset defining an elementary cycle.*

The definition of prime cycle is illustrated in the next example.

Example 2.3 *(Prime cycles in de Bruijn graphs) Consider again the de Bruijn graph depicted in Figure 2.1 on page 33. According to Definition 2.15, the cycle $((0,1), (1,1), (1,0), (0,1))$ is not prime since it contains the elementary cycle $((0,1), (1,0), (0,1))$ in it. However, the elementary cycle $((0,1), (1,0), (0,1))$ is a prime cycle since it does not contain another elementary cycle.*

To introduce the next theorem, we need the definition below.

Definition 2.16 *(Descendants of a node in a de Bruijn graph) Consider a de Bruijn graph $\mathcal{G}_{\mathcal{C}^n}$. For any $x \in \mathcal{C}^n$, the set of* descendants *of x is defined as*

$$\mathcal{D}_x := \{v \in \mathcal{C}^n : (x, v) \in \mathcal{E}\}\,. \tag{2.15}$$

We have the following result [210, Theorem 6]:

Theorem 2.1 *The prime cycles of the de Bruijn graph $\mathcal{G}_{\mathcal{C}^{n_m}}$ are in one-to-one correspondence with the elements of $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$, the set of vertices of $\mathcal{P}_\mathcal{C}$. In particular, each $v_i \in \mathcal{V}_{\mathcal{P}_\mathcal{C}}$ is a uniform pmf whose support is the set of elements of the associated prime cycle.*

Proof *As an induction hypothesis, we assume that all measures in $\mathcal{P}_\mathcal{C}$ with less than k support points are mixtures of prime cycle measures. To start the induction, it is enough to observe that the hypothesis is trivially true when $k = 1$.*

Let $p \in \mathcal{P}_\mathcal{C}$ have k points in its support. By Lemma 2.2, for any $x \in \mathrm{supp}(p)$, where $\mathrm{supp}(p)$ is the support of p, we have

$$0 < p(x) \le \sum_{v \in \mathcal{C}} p(v, x_2, \ldots, x_{n_m}) = \sum_{v \in \mathcal{C}} p(x_2, \ldots, x_{n_m}, v) = \sum_{y \in \mathcal{D}_x} p(y)\,. \quad (2.16)$$

Equation (2.16) shows that every $x \in \mathrm{supp}(p)$ has a descendant y in $\mathrm{supp}(p)$. Because $\mathrm{supp}(p)$ is a finite set, this implies the existence of a cycle and hence a prime cycle in $\mathrm{supp}(p)$.

Let $\mathbf{a} := (a_1, \ldots, a_e)$ be one such prime cycle, so that $a_i \in \mathrm{supp}(p)$ for $i \in \{1, 2\ \ldots, e\}$. Define

$$\alpha := \min_{i \in \{1,\, 2\, \ldots,\, e\}} e\, p(a_i)\,. \quad (2.17)$$

Note that $0 < \alpha \le 1$. If $\alpha = 1$ then $p = p_\mathbf{a}$, where $p_\mathbf{a}$ is the measure assigning equal probability to all the elements in $\mathbf{a}$, and the induction is over. If $\alpha < 1$, define the measure

$$p' := \frac{p - \alpha\, p_\mathbf{a}}{1 - \alpha}\,. \quad (2.18)$$

It is easy to verify that p' is a probability measure. Moreover, p' has at most $k - 1$ support points, because

$$(1 - \alpha)p'(a_i) = p(a_i) - \alpha\, p_\mathbf{a}(a_i) = p(a_i) - \min_{i \in \{1,\, 2\, \ldots,\, e\}} e\, p(a_i)\, \frac{1}{e}\,, \quad (2.19)$$

which implies that $p'(a_i) = 0$ for some $i \in \{1, 2\ \ldots, e\}$. Finally, $p' \in \mathcal{P}_\mathcal{C}$ because p' is a linear combination of the measures p and $p_\mathbf{a}$, and $\mathcal{P}_\mathcal{C}$ is defined by linear inequalities.

By the induction hypothesis, p' is a mixture of prime cycle measures, and

$$p = \alpha\, p_\mathbf{a} + (1 - \alpha)\, p'\,. \quad (2.20)$$

This shows that any $p \in \mathcal{P}_\mathcal{C}$ is a mixture of prime cycle measures. It only remains to show that all prime cycle measures are extreme points.

Let $p_\mathbf{a}$ be a prime cycle measure. If $p_\mathbf{a}$ is a mixture of stationary measures, by what has just been shown, $p_\mathbf{a}$ is a mixture of prime cycle measures. For any $p_\mathbf{b}$ in that mixture, defined for a prime cycle $\mathbf{b}$, we have $\mathrm{supp}(p_\mathbf{b}) \subset \mathrm{supp}(p_\mathbf{a})$. But $\mathbf{a}$ is prime cycle, so the only cycle contained in $\mathbf{a}$ is itself, and hence $\mathbf{b} = \mathbf{a}$. This shows that $p_\mathbf{a}$ is an extreme point. ■

Theorem 2.1 says that we can describe all the elements in $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$ by finding all the prime cycles associated with the de Bruijn graph $\mathcal{G}_{\mathcal{C}^{n_m}}$ drawn from $\mathcal{C}^{n_m}$. To find all the prime cycles in $\mathcal{G}_{\mathcal{C}^{n_m}}$, we use the following lemma [210, Lemma 4]:

Lemma 2.3 *The prime cycles associated with $\mathcal{G}_{\mathcal{C}^{n_m}}$ are in one-to-one correspondence with the elementary cycles of $\mathcal{G}_{\mathcal{C}^{n_m-1}}$.*

Proof *Let $x, y \in \mathcal{C}^{n_m-1}$ be such that*

$$x := (x_1, \ldots, x_{n_m-1}), \tag{2.21}$$
$$y := (y_1, \ldots, y_{n_m-1}), \tag{2.22}$$

with $x_i, y_i \in \mathcal{C}$, $i \in \{1, \ldots, n_m - 1\}$, satisfying

$$(x_2, \ldots, x_{n_m-1}) = (y_1, \ldots, y_{n_m-2}). \tag{2.23}$$

Define $< x, y > \in \mathcal{C}^{n_m}$ by

$$< x, y > := (x_1, x_2, \ldots, x_{n_m-1}, y_{n_m-1}) = (x_1, y_1, \ldots, y_{n_m-2}, y_{n_m-1}). \tag{2.24}$$

For a cycle $\mathbf{a} := (a^{(1)}, \ldots, a^{(e)})$, with $a^{(i)} \in \mathcal{C}^{n_m-1}$ for all $i \in \{1, \ldots, e\}$, we create a cycle $\mathbf{a}'$ with vertices in $\mathcal{C}^{n_m}$ by defining

$$\mathbf{a}' := (< a^{(1)}, a^{(2)} >, < a^{(2)}, a^{(3)} >, \ldots, < a^{(e)}, a^{(1)} >). \tag{2.25}$$

Finally, we have that $\mathbf{a}'$ in (2.25) is a prime cycle if and only if:

(i) $< a^{(j)}, a^{(j+1)} >$ is in the set of descendants of $< a^{(i)}, a^{(i+1)} >$ if and only if $i + 1 = j$.

(ii) From item (i), we have that $a^{(i+1)} = a^{(j)}$ if and only if $i + 1 = j$.

(iii) Item (ii) follows if and only if $\mathbf{a}$ is an elementary cycle.

Therefore, all the prime cycles in de Bruijn graph $\mathcal{G}_{\mathcal{C}^{n_m}}$ can be found by computing the elementary cycles in the de Bruijn graph $\mathcal{G}_{\mathcal{C}^{n_m-1}}$. ■

Lemma 2.3 states that finding all the prime cycles in $\mathcal{G}_{\mathcal{C}^{n_m}}$ is equivalent to finding all the elementary cycles in $\mathcal{G}_{\mathcal{C}^{n_m-1}}$, which can be determined using standard graph algorithms. Moreover, the proof of Lemma 2.3 presents a procedure for computing the prime cycles in $\mathcal{G}_{\mathcal{C}^{n_m}}$ given the elementary cycles in $\mathcal{G}_{\mathcal{C}^{n_m-1}}$.

Remark 2.3 *For the examples in Chapters 4 and 5, we employ the algorithm presented in [103, pp. 79–80] complemented with the one proposed in [184, pp. 157]. Appendix A presents the pseudo-codes of these algorithms.*

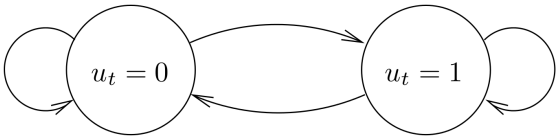

Figure 2.2: A 1-dimensional de Bruijn graph with symbols in $\mathcal{C}$, where $\mathcal{C} = \{0, 1\}$.

To illustrate the procedure for computing the prime cycles, we consider the graph depicted in Figure 2.2. One elementary cycle for the graph in Figure 2.2 is given by (0, 1, 0). Using the proof of Lemma 2.3, the elements of one prime cycle for the graph $\mathcal{G}_{\mathcal{C}^2}$ are obtained as a concatenation of the elements in the elementary cycle (0, 1, 0). Hence, the prime cycle in $\mathcal{G}_{\mathcal{C}^2}$ associated with this elementary cycle is ((0, 1), (1, 0), (0, 1)) (cf. Figure 2.1).

Once all the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_m}}$ are found, the set $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$ is fully determined. Then, for each $v_i \in \mathcal{V}_{\mathcal{P}_\mathcal{C}}$ we can generate a corresponding realization by running the corresponding prime cycle. This property will be useful in the input design method discussed in Chapters 4 and 5, where numerical approximations are needed for expressions depending on the probability measure of each prime cycle.

Example 2.4 *(Generation of a sequence from a prime cycle) Consider the de Bruijn graph depicted on Figure 2.1 on page 33. From Example 2.3, we know that one prime cycle for this graph is given by* $((0,1), (1,0), (0,1))$*. The pmf associated with this prime cycle is*

$$p(u_{t-1:t}) = \begin{cases} 0.5, & \text{if } (u_{t-1}, u_t) \in \{(0,1), (1,0)\} \\ 0, & \text{otherwise.} \end{cases} \tag{2.26}$$

In addition, a realization $u_{1:T}$ *associated with this prime cycle is computed by taking the last element of each node, i.e.,*

$$u_{1:T} = (1,\, 0,\, 1,\, 0,\, \ldots,\, ((-1)^{T-1} + 1)/2)\,. \tag{2.27}$$

We note that there is some degree of freedom on choosing from which node to start. In this example we start at node $(0,1)$ *at time* $t = 1$*.*

Before concluding this part, we show that every pmf in the set $\mathcal{P}_\mathcal{C}$ can be associated with the n_m-dimensional marginal pmf of a stationary process [210, Theorem 8]:

Theorem 2.2 *For every* $p \in \mathcal{P}_\mathcal{C}$*, there exists a stationary process* $\{u_t\}_{t\geq 1}$ *(*$u_t \in \mathcal{C}$*) such that its marginal pmf over vectors of* n_m *consecutive terms is* p*.*

Proof *Let* $a^{(i)} := (a_1^{(i)}, \ldots, a_{n_m-1}^{(i)}) \in \mathcal{C}^{n_m-1}$ *be the* i*-th vertex of the elementary cycle*

$$\mathbf{a} := (a^{(1)}, \ldots, a^{(e)}, a^{(1)})\,, \tag{2.28}$$

in the $n_m - 1$-dimensional de Bruijn graph with symbols in $\mathcal{C}$. From Lemma 2.3, the prime cycle in $\mathcal{G}_{\mathcal{C}^{n_m}}$ associated with $\mathbf{a}$ is given by

$$\mathbf{b} := ((a_1^{(1)}, \ldots, a_1^{(n_m)}), \ldots, (a_1^{(e)}, \ldots, a_1^{(e+n_m-1)}), (a_1^{(1)}, \ldots, a_1^{(n_m)})) , \quad (2.29)$$

where $a^{(me+k)} := a^{(k+1)}$ for all $k \in \{0, \ldots, e-1\}$ and all integers $m > 0$.

The prime cycle $\mathbf{b}$ can be extended to an elementary cycle on infinite dimensional de Bruijn graphs with symbols in $\mathcal{C}$ (denoted by $\mathcal{G}_{\mathcal{C}^\infty}$), which we describe in the sequel. In $\mathcal{G}_{\mathcal{C}^\infty}$, each vertex is a sequence $\{u_t\}_{t\geq 1}$, and a vertex y is the successor of the vertex $x := \{x_t\}_{t\geq 1}$ if and only if $y = \{x_t\}_{t\geq 2}$. Hence, every vertex of $\mathcal{G}_{\mathcal{C}^\infty}$ has a unique successor. A path on $\mathcal{G}_{\mathcal{C}^\infty}$ is thus completely specified by its first vertex, and it will be a cycle only if the first vertex is a repeating sequence. A prime cycle on $\mathcal{G}_{\mathcal{C}^\infty}$ thus consists of starting from a repeating sequence x, and successively shifting until after one period it returns to x. Therefore, all cycles are simply further repetitions of prime cycles. This leads to the conclusion that on $\mathcal{G}_{\mathcal{C}^\infty}$, the set of measures on cycles is identical to the set of measures on prime cycles.

Based on the previous discussion, we extend the prime cycle (2.29) *to a prime cycle on $\mathcal{G}_{\mathcal{C}^\infty}$. To this end, we make use of an extension of the proof of Lemma 2.3 to build a prime cycle on $\mathcal{G}_{\mathcal{C}^\infty}$ by extending $\mathbf{b}$ as*

$$f_{n_m,\infty}(\mathbf{b}) := \\ ((a_1^{(1)}, a_1^{(2)}, a_1^{(3)}, \ldots), \ldots, (a_1^{(e)}, a_1^{(1)}, a_1^{(2)}, \ldots), (a_1^{(1)}, a_1^{(2)}, a_1^{(3)}, \ldots)) . \quad (2.30)$$

Let $\mathbf{P}_{f_{n_m,\infty}(\mathbf{b})}$ denote the probability measure associated with the prime cycle $f_{n_m,\infty}(\mathbf{b})$ (a uniform distribution with support on the vertices of $f_{n_m,\infty}(\mathbf{b})$), and $\mathbf{P}^{(k)}_{f_{n_m,\infty}(\mathbf{b})}$ denote its k-dimensional marginal measure. Then, $p_{\mathbf{b}} = \mathbf{P}^{(n_m)}_{f_{n_m,\infty}(\mathbf{b})}$, where $p_{\mathbf{b}}$ is the pmf associated with the prime cycle $\mathbf{b}$ in $\mathcal{G}_{\mathcal{C}^{n_m}}$.

By Theorem 2.1, any $p \in \mathcal{P}_{\mathcal{C}}$ can be represented as

$$p = \sum_{i=1}^{n_{\mathcal{V}}} \alpha_i \, p_{\mathbf{b}^{(i)}} , \quad (2.31)$$

where $\{\mathbf{b}^{(i)}\}_{i=1}^{n_{\mathcal{V}}}$ are all the prime cycles in $\mathcal{G}_{\mathcal{C}^{n_m}}$, $p_{\mathbf{b}^{(i)}}$ is the pmf associated with the prime cycle $\mathbf{b}^{(i)}$, and $\{\alpha_i\}_{i=1}^{n_{\mathcal{V}}}$ are real coefficients such that $\alpha_i \geq 0$ and

$$\sum_{i=1}^{n_{\mathcal{V}}} \alpha_i = 1 . \quad (2.32)$$

Finally, the probability measure

$$\mathbf{P} = \sum_{i=1}^{n_{\mathcal{V}}} \alpha_i \, \mathbf{P}_{f_{n_m,\infty}(\mathbf{b}^{(i)})} , \quad (2.33)$$

characterizes a stationary process $\{u_t\}_{t\geq 1}$ satisfying the conditions of the theorem. ■

Remark 2.4 *The computational cost associated with the approach discussed in this part of the thesis is mostly dominated by the effort required to compute the elementary cycles, and the Fisher information matrix for these cycles. An upper bound on the complexity of computing all elementary cycles for the method presented in Appendix A is given by* $\mathcal{O}(n_{\mathcal{C}}^{n_m}(n_{\mathcal{C}}+1)(c_e+1))$*, where* c_e *is the number of elementary cycles for a fully connected graph, given by [103, p. 77]*

$$c_e := n_{\mathcal{C}} + \sum_{i=1}^{n_{\mathcal{C}}^{n_m-1}-1} \binom{n_{\mathcal{C}}^{n_m-1}}{n_{\mathcal{C}}^{n_m-1}-i+1} \left(n_{\mathcal{C}}^{n_m-1}-i\right)! \,. \tag{2.34}$$

2.3 Generation of stationary Markov processes

In Section 2.2, we introduced a parametrization of $\mathcal{P}_{\mathcal{C}}$ to obtain a computationally tractable description of the n_m-dimensional marginal pmfs of stationary processes. However, to use this technique in Chapters 4 and 5, we need to obtain an input vector $u_{1:T}$, where each u_t is sampled from a given $p \in \mathcal{P}_{\mathcal{C}}$, for $t \in \{1, \ldots, T\}$.

In this section we develop a procedure to generate an input sequence $u_{1:T}$ as a realization of a Markov process with stationary pmf $p(u_{1:n_m})$. To this end, notice that we can associate $\mathcal{G}_{\mathcal{C}^{n_m}}$ with the discrete-time Markov chain [58]

$$p_{t+1} = A\, p_t \,, \tag{2.35}$$

where $A \in \mathbb{R}^{\mathcal{C}^{n_m} \times \mathcal{C}^{n_m}}$ is a transition probability matrix[1], and $p_t \in \mathbb{R}^{\mathcal{C}^{n_m}}$ is a vector[2] whose entries are the probabilities assigned to the vertices in $\mathcal{C}^{n_m}$ at time t. In this case, there is a one-to-one correspondence between each entry of $p_t \in \mathbb{R}^{\mathcal{C}^{n_m}}$ and an element of $\mathcal{C}^{n_m}$.

Based on this association, $p(u_{1:n_m})$ corresponds to the stationary distribution of a Markov chain (2.35), defined as $p^{\mathrm{st}} \in \mathbb{R}^{\mathcal{C}^{n_m}}$. Therefore, in order to generate an input sequence $u_{1:T}$, we design a Markov chain having $p(u_{1:n_m})$ as its stationary distribution, and simulate this Markov chain to generate $u_{1:T}$ from its samples in stationary regime.

To continue, we denote by $A_{rl} \in \mathbb{R}$ the (r,l)-entry of A. For convenience, the indices of A are not numerical, but belong to $\mathcal{C}^{n_m}$. Thus, a valid A for the Markov chain (2.35) must satisfy

$$A_{rl} \geq 0 \,, \text{ for all } r, l \in \mathcal{C}^{n_m} \,, \tag{2.36}$$

$$\sum_{r \in \mathcal{C}^{n_m}} A_{rl} = 1 \,, \text{ for all } l \in \mathcal{C}^{n_m} \,, \tag{2.37}$$

$$A_{rl} = 0 \,, \text{ if } (l,\, r) \notin \mathcal{E} \,. \tag{2.38}$$

[1] Given a set X with finite cardinality, we denote by $\mathbb{R}^{X \times X}$ the matrices with real entries, with dimensions given by the cardinality of X.

[2] Note that equation (2.35) is not in standard Markov chain notation (defined as the transpose of (2.35)) [134].

It can be proven that a matrix A satisfying (2.36) and (2.37) has 1 as an eigenvalue [100]. Furthermore, if the Markov chain is ergodic, the unique eigenvector $p^{\mathrm{st}} \in \mathbb{R}^{\mathcal{C}^{n_m}}$ associated with this eigenvalue is the unique stationary pmf of $\mathcal{C}^{n_m}$ (up to a scaling factor), satisfying

$$p^{\mathrm{st}} = Ap^{\mathrm{st}}\,. \tag{2.39}$$

The task is to design a transition probability matrix satisfying (2.36)-(2.39). There is an extensive literature on how to optimize the mixing time of the resulting Markov chain (i.e., the time required to obtain samples distributed according to the stationary measure of the Markov chain, see, e.g., [19, 94] and the references therein). However, these works assume that the graph is undirected or reversible, which implies that A must have a particular structure (e.g., to be symmetric). Since the structure of the graph $\mathcal{G}_{\mathcal{C}^{n_m}}$ does not satisfy in general these properties, most existing methods cannot be applied here.

Below, we develop a method to design a transition probability matrix for the de Bruijn graph $\mathcal{G}_{\mathcal{C}^{n_m}}$. The idea is that if we parameterize the transition probabilities of a Markov process of memory n in terms of the stationary probabilities of a Markov process of memory $n+1$, we obtain a computationally tractable description of $\mathcal{P}_{\mathcal{C}}$, as discussed in Section 2.2. Given a pmf $p \in \mathcal{P}_{\mathcal{C}}$, the proposed algorithm gives a unique mapping between p and the transition matrix associated with a Markov process of memory n by setting

$$p(u_t|u_{t-1},\,\ldots,\,u_{t-n}) = \frac{p(u_t,\,\ldots,\,u_{t-n})}{\sum_{u_t\in\mathcal{C}} p(u_t,\,\ldots,\,u_{t-n})}\,, \tag{2.40}$$

To continue, we need the following:

Assumption 2.2 *The pmf $p(u_{1:n_m})$ satisfies*

$$\sum_{r=1}^{n_{\mathcal{C}}} p(v_1,\,\ldots,\,v_{n_m-1},\,s_r) = \sum_{r=1}^{n_{\mathcal{C}}} p(s_r,\,v_1,\,\ldots,\,v_{n_m-1})\,, \tag{2.41}$$

for all $(v_1,\,\ldots,\,v_{n_m-1}) \in \mathcal{C}^{n_m-1}$.

Assumption 2.2 is satisfied in our context, as $p(u_{1:n_m})$ is a marginal pmf of a stationary process (cf. Lemma 2.2 and Theorem 2.2). Based on this fact, we can design a transition probability matrix A for $\mathcal{G}_{\mathcal{C}^{n_m}}$ as described in Algorithm 2.1.

Algorithm 2.1 introduces a method to design valid transition probability matrices when p satisfies Assumption 2.2. For an element $r \in \mathcal{C}^{n_m}$ with nonzero probability $p(r)$, Algorithm 2.1 assigns nonzero transition probability only to those elements in the set of ancestors of r. If an element $v \in \mathcal{C}^{n_m}$ has zero probability, the algorithm assigns equal probability to those elements in the set of ancestors of v.

The next theorem establishes the correctness of the algorithm.

Theorem 2.3 *Let p be a pmf satisfying Assumption 2.2. Then, the matrix $A \in \mathbb{R}^{\mathcal{C}^{n_m}\times\mathcal{C}^{n_m}}$ designed by Algorithm 2.1 is a transition probability matrix satisfying* (2.36)-(2.39).

Algorithm 2.1 Design of a transition probability matrix

INPUTS: A pmf $p\colon \mathcal{C}^{n_m} \to \mathbb{R}$, and the alphabet $\mathcal{C}$.
OUTPUT: A transition probability matrix A with stationary distribution p.

1: For each $r \in \mathcal{C}^{n_m}$, define the set of ancestors of r by

$$\mathcal{A}_r := \{l \in \mathcal{C}^{n_m} : (l, r) \in \mathcal{E}\}\,. \tag{2.42}$$

2: For each r, $l \in \mathcal{C}^{n_m}$, let

$$A_{rl} = \begin{cases} \frac{p(r)}{\sum\limits_{k\in\mathcal{A}_r} p(k)}\,, & \text{if } l \in \mathcal{A}_r \text{ and } \sum\limits_{k\in\mathcal{A}_r} p(k) \neq 0\,, \\ \frac{1}{\#\mathcal{A}_r}\,, & \text{if } l \in \mathcal{A}_r \text{ and } \sum\limits_{k\in\mathcal{A}_r} p(k) = 0\,, \\ 0\,, & \text{otherwise.} \end{cases} \tag{2.43}$$

Proof *Properties* (2.36) *and* (2.38) *are trivially satisfied by the construction of* A. *To establish* (2.37) *and* (2.39), *we need to analyze the structure of the transition probability matrix* A *associated with a de Bruijn graph. From the definition of* $\mathcal{E}$ *(cf. Equation* (2.2)*), we have that*

$$\sum_{l\in\mathcal{A}_r} p(l) = \sum_{l=1}^{n_\mathcal{C}} p(s_l, r_1, \ldots, r_{n_m-1})\,. \tag{2.44}$$

To proceed, we need the set of descendants of r, *denoted by* $\mathcal{D}_r$. *From the definition of* $\mathcal{E}$ *in equation* (2.2), *we have that* $\#\mathcal{D}_r = \#\mathcal{A}_r = n_\mathcal{C}$.

First, we prove (2.37). *Consider first an* $l \in \mathcal{C}^{n_m}$ *such that* $\sum_{k\in\mathcal{A}_r} p(k) \neq 0$ *for all* $r \in \mathcal{D}_l$. *Then,*

$$\mathcal{D}_l = \{(l_2, \ldots, l_{n_m}, s_1), \ldots, (l_2, \ldots, l_{n_m}, s_{n_\mathcal{C}})\}\,. \tag{2.45}$$

In addition, for any $r \in \mathcal{D}_l$,

$$\mathcal{A}_r = \{(s_1, l_2, \ldots, l_{n_m}), \ldots, (s_{n_\mathcal{C}}, l_2, \ldots, l_{n_m})\}\,. \tag{2.46}$$

Equation (2.46) *shows that the sets* $\mathcal{A}_r$ *are equal for all* $r \in \mathcal{D}_l$. *Therefore, the sums* $\sum_{k\in\mathcal{A}_r} p(k)$ *are equal (and nonzero) for all* $r \in \mathcal{D}_l$, *hence*

$$\sum_{r\in\mathcal{C}^{n_m}} A_{rl} = \sum_{r\in\mathcal{D}_l} \frac{p(r)}{\sum_{k\in\mathcal{A}_r} p(k)} = \frac{\sum_{r\in\mathcal{D}_l} p(r)}{\sum_{k\in\mathcal{A}_{\tilde{r}}} p(k)}\,, \tag{2.47}$$

for any fixed $\tilde{r} \in \mathcal{D}_l$. *Furthermore, in the light of* (2.45)-(2.46), *we can rewrite* (2.47) *as*

$$\sum_{r\in\mathcal{C}^{n_m}} A_{rl} = \frac{\sum_{r=1}^{n_\mathcal{C}} p(l_2, \ldots, l_{n_m}, s_r)}{\sum_{k=1}^{n_\mathcal{C}} p(s_k, l_2, \ldots, l_{n_m})} = 1, \tag{2.48}$$

where the last equality follows from Lemma 2.2.

On the other hand, if $l \in \mathcal{C}^{n_m}$ *is such that* $\sum_{k\in\mathcal{A}_r} p(k) = 0$ *for all* $r \in \mathcal{C}^{n_m}$*, we write*

$$\sum_{r\in\mathcal{C}^{n_m}} A_{rl} = \sum_{r\in\mathcal{D}_l} \frac{1}{\#\mathcal{A}_r} = \sum_{r\in\mathcal{D}_l} \frac{1}{\#\mathcal{D}_l} = 1\,. \tag{2.49}$$

The results presented in (2.48)-(2.49) *establish* (2.37).

Now we prove (2.39)*. For each* $r \in \mathcal{C}^{n_m}$ *such that* $\sum_{k\in\mathcal{A}_r} p(k) \neq 0$*, we have that the* r*-th element of the product* $Ap^{\rm st}$ *(denoted by* $p_r^{\rm st}$*) is given by*

$$p_r^{\rm st} = \frac{p(r)}{\sum_{l\in\mathcal{A}_r} p(l)} \sum_{k\in\mathcal{A}_r} p(k) = p(r)\,. \tag{2.50}$$

On the other hand, for each $r \in \mathcal{C}^{n_m}$ *such that* $\sum_{k\in\mathcal{A}_r} p(k) = 0$*, we can consider an* $l \in \mathcal{C}^{n_m}$ *such that* $r \in \mathcal{D}_l$*. According to* (2.45)*,* (2.46)*, and using Lemma 2.2, we can conclude that*

$$\sum_{\tilde{r}\in\mathcal{D}_l} p(\tilde{r}) = \sum_{k\in\mathcal{A}_r} p(k) = 0\,, \tag{2.51}$$

which implies that

$$p(k) = 0 \tag{2.52}$$

for all $k \in \mathcal{D}_l$*, and in particular, for* $k = r$*. Since* $l \in \mathcal{C}^{n_m}$ *is arbitrary,* (2.52) *is true for all* $l \in \mathcal{C}^{n_m}$ *such that* $r \in \mathcal{D}_l$*. Hence,* (2.50) *is also satisfied for each* $r \in \mathcal{C}^{n_m}$ *such that* $\sum_{k\in\mathcal{A}_r} p(k) = 0$*, which establishes* (2.39)*. This concludes the proof.* ■

The transition probability matrix given by Algorithm 2.1 has the following property:

Theorem 2.4 *The transition probability matrix* $A \in \mathbb{R}^{\mathcal{C}^{n_m}\times\mathcal{C}^{n_m}}$ *designed by Algorithm 2.1 has all its eigenvalues in the region* $\mathbf{D} := \{z \in \mathbb{C} : |z| \leq 1\}$*. In addition,* A *has at most* $n_{\mathcal{C}}^{n_m-1}$ *nonzero eigenvalues in* $\mathbf{D}$*.*

Proof *The first statement follows since* A *is a transition probability matrix [100], according to Theorem 2.3.*

To establish the second statement, notice that, from (2.45)-(2.46)*, for each* $l \in \mathcal{C}^{n_m}$ *we have that* $\mathcal{A}_r$ *is the same for all* $r \in \mathcal{D}_l$*, which means that the columns of* A *can be partitioned into* $n_{\mathcal{C}}^{n_m-1}$ *groups of* $n_{\mathcal{C}}$ *identical columns. Therefore, the number of nonzero eigenvalues of* A *in* $\mathbf{D}$ *is at most* $n_{\mathcal{C}}^{n_m-1}$*, since there are at most* $n_{\mathcal{C}}^{n_m-1}$ *linearly independent columns in* A*. This concludes the proof.* ■

Remark 2.5 *There are, in general, several transition matrices having a given* $\mathbf{P}$ *as stationary probability measure, subject to a graph constraint (a prescribed set of edges). Algorithm 2.1 provides only one such choice. Among those transition matrices, it would be preferable to select the one with the fastest mixing time, i.e., for which the Markov chain reaches the stationary distribution as quickly as possible.*

The most common criterion to define mixing time is the second largest eigenvalue modulus (SLEM). A Monte Carlo study, for $n_{\mathcal{C}} = 2$ and $n_m = 2$, based on uniform sampling from the set of transition matrices giving a specific p^{st} (which can be shown to be a polytope) has empirically shown that the A matrix given by Algorithm 2.1 is within the 7 % of those with lowest SLEM, which suggests that Algorithm 2.1 gives a reasonable (but improvable) mixing time. One way to further reduce the SLEM of A is by performing gradient descent over the transition probabilities in A, starting from the matrix designed in Algorithm 2.1. Another option to reduce the SLEM is by exploiting the full memory of the Markov chain when designing the transition probability matrix. This is part of the future work on the subject.

Remark 2.6 *For simplicity, the results introduced in this thesis are discussed for scalar sequences. However, an immediate extension of this technique to the case of sequences of vectors can be done. In the case of sequences of vectors with n_u entries, the states associated with each node in the de Bruijn graph are the possible values of an $n_u \times n_m$ matrix, where the i-th row describes the feasible states for the stationary process in the i-th entry. With this modification, the method can be directly employed to solve input design problems for MIMO models.*

2.4 Conclusion

This chapter introduced a link between graph theoretical concepts and stationary processes. In particular, this chapter shows that a specific family of graphs, the de Bruijn graphs, can be employed to characterize marginal cdfs of stationary processes. The link between de Bruijn graphs and stationary processes is then used to parameterize the set of marginal pmfs of stationary processes in a tractable manner. By the use of standard algorithms to compute elementary cycles in a graph, the vertices of the polytope characterizing the set of marginal pmfs of stationary processes are obtained.

By restricting the pmfs to the set of marginal pmfs of stationary processes, this chapter also introduces a procedure to construct a Markov process with stationary pmf in such set. The Markov process built in this manner is then employed to generate samples distributed according to the stationary pmf.

Chapter 3

A review on sequential Monte Carlo methods

In this chapter we review the fundamentals of sequential Monte Carlo (SMC) methods, which are needed in the input design methods developed in Chapters 4 and 6.

SMC methods are a family of techniques that can be used, e.g., to estimate the filtering and smoothing distributions in state space models (SSM). Here we describe the idea behind SMC methods, and we introduce the required material to understand the estimation algorithms used in Chapters 4 and 6. The material in this chapter is based on [55, 61, 170].

3.1 SMC methods

The objective of SMC methods is to sample sequentially from target probability densities $\{p(x_{1:k})\}_{k\geq 1}$ of increasing dimension, where $x_k \in \mathcal{X}$, and each distribution $p(x_{1:k})$ is defined on the product space $\mathcal{X}^k$. Writing

$$p(x_{1:k}) = \frac{\gamma_k(x_{1:k})}{Z_k}\,, \tag{3.1}$$

it is only required that $\gamma_k \colon \mathcal{X}^k \to \mathbb{R}^+$ be computable; the normalizing constant

$$Z_k = \int_{\mathcal{X}^k} \gamma_k(x_{1:k}) dx_{1:k}\,, \tag{3.2}$$

might be unknown. SMC methods provides an approximation of $p(x_1)$ and an estimate of Z_1 at time 1, then an approximation of $p(x_{1:2})$, and an estimate of Z_2 at time 2, and so on.

Monte Carlo methods

Consider initially approximating a generic probability density function $p(x_{1:T})$ for some fixed T. If we sample N independent random vectors $\{x_{1:T}^{(i)}\}_{i=1}^{N}$ from $p(x_{1:T})$, then the Monte Carlo method approximates expectations of functions of $x_{1:T}$ in the mean square sense by using

$$\hat{p}(x_{1:T}) := \frac{1}{N}\sum_{i=1}^{N} \delta\left(x_{1:T} - x_{1:T}^{(i)}\right), \tag{3.3}$$

instead of $p(x_{1:T})$, where $\delta(x)$ denotes the Dirac delta at $x = 0$. Based on this approximation, it is possible to approximate any marginal $p(x_k)$ by

$$\hat{p}(x_k) = \frac{1}{N}\sum_{i=1}^{N} \delta\left(x_k - x_k^{(i)}\right), \tag{3.4}$$

and the expectation of any measurable function $\varphi\colon \mathcal{X}^T \to \mathbb{R}$, given by

$$\mathbf{E}\left\{\varphi(x_{1:T})\right\} = \int_{\mathcal{X}^T} \varphi(x_{1:T}) p(x_{1:T})\, dx_{1:T}, \tag{3.5}$$

is estimated as

$$\hat{\mathbf{E}}^{\mathrm{MC}}\left\{\varphi(x_{1:T})\right\} := \int_{\mathcal{X}^T} \varphi(x_{1:T})\hat{p}(x_{1:T})\, dx_{1:T} = \frac{1}{N}\sum_{i=1}^{N} \varphi\left(x_{1:T}^{(i)}\right). \tag{3.6}$$

It is easy to show that (3.6) is an unbiased estimate of (3.5), and that its variance is given by

$$\mathrm{Var}\left\{\hat{\mathbf{E}}^{\mathrm{MC}}\left\{\varphi(x_{1:T})\right\}\right\} = \frac{1}{N}\left\{\int_{\mathcal{X}^T} \varphi^2(x_{1:T}) p(x_{1:T})\, dx_{1:T} - \mathbf{E}\left\{\varphi(x_{1:T})\right\}^2\right\}. \tag{3.7}$$

The main advantage of Monte Carlo methods over standard approximation techniques is that the variance of the approximation error decreases at a rate of $\mathcal{O}(1/N)$ regardless of the dimension of the space $\mathcal{X}^T$. However, there are at least two main problems with this idea:

(i) If $p(x_{1:T})$ is a highly complex probability density function, then we may not be able to sample directly from it.

(ii) Even if we knew how to sample exactly from $p(x_{1:T})$, the computational complexity of such a sampling scheme is typically (at least) linear in the number of variables T. Therefore, an algorithm sampling exactly from $p(x_{1:T})$ sequentially for each value of T, would have a computational complexity increasing at least linearly with T.

Importance sampling

In this part we address Problem (i) using the importance sampling (IS) method. IS relies on the introduction of an *importance density* $p_{\mathrm{I}}(x_{1:T})$ such that

$$p(x_{1:T}) > 0 \text{ implies } p_{\mathrm{I}}(x_{1:T}) > 0\,, \tag{3.8}$$

i.e., the distributions satisfy $\mathrm{supp}(p) \subseteq \mathrm{supp}(p_{\mathrm{I}})$. In this case, we have from (3.1)-(3.2) the following identities:

$$p(x_{1:T}) = \frac{\tilde{w}_T(x_{1:T})p_{\mathrm{I}}(x_{1:T})}{Z_T}\,, \tag{3.9}$$

$$Z_T = \int_{\mathcal{X}^T} \tilde{w}_T(x_{1:T})p_{\mathrm{I}}(x_{1:T})\,dx_{1:T}\,, \tag{3.10}$$

where $w_T(x_{1:T})$ is the *unnormalized weight* function

$$\tilde{w}_T(x_{1:T}) = \frac{\gamma_T(x_{1:T})}{p_{\mathrm{I}}(x_{1:T})}\,. \tag{3.11}$$

In particular, we can select an importance density $p_{\mathrm{I}}(x_{1:T})$ from which it is easy to draw samples, e.g., the normal distribution, if this is consistent with the support of p.

If we assume we draw N independent samples $\{x_{1:T}^{(i)}\}_{i=1}^N$ from $p_{\mathrm{I}}(x_{1:T})$, then by inserting the Monte Carlo approximation of $p_{\mathrm{I}}(x_{1:T})$ (the empirical distribution of the samples $\{x_{1:T}^{(i)}\}_{i=1}^N$) into (3.9)-(3.10), we have

$$\hat{p}(x_{1:T}) = \sum_{i=1}^N w_T^{(i)}\,\delta\left(x_{1:T} - x_{1:T}^{(i)}\right)\,, \tag{3.12}$$

$$\hat{Z}_T = \frac{1}{N}\sum_{i=1}^N \tilde{w}_T\left(x_{1:T}^{(i)}\right)\,, \tag{3.13}$$

where

$$w_T^{(i)} = \frac{\tilde{w}_T\left(x_{1:T}^{(i)}\right)}{\sum_{j=1}^N \tilde{w}_T\left(x_{1:T}^{(j)}\right)}\,, \tag{3.14}$$

denotes the normalized weights.

Based on the estimate (3.12), we can compute an estimate of (3.5) as

$$\hat{\mathbf{E}}^{\mathrm{IS}}\left\{\varphi(x_{1:T})\right\} := \int_{\mathcal{X}^T} \varphi(x_{1:T})\hat{p}(x_{1:T})\,dx_{1:T} = \sum_{i=1}^N w_T^{(i)}\varphi\left(x_{1:T}^{(i)}\right)\,. \tag{3.15}$$

Unlike the estimate $\hat{\mathbf{E}}^{\mathrm{MC}}\left\{\varphi(x_{1:T})\right\}$, the expression (3.15) is in general biased for finite N. However, (3.15) is consistent as $N \to \infty$. The expression for the asymptotic bias is given by

$$\lim_{N\to\infty} N\left(\hat{\mathbf{E}}^{\mathrm{IS}}\left\{\varphi(x_{1:T})\right\} - \mathbf{E}\left\{\varphi(x_{1:T})\right\}\right) = \\ -\int_{\mathcal{X}^T} \frac{p^2(x_{1:T})}{p_{\mathrm{I}}(x_{1:T})}\left(\varphi(x_{1:T}) - \mathbf{E}\left\{\varphi(x_{1:T})\right\}\right) dx_{1:T}\,. \quad (3.16)$$

Furthermore, $\hat{\mathbf{E}}^{\mathrm{IS}}\left\{\varphi(x_{1:T})\right\}$ satisfies, under mild conditions,

$$\sqrt{N}\left(\hat{\mathbf{E}}^{\mathrm{IS}}\left\{\varphi(x_{1:T})\right\} - \mathbf{E}\left\{\varphi(x_{1:T})\right\}\right) \xrightarrow{d} \mathcal{N}\left(0,\, M^{\mathrm{IS}}\right)\,, \quad (3.17)$$

where

$$M^{\mathrm{IS}} := \int_{\mathcal{X}^T} \frac{p^2(x_{1:T})}{p_{\mathrm{I}}(x_{1:T})}\left(\varphi(x_{1:T}) - \mathbf{E}\left\{\varphi(x_{1:T})\right\}\right)^2 dx_{1:T}\,. \quad (3.18)$$

As we can see from (3.16)-(3.18), both the bias and the variance for this method are of order $\mathcal{O}(1/N)$.

The next sampling method addresses the increasing computational complexity in the sample length T.

Sequential importance sampling

Sequential importance sampling (SIS) is a method that admits a fixed computational complexity at each time step, and thus addresses Problem (ii). This method considers an importance distribution which has the following structure:

$$p_{\mathrm{I}}(x_{1:T}) = p_{\mathrm{I}}(x_{1:T-1})p_{\mathrm{I}}(x_T|x_{1:T-1}) \\ = p_{\mathrm{I}}(x_1)\prod_{t=2}^{T} p_{\mathrm{I}}(x_t|x_{1:t-1})\,. \quad (3.19)$$

From a practical perspective, Equation (3.19) means that to obtain particles $x_{1:T}^{(i)}$ distributed according to $p_{\mathrm{I},T}(x_{1:T})$ at time T, $x_1^{(i)}$ is sampled from $p_{\mathrm{I}}(x_1)$ at time 1, then $x_t^{(i)}$ is sampled from $p_{\mathrm{I}}(x_t|x_{1:t-1}^{(i)})$ at time t, for $t \in \{2, \ldots, T\}$. The associated unnormalized weights can be computed recursively using the decomposition

$$\tilde{w}_T(x_{1:T}) = \frac{\gamma_T(x_{1:T})}{p_{\mathrm{I}}(x_{1:T})} \\ = \frac{\gamma_{T-1}(x_{1:T-1})}{p_{\mathrm{I}}(x_{1:T-1})} \cdot \frac{\gamma_T(x_{1:T})}{\gamma_{T-1}(x_{1:T-1})p_{\mathrm{I}}(x_T|x_{1:T-1})}\,, \quad (3.20)$$

which can be written in the form

$$\tilde{w}_T(x_{1:T}) = \tilde{w}_{T-1}(x_{1:T-1}) \cdot \alpha_T(x_{1:T}) \\ = \tilde{w}_1(x_1)\prod_{k=2}^{T} \alpha_k(x_{1:k})\,, \quad (3.21)$$

Algorithm 3.1 Sequential importance sampling

INPUTS: N (number of samples), $p_{\mathrm{I}}(x_{1:T})$ (importance distribution), and $\gamma_T(x_{1:T})$ (unnormalized pdf).
OUTPUT: $\{x_{1:T}^{(i)}\}_{i=1}^N$ (realizations), and $\{w_T^{(i)}\}_{i=1}^N$ (normalized weights).

1: Sample $x_1^{(i)}$ from $p_{\mathrm{I}}(x_1)$ for $i = 1$ to N.
2: Compute the weights $\tilde{w}_1(x_1^{(i)}) = p_{\mathrm{I}}(x_1^{(i)})$, and

$$w_1^{(i)} = \frac{\tilde{w}_1(x_1^{(i)})}{\sum_{j=1}^N \tilde{w}_1(x_1^{(i)})}\,, \tag{3.23}$$

for $i = 1$ to N.
3: **for** $t = 2$ to T **do**
4: Sample $x_t^{(i)}$ from $p_{\mathrm{I}}(x_t|x_{1:t-1}^{(i)})$ for $i = 1$ to N.
5: Compute the weights

$$\tilde{w}_t(x_{1:t}^{(i)}) = \tilde{w}_{t-1}(x_{1:t-1}^{(i)})\,\alpha_t(x_{1:t}^{(i)})\,, \tag{3.24}$$

where α_t given in (3.22), and

$$w_t^{(i)} = \frac{\tilde{w}_t(x_{1:t}^{(i)})}{\sum_{j=1}^N \tilde{w}_t(x_{1:t}^{(i)})}\,, \tag{3.25}$$

for $i = 1$ to N.
6: **end for**

where the *incremental importance weight* function $\alpha_k(x_{1:k})$ is given by

$$\alpha_k(x_{1:k}) := \frac{\gamma_k(x_{1:k})}{\gamma_{k-1}(x_{1:k-1})p_{\mathrm{I}}(x_k|x_{1:k-1})}\,. \tag{3.22}$$

The SIS method is summarized in Algorithm 3.1.

Algorithm 3.1 provides the estimates $\hat{p}(x_{1:t})$ and $\hat{Z}_t$ (equations (3.12)-(3.13)) of $p(x_{1:t})$ and Z_t at any time t. In this framework, it seems that the only degree of freedom the user has at time t is the choice of $p_{\mathrm{I}}(x_t|x_{1:t-1})$ (the number of samples N is also a degree of freedom, but it is fixed before executing the algorithm). A sensible strategy consists in selecting $p_{\mathrm{I}}(x_t|x_{1:t-1})$ so as to minimize the variance of $\tilde{w}_t(x_{1:t})$. This is achieved by selecting

$$p_{\mathrm{I}}^{\mathrm{opt}}(x_t|x_{1:t-1}) = p(x_t|x_{1:t-1})\,, \tag{3.26}$$

as in this case the variance of $\tilde{w}_t(x_{1:t})$ conditioned on $x_{1:t-1}$ is zero, and the associated incremental weight is given by

$$\alpha_t^{\mathrm{opt}}(x_{1:t}) = \frac{\gamma_t(x_{1:t-1})}{\gamma_{t-1}(x_{1:t-1})} = \frac{\int_{\mathcal{X}} \gamma_t(x_{1:t})\,dx_t}{\gamma_{t-1}(x_{1:t-1})}\,. \tag{3.27}$$

We note that it is not always possible to sample from $p(x_t|x_{1:t-1})$ nor to compute $\alpha_t^{\text{opt}}(x_{1:t})$. In these cases, we need to employ an approximation of $p_{\text{I}}^{\text{opt}}(x_t|x_{1:t-1})$ for $p_{\text{I}}(x_t|x_{1:t-1})$.

In those scenarios in which the time required to sample from $p_{\text{I}}(x_t|x_{1:t-1})$ and to compute $\alpha_t(x_{1:t})$ is independent of t (which is the case if p_{I} is chosen carefully and one is concerned with a problem such as filtering), it appears that SIS provides a solution for issue (ii). However, the SIS method suffers from severe drawbacks. Even for standard IS, the variance of the resulting estimates increases exponentially with t [110]. A method to overcome this difficulty is presented in the next section.

Resampling

We have commented that IS (and therefore SIS) provides estimates whose variance increases with t. Resampling techniques are a key ingredient of SMC methods which (partially) solve this problem in some important scenarios.

Resampling is an intuitive idea with major practical and theoretical benefits. We consider first an IS approximation $\hat{p}(x_{1:T})$ of the target distribution $p(x_{1:T})$. This approximation is based on weighted samples from $p_{\text{I}}(x_{1:T})$, and does not provide samples approximately distributed according to $p(x_{1:T})$ as the samples are obtained from p_{I}. To obtain approximate samples from $p(x_{1:T})$, we can simply draw samples from its IS approximation $\hat{p}(x_{1:T})$, where $\hat{p}(x_{1:T})$ is defined in (3.12), with normalized weights $\{w_T^{(i)}\}_{i=1}^N$. Then we select $x_{1:T}^{(i)}$ with probability $w_T^{(i)}$. This operation is called *resampling* as it corresponds to sampling from an approximation $\hat{p}(x_{1:T})$ which was itself obtained by sampling. If we are interested in obtaining N samples from $\hat{p}(x_{1:T})$, then we can resample N times from $\hat{p}(x_{1:T})$. This is equivalent to associating a number of offsprings $N_T^{(i)}$ with each sample $x_{1:T}^{(i)}$ in such a way that $N_T^{(1:N)} := \left(N_T^{(1)}, \ldots, N_T^{(N)}\right)$ follows a multinomial distribution with parameter vector $(N, w_T^{(1:N)})$, and associating a weight of $1/N$ with each offspring. Thus, we approximate $\hat{p}(x_{1:T})$ by the resampled empirical measure

$$\overline{p}(x_{1:T}) = \sum_{i=1}^{N} \frac{N_T^{(i)}}{N} \delta\left(x_{1:T} - x_{1:T}^{(i)}\right) , \tag{3.28}$$

where $\mathbf{E}\left\{N_T^{(i)} \middle| w_T^{(1:N)}\right\} = N w_T^{(i)}$. Hence $\overline{p}(x_{1:T})$ is an unbiased approximation of $\hat{p}(x_{1:T})$.

Many resampling schemes have been proposed in the literature [61]. One of the most popular algorithms is *systematic resampling*, which is described in Algorithm 3.2. It can be shown that systematic resampling gives an unbiased estimate of the distribution $p(x_{1:T})$.

We should clarify that resampling retrieves an estimate of $\mathbf{E}\{\varphi(x_{1:T})\}$ with higher variance than the one obtained by using $\hat{p}(x_{1:T})$. However, by resampling we remove samples with low weights with high probability, which is useful when

Algorithm 3.2 Systematic resampling

INPUTS: $\{x_{1:T}^{(i)}\}_{i=1}^N$ (realizations), and $\{w_T^{(i)}\}_{i=1}^N$ (normalized weights).
OUTPUT: $\{x_{1:T}^{(i)}\}_{i=1}^N$ (realizations), and $\{N_T^{(i)}/N\}_{i=1}^N$ (resampled weights).

1: Sample $U_{(1)}$ from a uniform distribution defined on $[0,\, 1/N]$.
2: Define $U_{(i)} := U_{(1)} + (i-1)/N$ for $i \in \{2,\, \ldots,\, N\}$.
3: **for** $i = 1$ to N **do**
4: Set $N_T^{(i)} = \#\left\{U_{(j)} : \sum_{k=1}^{i-1} w_T^{(k)} \leq U_{(j)} \leq \sum_{k=1}^{i} w_T^{(k)}\right\}$, where $\sum_{k=1}^{0} \mu_k := 0$.
5: **end for**

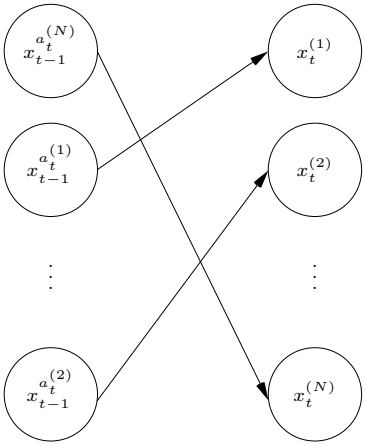

Figure 3.1: Illustration of the ancestor index $a_t^{(i)}$.

working with sequential techniques. If samples with low weights are preserved, then the approximation $\hat{p}(x_{1:T})$ will lose accuracy as we increase the path length T. This problem is known as particle degeneracy, and is discussed in more detail on page 58.

Remark 3.1 *The notion of offspring is not only important when talking about resampling, but also for some of the particle methods discussed in the next sections. For future reference, we introduce the ancestor index $a_t^{(i)}$. Given a set of particles at time t, $\{x_t^{(k)}\}_{k=1}^N$, we denote by $a_t^{(i)}$ the index of the particle at time $t-1$ from which $x_t^{(i)}$ was generated. Figure 3.1 illustrates this idea. In Figure 3.1, the directed edges between two nodes at different time instants express which particles at time $t-1$ generate which particles at time t.*

3.2 Particle filtering

The name *particle filter* (PF) alludes to an SMC method for obtaining estimates of the sequence of target densities $\{p(x_{1:t}|y_{1:t})\}_{t=1}^{T}$ or their marginals $\{p(x_t|y_{1:t})\}_{t=1}^{T}$. Here we assume that the underlying process is described for all $t \geq 1$ by Equation (1.5) on page 4.

The basic idea is to leverage SIS and resampling to propagate a collection of N weighted random samples $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$ forward in time. These samples (commonly referred to as *particles*) constitute an empirical approximation that converges to the underlying target density $p(x_t|y_{1:t})$ as $N \to \infty$.

The particle filter can be viewed as a framework to sequentially approximate the filtering densities $\{p(x_t|y_{1:t})\}_{t=1}^{T}$. The resulting approximation is an empirical distribution of the form

$$\hat{p}(x_t|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)}\, \delta(x_t - x_t^{(i)})\,. \tag{3.29}$$

Intuitively speaking, each particle $x_t^{(i)}$ can be understood as a possible state of the underlying system, where the corresponding weight $w_t^{(i)}$ contains information about how probable that particular state is.

In this subsection we introduce a method that will prove to be useful for the results presented in Chapter 4: the auxiliary particle filter (APF).

The auxiliary particle filter

The *auxiliary particle filter* (APF) [148] is a method that can be used to include the information available in the current observation y_t not only for proposing the new state x_t (as it is done in the PF method[1]), but also when proposing the ancestor index a_t (recall that a_t is the ancestor index of the particles at time t, cf. Remark 3.1). That is, we increase the probability of resampling particles at time $t-1$ that agree with the current observation y_t when compared with the PF approach.

To continue, given a function $\nu : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ to be specified by the user, for each particle i, $i \in \{1, \ldots, N\}$ we compute the factors

$$\nu_{t-1}^{(i)} := \nu(x_{t-1}^{(i)},\, y_t)\,, \tag{3.30}$$

referred to as *adjustment multipliers*. The adjustment multipliers are functions, which will be used to adjust the empirical distributions to sample and propagate the particles. We note that (3.30) depends only on the previous particles and on the current observation, which are available in the resampling step at time t.

[1] We refer to [61, 170] for more details of the particle filter algorithm.

Algorithm 3.3 Auxiliary particle filter (APF)

1: **Initialization** ($t = 1$)**:**
2: Sample $x_1^{(i)} \sim p_{\text{prop}}(x_1|y_1)$.
3: Compute the importance weights $\tilde{w}_1^{(i)} = g_\theta(y_1|x_1^{(i)})\mu_\theta(x_1^{(i)})/p_{\text{prop}}(x_1^{(i)}|y_1)$, and normalize $w_1^{(i)} = \tilde{w}_1^{(i)} / \sum_{j=1}^N \tilde{w}_1^{(j)}$.
4: **for** $t = 2$ to T **do**
5: Compute the adjustment multipliers $\nu_{t-1}^{(i)} = \nu(x_{t-1}^{(i)}, y_t)$.
6: **Resampling:** Resample $\{x_{t-1}^{(i)}\}_{i=1}^N$ with probabilities proportional to $\{w_{t-1}^{(i)}\nu_{t-1}^{(i)}\}_{i=1}^N$ to generate the equally weighted particle system $\{\overline{x}_{t-1}^{(i)}, 1/N\}_{i=1}^N$.
7: **Propagation:** Sample $x_t^{(i)} \sim p_{\text{prop}}(x_t|\overline{x}_{t-1}^{(i)}, y_t)$.
8: **Weighting:** Compute $\tilde{w}_t^{(i)} = w(\overline{x}_{t-1}^{(i)}, x_t^{(i)}, y_t)$ and normalize:

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^N \tilde{w}_t^{(j)}} \,.$$

9: **end for**

The adjustment multipliers are then used to define a proposal distribution for the ancestor index a_t according to

$$\mathbf{P}\left\{a_t = i \,\middle|\, \left\{x_{t-1}^{(j)}, w_{t-1}^{(j)}\right\}_{j=1}^N\right\} = \frac{w_{t-1}^{(i)}\nu_{t-1}^{(i)}}{\sum_{l=1}^N w_{t-1}^{(l)}\nu_{t-1}^{(l)}} \,. \tag{3.31}$$

Once the ancestor indices are generated, we propagate the particles to time t by simulating $x_t^{(i)} \sim p_{\text{prop}}(x_t|\overline{x}_{t-1}^{(i)}, y_t)$ for $i \in \{1, \ldots, N\}$, where $p_{\text{prop}}(x_t|x_{t-1}, y_t)$ is a prespecified proposal distribution, and $\overline{x}_{t-1}^{(i)} = x_{t-1}^{(a_t^{(i)})}$.

In addition to the elements introduced in this part, the APF needs the definition of a weight function from which $\{w_t^{(i)}\}_{i=1}^N$ are generated. For the APF, the weight function $w : \mathcal{X}^2 \times \mathcal{Y} \to \mathbb{R}$ is defined as[2]

$$w(x_{t-1}, x_t, y_t) = \frac{g_\theta(y_t|x_t)\, f_\theta(x_t|x_{t-1})}{\nu(x_{t-1}, y_t)\, p_{\text{prop}}(x_t|x_{t-1}, y_t)} \,. \tag{3.32}$$

Algorithm 3.3 summarizes the auxiliary particle filter method. We notice that, at time t, the APF performs three steps: (i) resampling of the particles at time $t-1$, (ii) propagation of the particles from time $t-1$ to time t, and (iii) normalized weighting of the set of particles at time t. These steps constitute the common framework of particle filtering methods [170].

[2]We note that the PF is obtained from the APF when $\nu(x_{t-1}, y_t) = 1$ [170].

Particle degeneracy

In principle, the SMC can be used to obtain an approximation of a sequence of densities $\{p(x_{1:t}|y_{1:t})\}_{t\geq 1}$ of growing dimension, where the approximations tend to the true distributions as $N \to \infty$. However, the task of approximating $\{p(x_{1:t}|y_{1:t})\}_{t\geq 1}$ using a finite number of particles N is inherently impossible. To illustrate this, assume that we use a particle filter to target the joint smooth density $p(x_{1:t}|y_{1:t})$. At time s we generate N unique[3] particles $\{x_s^{(i)}\}_{i=1}^N$ from the proposal density, and we append them to the existing particle trajectories $\{x_{1:s-1}^{(i)}\}_{i=1}^N$. Therefore, we have a weighted particle system $\{x_{1:s}^{(i)}, w_s^{(i)}\}_{i=1}^N$ approximating the joint smoothing density at time s. If we assume that the particle trajectories are resampled, then we obtain the particle system $\{\tilde{x}_{1:s}^{(i)}, 1/N\}_{i=1}^N$. Recall that the purpose of resampling is to remove particles with small weights, and to multiply particles with large weights. Thus, the resampling step has the effect of reducing the number of unique particles. In consequence, over time each consecutive resampling of the particle trajectories will reduce the number of unique particles at time s. Eventually the particle system at time s will collapse into a single trajectory. This problem is referred to as *particle degeneracy*. In other words, the resampling step inevitably results in that for any time s there exists a $t > s$ such that the PF approximation $\hat{p}_t(x_{1:t}|y_{1:t})$ consists of a single particle at time s.

A relevant question is to consider if it is possible (at least partly) to "undo" the particle degeneracy. One idea is to initiate a backwards sweep starting from the particle filter representation of $p(x_t|y_{1:t})$, and to reintroduce diversity among the particles by some form of backwards computation. This question[4] is addressed by a family of algorithms referred to as *particle smoothers*, which are introduced in the next subsection.

3.3 Particle smoothing

As previously mentioned, if we use the particle filter to compute the sequence of densities $\{p(x_{1:t}|y_{1:t})\}_{t\geq 1}$, then the approximations $\{\hat{p}(x_{1:t}|y_{1:t})\}_{t\geq 1}$ will have the particle degeneracy problem. To overcome this issue, we employ particle smoothers to compute the joint distribution. In this subsection, we introduce the forward filtering-backwards simulator (FFBSi) algorithm.

The forward filtering-backwards simulator

The FFBSi algorithm provides a solution to the problem of obtaining samples $\{x_{1:T}^{(i)}\}_{i=1}^N$ distributed according to the smoothing pdf $p(x_{1:T}|y_{1:T})$ [170]. To under-

[3] By unique we mean that each particle $x_s^{(i)}$, for $i \in \{1, \ldots, N\}$, is a different realization from the proposal density.

[4] Other solutions to the particle degeneracy problem involve increasing the number of particles N, or to use the fully-adapted PF [170].

stand the core idea of the method, we note that, by repeated use of conditional probabilities, the smoothing distribution $p(x_{1:T}|y_{1:T})$ can be written as

$$p(x_{1:T}|y_{1:T}) = p(x_T|y_{1:T}) \prod_{t=1}^{T-1} p(x_t|x_{t+1:T}, y_{1:T}) \,. \tag{3.33}$$

The expression (3.33) can be further simplified. By the Markov property of the NSSM (1.5) on page 4, we have that $p(x_t|x_{t+1:T}, y_{1:T}) = p(x_t|x_{t+1}, y_{1:T})$. In addition, since the measurements are conditionally independent given the state, we have $p(x_t|x_{t+1}, y_{1:T}) = p(x_t|x_{t+1}, y_{1:t})$. Hence, Equation (3.33) can be rewritten as

$$p(x_{1:T}|y_{1:T}) = p(x_T|y_{1:T}) \prod_{t=1}^{T-1} p(x_t|x_{t+1}, y_{1:t}) \,. \tag{3.34}$$

The pdf $p(x_t|x_{t+1}, y_{1:t})$ is often referred to as the *backward kernel*, since it works backwards in time. The backward kernel can be computed by noting that

$$p(x_t|x_{t+1}, y_{1:t}) = \frac{p(x_{t:t+1}|y_{1:t})}{p(x_{t+1}|y_{1:t})} = \frac{f_\theta(x_{t+1}|x_t)\, p(x_t|y_{1:t})}{\int_{\mathcal{X}} f_\theta(x_{t+1}|x_t)\, p(x_t|y_{1:t}) dx_t} \,. \tag{3.35}$$

Replacing (3.35) into (3.34) gives

$$p(x_{1:T}|y_{1:T}) = p(x_T|y_{1:T}) \prod_{t=1}^{T-1} \frac{p(x_{t:t+1}|y_{1:t})}{p(x_{t+1}|y_{1:t})} \,. \tag{3.36}$$

An alternative representation of the smoothing distribution $p(x_{1:T}|y_{1:T})$ is based on the backward kernel to write

$$p(x_{t:T}|y_{1:T}) = p(x_t|x_{t+1}, y_{1:t})\, p(x_{t+1:T}|y_{1:T}) \,, \tag{3.37}$$

for all $t \in \{1, \ldots, T\}$, where we start with the filtering density at time T, $p(x_T|y_{1:T})$.

In this context, the FFBSi method first computes the samples $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ for all $t \in \{1, \ldots, T\}$, approximating the filtering distributions $\{p(x_t|y_{1:t})\}_{t=1}^T$. Then, the method simulates backwards in time, i.e., by first simulating x_T, then x_{T-1}, and so on, until a complete state trajectory $x_{1:T}$ is generated. This procedure allows to compute samples $\{x_{1:T}^{(i)}\}_{i=1}^N$ distributed according to $p(x_{1:T}|y_{1:T})$. Specifically, the backward simulation begins by generating a sample from the filtering density at time T,

$$\tilde{x}_T \sim p(x_T|y_{1:T}) \,. \tag{3.38}$$

Then, the backward trajectory $x_{t:T}$ is successively generated by drawing samples from the backward kernel

$$\tilde{x}_t \sim p(x_t|\tilde{x}_{t+1}, y_{1:t}) \,, \tag{3.39}$$

for t starting at $T-1$ and ending at 1. We note that the computational complexity of the FFBSi algorithm is of order $\mathcal{O}(NMT)$, where N and M denote the number of filter and smoother particles, respectively.

Remark 3.2 *Another useful particle smoother is the fixed-lag (FL) smoother [108, 138], which has been employed in [186] for solving the nominal input design problem. The FL smoother provides a less computational expensive approach to approximate smoothing distributions than the FFBSi method. However, the reduction of computational complexity in the FL smoother is achieved by adding bias to the estimates. It can be shown that, under some strict mixing assumptions, the variance and the bias of the FL smoother are of order* $\mathcal{O}(T\log\{T\}/\sqrt{N})$ *and* $\mathcal{O}(T\log\{T\}/N)$ *respectively. We refer to [138] for more details.*

3.4 Conclusion

This chapter presented an overview of sequential Monte Carlo methods. In particular, we introduced the particle filtering and particle smoothing techniques, which are required for the implementation of the input design methods in Chapters 4 and 6.

Chapter 4

Robust input design for nonlinear models

The results in Chapter 2 provide the foundation for developing an input design methodology for dynamical models. Here we discuss how the graph-theoretical technique can be employed to formulate a robust input design framework for the identification of nonlinear state space models. The material in this chapter is based on [188].

4.1 Introduction

Input design is concerned with generating an excitation signal that maximizes the information retrieved from an experiment, quantified in terms of a cost function related to the intended model application. Some of the initial contributions are discussed in [45] and [87]. Since then, many contributions to the subject have been presented; see e.g. [66, 81, 95, 206] and the references therein.

As discussed in Section 1.3 on page 12, there has been an interest to extend the input design methods to nonlinear (NL) model structures. The main issue here is that the convex formulations in [102, 117] cannot be applied. The first approaches to the problem considered NL FIR models [99, 115], which have been extended in [194, 195] to structured NL models.

The class of NL model structures is also generalized in [69], where the input signal is optimized over an alphabet with finite cardinality. The multilevel excitation design is also considered in [53] for the identification of Wiener models. The restriction to a finite alphabet is relaxed in [88], where an AR process is designed as input for the identification of nonlinear state-space models (NSSMs). A graph theoretical methodology to design inputs for identification of NL output-error models has been developed by the author of this thesis in a collaborative work [189, 190], and it has been extended to NSSMs in [187].

The existing results on input design allow to optimize input signals when the system contains nonlinear functions, but the restrictions on the system dynamics and/or the input structure are the main limitations of most of the previous contributions. Moreover, with the exception of multilevel excitation [69, 115], and stationary processes [25, 187, 189, 190], most of the proposed methods cannot handle amplitude limitations on the input signal, which could arise due to physical and/or safety reasons.

An additional limitation of the input design methods previously mentioned is the assumption of having a prior estimate of the model parameters. The requirement of such knowledge is a common issue in input design and different solutions to this difficulty have been proposed, categorized into adaptive schemes [80, 165] and robust formulations [164, 167, 204].

Contribution

We present a robust input design method for the identification of NSSMs with input constraints, which extends the class of models considered in [189, 190], and the nominal input design presented in [187]. The optimal input signal is considered to be a realization of a stationary Markov process, which maximizes a scalar function of the Fisher information matrix (FIM). To pose a tractable convex problem, we restrict the optimization to a set of marginal distributions of stationary processes with a finite alphabet. This set is a polytope and hence it can be described by a convex combination of its vertices. The vertices are cumulative distribution functions that can be found using de Bruijn graphs (cf. Chapter 2). Once the vertices of the set are found, we can draw an input realization and compute an estimate of the FIM for each vertex using particle methods [55, 61]. The estimates of the information matrices are computed using the method introduced in [173], which only needs one realization of the input-output data, thus reducing the computational effort when estimating the FIM compared to [186, Chapter 4].

To make the input design robust against model uncertainty, the optimization problem considers a measure of the uncertainty of the parameters, which relaxes the requirements on the knowledge of the system assumed in [187, 189, 190]. The method is illustrated by numerical examples, where the designed input is employed to identify a NSSM using the expectation-maximization (EM) algorithm (outlined in Appendix B).

Structure of the chapter

This chapter is organized as follows. Section 4.2 states the problem and the main challenges when designing inputs for the identification of NSSM. Section 4.3 describes the graph theoretical approach to input design. Section 4.4 discusses the estimation of the FIM using particle methods. A summary of the proposed robust input design method is presented in Section 4.5. The generation of the optimal input signal is addressed in Section 4.6. A convergence analysis of the proposed technique is

presented in Section 4.7. To illustrate the correctness and utility of the method, two numerical examples are discussed in Section 4.8. Concluding remarks are presented in Section 4.9.

4.2 Problem formulation

Consider an NSSM described for all $t \geq 1$ by

$$x_t|x_{t-1} \sim f_\theta(x_t|x_{t-1}, u_{t-1}), \tag{4.1a}$$
$$y_t|x_t \sim g_\theta(y_t|x_t, u_t), \tag{4.1b}$$
$$x_0 \sim \mu_\theta(x_0), \tag{4.1c}$$

where f_θ, g_θ, and μ_θ denote probability density functions (pdf) parameterized by $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ (where Θ is an open set). Here, $u_t \in \mathbb{R}^{n_u}$ denotes the input signal, $x_t \in \mathbb{R}^{n_x}$ are the (unobserved/latent) internal states, and $y_t \in \mathbb{R}^{n_y}$ are the measured outputs.

The objective is to design an input signal $u_{1:T} = (u_1, \ldots, u_T)$, as a realization of a stationary process, such that the parameters characterizing the NSSM (4.1) can be identified with maximum accuracy as defined by a scalar function of the FIM [122]. In the sequel, we will assume that there exists at least one parameter $\theta_0 \in \Theta$ such that the model (4.1) exactly describes the pdfs of the system, i.e., there is no undermodelling [122].

Given $u_{1:T}$, the FIM is

$$\mathcal{I}_F^T(u_{1:T}, \theta_0) = \mathbf{E}\left\{\mathcal{S}_T(u_{1:T}, \theta_0)\mathcal{S}_T^\top(u_{1:T}, \theta_0)|\, u_{1:T}\right\}, \tag{4.2}$$

where $\mathcal{S}_T(u_{1:T}, \theta_0)$ denotes the score function, i.e.,

$$\mathcal{S}_T(u_{1:T}, \theta_0) = \nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta_0}. \tag{4.3}$$

We note that the expected value in (4.2) is with respect to the stochastic processes in (4.1). In the following, we will consider the *per-sample* FIM:

$$\begin{aligned}\mathcal{I}_F^s(P_u, \theta_0) &= \frac{1}{T}\mathbf{E}_u\left\{\mathcal{I}_F^T(u_{1:T}, \theta_0)\right\} \\ &= \frac{1}{T}\mathbf{E}\left\{\mathcal{S}_T(u_{1:T}, \theta_0)\mathcal{S}_T^\top(u_{1:T}, \theta_0)\right\},\end{aligned} \tag{4.4}$$

where the expected value in (4.4) is over both the stochastic processes in (4.1), and the random vector $u_{1:T}$.

We note that (4.4) depends on the cumulative distribution function (cdf) of $u_{1:T}$, denoted by $P_u(u_{1:T})$. Therefore, the input design problem is to find a cdf $P_u^{\text{opt}}(u_{1:T})$ which maximizes a scalar function of (4.4), $h\colon \mathbb{R}^{n_\theta \times n_\theta} \times \Theta \to \mathbb{R}$, where h is a matrix convex function in its first argument (cf. Definition 1.2).

Remark 4.1 *We let h depend on Θ as the function can explicitly depend on the model parameter.*

To simplify our problem, we assume that u_t can only adopt a finite number $n_{\mathcal{C}}$ of values. We denote this set of values as $\mathcal{C}$. With the previous assumption, we can define the following set:

$$\mathcal{P}_{\mathcal{C}} := \left\{ p_u : \mathcal{C}^T \to \mathbb{R} \middle| \begin{array}{l} p_u(\mathbf{x}) \geq 0,\, \forall \mathbf{x} \in \mathcal{C}^T; \\ \displaystyle\sum_{\mathbf{x} \in \mathcal{C}^T} p_u(\mathbf{x}) = 1; \\ \displaystyle\sum_{v \in \mathcal{C}} p_u(v,\, \mathbf{z}) = \sum_{v \in \mathcal{C}} p_u(\mathbf{z},\, v)\,, \forall \mathbf{z} \in \mathcal{C}^{T-1} \end{array} \right\}. \quad (4.5)$$

The set $\mathcal{P}_{\mathcal{C}}$ introduced in (4.5) constrains the probability mass function (pmf) $u_{1:T}$ of p_u to the set of T-dimensional marginal pmfs of stationary processes (cf. Chapter 2).

So far we have been concerned with the formulation of the problem in terms of the stationary process describing $u_{1:T}$. However, we still need to consider a remaining issue: Equation (4.4) depends on the parameter θ_0 describing the true system (4.1). Therefore, the optimal input sequence $u_{1:T}$ depends on the parameter we want to estimate, which limits the practical applicability of the previous formulation. To overcome this issue, we consider a function $\mathcal{R} : \Theta \to \mathbb{R}$ that measures the uncertainty over Θ. There are several options for defining $\mathcal{R}$. One possibility is to consider $\mathcal{R}\{h(\mathcal{I}_F^s(p_u, \theta),\, \theta)\} = h(\mathcal{I}_F^s(p_u, \theta_0),\, \theta_0)$, where $\theta_0 \in \Theta$ is a nominal parameter (known as nominal input design [187, 189, 190]). Another option is to consider the maximum value of h in Θ by setting $\mathcal{R} = \max_{\theta \in \Theta}\{\cdot\}$ (usually referred to as robust input design [167]). An additional example for $\mathcal{R}$ follows by adopting a Bayesian framework, where we assume that Θ is a probability space endowed with a known cdf[1] P_Θ. Under this setting, we can choose to optimize over the expected value of the cost function, i.e. $\mathcal{R} = \mathbf{E}_\Theta\{\cdot\}$. This choice makes sense in the context of input design, as it requires that the function h is maximized in average by the designed experiment. In the following, we will consider $\mathcal{R} = \mathbf{E}_\Theta\{\cdot\}$, and $\mathcal{R} = \max_{\theta \in \Theta}\{\cdot\}$.

To summarize, the problem we are interested in solving can be written as

Problem 4.1 *Design an optimal input signal $u_{1:T} \in \mathcal{C}^T$ as a realization from $p_u^{\mathrm{opt}}(u_{1:T})$, where*

$$p_u^{\mathrm{opt}} := \arg \min_{p_u \in \mathcal{P}_{\mathcal{C}}} \mathcal{R}\{h(\mathcal{I}_F^s(p_u, \theta),\, \theta)\}\,, \quad (4.6)$$

with $h\colon \mathbb{R}^{n_\theta \times n_\theta} \times \Theta \to \mathbb{R}$ a matrix convex function, $\mathcal{R} : \Theta \to \mathbb{R}$, and $\mathcal{I}_F^s(p_u, \theta) \in \mathbb{R}^{n_\theta \times n_\theta}$ defined as in (4.4).

[1] How to obtain the cdf P_Θ is beyond the scope of this chapter. However, we mention that P_Θ can be obtained from information about the system prior to perform the experiment.

Problem 4.1 is difficult to solve. The main challenge is that the set Θ may be uncountable, which implies that the computation of $\mathcal{R}\{h(\mathcal{I}_F^s(p_u, \theta), \theta)\}$ can be intractable. To address this issue, we consider a procedure that depends on $\mathcal{R}$. For $\mathcal{R} = \mathbf{E}_\Theta\{\cdot\}$, we solve a Monte Carlo approximation of Problem 4.1 by sampling N_s points from the set Θ according to the cdf P_Θ, and replace the expected value by its sample mean estimate. In the case where $\mathcal{R} = \max_{\theta\in\Theta}\{\cdot\}$, we employ the scenario approach [27, 204]. By sampling N_s points from the set Θ according to a given cdf P_s, we can rewrite Problem 4.1 as an optimization problem over a finite number of points in Θ. In this case we will obtain a sub-optimal solution to Problem 4.1 which, however, can be made close to the optimal solution by increasing N_s; we refer to Appendix C for more details.

In addition to the aforementioned issue, the parameterization of the set $\mathcal{P}_\mathcal{C}$ and the computation of the FIM (4.4) for the model (4.1) are also part of the complexity of solving Problem 4.1.

To parameterize the set $\mathcal{P}_\mathcal{C}$, we follow the graph theoretical approach proposed in Chapter 2. Finally, the computation of the FIM (4.4) will rely on particle methods, which are considered in Section 4.4.

Remark 4.2 *The assumptions of prior knowledge about the structure of the pdfs characterizing* (4.1) *and the finite cardinality of $\mathcal{C}$ might seem very restrictive. However, the requirement of a structure for the pdfs in* (4.1) *is not more restrictive than the requirement of a model structure for linear models, cf. [122]. On the other hand, the finite alphabet assumption for u_t is introduced to make Problem 4.1 tractable.*

Remark 4.3 *An alternative to solving Problem 4.1 is to directly optimize over* (4.2) *by designing every sample in $u_{1:T}$. However, the resulting optimization is non-convex, and hence the designed input can in general only be guaranteed to be locally optimal.*

4.3 Describing the set of stationary processes

To characterize the set $\mathcal{P}_\mathcal{C}$, we use the graph theoretical approach presented in Chapter 2. Here we briefly discuss this technique in connection with the current problem, and we refer to Chapter 2 for more details about this method.

One of the difficulties associated with $\mathcal{P}_\mathcal{C}$ is that $p_u \in \mathcal{P}_\mathcal{C}$ is of dimension T, where T can be very large. To address this issue, we restrict $u_{1:T}$ to be realization of a stationary Markov process of memory n_m, where $n_m << T$ (cf. Assumption 2.1 on page 36). Furthermore, the stationary distribution of the Markov process is constrained to the set $\mathcal{P}_\mathcal{C}$, where the pmfs are defined over $\mathcal{C}^{n_m}$. This assumption allows to solve an approximation of Problem 4.1 in the sense that the difference between the optimal cost for the solution considering $p_u(u_{1:T})$, and the optimal cost for $p_u(u_{1:n_m})$ can be made arbitrarily small by defining n_m sufficiently large.

Once the optimal pmf is obtained, the input $u_{1:T}$ is computed as the output of a Markov chain having the optimal pmf as its stationary distribution, as discussed in Section 2.3 on page 43. This aspect will be briefly revisited in Section 4.6.

A second difficulty associated with $\mathcal{P}_{\mathcal{C}}$ corresponds to its parameterization. To solve this issue, we employ the parameterization of $\mathcal{P}_{\mathcal{C}}$ presented in Section 2.2 on page 37, and which is briefly presented here.

First, we note that $\mathcal{P}_{\mathcal{C}}$ can be represented as a convex combination of its extreme points, as $\mathcal{P}_{\mathcal{C}}$ is described by a finite number of linear inequalities [163, Chapter 17]. In particular, $\mathcal{P}_{\mathcal{C}}$ is a polyhedron [163, pp. 170]. Following the notation in Chapter 2, we refer to $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}} = \{v_i\}_{i=1}^{n_{\mathcal{V}}}$ as the set of the extreme points of $\mathcal{P}_{\mathcal{C}}$. As the set $\mathcal{P}_{\mathcal{C}}$ contains pmfs, $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$ consists of pmfs describing every $p_u \in \mathcal{P}_{\mathcal{C}}$ as

$$p_u = \sum_{j=1}^{n_{\mathcal{V}}} \alpha_j v_j \,, \tag{4.7}$$

where $\alpha_j \geq 0$, for all $j \in \{1, \ldots, n_{\mathcal{V}}\}$, and

$$\sum_{j=1}^{n_{\mathcal{V}}} \alpha_j = 1 \,. \tag{4.8}$$

The problem of parameterizing $\mathcal{P}_{\mathcal{C}}$ becomes that of finding $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$. To this end, we make use of the results in Section 2.2.

Following the discussion in Section 2.2, we compute the pmfs defining the vertices of the set $\mathcal{P}_{\mathcal{C}}$ by finding the pmfs associated with the prime cycles in the equivalent n_m-dimensional de Bruijn graph with symbols in $\mathcal{C}$ (cf. Definition 2.6 on page 32 and Theorem 2.1 on page 39).

As has been shown in Lemma 2.3 on page 40, all the prime cycles associated with $\mathcal{G}_{\mathcal{C}^{n_m}}$ can be derived from the elementary cycles associated with $\mathcal{G}_{\mathcal{C}^{n_m-1}}$, which can be found using existing algorithms (cf. Remark 2.3 on page 40).

Based on the prime cycles, it is possible to generate an input sequence $u_{1:T}^{(j)}$ from v_j, which will be referred to as the *basis inputs*. We refer to Example 2.4 on page 41 for an illustration of this procedure.

Given $u_{1:T}^{(j)}$, we can use it to compute the corresponding information matrix for every $v_j \in \mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$, given by $\mathcal{I}_F^s(p_u^{(j)}, \theta)$. However, in general the matrix $\mathcal{I}_F^s(p_u^{(j)}, \theta)$ cannot be computed explicitly. This difficulty is overcome by using particle methods to approximate $\mathcal{I}_F^s(p_u^{(j)}, \theta)$, as discussed in the next section.

Remark 4.4 *To simplify the discussion, here we use T input samples to estimate $\mathcal{I}_F^s(p_u^{(j)}, \theta)$, where T is the length of the experiment specified in Problem 4.1. However, the number of samples to compute $\mathcal{I}_F^s(p_u^{(j)}, \theta)$ can be different from T in general.*

4.4 Estimation of the FIM

From (4.7), we have that every $p_u \in \mathcal{P}_\mathcal{C}$ is a convex combination of the elements in $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$. Hence, it is possible to approximate the FIM associated with p_u as

$$\mathcal{I}_F^{\text{app}}(\gamma, \theta) = \sum_{j=1}^{n_\mathcal{V}} \alpha_j \, \mathcal{I}_F^s(p_u^{(j)}, \theta)\,, \tag{4.9}$$

where $\gamma := \{\alpha_j\}_{j=1}^{n_\mathcal{V}}$. In Section 4.7 we further analyze how well $\mathcal{I}_F^{\text{app}}(\gamma, \theta)$ approximates $\mathcal{I}_F^s(p_u, \theta)$, when p_u is defined as in (4.7) and the NSSM (4.1) is restricted to the nonlinear output-error (NOE) model structure.

The main problem with the expression (4.9) is that the FIM $\mathcal{I}_F^s(p_u^{(j)}, \theta)$ is often not available in closed form for NSSM. Instead, we propose to approximate $\mathcal{I}_F^s(p_u^{(j)}, \theta)$ for every $j \in \{1, \ldots, n_\mathcal{V}\}$ using particle methods [61, 120]. For brevity, we omit in this section the superscript j corresponding to each vertex in $\mathcal{P}_\mathcal{C}$.

Estimating the score function

From (4.2), we know that we can compute the FIM by the use of the score function. Now, the score function can be estimated using particle smoothers. The key ingredient for this is the *Fisher identity* [30] presented in Appendix B, Theorem B.1. In the Fisher identity, $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$ denotes the complete data log-likelihood for (4.1) given by

$$\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T}) = \log \mu_\theta(x_0) + \sum_{t=1}^{T} \xi_\theta(x_{t-1:t}), \tag{4.10}$$

$$\xi_\theta(x_{t-1:t}) := \log f_\theta(x_t|x_{t-1}, u_{t-1}) + \log g_\theta(y_t|x_t, u_t). \tag{4.11}$$

Using (4.10) and (B.9) on page 194, we arrive at the estimator

$$\mathcal{S}_T(u_{1:T}, \theta) = \sum_{t=1}^{T} \mathcal{S}_{t,T}(u_{1:T}, \theta)\,, \tag{4.12}$$

where

$$\mathcal{S}_{t,T}(u_{1:T}, \theta) := \int_{\mathcal{X}^2} \nabla_\theta \, \xi_\theta(x_{t-1:t}) p_\theta(x_{t-1:t}|y_{1:T}, u_{1:T}) dx_{t-1:t}\,. \tag{4.13}$$

Here, we require an estimate of the two-step smoothing distribution[2] $p_\theta(x_{t-1:t}|y_{1:T}, u_{1:T})$. This can be provided by a so-called empirical distribution,

$$\widehat{p}_\theta(x_{t-1:t}|y_{1:T}, u_{1:T}) := \sum_{i=1}^{N} w_t^{(i)} \delta(x_{t-1:t} - x_{t-1:t}^{(i)}), \tag{4.14}$$

[2]We refer to Chapter 3 for a review on sequential Monte Carlo methods.

Algorithm 4.1 Bootstrap particle filter (bPF)

INPUTS: An SSM (4.1), $y_{1:T}$ (observations), $u_{1:T}$ (inputs), $N \in \mathbb{N}$ (number of particles).
OUTPUT: $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$, $t = 1, \ldots, T$.
All operations are carried out over $i, j = 1, \ldots, N$.

1: Sample $x_0^{(i)} \sim \mu_\theta(x_0)$ and set $w_0^{(i)} = 1/N$.
2: **for** $t = 1$ to T **do**
3: (Resampling) Sample $a_t^{(i)} \sim \mathsf{Cat}\big(\{w_{t-1}^{(j)}\}_{j=1}^N\big)$
4: (Propagation) Sample $x_t^{(i)} \sim f_\theta\Big(x_t^{(i)}\Big|x_{t-1}^{(a_t^{(i)})}, u_t\Big)$.
5: Set $x_{0:t}^{(i)} = \Big\{x_{0:t-1}^{a_t^{(i)}}, x_t^{(i)}\Big\}$.
6: (Weighting) Calculate $\widetilde{w}_t^{(i)} = g_\theta\Big(y_t\Big|x_t^{(i)}, u_t\Big)$.
7: Normalize $\widetilde{w}_t^{(i)}$ (over i) to obtain $w_t^{(i)}$.
8: **end for**

where $x_t^{(i)}$ and $w_t^{(i)}$ denote particle i and its normalized weight at time t. Here, $\{\{x_t^{(i)}, w_t^{(i)}\}_{t=1}^T\}_{i=1}^N$ denotes the particle system generated by a particle filter.

To estimate the FIM, we generate the particle system using the bootstrap particle filter (bPF) [60, Section 1.3.3] presented in Algorithm 4.1. Here, $\mathsf{Cat}(\{p^{(i)}\}_{i=1}^N)$ denotes the categorical distribution with $p^{(i)}$ denoting the probability of selecting element i. The limitation of implementing the estimator (4.14) based on the bPF is the poor accuracy of the latter. This is due to problems with particle degeneracy, as discussed in Chapter 3, on page 58. To mitigate this problem, we make use of a particle smoother that introduces a backward sweep after the forward run of the particle filter.

To address the particle degeneracy problem, we use the forward-filtering backwards simulator (FFBSi) with rejection sampling and early stopping [59, 183] presented in Algorithm 4.2. FFBSi makes use of the output from a run of the bPF. The parameter ρ (line 13 in Algorithm 4.2) is an upper bound for the pdf f_θ in the sense that $f_\theta(x_t|x_{t-1}, u_{t-1}) \leq \rho$ for all $t \in \{1, \ldots, T\}$. The computational complexity of FFBSi is of order $\mathcal{O}(NMT)$, where N and M denote the number of filter and smoother particles, respectively.

The statistical properties of the FFBSi are studied in [59] under some regularity assumptions. These include that the pdfs derived from the SSM (4.1) are bounded and that $\xi_\theta(x_{t-1:t})$ is measurable. These two conditions are usually fulfilled when the SSM is defined by densities f_θ and g_θ as in (4.1). Given these assumptions, it is possible to show that the error in the estimates obtained from the algorithm is bounded [59, Corollary 6] and obeys a central limit theorem [59, Corollary 9]. This implies that the estimates of the score function are (strongly) consistent and their variance decreases as $N, M \to \infty$. In practice, increasing N and M also increases the computational cost. We return to investigate the finite-data properties of FFBSi in Section 4.8.

Algorithm 4.2 Fast forward-filtering backward-simulator with early stopping (fFFBSi-ES)

INPUTS: Inputs to Algorithm 4.1, $M \in \mathbb{N}$ (No. backward trajectories), $N_{\text{limit}} \in \mathbb{N}$ (Limit for when to stop using rejection sampling), $\rho > 0$.
OUTPUT: $\widehat{\mathcal{I}}_F^s(u_{1:T}, \theta)$ (estimate of the FIM).

Run Algorithm 4.1 to obtain the particle system $\left\{x_t^{(i)}, w_t^{(i)}\right\}_{i=1}^N$ for $t = 1, \ldots, T$.
Sample $\left\{b_T(j)\right\}_{j=1}^M \sim \mathsf{Cat}\left(\{w_T^{(i)}\}_{i=1}^N\right)$.
Set $\tilde{x}_T^{(j)} = x_T^{(b_T(j))}$ for $j = 1, \ldots, M$.
for $t = T-1$ to 1 **do**
  $L \leftarrow \{1, \ldots, M\}$.
  {Rejection sampling until N_{limit} trajectories remain.}
  while $|L| \geq N_{\text{limit}}$ **do**
    $n \leftarrow \mathsf{Card}(L)$.
    $\delta \leftarrow \emptyset$.
    Sample $\left\{I(k)\right\}_{k=1}^n \sim \mathsf{Cat}\left(\{w_t^{(i)}\}_{i=1}^N\right)$.
    Sample $\left\{U(k)\right\}_{k=1}^n \sim \mathsf{Uniform}([0,1])$.
    for $k = 1$ to n **do**
      if $U(k) \leq f\left(\tilde{x}_{t+1}^{L(k)} | x_t^{I(k)}, u_t\right)/\rho$ **then**
        $b_t(L(k)) \leftarrow I(k)$.
        $\delta \leftarrow \delta \cup \{L(k)\}$.
      end if
    end for
    $L \leftarrow L \setminus \delta$.
  end while
  {Use standard FFBSi for the remaining trajectories.}
  for $j \in L$ **do**
    Compute $\tilde{w}_{t|T}^{(i,j)} \propto w_t^{(i)} f\left(\tilde{x}_{t+1}^{(j)} | x_t^{(i)}, u_t\right)$ for $i = 1, \ldots, N$.
    Normalize the smoothing weights $\left\{\tilde{w}_{t|T}^{(i,j)}\right\}_{i=1}^N$.
    Draw $b_t(j) \sim \mathsf{Cat}\left(\left\{\tilde{w}_{t|T}^{(i,j)}\right\}_{i=1}^N\right)$.
  end for
  Set $\tilde{x}_{t:T}^{(j)} = \left\{x_t^{b_t(j)}, \tilde{x}_{t+1:T}^{(j)}\right\}$ for $j = 1, \ldots, M$.
  Estimate the score function at t using (4.12) by

$$\widehat{\mathcal{S}}_{t,T}(u_{1:T}, \theta) = \frac{1}{M} \sum_{j=1}^{M} \nabla_\theta \xi_\theta \left(\tilde{x}_{t:t+1}^{(j)}\right).$$

end for
Compute $\widehat{\mathcal{I}}_F^s(u_{1:T}, \theta)$ using (4.15).

Remark 4.5 *We note that there are many alternative particle smoothers that can be useful in this context, see [120] for a recent survey. In [187], an FL particle*

smoother is employed to estimate the score function. The main advantage with using FFBSi compared to FL is the consistency of the former. In contrast, the FL smoother gives biased estimates even when $N \to \infty$. However, the FL smoother has a computational complexity of $\mathcal{O}(NT)$, which is smaller than that of the FFBSi smoother.

The resulting estimator

From (4.12), we note that $\mathcal{S}_T(u_{1:T}, \theta)$ can be written as a sum of conditional scores $\mathcal{S}_{t,T}(u_{1:T}, \theta)$. This can be seen as a martingale representation since each conditional score is conditionally independent given the past. Using results presented in [133, 173], we can compute an estimate of the FIM by

$$\widehat{\mathcal{I}}_F^s(u_{1:T}, \theta) = \frac{1}{T}\left\{\sum_{t=1}^{T}\left[\widehat{\mathcal{S}}_{t,T}(u_{1:T}, \theta)\right]\left[\widehat{\mathcal{S}}_{t,T}(u_{1:T}, \theta)\right]^\top - \frac{1}{T}\left[\widehat{\mathcal{S}}_T(u_{1:T}, \theta)\right]\left[\widehat{\mathcal{S}}_T(u_{1:T}, \theta)\right]^\top\right\}, \tag{4.15}$$

where

$$\widehat{\mathcal{S}}_T(u_{1:T}, \theta) := \sum_{t=1}^{T} \widehat{\mathcal{S}}_{t,T}(u_{1:T}, \theta)\,. \tag{4.16}$$

We note that the estimator (4.15) is based on the output $y_{1:T}$ generated from (4.1) using $\theta \in \Theta$ and $u_{1:T}$ as input.

There are other alternatives for computing an estimate of the FIM. One alternative approach is to make use of the Louis identity (cf. Theorem B.2). Another alternative is to compute the FIM using the sample covariance matrix of the score estimates (4.12) as proposed in [187]. The main advantage of the estimator (4.15) is that it only requires running a single particle smoother for estimating the score function. The approach based on the sample covariance matrix requires hundreds or thousands of runs to marginalize out the effect of the noisy realization, that is, to estimate the expectation operator in (4.12) with respect to $y_{1:T}$. The use of (4.15) results in a significant speed-up of the same order in the computations as well as in substantial improvements in the accuracy of the estimates of $\mathcal{I}_F^s(p_u, \theta)$.

To see the latter, we analyze the variance of this estimator using the martingale difference property when using Algorithm 4.2 to estimate the score function. To this end, we note that the expected value of (4.15) is given by

$$T\,\mathbf{E}\left\{\widehat{\mathcal{I}}_F^s(u_{1:T}, \theta)\right\} = \frac{T-1}{T}\mathbf{E}\left\{\sum_{t=1}^{T}\left[\widehat{\mathcal{S}}_{t,T}(u_{1:T}, \theta)\right]\left[\widehat{\mathcal{S}}_{t,T}(u_{1:T}, \theta)\right]^\top\right\}. \tag{4.17}$$

where the expected value in (4.17) is with respect to the stochastic processes characterizing the NSSM (4.1).

Before proceeding, let $\widehat{\mathcal{S}}_{T,\ell}(u_{1:T},\theta)$ and $\widehat{\mathcal{I}}^s_{F,\ell m}(u_{1:T},\theta)$ denote the ℓ-th and (ℓ,m) entry of $\widehat{\mathcal{S}}_T(u_{1:T},\theta)$ and $\widehat{\mathcal{I}}^s_F(u_{1:T},\theta)$, respectively. Then the variance of the (ℓ,m) entry of (4.15) is given by

$$T^2 \operatorname{Var}\left\{\widehat{\mathcal{I}}^s_{F,\ell m}(u_{1:T},\theta)\right\} = T^2\left[\mathbf{E}\left\{\widehat{\mathcal{I}}^s_{F,\ell m}(u_{1:T},\theta)^2\right\} - \mathbf{E}^2\left\{\widehat{\mathcal{I}}^s_{F,\ell m}(u_{1:T},\theta)\right\}\right]. \tag{4.18}$$

Based on (4.15), we compute $\mathbf{E}\left\{\widehat{\mathcal{I}}^s_{F,\ell m}(u_{1:T},\theta)^2\right\}$ as

$$T^2\mathbf{E}\left\{\widehat{\mathcal{I}}^s_{F,\ell m}(u_{1:T},\theta)^2\right\} = \mathbf{E}\left\{f_1^2(T,\ell,m)\right\} + \frac{1}{T^2}\mathbf{E}\left\{f_2^2(T,\ell,m)\right\} - \frac{2}{T}\mathbf{E}\left\{f_1(T,\ell,m)\, f_2(T,\ell,m)\right\}, \tag{4.19}$$

where

$$f_1(T,\ell,m) := \sum_{t=1}^{T} \widehat{\mathcal{S}}_{t,T,\ell}(u_{1:T},\theta)\widehat{\mathcal{S}}_{t,T,m}(u_{1:T},\theta), \tag{4.20}$$

$$f_2(T,\ell,m) := \widehat{\mathcal{S}}_{T,\ell}(u_{1:T},\theta)\widehat{\mathcal{S}}_{T,m}(u_{1:T},\theta). \tag{4.21}$$

Using the martingale difference property, we obtain

$$\mathbf{E}\left\{f_1(T,\ell,m)\, f_2(T,\ell,m)\right\} = \mathbf{E}\left\{\sum_{t=1}^{T} f_3^2(t,\ell,m)\right\}, \tag{4.22}$$

with

$$f_3(t,\ell,m) := \widehat{\mathcal{S}}_{t,T,\ell}(\theta)\widehat{\mathcal{S}}_{t,T,m}(\theta). \tag{4.23}$$

Replacing (4.22) into (4.19) we have

$$T^2\mathbf{E}\left\{\widehat{\mathcal{I}}^s_{F,\ell m}(u_{1:T},\theta)^2\right\} = \left[1 - \frac{(T-1)^2}{T^2}\right]\mathbf{E}\left\{f_1^2(T,\ell,m)\right\} + \frac{1}{T^2}\mathbf{E}\left\{f_2^2(T,\ell,m)\right\} - \frac{2}{T}\mathbf{E}\left\{\sum_{t=1}^{T} f_3^2(t,\ell,m)\right\} + \frac{(T-1)^2}{T^2}\mathbf{E}\left\{f_1^2(T,\ell,m)\right\}, \tag{4.24}$$

where we have added and subtracted the term

$$\frac{(T-1)^2}{T^2}\mathbf{E}\left\{f_1^2(T,\ell,m)\right\}. \tag{4.25}$$

Finally, replacing (4.17) and (4.24) into (4.18) and rearranging terms, we obtain

$$T^2 \operatorname{Var}\left\{\widehat{\mathcal{I}}_{F,\ell m}(u_{1:T},\theta)\right\} = \frac{(T-1)^2}{T^2}\operatorname{Var}\left\{f_1(T,\ell,m)\right\} + \frac{2}{T}f_4(T,\ell,m) + \frac{1}{T^2}f_5(T,\ell,m)\,, \quad (4.26)$$

with

$$f_4(T,\ell,m) := \mathbf{E}\left\{f_1^2(T,\ell,m)\right\} - \mathbf{E}\left\{\sum_{t=1}^{T} f_3^2(t,\ell,m)\right\}, \quad (4.27)$$

$$f_5(T,\ell,m) := \mathbf{E}\left\{f_2^2(T,\ell,m)\right\} - \mathbf{E}\left\{f_1^2(T,\ell,m)\right\}. \quad (4.28)$$

When the FFBSi is used to estimate the score function, we have by [59, Theorem 11] that the variance (4.26) is finite for a fixed T. Moreover, the estimator is consistent as $T, N, M \to \infty$, which follows from [59, Corollary 9]. We will investigate the finite sample accuracy of the estimate in Section 4.8.

We can carry out a similar analysis for the alternative approach based on the sample covariance matrix. Here, we estimate the score function $\widehat{\mathcal{S}}_T^{(k)}$ using particle methods over $k \in \{1, \ldots, K\}$ different noisy realizations based on a single realization of the input. The estimate of the FIM is computed by

$$\widehat{\mathcal{I}}_F^s(u_{1:T},\theta) = \frac{1}{KT}\sum_{k=1}^{K}\left[\widehat{\mathcal{S}}_T^{(k)}(u_{1:T},\theta)\right]\left[\widehat{\mathcal{S}}_T^{(k)}(u_{1:T},\theta)\right]^\top. \quad (4.29)$$

The variance of this estimator is given by

$$T^2 \operatorname{Var}\left\{\widehat{\mathcal{I}}_{F,\ell m}^s(u_{1:T},\theta)\right\} = \frac{1}{K}\operatorname{Var}\left\{\widehat{\mathcal{S}}_\ell^{(k)}(\theta)\widehat{\mathcal{S}}_m^{(k)}(\theta)\right\}, \quad (4.30)$$

for some $k \in \{1,\ldots,K\}$. This implies that the accuracy of (4.29) increases with the number of realizations K, provided that the variance of each element in $\left[\widehat{\mathcal{S}}_T^{(k)}(u_{1:T},\theta)\right]\left[\widehat{\mathcal{S}}_T^{(k)}(u_{1:T},\theta)\right]^\top$ is bounded, which again is satisfied by FFBSi as discussed above. Therefore, we conclude that the variance of the estimator (4.29) is bounded in T, N, M and it approaches zero as $K \to \infty$.

The main benefit of using (4.15) instead of (4.29) is a smaller computational cost. This results from that only one run of the particle smoother is required for (4.15) compared with K runs for (4.29). In practice, this decreases the computational time for a single estimate of the FIM from over a day to about an hour. Moreover, it is difficult to establish theoretically which of the two estimators has better accuracy. Using numerical evaluations, we have observed that the new estimator outperforms the latter in terms of both accuracy and computational cost.

Algorithm 4.3 New input design method

INPUTS: $\mathcal{C}$ (input values), n_m (memory of the Markov process), $\mathbf{P}$ (probability measure over Θ), and N_s (number of samples over Θ).
OUTPUT: γ^{opt} (optimal weighting of the basis inputs).

1: Sample $\{\theta_i\}_{i=1}^{N_\mathrm{s}}$ from Θ according to $\mathbf{P}$.
2: Compute all the elementary cycles of $\mathcal{G}_{\mathcal{C}^{n_m-1}}$ using, e.g., [103, pp. 79–80], [184, pp. 157].
3: Compute all the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_m}}$ from the elementary cycles of $\mathcal{G}_{\mathcal{C}^{n_m-1}}$ as explained in Lemma 2.3. Denote by $\{v_j\}_{j=1}^{n_\mathcal{V}}$ the pmfs associated with the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_m}}$.
4: Generate the input signals $u_{1:T}^{(j)}$ from the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_m}}$, for each $j \in \{1, \ldots, n_\mathcal{V}\}$.
5: **for** $i = 1$ to N_s **do**
6: Execute Algorithm 4.2 based on θ_i and $\{u_{1:T}^{(j)}\}_{j=1}^{n_\mathcal{V}}$ to obtain $\{\widehat{\mathcal{I}}_F^s(u_{1:T}^{(j)}, \theta_i)\}_{j=1}^{n_\mathcal{V}}$.
7: **end for**
8: Solve the optimization problem (4.31) to obtain $\gamma^\star$.

4.5 Final input design method

Under the considered approximations, the FIM (4.4) can be expressed as a convex combination of the information matrices evaluated at the vertex points of $\mathcal{P}_\mathcal{C}$ given by (4.9). Hence, the proposed method to design input signals in $\mathcal{C}^{n_m}$ for the identification of NL-SSMs is summarized in Algorithm 4.3. The method solves an approximation of Problem 4.1, which can be written as

$$\begin{aligned}
\gamma^{\mathrm{opt}} = \arg \min_{\gamma=\{\alpha_j\}_{j=1}^{n_\mathcal{V}}} \quad & \widehat{\mathcal{R}}(h(\widehat{\mathcal{I}}_F^{\mathrm{app}}(\gamma,\, \theta),\, \theta)) \\
\text{subject to} \quad & \widehat{\mathcal{I}}_F^{\mathrm{app}}(\gamma,\, \theta_i) = \sum_{j=1}^{n_\mathcal{V}} \alpha_j\, \widehat{\mathcal{I}}_F^{s}(u_{1:T}^{(j)}, \theta_i)\,, \text{ for all } i \in \{1, \ldots, N_\mathrm{s}\} \\
& \alpha_j \geq 0\,, \text{ for all } j \in \{1, \ldots, n_\mathcal{V}\} \\
& \sum_{j=1}^{n_\mathcal{V}} \alpha_j = 1\,,
\end{aligned} \tag{4.31}$$

where $\widehat{\mathcal{R}}$ denotes the approximation of the function $\mathcal{R}$ when the set Θ is approximated by $\{\theta_i\}_{i=1}^{N_\mathrm{s}}$. The implementation of $\widehat{\mathcal{R}}$ for the cost functions considered in this chapter follows the solution presented in Section 4.2. Thus, for $\mathcal{R} = \mathbf{E}_\Theta\{\cdot\}$ we have

$$\widehat{\mathcal{R}}(h(\widehat{\mathcal{I}}_F^{\mathrm{app}}(\gamma,\, \theta),\, \theta)) = \frac{1}{N_\mathrm{s}} \sum_{i=1}^{N_\mathrm{s}} h(\widehat{\mathcal{I}}_F^{\mathrm{app}}(\gamma,\, \theta_i),\, \theta_i)\,, \tag{4.32}$$

and for $\mathcal{R} = \max_{\theta\in\Theta}\{\cdot\}$ we obtain

$$\widehat{\mathcal{R}}(h(\widehat{\mathcal{I}}_F^{\mathrm{app}}(\gamma,\, \theta),\, \theta)) = \max_{\theta\in\{\theta_i\}_{i=1}^{N_\mathrm{s}}} h(\widehat{\mathcal{I}}_F^{\mathrm{app}}(\gamma,\, \theta),\, \theta)\,. \tag{4.33}$$

Algorithm 4.3 computes the vector $\gamma^{\text{opt}} = \{\alpha_j^{\text{opt}}\}_{j=1}^{n_\mathcal{V}}$ which defines the optimal pmf p_u^{opt} as a convex combination of the measures associated with the elements in $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$, according to

$$p_u^{\text{opt}} = \sum_{j=1}^{n_\mathcal{V}} \alpha_j^{\text{opt}}\, v_j\,. \tag{4.34}$$

4.6 Input signal generation

Given a pmf $p_u(u_{1:n_m}) \in \mathcal{P}_\mathcal{C}$ describing the stationary distribution of a Markov process with memory n_m, we need to obtain a realization $u_{1:T}$ from such Markov process. To this end, we employ the method introduced in Section 2.3 on page 43. We build a transition probability matrix with stationary distribution p_u by making use of Algorithm 2.1, and then obtain the input $u_{1:T}$ from the designed Markov chain running on stationary regime.

4.7 Convergence analysis

At this point we return to what has been mentioned about the expression (4.9): Equation (4.9) is an approximation of the per-sample FIM

$$\mathcal{I}_F^s(p_u^{\text{opt}}, \theta) = \frac{1}{T}\mathbf{E}\left\{\mathcal{I}_F^T(u_{1:T}, \theta)\right\}\,, \tag{4.35}$$

where p_u^{opt} is given by (4.34).

The discussion around (4.35) is restricted in this section to the NOE model structure, defined for all $t \geq 1$ as

$$x_{t+1} = f_\theta^x(x_t, u_t)\,, \tag{4.36a}$$
$$y_t|x_t \sim \mathcal{N}(g_\theta^y(x_t, u_t), \sigma^2)\,, \tag{4.36b}$$
$$x_0 = x^\star\,, \tag{4.36c}$$

where $x^\star \in \mathbb{R}^{n_x}$, $f_\theta^x \colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, and $g_\theta^x \colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ are differentiable functions in θ, for all $\theta \in \Theta$. We further assume that the variance σ^2 in (4.36b) is known.

Based on the NOE model (4.36), we have that the per-sample FIM is given by

$$\mathcal{I}_F^s(p_u, \theta_0) = \frac{1}{T\,\sigma^2}\sum_{t=1}^{T}\mathbf{E}\left\{\nabla_\theta\left[g_\theta^y(x_t, u_t)\right]\big|_{\theta=\theta_0}\,\nabla_\theta^\top\left[g_\theta^y(x_t, u_t)\right]\big|_{\theta=\theta_0}\right\}\,. \tag{4.37}$$

We need the following

Assumption 4.1 *For every $\theta_0 \in \Theta$,*

$$\left\{\nabla_\theta\left[g_\theta^y(x_t, u_t)\right]\big|_{\theta=\theta_0}\,\nabla_\theta^\top\left[g_\theta^y(x_t, u_t)\right]\big|_{\theta=\theta_0}\right\}_{t\geq 1} \tag{4.38}$$

is a sequence of bounded (with respect to t) and exponentially forgetting functions in the sense of Definition D.1.

Based on Assumption 4.1, we can establish the convergence of the per-sample FIM when $u_{1:T}^{(j)}$ is drawn from the pmf $p_u^{(j)}$ associated with the j-th vertex of the set $\mathcal{P}_{\mathcal{C}}$:

Corollary 4.1 *Consider the nonlinear output-error model* (4.36)*. Let* $\{u_t^{(j)}\}$, $u_t^{(j)} \in \mathcal{C}$*, be a sequence of period* $\mathbf{T_u}$*, generated from the prime cycle associated with* $p_u^{(j)} \in \mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$ *($j \in \{1, \ldots, n_{\mathcal{V}}\}$) satisfying* $|u_t| \leq K$ *for some* $K \geq 0$*, and*

$$\left\{ \nabla_\theta \left[g_\theta^y(x_t, u_t)\right]\big|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(x_t, u_t)\right]\big|_{\theta=\theta_0} \right\}_{t\geq 1} \tag{4.39}$$

satisfies Assumption 4.1. Then, for every $\theta_0 \in \Theta$ *and* $j \in \{1, \ldots, n_{\mathcal{V}}\}$*,*

$$\begin{aligned}
\lim_{T\to\infty} \frac{1}{T} \mathcal{I}_F^T(u_{1:T}^{(j)}, \theta_0) &= \lim_{T\to\infty} \frac{1}{T\,\sigma^2} \sum_{t=1}^{T} \nabla_\theta \left[g_\theta^y(x_t, u_t^{(j)})\right]\Big|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(x_t, u_t^{(j)})\right]\Big|_{\theta=\theta_0} \\
&= \lim_{t\to\infty} \frac{1}{\mathbf{T_u}\,\sigma^2} \sum_{k=1}^{\mathbf{T_u}} \nabla_\theta \left[g_\theta^y(x_t, u_t^{(k,j)})\right]\Big|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(x_t, u_t^{(k,j)})\right]\Big|_{\theta=\theta_0} \\
&= \lim_{t\to\infty} \frac{1}{\sigma^2} \int \nabla_\theta \left[g_\theta^y(x_t, u_t)\right]\big|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(x_t, u_t)\right]\big|_{\theta=\theta_0} d\mathbf{P}(u_{1:t}) \,,
\end{aligned} \tag{4.40}$$

where $\mathbf{P}$ *is the probability measure of a Markov chain generating* $\{u_t\}$ *(a uniform probability distribution on the set of at most* $\mathbf{T_u}$ *possible values of* $u_{1:\mathbf{T_u}}$*), and* $u_{1:t}^{(k,j)}$ *is obtained from* $\{u_t^{(j)}\}$ *after* $k-1$ *time shift units.*

Proof *Follows immediately from Theorem D.2 by setting*

$$g_t(u_{1:t}) = \nabla_\theta \left[g_\theta^y(x_t, u_t)\right]\big|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(x_t, u_t)\right]\big|_{\theta=\theta_0} \tag{4.41}$$

for all $t \geq 1$*, and considering*

$$g_{n_m}(u_{t-n_m+1:t}) = \nabla_\theta \left[g_\theta^y(\tilde{x}_t, u_t)\right]\big|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(\tilde{x}_t, u_t)\right]\big|_{\theta=\theta_0} \tag{4.42}$$

for all $n_m \geq 1$ *and all* $t \geq n_m + 1$*, where* $\tilde{x}_t$ *is the state of the nonlinear output-error model* (4.36) *at time* t *obtained with* $\theta = \theta_0$*, initial condition* $x_0 = 0$*, and input*

$$\tilde{u}_k^{t-n_m} = \begin{cases} 0, & \text{if } k \leq t - n_m \,, \\ u_k, & \text{otherwise.} \end{cases} \tag{4.43}$$

■

Finally, we establish a bound on the difference between the optimized and the actual Fisher information matrices attained with the proposed method:

Corollary 4.2 *Consider the assumptions in Theorem D.1 and the exponentially forgetting sequence of functions*

$$\left\{ \nabla_\theta \left[g_\theta^y(x_t, u_t) \right]|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(x_t, u_t) \right]\big|_{\theta=\theta_0} \right\}_{t\geq 1} . \tag{4.44}$$

Then, for all $\theta_0 \in \Theta$,

$$\left\| \lim_{T\to\infty} \frac{1}{T\,\sigma^2} \sum_{t=1}^{T} \sum_{u_{1:t}\in\mathcal{C}^t} \nabla_\theta \left[g_\theta^y(x_t, u_t) \right]|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(x_t, u_t) \right]\big|_{\theta=\theta_0} p_t^{\mathrm{opt}}(u_{1:t}|u_{-n_m+1:0}) - \sum_{i=1}^{n_{\mathcal{V}}} \alpha_i \, \overline{\mathcal{I}}_{F,n_m}^{(i)}(\theta_0) \right\| \leq 2C\lambda^{n_m} , \tag{4.45}$$

where

$$\overline{\mathcal{I}}_{F,n_m}^{(i)}(\theta_0) := \lim_{T\to\infty} \frac{1}{T\,\sigma^2} \sum_{t=1}^{T} \sum_{u_{1:t}\in\mathcal{C}^t} f_t(u_{1:t}) p_t^{(i)}(u_{1:t}|u_{-n_m+1:0}) , \tag{4.46}$$

$$f_t(u_{1:t}) := \nabla_\theta \left[g_\theta^y(x_t, u_t) \right]|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(x_t, u_t) \right]\big|_{\theta=\theta_0} , \tag{4.47}$$

and p_t^{opt}, $p_t^{(i)}$ *given as in Theorem D.1 for all* $i \in \{1, \ldots, n_{\mathcal{V}}\}$.

Proof *This is immediate from Theorem D.1 by setting*

$$g_t(u_{1:t}) = \nabla_\theta \left[g_\theta^y(x_t, u_t) \right]|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(x_t, u_t) \right]\big|_{\theta=\theta_0} , \tag{4.48}$$

for all $t \geq 1$, $\overline{g}_{n_m}^{(i)} = \overline{\mathcal{I}}_{F,n_m}^{(i)}(\theta_0)$ *for all* $i \in \{1, \ldots, n_{\mathcal{V}}\}$, *and considering*

$$g_{n_m}(u_{t-n_m+1:t}) = \nabla_\theta \left[g_\theta^y(\tilde{x}_t, u_t) \right]|_{\theta=\theta_0} \nabla_\theta^\top \left[g_\theta^y(\tilde{x}_t, u_t) \right]\big|_{\theta=\theta_0} \tag{4.49}$$

for all $n_m \geq 1$ *and all* $t \geq n_m + 1$, *where* $\tilde{x}_t$ *is the state of the nonlinear output-error model* (4.36) *at time* t *obtained with* $\theta = \theta_0$, *initial condition* $x_0 = 0$, *and input*

$$\tilde{u}_k^{t-n_m} = \begin{cases} 0, & \text{if } k \leq t - n_m , \\ u_k, & \text{otherwise.} \end{cases} \tag{4.50}$$

■

Remark 4.6 *Corollary 4.2 states that the difference between the optimal value of the Fisher information matrix and the actual value obtained in [190] is less than a given* $\varepsilon > 0$ *if*

$$n_m \geq \frac{\log(\varepsilon) - \log(2C)}{\log(\lambda)} . \tag{4.51}$$

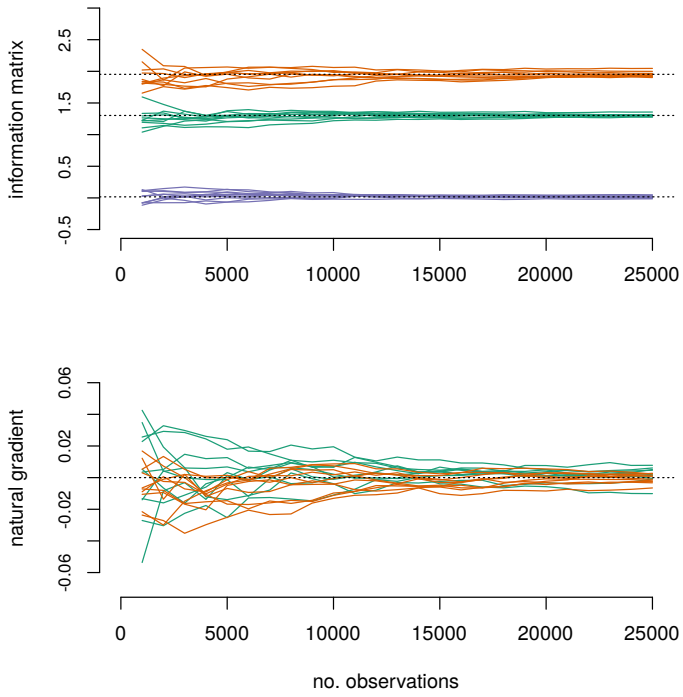

Figure 4.1: Top: the information matrix element for ϕ (orange), σ_v (green) and cross-term between ϕ and σ_v (purple). Bottom: the natural gradient of the log-likelihood function for different T. The results are computed using 25 Monte Carlo simulations.

4.8 Numerical examples

In this section, we present numerical simulations to illustrate some aspects of the proposed input design method in two SSMs. First, we make use of a LGSS model to evaluate the accuracy of the estimator for the FIM and the impact of the design parameters in Algorithm 4.2. We also compare the solution obtained for the input design problem to some standard input signals to verify how well the proposed method works.

Second, we consider an NL-SSM, which is more challenging from an input design perspective. All implementation details and settings for the algorithm are presented at the end of this section.

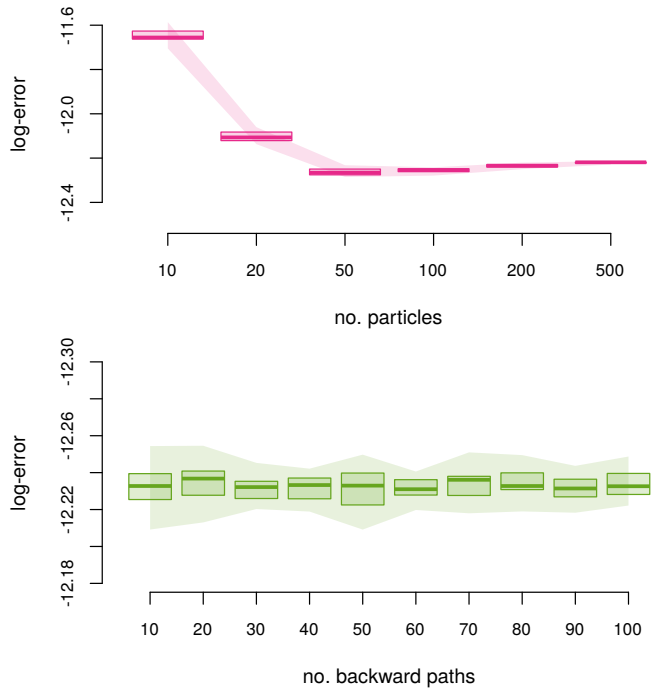

Figure 4.2: The logarithm of the Frobenius norm of the error of the estimate of the information matrix when varying N (upper) and M (lower). The plots show the results from 25 Monte Carlo simulations and the shaded areas indicate the span between the largest and smallest error.

Accuracy of information matrix estimation

Consider the LGSS model given by

$$x_t|x_{t-1} \sim \mathcal{N}\big(\phi x_{t-1} + u_{t-1}, \sigma_v^2\big), \tag{4.52a}$$

$$y_t|x_t \sim \mathcal{N}\big(x_t, 0.1^2\big), \tag{4.52b}$$

where the parameter vector is $\theta = \{\phi, \sigma_v\}$ with the state persistence $\phi \in (-1, 1)$ and the standard deviation of the innovations $\sigma_v \in \mathbb{R}^+$.

From Section 4.4, we know that the estimates of the FIM obtained by (4.15) are consistent. However, in practice we do not have an infinite amount of data and therefore the properties of the estimate can only be evaluated using numerical experiments. To this end, we make use of a Kalman method, which allows to compute the score function exactly for the LGSS model. Hence, we can isolate the influence of T on the accuracy of the estimate.

Figure 4.1 presents the estimate and the resulting natural gradient of the log-likelihood function (the gradient scaled by the inverse FIM) when varying T for 25 Monte Carlo executions. We conclude that the estimator stabilizes after about $T \approx 15 \cdot 10^3$ observations. Furthermore, we observe that the natural gradients are almost zero at this value of T, indicating that θ_0 is a parameter attaining a local maximum for the log-likelihood function.

For a general SSM, we cannot make use of Kalman methods and have to resort to approximations using, e.g., particle methods. In this setting, we want to investigate the influence of N and M in Algorithm 4.2 on the accuracy of the estimates, when T is fixed to $15 \cdot 10^3$ based on the results above. Figure 4.2 presents the logarithm of the Frobenius norm of the error in the information matrix when varying N and M. In the former case, we fix $M = \lfloor N/4 \rfloor$ (the closest integer to $N/4$ from below) and vary N. We conclude that 200 particles are enough to obtain reasonable accuracy in this model. The increase of the log-error for $N \geq 50$ is probably due to the variability of the Monte Carlo simulations. In the latter case, we fix $N = 200$ and vary the number of backward paths M. We conclude from this that the accuracy of the estimates is quite robust to the choice of the number of backward trajectories, and that $M = 10$ seems to be a reasonable choice for this example.

Input design for the linear Gaussian state space model

The analysis in the previous subsection provides some guidance for selecting T, N and M to obtain reasonable estimates of the FIM for the LGSS model. Therefore, we are ready to apply Algorithm 4.3 to construct an input sequence with the aim to accurately estimate $\theta = \{\phi, \sigma_v\}$ in (4.52). We make use of the EM algorithm proposed in [171] to estimate the parameters of the model when different inputs are applied. This is done to investigate if the designed input actually increases the accuracy of the estimates with respect to standard inputs.

Algorithm 4.3 spends most of the time in Line 6 (5 minutes for each realization i on a standard stationary computer from 2012) and Line 8 (1 minute). Note that it is possible to parallelize Line 6 to decrease the computational time.

Table 4.1 presents the logarithm of the mean squared error (MSE) of the parameter estimates for different inputs: none ($u_t \equiv 0$), constant ($u_t \equiv 1$), uniform ($u_t \sim \mathcal{U}[-1, 1]$) and binary ($u_t \sim 1 - 2 \cdot \mathsf{Bernoulli}(0.5)$). As mentioned above, Kalman methods can be applied to compute the score function (and the Q-function in the EM algorithm) exactly. Hence, Kalman methods illustrate the optimal performance of the proposed algorithm when N and M tends to infinity. Moreover, we make use of $\mathcal{R} = \mathbf{E}_\theta\{\cdot\}$ (mean case), and $\mathcal{R} = \max_{\theta \in \Theta}\{\cdot\}$ (worst case) to include robustness with respect to the uncertainty in the parameters. The input design technique with particle methods (PM) attains similar log-MSE values to those obtained with a constant signal ($u_t = 1$), which is the optimal choice.

To complement the results in this example, we compute $\log\{\det(\mathcal{I}_{\text{MSE}}(\theta_0))\}$ for the different inputs and methods, where $\mathcal{I}_{\text{MSE}}(\theta_0)$ corresponds to the sample MSE

Table 4.1: Parameter estimates with the 1.96 times the standard deviation (parenthesis) and the log-MSE in the LGSS model computed from 40 independent runs of the EM algorithm using different (but fixed) input signals. Bold face marks the best values and KM/PM indicate that Kalman methods/Algorithm 4.2 is used to compute the Q-function in the EM algorithm. The true parameter values for θ are indicated between square brackets.

	Kalman (KM)				Particle (PM)			
Input	$\widehat{\phi}$ [0.5]	$\mathsf{MSE}(\widehat{\phi})$	$\widehat{\sigma}_v$ [0.1]	$\mathsf{MSE}(\widehat{\sigma}_v)$	$\widehat{\phi}$ [0.5]	$\mathsf{MSE}(\widehat{\phi})$	$\widehat{\sigma}_v$ [0.1]	$\mathsf{MSE}(\widehat{\sigma}_v)$
none	0.50 (0.02)	-9.06	0.10 (0.002)	-13.25	0.30 (0.02)	-3.21	0.11 (0.002)	**-8.89**
constant	0.50 (0.00)	**-15.33**	0.10 (0.002)	-13.47	0.50 (0.00)	-14.15	0.12 (0.002)	-8.33
uniform	0.50 (0.00)	-13.02	0.10 (0.002)	**-13.49**	0.49 (0.00)	-10.31	0.11 (0.002)	-8.68
binary	0.50 (0.00)	-14.06	0.10 (0.002)	-13.43	0.50 (0.00)	-12.26	0.11 (0.002)	-8.52
mean case	0.50 (0.00)	-14.56	0.10 (0.002)	-13.46	0.50 (0.00)	-14.15	0.12 (0.002)	-8.33
worst case	0.50 (0.00)	-14.52	0.10 (0.002)	-13.48	0.50 (0.00)	**-14.18**	0.12 (0.001)	-8.34

Table 4.2: $\log\{\det(\mathcal{I}_{\text{MSE}}(\theta_0))\}$ for the parameter estimates of the LGSS model. Bold face marks the best value and KM/PM indicate that Kalman methods/Algorithm 4.2 is used to compute the Q-function in the EM algorithm.

Input	Kalman (KM)	Particle (PM)
none	-22.53	-16.63
constant	**-28.83**	-23.56
uniform	-26.53	-21.56
binary	-27.69	-22.67
mean case	**-28.83**	-23.56
worst case	**-28.83**	**-23.67**

matrix

$$\mathcal{I}_{\text{MSE}}(\theta_0) := \frac{1}{N_{\text{est}}} \sum_{i=1}^{N_{\text{est}}} (\widehat{\theta}_{T,i} - \theta_0)(\widehat{\theta}_{T,i} - \theta_0)^\top, \tag{4.53}$$

where $\{\widehat{\theta}_{T,i}\}_{i=1}^{N_{\text{est}}}$ are the estimated parameters for the different inputs and methods in Table 4.1 ($N_{\text{est}} = 40$). The results for $\log\{\det(\mathcal{I}_{\text{MSE}}(\theta_0))\}$ are summarized in Table 4.2. From this table we conclude that the mean and worst case designs based on the proposed method attain the minimum value of $\log\{\det(\mathcal{I}_{\text{MSE}}(\theta_0))\}$ among the inputs considered in this example.

Input design for a nonlinear model

The second example is an NL-SSM given by

$$x_t | x_{t-1} \sim \mathcal{N}\Big(\frac{1}{\gamma + x_{t-1}^2} + u_{t-1}, 0.1^2\Big), \tag{4.54a}$$

$$y_t | x_t \sim \mathcal{N}(\beta x_t^2, 1^2), \tag{4.54b}$$

where the parameters are $\theta = \{\gamma, \beta\}$. The sign of the state is lost due to the term x_t^2 in the measurement process, which implies that two different values for the state can equally well represent a value for y_t. The term x_t^2 at the output and the NL state process prohibit the use of Kalman methods for this model, hence only Algorithm 4.2 is considered. We apply the same approach as in the previous example to construct an input to excite the system and then apply an EM algorithm to estimate the parameters. The computational time for executing Line 6 in Algorithm 4.3 increases to 38 minutes in this model.

Figure 4.3 presents estimates at each iteration of the EM algorithm from 35 independent runs on the same input/output data. The parameter β seems easier to estimate than the parameter γ, probably because the model is linear in β. Another interesting result from this figure can be seen by comparing the estimates for the different inputs. We observe that there exists a trade-off on the accuracy achieved

for the estimates of γ and β. While a zero input gives more accurate estimates for γ than using a nonzero input, the accuracy for the estimates of β is increased by including a non-zero input.

We can also analyze the overall performance of the estimates for different inputs. In Table 4.3 the parameter estimates and the corresponding log-MSE for different inputs is presented. We note that the log-MSE is computed using θ_0 and not the maximum likelihood estimate of θ. We also present the half-length of the bootstrapped 95% confidence intervals (CIs) for the estimated parameters in parentheses. These are obtained after 10^3 resamplings and using the adjusted bootstrap percentile. We refer to [51] for further details on the computation of this quantity.

Table 4.3 shows that no overall best input signal can be determined from this experiment, as the decision depends on the relative importance of the two parameters. In the case of the optimal inputs designed by Algorithm 4.3, we see that we can improve the log-MSE for the estimate of γ when compared to the uniform and binary inputs. As expected from the previous discussion, the improvement in the log-MSE for $\widehat{\gamma}$ is achieved by degrading the log-MSE for $\widehat{\beta}$ when compared to the binary input. However, the log-MSE of $\widehat{\beta}$ for the designed inputs is better than the one obtained with no input.

As a final comparison, Table 4.4 presents the value of $\log\{\det(\mathcal{I}_{\text{MSE}}(\theta_0))\}$ for the different inputs, with $\mathcal{I}_{\text{MSE}}(\theta_0)$ given by (4.53) using $\{\widehat{\theta}_{T,i}\}_{i=1}^{N_{\text{est}}}$, the estimated parameters employed in Table 4.3 ($N_{\text{est}} = 35$). The results presented in this table show that the designed inputs reduce the volume of the uncertainty set associated with $\widehat{\theta}_T$ when compared to the other inputs. Hence, the proposed method designs an input that optimally distributes the uncertainty over the estimated parameters in the sense that the volume of the associated uncertainty set is minimized.

In conclusion, the input design method presented here can be employed to provide a better trade-off between the accuracies of the parameter estimates in this challenging example, when compared to standard inputs.

Implementation details

LGSS model

For the Monte Carlo study in Figure 4.1, we simulate 25 data sets with $T = 2.5 \cdot 10^4$ observations from (4.52) with $\theta_0 = \{0.5, 1.0\}$, no input and a known initial zero state. For the approach based on Kalman methods, an RTS smoother [156] is used to compute the score function and the FIM is estimated by (4.15). For the particle method, we make use of a fully adapted particle filter (faPA; [148]) in Algorithm 4.2 to estimate the FIM. This reduces the computational cost and increases the accuracy compared with bPF. However, the faPF can only be implemented for a small number of SSMs due to the assumptions on the sampling procedure.

For Algorithm 4.3, we make use of $T = 15 \cdot 10^3$, $n_m = 2$, and $n_{\mathcal{C}} = 3$ values (-1, 0 and 1). This results in 8 different basis inputs. Moreover, we use $h(\mathcal{I}_F^s(p_u, \theta), \theta) =$

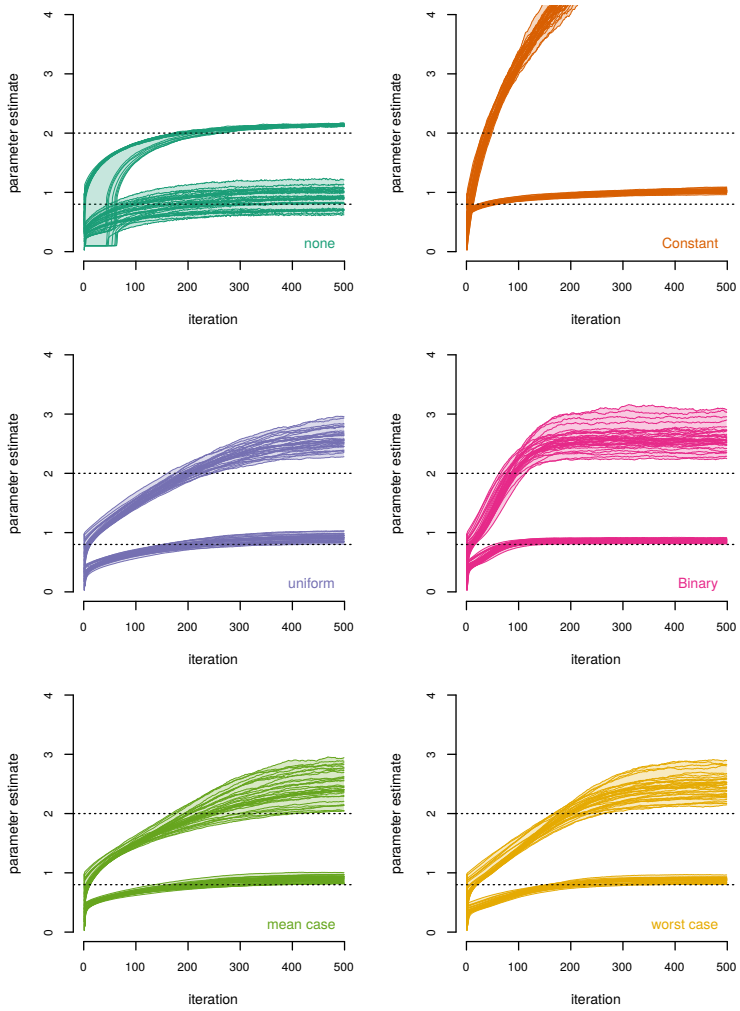

Figure 4.3: The evolution of the parameter estimates for (4.54) obtained from 35 runs of the EM algorithm with random initializations and different inputs. The dotted lines indicate the values $\gamma_0 = 2$ and $\beta_0 = 0.8$ used to generate the data from the model.

$-\log\det(\mathcal{I}_F^s(p_u, \theta))$ as the scalar cost function of the FIM. For Algorithm 4.2, we make use of $N = 200$, $M = 10$, $N_{\text{limit}} = 3$ and $\rho = 1$. To account for the uncertainty in the parameters, we sample $N_{\text{s}} = 100$ realizations uniformly from the parameter space $\Theta = \{(\phi, \sigma_v) : \phi \in (0.4, 0.6), \sigma_v \in (0.8, 1.2)\}$ and design the input accordingly. For the EM algorithm, we make use of $N = 200$, $M = 20$ and run it for 150

Table 4.3: Parameter estimates with the half-width of the 95% Bootstrapped CIs (parenthesis) and the log-MSE in the NL-SSM model computed from 35 independent runs of the EM algorithm using different (but fixed) input signals. Bold face marks the best values and Algorithm 4.2 is used to compute the Q-function in the EM algorithm. The true parameter values for θ are indicated between square brackets.

Input	$\widehat{\gamma}$ [2]	$\mathsf{MSE}(\widehat{\gamma})$	$\widehat{\beta}$ [0.8]	$\mathsf{MSE}(\widehat{\beta})$
none	2.14 (**0.01**)	**-3.99**	0.90 (0.05)	-3.46
constant	6.60 (0.23)	3.07	1.03 (**0.01**)	-2.96
uniform	2.59 (0.05)	-0.97	0.91 (0.02)	-4.14
binary	2.59 (0.06)	-0.95	0.86 (**0.01**)	**-5.38**
mean case	2.47 (0.09)	-1.26	0.89 (0.02)	-4.47
worst case	2.47 (0.06)	-1.36	0.88 (**0.01**)	-4.93

Table 4.4: $\log\{\det(\mathcal{I}_{\mathrm{MSE}}(\theta_0))\}$ for the parameter estimates of the NL-SSM model. Bold face marks the best value.

Input	$\log\{\det(\mathcal{I}_{\mathrm{MSE}}(\theta_0))\}$
none	-7.76
constant	-3.84
uniform	-7.31
binary	-7.94
mean case	-8.37
worst case	**-8.47**

iterations.

NSSM

For the Monte Carlo study performed with the NSSM, we generate a single data set with $T = 10^3$ observations from (4.54) using $\theta_0 = \{2, 0.8\}$. For Algorithm 4.3, we use $n_m = 2$, and $n_{\mathcal{C}} = 4$ values (-1, $-1/3$, $1/3$ and 1), resulting in 24 different basis inputs. Moreover, we use $h(\mathcal{I}_F^s(p_u, \theta), \theta) = -\log\det(\mathcal{I}_F^s(p_u, \theta))$ as the scalar cost function of the FIM. We make use of the bPF in Algorithm 4.2 with $N = 2.5 \cdot 10^3$, $M = 100$, $M_{\text{limit}} = 10$ and $\rho = 5$. For the mean and worst case costs, we sample $N_{\text{s}} = 30$ parameters uniformly from the set $\Theta = \{(\beta, \gamma) : \gamma \in (1.6, 2.4), \beta \in (0.6, 1)\}$. We run the EM algorithm for 500 iterations.

4.9 Conclusions

We have proposed a robust input design method for the identification of NSSMs. It is based on designing an input signal as a realization of a stationary process maximizing

a scalar cost function of the FIM. Since the model parameters are unknown a priori, the optimization of the experiment considers a measure of the uncertainty over the space of model parameters. Furthermore, we have provided numerical illustrations indicating that the designed input signal improves the accuracy of parameter estimates compared with other standard inputs.

One of the limitations of the proposed technique is its scalability to more complex scenarios. The computational complexity of the method is not only associated with the computation of the Fisher information matrix for every basis input (already discussed in Chapter 2), but also with the dimension of the parameter set considered for the robust framework. How to reduce the computational complexity for the technique presented here is part of the future research in the subject.

Chapter 5

Robust input design for systems with quantized inputs and measurements

In the previous chapter we introduced a framework to design inputs for general nonlinear state space models. By restricting the class of dynamical models, it is possible to obtain expressions for the Fisher information matrix that can be computed without the use of particle methods. This is the case in this chapter, where we use the graph theoretical framework for input design in FIR models with quantized outputs.

5.1 Introduction

Many applications in signal processing require to accurately estimate a model where only quantized measurements are available. This is of crucial importance, for example, in echo cancellation and blind communication channel estimation, see e.g. [84, 92, 130, 207] and the references therein.

In the case of quantized systems, the problem of model estimation and input design has been addressed in the literature [36, 37, 38, 199, 200, 201, 213]. In the case of input design, the assumptions on the quantizer properties (e.g., binary or uniform quantization) can limit their practical applicability. An attempt to solve this limitation is presented in [85], but it requires the prior knowledge of a nominal model.

Contribution

We present a robust input design method for the identification of finite impulse response (FIR) systems with quantized measurements, which extends the nominal input design proposed in [85]. The method relies on the technique presented in

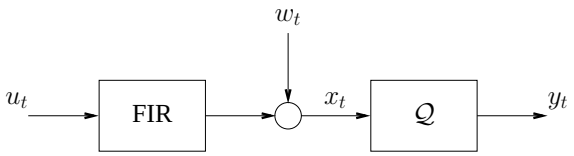

Figure 5.1: FIR system with quantized output.

Chapter 4. The method optimizes a scalar cost function of the Fisher information matrix, which is difficult to compute in closed form for systems with quantized measurements. To overcome this issue, we use a result in [84] to approximate such matrix. Moreover, since the true parameter is unknown before performing an experiment, here we consider a robust formulation, where the input is designed to minimize the maximum value of the objective function over the parameter space. The implementation of the resulting optimization is based on the scenario approach for approximating the robust formulation (see Appendix C).

The results in this chapter are connected to [188], where a general approach to robust input design for nonlinear models is proposed (cf. Chapter 4). However, the approach in [188] is based on particle methods, and hence it is computationally more demanding than this tailored approach to quantized FIR models.

Structure of the chapter

This chapter is organized as follows. Section 5.2 formulates the input design problem. A review on the ML estimation and some results from the theory of the expectation-maximization algorithm are detailed in Section 5.3. The use of the graph theoretical approach for input design in the context of this chapter is described in Section 5.4. Section 5.5 illustrates the discussion with a numerical example. Finally, Section 5.6 presents concluding remarks for this chapter.

5.2 Problem formulation

Consider the system described in Figure 5.1. The input u_t is filtered by a FIR system of order n_θ (i.e., its output at time t is linear in $u_{t-1}, \ldots, u_{t-n_\theta}$), and the output of this filter is corrupted by $\{w_t\}$, a Gaussian white noise with zero mean and known variance σ^2.

Remark 5.1 *The assumption about the noise variance being known is only necessary for designing the experiment, and not for the estimation problem. As shown in [84], the noise variance can also be estimated, and the asymptotic variance of the parameters of the FIR model is not affected by this.*

The output y_t is quantized via the quantizer $\mathcal{Q}\colon \mathbb{R} \to \mathcal{V}$, where $\mathcal{V} = \{v_i\}_{i=1}^{M}$ is a finite set of scalars, i.e.

$$\mathcal{Q}[x] = \begin{cases} v_1, & \text{if } x \in \Omega_1, \\ \vdots & \\ v_M, & \text{if } x \in \Omega_M, \end{cases} \tag{5.1}$$

where $\{\Omega_i\}_{i=1}^{M}$ is a partition of $\mathbb{R}$ such that Ω_i is a connected set, for all $i \in \{1, \ldots, M\}$. Hence, the system in Figure 5.1 can be written as

$$\begin{aligned} x_t &= \varphi_t^\top \theta + w_t, \quad w_t \sim \mathcal{N}(0, \sigma^2) \\ y_t &= \mathcal{Q}[x_t], \end{aligned} \tag{5.2}$$

where $x_t \in \mathbb{R}$, $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ is the parameter vector, $\varphi_t = [u_{t-1} \ \ldots \ u_{t-n_\theta}]^\top \in \mathbb{R}^{n_\theta}$ is the regressor vector, and $\sigma^2 \in \mathbb{R}$ is the noise variance. In the following, we assume that there exists a $\theta_0 \in \Theta$ such that (5.2) describes the true system.

The objective is to design the input sequence $u_{1:T}$ that maximizes the information about the parameter θ, quantified as a function of the Fisher information matrix

$$\mathcal{I}_F^T(u_{1:T}, \theta_0) = \mathbf{E}\left\{ \mathcal{S}_T(u_{1:T}, \theta_0)\, \mathcal{S}_T^\top(u_{1:T}, \theta_0) \middle|\, u_{1:T} \right\}, \tag{5.3}$$

where $\mathcal{S}_T(u_{1:T}, \theta_0)$ is the score function,

$$\mathcal{S}_T(u_{1:T}, \theta_0) = \nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta_0}, \tag{5.4}$$

$\log p_\theta(y_{1:T}|u_{1:T})$ is the log-likelihood function, and $p_\theta(y_{1:T}|u_{1:T})$ is the pdf of the measurements $y_{1:T}$, given the input sequence $u_{1:T}$ and $\theta \in \Theta$. We note that the expected value in (5.3) is with respect to $\{w_t\}$.

To quantify the Fisher information matrix, we consider a matrix convex function $h\colon \mathbb{R}^{n_\theta \times n_\theta} \times \Theta \to \mathbb{R}$. In this chapter we leave the choice of h to the user. Common definitions for h are listed in Table 1.1 on page 13.

To continue, we restrict the sequence $u_{1:T}$ to be a realization from an n_θ-dimensional stationary Markov process. In addition, we restrict u_t to a finite set $\mathcal{C}$ with $n_\mathcal{C}$ elements. The characterization of the stationary distribution of the Markov process is made in terms of n_θ-dimensional marginal pmfs of stationary processes as [190]

$$\begin{aligned} \mathcal{P}_\mathcal{C} = \Big\{ p_u : \mathcal{C}^{n_\theta} \to \mathbb{R} \Big|\, & p_u(\mathbf{x}) \geq 0, \, \forall \mathbf{x} \in \mathcal{C}^{n_\theta}; \\ & \sum_{\mathbf{x} \in \mathcal{C}^{n_\theta}} p_u(\mathbf{x}) = 1; \\ & \sum_{v \in \mathcal{C}} p_u(v, \mathbf{z}) = \sum_{v \in \mathcal{C}} p_u(\mathbf{z}, v), \forall \mathbf{z} \in \mathcal{C}^{n_\theta - 1} \Big\}. \end{aligned} \tag{5.5}$$

In the following, we will be interested in the *per-sample* Fisher information matrix,

$$\mathcal{I}_F^s(p_u, \theta_0) := \frac{1}{T}\mathbf{E}_u \left\{ \mathcal{I}_F^T(u_{1:T}, \theta_0) \right\} , \tag{5.6}$$

where the expected value in (5.6) is with respect to $u_{1:T}$.

Before stating the problem, we note that the Fisher information matrix (5.6) depends on the parameter θ_0 we want to estimate. To circumvent this issue, we consider a robust formulation, which minimizes the maximum value of $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$ over Θ. Thus, the problem can be stated as follows:

Problem 5.1 *Design $u_{1:T}$ as a realization of a Markov process with stationary distribution p_u^{opt}, where*

$$p_u^{\text{opt}} = \arg \min_{p_u \in \mathcal{P}_\mathcal{C}} \max_{\theta_0 \in \Theta} h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0) . \tag{5.7}$$

As mentioned in Chapter 4, Problem 5.1 is difficult to solve. The main difficulties are that the set Θ can be uncountable, and an efficient parametrization of $\mathcal{P}_\mathcal{C}$ is needed. These issues will be solved using the ideas presented in Chapter 4, where the scenario approach (cf. Appendix C) is used to approximate the set Θ by a finite number of samples, and the graph theory approach is used to characterize the set $\mathcal{P}_\mathcal{C}$. To continue, we need expressions derived from the expectation-maximization (EM) algorithm, which are presented in the next section.

5.3 Maximum likelihood and the EM algorithm for quantized systems

The maximum likelihood (ML) estimate of θ is given by

$$\hat{\theta}_T = \arg \max_{\theta \in \Theta} \log p_\theta(y_{1:T}|u_{1:T}) , \tag{5.8}$$

where $\log p_\theta(y_{1:T}|u_{1:T})$ can be rewritten as [87]

$$\log p_\theta(y_{1:T}|u_{1:T}) = \sum_{t=1}^{T} \ell_t(\theta) , \tag{5.9}$$

$$\ell_t(\theta) = \log p_\theta(y_t|y_{1:t-1}, u_{1:T}) . \tag{5.10}$$

Since the system (5.2) is FIR, $\ell_t(\theta)$ simplifies to

$$\begin{aligned} \ell_t(\theta) &= \log p_\theta(y_t|y_{1:t-1}, u_{1:T}) \\ &= \log p_\theta(y_t|u_{1:t-1}) . \end{aligned} \tag{5.11}$$

To continue, we require the following

Assumption 5.1 *The variance σ^2 is finite, and $\mathbf{E}\{\ell_t(\theta)\}$ has a unique (global) minimizer at θ_0, where the expected value is taken with respect to $\{w_t\}$.*

Assumption 5.1 guarantees that the solution $\hat{\theta}_T$ of the optimization problem in (5.8) converges (with probability 1) to the true solution θ_0 [205, Theorem 3.6]. In addition, Assumption 5.1 ensures identifiability of the system (5.2).

The maximum likelihood estimation (5.8) requires the computation of the log-likelihood function associated with a model described by (5.2), which is difficult to handle due to the quantizer in (5.2). To overcome this issue, we use the expectation maximization (EM) algorithm to solve the maximum likelihood problem (5.8) (cf. Appendix B). Moreover, some intermediate results of the EM algorithm are used to compute the Fisher information matrix associated with the model (5.2), as shown in Section 5.4. In the following, we briefly review the EM algorithm and present useful results for computing the Fisher information matrix.

The EM algorithm [56, 131] is an iterative procedure that at the k-th step seeks a value $\hat{\theta}_{T,k}$ such that the likelihood is increased in the sense that $\log p_{\hat{\theta}_{T,k}}(y_{1:T}|u_{1:T}) > \log p_{\hat{\theta}_{T,k-1}}(y_{1:T}|u_{1:T})$.

The main idea of the EM algorithm lies in the postulation of a *missing* data set $x_{1:T}$. In this chapter, the missing data $x_{1:T}$ will be understood as the state sequence $\{x_t\}$ in the model (5.2), but other choices are possible, and it can be considered as a design variable. This approach assumes that maximizing the joint log-likelihood $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$ is easier than maximizing the marginal one $\log p_\theta(y_{1:T}|u_{1:T})$. Here we briefly describe the EM method, and refer to Appendix B for more details.

The EM algorithm consists of two steps:

(i) the calculation of an auxiliary function (E-step), and

(ii) the optimization of this auxiliary function (M-step).

The E-step is obtained by calculating the function $\mathbf{Q}(\theta, \hat{\theta}_{T,i})$:

$$\mathbf{Q}(\theta, \hat{\theta}_{T,i}) = \int_{\mathbb{R}^T} \log[p_\theta(x_{1:T}, y_{1:T}|u_{1:T})]\, p_{\hat{\theta}_{T,i}}(x_{1:T}|y_{1:T}, u_{1:T}) dx_{1:T}. \tag{5.12}$$

The M-step is then given by

$$\hat{\theta}_{T,i+1} = \arg\max_{\theta\in\Theta} \mathbf{Q}(\theta, \hat{\theta}_{T,i}). \tag{5.13}$$

Before introducing one of the main results in this section, we need the following lemma [84, Lemma 5]:

Lemma 5.1 *The joint log-likelihood $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$, and the conditional pdf $p_{\hat{\theta}_{T,i}}(x_{1:T}|y_{1:T}, u_{1:T})$ are given by*

$$\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T}) = \begin{cases} \sum_{t=1}^{T} \log p_\theta(x_t|u_{1:t-1}), & x_t \in \mathcal{Q}^{-1}[y_t]\,, \\ 0, & \textit{otherwise,} \end{cases} \tag{5.14}$$

$$p_{\hat{\theta}_{T,i}}(x_{1:T}|y_{1:T}, u_{1:T}) = \prod_{t=1}^{T} p_{\hat{\theta}_{T,i}}(x_t|y_t, u_{1:t-1})\,, \tag{5.15}$$

with

$$p_{\hat{\theta}_{T,i}}(x_t|y_t, u_{1:t-1}) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} \exp\left\{-\frac{1}{2\sigma^2}(x_t - \varphi_t^\top \theta)^2\right\} dx_t\,. \tag{5.16}$$

Also, the E-step in the EM algorithm is

$$\mathbf{Q}(\theta,\, \hat{\theta}_{T,i}) = -\frac{1}{2}\sum_{t=1}^{T}\left[\log\{2\pi\sigma^2\} + \frac{1}{\sigma^2}\int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t}(x_t - \varphi_t^\top\theta)^2 p_{\hat{\theta}_{T,i}}(x_t|y_t)dx_t\right], \tag{5.17}$$

where $\mathcal{S}_t = \{x_t \in \mathbb{R}\colon \mathcal{Q}[x_t] = y_t\}$.

Proof *(Taken from [84]) The expression of the joint log-likelihood can be obtained as*

$$\begin{aligned} p_\theta(x_{1:T}, y_{1:T}|u_{1:T}) &= \prod_{t=1}^{T} p_\theta(x_t,\, y_t|x_{1:t-1}, y_{1:t-1}, u_{1:t-1}) \\ &= \prod_{t=1}^{T} p_\theta(x_t,\, y_t|u_{1:t-1})\,, \end{aligned} \tag{5.18}$$

with $p_\theta(x_t,\, y_t|u_{1:t-1})$ *defined by*

$$p_\theta(x_t,\, y_t|u_{1:t-1}) = \begin{cases} p_\theta(x_t|u_{1:t-1}), & x_t \in \mathcal{Q}^{-1}[y_t]\,, \\ 0, & \textit{otherwise,} \end{cases} \tag{5.19}$$

and from [87, p. 210] we know that

$$x_t \sim \mathcal{N}(\varphi_t^\top\theta,\, \sigma^2)\,. \tag{5.20}$$

Hence,

$$\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T}) = \sum_{t=1}^{T} \log p_\theta(x_t,\, y_t|u_{1:t-1})\,, \tag{5.21}$$

and (5.14) *follows from* (5.19). *Also, from* (5.18) *and*

$$p_\theta(y_{1:T}|u_{1:T}) = \prod_{t=1}^{T} p_\theta(y_t|y_{1:t-1}, u_{1:t-1}) = \prod_{t=1}^{T} p_\theta(y_t|u_{1:t-1})\,, \tag{5.22}$$

we have that

$$p_{\hat{\theta}_{T,i}}(x_{1:T}|y_{1:T}, u_{1:T}) = \frac{p_{\hat{\theta}_{T,i}}(x_{1:T}, y_{1:T}|u_{1:T})}{p_{\hat{\theta}_{T,i}}(y_{1:T}|u_{1:T})}$$

$$= \prod_{t=1}^{T} \frac{p_{\hat{\theta}_{T,i}}(x_t, y_t|u_{1:t-1})}{p_{\hat{\theta}_{T,i}}(y_t|u_{1:t-1})} = \prod_{t=1}^{T} p_{\hat{\theta}_{T,i}}(x_t|y_t, u_{1:t-1})\,. \quad (5.23)$$

In the view of (5.12)*, to compute the E-step we need the expressions for the joint log-likelihood* $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$*, and the conditional pdf* $p_{\hat{\theta}_{T,i}}(x_t|y_t, u_{1:t-1})$*. Combining* (5.12) *with* (5.21) *and* (5.23) *we have that*

$$\mathbf{Q}(\theta, \hat{\theta}_{T,i}) = \int_{\mathbb{R}^T} \sum_{t=1}^{T} \log[p_\theta(x_t, y_t|u_{1:t-1})] \prod_{s=1}^{T} p_{\hat{\theta}_{T,i}}(x_s|y_s, u_{1:s-1}) dx_{1:T} \quad (5.24)$$

$$= \sum_{t=1}^{T} \int_{\mathbb{R}} \log[p_\theta(x_t, y_t|u_{1:t-1})]\, p_{\hat{\theta}_{T,i}}(x_t|y_t, u_{1:t-1}) dx_t\,, \quad (5.25)$$

and (5.17) *follows from* (5.20) *and* (5.19)*.* ■

The EM algorithm is summarized in the following theorem:

Theorem 5.1 *Consider the system* (5.2)*, and the maximization problem stated in* (5.13)*. If* $\{u_t\}$ *and* $\{w_t\}$ *are mutually independent, the M-step of the EM algorithm is given by:*

$$\hat{\theta}_{T,i+1} = \left[\sum_{t=1}^{T} \varphi_t \varphi_t^\top\right]^{-1} \sum_{t=1}^{T} \varphi_t \hat{x}_t, \quad (5.26)$$

where

$$\hat{x}_t = \varphi_t^\top \hat{\theta}_{T,i} + \sigma \frac{I_t^{(1)}}{I_t^{(0)}}, \quad (5.27)$$

with $I_t^{(0)}, I_t^{(1)} \in \mathbb{R}$ *given by*

$$I_t^{(0)} = \frac{1}{\sqrt{2\pi}} \int_{\inf \tilde{\mathcal{S}}_t}^{\sup \tilde{\mathcal{S}}_t} \exp\left\{-\frac{1}{2}\tilde{x}_t^2\right\} d\tilde{x}_t, \quad (5.28)$$

$$I_t^{(1)} = \frac{1}{\sqrt{2\pi}} \int_{\inf \tilde{\mathcal{S}}_t}^{\sup \tilde{\mathcal{S}}_t} \tilde{x}_t \exp\left\{-\frac{1}{2}\tilde{x}_t^2\right\} d\tilde{x}_t, \quad (5.29)$$

and $\tilde{\mathcal{S}}_t = \{\tilde{x}_t \in \mathbb{R}\colon \mathcal{Q}[\sigma\tilde{x}_t + \varphi_t^\top \hat{\theta}_{T,i}] = y_t\}$*.*

Proof *(Taken from [84]) The maximization of* (5.13) *with respect to* θ *can be found by taking the derivative in* (5.17) *with respect to* θ*, and using the definition of* $\mathcal{S}_t$*. Thus,*

$$-2\nabla_\theta \mathbf{Q}(\theta, \hat{\theta}_{T,i}) = \nabla_\theta \left(\sum_{t=1}^{T} \left[\log\{2\pi\sigma^2\} + \frac{1}{\sigma^2} \int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} (x_t - \varphi_t^\top \theta)^2 p_{\hat{\theta}_{T,i}}(x_t|y_t, u_{1:t-1}) dx_t \right] \right)$$

$$= \sum_{t=1}^{T} \left[\frac{1}{\sigma^2} \int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} \nabla_\theta (x_t - \varphi_t^\top \theta)^2 p_{\hat{\theta}_{T,i}}(x_t | y_t, u_{1:t-1}) dx_t \right]$$
$$= \sum_{t=1}^{T} \left[-\frac{2}{\sigma^2} \int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} \varphi_t (x_t - \varphi_t^\top \theta) p_{\hat{\theta}_{T,i}}(x_t | y_t, u_{1:t-1}) dx_t \right] . \quad (5.30)$$

Setting (5.30) *equal to zero we have that*

$$\hat{\theta}_{T,i+1} = \left[\sum_{t=1}^{T} \varphi_t \varphi_t^\top \right]^{-1} \sum_{t=1}^{T} \varphi_t \hat{x}_t \,, \quad (5.31)$$

where

$$\hat{x}_t = \int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} x_t \, p_{\hat{\theta}_{T,i}}(x_t | y_t, u_{1:t-1}) \, dx_t \,. \quad (5.32)$$

From (5.32) *we have*

$$\hat{x}_t = \int_{\mathbb{R}} x_t \frac{p_{\hat{\theta}_{T,i}}(x_t,\, y_t | u_{1:t-1})}{p_{\hat{\theta}_{T,i}}(y_t | u_{1:t-1})} \, dx_t = \frac{1}{p_{\hat{\theta}_{T,i}}(y_t | u_{1:t-1})} \int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} x_t p_{\hat{\theta}_{T,i}}(x_t | u_{1:t-1}) \, dx_t$$
$$= \frac{1}{p_{\hat{\theta}_{T,i}}(y_t | u_{1:t-1})} \int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} \frac{x_t}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} (x_t - \varphi_t^\top \hat{\theta}_{T,i})^2 \right\} dx_t \,. \quad (5.33)$$

Making the change of variables $\tilde{x}_t = \sigma^{-1}(x_t - \varphi_t^\top \hat{\theta}_{T,i})$*, we have*

$$\hat{x}_t = \frac{1}{p_{\hat{\theta}_{T,i}}(y_t | u_{1:t-1})} \int_{\inf \tilde{\mathcal{S}}_t}^{\sup \tilde{\mathcal{S}}_t} \frac{(\sigma \tilde{x}_t + \varphi_t^\top \hat{\theta}_{T,i})}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\tilde{x}_t^2} \, d\tilde{x}_t$$
$$= \frac{1}{p_{\hat{\theta}_{T,i}}(y_t | u_{1:t-1})} \left[\int_{\inf \tilde{\mathcal{S}}_t}^{\sup \tilde{\mathcal{S}}_t} \frac{\sigma \tilde{x}_t}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\tilde{x}_t^2} \, d\tilde{x}_t + \varphi_t^\top \hat{\theta}_{T,i} \int_{\inf \tilde{\mathcal{S}}_t}^{\sup \tilde{\mathcal{S}}_t} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\tilde{x}_t^2} \, d\tilde{x}_t \right]$$
$$= \frac{1}{p_{\hat{\theta}_{T,i}}(y_t | u_{1:t-1})} \left[\sigma I_t^{(1)} + \varphi_t^\top \hat{\theta}_{T,i} I_t^{(0)} \right] . \quad (5.34)$$

Also,

$$p_{\hat{\theta}_{T,i}}(y_t | u_{1:t-1}) = \int_{\mathbb{R}} p_{\hat{\theta}_{T,i}}(x_t,\, y_t | u_{1:t-1}) \, dx_t = \int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} p_{\hat{\theta}_{T,i}}(x_t | u_{1:t-1}) \, dx_t$$
$$= \int_{\inf \tilde{\mathcal{S}}_t}^{\sup \tilde{\mathcal{S}}_t} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2} \tilde{x}_t^2 \right\} \sigma \, d\tilde{x}_t$$
$$= I_t^{(0)} \,. \quad (5.35)$$

Finally, the expression for $\hat{x}_t$ *follows from* (5.34) *and* (5.35). ■

Under the assumption that $\tilde{\mathcal{S}}_t$ is a connected set, we can develop further the expressions for the integrals $I_t^{(0)}$, $I_t^{(1)}$ to obtain:

$$I_t^{(0)} = \frac{1}{2}\left(\operatorname{erf}\left(\frac{\sup \tilde{\mathcal{S}}_t}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{\inf \tilde{\mathcal{S}}_t}{\sqrt{2}}\right)\right), \tag{5.36}$$

$$I_t^{(1)} = \frac{e^{-[\inf \tilde{\mathcal{S}}_t]^2/2} - e^{-[\sup \tilde{\mathcal{S}}_t]^2/2}}{\sqrt{2\pi}}, \tag{5.37}$$

where

$$\operatorname{erf}(x) := \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du \tag{5.38}$$

is the error function.

5.4 Input design

Input design via graph theory

From Chapter 2, we know that $\mathcal{P}_{\mathcal{C}}$ corresponds to the convex hull of pmfs associated with prime cycles in a graph.

As discussed in Chapter 2, the set $\mathcal{C}^{n_\theta}$ of possible values for $u_{1:n_\theta}$ can be mapped to a de Bruijn graph $\mathcal{G}_{\mathcal{C}^{n_\theta}}$ (Definition 2.6 on page 32). Moreover, it can be shown that the convex hull of the pmfs associated with the prime cycles of de Bruijn graph $\mathcal{G}_{\mathcal{C}^{n_\theta}}$ defines the set of stationary pmfs over the graph (Theorem 2.1). Finally, the prime cycles of the de Bruijn graph can be derived from the elementary cycles of $\mathcal{G}_{\mathcal{C}^{n_\theta-1}}$ (cf. Lemma 2.3), which, in turn, can be determined using standard graph algorithms [103, 184]. We refer to Chapter 2 for a more detailed discussion about this topic.

Robust input design algorithm

The robust version of the input design method in [85] can be described as follows:

1. Compute all the elementary cycles of $\mathcal{G}_{\mathcal{C}^{n_\theta-1}}$.
2. Compute all the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_\theta}}$ by extending the elementary cycles of $\mathcal{G}_{\mathcal{C}^{n_\theta-1}}$ (cf. Lemma 2.3).
3. Generate the input signals $\{u_{1:N_{\text{sim}}}^{(i)}\}_{i=1}^{n_{\mathcal{V}}}$ by traversing the prime cycles, with $n_{\mathcal{V}}$ the number of prime cycles (N_{sim} must be sufficiently large as will be described in the next subsection).
4. Sample $\{\theta_j\}_{j=1}^{N_{\text{s}}}$ from Θ according to the predefined cdf P_Θ.
5. Approximate $\mathcal{I}_F^s(p_u^{(i)}, \theta_j)$ using $u_{1:N_{\text{sim}}}^{(i)}$ as input and θ_j as parameter. This step will be detailed in the next subsection.

6. Define $\gamma = \{\alpha_1, \ldots, \alpha_{n_\mathcal{V}}\}$ Find $\gamma^{\text{opt}} = \{\alpha_1^{\text{opt}}, \ldots, \alpha_{n_\mathcal{V}}^{\text{opt}}\}$ by solving:

$$\gamma^{\text{opt}} = \arg\max_{\gamma \in \mathbb{R}^n} \min_{\theta \in \{\theta_j\}_{j=1}^{N_\text{s}}} h(\mathcal{I}_F^{\text{app}}(\gamma, \theta), \theta)\,, \tag{5.39}$$

where

$$\mathcal{I}_F^{\text{app}}(\gamma, \theta_j) = \sum_{i=1}^{n} \alpha_i \mathcal{I}_F^s(p_u^{(i)}, \theta_j)\,, \text{ for all } j \in \{1, \ldots, N_\text{s}\}. \tag{5.40}$$

$$\sum_{i=1}^{n_\mathcal{V}} \alpha_i = 1, \tag{5.41}$$

$$\alpha_i \geq 0\,, \text{ for all } i \in \{1, \ldots, n_\mathcal{V}\}. \tag{5.42}$$

This procedure computes γ^{opt} defining $p_u^{\text{opt}} = \sum_{i=1}^{n_\mathcal{V}} \alpha_i^{\text{opt}} p_u^{(i)}$. Notice that the problem (5.39)-(5.42) is convex.

To generate the resulting input signal from the optimal pmf p_u^{opt}, we run a Markov chain with p_u^{opt} as stationary distribution. We refer to Section 4.6 on page 74 for more details.

Computation of the Fisher information matrix

Before introducing the main result for estimating the Fisher information matrix, we state the following lemma:

Lemma 5.2 *The partial derivative of* $\mathbf{Q}(\theta, \theta_i)$ *with respect to* θ *can be expressed as*

$$\nabla_\theta \mathbf{Q}(\theta, \theta_i) = \sum_{t=1}^{T} \left(\frac{1}{\sigma^2} \varphi_t \varphi_t^\top [\theta_i - \theta] + \frac{\varphi_t}{\sigma} \frac{I_t^{(1)}}{I_t^{(0)}} \right). \tag{5.43}$$

Proof *We start from* (5.30) *to write*

$$\begin{aligned} \nabla_\theta \mathbf{Q}(\theta, \theta_i) &= \sum_{t=1}^{T} \left[\frac{1}{\sigma^2} \int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} \varphi_t (x_t - \varphi_t^\top \theta) p_{\hat{\theta}_{T,i}}(x_t | y_t, u_{1:t-1}) dx_t \right] \\ &= \sum_{t=1}^{T} \frac{\varphi_t}{\sigma^2} \left[\hat{x}_t - \varphi_t^\top \theta \int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} p_{\theta_i}(x_t | y_t, u_{1:t-1}) dx_t \right], \end{aligned} \tag{5.44}$$

where $\hat{x}_t$ *is defined according to Theorem 5.1. Moreover, since* $p_{\theta_i}(x_t | y_t, u_{1:t-1})$ *is nonzero only in* $\mathcal{S}_t$*, we have*

$$\int_{\inf \mathcal{S}_t}^{\sup \mathcal{S}_t} p_{\theta_i}(x_t | y_t, u_{1:t-1}) dx_t = 1\,. \tag{5.45}$$

Finally, using (5.27) *and* (5.45) *into* (5.44) *we obtain* (5.43). ■

The expression for $\mathcal{I}_F^s$ for the system described in equation (5.2) is provided in the following theorem:

Theorem 5.2 *(Fisher information matrix for models with quantized output) For systems with quantized output data of the form* (5.2) *and a connected set* $\tilde{\mathcal{S}}_t = \{\tilde{x}_t \in \mathbb{R}\colon \mathcal{Q}[\sigma\tilde{x}_t + \varphi_t^\top\theta] = y_t\}$*, the Fisher information matrix is given by*

$$\mathcal{I}_F^s(p_u,\theta) = \lim_{N_{\text{sim}}\to\infty} \frac{2}{\pi\sigma^2 N_{\text{sim}}} \sum_{t=1}^{N_{\text{sim}}} \left(\frac{e^{-[\sup\tilde{\mathcal{S}}_t]^2/2} - e^{-[\inf\tilde{\mathcal{S}}_t]^2/2}}{\operatorname{erf}([\sup\tilde{\mathcal{S}}_t]/\sqrt{2}) - \operatorname{erf}([\inf\tilde{\mathcal{S}}_t]/\sqrt{2})} \right)^2 \varphi_t\varphi_t^\top, \tag{5.46}$$

with probability 1.

Proof *We start by using the Fisher identity*[1]

$$\nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta_0} = \nabla_\theta \mathbf{Q}(\theta,\theta_0)|_{\theta=\theta_0} \tag{5.47}$$

where $\mathbf{Q}(\theta,\theta_0)$ *is the auxiliary function arising from the EM algorithm.*

From Lemma 5.2 we have that

$$\nabla_\theta \mathbf{Q}(\theta,\theta_0)|_{\theta=\theta_0} = \frac{1}{\sigma}\sum_{t=1}^{T} \varphi_t \frac{I_t^{(1)}}{I_t^{(0)}}. \tag{5.48}$$

For the SISO case, the quotient $I_t^{(1)}/I_t^{(0)}$ *can be expressed as:*

$$\frac{I_t^{(1)}}{I_t^{(0)}} = -\frac{\sqrt{2}}{\sqrt{\pi}} \frac{e^{-[\sup\tilde{\mathcal{S}}_t]^2/2} - e^{-[\inf\tilde{\mathcal{S}}_t]^2/2}}{\operatorname{erf}([\sup\tilde{\mathcal{S}}_t]/\sqrt{2}) - \operatorname{erf}([\inf\tilde{\mathcal{S}}_t]/\sqrt{2})}. \tag{5.49}$$

Thus, we can write the Fisher information matrix as

$$\begin{aligned}\mathcal{I}_F^s(p_u,\theta) &= \frac{1}{T}\mathbf{E}\left\{ \nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta_0}\ \nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})^\top\big|_{\theta=\theta_0} \right\} \\ &= \frac{2}{\pi\sigma^2}\mathbf{E}\left\{ \left(\frac{e^{-[\sup\tilde{\mathcal{S}}_t]^2/2} - e^{-[\inf\tilde{\mathcal{S}}_t]^2/2}}{\operatorname{erf}([\sup\tilde{\mathcal{S}}_t]/\sqrt{2}) - \operatorname{erf}([\inf\tilde{\mathcal{S}}_t]/\sqrt{2})} \right)^2 \varphi_t\varphi_t^\top \right\}. \end{aligned} \tag{5.50}$$

Since x_t *and* φ_t *are asymptotically jointly stationary processes, which are asymptotically uncorrelated (for sufficiently large lags), by the Birkhoff-Khinchin ergodic theorem [62, Theorem 7.2.1], we can write* (5.50) *as:*

$$\mathcal{I}_F^s(p_u,\theta) = \lim_{N_{\text{sim}}\to\infty} \frac{2}{\pi\sigma^2 N_{\text{sim}}} \sum_{t=1}^{N_{\text{sim}}} \left(\frac{e^{-[\sup\tilde{\mathcal{S}}_t]^2/2} - e^{-[\inf\tilde{\mathcal{S}}_t]^2/2}}{\operatorname{erf}([\sup\tilde{\mathcal{S}}_t]/\sqrt{2}) - \operatorname{erf}([\inf\tilde{\mathcal{S}}_t]/\sqrt{2})} \right)^2 \varphi_t\varphi_t^\top, \tag{5.51}$$

with probability 1, which is the expression in (5.46). ■

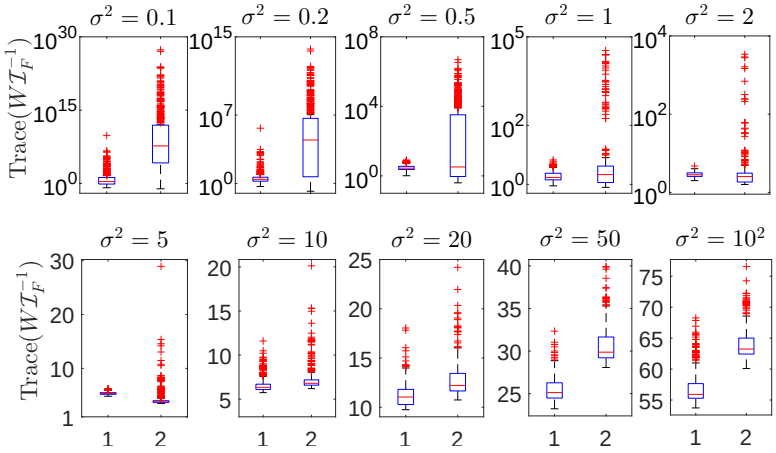

Figure 5.2: Box plots for $-h(\mathcal{I}_F(\theta))$ for $5 \cdot 10^2$ samples uniformly distributed over Θ, and different values of σ^2. In each plot, the left box corresponds to the optimal input, and the right box corresponds to using binary distributed white noise as input.

5.5 Numerical example

Consider a SISO FIR system given by (5.2), where $\varphi_t^\top = \begin{bmatrix} u_{t-1} & u_{t-2} \end{bmatrix}$, $\theta \in \Theta \subset \mathbb{R}^2$, and the noise variance σ^2 is known. We assume that $\Theta = \{\begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}^\top : \theta_1 \in [-1, 1],\, \theta_2 \in [-1, 1]\}$, which is approximated by samples $\{\theta_j\}_{j=1}^{500}$ obtained from a uniform distribution over Θ. These samples guarantee that, with given probability $1 - \beta$, the solution to (5.39) is a lower bound for $h(\mathcal{I}_F^s(p_u^{\mathrm{opt}}, \theta), \theta)$ over Θ (with probability at least $1 - \varepsilon$) if

$$\sum_{i=0}^{n_{\mathcal{V}}} \binom{500}{i} \varepsilon^i (1-\varepsilon)^{500-i} \leq \beta\,, \tag{5.52}$$

holds, where $n_{\mathcal{V}}$ is the number of prime cycles and $\varepsilon \in (0, 1)$ is defined by the user [29]. For example, if $\beta = 1.6 \cdot 10^{-5}$ and $n_{\mathcal{V}} = 24$, then with probability $1 - 1.6 \cdot 10^{-5}$ the solution to (5.39) provides a lower bound for $h(\mathcal{I}_F^s(p_u, \theta), \theta)$ over 90% of the θ's in Θ. We refer to Appendix C for more details on the scenario approach.

[1] We refer to Appendix B, page 194 for the derivation of the Fisher identity.

We consider a 2-bit quantizer $\mathcal{Q}$ given by

$$\mathcal{Q}[x] = \begin{cases} 7.5, & \text{if } x > 2.5, \\ 2.5, & \text{if } x \in (0,\, 2.5], \\ -2.5, & \text{if } x \in [-2.5,\, 0], \\ -7.5, & \text{if } x < -2.5. \end{cases} \tag{5.53}$$

We design an input with $\mathcal{C} = \{-5, -10/3, 10/3, 5\}$ (hence $n_{\mathcal{V}} = 24$), and solve (5.39) for $h(\cdot) = \mathrm{tr}\{W(\cdot)^{-1}\}$, where

$$W = \begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}. \tag{5.54}$$

The approximation of each $\mathcal{I}_F^s(p_u^{(i)}\theta_j)$ in (5.40) is obtained by using (5.46) with $N_{\mathrm{sim}} = 10^3$. The problem is solved in `Matlab` using the `cvx` toolbox [21]. Finally, we generate $u_{1:T}^{\mathrm{opt}}$ with $T = 10^3$ from p_u^{opt} by running a Markov chain as in Chapter 2.

Figure 5.2 presents the box plots obtained for $\mathrm{tr}\{W[\mathcal{I}_F^s(p_u^{\mathrm{opt}}, \theta)]^{-1}\}$ when the optimal input is employed to excite the system (5.2), for $5{\cdot}10^2$ samples of θ uniformly distributed over Θ, and different values of σ^2. To study the performance of the designed input, we also excite the system (5.2) for the same parameter realizations with a binary distributed white noise process, with values in $\{-5, 5\}$. The desired behavior for $\mathrm{tr}\{W[\mathcal{I}_F^s(p_u^{\mathrm{opt}}, \theta)]^{-1}\}$ is to be as close to zero as possible for all $\theta \in \Theta$. Hence, inputs yielding $\mathrm{tr}\{W[\mathcal{I}_F^s(p_u^{\mathrm{opt}}, \theta)]^{-1}\}$ close to zero are preferred.

Based on Figure 5.2, we see that the optimal input improves the amount of information retrieved from the system in the worst case (i.e., for the maximum value of $\mathrm{tr}\{W[\mathcal{I}_F^s(p_u^{\mathrm{opt}}, \theta)]^{-1}\}$ over Θ), when compared to the binary input. Moreover, this improvement is significant for $\sigma^2 \leq 2$, where the difference for the worst case value is of several orders of magnitude. However, this improvement is achieved by sacrificing the performance for the best case (i.e., for the minimum value of $\mathrm{tr}\{W[\mathcal{I}_F^s(p_u^{\mathrm{opt}}, \theta)]^{-1}\}$ over Θ), which can be seen for values of $\sigma^2 \leq 5$. This is a common trade-off in robust input design, where the uncertainty on the model parameters degrades the maximum achievable value for the cost function.

5.6 Conclusions

This chapter has introduced an input design procedure for identification of FIR systems with quantized measurements under amplitude limitations. Since the model parameters are unknown, the technique considers a robust approach, where the input is designed by minimizing the maximum value of the objective function over the parameter set. To solve the resulting optimization, we employ the scenario approach for approximating the parameter set, and a graph theoretical technique to optimize the input sequence.

Part II

Bayesian techniques in system identification

Chapter 6

A Gaussian process optimization method to input design for the identification of nonlinear models

In the first part of this thesis, we have presented a graph theoretical method for designing inputs for identification of nonlinear models. The assumption of a Markov process for the input allows us to solve a convex problem, which is an approximation of the original formulation. In this chapter, we explore another alternative for solving the input design problem, where we make use of Bayesian optimization and Gaussian processes.

6.1 Introduction

As mentioned in Chapter 1, the objective in input design is to maximize the information in the experiment about the system to be identified, quantified as a scalar associated with the intended model application. A standard choice for the scalar cost is a function of the Fisher information matrix. However, for nonlinear models, the Fisher information matrix is often unavailable in closed form.

As the Fisher information matrix does not have a closed form expression, we need to rely on estimates. However, such estimates are always uncertain, due to the uncertainty associated with the Monte Carlo estimators. This results in difficulties when implementing part of the available optimization methods, as the information about the uncertainty of the estimates needs to be considered in the problem formulation.

Contribution

In this chapter, we explore the reduction of the computational complexity when calculating the objective function used in input design for nonlinear dynamical

models. To this end, a Gaussian process optimization (GPO) algorithm is presented. By specifying that the scalar function of the Fisher information matrix is a realization of a Gaussian process (GP), we can compute its predictive posterior distribution given a set of samples over the feasible set. The predictive posterior distribution acts as a surrogate of the objective function, and is employed to compute the next sample over the feasible set by using an acquisition rule. This technique recursively explores the feasible set to determine the element minimizing a surrogate function. The advantage of this approach when compared with existing techniques is that it can handle uncertainty in the estimates of the objective function, and it drives the exploration of the input space towards those regions where either an improvement of the objective function is expected or there is large uncertainty on the current estimate.

To simplify the presentation of the method, in this chapter we rely on prior information about the system for computing an optimal design. This assumption can be overcome by implementing an adaptive scheme on top of it [80], where the input is recursively designed as the estimated parameter is updated, or by using a robust input design scheme as presented in the first part of this thesis.

Structure of the chapter

This chapter is organized as follows. Section 6.2 presents the problem statement. Section 6.3 introduces the Gaussian process optimization to solve the input design problem. Section 6.4 presents numerical examples. Finally, Section 6.5 concludes this chapter.

6.2 Problem formulation

Consider the discrete time, nonlinear state space model defined for all $t \geq 1$ by

$$x_t|x_{t-1} \sim f_\theta(x_t|x_{t-1}, u_{t-1}), \tag{6.1a}$$
$$y_t|x_t \sim g_\theta(y_t|x_t, u_t), \tag{6.1b}$$
$$x_0 \sim \mu_\theta(x_0), \tag{6.1c}$$

where f_θ, g_θ, and μ_θ are pdfs parameterized by $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$. Here, $u_t \in \mathcal{C} \subseteq \mathbb{R}^{n_u}$ denotes the input signal, $x_t \in \mathbb{R}^{n_x}$ are the (unobserved/latent) internal states, and $y_t \in \mathbb{R}^{n_y}$ are the measured outputs. We assume that there exists a $\theta_0 \in \Theta$ such that the pdfs in (6.1) describe the true pdfs of the system when $\theta = \theta_0$ (i.e., there is no undermodelling).

The objective is to design $u_{1:T} \in \mathcal{C}^T$, such that the parameter θ in the model (6.1) can be identified with maximum accuracy as defined by a scalar function of the Fisher information matrix $\mathcal{I}_F^T(u_{1:T}, \theta_0)$ [87],

$$\mathcal{I}_F^T(u_{1:T}, \theta_0) = \mathbf{E}\left\{\mathcal{S}_T(u_{1:T}, \theta_0)\mathcal{S}_T^\top(u_{1:T}, \theta_0)|\, u_{1:T}\right\}, \tag{6.2}$$

where $\mathcal{S}_T(u_{1:T}, \theta_0)$ is the score function,

$$\mathcal{S}_T(u_{1:T}, \theta_0) = \nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta_0} \ . \tag{6.3}$$

We note that the expected value in (6.2) is with respect to the stochastic processes in (6.1).

In the following, we consider $u_{1:T}$ as a realization of a stationary process and we will be interested in the per-sample Fisher information matrix,

$$\mathcal{I}_F^s(p_u, \theta_0) = \frac{1}{T}\mathbf{E}_u \left\{\mathcal{I}_F^T(u_{1:T}, \theta_0)\right\} \ . \tag{6.4}$$

The input $u_{1:T}$ optimizes a scalar function of (6.4). We define this scalar function as $h\colon \mathbb{R}^{n_\theta \times n_\theta} \times \Theta \to \mathbb{R}$, which is assumed to be a matrix convex function in its first argument for every $\theta \in \Theta$.

The problem presented here can be summarized as

Problem 6.1 *Find an input signal $u_{1:T}^{\text{opt}} \in \mathcal{C}^T$ as a realization from p_u^{opt}, where*

$$p_u^{\text{opt}} := \arg\min_{p_u \in \mathcal{P}} h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0) \, , \tag{6.5}$$

where $h\colon \mathbb{R}^{n_\theta \times n_\theta} \times \Theta \to \mathbb{R}$ is a matrix convex function in its first argument for every $\theta \in \Theta$, and $\mathcal{I}_F^s(p_u, \theta_0)$ is given in (6.4).

6.3 Gaussian process optimization in input design

Problem 6.1 is difficult to solve. One of the main challenges is the characterization of $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$ for all p_u associated with stationary processes. Unless assumptions on the model structure (6.1) and the input properties are made, the expression $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$ is often unavailable, and we need to rely on approximations. Moreover, even if an estimate of $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$ is available, part of the existing optimization methods do not handle the uncertainty of the estimate employed to perform the computations.

Remark 6.1 *An alternative to solving optimization problems under noisy observations is to use stochastic approximation methods [39, 181]. It can be shown that, under some technical conditions, the output of the stochastic approximation method converges to the optimal solution as the number of iterations goes to infinity. This technique will be explored in Chapter 9 in the context of robust application oriented input design.*

To deal with the problem of noisy estimates, here we employ the iterative procedure discussed in [48] to solve Problem 6.1. The procedure generates a sequence of iterates $\{p_u^{(k)}\}_{k\geq 0}$ for the input excitation, where k denotes the k-th iteration. Each iteration consists of three steps:

(i) Given $p_u^{(k)}$, compute an estimate of the objective function $h(\mathcal{I}_F^s(p_u^{(k)}, \theta_0), \theta_0)$, denoted by $\widehat{h}_k$.

(ii) Given the collection of tuples $\{p_u^{(j)}, \widehat{h}_j\}_{j=0}^k$, create a model of the objective function $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$.

(iii) Use the model as a surrogate for $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$ to generate a new iterate $p_u^{(k+1)}$.

The procedure only requires one estimate of $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$ at each iteration, hence keeping the number of estimates as low as possible. Moreover, it requires fewer iterations than a random search, since it focuses on regions of $\mathcal{P}$ where an improvement is expected.

For step (i), we employ particle methods to estimate $h(\mathcal{I}_F^s(p_u^{(k)}, \theta_0), \theta_0)$. This is briefly discussed in the next subsection, where we follow the method introduced in Chapter 4.

For steps (ii) and (iii) we use the GPO framework [139, 174]. We first compute a surrogate of the objective function by modelling it as a Gaussian process, and compute the predictive posterior distribution based on $\{p_u^{(j)}, \widehat{h}_j\}_{j=0}^k$. This is discussed on page 107.

Then we make use of a heuristic, referred to as the *acquisition rule* (presented on page 116), to compute $p_u^{(k+1)}$ based on the Gaussian process (GP) model. The acquisition rule favours values of $u_{1:T}$ for which the model predicts a low value of the objective function and/or where there is high uncertainty. This establishes a trade-off between exploration and exploitation of the input set. Finally, to employ the GPO framework in input design, we need tractable parameterizations of $\mathcal{P}$, which are discussed on page 118.

Estimating the Fisher information matrix

Given $p_u^{(k)} \in \mathcal{P}$, we need to approximate (6.4). To this end, we sample $u_{1:T}^{(k)} \in \mathcal{C}^T$ from $p_u^{(k)}$ and we estimate the Fisher information matrix as in Chapter 4. This estimator is based on one estimate of $\mathcal{S}_T(u_{1:T}, \theta_0)$ (provided a sufficiently large T) to approximate (6.4) by using [173]

$$\widehat{\mathcal{I}}_F^s(p_u^{(k)}, \theta_0) := \frac{1}{T}\left[\sum_{t=1}^{T} \widehat{\mathcal{S}}_{t,T}(u_{1:T}^{(k)}, \theta_0)(\widehat{\mathcal{S}}_{t,T}(u_{1:T}^{(k)}, \theta_0))^\top - \frac{1}{T}\widehat{\mathcal{S}}_T(u_{1:T}^{(k)}, \theta_0)(\widehat{\mathcal{S}}_T(u_{1:T}^{(k)}, \theta_0))^\top\right], \quad (6.6)$$

where the Fisher identity (cf. Theorem B.1) can be used to write

$$\mathcal{S}_T(u_{1:T}, \theta') = \sum_{t=1}^{T} \mathcal{S}_{t,T}(u_{1:T}, \theta'), \quad (6.7)$$

$$\mathcal{S}_{t,T}(u_{1:T}, \theta') = \int \nabla_\theta \, \xi_\theta(x_{t-1:t})|_{\theta=\theta'} p_{\theta'}(x_{t-1:t}|y_{1:T}, u_{1:T}) \, \mathrm{d}x_{t-1:t} \,, \tag{6.8}$$

with

$$\xi_\theta(x_{t-1:t}) = \log f_\theta(x_t|x_{t-1}, u_{t-1}) + \log g_\theta(y_t|x_t, u_t) \,.$$

As we can see from (6.6), we require an estimate for (6.7), which we obtain using particle methods [120].

To estimate the score function in (6.7), we require the two-step smoothing distribution $p_\theta(x_{t-1:t}|y_{1:T}, u_{1:T})$, which is not available analytically for a general SSM. Instead, we approximate it using an empirical distribution

$$\widehat{p}_\theta(x_{t-1:t}|y_{1:T}, u_{1:T}) = \sum_{i=1}^{N} w_t^{(i)} \delta(x_{t-1:t} - x_{t-1:t}^{(i)}), \tag{6.9}$$

where $x_t^{(i)}$ and $w_t^{(i)}$ denote particle i and its normalized weight at time t. Here, $\{\{x_t^{(i)}, w_t^{(i)}\}_{t=1}^T\}_{i=1}^N$ denotes the particle system generated by a particle filter and $\delta(x)$ is the Dirac delta at $x = 0$.

Following the approach in Chapter 4, here we use the bootstrap particle filter (bPF), presented in Algorithm 4.1 on page 68, combined with the fast forward filtering backwards simulator with early stopping (FFBSi-ES) presented in Algorithm 4.2 on page 69. The FFBSi is used in order to solve the particle degeneracy problem when estimating the smoothing distribution with a particle filter. We refer to Chapter 3 for more details on Sequential Monte Carlo methods.

Modelling the objective function

We explore the use of GP to model the objective function $h(\mathcal{I}_F^{\mathrm{s}}(p_u, \theta_0), \theta_0)$ [155]. GPs can be understood as a generalization of the multivariate Gaussian distribution and are commonly used as priors over functions [22]. In this perspective, the posterior distribution obtained by conditioning on the observations corresponds to a distribution over the functions that could have generated the observations. In the following, we make a brief review of Gaussian processes.

Gaussian processes

The Gaussian process is a model widely used in machine learning, with applications in different fields [174]. Its definition is as follows:

Definition 6.1 *(Gaussian process) The function* $\varphi\colon \mathbb{R}^{n_r} \to \mathbb{R}$ *is said to be a* Gaussian process *if for every positive integer* N *and for any collection of indices* $\{r^{(i)}\}_{i=1}^N$, $r^{(i)} \in \mathbb{R}^{n_r}$, *the distribution of the vector*

$$\boldsymbol{\varphi} := \begin{bmatrix} \varphi(r^{(1)}) \\ \vdots \\ \varphi(r^{(N)}) \end{bmatrix} \tag{6.10}$$

is multivariate Gaussian.

We note that Definition 6.1 is valid for functions on uncountable sets, as r is allowed to adopt any value in $\mathbb{R}^{n_r}$. In addition, it follows from its definition that a Gaussian process is fully determined by specifying its mean and autocovariance functions.

If φ is described by a Gaussian process, we define its mean and autocovariance functions as

$$\mathbf{E}\left\{\varphi(r)\right\} := m(r)\,, \tag{6.11}$$

$$\mathbf{E}\left\{\left(\varphi(r) - m(r)\right)\left(\varphi(\tilde{r}) - m(\tilde{r})\right)\right\} := \kappa(r, \tilde{r})\,, \tag{6.12}$$

where $m\colon \mathbb{R}^{n_r} \to \mathbb{R}$ and $\kappa\colon \mathbb{R}^{n_r} \times \mathbb{R}^{n_r} \to \mathbb{R}$.

In the following, the expression

$$\varphi(\cdot) \sim \mathcal{GP}\left(m(\cdot),\, \kappa(\cdot,\, \cdot)\right) \tag{6.13}$$

denotes that the function $\varphi\colon \mathbb{R}^{n_r} \to \mathbb{R}$ is distributed according to a Gaussian process with mean and autocovariance functions m and κ, respectively. From Definition 6.1, expression (6.13) means that, for every positive integer N and for any collection of indices $\{r^{(i)}\}_{i=1}^{N}$, $r^{(i)} \in \mathbb{R}^{n_r}$, we have

$$\varphi(\mathbf{r}) \sim \mathcal{N}\left(m(\mathbf{r}), \kappa(\mathbf{r}, \mathbf{r})\right)\,, \tag{6.14}$$

where

$$\mathbf{r} := \begin{bmatrix} r^{(1)} \\ \vdots \\ r^{(N)} \end{bmatrix}\,, \tag{6.15}$$

$\varphi(\mathbf{r})$, $m(\mathbf{r})$ denote vectors whose i-th entry is $\varphi(r^{(i)})$ and $m(r^{(i)})$ respectively, and $\kappa(\mathbf{r}, \mathbf{r})$ is a matrix whose (i, j) entry is $\kappa(r^{(i)}, r^{(j)})$, with $r^{(i)} \in \mathbb{R}^{n_r}$, $i, j \in \{1, \ldots, N\}$.

When specifying GP priors over functions, the user has the freedom to define the mean and autocovariance functions. The mean function reflects the prior knowledge (or belief) the user has about the shape of the function before the data is available. On the other hand, the autocovariance function κ encodes additional information about the function, such as smoothness. Standard choices for the autocovariance function are presented in Table 6.1. In this table, $\Gamma\colon \mathbb{R}^{+} \to \mathbb{R}$ is the Gamma function [168, p. 192]

$$\Gamma(x) := \int_0^{\infty} t^{x-1} e^{-t} dt\,. \tag{6.16}$$

To illustrate the effect of the choice of the autocovariance function κ on the GP, we consider the following example:

Table 6.1: Standard choices for the autocovariance function κ, where $d := \sqrt{(r-\tilde{r})^\top (r-\tilde{r})}$ and $p \in \mathbb{N}$ [155].

Name	$\kappa(r, \tilde{r})$	hyperparameters
Squared exponential	$\exp(-d^2/2l^2)$	$l > 0$
Matérn $s/2$	$\exp\left(\frac{\sqrt{s}\,d}{l}\right) \frac{\Gamma((s+1)/2)}{\Gamma(s)} \sum_{i=0}^{(s-1)/2} \frac{((s-1)/2+i)!}{i!((s-1)/2-i)!} \left(\frac{2\sqrt{s}\,d}{l}\right)^{(s-1)/2-i}$	$s = 2p-1$, $l > 0$
γ-exponential	$\exp(-(d/l)^\gamma)$	$0 < \gamma \leq 2$, $l > 0$
Rational quadratic	$\left(1 + \frac{d^2}{2\alpha\, l^2}\right)^{-\alpha}$	$\alpha > 0$, $l > 0$

Example 6.1 *Consider a function* $\varphi\colon [-1,1] \to \mathbb{R}$ *distributed according to the GP*

$$\varphi(\cdot) \sim \mathcal{GP}\left(0,\, \kappa(\cdot,\, \cdot)\right)\,, \tag{6.17}$$

where we consider five different choices for κ*:*

- *Matérn* $1/2$*.*
- *Matérn* $5/2$*.*
- *Squared exponential.*
- *Gamma exponential* ($\gamma = 0.5$)*.*
- *Rational quadratic* ($\alpha = 1$)*.*

For all the choices, the scale parameter is set to $l = 1$*.*

Figure 6.1 presents the $\kappa(x, x')$ *as a function of* $|x - x'|$*, as well as one realization of* φ *from Equation* (6.17)*. From this figure, we see that the choice of* κ *affects the smoothness of the realizations obtained from* (6.17)*. Hence, the choice of the covariance function* κ *significantly affects the properties of the function* φ *distributed according to* (6.17)*.*

Estimation using GP

We conclude this section by illustrating how to use a GP prior to build posterior estimates of a function φ distributed according to (6.14). To this end, assume that we can measure $\varphi(r)$ for all $r \in \mathbb{R}^{n_r}$ as

$$\widehat{\varphi}(r) = \varphi(r) + z\,, \tag{6.18}$$

where $z \sim \mathcal{N}(0, \sigma_z^2)$ is a random variable independent of $\varphi(r)$, for all $r \in \mathbb{R}^{n_r}$.

The objective in this section is to derive the posterior distribution of $\varphi(r)$, $r \in \mathbb{R}^{n_r}$ given $\widehat{\varphi}(\mathbf{r}^{(k)}) := \begin{bmatrix}\widehat{\varphi}(r^{(1)}) & \cdots & \widehat{\varphi}(r^{(k)})\end{bmatrix}^\top$ and $\mathbf{r}^{(k)} := \begin{bmatrix}[r^{(1)}]^\top & \cdots & [r^{(k)}]^\top\end{bmatrix}^\top$, with $k \in \mathbb{N}$. By using the definition of Gaussian process and the expression (6.18), the joint distribution for $\varphi(r)$ and $\widehat{\varphi}(\mathbf{r}^{(k)})$ is given by

$$\begin{bmatrix}\varphi(r) \\ \widehat{\varphi}(\mathbf{r}^{(k)})\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}m(r) \\ m(\mathbf{r}^{(k)})\end{bmatrix}, \begin{bmatrix}\kappa(r,r) & \kappa(r,\mathbf{r}^{(k)}) \\ \kappa(\mathbf{r}^{(k)},r) & \kappa(\mathbf{r}^{(k)},\mathbf{r}^{(k)}) + \sigma_z^2\mathbf{I}_k\end{bmatrix}\right)\,, \tag{6.19}$$

where $\kappa(r, \mathbf{r}^{(k)})$ denotes a row vector whose i-th entry is $\kappa(r, r^{(i)})$, $i \in \{1, \ldots, k\}$, $\kappa(\mathbf{r}^{(k)}, r) = \kappa(r, \mathbf{r}^{(k)})^\top$, and $\mathbf{I}_k$ denotes the $k \times k$ identity matrix. The term $\sigma_z^2\mathbf{I}_k$ in Equation (6.19) comes from the autocovariance function of the measurements $\widehat{\varphi}(\mathbf{r}^{(k)})$, given by (6.18).

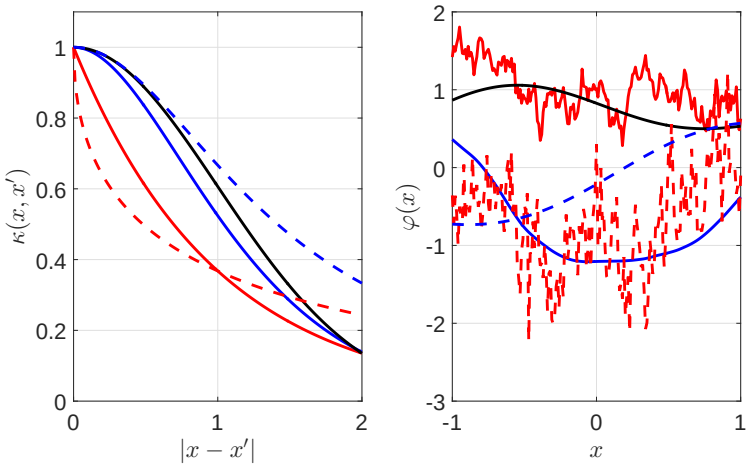

Figure 6.1: Left: Plot of the autocovariance functions κ in Table 6.1 for the hyperparameters specified in Example 6.1. Right: One realization of φ from $\mathcal{GP}(0, \kappa(\cdot,\cdot))$ for the 5 choices of κ in Example 6.1. Red, continuous line: Matérn 1/2. Blue, continuous line: Matérn 5/2. Black, continuous line: Squared exponential. Red, dashed line: γ-exponential. Blue, dashed line: Rational quadratic.

By using standard computations for conditional distributions of Gaussian distributed random variables and (6.19), the posterior distribution of $\varphi(r)$ given $\mathcal{Z}_k := \{\mathbf{r}^{(k)}, \widehat{\varphi}(\mathbf{r}^{(k)})\}$ is

$$\varphi(r)|\mathcal{Z}_k \sim \mathcal{N}\left(\mu(r|\mathcal{Z}_k), \sigma^2(r|\mathcal{Z}_k)\right) , \tag{6.20}$$

where

$$\mu(r|\mathcal{Z}_k) := m(r) + \kappa(r, \mathbf{r}^{(k)})\Sigma^{-1}\left\{\widehat{\varphi}(\mathbf{r}^{(k)}) - m(\mathbf{r}^{(k)})\right\} \tag{6.21a}$$

$$\sigma^2(r|\mathcal{Z}_k) := \kappa(r, r) - \kappa(r, \mathbf{r}^{(k)})\Sigma^{-1}\kappa(\mathbf{r}^{(k)}, r) . \tag{6.21b}$$

with $\Sigma := \kappa(\mathbf{r}^{(k)}, \mathbf{r}^{(k)}) + \sigma_z^2\mathbf{I}_k$.

The following example illustrates the idea presented in this section:

Example 6.2 *Consider the function $\varphi\colon [-1, 1] \to \mathbb{R}$, defined as*

$$\varphi(r) = \sin(\pi r) . \tag{6.22}$$

The objective in this example is to compute the predictive posterior distribution of φ given $\mathcal{Z}_3$ and using the GP prior

$$\varphi(\cdot) \sim \mathcal{GP}(0, \kappa(\cdot, \cdot)) , \tag{6.23}$$

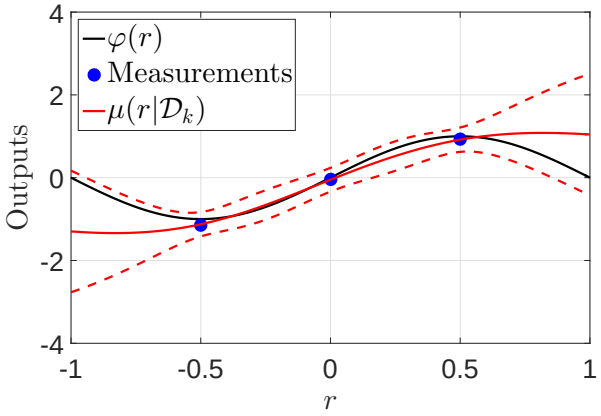

Figure 6.2: Plot of the function $\varphi(\cdot)$, the measurements $\widehat{\varphi}(\mathbf{r}^{(3)})$, and the posterior mean $\mu(r|\mathcal{Z}_3)$, Example 6.2. The figure also presents the confidence interval $[\mu(r|\mathcal{Z}_3) - 3\sigma(r|\mathcal{Z}_3), \mu(r|\mathcal{Z}_3) + 3\sigma(r|\mathcal{Z}_3)]$ as the region enclosed by the red dashed lines.

where $\kappa\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ *is a Matérn* $5/2$ *autocovariance function. The measurements of the function* (6.22) *are collected via* (6.18) *at* $r^{(1)} = -0.5$, $r^{(2)} = 0$, $r^{(3)} = 0.5$, *with* $\sigma_z^2 = 10^{-2}$.

Figure 6.2 presents the function φ, *as well as the measurements* $\widehat{\varphi}(\mathbf{r}^{(3)})$, *and the posterior mean* $\mu(r|\mathcal{Z}_3)$. *The figure also presents the functions* $\mu(r|\mathcal{Z}_3) + 3\sigma(r|\mathcal{Z}_3)$ *and* $\mu(r|\mathcal{Z}_3) - 3\sigma(r|\mathcal{Z}_3)$ *(red, dashed lines). We see that the predictive posterior distribution has higher accuracy in the interval* $[-0.5, 0.5]$, *and that the posterior mean is close to* φ *on that interval. This follows naturally from the fact that the samples* $\mathbf{r}^{(3)}$ *lie in this interval, which gives information on the value of the function on this interval.*

We also observe that the posterior mean does not coincide with φ *outside* $[-0.5, 0.5]$, *where there is no information about* φ. *This is also reflected in the confidence interval, which increases because of the lack of information in* $[-1, -0.5) \cup (0.5, 1]$.

Gaussian processes in input design

Based on the previous subsection, we model the function $h(\mathcal{I}_F^s(\cdot, \theta_0), \theta_0)$ as being a priori distributed according to a GP. That is

$$h(\mathcal{I}_F^s(\cdot, \theta_0), \theta_0) \sim \mathcal{GP}\left(m(\cdot),\, \kappa(\cdot, \cdot)\right), \tag{6.24}$$

where the process is fully described by the mean function $m(\cdot)$ and the autocovariance function $\kappa(\cdot, \cdot)$. We note that the mean and autocovariance functions are defined

over the vector parameterizing p_u. In the following, $m(p_u^{(i)})$ and $\kappa(p_u^{(i)}, p_u^{(j)})$ denote the functions m and κ evaluated at the vectors parameterizing $p_u^{(i)}$ and $p_u^{(j)}$ in $\mathcal{P}$.

To simplify the discussion, we will focus on a specific iteration k of the proposed procedure. Let $\mathcal{Z}_k := \{\mathbf{p}_u^{(k)}, \widehat{\mathbf{h}}_k\}$ denote a set of iterates, where $\mathbf{p}_u^{(k)}$ and $\widehat{\mathbf{h}}_k$ denote matrices obtained by stacking the samples from $\mathcal{P}$ and estimates of the objective function up to iteration k, respectively. In addition, we will assume that

$$\widehat{h}_k = h(\mathcal{I}_F^s(p_u^{(k)}, \theta_0), \theta_0) + z_k \,, \tag{6.25}$$

where $z_k \sim \mathcal{N}(0, \sigma_z^2)$, and $\sigma_z > 0$. We note that σ_z is unknown a priori, and it needs to be estimated using $\mathcal{Z}_k$.

The assumption (6.25) seems strict, but the continuous mapping theorem [16, Theorem 2.7] shows that the central limit theorem also applies to the estimate $\widehat{h}_k$, as it is satisfied by the estimates of $\log p_{\theta_0}(y_{1:T}|u_{1:T})$ asymptotically in the number of particles:

Theorem 6.1 *Let $\log \widehat{p}_{\theta_0}(y_{1:T}|u_{1:T})$ denote the particle estimate of the log likelihood function $\log p_{\theta_0}(y_{1:T}|u_{1:T})$ based on M particles, and let $\mathcal{H}$ be the mapping from $\log p_{\theta_0}(y_{1:T}|u_{1:T})$ to $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$. Assume that $\mathcal{H}$ is measurable on the probability space for $\log \widehat{p}_{\theta_0}(y_{1:T}|u_{1:T})$ and that its discontinuities lie in a set with zero probability. If*

$$\sqrt{M}\left(\log \widehat{p}_{\theta_0}(y_{1:T}|u_{1:T}) - \log p_{\theta_0}(y_{1:T}|u_{1:T})\right) \xrightarrow{d} \mathcal{N}(0, \sigma_p^2) \tag{6.26}$$

holds for some $0 < \sigma_p < \infty$, then

$$\sqrt{M}\left(h(\widehat{\mathcal{I}}_F^s(p_u, \theta_0), \theta_0) - h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)\right) \xrightarrow{d} \mathcal{N}(0, \sigma_h^2) \tag{6.27}$$

holds for some $\sigma_h > 0$.

Proof *Follows from the continuous mapping theorem [16, Theorem 2.7].* ■

To illustrate the continuous mapping theorem in our context, we consider the following example:

Example 6.3 *Consider*

$$x_{t+1}|x_t \sim \mathcal{N}\Big(\phi\, x_t + u_t, 0.1^2\Big), \tag{6.28a}$$

$$y_t|x_t \sim \mathcal{N}\Big(\alpha\, x_t, 0.1^2\Big), \tag{6.28b}$$

where the parameters are $\theta = \{\phi, \alpha\}$. We generate $T = 10^3$ observations from (6.28) with $\theta_0 = \{0.8, 1\}$.

We are interested in estimating $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0) = -\log\det(\mathcal{I}_F^s(p_u, \theta_0))$, where $\{u_t\}$ is a binary white noise process with values $\{-1, 1\}$.

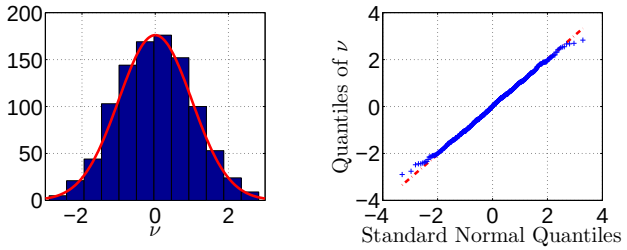

Figure 6.3: Left: Histogram of ν and plot of the scaled pdf of an $\mathcal{N}(0, 1^2)$ distribution (continuous line), Example 6.3. Right: Quantile-quantile plot of the samples of ν and the $\mathcal{N}(0, 1^2)$ distribution, Example 6.3.

An estimate of the Fisher information matrix is obtained using Algorithms 4.1-4.2, with $N = 2.5 \cdot 10^3$ *particles,* $M = 100$ *backward trajectories and* $N_{\text{limit}} = \lfloor \sqrt{N} \rfloor$ *in the fFFBSi smoother. Figure 6.3 shows a histogram based on* 10^3 *realizations of the random variable*

$$\nu := \frac{\sqrt{M}(\widehat{h} - \overline{h})}{\sigma_{\sqrt{M}\widehat{h}}}, \tag{6.29}$$

where $\widehat{h} := h(\widehat{\mathcal{I}}_F^s(p_u, \theta_0), \theta_0)$*, and* $\overline{h}$*,* $\sigma^2_{\sqrt{M}\widehat{h}}$ *are the sample mean of* $\widehat{h}$ *and variance of* $\sqrt{M}\,\widehat{h}$*, respectively. As a comparison, we also present the scaled pdf of an* $\mathcal{N}(0, 1^2)$ *distribution. We can see that the histogram follows the shape of the pdf of an* $\mathcal{N}(0, 1^2)$ *distribution. This is also confirmed by the quantile-quantile (QQ) plot in Figure 6.3, where the quantiles of* ν *coincides with those given by an* $\mathcal{N}(0, 1^2)$ *distribution.*

Based on (6.25), it follows that the predictive posterior distribution is

$$h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0) | \mathcal{Z}_k \sim \mathcal{N}\left(\mu(p_u|\mathcal{Z}_k),\, \sigma^2(p_u|\mathcal{Z}_k)\right), \tag{6.30}$$

where $\mu(p_u|\mathcal{Z}_k)$ and $\sigma^2(p_u|\mathcal{Z}_k)$ denote the posterior mean and variance given $\mathcal{Z}_k$. From (6.21), we have

$$\mu(p_u|\mathcal{Z}_k) = m(p_u) + \kappa(p_u,\, \mathbf{p}_u^{(k)})\Sigma^{-1}\left\{\widehat{\mathbf{h}}_k - m(\mathbf{p}_u^{(k)})\right\}, \tag{6.31a}$$

$$\sigma^2(p_u|\mathcal{Z}_k) = \kappa(p_u,\, p_u) - \kappa(p_u,\, \mathbf{p}_u^{(k)})\Sigma^{-1}\kappa(\mathbf{p}_u^{(k)},\, p_u), \tag{6.31b}$$

with $\Sigma := \kappa(\mathbf{p}_u^{(k)}, \mathbf{p}_u^{(k)}) + \sigma_z^2 \mathbf{I}_k$. The function $\kappa(\mathbf{p}_u^{(k)}, \mathbf{p}_u^{(k)})$ denotes a matrix for which its (i, j) entry is given by $\kappa(p_u^{(i)}, p_u^{(j)})$.

In the GP model introduced here, we use mean and autocovariance functions that possibly depend on some unknown hyperparameters. In addition, we also need to estimate σ_z characterizing the random variable z in (6.25). To estimate these quantities, we adopt the empirical Bayes procedure, which is briefly explained in the next subsection.

Empirical Bayes

In the literature, the term *empirical Bayes* has many different interpretations [172, p. 325]. In the context of this chapter, by empirical Bayes we refer to maximizing the marginal likelihood of the data with respect to the hyperparameters characterizing the prior distribution and the likelihood function [33].

To illustrate the use of marginal likelihood when we specify priors over functions, we assume we have N_p evaluations of $h(\mathcal{I}_F^s(\cdot,\theta_0),\theta_0)$, which are stacked to define the vector

$$\mathbf{h}^{N_p} := \begin{bmatrix} h(\mathcal{I}_F^s(p_u^{(1)},\theta_0),\theta_0) & \cdots & h(\mathcal{I}_F^s(p_u^{(N_p)},\theta_0),\theta_0) \end{bmatrix}^\top , \tag{6.32}$$

where

$$\mathbf{p}_u^{(N_p)} = \begin{bmatrix} p_u^{(1)} & \cdots & p_u^{(N_p)} \end{bmatrix}^\top , \tag{6.33}$$

is a matrix containing the vectors parameterizing the points in $\mathcal{P}$ on which $h(\mathcal{I}_F^s(\cdot,\theta_0),\theta_0)$ is evaluated. Since $h(\mathcal{I}_F^s(\cdot,\theta_0),\theta_0)$ has a GP prior, then

$$\mathbf{h}^{N_p} \sim \mathcal{N}\left(m_\eta(\mathbf{p}_u^{(N_p)}), \kappa_\eta(\mathbf{p}_u^{(N_p)},\mathbf{p}_u^{(N_p)}) \right) , \tag{6.34}$$

where $m_\eta(\mathbf{p}_u^{(N_p)})$ denotes a vector for which the i-th entry is $m_\eta(p_u^{(i)})$, $i \in \{1,\ldots,N_p\}$. Note that we allow m_η and κ_η to be functions of the vector of hyperparameters $\eta \in \mathbb{R}^{n_\eta}$.

Defining $\tau \in \mathbb{R}^{n_\eta+1}$ by

$$\tau := \begin{bmatrix} \sigma_z & \eta^\top \end{bmatrix}^\top , \tag{6.35}$$

as the vector of hyperparameters characterizing the likelihood function and the prior distribution, the marginal likelihood given the observations $\widehat{\mathbf{h}}_{N_p}$ of $\mathbf{h}^{N_p}$ is the integral of the likelihood function times the prior:

$$p_\tau(\widehat{\mathbf{h}}_{N_p}) = \int p_{\sigma_z}(\widehat{\mathbf{h}}_{N_p}|\mathbf{h}^{N_p})\, \pi_\eta(\mathbf{h}^{N_p}) d\mathbf{h}^{N_p} , \tag{6.36}$$

where π_η is the pdf associated with the distribution (6.34).

Following (6.25), we have that

$$\widehat{\mathbf{h}}_{N_p}|\mathbf{h}^{N_p} \sim \mathcal{N}(\mathbf{h}^{N_p}, \sigma_z^2\, \mathbf{I}_{N_p\times N_p}) . \tag{6.37}$$

Since the prior for $\mathbf{h}^{N_p}$ and the distribution of $\widehat{\mathbf{h}}_{N_p}$ given $\mathbf{h}_{N_p}$ are Gaussian distributed, then it can be shown that the product of the pdfs inside the integral in (6.36) is an un-normalized Gaussian pdf [155, Equation (A.7)]. In fact,

$$p_{\sigma_z}(\widehat{\mathbf{h}}_{N_p}|\mathbf{h}^{N_p})\, \pi_\eta(\mathbf{h}^{N_p}) = K(\mathbf{h}^{N_p},\sigma_z^2,\eta)\, \tilde{p}_\tau(\widehat{\mathbf{h}}_{N_p}|\mathbf{h}^{N_p}) , \tag{6.38}$$

where $\tilde{p}_\tau(\widehat{\mathbf{h}}_{N_p}|\mathbf{h}^{N_p})$ is the pdf of a Gaussian distribution with mean

$$\left(\frac{1}{\sigma_z^2}\mathbf{I}_{N_p} + \kappa_\eta(\mathbf{p}_u^{(N_p)}, \mathbf{p}_u^{(N_p)})^{-1}\right)^{-1}\left(\frac{1}{\sigma_z^2}\mathbf{h}^{N_p} + \kappa_\eta(\mathbf{p}_u^{(N_p)}, \mathbf{p}_u^{(N_p)})^{-1} m_\eta(\mathbf{p}_u^{(N_p)})\right), \tag{6.39}$$

and covariance matrix

$$\left(\frac{1}{\sigma_z^2}\mathbf{I}_{N_p} + \kappa_\eta(\mathbf{p}_u^{(N_p)}, \mathbf{p}_u^{(N_p)})^{-1}\right)^{-1}. \tag{6.40}$$

In addition, the constant $K(\mathbf{h}^{N_p}, \sigma_z^2, \tau)$ in (6.38) is given by

$$K(\mathbf{h}^{N_p}, \sigma_z^2, \eta) = \frac{1}{(2\pi)^{N_p/2}} \det M^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{h}^{N_p} - m_\eta(\mathbf{p}_u^{(N_p)}))^\top M^{-1}(\mathbf{h}^{N_p} - m_\eta(\mathbf{p}_u^{(N_p)}))\right\}, \tag{6.41}$$

with

$$M := \sigma_z^2\mathbf{I}_{N_p} + \kappa_\eta(\mathbf{p}_u^{(N_p)}, \mathbf{p}_u^{(N_p)}). \tag{6.42}$$

Based on (6.39)-(6.42), the logarithm of the marginal likelihood in (6.36) is

$$\log p_\tau(\widehat{\mathbf{h}}_{N_p}) = -\frac{1}{2}(\widehat{\mathbf{h}}_{N_p} - m_\eta(\mathbf{p}_u^{(N_p)}))^\top M^{-1}(\widehat{\mathbf{h}}_{N_p} - m_\eta(\mathbf{p}_u^{(N_p)})) - \frac{1}{2}\log\det(M) - \frac{N_p}{2}\log 2\pi. \tag{6.43}$$

Hence, the hyperparameter vector $\widehat{\tau}$ to be employed in the GPO framework is computed as

$$\widehat{\tau} = \arg\max_{\tau\in\mathbb{R}^{n_\eta+1}} \log p_\tau(\widehat{\mathbf{h}}_{N_p}), \tag{6.44}$$

where $\log p_\tau(\widehat{\mathbf{h}}_{N_p})$ is given by (6.43).

A closer inspection to (6.43) reveals that, for the particular problem addressed in this chapter, the logarithm of the marginal likelihood is maximized by choosing (σ_z^2, η) such that $\widehat{\mathbf{h}}_{N_p} - m_\eta(\mathbf{p}_u^{(N_p)})$ and the eigenvalues of $\sigma_z^2\mathbf{I}_{N_p} + \kappa_\eta(\mathbf{p}_u^{(N_p)}, \mathbf{p}_u^{(N_p)})$ are close to zero. This means that the GP prior evaluated at $\widehat{\tau}$ makes a trade-off between the fit to the observed data $\widehat{\mathbf{h}}_{N_p}$ and the complexity of the model, quantified in terms of the eigenvalues of the covariance matrix $\sigma_z^2\mathbf{I}_{N_p} + \kappa_\eta(\mathbf{p}_u^{(N_p)}, \mathbf{p}_u^{(N_p)})$.

Acquisition rules

To implement step (iii), we need to generate $p_u^{(k+1)} \in \mathcal{P}$. One option is to perform a random walk over $\mathcal{P}$, which works well provided that the parameterization of $\mathcal{P}$ is of small dimension. However, this approach is inefficient as the dimension of the parameterization for $\mathcal{P}$ increases.

Instead, we make use of *acquisition rules* that balance exploration and exploitation of the parameter space and employ the posterior distribution obtained from the GP. The objective of acquisition rules (or selection strategies) is to guide the sequential search over $\mathcal{P}$, i.e., by determining the next sample $p_u^{(k+1)} \in \mathcal{P}$ given $\mathcal{Z}_k$. A critical requirement of acquisition rules is that they should be cheaper to evaluate (or approximate) than evaluating the cost function $h(\mathcal{I}_F^s(\cdot, \theta_0), \theta_0)$. This holds for most acquisition functions [174, p. 150].

There exists a vast literature on selection strategies, and we refer to [174] for a more detailed discussion in this subject. In the context of this chapter, we use the expected improvement (EI) technique [104] to sequentially sample over $\mathcal{P}$.

The expected improvement approach involves computing how much improvement we expect to achieve if we sample at a given point. Intuitively speaking, if the current best function value predicted is $\mu_{\min}$, then the predicted improvement at a point $p_u \in \mathcal{P}$ is defined as

$$I(p_u) := \max\left\{0,\ -h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0) + \mu_{\min} + \xi\right\}, \tag{6.45}$$

where $\xi \in \mathbb{R}$ is a predefined parameter, and

$$\mu_{\min} := \min_{p_u \in \mathbf{p}_u^{(k)}} \mu(p_u | \mathcal{Z}_k), \tag{6.46}$$

is the expected minimum of $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$ at iteration k.

Remark 6.2 *The parameter ξ in* (6.45) *establishes a trade-off between exploration and exploitation of the method. If $\xi < 0$, then $I(p_u)$ is greater than zero if and only if $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0) - \xi < \mu_{\min}$, i.e., the predicted improvement can be zero even if $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0) < \mu_{\min}$. Thus, $\xi < 0$ promotes exploitation of the value of $p_u \in \mathbf{p}_u^{(k)}$ attaining $\mu_{\min}$. The same reasoning can be reversed to establish that $\xi > 0$ promotes exploration of the set $\mathcal{P}$, as the predicted improvement can be nonzero even if $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0) > \mu_{\min}$.*

By using the posterior distribution obtained from the GP, the EI $\mathbf{E}\{I(p_u)|\mathcal{Z}_k\}$ can be computed using integration by parts as [104, p. 371]

$$\mathbf{E}\{I(p_u)|\mathcal{Z}_k\} = \sigma(p_u|\mathcal{Z}_k)\left\{L(p_u|\mathcal{Z}_k)\Phi(L(p_u|\mathcal{Z}_k))\ -\phi(L(p_u|\mathcal{Z}_k))\right\}, \tag{6.47a}$$

$$L(p_u|\mathcal{Z}_k) := \sigma^{-1}(p_u|\mathcal{Z}_k)\left\{-\mu(p_u|\mathcal{Z}_k) + \mu_{\min} + \xi\right\}, \tag{6.47b}$$

with Φ and ϕ denoting the cumulative distribution function and the pdf of the standard Gaussian distribution, respectively. Then, the acquisition rule based on EI becomes

$$p_u^{(k+1)} = \arg\max_{p_u \in \mathcal{P}} \mathbf{E}\left\{I(p_u)\,|\mathcal{Z}_k\right\}, \tag{6.48}$$

i.e., the element maximizing the EI. From (6.47) we see that the EI assigns a large value when both the variance $\sigma^2(p_u|\mathcal{Z}_k)$ and the mean difference $-\mu(p_u|\mathcal{Z}_k) + \mu_{\min}$ are large, in line with the desired behavior of an acquisition function, as explained at the beginning of Section 6.3.

Parameterizing the input

To implement the GPO for solving the input design problem, we need a parameterization of $\mathcal{P}$. Here we explain two options.

Stationary Markov processes

If we restrict $\mathcal{C}$ to be finite and $u_{1:T}$ to be realization from an n-dimensional stationary Markov process, then the parameterization employed in [190] can be used, which is presented in Chapter 2. The parameterization of the input is given by the stationary distribution of the Markov process, which is constrained to

$$\mathcal{P}_{\mathcal{C}} := \left\{ p_u : \mathcal{C}^n \to \mathbb{R} \,\middle|\, p_u(\mathbf{x}) \geq 0,\, \forall \mathbf{x} \in \mathcal{C}^n; \right. \\ \sum_{\mathbf{x} \in \mathcal{C}^n} p_u(\mathbf{x}) = 1; \\ \left. \sum_{v \in \mathcal{C}} p_u(v,\, \mathbf{z}) = \sum_{v \in \mathcal{C}} p_u(\mathbf{z},\, v)\,, \forall \mathbf{z} \in \mathcal{C}^{n-1} \right\}. \quad (6.49)$$

Following Chapter 2, we parameterize (6.49) as the convex hull of its extreme points, which are computed using graph theoretical techniques. Therefore, the decision variable in this case corresponds to the weighting vector of the extreme points describing an element in $\mathcal{P}_{\mathcal{C}}$. Assuming that $\mathcal{P}_{\mathcal{C}}$ has $n_{\mathcal{V}}$ extreme points, then the weighting vector $\alpha := [\alpha_1 \quad \dots \quad \alpha_{n_{\mathcal{V}}}]^\top \in \mathbb{R}^{n_{\mathcal{V}}}$ is used to compute $p \in \mathcal{P}_{\mathcal{C}}$ as

$$p = \sum_{i=1}^{n_{\mathcal{V}}} \alpha_i \, p^{(i)}\,, \quad (6.50)$$

with α satisfying

$$\alpha_i \geq 0\,, \text{ for all } i \in \{1\,, \dots,\, n_{\mathcal{V}}\}\,, \quad (6.51a)$$

$$\sum_{i=1}^{n_{\mathcal{V}}} \alpha_i = 1\,. \quad (6.51b)$$

In (6.50), $\{p^{(i)}\}_{i=1}^{n_{\mathcal{V}}}$ corresponds to the probability mass functions (pmf) that are the vertices of $\mathcal{P}_{\mathcal{C}}$.

Once a new sample $\alpha \in \mathbb{R}^{n_{\mathcal{V}}}$ satisfying (6.51) is generated, we compute the associated pmf $p \in \mathcal{P}_{\mathcal{C}}$ by (6.50), and we generate $u_{1:T}$ by running a Markov chain with stationary distribution p. The sample α is computed at every iteration k by a random walk centered at the parametrization of the solution of (6.48). We refer to Chapter 2 for more details about the graph theoretical parameterization.

Algorithm 6.1 GPO for input design

INPUTS: Algorithm 4.2, K (no. iterations) and $p_u^{(0)} \in \mathcal{P}$ (initial pdf characterizing $u_{1:T}$).
OUTPUT: A realization $u_{1:T}^{\text{opt}}$ from p_u^{opt}.

1: Sample $p_u^{(0)} \in \mathcal{P}$.
2: **for** $k = 0$ to K **do**
3: Use Algorithm 4.2 to compute $\widehat{h}_k := h(\widehat{\mathcal{I}}_F^s(p_u^{(k)}, \theta_0), \theta_0)$.
4: Compute (6.30)-(6.31) to obtain $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)|\mathcal{Z}_k$.
5: Compute (6.46) to obtain $\mu_{\min}$.
6: Compute (6.48) to obtain $\tilde{p}_u^{(k+1)}$.
7: Compute $p_u^{(k+1)}$ as a realization of a random walk centered at $\tilde{p}_u^{(k+1)}$.
8: **end for**
9: Compute the maximizer of $\mu(p_u|\mathcal{Z}_K)$ to obtain p_u^{opt}.

Stationary AR processes

We can restrict $u_{1:T}$ to be filtered white noise, as proposed in [88]. In this case, the decision variables are the filter coefficients, and the properties of the white noise. For example, we can assume that $u_{1:T}$ is a realization from a stationary AR process

$$A(q)\, u_t = e_t\,, \tag{6.52}$$

where $\{e_t\}$ is Gaussian white noise of zero mean and variance σ_e^2, and

$$A(q) := \sum_{i=0}^{n_a} a_i\, q^{-i}\,, \tag{6.53}$$

with $n_a > 0$ given, $a_i \in \mathbb{R}$ for all $i \in \{1, \ldots, n_a\}$, and $a_0 = 1$. For this example, the parameters are $\sigma_e > 0$, and $\{a_i\}_{i=1}^{n_a}$, such that $A(q)$ has all its zeros strictly inside the complex unit disc[1].

The final procedure

Algorithm 6.1 presents the resulting procedure for input design using Gaussian process optimization. We note that line 7 introduces a random walk centered at (6.48) to promote exploration around the expected improvement. We also note that only one functional evaluation is required per iteration, reducing the computational effort with respect to a search over the entire space when optimizing over $\mathcal{P}$.

[1]This can be guaranteed by factorizing $A(q)$ into first and second order polynomials in q, and imposing the constraint on each of these factors.

6.4 Numerical example

Input design for linear Gaussian SSM

Consider the linear Gaussian state space model in Example 6.3. We are interested in minimizing $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0) = -\log\det(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$, where $u_{1:T}$ ($T = 10^3$) is a realization of a stationary Markov process (see Section 6.3), with $n_m = 1$ and $\mathcal{C} = \{-1, 1\}$.

For Algorithm 6.1, we use $K = 500$, $\xi = 0.01$, and a random walk centered around the current parametrization of $\tilde{u}_{1:T}^{(k+1)}$, uniformly distributed on $[-0.01, 0.01]$. The estimate of the Fisher information matrix is obtained using Algorithms 4.1-4.2, which are implemented as in Example 6.3. For the prior distribution of $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$, we consider a constant mean function, and an autocovariance function composed of a Matérn 5/2 structure and a constant. The Matérn 5/2 structure is chosen in this example as it imposes information about the smoothness of $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$.

Algorithm 6.1 is implemented in Matlab using the `fmincon` command for (6.48) and the `GPML` toolbox [154] to infer the hyperparameters and estimate the predictive posterior distribution of $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0)$.

The solution obtained from Algorithm 6.1 is $u_t = 1$ for all $t \geq 0$. In this example, a nonzero constant input introduces a nonzero offset in the measurements, which helps to estimate θ in the presence of process disturbance and measurement noise. As a reference, we draw $u_{1:T}$ as a realization from binary white noise with values $\{-1, 1\}$. The results are $h(\mathcal{I}_F^s(p_u^{\text{opt}}, \theta_0), \theta_0) = -14.57$ for the optimal input and $h(\mathcal{I}_F^s(p_u, \theta_0), \theta_0) = -10.18$ for the binary white noise process. Hence, the GPO algorithm provides an input that improves the experimental results in this example when compared to a binary white noise.

Input design for nonlinear SSM

Consider the system

$$x_{t+1}|x_t \sim \mathcal{N}\Big(\frac{1}{\gamma + x_t^2} + u_t, 0.1^2\Big), \tag{6.54a}$$

$$y_t|x_t \sim \mathcal{N}\Big(\beta x_t^2, 1^2\Big), \tag{6.54b}$$

where the parameters are $\theta = \{\gamma, \beta\}$. We generate $T = 10^3$ observations from the model with $\theta_0 = \{2, 0.8\}$. We note that estimating γ in (6.54) is inherently difficult, since two different values of x_t can explain y_t equally well.

We consider the same setting and function h as in the previous example, but we consider three cases for $\mathcal{C}$:

- Case 1: $\mathcal{C} = \{-1, 1\}$.
- Case 2: $\mathcal{C} = \{-1, 0, 1\}$.

Table 6.2: h^{opt} for different input realizations, NSSM example.

Input	Binary	opt. Case 1	opt. Case 2	opt. Case 3
h^{opt}	-4.11	-4.11	-4.15	-4.44

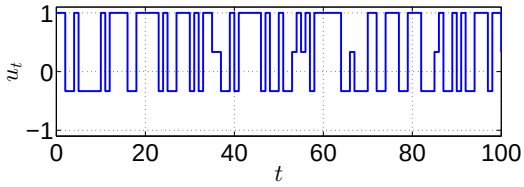

Figure 6.4: Optimal input $u_{1:T}^{\text{opt}}$ for Case 3, NSSM example.

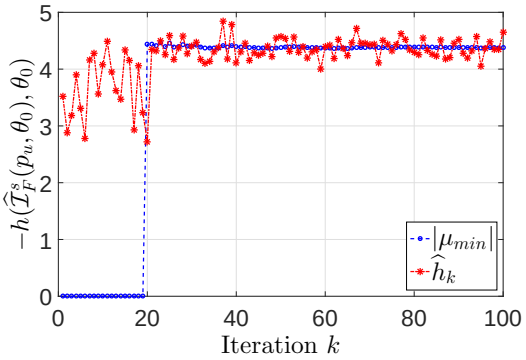

Figure 6.5: Value of $\widehat{h}_k$ and $|\mu_{\min}|$ at iteration k for Case 3 in Example 1.1.

- Case 3: $\mathcal{C} = \{-1, -1/3, 1/3, 1\}$.

Table 6.2 presents the value of $h^{\text{opt}} := h(\widehat{\mathcal{I}}_F^s(p_u^{\text{opt}}, \theta_0), \theta_0)$ for each case, where p_u^{opt} corresponds to the optimal input obtained from Algorithm 6.1. As comparison, we also compute the value of $h(\widehat{\mathcal{I}}_F^s(p_u, \theta_0), \theta_0)$, with $\{u_t\}$ binary distributed white noise with values $\{-1, 1\}$ (Binary in Table 6.2). We see that the binary white noise process seems to be optimal when $\mathcal{C} = \{-1, 1\}$, as it is confirmed by the value of h^{opt} for Case 1. We also note that adding intermediate values to the input alphabet increases the amount of information in the data, as h^{opt} is greater in Cases 2 and 3 than in Case 1.

Figure 6.4 presents the optimal input obtained for Case 3. We note that the optimal input includes a nonzero offset to improve the accuracy of the parameter estimates.

To illustrate the evolution of $\widehat{h}_k$, we present in Figure 6.5 the samples $\{\widehat{h}_k\}_{k=1}^{100}$, together with the value of $|\mu_{\min}|$ at every iteration. The first 20 samples are drawn at random from $\mathcal{C}^T$ to provide an initial estimate of the hyperparameters in the GP prior. We note that some of the samples in $\{\widehat{h}_k\}_{k=1}^{20}$ are not close to the optimal cost, which is expected due to random sampling. However, once Algorithm 6.1 is executed from iteration 21 onwards, we observe that the samples are close to $\mu_{\min}$, which implies that the space $\mathcal{P}$ is explored only in those regions where h can only increase with respect to the current estimates. Hence, the proposed technique drives the parameter search towards those regions where an improvement in the objective function is expected.

6.5 Conclusions

A Gaussian process optimization algorithm for input design for nonlinear dynamical models has been introduced. The method maximizes a scalar cost function of the Fisher information matrix over the parameter set for the input sequence. Since the objective function is unavailable in closed form, a Gaussian process approach is employed to compute a surrogate function. Numerical examples show that the algorithm can provide a good alternative to solving the input design problem.

Chapter 7

On model order priors for Bayesian identification of linear models

In Chapter 6 we discussed how Bayesian optimization can provide a solution to the problem of input design for identification of nonlinear dynamical models. This chapter continues exploring the use of Bayesian methods, but now focused on the estimation of single input-single output (SISO) LTI models.

7.1 Introduction

As mentioned in Chapter 1, most of the Bayesian methods for system identification makes use of prior distributions that are conjugate of the likelihood function [73, 98, 208]. This choice is usually taken since it leads to closed form expressions for the posterior distributions and the intermediate quantities required in the Gibbs sampler [78]. However, this choice restricts the class of distributions to encode prior information about the dynamical model. In particular, imposing a prior distribution on the model complexity becomes difficult in this context, since it is not immediate to determine a prior distribution that is conjugate to the likelihood function which penalizes the model complexity.

Contribution

This chapter presents a new approach for Bayesian identification of single input, single output (SISO) linear dynamical systems, where priors over the model order are imposed. The method computes samples that are approximately distributed according to the posterior probability density function (pdf) of the model parameters. To this end, a Metropolis Hastings sampler is implemented [94, 160]. An advantage of using a Metropolis Hastings sampler when compared with other methods is that it gives more freedom to choose the prior distribution of the model parameters, which allows to define priors that do not necessarily lead to posterior distributions available

in closed form. We exploit this feature by imposing a prior distribution over the Hankel singular values of the model, and hence accounting for prior knowledge or belief about the order of the model.

The problem of accounting for the model complexity in a Bayesian setting has been previously considered in the literature. For example, [91] presents a Markov chain algorithm to sample over model structures with different complexity. However, the main limitation in [91] is the requirement of a differential bijective mapping between model structures of different complexity, which is not directly implementable in the case of dynamical models. The model complexity has also been addressed in [73] by employing an automatic relevance determination (ARD) approach. In this case, the ARD method avoids that the models sampled from the posterior distribution are of high order. Nevertheless, the Bayesian method in [73] does not ensure directly that properties such as stability are preserved in the models sampled from the posterior distribution. Leaving the Bayesian framework, [96] discusses the problem of choosing the appropriate model complexity for controller design. In this context, [96] shows that a well designed experiment can lead to identify restricted complexity models that are near optimal.

Structure of the chapter

This chapter is organized as follows. Section 7.2 introduces the problem formulation. Section 7.3 discusses the Metropolis Hastings sampler. The specification of prior distributions over the parameter set is considered in Section 7.4. The implementation of a random walk over the parameter space is presented in Section 7.5. Section 7.6 illustrates the method with a numerical example. Finally, Section 7.7 presents concluding remarks.

In the following, for a given scalar, rational function G without poles on the circle $\{z \in \mathbb{C}\colon\ |z| = 1\}$, we define its 2-norm as

$$\|G\|_2 := \sqrt{\frac{1}{2\pi} \int_{-\pi}^{\pi} |G(e^{i\omega})|^2 \, d\omega}\,. \tag{7.1}$$

7.2 Problem formulation

Consider the discrete time, single input, single output (SISO) linear time invariant system defined for all $t \geq 1$ as

$$y_t = G_o(q)\, u_t + H_o(q)\, e_t\,, \tag{7.2}$$

where G_o, H_o are scalar real rational functions in the time shift operator q, $u_t \in \mathbb{R}$ is the input, $e_t \in \mathbb{R}$ is white noise with pdf p_e, and $y_t \in \mathbb{R}$ is the measured output. In addition, we consider the following:

Assumption 7.1 *The pair (G_o, H_o) of scalar rational functions in q has all its poles in the set $\mathbb{D} := \{z \in \mathbb{C} \colon |z| < 1\}$. Also, H_o^{-1} has all its poles in $\mathbb{D}$ and*

$$\lim_{q\to\infty} H_o(q) = 1\,. \tag{7.3}$$

Assumption 7.1 is standard in system identification to guarantee that the signals derived from (7.2) are quasi-stationary [122, Definition 2.1]:

Definition 7.1 *(Quasi-stationary signals) A signal $\{s_t\}$ is* quasi-stationary *if there exists functions $m_s \colon \mathbb{Z} \to \mathbb{R}$, $R_s \colon \mathbb{Z} \times \mathbb{Z} \to \mathbb{R}$, $\tilde{R}_s \colon \mathbb{Z} \to \mathbb{R}$ and a positive constant $C < \infty$ such that, for all $t, r, \tau \in \mathbb{Z}$,*

$$\mathbf{E}\,\{s_t\} = m_s(t)\,,\ |m_s(t)| \leq C\,, \tag{7.4a}$$

$$\mathbf{E}\,\{s_t s_r\} = R_s(t,r)\,,\ |R_s(t,r)| \leq C\,, \tag{7.4b}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} R_s(t, t-\tau) = \tilde{R}_s(\tau)\,, \tag{7.4c}$$

Our aim is to estimate the parameters $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ in the model

$$y_t = G_\theta(q)\, u_t + H_\theta(q)\, e_t^\theta\,, \tag{7.5}$$

given the observed data set $\mathbf{Z}_T := \{(y_t, u_t)\}_{t=1}^T$. We note that the white noise sequence e_t^θ is parameterized through its pdf, denoted by p_e^θ. In the following, we assume that the model (7.5) satisfies Assumption 7.1 for all $\theta \in \Theta$ (with (G_θ, H_θ) instead of (G_o, H_o)). In addition, we assume that there exists a parameter $\theta_0 \in \Theta$ such that the model (7.5) describes the system (7.2) when $\theta = \theta_0$, i.e., there is no undermodelling.

Here we are interested in exploring a Bayesian approach to estimate $\hat{\theta}_T \in \Theta$ describing the system (7.2). In this setting, we require the specification of a prior distribution over Θ, which reflects the available knowledge (or belief) that the user has about θ_0 before observing the data set $\mathbf{Z}_T$. In the following, we denote by π_Θ the pdf associated with the prior distribution over Θ.

Given π_Θ and the data set $\mathbf{Z}_T$, the posterior distribution over Θ can be computed using Bayes' rule as

$$p(\theta|\mathbf{Z}_T) = \frac{1}{K_\Theta}\, p_\theta(y_{1:T}|u_{1:T})\, \pi_\Theta(\theta)\,, \tag{7.6}$$

where

$$K_\Theta := \int_\Theta p_\theta(y_{1:T}|u_{1:T})\, \pi_\Theta(\theta)\, d\theta \tag{7.7}$$

is a normalizing constant. In (7.6), the expression $p_\theta(y_{1:T}|u_{1:T})$ denotes the likelihood of $\mathbf{Z}_T$.

Once the pdf (7.6) is computed, there are several options to obtain an estimate of the model parameters $\hat{\theta}_T$. The most commonly used ones are

$$\hat{\theta}_T = \arg\max_{\theta\in\Theta} p(\theta|\mathbf{Z}_T)\,, \tag{7.8}$$

referred to as the maximum a posteriori estimator, and

$$\hat{\theta}_T = \mathbf{E}\left\{\theta|\mathbf{Z}_T\right\}\,, \tag{7.9}$$

the conditional mean of the model parameters given the observed data set $\mathbf{Z}_T$.

Remark 7.1 *We note that the maximum a posteriori estimator* (7.8) *can be obtained by maximizing the product* $p_\theta(y_{1:T}|u_{1:T})\,\pi_\Theta(\theta)$. *Indeed the optimization in* (7.8) *can be performed directly in a numerical fashion. However, this approach only delivers the maximizer of* $p(\theta|\mathbf{Z}_T)$ *and not the uncertainty of this estimator.*

The key step to estimate the model parameters in a Bayesian setting is the computation of (7.6). Unfortunately, obtaining such an expression in closed form is often difficult in practice, which has been the main limitation of this approach for a long time.

In an effort to compute expressions like (7.6), Markov chain Monte Carlo (MCMC) algorithms for approximating pdfs have received considerable attention, and methods to obtain estimates for the posterior pdf (7.6) have been developed and analyzed. In particular, the Gibbs sampler has been mainly employed to generate an estimate of the posterior distribution [34, 73, 208]. The main limitation of this approach is that the prior cannot be arbitrarily chosen, since it must allow to sample from conditional pdfs derived from the posterior distribution. To achieve this, the prior is usually chosen as the conjugate of the likelihood out of mathematical convenience [170]. In case the user wants to specify a prior that is not conjugate to the likelihood, then the Gibbs sampler approach in [73, 208] cannot be directly utilized to approximate the posterior distribution. For example, this is typically the case if the user specifies a prior over the order of the model (7.5).

An alternative to the Gibbs sampler is provided by its ancestor, the Metropolis Hastings method, where the assumption of conjugate distributions can be relaxed. In the next section we describe the Metropolis Hastings algorithm and its application to compute the posterior pdf (7.6).

7.3 The Metropolis Hastings sampler

The Metropolis Hastings (MH) sampler is an MCMC method that generates samples approximately distributed according to a prescribed pdf Υ, called the target distribution. The samples are obtained by running a Markov chain with appropriate transition kernels, ensuring that Υ is the stationary pdf associated with the chain [170]. In the context of this chapter, the target distribution is $\Upsilon(\theta) = p(\theta|\mathbf{Z}_T)$ (i.e., the posterior pdf of the parameter vector).

Algorithm 7.1 Metropolis Hastings sampler

INPUTS: Target pdf Υ and proposal distribution f.
OUTPUT: $\{\theta^{(i)}\}_{i=1}^{N_s}$ (samples distributed approximately according to Υ).

1: Set the initial state of the Markov chain $\theta^{(1)}$.
2: **for** $i = 1$ to N_s **do**
3: Sample $\theta^\star \sim f(\theta|\theta^{(i)})$.
4: Sample $u \sim \mathcal{U}[0,1]$.
5: Compute the acceptance probability

$$a(\theta^\star, \theta^{(i)}) = \min\left(1, \frac{\Upsilon(\theta^\star)\, f(\theta^{(i)}|\theta^\star)}{\Upsilon(\theta^{(i)})\, f(\theta^\star|\theta^{(i)})}\right). \tag{7.11}$$

6: Set the next state $\theta^{(i+1)}$ of the Markov chain according to

$$\theta^{(i+1)} = \begin{cases} \theta^\star, & u \leq a(\theta^\star, \theta^{(i)}) \\ \theta^{(i)}, & \text{otherwise.} \end{cases} \tag{7.12}$$

7: **end for**

The idea behind the MH sampler is as follows: starting from an initial state $\theta^{(1)} \in \Theta$, we sample a new candidate $\theta^\star \in \Theta$ from a user defined conditional pdf $f(\theta|\theta^{(1)})$, called the proposal distribution. Then, with probability

$$a(\theta^\star, \theta^{(1)}) := \min\left(1, \frac{\Upsilon(\theta^\star)\, f(\theta^{(1)}|\theta^\star)}{\Upsilon(\theta^{(1)})\, f(\theta^\star|\theta^{(1)})}\right) \tag{7.10}$$

we define the next state as $\theta^{(2)} = \theta^\star$, otherwise we keep the previous state of the chain, i.e., $\theta^{(2)} = \theta^{(1)}$. This procedure is repeated until the required number of samples have been generated.

Algorithm 7.1 presents the details of the MH sampler. Under the assumption that the support of the proposal pdf f includes the support of the target distribution Υ, it can be proven that Algorithm 7.1 has Υ as a stationary distribution of the chain [170].

By using $\Upsilon(\theta) = p(\theta|\mathbf{Z}_T)$ and (7.6), the acceptance probability for the MH sampler to generate samples from $p(\theta|\mathbf{Z}_T)$ is

$$a(\theta^\star, \theta^{(i)}) = \min\left(1, \frac{p_{\theta^\star}(y_{1:T}|u_{1:T})\, \pi_\Theta(\theta^\star)\, f(\theta^{(i)}|\theta^\star)}{p_{\theta^{(i)}}(y_{1:T}|u_{1:T})\, \pi_\Theta(\theta^{(i)})\, f(\theta^\star|\theta^{(i)})}\right). \tag{7.13}$$

From (7.13), we see that the acceptance probability only needs the knowledge of the posterior pdf up to a normalization constant.

On the other hand, the likelihood $p_\theta(y_{1:T}|u_{1:T})$ can be rewritten as

$$p_\theta(y_{1:T}|u_{1:T}) = p_\theta(y_1|u_1) \prod_{t=2}^{T} p_\theta(y_t|y_{1:t-1}, u_{1:t}), \tag{7.14}$$

where $p_\theta(y_t|y_{1:t-1}, u_{1:t})$ denotes the pdf of the measurement y_t given the tuple $(y_{1:t-1}, u_{1:t}, \theta)$. The pdf $p_\theta(y_t|y_{1:t-1}, u_{1:t})$ can be obtained from the model (7.5) as

$$p_\theta(y_t|y_{1:t-1}, u_{1:t}) = p_e^\theta(\varepsilon_t(\theta))\,, \tag{7.15}$$

where

$$\varepsilon_t(\theta) := H_\theta^{-1}(q)\,(y_t - G_\theta(q)\,u_t) \tag{7.16}$$

is the one step ahead prediction error associated with the model (7.5) at time t. Inserting (7.14)-(7.16) into (7.13) we finally obtain

$$a(\theta^\star, \theta^{(i)}) = \min\left(1,\, \frac{p_{\theta^\star}(y_1|u_1)\,\pi_\Theta(\theta^\star)\,f(\theta^{(i)}|\theta^\star)\prod_{t=2}^T p_e^{\theta^\star}(\varepsilon_t(\theta^\star))}{p_{\theta^{(i)}}(y_1|u_1)\,\pi_\Theta(\theta^{(i)})\,f(\theta^\star|\theta^{(i)})\prod_{t=2}^T p_e^{\theta^{(i)}}(\varepsilon_t(\theta^{(i)}))}\right). \tag{7.17}$$

From (7.17) we see that the acceptance probability can be computed using the model (7.5) for every $\theta \in \Theta$. However, to implement the MH sampler over the parameter set Θ we need to address two aspects:

1. the specification of the prior π_Θ over Θ, and
2. the implementation of a random walk[1] over Θ using the proposal distribution f.

The first point will be addressed in the next section, while the second point will be discussed in Section 7.5.

7.4 Specifying priors over the parameter set

As mentioned in Section 7.2, the prior pdf π_Θ reflects the available knowledge (or belief) the user has before the data set $\mathbf{Z}_T$ is obtained. In the literature, most of the methods using Gibbs and MH samplers specify an explicit prior over Θ (i.e., the pdf π_Θ is an explicit function of θ [49, 73, 208]). In addition, the requirement of a conjugate prior is generally imposed, to analytically compute posterior distributions. In case the user wants to use more general priors, then the methods in [49, 73, 208] cannot be employed, and hence other alternatives are needed.

Model order priors

In this chapter we are interested in exploring more general priors than those introduced in [73, 208]. Extending the class of priors will allow us to encode particular properties of the model in the problem.

Here we are interested in defining priors over the order of the model (7.5). To proceed, we assume that the set

$$\mathcal{M} := \left\{(G_\theta, H_\theta, p_e^\theta)\colon \theta \in \Theta\right\} \tag{7.18}$$

[1] By random walk we mean a succession of random steps.

contains all rational functions in q for the pair (G_θ, H_θ) that are of order at most $n_{\mathcal{M}}$, where $n_{\mathcal{M}}$ is a fixed positive integer. This assumption is similar to the one introduced in [208] in the sense that prior knowledge about an upper bound on the order of the system is required.

At this point we may consider using one of the available methods for Bayesian model order selection [91]. However, the assumption of a differential bijective mapping between model structures of different order limits its applicability in the present problem. In particular, the main difficulty of applying the method of [91] is the requirement of a closed form expression for the bijective mapping and its Jacobian, which are needed in order to run an MH sampler targeting the posterior distribution. To circumvent this issue, we work instead with priors defined over the Hankel singular values of the rational function G_θ. The definition of Hankel singular values is given next [90]:

Definition 7.2 *(Hankel singular values) Consider the discrete time, stable, linear state space system*

$$x_{t+1} = A\,x_t + B\,u_t\,, \tag{7.19a}$$

$$y_t = C\,x_t + D\,u_t\,, \tag{7.19b}$$

with $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$, $D \in \mathbb{R}^{n_y \times n_u}$, and the matrices P, $Q \in \mathbb{R}^{n_x \times n_x}$ which are the positive definite solutions to the corresponding Lyapunov equations

$$APA^\top - P + BB^\top = 0\,, \tag{7.20a}$$

$$A^\top QA - Q + C^\top C = 0\,. \tag{7.20b}$$

The values $\sigma_i = \sqrt{\lambda_i(PQ)}$, $i \in \{1, \ldots, n_x\}$, are called the Hankel singular values of the system (7.19)*.*

Since the Hankel singular values are closely related to the order of a model, they have been widely employed in model order reduction [90]. A Hankel singular value close to zero indicates that the order of the model can be reduced by one with almost negligible effect on the model properties (e.g., its bandwidth, step response, DC gain, etc.). Thus, the specification of a prior over the Hankel singular values of G_θ reflects prior knowledge (or belief) about the order of the system (7.2).

To evaluate the prior over the Hankel singular values associated with G_θ, we first write G_θ in the state space form (7.19), and then we compute the Hankel singular values according to Definition 7.2. In this manner, a prior pdf over the Hankel singular values of G_θ implicitly accounts for its complexity.

Improper priors

At this point we need to emphasize that a prior distribution over the Hankel singular values of G_θ is not necessarily normalizable[2] (or *proper*). In case the

[2]By normalizable we mean that $\int_\Theta \pi_\Theta(\theta) d\theta < \infty$.

prior is proper but not normalized, the MH sampler can still be used, since the computation of the acceptance probability (7.17) only requires the knowledge of π_Θ up to a normalization constant. On the other hand, if the prior π_Θ is improper, the computation of (7.6) is still possible provided that the normalization constant (7.7) is finite [159, Section 1.5]. However, the samples obtained from the MH sampler can no longer be interpreted as samples from the posterior distribution (7.6), since π_Θ is not a pdf anymore.

As noted above, the correctness of the MH sampler is guaranteed as long as the normalization constant (7.7) is finite. However, the boundedness of (7.7) is difficult to verify in complex settings [159, p. 30], which is the case here as we impose priors over the Hankel singular values of G_θ.

To guarantee the correctness of the MH sampler, we impose the following assumptions:

Assumption 7.2 *The set Θ can be written as*

$$\Theta = \Theta_G \times \Theta_H \times \Theta_e \,, \tag{7.21}$$

where $\theta = (\theta_G^\top, \theta_H^\top, \theta_e^\top)^\top \in \Theta$ is such that the model (7.5) is given by $G_\theta = G_{\theta_G}$, $H_\theta = H_{\theta_H}$ and $e_t^\theta = e_t^{\theta_e}$ for all t.

Assumption 7.3 *For the set Θ satisfying Assumption 7.2, the set Θ_G is bounded.*

Assumption 7.2 requires that the rational transfer functions G_θ, H_θ and the white noise process $\{e_t^\theta\}$ in (7.5) are independently parameterized. In addition, the boundedness of the set Θ_G required by Assumption 7.3 does not introduce considerable conservatism to the model (7.5). Indeed, for reasonable parameterizations of (G_θ, H_θ), from Assumption 7.1 it follows that the parameters describing the denominator of G_θ and H_θ lie in a bounded set. Hence, Assumption 7.3 only implies that, for reasonable parameterizations of G_θ, the parameters describing the numerator of G_θ also lie in a bounded set. This is a natural assumption as the model (7.5) describes systems satisfying the bounded input-bounded output property.

To guarantee the boundedness of the normalization constant (7.7), we also impose the following requirement over π_Θ:

Assumption 7.4 *For the set Θ satisfying Assumption 7.2, the prior π_Θ is given for all $\theta \in \Theta$ by*

$$\pi_\Theta(\theta) = \pi_{\Theta_G}(\theta_G)\, \pi_{\Theta_H}(\theta_H)\, \pi_{\Theta_e}(\theta_e) \,, \tag{7.22}$$

where π_{Θ_G}, π_{Θ_H} and π_{Θ_e} are proper priors defined over Θ_G, Θ_H and Θ_e, respectively. In addition, π_{Θ_G} is a bounded function on Θ_G.

We note that Assumption 7.4 considers priors over Θ that are possibly not normalized.

In our setting, the prior π_{Θ_G} is specified over the Hankel singular values associated with G_{θ_G}. If we denote by π_Σ a proper and bounded prior imposed over the Hankel singular values $(\sigma_1, \ldots, \sigma_{n_\mathcal{M}})$ of models with $\theta_G \in \Theta_G$, then a prior π_{Θ_G} is given by

$$\pi_{\Theta_G}(\theta_G) = \frac{\pi_\Sigma(\sigma_1(\theta_G), \ldots, \sigma_{n_\mathcal{M}}(\theta_G))}{K_{\Theta_G}}\,, \tag{7.23}$$

where $\sigma_i(\theta_G)$ denotes the i-th Hankel singular value associated with G_{θ_G} ($i \in \{1, \ldots, n_\mathcal{M}\}$), and

$$K_{\Theta_G} := \int_{\Theta_G} \pi_\Sigma(\sigma_1(\theta_G), \ldots, \sigma_{n_\mathcal{M}}(\theta_G))\, d\theta_G\,, \tag{7.24}$$

is the normalization constant.

Remark 7.2 *A closer inspection at π_{Θ_G} in (7.23) tells us that the prior imposed in this chapter results in a uniform distribution over all $\theta_G \in \Theta_G$ having the same Hankel singular values $\{\sigma_i(\theta_G)\}_{i=1}^{n_\mathcal{M}}$.*

We note that (7.24) is finite as the set Θ_G is bounded. Indeed,

$$K_{\Theta_G} \leq \sup_{\substack{(z_1,\ldots,z_{n_\mathcal{M}})=(\sigma_1(\theta_G),\ldots,\sigma_{n_\mathcal{M}}(\theta_G)) \\ \theta_G \in \Theta_G}} \pi_\Sigma(z_1, \ldots, z_{n_\mathcal{M}}) \int_{\Theta_G} d\theta < \infty\,. \tag{7.25}$$

Hence, the prior pdf (7.23) fulfills the requirements in Assumption 7.4.

Finally, if the set Θ satisfies Assumption 7.2 and π_Θ satisfies Assumption 7.4 with π_{Θ_G} given by (7.23), then

$$\int_\Theta \pi_\Theta(\theta)\, d\theta \leq \int_{\Theta_G} \pi_{\Theta_G}(\theta_G)\, d\theta_G \int_{\Theta_H \times \Theta_e} \pi_{\Theta_H}(\theta_H)\, \pi_{\Theta_e}(\theta_e)\, d\theta_H d\theta_e < \infty\,. \tag{7.26}$$

Thus, we conclude that the normalization constant (7.7) is bounded for almost every value of $\mathbf{Z}_T$ [159, Problem 1.47.b]. Therefore, the MH sampler draws samples that are approximately distributed according to (7.6), when π_{Θ_G} is a proper prior defined over the Hankel singular values of G_{θ_G} (according to (7.23)) and Assumptions 7.2-7.4 are satisfied.

The correctness of the MH sampler for the proposed setup is established by

Theorem 7.1 *Consider the set Θ satisfying Assumptions 7.2-7.3 and a prior π_Θ satisfying Assumption 7.4, with π_{Θ_G} given by (7.23). Let $\{\theta^{(i)}\}_{i\geq 1}$ be the samples generated by the Markov chain in Algorithm 7.1, with $\Upsilon(\theta) = p_\theta(y_{1:N}|u_{1:N})\pi_\Theta(\theta)/K_\Theta$, and the conditional pdf $f(\theta|\theta^\star)$ whose support includes Θ. If*

$$\mathbf{P}\left\{p_{\theta^{(i)}}(y_{1:N}|u_{1:N})\pi_\Theta(\theta^{(i)})f(\theta^\star|\theta^{(i)}) \leq p_{\theta^\star}(y_{1:N}|u_{1:N})\pi_\Theta(\theta^\star)f(\theta^{(i)}|\theta^\star)\right\} < 1 \tag{7.27}$$

is satisfied for all $\theta^{(i)}$*, then*

$$\lim_{i\to\infty} \left\| \int_{\Theta} p^{(i)}(\cdot|\theta^{(1)})\mu(\theta^{(1)})d\theta^{(1)} - p(\cdot|\mathbf{Z}_T) \right\|_{\mathrm{TV}} = 0 \tag{7.28}$$

for almost every value of $\mathbf{Z}_T$ *and any initial pdf* μ*, where* $p^{(i)}$ *denotes the conditional pdf of* $\theta^{(i)}$ *given* $\theta^{(1)}$*, and*

$$\|\beta\|_{\mathrm{TV}} := \frac{1}{2}\int_{\Theta} |\beta(\theta)|d\theta\,, \tag{7.29}$$

with $\beta\colon \Theta \to \mathbb{R}$*.*

Proof *Follows from the existence of the posterior pdf* $p(\cdot|\mathbf{Z}_T)$ *according to [159, Problem 1.47.b] and the convergence of the MH sampler provided the existence of* $p(\cdot|\mathbf{Z}_T)$ *from [160, Corollary 7.5].* ■

Remark 7.3 *The expression* (7.27) *establishes a sufficient condition for guaranteeing that the Markov chain generated by the MH sampler is aperiodic. The statement in Theorem 7.1 can also be proven under less restrictive conditions than* (7.27)*; we refer to [160, Section 7.3.2] for more details.*

7.5 A random walk over the parameter set

To run the MH sampler over Θ, we need to define a random walk over Θ based on the proposal distribution f. One of the main difficulties in implementing such a random walk is to ensure the stability of the pair (G_θ, H_θ) and H_θ^{-1} for the value of θ being sampled. To address this point, we note that the real rational functions G_θ, H_θ can be written as

$$G_\theta(q) = K\prod_{i=1}^{n_{\mathcal{M}}/2} \frac{1 + b_1^i q^{-1} + b_2^i q^{-2}}{1 + f_1^i q^{-1} + f_2^i q^{-2}}\,, \tag{7.30a}$$

$$H_\theta(q) = \prod_{i=1}^{n_{\mathcal{M}}/2} \frac{1 + c_1^i q^{-1} + c_2^i q^{-2}}{1 + d_1^i q^{-1} + d_2^i q^{-2}}\,, \tag{7.30b}$$

if $n_{\mathcal{M}}$ is even, and

$$G_\theta(q) = K\frac{1 + b_0 q^{-1}}{1 + f_0 q^{-1}}\prod_{i=1}^{n_{\mathcal{M}}/2-1} \frac{1 + b_1^i q^{-1} + b_2^i q^{-2}}{1 + f_1^i q^{-1} + f_2^i q^{-2}}\,, \tag{7.31a}$$

$$H_\theta(q) = \frac{1 + c_0 q^{-1}}{1 + d_0 q^{-1}}\prod_{i=1}^{n_{\mathcal{M}}/2-1} \frac{1 + c_1^i q^{-1} + c_2^i q^{-2}}{1 + d_1^i q^{-1} + d_2^i q^{-2}}\,, \tag{7.31b}$$

if $n_{\mathcal{M}}$ is odd. In Equations (7.30)-(7.31) the gain K and the polynomial coefficients are real valued functions of the parameter θ. The requirements in Assumption 7.1 are satisfied by guaranteeing that all the poles of G_θ and all the poles and zeros of H_θ belong to $\mathbb{D}$. Assumption 7.1 is satisfied if and only if [11]

$$|f_0| < 1\,, \qquad |c_0| < 1\,, \qquad |d_0| < 1\,, \tag{7.32a}$$

$$\left|f_2^i\right| < 1\,, \qquad \left|c_2^i\right| < 1\,, \qquad \left|d_2^i\right| < 1\,, \text{ for all } i\,, \tag{7.32b}$$

$$\left|f_1^i\right| < 1 + f_2^i\,, \qquad \left|c_1^i\right| < 1 + c_2^i\,, \qquad \left|d_1^i\right| < 1 + d_2^i\,, \text{ for all } i\,. \tag{7.32c}$$

Thus, given the proposal distribution f, at step i of the MH sampler we generate $\theta^\star$ from $f(\theta|\theta^{(i)})$ and check if the conditions (7.32) are satisfied. If they are not satisfied, we then sample a new parameter $\theta^\star$ until the inequalities (7.32) are fulfilled. In this manner we guarantee that the samples $\theta^\star$ generated from f belong to Θ. Note that this method can be seen as a rejection sampling procedure.

Remark 7.4 *The approach presented here for implementing a random walk over the parameter set does not guarantee that the resulting Markov chain will possess a fast convergence rate to its stationary distribution. A fast convergence rate of the Markov chain implies that a small number of samples have to be discarded at the beginning (to avoid the influence of the initial conditions in the samples), and that the stationary distribution of the chain can be approximated by a small number of samples. Hence, the convergence rate of the resulting Markov chain affects the computational cost of approximating the stationary distribution by its samples.*

One way to improve the convergence rate is to scale the covariance matrix of the random walk proposal by the covariance matrix of the posterior distribution [50]. To this end, we first run the MH sampler to estimate the covariance matrix of the posterior distribution based on the generated samples, which is then used to build the random walk for the second run of the MH sampler. In the second run of the MH sampler, the covariance matrix of the random walk is the one employed in the first run, scaled by the estimated covariance matrix of the posterior distribution. In this manner, the samples from the second run are obtained from a Markov chain with better convergence rate than the first run.

Another option to improve the convergence rate is to reparameterize the model (7.5) *so that the resulting MH sampler is performed over an unconstrained space. To this end, the acceptance probability* (7.17) *needs to be modified to include the gradient of the transformation between parameters in Θ and parameters in the unconstrained space. For further details we refer to [50], where a more elaborate discussion on extensions for the MH sampler is presented.*

7.6 Numerical example

System and algorithm setup

Consider the system

$$y_t = G_o(q)\, u_t + e_t\,, \tag{7.33}$$

where

$$G_o(q) = \frac{2\,q^{-2}}{1 - 1.4\,q^{-1} + 0.81\,q^{-2}}\,, \tag{7.34}$$

and $\{u_t\}$, $\{e_t\}$ are Gaussian white noise processes of zero mean and unit variance. To simplify the example, we assume that the system (7.33) is at rest for all $t \leq 0$. The system (7.33) is estimated based on $\mathbf{Z}_T$, with $T = 100$, using the model

$$y_t = G_{\theta_G}(q)\,u_t + H_{\theta_H}(q)\,e_t^{\theta_e}\,, \tag{7.35}$$

where $\{e_t^{\theta_e}\}$ is assumed to be Gaussian white noise of zero mean and variance $\theta_e > 0$, which has to be estimated. In this example, the pair $(G_{\theta_G}, H_{\theta_H})$ can describe all real rational functions in q satisfying Assumption 7.1 and that are of order at most $n_{\mathcal{M}} = 10$. In particular, the parameter vectors $\theta_G \in \Theta_G \subset \mathbb{R}^{2n_{\mathcal{M}}+1}$, $\theta_H \in \Theta_H \subset \mathbb{R}^{2n_{\mathcal{M}}}$ contain the gain and the numerator and denominator polynomial coefficients of the rational functions G_{θ_G} and H_{θ_H}, i.e.,

$$G_{\theta_G}(q) = \theta_{G,\,0}\frac{1 + \sum_{i=1}^{n_{\mathcal{M}}} \theta_{G,\,i}\,q^{-i}}{1 + \sum_{i=1}^{n_{\mathcal{M}}} \theta_{G,\,i+n_{\mathcal{M}}}\,q^{-i}}\,, \tag{7.36a}$$

$$H_{\theta_H}(q) = \frac{1 + \sum_{i=1}^{n_{\mathcal{M}}} \theta_{H,\,i}\,q^{-i}}{1 + \sum_{i=1}^{n_{\mathcal{M}}} \theta_{H,\,i+n_{\mathcal{M}}}\,q^{-i}}\,, \tag{7.36b}$$

with $\theta_G = (\theta_{G,\,0},\,\ldots,\,\theta_{G,\,2n_{\mathcal{M}}})^\top$, $\theta_H = (\theta_{H,\,1},\,\ldots,\,\theta_{H,\,2n_{\mathcal{M}}})^\top$. Hence, the parameter vector is given by $\theta = (\theta_G^\top,\,\theta_H^\top,\,\theta_e)^\top \in \Theta \subset \mathbb{R}^{2(2n_{\mathcal{M}}+1)}$. In addition, we require that each entry of the parameter vector $(\theta_{G,\,0},\,\ldots,\,\theta_{G,\,n_{\mathcal{M}}})^\top$ satisfies $|\theta_{G,i}| \leq 10^3$ (hence Assumption 7.3 is satisfied).

To implement the MH sampler for this example, we define a proposal distribution over Θ following the discussion in Section 7.5. Based on the rational functions (7.30), we define f as the product of the proposals for each parameter in (7.30) and the proposal for θ_e, where

$$\theta_e{}^\star \sim \mathcal{N}(\theta_e,\,10^{-3})\,, \tag{7.37a}$$

$$K^\star \sim \mathcal{N}(K,\,10^{-4})\,, \tag{7.37b}$$

$$f_j^{i\,\star} \sim \mathcal{N}(f_j^i,\,10^{-4})\,, i \in \{1,\,\ldots,\,n_{\mathcal{M}}/2\},\, j \in \{1,\,2\}\,, \tag{7.37c}$$

$$c_j^{i\,\star} \sim \mathcal{N}(c_j^i,\,10^{-4})\,, i \in \{1,\,\ldots,\,n_{\mathcal{M}}/2\},\, j \in \{1,\,2\}\,, \tag{7.37d}$$

$$d_j^{i\,\star} \sim \mathcal{N}(d_j^i,\,10^{-4})\,, i \in \{1,\,\ldots,\,n_{\mathcal{M}}/2\},\, j \in \{1,\,2\}\,. \tag{7.37e}$$

From the proposals (7.37) we note that $f(\theta|\theta^\star) = f(\theta^\star|\theta)$. Hence, the acceptance probability (7.17) for this example becomes

$$a(\theta^\star, \theta) = \min\left(1,\,\frac{p_{\theta^\star}(y_1|u_1)\prod_{t=2}^{T} p_e^{\theta^\star}(\varepsilon_t(\theta^\star))\,\pi_\Theta(\theta^\star)}{p_\theta(y_1|u_1)\prod_{t=2}^{T} p_e^{\theta}(\varepsilon_t(\theta))\,\pi_\Theta(\theta)}\right). \tag{7.38}$$

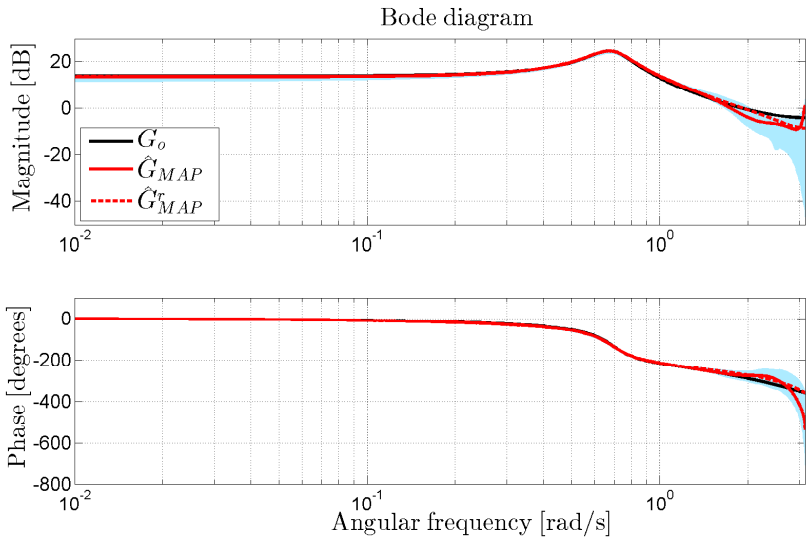

Figure 7.1: Bode diagram of the true system G_o (black), the maximum a posteriori estimate $\hat{G}_{\text{MAP}}$ (red, continuous line), and the reduced model $\hat{G}^r_{\text{MAP}}$ (red, dashed line), obtained by balanced truncation of $\hat{G}_{\text{MAP}}$. The shaded area corresponds to the 98% credible interval for the posterior distribution, centered at its mean value.

For the prior distribution over Θ, we consider

$$\pi_{\Theta}(\theta) = \pi_{\Theta_G}(\theta_G)\, \pi_{\Theta_H}(\theta_H)\, \pi_{\Theta_e}(\theta_e)\,, \tag{7.39}$$

where

$$\pi_{\Theta_G}(\theta_G) = \prod_{i=1}^{n_{\mathcal{M}}} \pi_{\sigma_i}(\theta_G)\,, \tag{7.40}$$

with π_{σ_i} denoting the prior pdf associated with the i-th Hankel singular value of G_{θ_G}, for $i \in \{1, \ldots, n_{\mathcal{M}}\}$. In addition, $\{\pi_{\sigma_i}\}_{i=1}^{n_{\mathcal{M}}}$, and π_{Θ_e} are defined as the pdf of a Gamma distribution with scale parameter 1 and shape parameter 1, and π_{Θ_H} is defined as the pdf of a uniform distribution over Θ_H.

To approximate the posterior (7.6) we compute samples with the MH sampler introduced in Algorithm 7.1, with which we generate 10^5 samples and discard the first $2 \cdot 10^4$ ones to account for the transient behavior of the Markov chain. The last $8 \cdot 10^4$ samples are employed to approximate the posterior distribution.

Model estimates

The samples obtained by running Algorithm 7.1 are employed to compute the parameters estimates (7.8) and (7.9), where the posterior distribution (7.6) is

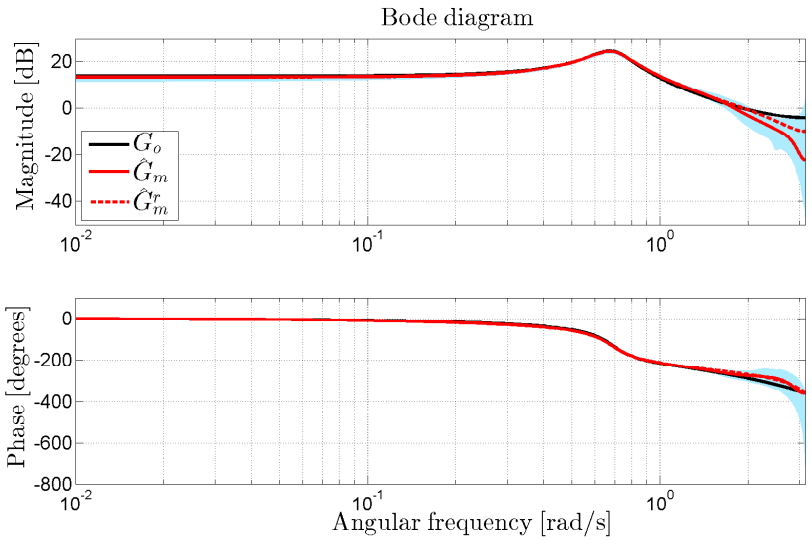

Figure 7.2: Bode diagram of the true system G_o (black), the mean value estimate $\hat{G}_m$ (red, continuous line), and the reduced model $\hat{G}_m^r$ (red, dashed line), obtained by balanced truncation of $\hat{G}_m$. The shaded area corresponds to the 98% credible interval.

replaced by its approximation based on $\{\theta^{(i)}\}_{i=2\cdot10^4}^{10^5}$.

Figure 7.1 presents[3] the Bode diagram of $\hat{G}_{\text{MAP}} := G_{\hat{\theta}_T}$, where $\hat{\theta}_T$ is the maximum a posteriori estimate (7.8). In addition, Figure 7.1 also presents the balanced truncation of the estimate $\hat{G}_{\text{MAP}}$, denoted by $\hat{G}^r_{\text{MAP}}$, which is obtained by removing the states in the balanced state-space representation of $\hat{G}_{\text{MAP}}$ associated with the Hankel singular values with a value less than 10% of the largest Hankel singular value. The same results are presented in Figure 7.2 for the mean estimate (7.9), where $\hat{G}_m$ and $\hat{G}_m^r$ denote the mean estimate and its balanced truncation, respectively. From these figures we see that the estimates capture well the dynamics of the true system (7.33). Moreover, we also note that the degradation in the estimates is larger in the frequency range where the magnitude of the response of G_o is at most 0 [dB], which is the value of the input to noise variance ratio. This can also be seen in the credible intervals in Figures 7.1 and 7.2, where the uncertainty is large when the magnitude of the response of G_o is less than or equal to the input to noise variance ratio.

The models obtained by balanced truncations of the estimated functions $\hat{G}^r_{\text{MAP}}$,

[3]The phase of the estimates in Figures 7.1-7.2 is shifted by 2π [rad] to ease the discussion.

and $\hat{G}_m^r$ are

$$\hat{G}_{\text{MAP}}^r(q) = \frac{-0.29\,(1 - 3.26\,q^{-1})\,(1 + 1.99\,q^{-1})}{1 - 1.39\,q^{-1} + 0.80\,q^{-2}}\,, \tag{7.41}$$

$$\hat{G}_m^r(q) = \frac{-0.35\,(1 - 2.96\,q^{-1})\,(1 + 1.73\,q^{-1})}{1 - 1.39\,q^{-1} + 0.80\,q^{-2}}\,. \tag{7.42}$$

From these expressions we see that the order of the reduced models coincides with the order of the system (7.33). In addition, the relative degree of G_o is estimated by the finite nonminimum phase zeros in (7.41) and (7.42). Indeed, the relative degree of a discrete time linear system is associated with time delays, which can be interpreted as nonminimum phase zeros at infinity. Hence, the models (7.41) and (7.42) try to capture the time delays (and thus the relative degree of G_o) by finite nonminimum phase zeros.

We can also compute the model fit for the estimated functions (7.41) and (7.42), by defining

$$\mathcal{F}(G_1, G_2) := 100 \cdot \left(1 - \frac{\|G_1 - G_2\|_2}{\|G_2\|_2}\right)\,[\%]\,, \tag{7.43}$$

where G_1, and G_2 are rational functions in q. Hence, for the reduced models we obtain $\mathcal{F}(\hat{G}_{\text{MAP}}^r, G_o) = 93.86\,\%$ for the MAP estimate, and $\mathcal{F}(\hat{G}_m^r, G_o) = 91.98\,\%$ for the mean estimate. Therefore, the estimated models describe the main characteristics in the frequency response of G_o.

As a comparison, we run an MH sampler targeting the posterior distribution of θ when π_{Θ_G} is defined as a normal distribution with zero mean and identity covariance matrix for the parameter vector $\{\theta_{G,\,i}\}_{i=0}^{2n_{\mathcal{M}}}$. However, due to convergence issues, this time the MH sampler runs over $5 \cdot 10^5$ iterations, and we discard the first 10^5 samples. The convergence issues are related to the mixing properties of the designed Markov chain, resulting from the difficulty of properly tuning the random walk over the parameter set in this case.

As before, we compute the balanced truncations of the new estimated functions $\hat{G}_{\text{MAP}}^r$, and $\hat{G}_m^r$, which are given by

$$\hat{G}_{\text{MAP}}^{r,\mathcal{N}}(q) = \frac{3.2 \cdot 10^{-2}\,(1 - 15.73\,q^{-1} + 74.8\,q^{-2})}{1 - 1.39\,q^{-1} + 0.80\,q^{-2}}\,, \tag{7.44}$$

$$\hat{G}_m^{r,\mathcal{N}}(q) = \frac{4.5 \cdot 10^{-2}\,(1 - 14.44\,q^{-1} + 54.79\,q^{-2})}{1 - 1.39\,q^{-1} + 0.80\,q^{-2}}\,. \tag{7.45}$$

The model fit for the new estimated models are $\mathcal{F}(\hat{G}_{\text{MAP}}^{r,\mathcal{N}}, G_o) = 87.27\,\%$, and $\mathcal{F}(\hat{G}_m^{r,\mathcal{N}}, G_o) = 83.49\,\%$. Hence, adding priors over the Hankel singular values in this example helps to improve the model estimates by capturing the main properties of the underlying system.

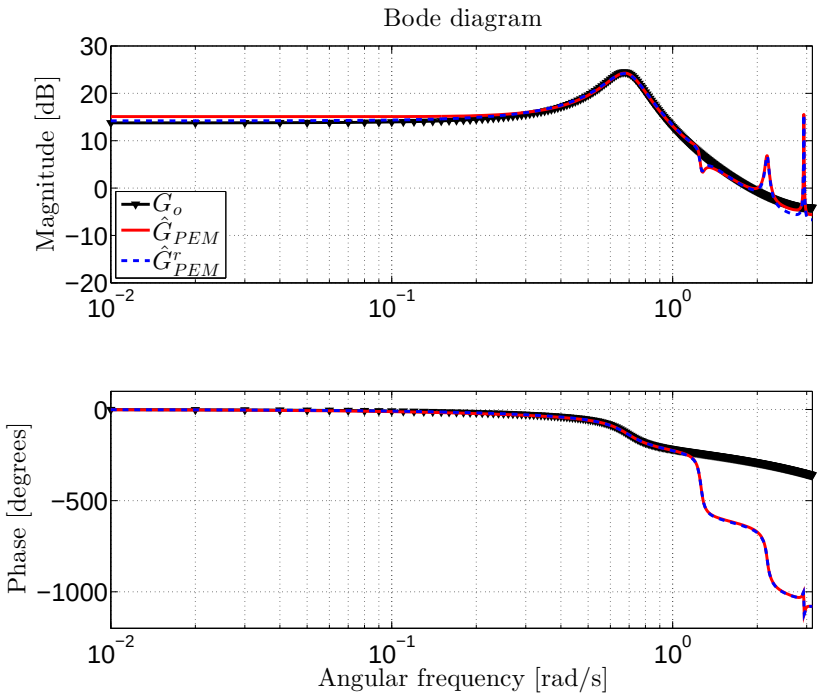

Figure 7.3: Bode diagram of the true system G_o (black, triangle line), the PEM estimate $\hat{G}_{PEM}$ (red, continuous line), and the reduced PEM model $\hat{G}^r_{PEM}$ (blue, dashed line), obtained by balanced truncation of $\hat{G}_{PEM}$.

Comparison with available identification methods

We can further compare the results obtained from the MH sampler with those computed by standard identification methods. To this end, the same data set $\mathbf{Z}_T$ used by the MH sampler is employed to estimate a model based on PEM and subspace identification. Both methods are implemented in Matlab using the commands `pem` for PEM and `n4sid` for subspace identification. For PEM, a Box-Jenkins model is chosen, where both G_θ and H_θ are rational functions in q of order 10. For subspace identification, a linear state space model in innovations form of order 10 is chosen as the model set. The estimated models from PEM and subspace identification are reduced by the same balanced truncation procedure considered for the models obtained from the MH sampler.

The Bode diagrams of the estimated models for G_o using PEM and subspace

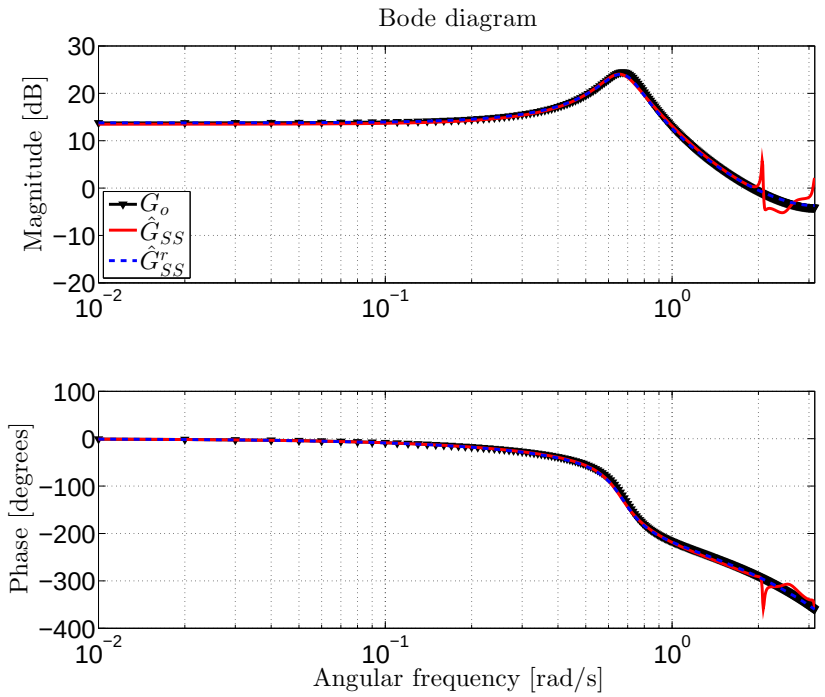

Figure 7.4: Bode diagram of the true system G_o (black, triangle line), the subspace estimate $\hat{G}_{SS}$ (red, continuous line), and the reduced subspace model $\hat{G}^r_{SS}$ (blue, dashed line), obtained by balanced truncation of $\hat{G}_{SS}$.

identification are presented in Figure 7.3 and Figure 7.4, respectively[4]. The models estimated by both identification methods are close in magnitude to the frequency response of G_o. In addition, they present a degradation in the model fit for the frequency region where the magnitude of G_o is close to the input to noise variance ratio, which is also observed in the estimates derived from the MH sampler.

Once the balanced truncation is performed over the models obtained by PEM ($\hat{G}_{\text{PEM}}$) and subspace identification ($\hat{G}_{\text{SS}}$), the reduced order models are[5]

$$\hat{G}^r_{\text{PEM}}(q) = G_1(q)\, G_2(q)\, G_3(q)\,, \tag{7.46}$$

$$\hat{G}^r_{\text{SS}}(q) = \frac{-0.08\, q^{-1}(1 - 25\, q^{-1})}{1 - 1.41\, q^{-1} + 0.80\, q^{-2}}\,, \tag{7.47}$$

[4]To ease the presentation, the phase of the estimates in Figures 7.3 and 7.4 is shifted by 6π [rad] and 2π [rad] respectively.

[5]The model obtained with PEM is factorized to ease the presentation.

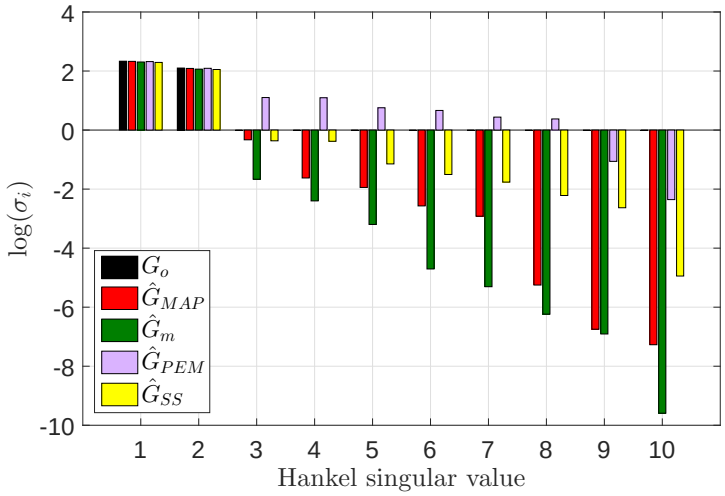

Figure 7.5: Hankel singular values for the estimated models and the true system G_o (in logarithmic scale).

where $\hat{G}^r_{\text{PEM}}$ and $\hat{G}^r_{\text{SS}}$ denote the reduced models for PEM and subspace identification respectively, and

$$G_1(q) = \frac{-0.03\,(1 - 12.2\,q^{-1})\,(1 + 4.35\,q^{-1})\,(1 + 1.89\,q^{-1} + 0.93\,q^{-2})}{(1 + 1.95\,q^{-1} + 0.99\,q^{-2})\,(1 - 1.39\,q^{-1} + 0.80\,q^{-2})}\,, \tag{7.48}$$

$$G_2(q) = \frac{1 + 1.26\,q^{-1} + 1.23\,q^{-2}}{1 + 1.09\,q^{-1} + 0.94\,q^{-2}}\,, \tag{7.49}$$

$$G_3(q) = \frac{1 - 0.62\,q^{-1} + 1.05\,q^{-2}}{1 - 0.60\,q^{-1} + 0.93\,q^{-2}}\,. \tag{7.50}$$

From the expression for $\hat{G}^r_{\text{PEM}}$ we observe that the reduced model for the PEM estimate is of order 8, and hence it overestimates the order of G_o. On the other hand, the reduced order model $\hat{G}^r_{\text{SS}}$ results in a second order rational function. Thus, the reduced order model derived from the subspace estimate coincides with the order of the rational function G_o. The model fit for the reduced order models is $\mathcal{F}(\hat{G}^r_{\text{PEM}}, G_o) = 81.34\%$ for PEM and $\mathcal{F}(\hat{G}^r_{\text{SS}}, G_o) = 90.43\%$ for subspace identification. Thus, even though the subspace estimate $\hat{G}^r_{\text{SS}}$ captures the true order of the system, its performance is still slightly below the one achieved by estimates generated from the MH sampler when priors over the Hankel singular values of G_θ are imposed.

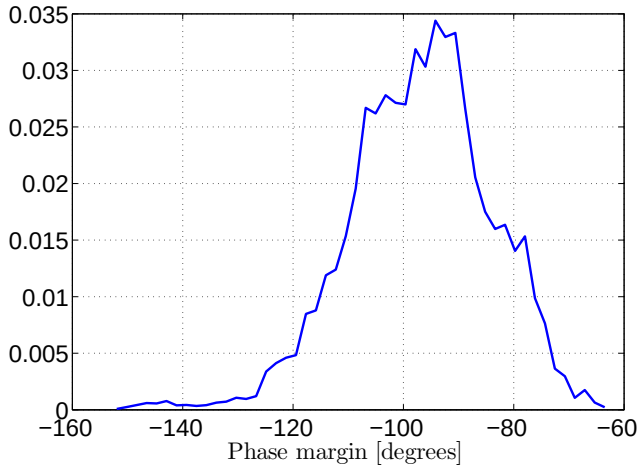

Figure 7.6: Empirical pdf of the phase margin.

Hankel singular values

To illustrate the effect of imposing priors over the Hankel singular values of G_θ, we present in Figure 7.5 the Hankel singular values (in logarithmic scale) for the estimated models $\hat{G}_{\text{MAP}}$ and $\hat{G}_m$ obtained from the MH sampler with prior distribution (7.39)-(7.40) and the estimated models $\hat{G}_{\text{PEM}}$ and $\hat{G}_{\text{SS}}$. From this plot we see that all the estimated models behave similarly for the first and second Hankel singular values. In the case of $\hat{G}_{\text{PEM}}$, the remaining Hankel singular values cannot be completely discarded, as they indicate that the associated states have a noticeable effect in the resulting model. The situation improves for $\hat{G}_{\text{SS}}$, where we see that the states associated with the first and second Hankel singular values can almost completely explain the resulting model. The behavior observed for the Hankel singular values of $\hat{G}_{\text{SS}}$ is sharpened for $\hat{G}_{\text{MAP}}$ and $\hat{G}_m$, where the evidence for a reduced model order is clearer than for $\hat{G}_{\text{SS}}$. Hence, imposing priors over the Hankel singular values of G_θ helps to account for the model complexity of G_θ.

Posterior distributions

As a final illustration, we can analyze the posterior distribution of functions of the parameter vector based on the samples obtained from the MH algorithm. For example, Figure 7.6 presents the empirical pdf obtained for the phase margin using the samples of the posterior pdf $p(\theta|\mathbf{Z}_T)$, when the prior π_{Θ_G} is defined over the Hankel singular values of the transfer function G_{θ_G}. We note in Figure 7.6 that the phase margin is nonpositive with probability one (within the sampling variability

of the MCMC samples), which implies that the resulting closed loop with negative unitary feedback would be unstable with very high probability.

7.7 Conclusions

This chapter introduced a Metropolis Hastings sampler for Bayesian identification of SISO linear systems. By using the Metropolis Hastings sampler, we allow the choice of more general priors over the model parameters than those considered by the Gibbs sampler. This property has been exploited by specifying priors over the Hankel singular values of the model, which are closely connected to the order of the model.

Part III

Risk coherent framework for system identification

Chapter 8

On risk theory in system identification

As it can be noted in the first part of this thesis, accounting for the uncertainty on the true model description is one of the issues to be considered in input design. This issue was circumvented by formulating a robust problem accounting for the model uncertainty. Here we further explore how to properly account for this lack of knowledge on the true model description. The content of this chapter is based on [191].

8.1 Introduction

Uncertainty is an issue common to many research areas. By uncertainty we mean the lack of knowledge to fully describe a phenomenon. The lack of knowledge causes severe difficulties when we are interested in determining the best decision. Examples of this problem can be found in control design, where the controller must be designed with limited information about the plant dynamics [214, 215], and in portfolio optimization, where the returns are maximized subject to limited information about the future evolution of the asset prices [15, 17, 74, 111, 143, 149].

In the same line, many problems arising in system identification are solved under limited information. In system identification, one source of uncertainty can be understood as the lack of knowledge about the true dynamics of the process to be modeled. The uncertainty associated with the process dynamics is of importance in applications where the optimal decision depends on the true model description. This is the case in input design, where the optimal input sequence depends on true process dynamics [122]. Approaches to solve this issue have been presented in the literature, which can be classified into two classes: (i) sequential or adaptive procedures, where a new design is obtained based on the current estimates of the process dynamics [79, 80, 118, 151]; and (ii) robust procedures, where the design is obtained by including the uncertainty in the optimization problem [102, 153, 167].

In this chapter, we are interested in addressing the uncertainty in input design by using the robust approach.

The robust approach to uncertainty in system identification has been analyzed in the literature, and several techniques have been proposed [114, 128, 164]. The main idea behind these results is the inclusion of a mapping from the space of functions with uncertainty to a scalar value. The scalar value takes into account the uncertainty associated with the process dynamics. Some examples of the mappings employed are the expected value and the supremum over the set of possible descriptions of the process dynamics. However, there is no analysis of how well the mappings address the issue of uncertainty in system identification. By *how well* we mean if the mapping is either a weak or a conservative measure of the uncertainty in the optimization problem.

The problem of properly measuring the uncertainty has been addressed in the theory of risk measures. In the theory of risk, the objective is to maximize the return of portfolios of assets which are subject to uncertainty. The uncertainty in this context is understood as the risk associated with the returns of financial portfolios of assets [46]. To properly measure the risk associated with portfolios, the notion of coherent measure of risk has been introduced [10]. Some of the requirements for a functional to be a coherent measure of risk are convexity and monotonicity, which imply that the resulting optimization problem is convex if the original problem with uncertainty is convex. In addition, a coherent measure of risk encourages diversification, i.e., it is always better to invest in several assets rather than in a single one, which is a property usually required by investors to reduce the risk associated with the portfolios.

Contribution

In this chapter we explore the connection between uncertainty in the theory of risk and uncertainty in system identification. In particular, we discuss how the notion of a coherent measure of risk can be employed to properly measure the risk of taking a decision under uncertainty in system identification. In addition, we introduce a coherent measure of risk that can be useful in system identification: the conditional value at risk (CVaR) [162]. The usefulness of coherent risk measures will be illustrated by its application to input design. However, we emphasize that the applicability of this approach is not only limited to this topic, it can also help to address the uncertainty issue in other areas (e.g., a risk theoretical approach has already been used in Markov control processes and model predictive control (MPC), see [42, 43, 175]).

We note that the contribution of this chapter (as well as the contribution of Chapter 9) lies in a Bayesian setting, as it requires information about the uncertainty in terms of a prior distribution.

Structure of the chapter

This chapter is organized as follows. Section 8.2 presents the problem of uncertainty in system identification. We analyze the problem of uncertainty from a risk theoretical perspective in Section 8.3. In Section 8.4, we explore the use of the conditional value at risk to measure uncertainty in system identification. A numerical illustration of the discussion is presented in Section 8.5. Finally, Section 8.6 presents some concluding remarks for this chapter.

8.2 The role of uncertainty in system identification

To illustrate the discussion, we consider a nonlinear state-space model with states $x_t \in \mathbb{R}^{n_x}$, inputs $u_t \in \mathbb{R}^{n_u}$, and measurements $y_t \in \mathbb{R}^{n_y}$ given by

$$x_t|x_{t-1} \sim f_\theta(x_t|x_{t-1}, u_{t-1}), \tag{8.1a}$$
$$y_t|x_t \sim g_\theta(y_t|x_t, u_t), \tag{8.1b}$$
$$x_0 \sim \mu_\theta(x_0), \tag{8.1c}$$

where $f_\theta(\cdot)$, $g_\theta(\cdot)$, and $\mu_\theta(\cdot)$ denote known pdfs parameterized by $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$. We assume that there exists a $\theta_0 \in \Theta$ such that (8.1) describes the true system when $\theta = \theta_0$, i.e., there is no undermodelling.

The main objective in system identification is to estimate the model parameters θ from the collected input-output data $\mathbf{Z}_T = \{(y_t, u_t)\}_{t=1}^T$ [122]. Since the estimated model parameter relies on the input-output data, it is typically required that the data must provide as much information from the process as possible, which implies that the model parameters can be estimated with maximum accuracy for a given experiment length T (cf. Section 1.3 on page 12). However, as we discuss below, the design of the input to maximize the information content about the process in the data usually depends on the true model parameters θ_0, which are uncertain prior to performing an experiment.

Input design

A standard approach to maximize the accuracy of the estimated model (8.1) is by optimizing a scalar function of the Fisher information matrix $\mathcal{I}_F^T$ [87, 122]. The Fisher information matrix is defined as

$$\mathcal{I}_F^T(u_{1:T}, \theta_0) = \mathbf{E}\left\{ \mathcal{S}_T(u_{1:T}, \theta_0)\mathcal{S}_T^\top(u_{1:T}, \theta_0) \middle| u_{1:T} \right\}, \tag{8.2a}$$
$$\mathcal{S}_T(u_{1:T}, \theta_0) = \nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta_0}. \tag{8.2b}$$

where $p_\theta(y_{1:T}|u_{1:T})$ is the likelihood function. We note that the expected value in (8.2a) is over the stochastic processes in (8.1).

The input design problem is to find an input sequence $u_{1:N}$, $u_t \in \mathcal{C} \subset \mathbb{R}$ which optimizes a scalar function of (8.2a), where $\mathcal{C}$ denotes the alphabet for u_t [186, 187].

The scalar function is given by $h\colon \mathbb{R}^{n_\theta \times n_\theta} \times \Theta \to \mathbb{R}$. To properly quantify $\mathcal{I}_F^T$, we require h to be a matrix convex function in its first argument.

The problem described can be summarized as

Problem 8.1 *Design an optimal input signal* $u_{1:T}^{\mathrm{opt}}$*, where*

$$u_{1:T}^{\mathrm{opt}} = \arg \min_{u_{1:T} \in \mathcal{C}^T} h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)\,, \tag{8.3}$$

with $h\colon \mathbb{R}^{n_\theta \times n_\theta} \times \Theta \to \mathbb{R}$ *a matrix convex function in its first argument, and* $\mathcal{I}_F^T(u_{1:T}, \theta_0) \in \mathbb{R}^{n_\theta \times n_\theta}$ *defined as in* (8.2)*.*

As we can see from (8.2), the main difficulty to solve Problem 8.1 is that the input design relies on the knowledge of the parameter θ_0 describing the true system, which will be estimated using the excitation to be designed as the solution of Problem 8.1. To solve this problem, a robust input design scheme has been proposed, where the input sequence is designed by incorporating the uncertainty on the model parameters into the optimization problem. However, so far, this scheme has only considered ad-hoc functions (e.g. the expected value of $h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)$ over Θ, or the supremum of $h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)$ over Θ), and they are usually considered without analyzing its consequences over the optimal decision. Therefore, there is a need for characterizing the set of functions that can be employed to measure the risk of designing inputs under uncertainty for system identification.

Application oriented input design

Another alternative to guarantee a prescribed accuracy in the model parameters is by imposing a quality constraint over the estimated parameters, while we minimize the experimental effort. This is the approach in least-costly and application oriented input design [18, 97, 113, 116].

To measure the quality of the estimated model parameters, we consider a function $J\colon \mathcal{C}^T \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$, which assesses the performance degradation when a particular parameter is employed in the model application, and is compared to the performance achieved by the true description. The purpose of this cost function is to incorporate the intended model application into the input design problem. On the other hand, the designed input sequence must minimize the experimental effort required to fulfill a specified quality constraint on the estimated model parameters. To this end, we introduce a function $H\colon \mathcal{C}^T \to \mathbb{R}$, which quantifies the required effort for a particular experiment $u_{1:T}$. We will assume that H is convex.

The application oriented input design problem can be summarized as

Problem 8.2 *Design an optimal input signal* $u_{1:T}^{\mathrm{opt}}$*, where*

$$\begin{aligned} u_{1:T}^{\mathrm{opt}} = \arg \min_{u_{1:T} \in \mathcal{C}^T} \quad & H(u_{1:T}) \\ \textit{subject to} \quad & J(u_{1:T}, \theta, \theta_0) \leq 0\,, \end{aligned} \tag{8.4}$$

with $H\colon \mathcal{C}^T \to \mathbb{R}$ *a convex function, and* $J\colon \mathcal{C}^T \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$*.*

As with Problem 8.1, Problem 8.2 has the difficulty that the optimal solution depends on the true model parameters θ_0. Therefore, an exact solution to Problem 8.2 cannot be achieved in practical applications, and a method to include the uncertainty in the optimization problem is required.

8.3 A risk theoretical approach to uncertainty

Measuring uncertainty is one of the challenges to be addressed in system identification. There are different approaches to uncertainty in the literature. However, there is a need on properly characterizing functions that measure the risk associated with taking a decision with limited information in system identification.

In this section we analyze the problem of uncertainty from a risk theoretical perspective. Quoting [161] "*risk is associated with having to make a decision without fully knowing its consequences*". The objective in risk theory is to determine the decision minimizing a cost under risk. The definition of cost can be very general, and is not tied to a particular application [161].

Remark 8.1 *We note that the notion of risk is also used in statistical decision theory [14], but its definition differs from the notion of risk considered here. Indeed, in [14, Section 1.3] the term Bayes risk is employed to refer to the expected value of a risk function associated with a decision rule.*

From a risk theoretical point of view, the cost $c\colon \mathbb{S} \times \Omega \to \mathbb{R}$ associates with each action $s \in \mathbb{S}$ (determined by the user) the value $c(s, \omega) \in \mathbb{R}$, where $\omega \in \Omega$ is a realization from the uncertainty associated with the cost. The set Ω is assumed to be a probability space with probability measure $\mathbf{P}$, where $\mathbf{P}$ must reflect the lack of knowledge or prior information regarding ω. Since c establishes a map from Ω to $\mathbb{R}$ for every $s \in \mathbb{S}$, we can interpret the cost as a random variable.

Example 8.1 *Consider Problem 8.1. The function h can be seen as the cost c, where $\mathbb{S} = \mathcal{C}$, and $\Omega = \Theta$. Here, we assume that Θ is a probability space with probability measure $\mathbf{P}_\Theta$.*

Example 8.2 *Consider Problem 8.2. The function J can be also interpreted as the cost c, where $\mathbb{S} = \mathcal{C}$, and $\Omega = \Theta \times \Theta$. Here, it is assumed that $\Theta \times \Theta$ is a probability space with probability measure $\mathbf{P}_{\Theta\times\Theta}$.*

To continue, we let $\mathcal{X}$ denote a scalar random variable on Ω for which its mean and standard deviation

$$\mu\{\mathcal{X}\} := \mathbf{E}\{\mathcal{X}\}\,, \tag{8.5a}$$

$$\sigma\{\mathcal{X}\} := \mathbf{E}\{(\mathcal{X} - \mu(\mathcal{X}))^2\}^{1/2}\,, \tag{8.5b}$$

are well defined. The expected values in (8.5) are with respect to the probability measure $\mathbf{P}$. We will denote by $\mathcal{L}^2$ the set of functions $\mathcal{X}$ such that (8.5) are finite. Here, we will assume that the function $c(s, \cdot)$ is in $\mathcal{L}^2$ for every $s \in \mathbb{S}$.

Example 8.3 *Consider Problem 8.1. The function h is in $\mathcal{L}^2$ provided that $\mu\{h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)\}$ and $\sigma\{h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)\}$ are finite for every $u_{1:T} \in \mathcal{C}^T$. In this case, the expected values in* (8.5) *are with respect to the probability measure $\mathbf{P}_\Theta$.*

The task is to measure the risk of loss associated with $\mathcal{X}$. To this end, for every $\mathcal{X} \in \mathcal{L}^2$ we associate a value $\mathcal{R}(\mathcal{X})$, where $\mathcal{R}\colon \mathcal{L}^2 \to (-\infty, \infty]$ is a functional. Note that $\mathcal{R}$ is allowed to take the value ∞.

A question that arises at this point is which properties $\mathcal{R}$ must satisfy to properly measure the risk associated with $\mathcal{X}$. This issue has been addressed in the literature of theory of risk, where the notion of coherent measure of risk has been developed [10]. Its definition is presented below [161]:

Definition 8.1 *A functional $\mathcal{R}\colon \mathcal{L}^2 \to (-\infty,\ \infty]$ is a* coherent measure of risk in the extended sense *if*

(i) $\mathcal{R}(C) = C$ *for all constant functions C,*

(ii) for $\mathcal{X}_1,\ \mathcal{X}_2 \in \mathcal{L}^2$ *and* $\lambda \in (0,\ 1)$,

$$\mathcal{R}((1-\lambda)\mathcal{X}_1 + \lambda\mathcal{X}_2) \leq (1-\lambda)\mathcal{R}(\mathcal{X}_1) + \lambda\mathcal{R}(\mathcal{X}_2)\,,$$

(iii) $\mathcal{R}(\mathcal{X}_1) \leq \mathcal{R}(\mathcal{X}_2)$ *when* $\mathcal{X}_1 \leq \mathcal{X}_2$,

(iv) $\mathcal{R}(\mathcal{X}) \leq 0$ *when* $\mathbf{E}\{(\mathcal{X}^k - \mathcal{X})^2\}^{1/2} \to 0$ *with* $\mathcal{R}(\mathcal{X}^k) \leq 0$ *for all k.*

The functional $\mathcal{R}$ will be called a coherent measure of risk in the basic sense *if it also satisfies*

(v) $\mathcal{R}(\lambda\mathcal{X}) = \lambda\mathcal{R}(\mathcal{X})$ *for* $\lambda > 0$.

Definition 8.1 characterizes the desired properties of measures of risk. Property (i) is motivated by the need that, if a random variable has a constant outcome C, the result of measuring its risk must be C. Property (ii) requires that $\mathcal{R}$ must be a convex functional, which follows from the fact that the risk associated with the convex combination of any two random variables must be always less or equal than the convex combination of the risk associated with each random variable separately. Property (iii) requires that $\mathcal{R}$ must associate a higher risk to those random variables having a higher cost. Property (iv) requires the closedness of $\mathcal{R}$. Finally, (v) requires $\mathcal{R}$ to be positive homogeneous.

If $\mathcal{R}$ is a coherent measure of risk in the basic sense, then by combining (ii) and (v) we obtain the subadditivity property: for all $\mathcal{X}_1,\ \mathcal{X}_2 \in \mathcal{L}^2$,

$$\mathcal{R}(\mathcal{X}_1 + \mathcal{X}_2) \leq \mathcal{R}(\mathcal{X}_1) + \mathcal{R}(\mathcal{X}_2)\,. \tag{8.6}$$

Equation (8.6) is an important result: it says that the risk obtained by combining two random variables is less than the sum of the risk associated to each random

variable separately, which leads to diversification. This result is in line with the intuition, as we expect that combining several possibilities leads to a lower risk than considering a single option.

From the perspective of Definition 8.1, we can analyze the approaches employed in system identification to measure uncertainty. It is easy to show that $\sup \mathcal{X}$ and $\mathbf{E}\{\mathcal{X}\}$ are coherent measures of risk in the basic sense, but the α-quantile of the distribution of $\mathcal{X}$ ($\alpha \in (0, 1)$), defined as

$$q_\alpha(\mathcal{X}) := \min \{z \in \mathbb{R} \colon \mathbf{P}\{\mathcal{X}(\omega) \leq z\} \geq \alpha\} \,, \tag{8.7}$$

is not a coherent measure of risk (property (iii) in Definition 8.1 is not satisfied). Notice that chance constrained optimization [142, 166] corresponds to constraining q_α. Hence, chance constrained optimization problems are incoherent.

Even though $\sup \mathcal{X}$ and $\mathbf{E}\{\mathcal{X}\}$ are coherent measures of risk in the basic sense, they have disadvantages. In the case of $\sup \mathcal{X}$, it can be infinity when $\mathcal{X}$ does not have a bounded support [152, Chapter 8]. On the other hand, $\mathbf{E}\{\mathcal{X}\}$ is a weak measure of risk, since it only imposes a requirement on average, and it can lead to realizations of $\mathcal{X}$ with poor results. In recent years, a new coherent measure of risk in the basic sense has been proposed: the conditional value at risk. Its definition is given in the next subsection.

Conditional value at risk

The notion of conditional value at risk has been proposed in [9]. Its definition is as follows. Given $\alpha \in (0, 1)$, the α-conditional value at risk is

$$\mathrm{CVaR}_\alpha(\mathcal{X}) := \frac{1}{1-\alpha} \int_{\{\omega \in \Omega \colon \mathcal{X}(\omega) \geq q_\alpha(\mathcal{X})\}} \mathcal{X}(\omega) dP(\omega) \,. \tag{8.8}$$

Note that the set of integration in (8.8) is defined in terms of (8.7). Expression (8.7) is referred to in the literature of theory of risk as value at risk, and it has been widely employed in this area [64, 65, 74]. The value at risk (8.7) is understood as the value $x \in \mathbb{R}$ such that $\mathcal{X}(\omega) \leq x$ is satisfied with probability α, i.e., it is the α-quantile associated to the distribution of $\mathcal{X}$. Even though (8.7) has been extensively used in theory of risk, it has been proved in [9] that value at risk is not a coherent measure of risk, which implies that it is not suitable for optimizing decisions under risk since it can lead to decisions that do not account appropriately for the distribution of the uncertainty.

The definition of conditional value at risk in (8.8) is the expected value of $\mathcal{X}$ with respect to the conditional distribution of its upper q_α-tail. Therefore, the minimization of (8.8) does not only guarantee that $\mathcal{X}^{\mathrm{opt}}(\omega) \leq q_\alpha(\mathcal{X}^{\mathrm{opt}})$ with probability α, but it also guarantees that, with probability $1 - \alpha$, the mean value of the loss will be small. In addition, it has been shown that (8.8) is a convex and monotone function of $\mathcal{X}$ [145], which is a property of coherent measures of risk in the basic sense.

Remark 8.2 *Note that the conditional value at risk* (8.8) *differs from the notion of mean excess function employed in the literature of risk. Indeed, Equation* (8.8) *is computed based on the α-quantile of $\mathcal{X}$, while the mean excess function computes the expected value of $\mathcal{X}(\omega) - d$ conditioned on $\mathcal{X}(\omega) \geq d$, where $d \in \mathbb{R}$ is given. Hence, the mean excess function is a less useful measure of risk than* (8.8) *in this context as the α-quantile of the distribution is not considered in the optimization.*

From Equation (8.8) we see that the optimization of the conditional value at risk requires the user defined parameter α. This parameter is chosen according to the confidence level desired by the user (typically $\alpha = 0.98$ in finance applications).

On the other hand, the optimization of (8.8) requires to solve (8.7), which can be difficult to compute. However, [162, Theorem 1] shows that it is possible to circumvent this issue:

Lemma 8.1 *(Computation of* CVaR_α*) Let* $\mathbf{P}$ *be the probability measure defined over Ω. If*

$$\Psi(s, C) := \int_{\{\omega \in \Omega \colon \mathcal{X}(s,\omega) \leq C\}} d\mathbf{P}(\omega)\,, \tag{8.9}$$

is everywhere continuous with respect to $C \in \mathbb{R}$, then

(i) The function

$$F_\alpha(s, C) := C + \frac{1}{1-\alpha}\mathbf{E}\left\{\max\left(0,\, \mathcal{X}(s,\omega) - C\right)\right\}\,, \tag{8.10}$$

is a convex and continuously differentiable function of $C \in \mathbb{R}$.

*(ii) The α-*CVaR *of $\mathcal{X}$ can be computed by*

$$\mathrm{CVaR}_\alpha(\mathcal{X}(s,\omega)) = \min_{C \in \mathbb{R}} \left[C + \frac{1}{1-\alpha}\mathbf{E}\left\{\max\left(0,\, \mathcal{X}(s,\omega) - C\right)\right\}\right]\,. \tag{8.11}$$

(iii) The set

$$A_\alpha(s) = \arg\min_{C \in \mathbb{R}} \left[C + \frac{1}{1-\alpha}\mathbf{E}\left\{\max\left(0,\, \mathcal{X}(s,\omega) - C\right)\right\}\right]\,, \tag{8.12}$$

is a nonempty, closed and bounded interval (perhaps reducing to a single point), and the α-value at risk of $\mathcal{X}$ is given by

$$q_\alpha(s) = \min_{z \in A_\alpha(s)} z\,. \tag{8.13}$$

Proof *We refer to the appendix in [162] for the proof of this Lemma.* ■

Lemma 8.1 shows that the computation of CVaR_α is reduced to a convex optimization problem, where the cost function is continuously differentiable with respect to the auxiliary variable $C \in \mathbb{R}$. This implies that the function (8.10) can be easily minimized as a function of $C \in \mathbb{R}$ by standard numerical solvers. In addition, the minimum of the solution set (8.12) corresponds to the α-value at risk, which implies that the α-quantile can be obtained without solving the nonconvex optimization (8.7). Finally, if it is assumed that samples $\{\omega_i\}_{i=1}^{N_{\mathrm{sim}}}$ can be drawn from the distribution $\mathbf{P}$, then the expected value in (8.11) can be approximated based on that

$$\mathbf{E}\left\{\max\left(0,\, \mathcal{X} - C\right)\right\} = \lim_{N_{\mathrm{sim}} \to \infty} \frac{1}{N_{\mathrm{sim}}} \sum_{i=1}^{N_{\mathrm{sim}}} \max\left(0,\, \mathcal{X}(\omega_i) - C\right) \tag{8.14}$$

holds with probability 1. Therefore, Equation (8.14) can be put in (8.11) to obtain an approximate value for the conditional value at risk, when the expected value in (8.11) is difficult to compute explicitly.

8.4 Measuring risk in system identification

In this section we illustrate how the conditional value at risk can be used to measure the risk associated with taking decision under uncertainty in system identification. To this end, we notice that the conditional value at risk is a convex and monotone function. Therefore, the composition of a convex function with the conditional value at risk results in a convex function [163, pp. 32]. As a consequence, the conditional value at risk can be used to obtain robust optimization problems that are convex, provided that the original problem is convex in the decision variables.

Robust input design

One fundamental assumption to solve the input design problem is that the true values of the model parameters should be available. One alternative to relax this assumption is to remove the dependence of the design with respect to the model parameters, which is the focus of robust input design. However, as discussed in Section 8.3, the common choices to measure uncertainty are either very conservative or weak.

Here we propose the use of conditional value at risk to obtain a robust input design problem. To this end, we assume that a probability density function of the model parameters is available to the user, denoted by p_Θ. Note that the assumption is close to the requirement in Bayesian techniques in the sense that prior information on the model parameters is taken into account in the optimization problem (cf. Chapter 7). However, the notion of posterior distributions plays no role here, as the prior distribution is only employed to compute the risk measure.

Following the discussion in Section 8.3, we can interpret h as a cost in $\mathcal{L}^2$ with respect to the probability measure given by p_Θ. Thus, we can use the conditional

value at risk to obtain a robust formulation of the input design problem as

$$u_{1:T}^{\text{opt}} = \arg \min_{u_{1:T} \in \mathcal{C}^T} \text{CVaR}_\alpha(h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)) \,, \tag{8.15}$$

where $\alpha \in (0, 1)$ is a given parameter. The intuition behind minimizing (8.15) is as follows: the mean value of $h(\mathcal{I}_F^T(u_{1:T}^{\text{opt}}, \theta_0), \theta_0)$ associated with its $1 - \alpha$ tail distribution will be equal to $\text{CVaR}_\alpha(h(\mathcal{I}_F^T(u_{1:T}^{\text{opt}}, \theta_0), \theta_0))$.

The optimization problem (8.15) is solved by employing (8.11), where $C \in \mathbb{R}$ is an additional decision variable. The resulting optimization is

$$u_{1:T}^{\text{opt}} = \arg \min_{\substack{u_{1:T} \in \mathcal{C}^T \\ C \in \mathbb{R}}} C + \frac{1}{1-\alpha} \mathbf{E}\{\max(0, h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0) - C)\} \,, \tag{8.16}$$

where the expected value in (8.16) is with respect to p_Θ. If the original problem (8.3) is convex in the decision variables, then the optimization problem (8.16) is also convex.

Robust application oriented input design

The application oriented input design problem also suffers from the uncertainty in the true model parameters. To solve this issue, we use the theory of risk approach to obtain a robust version of Problem 8.2. Under the assumption that the uncertainty in the true model parameters is modeled by a probability density function $p_{\Theta \times \Theta}$, the robust application oriented input design problem can be written as

$$\begin{aligned} u_{1:T}^{\text{opt}} = \arg \min_{u_{1:T} \in \mathcal{C}^T} \quad & H(u_{1:T}) \\ \text{subject to} \quad & \text{CVaR}_\alpha(J(u_{1:T}, \theta, \theta_0)) \leq 0 \,, \end{aligned} \tag{8.17}$$

where $\alpha \in (0, 1)$. Note that the constraint in (8.17) follows from the monotonicity of the conditional value at risk, and that $\text{CVaR}_\alpha(0) = 0$. The meaning of the constraint in (8.17) is as follows: the mean value of the performance degradation associated with the $1 - \alpha$ tail distribution of J must be less or equal than 0.

Using (8.11), (8.17) can be formulated as

$$\begin{aligned} u_{1:T}^{\text{opt}} = \arg \min_{\substack{u_{1:T} \in \mathcal{C}^T \\ C \in \mathbb{R}}} \quad & H(u_{1:T}) \\ \text{subject to} \quad & C + \frac{1}{1-\alpha} \mathbf{E}\{\max(0, J(u_{1:T}, \theta, \theta_0) - C)\} \leq 0 \,, \end{aligned} \tag{8.18}$$

where the expected value in (8.18) is with respect to $p_{\Theta \times \Theta}$. Finally, the resulting optimization problem will be convex if the original problem is convex.

Remark 8.3 *In Chapter 9, the issue of accounting for the uncertainty in application oriented input design is revisited, where we provide further details on how risk coherent measures can be used to formulate a robust problem. In particular, in Chapter 9 we discuss how to implement the risk coherent problem* (8.18) *in a tractable manner.*

8.5 Numerical example

Consider the discrete-time, LTI model

$$y_t = \frac{q^{-1}}{1 - 2r\cos(\theta)q^{-1} + r^2 q^{-2}} u_t + e_t \,, \tag{8.19}$$

where $\theta \in [0, \pi]$, and $\{e_t\}$ is Gaussian white noise of zero mean and unit variance. We assume that $r = 0.95$ is known.

In this example, we are interested in identifying the location of the resonance frequency θ. To this end, we design an input sequence $u_{1:T}$ minimizing $h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0) = -\log\det(\mathcal{I}_F^T(u_{1:T}, \theta_0))$. We assume that $u_{1:T}$ is a realization of a stationary process with zero mean, and power spectrum given by

$$\Phi_u(\omega) = \sum_{\tau=-20}^{20} c_{|\tau|} e^{j\omega\tau} \,, \tag{8.20}$$

where $c_{0:20} \in \mathbb{R}^{21}$ are the design variables. Due to power constraints on the input, the optimal input design must satisfy $\mathbf{E}\{u_t^2\} \leq 5$.

To address the uncertainty issue of the parameter θ, we assume that θ is uniformly distributed over [0, π].

Finally, the optimization problem is given by[1]

$$\begin{aligned} \min_{\Phi_u} \quad & \mathcal{R}(h(\mathcal{I}_F^T(u_{1:T}, \theta_0))) \\ \text{subject to} \quad & \Phi_u(\omega) = \sum_{\tau=-20}^{20} c_{|\tau|} e^{j\omega\tau} \\ & \Phi_u(\omega) \geq 0, \ \text{ for all } \omega \in [-\pi, \pi] \,, \\ & \mathbf{E}\{u_t^2\} \leq 5 \,, \end{aligned} \tag{8.21}$$

where $\mathcal{R}$ is a measure of risk over the uncertainty in θ. We consider four cases for $\mathcal{R}$:

- Case i: $\mathcal{R}(h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)) = h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)$, $\mathcal{I}_F^T$ computed at the nominal value $\theta_0 = \pi/2$.
- Case ii: $\mathcal{R}(h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)) = \mathbf{E}\{h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)\}$.
- Case iii: $\mathcal{R}(h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)) = q_\alpha(h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0))$ (value at risk), with $\alpha = 0.98$.
- Case iv: $\mathcal{R}(h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)) = \mathrm{CVaR}_\alpha(h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0))$, with $\alpha = 0.98$.

[1] We refer to Appendix E for the details on the implementation of nonnegative constraint on Φ_u and the computation of the Fisher information matrix.

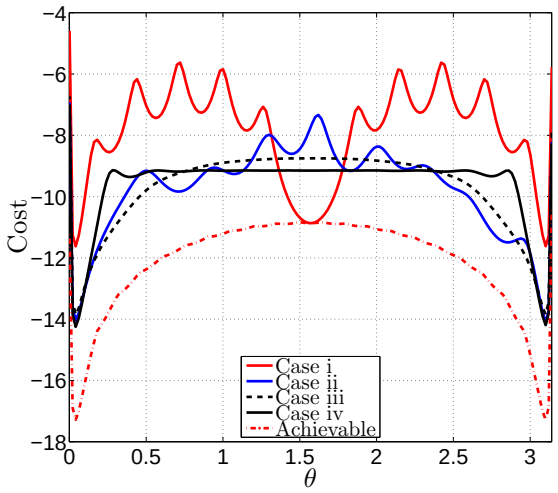

Figure 8.1: $h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)$ for $\theta_0 \in [0, \pi]$.

For Cases ii-iv we solve an approximation of (8.21), by replacing $\mathbf{E}\{h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)\}$, the expected value in (8.11), and the probability in (8.7) by their Monte-Carlo approximations, with $N_{\text{sim}} = 200$ realizations for the model parameter θ. The optimization problem (8.21) for Cases i, ii, and iv is solved using the `cvx` toolbox available for Matlab [89], and Case iii is solved using the command `fmincon` in Matlab (since the problem is nonconvex).

Figure 8.1 shows $h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)$ computed for different values of $\theta_0 \in [0, \pi]$, when the input is given by the solution of Cases i-iv, and the achievable cost for the optimal design assuming that the nominal value θ_0 coincides with the true parameter, for $\theta_0 \in [0, \pi]$. From this figure we can see that the optimal input for Case i results in informative experiments as long as θ_0 is close to the nominal value $\pi/2$, but it can be poor if θ_0 is not close to $\pi/2$. The behavior of $h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)$ is due to that the location of the resonance θ coincides with local minima and maxima of the optimal power spectrum $\Phi_u(\omega)$ for Case i. Case ii helps to obtain a more robust design, but results in poor experiments for θ close to the nominal value θ_0. Case iii reduces the value of $h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)$ even further for the poor experiments when compared to Case ii. Finally, Case iv improves the results obtained with Case iii for θ resulting in poor values of $h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)$. Moreover, Case iv guarantees the convergence to the optimal solution due to the convexity of the problem, which is not guaranteed by solving Case iii. The design in Case iv results in a more constant behavior for $h(\mathcal{I}_F^T(u_{1:T}, \theta_0), \theta_0)$ over $\theta_0 \in [0, \pi]$, except for the values of θ_0 close to 0 and π, where the experiment is more informative. As usual, the robustification in Case iv is obtained by sacrificing the information obtained for values of θ leading to

good experiments with the more non-robust approaches.

In conclusion, the example illustrates that the conditional value at risk can be useful to design robust experiments for identification of dynamical systems.

8.6 Conclusion

Motivated by the difficulty of properly accounting for the uncertainty in system identification problems, this chapter explores the use of a risk theoretical perspective to measure the risk of taking decisions under uncertainty. The notion of coherent measure of risk has been introduced, and it has been shown how this notion can be used to obtain a better understanding of the role of risk measures in system identification. To illustrate the usefulness of coherent measures of risk, the definition of conditional value at risk has been presented, and applications of this function to input design have been shown.

Chapter 9

Risk coherent application oriented input design

In Chapter 8 we have seen that the notion of coherent measures of risk can provide an approach to properly account for the uncertainty in system identification. This chapter extends the discussion in that direction, where we present a methodology to use coherent measures of risk in application oriented input design.

9.1 Introduction

As already mentioned in Chapter 8, uncertainty is a frequent issue in engineering and science. Since uncertainty corresponds to a lack of knowledge about a phenomenon, it can be interpreted in different manners depending on the subject of interest. As exemplified in Chapter 8, fields where the uncertainty plays a crucial role are robust control and portfolio optimization [74, 143, 215].

In system identification, one of the sources of uncertainty is the lack of knowledge about the location of the parameter defining the true system. Under suitable assumptions on the model and the parameter set, it is possible to characterize such uncertainty by the use of the Central limit theorem [122]. This information can be then used to provide confidence bounds on the location of the estimated parameter in the parameter set, which has been the main approach used in system identification to account for uncertainty [18, 96].

A topic in system identification where the uncertainty plays an important role is input design. The main issue in input design is that it usually relies on prior information about the model parameters. This difficulty is often addressed by (i) adaptive schemes, where collected information is employed to update the input sequence [80, 151], or (ii) robust schemes, where the uncertainty on the model parameters is included in the problem formulation [114, 167].

In regards to robust approaches to input design, we can quantify the uncertainty in different manners. For example, we can simply consider a nominal parameter

to design the experiment [186], use chance constraints to guarantee a prescribed accuracy on the estimated parameter [166], or a worst case scenario, where the input is designed by considering the parameter in the set delivering the worst performance [164]. Even though many approaches have been considered to account for the uncertainty on the model parameter, there is a need for properly characterizing the functions that measure the risk associated with taking decisions under uncertainty.

On the other hand, the issue of properly accounting for the uncertainty in optimization problems has been well studied in the risk theory literature [9, 17]. The focus on risk theory is on minimizing the losses due to uncertainty in the returns of a portfolio of assets. To this end, the notion of coherent measures of risk has been introduced [10, 161], namely, to provide a systematic approach to consider the uncertainty in the returns of the assets.

Contribution

In this chapter we discuss how the notion of coherent measures of risk can be used to properly account for the uncertainty in application oriented input design (AOID). A first discussion in this direction was presented in Chapter 8. However, the discussion in Chapter 8 only mentions the use of coherent measures of risk in application oriented input design, and does not provide a guideline on how to incorporate them in existing frameworks. Here we discuss how risk coherent measures can be properly included in the problem of application oriented input design. We also show that there exist difficulties in implementing such formulation, as the decision variables influence the distribution over which the risk coherent measure is computed. To circumvent this difficulty, we propose a stochastic approximation method to find a sub-optimal input satisfying the constraints. A convergence proof of the proposed method is presented.

Structure of the chapter

The structure of this chapter is as follows. Section 9.2 introduces the problem of uncertainty in AOID. Section 9.3 presents the risk theoretical approach in the context of AOID. Section 9.4 considers the use of risk coherent measures to account for the uncertainty in AOID, and formulates a stochastic approximation algorithm to obtain a sub-optimal solution to the proposed problem. A numerical illustration is presented in Section 9.5. Finally, Section 9.6 presents concluding remarks.

9.2 The role of uncertainty in application oriented input design

Consider the discrete time, linear time invariant, multiple input-multiple output (MIMO) system

$$y_t = G_o(q)u_t + H_o(q)e_t\,, \tag{9.1}$$

where $u_t \in \mathbb{R}^{n_u}$ is the input, $y_t \in \mathbb{R}^{n_y}$ is the measured output and $\{e_t\}$ ($e_t \in \mathbb{R}^{n_y}$) is Gaussian white noise of zero mean and covariance matrix $\Sigma_e \succ 0$. The transfer functions G_o and H_o are real rational functions in the time shift operator q. In the following, we assume that G_o, H_o and H_o^{-1} are stable, and that

$$\lim_{q\to\infty} H_o(q) = I\,. \tag{9.2}$$

System identification

The objective in system identification is to estimate a model for the system (9.1) based on the input-output data $\mathbf{Z}_T := \{(y_t, u_t)\}_{t=1}^T$ and the model

$$y_t = G_\theta(q)u_t + H_\theta(q)e_t\,, \tag{9.3}$$

where the pair (G_θ, H_θ) is a real rational transfer function in q parameterized by $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$. We assume that the pair (G_θ, H_θ) satisfies the same assumptions as (G_o, H_o). In addition, we assume that there exists $\theta_0 \in \Theta$ such that if $\theta = \theta_0$ the model (9.3) coincides with the system (9.1), i.e., there is no undermodelling.

One of the most applied techniques to estimate $\hat{\theta}_T \in \Theta$ based on $\mathbf{Z}_T$ is the prediction error method (PEM)

$$\hat{\theta}_T := \arg\min_{\theta\in\Theta} \frac{1}{2}\sum_{t=1}^{T} \varepsilon_t^\top(\theta)\Sigma_e^{-1}\varepsilon_t(\theta)\,, \tag{9.4}$$

where ε_t corresponds to the one step ahead prediction error,

$$\varepsilon_t(\theta) := y_t - \hat{y}_{t|t-1}(\theta)\,, \tag{9.5}$$

and $\hat{y}_{t|t-1}(\theta)$ is the one step ahead predictor of (9.3),

$$\hat{y}_{t|t-1}(\theta) := H_\theta^{-1}(q)\,G_\theta(q)u_t + (I - H_\theta^{-1}(q))y_t\,. \tag{9.6}$$

Under some regularity conditions on the system (9.1), it can be shown that the estimator (9.4) is consistent [122]. Moreover, under the Gaussian assumption on $\{e_t\}$, it can be also shown that the estimator (9.4) is asymptotically efficient (cf. Definition 1.1 on page 11). This means that the covariance matrix of $\hat{\theta}_T$ attains the Cramér-Rao bound:

$$\lim_{T\to\infty} \mathbf{E}\left\{ T\left(\hat{\theta}_T - \theta_0\right)\left(\hat{\theta}_T - \theta_0\right)^\top \right\} = \overline{\mathcal{I}}_F^{-1}(\theta_0)\,, \tag{9.7}$$

where

$$\overline{\mathcal{I}}_F(\theta_0) := \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\left\{ \nabla_\theta \hat{y}_{t|t-1}(\theta)\big|_{\theta=\theta_0}\, \Sigma_e^{-1}\, \nabla_\theta^\top \hat{y}_{t|t-1}(\theta)\big|_{\theta=\theta_0} \right\}. \tag{9.8}$$

The expected value in (9.8) is with respect to the stochastic process $\{e_t\}$. Expression (9.8) assumes that $\{u_t\}$ is a quasi-stationary process [122]. The covariance matrix (9.8) is usually referred to as the Fisher information matrix (FIM).

A key result for $\hat{\theta}_T$ is the central limit theorem, which states that

$$\sqrt{T}\left(\hat{\theta}_T - \theta_0\right) \xrightarrow{d} \mathcal{N}(0, \overline{\mathcal{I}}_F^{-1}(\theta_0))\,. \tag{9.9}$$

The result (9.9) is useful to derive confidence sets for the estimated parameter $\hat{\theta}_T$. Indeed, from (9.9) we can conclude that

$$\left(\hat{\theta}_T - \theta_0\right)^\top T\,\overline{\mathcal{I}}_F(\theta_0)\left(\hat{\theta}_T - \theta_0\right) \tag{9.10}$$

is asymptotically Chi-squared distributed with n_θ degrees of freedom. Therefore, the set

$$\Theta_{\mathrm{id}}(\alpha) := \left\{\theta \in \Theta\colon\ (\theta - \theta_0)^\top T\,\overline{\mathcal{I}}_F(\theta_0)\,(\theta - \theta_0) \leq \chi^2_\alpha(n_\theta)\right\} \tag{9.11}$$

represents a subset of Θ where $\hat{\theta}_T$ lies with probability at least α ($\alpha \in (0,1)$). In (9.11), $\chi^2_\alpha(n_\theta)$ represents the α quantile of a Chi-squared distributed random variable with n_θ degrees of freedom. In the following, the set (9.11) will be referred to as the *identification set*.

From (9.11), we note that the identification set depends on the location of the parameter θ_0 describing the system (9.1), which is typically unknown before performing an experiment. Therefore, any method using the identification set (9.11) also requires to take the uncertainty on the location of θ_0 into account.

The application set

From the user perspective, the estimated parameter $\hat{\theta}_T$ must ensure that a certain performance is achieved when the model (9.3) with $\theta = \hat{\theta}_T$ is employed in the application. To quantify the performance achieved by $\hat{\theta}_T$, we introduce the *application cost* $V_{\mathrm{app}}\colon \Theta \times \Theta \to \mathbb{R}_0^+$ measuring the degradation in performance when a parameter $\theta \neq \theta_0$ is employed in the application. In the following, we assume that the application cost is such that $V_{\mathrm{app}}(\theta, \theta_0) > 0$ if $\theta \neq \theta_0$, and $V_{\mathrm{app}}(\theta, \theta_0) = 0$ if and only if $\theta = \theta_0$.

Based on the application cost, and for a given $\gamma_0 > 0$, the set

$$\Theta_{\mathrm{app}}(\gamma_0) := \left\{\theta \in \Theta\colon\ V_{\mathrm{app}}(\theta, \theta_0) \leq \frac{1}{\gamma_0}\right\} \tag{9.12}$$

defines the parameters in Θ with acceptable performance degradation in the application, specified by the parameter γ_0. In the following, we refer to the set (9.12) as the *application set*.

As with the identification set, the application set (9.12) is defined by employing the parameter θ_0 describing the system (9.1). Hence, any method using the application set (9.12) must take the uncertainty on θ_0 into account.

At this point we need to emphasize that the uncertainty associated with θ_0 is different from the one on $\hat{\theta}_T$. In the case of θ_0, the uncertainty is described through the information (or belief) about the location of θ_0 in Θ prior to performing the experiment. On the other hand, the uncertainty on $\hat{\theta}_T$ obeys the central limit theorem (9.9) for a particular choice of θ_0.

Application oriented input design

The objective in application oriented input design is to design the input signal $u_{1:T}$ to be applied to system (9.1), such that the estimated parameter $\hat{\theta}_T$ satisfies the performance requirements in the application and minimizes an experimental cost $H\colon \mathbb{R}^{n_u \times T} \to \mathbb{R}$ [8, 63, 112]. The satisfaction of performance requirements is usually achieved by requiring the identification set (9.11) to be a subset of the application set (9.12). Hence, the application oriented input design problem can be formulated as follows:

Problem 9.1 *Design* $u_{1:T}^{\text{opt}}$ *as*

$$\begin{aligned} u_{1:T}^{\text{opt}} = \quad & \arg \min_{u_{1:T} \in \mathbb{R}^{n_u \times T}} \quad H(u_{1:T}) \\ & \text{subject to} \qquad \Theta_{\text{id}}(\alpha) \subseteq \Theta_{\text{app}}(\gamma_0)\,. \end{aligned} \tag{9.13}$$

The experimental cost H in Problem 9.1 refers to a user defined cost. For example, we can minimize the experimental length T or the input power $\mathbf{E}\{u_t^2\}$ (if we assume that $\{u_t\}$ is a zero mean wide sense stationary process). In the following, we require H to be a convex function over the variables parameterizing $u_{1:T}$.

Problem 9.1 is usually nonconvex due to the set constraint, and difficult to solve as the set constraint usually translates into an infinite number of inequalities. Instead, Problem 9.1 is usually relaxed to a convex formulation by introducing a convex function $J\colon \mathbb{R}^{n_u \times T} \times \Theta \times \Theta \to \mathbb{R}$ approximating the constraint $\Theta_{\text{id}}(\alpha) \subseteq \Theta_{\text{app}}(\gamma_0)$. Some examples of convex relaxations of Problem 9.1 are the ellipsoidal approximation and the scenario approach [112, 196].

Based on the convex relaxations, the convex formulation of Problem 9.1 can be stated as

Problem 9.2 *Design* $u_{1:T}^{\text{opt}}$ *as*

$$\begin{aligned} u_{1:T}^{\text{opt}} = \quad & \arg \min_{u_{1:T} \in \mathbb{R}^{n_u \times T}} \quad H(u_{1:T}) \\ & \text{subject to} \qquad J(u_{1:T}, \hat{\theta}_T, \theta_0) \leq \frac{1}{\gamma_0}\,, \end{aligned} \tag{9.14}$$

where $H\colon \mathbb{R}^{n_u \times T} \to \mathbb{R}$ *is the experimental cost and* $J\colon \mathbb{R}^{n_u \times T} \times \Theta \times \Theta \to \mathbb{R}$ *is a quality constraint on the parameter estimate* $\hat{\theta}_T$*.*

In Problem 9.2 we assume (without loss of generality) a constraint on J in terms of the parameter γ_0 defining the application set (9.12), as J can account for different approaches to handle the set constraint in Problem 9.1.

Example 9.1 *If the application cost V_{app} is two times differentiable function on Θ with respect to its first argument, satisfying $\nabla^2_\theta V_{\text{app}}(\theta_0, \theta_0) \succeq 0$, then an ellipsoidal approximation of the set constraint in Problem 9.1 can be written as (cf. Appendix G)*

$$\overline{\mathcal{I}}_F(\theta_0) \succeq \frac{\chi^2_\alpha(n_\theta)}{T} \frac{\gamma_0}{2} \nabla^2_\theta V_{\text{app}}(\theta_0, \theta_0)\,, \tag{9.15}$$

where $\chi^2_\alpha(n_\theta)$ denotes the α-quantile of a Chi-squared distribution with n_θ degrees of freedom, and $\alpha \in (0, 1)$. The linear matrix inequality (9.15) *can be rewritten in the form of the constraint in Problem 9.2 as*

$$\lambda_{\max}\left(\frac{\chi^2_\alpha(n_\theta)}{T} \frac{\gamma_0}{2} \nabla^2 V_{\text{app}}(\theta_0, \theta_0) - \overline{\mathcal{I}}_F(\theta_0) + \frac{1}{\gamma_0} I\right) \leq \frac{1}{\gamma_0}\,, \tag{9.16}$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of its argument.

From the constraints in Problems 9.1-9.2 we conclude that the optimization is affected by uncertainty on the location of both the estimated parameter $\hat{\theta}_T$, and the parameter θ_0 describing the system (9.1). This is a crucial element in application oriented input design, as it can greatly affect the optimal solution $u^{\text{opt}}_{1:T}$. In the next section, we explain how the elements in theory of risk can provide a systematic solution to the uncertainty problem.

9.3 Uncertainty and risk theory

As highlighted in the previous section, the main difficulty for solving Problem 9.2 is the uncertainty on both the estimated parameter $\hat{\theta}_T$ and the nominal parameter θ_0. Efforts in relaxing the exact knowledge of θ_0 have been made considering, e.g., iterative input design and robust formulations [80, 167]. However, a proper characterization of the risk associated with taking actions under uncertainty in Problem 9.2 is missing.

Adopting the risk theoretical perspective of Chapter 8, we consider the function $c\colon \mathbb{S} \times \Omega \to \mathbb{R}$ that associates every action $s \in \mathbb{S}$ with the cost $c(s, \omega)$, where $\omega \in \Omega$ corresponds to a realization of the uncertainty associated with the cost.

As in Chapter 8, we require Ω to be a probability space with probability measure $\mathbf{P}$. As the function c maps Ω into $\mathbb{R}$ for every $s \in \mathbb{S}$, we can interpret c as a random variable.

Example 9.2 *Consider Problem 9.2. The function J can be seen as the cost c, where $\mathbb{S} = \mathbb{R}^{n_u \times N}$, and $\Omega = \Theta \times \Theta$. Here, it is assumed that $\Theta \times \Theta$ is a probability*

space with probability measure $\mathbf{P} = \mathbf{P}_{\hat{\Theta}\times\Theta_0}$. *In this case, the probability measure can be rewritten as*

$$\mathbf{P}_{\hat{\Theta}\times\Theta_0} = \mathbf{P}_{\hat{\Theta}|\Theta_0}\mathbf{P}_{\Theta_0}\,, \tag{9.17}$$

where $\mathbf{P}_{\hat{\Theta}|\Theta_0}$ *corresponds to the conditional probability measure of* $\hat{\theta}_T$ *given* θ_0 *(which is the probability measure associated with a normal distribution given by* $\mathcal{N}(\theta_0, T^{-1}\overline{\mathcal{I}}_F^{-1}(\theta_0))$*), and* $\mathbf{P}_{\Theta_0}$ *corresponding to the probability measure associated with* θ_0 *prior to using* $u_{1:T}^{\text{opt}}$ *in the system* (9.1). *We note that* $\overline{\mathcal{I}}_F(\theta_0)$ *depends on the input* $\{u_t\}$*, as it is indicated in Appendix E.*

Remark 9.1 *The requirement of a probability measure associated with* θ_0 *in Example 9.2 can be interpreted as a Bayesian framework. The probability measure* $\mathbf{P}_{\Theta_0}$ *accounts for information about the true system prior to perform an identification experiment. This information can be available through, e.g., historical data available about the system to be identified.*

To use the risk theoretical framework, we require that, for every $s \in \mathbb{S}$, the random variable c belongs to the class of random variables $\mathcal{X}\colon \Omega \to \mathbb{R}$ satisfying that $\mu\{\mathcal{X}\}$ and $\sigma\{\mathcal{X}\}$ are well defined and finite, i.e., that $\mathcal{X} \in \mathcal{L}^2$ (cf. Equation (8.5) on page 149).

Example 9.3 *Consider Problem 9.2. The function* J *is in* $\mathcal{L}^2$ *provided that* $\mu\{J(u_{1:T}, \hat{\theta}_T, \theta_0)\}$ *and* $\sigma\{J(u_{1:T}, \hat{\theta}_T, \theta_0)\}$ *are finite for every action* $u_{1:T} \in \mathbb{R}^{n_u \times T}$. *In this case, the expected values in* (8.5) *are with respect to the probability measure* $\mathbf{P}_{\hat{\Theta}\times\Theta_0}$.

The task is to measure the risk of loss associated with $\mathcal{X}$. To this end, for every $\mathcal{X} \in \mathcal{L}^2$ we associate a value $\mathcal{R}(\mathcal{X})$, where $\mathcal{R}\colon \mathcal{L}^2 \to (-\infty, \infty]$ is a functional. We note that $\mathcal{R}$ is allowed to take the value ∞.

As discussed in Chapter 8, it is of importance to determine the set of properties that $\mathcal{R}$ must satisfy to properly account for the uncertainty. This led to the notion of coherent measures or risk presented in Definition 8.1 on page 150. From Definition 8.1, we see that the a risk coherent functional $\mathcal{R}$ is convex and preserves inequalities, among other properties. The most important property of a risk coherent measure is that it promotes diversification (cf. Equation (8.6) on page 150).

It is shown in Section 8.3 on page 149 that $\sup \mathcal{X}$ and $\mathbf{E}\{\mathcal{X}\}$ are coherent measures of risk in the basic sense, but that the α-quantile of the distribution of $\mathcal{X}$ ($\alpha \in (0, 1)$), defined as in Equation (8.7) is not.

The problem of using q_α to account for the uncertainty is illustrated next.

Example 9.4 *Consider the function* J *in Problem 9.2. To account for the uncertainty in* $(\hat{\theta}_T, \theta_0)$*, we can consider the inequality*

$$q_\alpha(J(u_{1:T}, \hat{\theta}_T, \theta_0)) \leq \frac{1}{\gamma_0}\,, \tag{9.18}$$

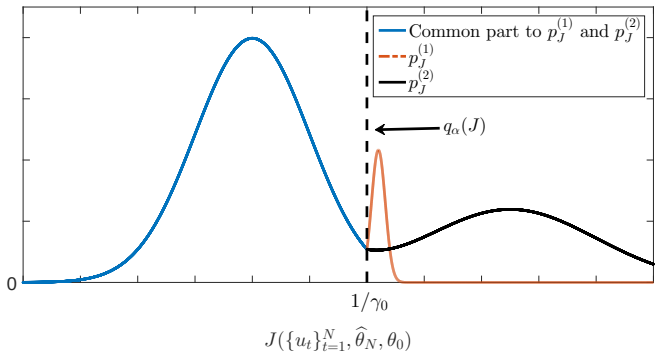

Figure 9.1: Possible values of p_J, Example 9.4.

for a fixed $\alpha \in (0,1)$. *The inequality* (9.18) *requires that the original inequality constraint in Problem 9.2 is satisfied with probability at least* α.

If Problem 9.2 is solved by replacing the inequality $J(u_{1:T}, \hat{\theta}_T, \theta_0) \leq 1/\gamma_0$ *by* (9.18) *we can obtain an undesired effect on the optimal solution. In the following, we study more in detail the effect of using* (9.18) *in the optimization.*

If we assume that J *has probability density function* p_J, *then* p_J *can adopt the values presented in Figure 9.1, where* $p_J^{(1)}$ *and* $p_J^{(2)}$ *correspond to the pdfs associated with* $J(u_{1:T}^{(1)}, \cdot, \cdot)$ *and* $J(u_{1:T}^{(2)}, \cdot, \cdot)$, *respectively. Here,* $u_{1:T}^{(1)}$ *and* $u_{1:T}^{(2)}$ *correspond to two different input realizations in* $\mathbb{R}^{n_u \times T}$. *In this case, both pdfs lead to that the constraint* (9.18) *is satisfied. However, it is clear from Figure 9.1 that* $p_J^{(1)}$ *gives a better scenario than* $p_J^{(2)}$ *for values of* J *larger than* γ_0^{-1}, *since the probability of large values for* J *is smaller for* $p_J^{(1)}$ *than for* $p_J^{(2)}$. *This information is not captured by the inequality* (9.18), *which is problematic if both* $u_{1:T}^{(1)}$ *and* $u_{1:T}^{(2)}$ *are optimal solutions to the problem considered here.*

As for q_α, we can also find disadvantages of using $\sup \mathcal{X}$ and $\mathbf{E}\{\mathcal{X}\}$ to account for the uncertainty. Recalling the discussion in Section 8.3, we see that $\sup \mathcal{X}$ can lead to very conservative results or even unfeasible problems, while $\mathbf{E}\{\mathcal{X}\}$, it is a poor measure of risk as it considers only an average value, which can lead to realizations of $\mathcal{X}$ with very poor performance.

An alternative to standard measures of risk is provided by the conditional value at risk, defined in Equation (8.8). Recalling the discussion in Section 8.3, given $\alpha \in (0,1)$, the α-conditional value at risk corresponds to the expected value of $\mathcal{X}$ with respect to the conditional distribution of its upper q_α-tail. Therefore, if (8.8) is used in a problem to constrain a function J subject to uncertainty, the constraint

involving (8.8) will not only guarantee that $J \leq 1/\gamma_0$ with probability α, but it will also guarantee that, if $J > q_\alpha(J)$, the mean value of the loss will be at most $1/\gamma_0$.

Example 9.5 *Consider Example 9.4. For the pdfs presented in Figure 9.1, we see that the α-conditional value at risk associated with J is smaller for $p_J^{(1)}$ than for $p_J^{(2)}$. Hence, the α-conditional value at risk captures more information about the effects of the uncertainty on J than the α-quantile q_α.*

In the next section we discuss how risk coherent measures can be used in application oriented input design.

9.4 Accounting for the uncertainty in application oriented input design

As it has been mentioned, the application oriented input design problem in (9.2) has two sources of uncertainty: uncertainty on the location of the estimated parameter $\hat{\theta}_T$, which is characterized asymptotically by (9.9), and uncertainty on the parameter θ_0 describing the true system, which is characterized by a prior distribution on Θ.

To account for the uncertainty in Problem 9.2, we replace the nominal constraint by

$$\mathcal{R}_{\hat{\Theta},\Theta_0}\left(J\left(u_{1:T},\hat{\theta}_T,\theta_0\right)\right) \leq \frac{1}{\gamma_0}\,, \tag{9.19}$$

where $\mathcal{R}_{\hat{\Theta},\Theta_0}\colon \mathcal{L}^2 \to \mathbb{R}$ is a coherent measure of risk. The sub-indices $\hat{\Theta},\Theta_0$ in the risk measure in (9.19) indicate that the risk measure is computed with respect to the joint distribution of $(\hat{\theta}_T,\theta_0)$.

Based on (9.19), the application oriented input design problem can be rewritten as:

Problem 9.3 *Design $u_{1:T}^{\text{opt}}$ as*

$$\begin{aligned} u_{1:T}^{\text{opt}} = \quad & \arg \min_{u_{1:T}\in\mathbb{R}^{n_u\times T}} \quad H(u_{1:T}) \\ & \text{subject to} \qquad \mathcal{R}_{\hat{\Theta},\Theta_0}\left(J\left(u_{1:T},\hat{\theta}_T,\theta_0\right)\right) \leq \frac{1}{\gamma_0}\,, \end{aligned} \tag{9.20}$$

with $\mathcal{R}_{\hat{\Theta},\Theta_0}\colon \mathcal{L}^2 \to \mathbb{R}$ a coherent measure of risk.

As the constraint in Problem 9.3 accounts for the uncertainty of both $\hat{\theta}_T$ and θ_0, the joint distribution of $(\hat{\theta}_T,\theta_0)$ is required. However, the distribution of $\hat{\theta}_T$ depends on the input signal, which is the decision variable in Problem 9.3. Unless assumptions on the dynamical model and the input characteristics are made, the dependence of the distribution of $\hat{\theta}_T$ on the input signal makes Problem 9.3 difficult to solve.

To circumvent the issue with the joint distribution of $(\hat{\theta}_T, \theta_0)$, we propose an iterative approach to satisfy the constraint in (9.20). The method iteratively updates the parameter $\gamma > 0$ defining the application set $\Theta_{\text{app}}(\gamma)$ by using a stochastic approximation method. At every iteration, we proceed in three steps. First, we solve an approximation of Problem 9.3 by only accounting for the uncertainty on θ_0. We denote by $u_{1:T}^{\gamma}$ the solution of the resulting approximate problem. In a second step, we compute $\mathcal{R}_{\hat{\Theta},\Theta_0}(J)$ based on $u_{1:T}^{\gamma}$ and the pdf of $(\hat{\theta}_T, \theta_0)$. Finally, the third step updates the value of γ to use in the next iteration by employing a stochastic approximation method. The objective of this procedure is to converge to a value $\gamma^\star$ such that risk coherent measure of J based on $u_{1:T}^{\gamma^\star}$ satisfies the inequality in Problem 9.3. We note that $u_{1:T}^{\gamma^\star}$ is not necessarily the optimal solution to Problem 9.3: $u_{1:T}^{\gamma^\star}$ is only guaranteed to be a feasible choice for the optimization in Problem 9.3.

A detailed description of the proposed method at iteration k is as follows:

1. Find $u_{1:T}^{\gamma_k}$ by solving

$$\begin{aligned} u_{1:T}^{\gamma_k} = \quad & \arg \min_{u_{1:T} \in \mathbb{R}^{n_u \times T}} \quad H(u_{1:T}) \\ & \text{subject to} \qquad \mathcal{R}_{\Theta_0}\left(J\left(u_{1:T}, \hat{\theta}_T, \theta_0\right)\right) \leq \frac{1}{\gamma_k}\,, \\ & \qquad\qquad\qquad\quad \text{for all } \hat{\theta}_T \in \Theta_{\text{app}}(\gamma_k)\,, \end{aligned} \tag{9.21}$$

   where $\mathcal{R}_{\Theta_0}(\cdot)$ is computed over the uncertainty associated with θ_0.

2. Given $u_{1:T}^{\gamma_k}$, compute $\mathcal{R}_{\hat{\Theta},\Theta_0}\left(J\left(u_{1:T}^{\gamma_k}, \hat{\theta}_T, \theta_0\right)\right)$.

3. For a predefined parameter $\beta > 0$, if the inequality

$$\left|\mathcal{R}_{\hat{\Theta},\Theta_0}\left(J\left(u_{1:T}^{\gamma_k}, \hat{\theta}_T, \theta_0\right)\right) - \frac{1}{\gamma_0}\right| \leq \beta\,, \tag{9.22}$$

   holds, then the algorithm terminates. Otherwise, compute

$$\gamma_{k+1} = x_{k+1}\,\mathbf{1}_{x_{k+1} \in \Gamma_\delta^M} + \mu\,\mathbf{1}_{x_{k+1} \in \left(\Gamma_\delta^M\right)^c}, \tag{9.23}$$

   where $M > 0$, $\delta \in (0, M)$ are given,

$$\Gamma_\delta^M := \{x \in \mathbb{R} \colon \delta < x < M\}\,, \tag{9.24}$$

$$\mathbf{1}_A := \begin{cases} 1, & \text{if } A \text{ is true,} \\ 0, & \text{otherwise,} \end{cases} \tag{9.25}$$

$$x_{k+1} := \gamma_k + \gamma_k\, a_k \left(-1 + \gamma_0 \hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma_k}, \hat{\theta}_T, \theta_0)\right)\right)\,, \tag{9.26}$$

   and where $\mu \in \Gamma_\delta^M$ is a constant to be defined in the next subsection.

   Set $k = k + 1$ and return to Step 1.

In Step 1, the constraint in the optimization can be difficult to implement as the set $\Theta_{\text{app}}(\gamma_k)$ can be uncountable. To overcome this issue, we employ the scenario approach [116]. By taking N_{s} samples from $\Theta_{\text{app}}(\gamma_k)$ based on a user defined probability measure $\mathbf{P}_{\text{s}}$, we replace the constraint in (9.21) by

$$\mathcal{R}_{\Theta_0}\left(J\left(u_{1:T}, \hat{\theta}_T, \theta_0\right)\right) \leq \frac{1}{\gamma_k}, \text{ for all } \hat{\theta}_T \in \{\theta_i\}_{i=1}^{N_{\text{s}}}, \tag{9.27}$$

where $\{\theta_i\}_{i=1}^{N_{\text{s}}} \in \Theta_{\text{app}}(\gamma_k)$ are the samples drawn according to $\mathbf{P}_{\text{s}}$. We refer to Appendix C for more details on the scenario approach.

Finally, the expression (9.27) is usually approximated by Monte Carlo methods. We denote by $\hat{\mathcal{R}}_{\Theta_0}(J(u_{1:T}, \hat{\theta}_T, \theta_0))$ the Monte Carlo approximation of (9.27) based on the samples $\{\theta_{0,i}\}_{i=1}^{N_{\text{sim}_1}}$ drawn from $\mathbf{P}_{\Theta_0}$.

As for (9.27), the expression in Step 2 usually relies on Monte Carlo methods, so we will denote by $\hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}(J(u_{1:T}, \hat{\theta}_T, \theta_0))$ the Monte Carlo approximation of the expression in Step 2 based on the samples $\{\theta_{0,i}\}_{i=1}^{N_{\text{sim}_1}}$ and $\{\hat{\theta}_{N,ji}\}_{j=1}^{N_{\text{sim}_2}}$ for $i \in \{1, \ldots, N_{\text{sim}_1}\}$, taken from $\mathbf{P}_{\Theta_0}$ and $\mathbf{P}_{\hat{\Theta}|\Theta_0}(\cdot|\theta_{0,i})$, respectively.

Finally, the properties of the sequence $\{a_k\}_{k\geq 1}$ in Step 3 follows the conditions for convergence of stochastic approximation methods, which are discussed in the next section.

Convergence of the algorithm

A natural question arising at this point is whether the proposed algorithm converges to the desired value. In other words, we want to see if $\gamma_k \to \gamma^\star$ as $k \to \infty$, where $\gamma^\star$ satisfies

$$\mathcal{R}_{\hat{\theta}_N,\theta_0}\left(J\left(u_{1:T}^{\gamma^\star}, \hat{\theta}_T, \theta_0\right)\right) = \frac{1}{\gamma_0}. \tag{9.28}$$

As we can see from (9.28), the value $\gamma^\star$ is such that $u_{1:T}^{\gamma^\star}$ is a feasible solution to the optimization problem (9.20), and it can then be used to guarantee a prescribed accuracy on $\hat{\theta}_T$ independent of the value of $\theta_0 \in \Theta$.

To continue, we require the following:

Condition 9.1 *The parameters $\gamma^\star, \gamma_0$ satisfy $\gamma^\star, \gamma_0 \in \Gamma_\delta^M$.*

Condition 9.2 *$\mathcal{R}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma}, \hat{\theta}_T, \theta_0)\right)$ is a bounded and monotonically decreasing function of $\gamma \in \Gamma_\delta^M$.*

Condition 9.3 *$J(u_{1:T}^{\gamma}, \cdot, \cdot) \in \mathcal{L}^2$ for every $\gamma \in \Gamma_\delta^M$.*

Condition 9.1 is required to guarantee the boundedness of γ_0^{-1} and $(\gamma^\star)^{-1}$. Furthermore, Condition 9.2 does not introduce severe conservatism as increasing γ usually implies a more accurate estimate $\hat{\theta}_T$, and thus a lower value for $\mathcal{R}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma}, \hat{\theta}_T, \theta_0)\right)$.

Finally, Condition 9.3 guarantees that the variance of the Monte Carlo estimate

$$\hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma_k},\hat{\theta}_T,\theta_0)\right) \tag{9.29}$$

is finite.

Remark 9.2 *Condition 9.1 requires prior knowledge about the parameter $\gamma^\star$ yielding* (9.28)*. This is not a very strict assumption, as the constant δ can be made arbitrarily close to zero and M sufficiently large to guarantee $\gamma^\star \in \Gamma_\delta^M$.*

To continue, we define

$$f_{\gamma_0}\left(\gamma_k, a_k, \hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(J\left(u_{1:T}^{\gamma_k},\hat{\theta}_T,\theta_0\right)\right)\right) := x_{k+1}\,\mathbf{1}_{x_{k+1}\in\Gamma_\delta^M} + \mu\,\mathbf{1}_{x_{k+1}\in\left(\Gamma_\delta^M\right)^c}, \tag{9.30}$$

The objective of the mapping (9.30) is to drive the exploration of the set Γ_δ^M to the desired direction. Indeed, if $\hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma_k},\hat{\theta}_T,\theta_0)\right) > 1/\gamma_0$, then $\gamma_{k+1} > \gamma_k$, implying a more conservative design for $u_{1:T}$ at iteration $k+1$. The opposite holds when $\hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma_k},\hat{\theta}_T,\theta_0)\right) < 1/\gamma_0$.

We note that the recursive equation (9.30) satisfies

$$f_{\gamma_0}\left(\gamma^\star, a_k, \mathcal{R}_{\hat{\Theta},\Theta_0}\left(J\left(u_{1:T}^{\gamma^\star},\hat{\theta}_T,\theta_0\right)\right)\right) = \gamma^\star\,, \tag{9.31}$$

with $\gamma^\star$ satisfying (9.28). Hence, $\gamma^\star$ is a fixed point of the recursive equation (cf. Equation (9.23))

$$\gamma_{k+1} = f_{\gamma_0}\left(\gamma_k, a_k, \mathcal{R}_{\hat{\Theta},\Theta_0}\left(J\left(u_{1:T}^{\gamma_k},\hat{\theta}_T,\theta_0\right)\right)\right)\,. \tag{9.32}$$

To analyze the recursive equation (9.32), we require

Condition 9.4 *The sequence $\{a_k\}_{k\geq 1}$ satisfies $a_k > 0$ for all $k \geq 1$ and*

$$\sum_{k=1}^{\infty} a_k = \infty\,, \tag{9.33}$$

$$\sum_{k=1}^{\infty} a_k^2 < \infty\,. \tag{9.34}$$

The requirements in Condition 9.4 are standard in stochastic approximation methods [39], and will be useful to prove the convergence of the proposed algorithm.

One of the difficulties in establishing the convergence of γ_k to $\gamma^\star$ is that the recursive equation (9.32) in practice is based on the Monte Carlo estimate $\hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma_k},\hat{\theta}_T,\theta_0)\right)$, which introduces a stochastic behavior in the sequence

$\{\gamma_k\}_{k\geq 1}$. In other words, the recursive equation in Step 3 of the proposed algorithm is given for all $k \geq 1$ by

$$\gamma_{k+1} = f_{\gamma_0}\left(\gamma_k, a_k, \hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(J\left(u_{1:T}^{\gamma_k}, \hat{\theta}_T, \theta_0\right)\right)\right), \tag{9.35}$$

with f_{γ_0} defined as in (9.30).

Thus, to proceed we write

$$\hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma_k}, \hat{\theta}_T, \theta_0)\right) = \mathcal{R}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma_k}, \hat{\theta}_T, \theta_0)\right) + \boldsymbol{\nu}_{k+1|\gamma_k}, \tag{9.36}$$

where $(\boldsymbol{\nu}_{k+1|\gamma_k}, \mathcal{F}_{k+1})$ is such that for all $k \geq 1$

$$\mathbf{E}\{\boldsymbol{\nu}_{k+1|\gamma_k}|\mathcal{F}_k\} = 0, \tag{9.37}$$

$$\mathbf{E}\{\boldsymbol{\nu}^2_{k+1|\gamma_k}|\mathcal{F}_k\} =: K_{\gamma_k} < \infty, \tag{9.38}$$

with $\{\mathcal{F}_k\}_{k\geq 1}$ a sequence of nondecreasing σ-algebras. Expressions (9.36)-(9.38) follow from the fact that

$$\hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma}, \hat{\theta}_T, \theta_0)\right) \tag{9.39}$$

is an unbiased estimate of $\mathcal{R}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma}, \hat{\theta}_T, \theta_0)\right)$ with finite variance for all $\gamma \in \Gamma_\delta^M$.

To prove the convergence of the proposed algorithm, we rely on a slightly modified version of [39, Theorem 1.4.1], which presents a convergence result for the truncated Robbins-Monro algorithm. To this end, we require the following conditions to hold[1]:

Condition 9.5 *There exist a continuously differentiable convex Lyapunov function v such that $v(\gamma^\star) = 0$, $v(\gamma) \neq 0$ for all $\gamma \neq \gamma^\star$, $\nabla_\gamma v(\gamma) < 0$ for all $\gamma < \gamma^\star$, $\nabla_\gamma v(\gamma) > 0$ for all $\gamma > \gamma^\star$, and constants $0 < \xi < \Delta$ such that*

$$\sup_{\xi \leq |\gamma - \gamma^\star| \leq \Delta} \left(-1 + \gamma_0 \mathcal{R}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma}, \hat{\theta}_T, \theta_0)\right)\right) \nabla_\gamma v(\gamma) < 0, \tag{9.40}$$

and for $\mu \in \Gamma_\delta^M$ in (9.30), we have $v(\mu) < \min(v(\delta), v(M))$.

Condition 9.6 *For any convergent subsequence $\{\gamma_{n_k}\}$ of $\{\gamma_k\}$*

$$\lim_{T_a \to 0} \limsup_{k\to\infty} \frac{1}{T_a} \left| \sum_{i=n_k}^{m(n_k,t)} a_i \boldsymbol{\nu}_{i+1|\gamma_i} \right| = 0, \text{ for all } t \in [0, T_a], \tag{9.41}$$

where

$$m(k, T_a) := \max\left\{ m \colon \sum_{i=k}^{m} a_i \leq T_a \right\}. \tag{9.42}$$

[1]Even though Condition 9.6 is immediately fulfilled in our context, it is still presented for completeness of the proof of Theorem 9.1.

Condition 9.7 $\mathcal{R}_{\hat{\Theta},\Theta_0}\left(J(u^{\gamma}_{1:T},\hat{\theta}_T,\theta_0)\right)$ *is measurable and locally bounded with respect to* γ.

The requirement on μ in Condition 9.5 seems restrictive, as it needs prior knowledge about the value of $\gamma^\star$. This issue can be circumvented by choosing M sufficiently large, and μ sufficiently close to δ to guarantee $v(\mu) < v(\delta) < v(M)$. Note that v satisfying Condition 9.5 exists: take, for example, $v(\gamma) = (\gamma - \gamma^\star)^2$. Indeed, under this choice, the sign of the factor multiplying $\nabla_\gamma v(\gamma)$ is the opposite sign of $\nabla_\gamma v(\gamma)$ for every $\gamma \neq \gamma^\star$. In addition, Condition 9.6 is satisfied in the current framework, as $(\boldsymbol{\nu}_{k+1|\gamma_k}, \mathcal{F}_{k+1})$ is a martingale difference sequence, and expression (9.41) has been shown to hold for this class of sequences for almost all sample paths $\boldsymbol{\omega}$ under Condition 9.4 [39, Example 1.3.2].

Given the elements presented in this section, the next result follows:

Theorem 9.1 *Assume Conditions 9.1-9.5 and 9.7 to hold. If Condition 9.6 holds, then* γ_k *given by* (9.35) *converges to* $\gamma^\star$ *for almost all sample paths* $\boldsymbol{\xi}$.

We refer to Appendix F for the proof of Theorem 9.1.

9.5 Numerical example

Consider the system

$$y_t = K\,\frac{(1 + 2r\,\cos(\theta_0) + r^2)q^{-2}}{1 + 2r\,\cos(\theta_0)q^{-1} + r^2q^{-2}}\,u_t + e_t\,, \tag{9.43}$$

where y_t is the measured output, u_t is the input signal, and $\{e_t\}$ is a Gaussian white noise of zero mean and unit variance. We assume that $K = 2$ and $r = 0.85$ are known, but that the value of θ_0 is uncertain, i.e., the location of the resonance frequency (determined by θ_0) is uncertain. However, it is known that $\theta_0 \in [0.5, 1.5]$. Since any value on $[0.5, 1.5]$ is equally likely for θ_0, we consider $\theta_0 \sim \mathcal{U}[0.5, 1.5]$.

The objective in this example is to design $u_{1:T}$ ($T = 10^2$) as a realization of a zero mean wide sense stationary process, such that the estimate $\hat{\theta}_T$ obtained from $(y_{1:T}, u_{1:T})$ satisfies a performance constraint defined by the model application. In this case, the model application is controller design, where the control law is given by

$$u_t = C_{\hat{\theta}_T}(q)\,(r_t - y_t)\,, \tag{9.44}$$

with r_t the reference signal and $C_{\hat{\theta}_T}(q)$ the controller, which is a real rational function in q. Note that the controller depends on $\hat{\theta}_T$ as it is designed based on the model $G_{\hat{\theta}_T}$.

The controller $C_{\hat{\theta}_T}$ is designed such that

$$\lim_{t\to\infty} \mathbf{E}\{y_t\} = r_t\,, \tag{9.45}$$

when $\{r_t\}$ is a constant signal. Moreover, $C_{\hat{\theta}_T}$ is such that the closed loop poles are $\{0.3, 0.5, 0.6, 0.7\}$. To measure the performance degradation, we consider the index

$$V_{\text{app}}(\theta, \theta_0) = \left(\text{pm}\{G_{\theta_0} C_{\theta_0}\} - \text{pm}\{G_{\theta_0} C_{\theta}\}\right)^2 , \tag{9.46}$$

with $\text{pm}\{F\}$ the phase margin associated with the function $F(q)$. Hence, the application set $\Theta_{\text{app}}(\gamma_0)$ in (9.12) is defined in terms of (9.46) with $\gamma_0 = 10^2$.

As the controller is designed in terms of $G_{\hat{\theta}_T}$, it is important to guarantee that the closed-loop resulting of using $C_{\hat{\theta}_T}$ is stable. Hence, it is important to avoid parameter estimates $\hat{\theta}_T$ causing closed-loop instability.

To obtain a tractable problem, we parameterize $\{u_t\}$ in terms of its power spectrum Φ_u. In addition, we restrict Φ_u to the FIR structure

$$\Phi_u(\omega) = \sum_{\tau=-30}^{30} c_{|\tau|} e^{j\omega\tau} , \tag{9.47}$$

where $c_{0:30} \in \mathbb{R}^{31}$ is the decision vector, satisfying $\Phi(\omega) \geq 0$ for all $\omega \in [-\pi, \pi]$.

Finally, for the application cost, we consider $H(u_{1:T}) = \mathbf{E}\{u_t^2\}$.

The original application oriented input design problem is then written as[2]

$$\begin{aligned} \min_{\Phi_u} \quad & \mathbf{E}\{u_t^2\} \\ \text{subject to} \quad & \Phi_u(\omega) = \sum_{\tau=-30}^{30} c_{|\tau|} e^{j\omega\tau} , \\ & \Phi_u(\omega) \geq 0, \ \text{ for all } \omega \in [-\pi, \pi] , \\ & V_{\text{app}}(\hat{\theta}_T, \theta_0) \leq \frac{1}{\gamma_0} . \end{aligned} \tag{9.48}$$

The difficulty with the formulation (9.48) is that both $\hat{\theta}_T$ and θ_0 are uncertain, and their uncertainties are modeled by probability distributions. One possibility to deal with these uncertainties is to consider the nominal parameter $\theta^{\mathbf{E}} := \mathbf{E}\{\theta_0\} = 1$, and solve the AOID (9.48) for the ellipsoidal approximation of the application set. This formulation can be written as

$$\begin{aligned} \min_{\Phi_u} \quad & \mathbf{E}\{u_t^2\} \\ \text{subject to} \quad & \Phi_u(\omega) = \sum_{\tau=-30}^{30} c_{|\tau|} e^{j\omega\tau} , \\ & \Phi_u(\omega) \geq 0, \ \text{ for all } \omega \in [-\pi, \pi] , \\ & \frac{T}{\chi^2_\alpha(n_\theta)} \overline{\mathcal{I}}_F(\theta^{\mathbf{E}}) \succeq \frac{\gamma_0}{2} \nabla^2_\theta V_{\text{app}}(\theta, \theta^{\mathbf{E}})\big|_{\theta=\theta^{\mathbf{E}}} , \end{aligned} \tag{9.49}$$

[2]We refer to Appendix E for more details on the implementation of the constraint on Φ_u and the computation of $\overline{\mathcal{I}}_F(\theta_0)$ for wide sense stationary processes.

where $\alpha = 0.95$. We refer to Appendix G, where the derivation of the linear matrix inequality (LMI) constraint in (9.49) is presented. The power spectrum solving (9.49) will be referred to as the nominal input spectrum.

The robust alternative to the problem (9.48) is given by

$$\begin{aligned} \min_{\Phi_u} \quad & \mathbf{E}\{u_t^2\} \\ \text{subject to} \quad & \Phi_u(\omega) = \sum_{\tau=-30}^{30} c_{|\tau|} e^{j\omega\tau}\,, \\ & \Phi_u(\omega) \geq 0, \ \text{ for all } \omega \in [-\pi,\, \pi]\,, \\ & \mathcal{R}_{\hat{\Theta},\Theta_0}\left(V_{\text{app}}(\hat{\theta}_T, \theta_0)\right) \leq \frac{1}{\gamma_0}\,, \\ & \theta_0 \sim \mathcal{U}[0.5, 1.5]\,,\ \hat{\theta}_T|\theta_0 \sim \mathcal{N}\left(\theta_0, T^{-1}\,\overline{\mathcal{I}}_F^{-1}(\theta_0)\right)\,. \end{aligned} \tag{9.50}$$

Since the distribution of $\hat{\theta}_T$ depends on Φ_u (the decision variable), the problem (9.50) becomes difficult to solve as $\mathcal{R}_{\hat{\Theta},\Theta_0}$ requires the distribution of $\hat{\theta}_T$ given θ_0, which depends on Φ_u. To provide a suboptimal solution to problem (9.50), we employ the stochastic approximation algorithm presented in Section 9.4 to obtain a power spectrum Φ_u such that the risk coherent inequality in (9.50) is fulfilled. For the purpose of this example, we consider $\mathcal{R}_{\hat{\Theta},\Theta_0} = \text{CVaR}_\alpha$, with $\alpha = 0.95$ (Defined in Chapter 8, Section 8.3).

For this example, the stochastic approximation algorithm in Section 9.4 is formulated as follows. At iteration k, the first step of the algorithm computes the input power spectrum by solving

$$\begin{aligned} \min_{\Phi_u} \quad & \mathbf{E}\{u_t^2\} \\ \text{subject to} \quad & \Phi_u(\omega) = \sum_{\tau=-30}^{30} c_{|\tau|} e^{j\omega\tau}\,, \\ & \Phi_u(\omega) \geq 0, \ \text{ for all } \omega \in [-\pi,\, \pi]\,, \\ & \mathcal{R}_{\Theta_0}\left(\max_{\theta \in \Theta_{\text{app}}(\gamma_k)} \gamma_k \chi_\alpha^2(n_\theta) V_{\text{app}}(\theta, \theta_0) - T\,\overline{\mathcal{I}}_F(\theta_0)(\theta - \theta_0)^2\right) \leq 0\,, \\ & \theta_0 \sim \mathcal{U}[0.5, 1.5]\,, \end{aligned} \tag{9.51}$$

with $\alpha = 0.95$ and $\mathcal{R}_{\Theta_0} = \text{CVaR}_\alpha$. We refer to Appendix G for the derivation of the robust constraint in (9.51). The robust constraint in (9.51) is implemented by using the scenario approach to approximate the maximum operator (cf. Appendix C) based on $N_s = 80$ samples for θ in $\Theta_{\text{app}}(\gamma_k)$ (given θ_0), and a Monte Carlo approximation for CVaR_α based on $N_{\text{sim}_1} = 80$ samples for θ_0.

In the second step, we compute $\mathcal{R}_{\hat{\Theta},\Theta_0}\left(V_{\text{app}}(\hat{\theta}_T, \theta_0)\right)$, with $\mathcal{R}_{\hat{\Theta},\Theta_0} = \text{CVaR}_\alpha$. This computation is implemented based on the optimal input spectrum for the

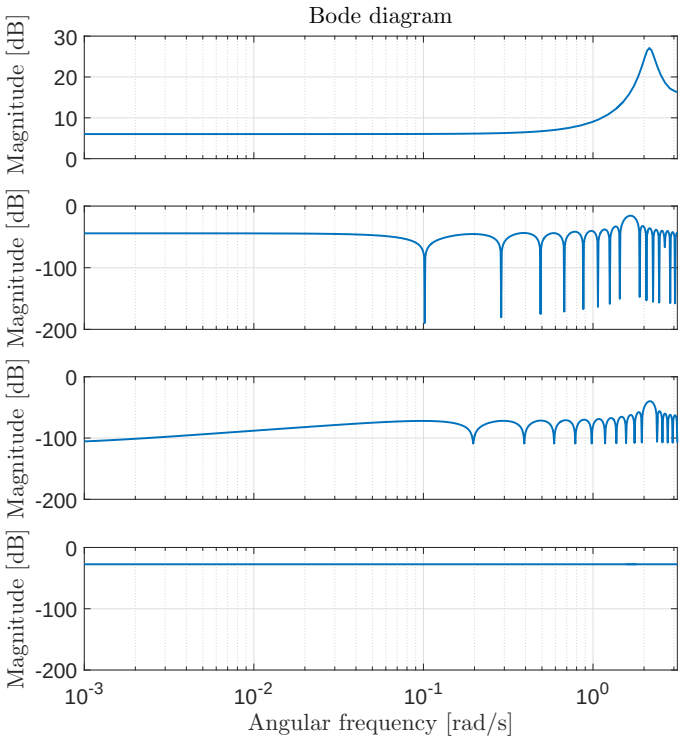

Figure 9.2: Bode diagram for the power input spectra and $G_{\theta \mathbf{E}}$. Top: magnitude of the Bode diagram of $G_{\theta \mathbf{E}}$. Second from top: robust input spectrum. Third from top: nominal input spectrum. Bottom: spectrum of white noise with variance equal to the one from the robust input spectrum.

problem (9.51), with $N_{\text{sim}_1} = 80$ samples for θ_0 and $N_{\text{sim}_2} = 80$ samples for $\hat{\theta}_T$ given θ_0, which is distributed according to the last expression in problem (9.50).

In the last step, we compute the value of γ_{k+1} using (9.30)-(9.26), where $\delta = 10^{-9}$, $M = 10^9$, $\mu = 5 \cdot 10^{-3}$ and $a_k = 1/k$ for all $k \geq 1$. In Equation (9.30) we have that

$$\hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma_k}, \hat{\theta}_T, \theta_0)\right) = \hat{\mathcal{R}}_{\hat{\Theta},\Theta_0}\left(V_{\text{app}}(\hat{\theta}_T, \theta_0)\right) . \tag{9.52}$$

Finally, the termination criterion β is chosen as $\beta = 10^{-4}$. In the following, we refer to the solution obtained for the power spectrum by the stochastic approximation algorithm as the robust input spectrum.

The nominal problem (9.49) and the stochastic approximation method in Section 9.4 are implemented in Matlab and solved using the `cvx` toolbox. The resulting

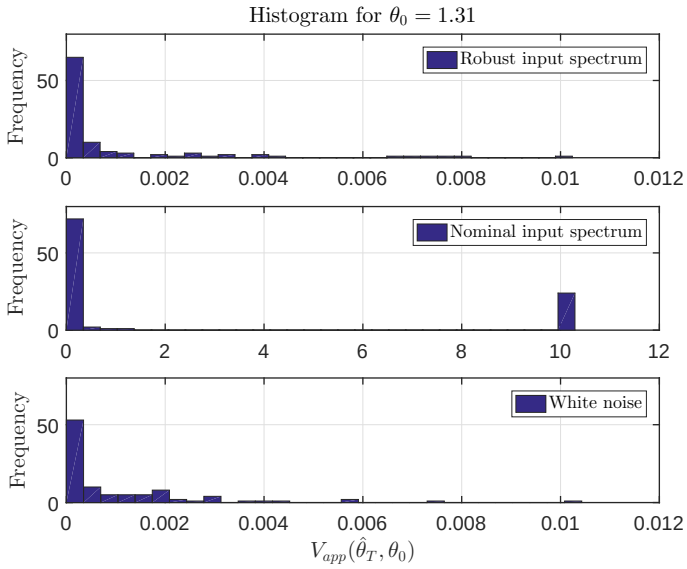

Figure 9.3: Histogram of V_{app} for $\theta_0 = 1.31$. Top: robust input spectrum. Middle: nominal input spectrum. Bottom: white noise process with variance equal to the one of the robust input spectrum.

input power spectra, together with the magnitude of the Bode diagram of $G_{\theta^{\mathbf{E}}}$ are presented in Figure 9.2.

We note in Figure 9.2 that both the nominal and robust input spectra have a peak in their frequency response. It is interesting to note that the peak of the frequency response of the nominal input spectrum coincides with the resonance peak of $G_{\theta^{\mathbf{E}}}(e^{j\omega})$. This behavior is expected, as the problem (9.49) only employs the mean value of θ_0, and assumes that $\theta^{\mathbf{E}}$ describes exactly the underlying system. Therefore, most of the power in the spectrum Φ_u can be allocated at the resonance peak of $|G_{\theta^{\mathbf{E}}}(e^{j\omega})|$.

When analyzing the robust input spectrum in Figure 9.2, we see that the uncertainty on θ_0 implies a more conservative design than the one attained with the nominal input spectrum. By conservative we mean that the power content per frequency for the robust input spectrum is greater than the one for the nominal input spectrum. Moreover, the peak of the robust input spectrum does not occur at the resonance peak of $|G_{\theta^{\mathbf{E}}}(e^{j\omega})|$. This can be interpreted as diversifying the power content on the frequency range where the resonance peak of $|G_{\theta_0}(e^{j\omega})|$ can be attained. Hence, the robust power spectrum tries to ensure an experiment attaining an application performance that is independent of the location of the resonance frequency.

As a reference, we also present in Figure 9.2 the power spectrum of a zero mean

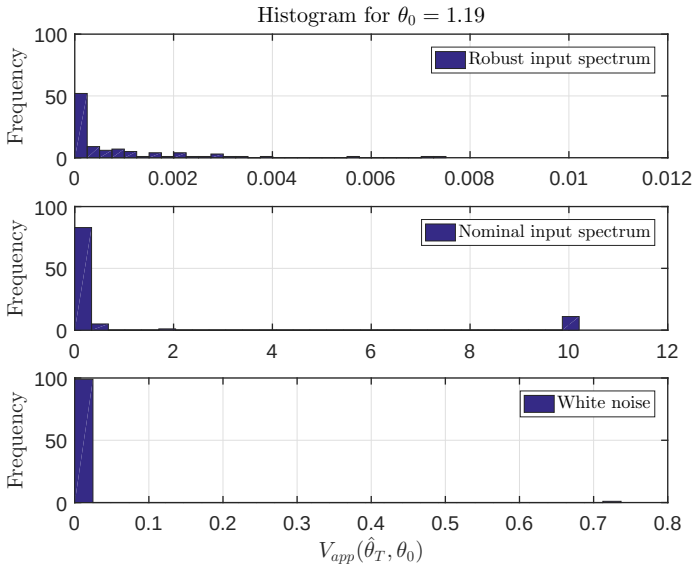

Figure 9.4: Histogram of V_{app} for $\theta_0 = 1.19$. Top: robust input spectrum. Middle: nominal input spectrum. Bottom: white noise process with variance equal to the one of the robust input spectrum.

white noise process with the variance equal to the one associated with the robust input spectrum. We note that the power spectrum of the white noise process is greater than the nominal input spectrum at every frequency, and it is smaller than the peak of the frequency response of the robust input spectrum. Hence, we expect to attain better results in terms of application performance for the white noise process than for the nominal input spectrum.

We can also test the different designs to analyze the application performance. To this end, we generate $u_{1:T}$ by filtering a zero mean Gaussian white noise process with unit variance with the spectral factor for every power spectra. These realizations are kept fixed and then the system (9.43) is simulated for $N_{MC} = 100$ different realizations of $e_{1:T}$ for $\theta_0 \in \{1.19, 1.31\}$. For every realization of $e_{1:T}$, the data set $\mathbf{Z}_T$ is used for estimating θ_0 with PEM. Finally, for every input and every $\theta_0 \in \{1.19, 1.31\}$, the estimated parameters $\{\hat{\theta}_T^{(i)}\}_i^{N_{MC}}$ are employed to evaluate $V_{\text{app}}(\hat{\theta}_T, \theta_0)$.

Figures 9.3-9.4 present the histograms of V_{app} for the different inputs, when $\theta_0 = 1.31$ and $\theta_0 = 1.19$, respectively. In both figures, we see that the performance degradation using the nominal input spectrum is significantly greater than the desired value $1/\gamma_0 = 0.01$ for a considerable number of samples. Moreover, for $\theta_0 = 1.31$, two samples of $\{\hat{\theta}_T\}_{i=1}^{N_{MC}}$ deliver an unstable closed loop system. The

same occurs to four samples of $\{\hat{\theta}_T\}_{i=1}^{N_{MC}}$ when $\theta_0 = 1.19$. Hence, the nominal input spectrum does not guarantee the desired application performance, causing even instability in closed loop.

For the white noise process with the variance of the robust input spectrum, we see in Figures 9.3-9.4 that the application performance significantly improves when compared to the one achieved with the nominal input spectrum. Moreover, for $\theta_0 = 1.31$, the histogram of the application performance is very close to the one attained with robust input spectrum. However, when $\theta_0 = 1.19$, there is one sample that violates significantly the application performance requirement $V_{\text{app}}(\hat{\theta}_T, \theta_0) < 0.01$, with a value greater than 0.7. Even though the application performance is fulfilled for the remaining 99 samples, when it fails to fulfill the inequality the estimated parameter generates very poor results in terms of application performance. This is in line with Example 9.4 in the sense of that the use of a wrong approach to account for the uncertainty can lead to undesired results.

When inspecting the histograms obtained with the robust input spectrum in Figures 9.3-9.4, we see that the distribution of the application cost is almost independent of the value of θ_0, which is a desirable behavior when considering a robust framework. We emphasize that this behavior is achieved at the expense of increasing the input variance $\mathbf{E}\{u_t^2\}$, as we have $\mathbf{E}\{u_t^2\} = 1.8 \cdot 10^{-3}$ for the robust input design and $\mathbf{E}\{u_t^2\} = 6.8 \cdot 10^{-6}$ for the nominal input design. In conclusion, the introduction of a robust framework in AOID significantly improves the behavior of the application cost under the presence of model uncertainties.

9.6 Conclusions

This chapter considered the problem of uncertainty in application oriented input design. Due to the uncertainty in the location of both the estimated parameter and the parameter describing the true system, a risk theoretical approach to account for the uncertainty in the optimization has been presented. The approach employs the notion of coherent measure of risk, which has been proven to be a useful concept in the literature of theory of risk. In particular, the use of the conditional value at risk in application oriented input design has been explored, and a risk coherent method to compute a sub-optimal robust solution has been introduced. A numerical example shows that a robust approach in application oriented input design helps to obtain the desired application performance in the presence of model uncertainty.

Chapter 10

Conclusions and future work

This thesis has presented a compendia of contributions in different areas of system identification. Specifically, the contribution of this thesis is on three topics:

(i) input design methods for the identification of nonlinear dynamical systems,

(ii) Bayesian methods for system identification, and

(iii) a risk theoretical framework for robustness in system identification.

In the following, we provide concluding remarks for each topic.

10.1 Input design for nonlinear dynamical models

A robust input design procedure for the identification of nonlinear SSM has been presented. The method has considered the minimization of a scalar cost function of the Fisher information matrix over the set of stationary Markov processes of a given order. As the model parameters are unknown, the method optimized a cost function taking the uncertainty on the model parameters into account.

To obtain a tractable problem, the Markov process is parameterized in terms of its stationary distribution, which is defined over a vector of consecutive input realizations and where the input has finite alphabet. Moreover, the stationary distribution of the Markov process is restricted to the set of marginal distributions of stationary processes. Based on this restriction, the feasible set is characterized by the use of graph theoretical methods, where the vertices of such set are associated with a particular class of elementary cycles in the equivalent de Bruijn graph.

As the vertices of the feasible set are available (which are probability mass functions), then the input design problem is solved by computing the optimal convex combination of the quantities associated with the pmfs in the vertices of the set. This equivalence made the problem convex in the decision variables. Once the optimization is solved, an input realization is obtained by running a Markov chain with the optimal pmf as stationary distribution.

As the problem is solved by optimizing a convex formulation approximating the cost in the original problem, the proposed method introduces an error between the optimized cost and the actual cost obtained by applying the designed input to the system. However, in this thesis is shown that it is possible to bound such difference under some assumptions on the model structure.

As an application of the proposed method, this thesis presented the use of the robust input design technique for the identification of FIR systems with quantized measurements. Numerical examples showed that the method improves the accuracy of the parameter estimates in the presence of model uncertainty.

10.2 Bayesian methods for system identification

This part of the thesis presented the use of Bayesian techniques for input design and model estimation for dynamical systems. In the case of input design, a Gaussian process optimization method has been developed. The aim with this method is to provide an alternative to the optimization of the scalar cost function of the Fisher information matrix, which is unavailable in closed form for general NSSMs. To this end, the optimization of a surrogate model for the objective function has been proposed. The surrogate model is obtained in a Bayesian framework, where a predictive posterior is built based on a Gaussian process prior on the cost function, and a set of estimates of the cost function over the feasible set. The points of the feasible set where the cost function is estimated are computed using a Bayesian optimization framework, where we recursively explored the feasible set based on the solution of an associated optimization. The associated optimization established a trade-off between exploration and exploitation of the current estimate of the cost function, which is reflected in the points employed for updating the surrogate cost function.

Motivated by the Bayesian framework in input design, this thesis also explored the use of Bayesian methods for the identification of SISO LTI models. Based on a prior pdf over the parameter set and the likelihood function, the posterior distribution of the model parameters given the observed data is computed. In particular, this thesis proposed the use of prior knowledge on the model complexity by specifying a prior pdf over the Hankel singular values of the system. As the posterior distribution of the parameters is often unavailable in closed form, a Metropolis-Hastings sampler is implemented to obtain samples distributed according to the posterior pdf. The correctness and convergence of the proposed algorithm is established, and the numerical example showed that the method is an attractive technique to encode information about the model complexity in a Bayesian setting.

10.3 Risk theoretical framework in system identification

The last part of this thesis concerned the issue of uncertainty in system identification. In particular, this thesis focused on how to properly account for the model

uncertainty in input design. To this end, we adopted a risk theoretical perspective. By reinterpreting the cost function in input design, we used the notion of coherent measures of risk to provide an attractive manner to properly account for the model uncertainty. The standard robust measures employed in input design are analyzed from a risk coherent perspective, which can be useful for establishing the disadvantages of using robust measures without fully understanding its consequences in the resulting optimization.

The same risk theoretical approach is employed for application oriented input design, where the focus is the minimization of an experimental cost, subject to quality constraints on the estimated parameters. In this framework, the quality constraint on the estimated parameters is established by restricting the performance degradation obtained in the intended model application. The main difficulty here is that the formulation suffered from uncertainty on both the true model parameter and the estimated model parameter. In this case, the risk coherent formulation is not immediate, as the asymptotic distribution of the estimated parameter depended on the decision variable. To address this difficulty, a stochastic approximation method has been proposed, where a suboptimal input is computed. Nevertheless, the suboptimal input ensures that a risk coherent measure of the model quality satisfies the desired performance. The designed algorithm is analyzed and its convergence is established.

10.4 Future work

The ideas discussed in this thesis leave several research directions that can be explored in future work. In the next subsections, we describe some of the ideas that can be subject of further study.

Input design

The graph theoretical approach to input design provides a methodology relying on convex optimization tools. Even though the proposed formulation guarantees the convergence of the numerical solvers to the global optimum, the technique still needs to deal with the problem of computational complexity, as the number of vertices of the feasible set grows exponentially with the length of the input vector over which the pmfs are defined. A solution to this point is to explore the reduction in computational complexity by relating the optimal pmf over input vectors of length n_m to the optimal pmf over input vectors of length $n_m + 1$. This can be made by noticing that the prime cycles describing the vertices of the set of marginal pmfs of stationary processes of dimension n_m determine a subset of the prime cycles describing the marginal pmfs of stationary processes of dimension $n_m + 1$.

Another aspect to be considered as a future work is the design of the alphabet for the input. So far, the alphabet has been arbitrarily chosen to illustrate the method, but it is also possible to think in a simultaneous design of the input alphabet and optimization of the scalar cost of the Fisher information matrix.

Bayesian methods for system identification

The GPO framework provides an alternative solution to the problem of input design for the identification of nonlinear state space models. Currently, the method only considers nominal experiment design, i.e., the input is designed based on prior knowledge of the model parameters. A natural step in this area is to formulate its robust equivalent, where the uncertainty on the model parameters is taken into account when designing the input.

Even though the numerical illustration shows that the GPO technique can be a useful method for input design, the method does not provide any guarantees of convergence or optimality of the designed input in terms of the cost function being optimized in a Bayesian setting. In this line, future work in this direction should focus on establishing convergence results and bounds on the optimality of the designed input.

Another research path in the GPO framework for input design is the implementation of a recursive algorithm, which provides an online procedure to design the sample u_t given the information up to time $t-1$. The current formulation designs the full input realization $u_{1:T}$, and hence the optimization needs to be solved offline.

For the Bayesian identification method discussed in this part of the thesis, we remark that the current implementation of the Metropolis-Hastings sampler can only be applied to SISO LTI models. Thus, future work in this framework should focus on extending the methodology to MIMO LTI models. This formulation is not immediate, since the assumptions on the inverse stability of the noise model imply that the random walk must enforce, at every iteration, that the multivariable zeros of the noise model lie strictly inside the complex unit disc.

The Bayesian identification method requires the specification of a prior pdf over the model parameters. It is often the case that the prior pdf requires the value of hyperparameters characterizing the distribution. This issue has been circumvented in this thesis by fully specifying the prior pdf. How to optimally tune the hyperparameters defining the prior pdf is subject of future work.

Risk theoretical framework in system identification

The final part of this thesis discusses how the risk theoretical framework can provide a framework to properly account for the uncertainty in system identification. In particular, this thesis explores the use of risk coherent measures to account for the model uncertainty in input design and application oriented input design. However, the same approach can also be employed in other areas of system identification and automatic control, where the uncertainty plays a fundamental role. To this end, we require to reinterpret the concepts in system identification and automatic control from a risk theoretical perspective, where the notion of risk coherent measures can help to solve the problem of uncertainty. This point may be part of future work on the subject.

Appendices

Appendix A

Algorithms for the computation of elementary cycles

In this appendix we present the algorithms implemented for the input design methods of Chapters 4 and 5.

A.1 Preliminaries

Here we extend the definitions of directed graph and elementary cycles given in Chapter 2. Before introducing the algorithms required in Chapters 4 and 5, we need the following definitions [184]:

Definition A.1 *(Undirected graph) An* undirected graph $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$ *is a pair consisting of a finite nonempty set of points (called* nodes *or* vertices*)* $\mathcal{V}$ *and a set* $\mathcal{E}$ *of unordered pairs* $\{z_i, z_j\}$ *of vertices* $z_i, z_j \in \mathcal{V}$ *called* edges*.*

Definition A.2 *(Tail and head of an edge) Let* $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$ *be a directed graph. Given an edge* $(v, w) \in \mathcal{E}$*,* v *is defined as the* tail *of the edge* (v, w)*, and* w *is defined as the* head *of the edge* (v, w)*.*

Definition A.3 *(Undirected version of a directed graph) Let* $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$ *be a directed graph. The* undirected version of a directed graph $\mathcal{G}_\mathcal{V}$ *is defined as the graph formed by converting each edge in* $\mathcal{E}$ *into an undirected edge, and removing duplicate edges.*

Definition A.4 *(Connected undirected graph) Let* $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$ *be a undirected graph. Then* $\mathcal{G}_\mathcal{V}$ *is said to be* connected *if there is a path between every pair of vertices.*

Definition A.5 *(Tree) A* tree $\mathcal{T}$ *is a directed graph whose undirected version is connected, having one vertex which is the head of no edges (called the* root*), and*

such that all vertices except the root are head of exactly one edge. If there is a path in $\mathcal{T}$ such that the starting vertex is v and the ending vertex is w, then v is an ancestor *of w, and w is a* descendant *of v.*

Definition A.6 *(Subtree) Let $\mathcal{T}$ be a tree, and v a node in $\mathcal{T}$. Then $\mathcal{T}_v$ is the* subtree *of $\mathcal{T}$ with respect to v if and only if $\mathcal{T}_v$ has as vertices all the descendants of v in $\mathcal{T}$.*

Definition A.7 *(Spanning tree) Let $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$ be a directed graph. A tree $\mathcal{T}$ is a* spanning tree *of $\mathcal{G}_\mathcal{V}$ if $\mathcal{T}$ is a subgraph of $\mathcal{G}_\mathcal{V}$, and $\mathcal{T}$ contains all the vertices of $\mathcal{G}_\mathcal{V}$.*

Definition A.8 *(Palm tree) Let $\mathcal{G}_\mathcal{V}$ be a directed graph, consisting of two disjoint sets of edges, denoted by $\mathcal{E}_1$ and $\mathcal{E}_2$, respectively. Suppose $\mathcal{G}_\mathcal{V}$ satisfies the following properties:*

(i) The subgraph $\mathcal{T}$ containing the edges $\mathcal{E}_1$ is a spanning tree of $\mathcal{G}_\mathcal{V}$.

(ii) Each edge which is not in the spanning tree $\mathcal{T}$ of $\mathcal{P}$ connects a vertex with one of its ancestors in $\mathcal{T}$.

Then $\mathcal{G}_\mathcal{V}$ is called a palm tree. *The edges $\mathcal{E}_2$ are called the* fronds *of $\mathcal{G}_\mathcal{V}$.*

Definition A.9 *(Cross-link) Let $\mathcal{T}$ be a tree, and assume there are subtrees $\mathcal{T}_v$, $\mathcal{T}_w$ in $\mathcal{T}$. Then, the edge (t_v, t_w) is called a* cross-link *if and only if t_v is in $\mathcal{T}_v$ and t_w is in $\mathcal{T}_w$.*

A.2 Strongly connected components of a graph

A term required in the discussion of the algorithms in this thesis is the concept of strongly connected graphs, whose definition is given below [184, Definition 4]:

Definition A.10 *(Strongly connected graph) Let $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$ be a directed graph. Suppose that for each pair of vertices v, $z \in \mathcal{V}$, there exist paths $\boldsymbol{\omega}_{vz} = (v, \ldots, z)$ and $\boldsymbol{\omega}_{zv} = (z, \ldots, v)$. Then $\mathcal{G}_\mathcal{V}$ is said to be* strongly connected.

Based on Definition A.10, we have the following result [184, Lemma 9]:

Lemma A.1 *(Strongly connected components) Let $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$ be a directed graph. We may define an equivalence relation on the set of vertices as follows: two vertices v and w are equivalent if there is a cycle $\boldsymbol{\omega}_{vv} = (v, \ldots, v)$ which contains w. Let the distinct equivalence classes under this relation be $\mathcal{V}_i$, $i \in \{1, \ldots, n\}$. Let $\mathcal{G}_{\mathcal{V}_i} = (\mathcal{V}_i, \mathcal{E}_i)$, where $\mathcal{E}_i := \{(v, w) \in \mathcal{E} : v, w \in \mathcal{V}_i\}$. Then:*

(i) Each $\mathcal{G}_{\mathcal{V}_i}$ is strongly connected.

(ii) No $\mathcal{G}_{\mathcal{V}_i}$ is a proper subgraph of a strongly connected subgraph of $\mathcal{G}$.

Then the subgraphs $\{\mathcal{G}_{\mathcal{V}_i}\}_{i=1}^{n}$ *are called the* strongly connected components *of* $\mathcal{G}_{\mathcal{V}}$.

It is shown in [184] that the problem of finding the strongly connected components of a graph $\mathcal{G}_{\mathcal{V}}$ can be reduced to the problem of finding the roots of the strongly connected components. Here, we define LOWLINK(v) as the smallest vertex (according to a user defined indexing for the vertices in $\mathcal{G}_{\mathcal{V}}$) which is in the same component as v and is reachable by traversing zero or more tree arcs followed by at most one frond or cross-link. We refer to [184, p. 156] for more details about the definition of this function.

Algorithm A.1 on page 188 presents a pseudo-code for computing the set of strongly connected components in a given graph $\mathcal{G}_{\mathcal{V}}$ [184, p. 157]. It is shown in [184] that Algorithm A.1 has $\mathcal{O}(\#\mathcal{V}) + \mathcal{O}(\#\mathcal{E})$ space and time complexity, where $\mathcal{O}(\cdot)$ is defined as follows:

Definition A.11 *(Order of complexity [109, p. 36]) Let* $T\colon \mathbb{R}_0^+ \to \mathbb{C}$ *and* $f\colon \mathbb{R}_0^+ \to \mathbb{C}$. *We say that* $T(x)$ *is* $\mathcal{O}(f(x))$ *if and only if there exist constants* $0 < c < \infty$ *and* $x_0 \geq 0$ *such that for all* $x \geq x_0$, *it holds that* $|T(x)| \leq c\,|f(x)|$.

The pseudo-code for determining the set of strongly connected components in a graph $\mathcal{G}_{\mathcal{V}}$ will be employed in the computation of elementary cycles in $\mathcal{G}_{\mathcal{V}}$, which is discussed in the next section.

A.3 Elementary cycles of a graph

As mentioned in Chapter 2, the computation of prime cycles in a graph $\mathcal{G}_{\mathcal{C}^{n_m}}$ can be performed by finding all the elementary cycles in $\mathcal{G}_{\mathcal{C}^{n_m-1}}$ (cf. Definition 2.3). In this section we provide an algorithm to find the set of elementary cycles in a directed graph $\mathcal{G}_{\mathcal{V}}$. The algorithm is based on the one introduced in [103, pp. 79–80]. The pseudo-code associated with this algorithm is presented in Algorithm A.2.

We describe Algorithm A.2 based on [103, p. 79]. The algorithm proceeds by building elementary paths from s. The vertices of the current elementary path are kept on a stack. A vertex is appended to an elementary path by a call to the procedure CIRCUIT and is deleted upon return from this call. When a vertex v is appended to a path it is blocked by setting BLOCKED(v) = true, so that v cannot be used twice on the same path. However, when we return from the call which blocks v, v is not necessarily unblocked. The idea is that we unblock a node with a sufficient delay such that any two unblockings of v are separated by either an output of a new circuit or a return to the main procedure.

Algorithm A.1 Computation of strongly connected components in a graph

INPUTS: A directed graph $\mathcal{G}_{\mathcal{V}} = (\mathcal{V}, \mathcal{E})$.
OUTPUT: The set of strongly connected components in $\mathcal{G}_{\mathcal{V}}$.

```
 1: Integer i;
 2: Procedure STRONGCONNECT(v);
 3:     Begin
 4:     i := i + 1;
 5:     NUMBER(v) := i;
 6:     LOWLINK(v) := NUMBER(v);
 7:     put v on stack of points;
 8:     For w in the set of descendants D_v do
 9:         Begin
10:         If w is not yet numbered then
11:             Begin comment (v, w) is a tree arc;
12:             STRONGCONNECT(w);
13:             LOWLINK(v) := min{LOWLINK(v), NUMBER(w)};
14:             end
15:         Else If NUMBER(w) < NUMBER(v) do
16:             Begin comment (v, w) is a frond or cross-link;
17:             If w is on stack of points then
18:                 Begin
19:                 LOWLINK(v) := min{LOWLINK(v), NUMBER(w)};
20:                 end
21:             end
22:         end
23:     If LOWLINK(v) = NUMBER(v) then
24:         Begin comment v is the root of a component;
25:         start new strongly connected component;
26:         While w on top of point stack satisfies NUMBER(w) ≥ NUMBER(v) do
27:             Begin
28:             delete w from point stack and put w in current component;
29:             end
30:         end
31:     end
32: i := 0;
33: empty stack of points;
34: For w a vertex do
35:     Begin
36:     If w is not yet numbered then
37:         Begin
38:         STRONGCONNECT(w);
39:         end
40:     end
41: end
```

Algorithm A.2 Computation of elementary cycles in a graph

INPUTS: A directed graph $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$.
OUTPUT: The set of elementary cycles in $\mathcal{G}_\mathcal{V}$.

```
 1: Begin
 2: Integer list array A_K(n), B(n);
 3: Logical array BLOCKED(N);
 4: Integer s;
 5: Logical Procedure CIRCUIT(v);
 6:     Begin Logical f;
 7:     Procedure UNBLOCK(u);
 8:         Begin
 9:         BLOCKED(u) := false;
10:         For w ∈ B(u) do
11:             Begin
12:             delete w from B(u);
13:             If BLOCKED(w) then
14:                 Begin
15:                 UNBLOCK(w);
16:                 end
17:             end
18:         end
19:     f := false;
20:     stack v;
21:     BLOCKED(v) := true;
22:     For w ∈ A_K(v) do
23:         Begin
24:         If w = s then
25:             Begin
26:             output circuit composed of stack followed by s;
27:             f := true;
28:             end
29:         Else If BLOCKED(w) = false then
30:             Begin
31:             If CIRCUIT(w) = true then
32:                 Begin
33:                 f := true;
34:                 end
35:             end
36:         end
37:     If f = true then
38:         Begin
39:         UNBLOCK(v);
40:         end
41:     Else For w ∈ A_K(v) do
42:         Begin
43:         If v ∉ B(w) then
44:             Begin
45:             put v on B(w);
46:             end
47:         end
48:     unstack v;
49:     CIRCUIT := f;
50:     end
```

Algorithm A.2 Computation of elementary cycles in a graph (cont.)

```
51: empty stack;
52: s := 1;
53: while s < n do
54:     Begin
55:     A_K := adjacency matrix of strong component K with least vertex in subgraph
56:             of G_V induced by {s, s + 1, ..., n};
57:     If A_K ≠ ∅ then
58:         Begin
59:         s := least vertex in V_K;
60:         For i ∈ V_K do
61:             Begin
62:             BLOCKED(i) := false;
63:             B(i) := ∅;
64:             end
65:         dummy := CIRCUIT(s);
66:         s := s + 1;
67:         end
68:     Else
69:         Begin
70:         s := n;
71:         end
72:     end
73: end
```

Appendix B

The expectation-maximization algorithm

In this appendix we present the expectation-maximization algorithm, and useful identities for the methods discussed in Chapters 4 and 5 of this thesis. The material in this appendix is based on [31, 170, 171].

B.1 The expectation-maximization algorithm

The expectation-maximization (EM) algorithm [56, 131] is an iterative procedure that at the k-th step seeks a value θ_k such that the likelihood is increased in the sense that $\log p_{\theta_k}(y_{1:T}|u_{1:T}) > \log p_{\theta_{k-1}}(y_{1:T}|u_{1:T})$.

The main idea of the EM algorithm is the postulation of a *missing* data set $x_{1:T}$. In this thesis, the missing data $x_{1:T}$ will be understood as the state sequence in the model structure (4.1) on page 63, but other choices are possible, and it can be considered as a design variable. Thus, we consider the joint log-likelihood function $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$ with respect to both the observed data $y_{1:T}$ and the missing data $x_{1:T}$. This approach assumes that maximizing the joint log-likelihood $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$ is easier than maximizing the marginal one $\log p_\theta(y_{1:T}|u_{1:T})$.

The EM algorithm then copes with $x_{1:T}$ being unavailable by forming an approximation $\mathbf{Q}(\theta, \theta_k)$ of $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$. The approximation used is the minimum variance estimate of $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$ given the observed data $(y_{1:T}, u_{1:T})$, and an assumption θ_k of the true parameter value. This minimum variance estimate is given by the conditional expectation [5]:

$$\begin{aligned}\mathbf{Q}(\theta, \theta_k) &:= \mathbf{E}\left\{\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T}) \middle| y_{1:T}, u_{1:T}\right\} \\ &= \int_{\mathcal{X}_{1:T}} \log p_\theta(x_{1:T}, y_{1:T}|u_{1:T}) p_{\theta_k}(x_{1:T}|y_{1:T}, u_{1:T})\, dx_{1:T}\,. \end{aligned} \tag{B.1}$$

The utility of this approach depends on the relationship between $\log p_\theta(y_{1:T}|u_{1:T})$, and the approximation $\mathbf{Q}(\theta, \theta_k)$ of $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$. This can be examined by

using the definition of conditional probability to write

$$\log p_\theta(x_{1:T},\, y_{1:T}|u_{1:T}) = \log p_\theta(x_{1:T}|y_{1:T},u_{1:T}) + \log p_\theta(y_{1:T}|u_{1:T})\,. \tag{B.2}$$

Taking the conditional expectation $\mathbf{E}\{\cdot|y_{1:T},u_{1:T}\}$ on both sides of equation (B.2) we obtain

$$\mathbf{Q}(\theta,\theta_k) = \log p_\theta(y_{1:T}|u_{1:T}) + \int_{\mathcal{X}_{1:T}} \log p_\theta(x_{1:T}|y_{1:T},u_{1:T}) p_{\theta_k}(x_{1:T}|y_{1:T},u_{1:T})\,dx_{1:T}\,. \tag{B.3}$$

Therefore,

$$\begin{aligned}\log p_\theta(y_{1:T}|u_{1:T}) - \log p_{\theta_k}(y_{1:T}|u_{1:T}) &= \mathbf{Q}(\theta,\theta_k) - \mathbf{Q}(\theta_k,\theta_k) \\ &+ \int_{\mathcal{X}_{1:T}} \log \frac{p_{\theta_k}(x_{1:T}|y_{1:T},u_{1:T})}{p_\theta(x_{1:T}|y_{1:T},u_{1:T})} p_{\theta_k}(x_{1:T}|y_{1:T},u_{1:T})\,dx_{1:T}\,.\end{aligned} \tag{B.4}$$

The integral in the right hand side of the equality in (B.4) is known as the *Kullback-Leibler divergence* metric, which is non-negative. Indeed, for $x>0$ we have that $-\log x \geq 1-x$, which implies

$$\begin{aligned}&-\int_{\mathcal{X}_{1:T}} \log \frac{p_\theta(x_{1:T}|y_{1:T},u_{1:T})}{p_{\theta_k}(x_{1:T}|y_{1:T},u_{1:T})} p_{\theta_k}(x_{1:T}|y_{1:T},u_{1:T})\,dx_{1:T} \\ &\qquad \geq \int_{\mathcal{X}_{1:T}} \left(1 - \frac{p_\theta(x_{1:T}|y_{1:T},u_{1:T})}{p_{\theta_k}(x_{1:T}|y_{1:T},u_{1:T})}\right) p_{\theta_k}(x_{1:T}|y_{1:T},u_{1:T})\,dx_{1:T} = 0\,,\end{aligned} \tag{B.5}$$

where the equality to zero is due to the fact that $p_\theta(x_{1:T}|y_{1:T},u_{1:T})$ is of unit area for any value of θ. As a consequence of (B.5) we have that

$$\log p_\theta(y_{1:T}|u_{1:T}) - \log p_{\theta_k}(y_{1:T}|u_{1:T}) \geq \mathbf{Q}(\theta,\theta_k) - \mathbf{Q}(\theta_k,\theta_k)\,. \tag{B.6}$$

Equation (B.6) is the key of the EM algorithm. Namely, choosing θ so that $\mathbf{Q}(\theta,\theta_k) > \mathbf{Q}(\theta_k,\theta_k)$ implies that the log-likelihood is also increased in that

$$\log p_\theta(y_{1:T}|u_{1:T}) > \log p_{\theta_k}(y_{1:T}|u_{1:T}).$$

The EM algorithm exploits this to deliver a sequence of values $\{\theta_k\}$ designed to be increasingly good approximations of the maximum likelihood estimate (1.6) on page 5. Algorithm B.1 summarizes the steps of the EM method.

Remark B.1 *For reference in the next sections, we define*

$$\mathcal{H}(\theta,\theta') := -\int_{\mathcal{X}_{1:T}} \log p_\theta(x_{1:T}|y_{1:T},u_{1:T}) p_{\theta'}(x_{1:T}|y_{1:T},u_{1:T})\,dx_{1:T}\,. \tag{B.8}$$

Algorithm B.1 EM algorithm

INPUTS: $y_{1:T}$ (observations), and $\log p_\theta(x_{1:T}, y_{1:T}|u_{1:T})$ (joint log-likelihood function).
OUTPUT: $\hat{\theta}_T$ (parameter estimate).

1: Set $k = 0$ and initialize θ_k such that $\log p_\theta(y_{1:T}|u_{1:T})$ is finite.
2: (Expectation (E) step): Compute $\mathbf{Q}(\theta, \theta_k)$.
3: (Maximization (M) step): Compute

$$\theta_{k+1} = \arg\max_{\theta \in \Theta} \mathbf{Q}(\theta, \theta_k)\,. \tag{B.7}$$

4: If not converged, update $k = k + 1$ and return to line 2. Otherwise, set $\hat{\theta}_T = \theta_{k+1}$, and stop the algorithm.

B.2 EM algorithm: useful identities

In this section we introduce useful results that can be derived from the intermediate expressions for the EM algorithm. In particular, we present the Fisher and Louis identities. To this end, we need the following assumption [31]:

Assumption B.1 *Assume that the following conditions hold:*

(i) The parameter Θ is an open subset of $\mathbb{R}^{n_\theta}$ (for some integer n_θ).

(ii) For any $\theta \in \Theta$, $p_\theta(y_{1:T}|u_{1:T})$ is positive and finite.

(iii) For any $(\theta, \theta') \in \Theta \times \Theta$,

$$\int_{\mathcal{X}_{1:T}} |\log p_\theta(x_{1:T}|y_{1:T}, u_{1:T})|\, p_{\theta'}(x_{1:T}|y_{1:T}, u_{1:T})\, dx_{1:T}\,,$$

is finite.

(iv) $p_\theta(y_{1:T}|u_{1:T})$ is twice continuously differentiable on Θ.

(v) For any $\theta' \in \Theta$, $\theta \to \mathcal{H}(\theta, \theta')$ is twice continuously differentiable on Θ. In addition,

$$\int_{\mathcal{X}_{1:T}} \left|\nabla_\theta^k \log p_\theta(x_{1:T}|y_{1:T}, u_{1:T})\right| p_{\theta'}(x_{1:T}|y_{1:T}, u_{1:T})\, dx_{1:T}\,,$$

is finite for $k = 1, 2$ and any $(\theta, \theta') \in \Theta \times \Theta$, and

$$\nabla_\theta^k \int_{\mathcal{X}_{1:T}} \log p_\theta(x_{1:T}|y_{1:T}, u_{1:T}) p_{\theta'}(x_{1:T}|y_{1:T}, u_{1:T})\, dx_{1:T} = \int_{\mathcal{X}_{1:T}} \nabla_\theta^k \log p_\theta(x_{1:T}|y_{1:T}, u_{1:T}) p_{\theta'}(x_{1:T}|y_{1:T}, u_{1:T})\, dx_{1:T}\,.$$

Fisher identity

A result derived from the EM algorithm is known as the *Fisher identity* [31, 68]:

Theorem B.1 *Consider that Assumption B.1 holds. Then:*

$$\nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta'} = \int_{\mathcal{X}_{1:T}} \nabla_\theta \log p_\theta(x_{1:T},\, y_{1:T}|u_{1:T})|_{\theta=\theta'}\, p_{\theta'}(x_{1:T}|y_{1:T},u_{1:T})\, dx_{1:T}\,, \quad \text{(B.9)}$$

and

$$\begin{aligned} -\,\nabla^2_\theta \log p_\theta(y_{1:T}|u_{1:T})\big|_{\theta=\theta'} = & \\ -\int_{\mathcal{X}_{1:T}} \nabla^2_\theta \log p_\theta(x_{1:T},\, y_{1:T}|u_{1:T})\big|_{\theta=\theta'}\, p_{\theta'}(x_{1:T}|y_{1:T},u_{1:T})\, dx_{1:T} & \\ +\int_{\mathcal{X}_{1:T}} \nabla^2_\theta \log p_\theta(x_{1:T}|y_{1:T},u_{1:T})\big|_{\theta=\theta'}\, p_{\theta'}(x_{1:T}|y_{1:T},u_{1:T})\, dx_{1:T}\,. & \quad \text{(B.10)} \end{aligned}$$

Proof *Expression* (B.9) *is* (B.1) *differentiated once under the integral sign (using* (B.3)*), and expression* (B.10) *is* (B.3) *differentiated twice under the integral sign.* ■

Remark B.2 *Expression* (B.9) *is known as the* Fisher identity*, and equation* (B.10) *is normally referred to as the* missing information principle *[124].*

Louis identity

The second result useful in this thesis is known as the *Louis identity* [31, 124]:

Theorem B.2 *(Louis identity) Consider that Assumption B.1 holds. Then:*

$$\begin{aligned} \nabla^2_\theta \log p_\theta(y_{1:T}|u_{1:T})\big|_{\theta=\theta'} + \{\nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta'}\}\{\nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta'}\}^\top = & \\ \int_{\mathcal{X}_{1:T}} \big[\nabla^2_\theta \log p_\theta(x_{1:T},\, y_{1:T}|u_{1:T})\big|_{\theta=\theta'} & \\ + \{\nabla_\theta \log p_\theta(x_{1:T},\, y_{1:T}|u_{1:T})|_{\theta=\theta'}\}\{\nabla_\theta \log p_\theta(x_{1:T},\, y_{1:T}|u_{1:T})|_{\theta=\theta'}\}^\top\big] & \\ p_{\theta'}(x_{1:T}|y_{1:T},u_{1:T})\, dx_{1:T}\,. & \quad \text{(B.11)} \end{aligned}$$

Proof *To prove* (B.11)*, we start from* (B.10) *and note that the second term on the right-hand side of the equality is the negative of an information matrix for the parameter θ associated with the probability density function $p_\theta(\cdot|y_{1:T},u_{1:T})$ evaluated at $\theta=\theta'$. Therefore, we can use the information matrix identity*

$$\int_{\mathcal{X}_{1:T}} \nabla^2_\theta \log p_\theta(x_{1:T}|y_{1:T},u_{1:T})\big|_{\theta=\theta'}\, p_{\theta'}(x_{1:T}|y_{1:T},u_{1:T})\, dx_{1:T} =$$

$$- \int_{\mathcal{X}_{1:T}} \{ \nabla_\theta \log p_\theta(x_{1:T}|y_{1:T}, u_{1:T})|_{\theta=\theta'} \} \{ \nabla_\theta \log p_\theta(x_{1:T}|y_{1:T}, u_{1:T})|_{\theta=\theta'} \}^\top p_{\theta'}(x_{1:T}|y_{1:T}, u_{1:T})\, dx_{1:T} \,. \quad \text{(B.12)}$$

This is a consequence of Assumption B.1, point (v), and the fact that $p_\theta(\cdot|y_{1:T}, u_{1:T})$ *is a probability density function for all values of* θ*, implying that*

$$\int_{\mathcal{X}_{1:T}} \nabla_\theta \log p_\theta(x_{1:T}|y_{1:T}, u_{1:T})|_{\theta=\theta'}\, p_{\theta'}(x_{1:T}|y_{1:T}, u_{1:T})\, dx_{1:T} = 0 \,. \quad \text{(B.13)}$$

Using the identity (B.2)*, and* (B.9) *we conclude that*

$$\begin{aligned} &\int_{\mathcal{X}_{1:T}} \{ \nabla_\theta \log p_\theta(x_{1:T}|y_{1:T}, u_{1:T})|_{\theta=\theta'} \} \{ \nabla_\theta \log p_\theta(x_{1:T}|y_{1:T}, u_{1:T})|_{\theta=\theta'} \}^\top p_{\theta'}(x_{1:T}|y_{1:T}, u_{1:T})\, dx_{1:T} = \\ &\int_{\mathcal{X}_{1:T}} \{ \nabla_\theta \log p_\theta(x_{1:T},\, y_{1:T}|u_{1:T})|_{\theta=\theta'} \} \{ \nabla_\theta \log p_\theta(x_{1:T},\, y_{1:T}|u_{1:T})|_{\theta=\theta'} \}^\top p_{\theta'}(x_{1:T}|y_{1:T}, u_{1:T})\, dx_{1:T} \\ &\quad + \{ \nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta'} \} \{ \nabla_\theta \log p_\theta(y_{1:T}|u_{1:T})|_{\theta=\theta'} \}^\top \,, \end{aligned} \quad \text{(B.14)}$$

which completes the proof. ■

Appendix C

The scenario approach

In this appendix we provide a brief discussion of the scenario approach for robust convex programs [27, 29], which is employed in this thesis.

C.1 Robust convex program and scenario approach

A robust program considers the optimization of a cost function when the problem is subject to uncertainty. The common structure for a robust program is given by

$$\begin{aligned} a^{\text{opt}} = \arg\min_{a \in \mathbb{A}} \quad & \max_{\delta \in \Delta} f(a, \delta) \\ \text{subject to} \quad & F(a, \delta) \leq 0, \text{ for all } \delta \in \Delta \,. \end{aligned} \tag{C.1}$$

In (C.1), $\mathbb{A} \subseteq \mathbb{R}^{n_a}$ is a convex and closed set representing the feasible actions, and $\Delta \subseteq \mathbb{R}^{n_\delta}$ is the set representing the uncertainty. We assume that $f \colon \mathbb{A} \times \Delta \to \mathbb{R}$ and $F \colon \mathbb{A} \times \Delta \to \mathbb{R}^{n_F}$ are continuous and convex in a, for any fixed value of $\delta \in \Delta$. Under these assumptions, the optimization (C.1) is a robust convex program.

The main difficulty to solve (C.1) relies in that the inequality constraint must be satisfied for the often infinite (even uncountable) number of elements in Δ. This difficulty makes the problem (C.1) computationally intractable [13].

An alternative to solve (C.1) is given by the scenario approach [27, 29]. The scenario approach provides a method to obtain a solution for an approximation of (C.1) based on an finite number of constraints.

To employ the scenario approach in (C.1), we first note that (C.1) can be rewritten without loss of generality as

$$\begin{aligned} u^{\text{opt}} = \arg\min_{u \in \mathbb{U}} \quad & c^\top u \\ \text{subject to} \quad & G(u, \delta) \leq 0, \text{ for all } \delta \in \Delta \,, \end{aligned} \tag{C.2}$$

where $c \in \mathbb{R}^{n_u}$, $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ is a convex and closed set and the entries of $G \colon \mathbb{U} \times \Delta \to \mathbb{R}^{n_G}$ are continuous and convex in u, for any fixed value of $\delta \in \Delta$. Then, the scenario

approach is applied to (C.2) as follows: draw samples $\{\delta_i\}_{i=1}^{N_\mathrm{s}} \in \Delta$ according to a probability measure $\mathbf{P}_\mathrm{s}$ defined over Δ (specified by the user) and solve

$$\begin{aligned} u_{N_\mathrm{s}}^{\mathrm{opt}} = \arg\min_{u \in \mathbb{U}} \quad & c^\top u \\ \text{subject to} \quad & G(u,\delta) \leq 0, \text{ for all } \delta \in \{\delta_i\}_{i=1}^{N_\mathrm{s}} . \end{aligned} \tag{C.3}$$

By comparing (C.2) with (C.3), we see that the infinite number of constraints in (C.2) is approximated by a finite number of them in (C.3). This implies that the solution given by (C.3) may not be feasible for the original problem (C.2). The feasibility of the solution (C.3) in the original problem (C.2) has been analyzed in [27, 29], and several bounds on the value of N_s providing specific confidence bounds on the feasibility of $u_{N_\mathrm{s}}^{\mathrm{opt}}$ in (C.2) have been developed [3, 4, 28, 29]. To date, the tightest bound on N_s is provided in [29], which will be introduced here.

To introduce the bound on [29], we define the *probability of violation*

$$V(u) := \mathbf{P}\left\{\delta \in \Delta \colon G(u,\delta) > 0\right\} . \tag{C.4}$$

The result in [29] is as follows: if for fixed numbers $\beta \in (0,1)$ and $\epsilon \in (0,1)$, N_s is chosen such that

$$\beta \leq \sum_{i=1}^{n_u - 1} \binom{N_\mathrm{s}}{i} \epsilon^i (1-\epsilon)^{N_\mathrm{s}-i} , \tag{C.5}$$

then, with probability no smaller than $1-\beta$, either the scenario problem (C.3) is unfeasible and, hence, also the initial robust convex problem (C.2); or the problem (C.3) is feasible and then its optimal solution $u_{N_\mathrm{s}}^{\mathrm{opt}}$ satisfies $V\left(u_{N_\mathrm{s}}^{\mathrm{opt}}\right) \leq \epsilon$.

Appendix D

Convergence results for Chapter 4

This appendix provides the theorems required to establish the convergence results in Chapter 4.

D.1 Convergence analysis

In this section we provide the main theorems to analyze the convergence properties of the graph theoretical method in input design. Specifically, we want to analyze how the quantities computed for every v_i ($v_i \in \mathcal{V}_\mathcal{C}$) relate to those computed using p defined as

$$p = \sum_{i=1}^{n_\mathcal{V}} \alpha_i v_i \,, \tag{D.1}$$

where $\alpha_i \geq 0$ for all $i \in \{1, \ldots, n_\mathcal{V}\}$, and

$$\sum_{i=1}^{n_\mathcal{V}} \alpha_i = 1 \,. \tag{D.2}$$

In addition, we want to study under which conditions the quantities computed using $u_{1:T}^{(i)}$ (where $u_{1:T}^{(i)}$ is generated from $v_i \in \mathcal{V}_\mathcal{C}$) converge to the inexact values as $T \to \infty$. To this end, we require the following definition:

Definition D.1 *(Exponentially forgetting functions [190]) A sequence of bounded functions $\{g_t\}_{t\geq 1}$, where $g_t \colon \mathcal{C}^t \to \mathcal{S}$, with $\mathcal{S}$ a normed space, is* exponentially forgetting *if there exists a $\lambda \in [0, 1)$ and $C > 0$ such that for every integer $n_m \geq 1$,*

$$\|g_t(u_{1:t}) - g_{n_m}(u_{t-n_m+1:t})\| \leq C\lambda^{n_m} \,, \tag{D.3}$$

for all $t > n_m$ and all $u_{1:t} \in \mathcal{C}^t$.

Remark D.1 *The constant C in Definition D.1 corresponds to a bound on the influence of $u_{1:t-n_m} \in \mathcal{C}^{t-n_m}$ on $\{g_t\}_{t\geq n_m+1}$ for all positive integer n_m and all $t \geq n_m + 1$.*

Based on the discussion provided in Chapter 2, we obtain the following lemma:

Lemma D.1 *Consider the Markov process $\{u_t\}_{t\geq -n_m+1}$ satisfying Assumption 2.1, associated with a sequence of probability mass functions $\{p_t\}_{t\geq 1}$ defined for all $t \geq 1$ as*

$$p_t(u_{1:t}|u_{-n_m+1:0}) := \prod_{k=1}^{t} p(u_{k-n_m+1:k}|u_{k-n_m:k-1})\,, \tag{D.4}$$

and an exponentially forgetting sequence of functions $\{g_t\}_{t\geq 1}$. Then, for every $n_m \in \mathbb{N}$ and $u_{-n_m+1:0} \in \mathcal{C}^{n_m}$,

$$\lim_{T\to\infty} \left\| \frac{1}{T}\sum_{t=1}^{T} \sum_{u_{1:t}\in\mathcal{C}^t} g_t(u_{1:t}) p_t(u_{1:t}|u_{-n_m+1:0}) \right. \\ \left. - \sum_{u_{1:n_m}\in\mathcal{C}^{n_m}} g_{n_m}(u_{1:n_m}) p^{\mathrm{st}}(u_{1:n_m}) \right\| \leq C\lambda^{n_m}\,. \tag{D.5}$$

Proof *For every $n_m \in \mathbb{N}$ we have that*

$$\begin{aligned}
&\left\| \frac{1}{T}\sum_{t=1}^{T} \sum_{u_{1:t}\in\mathcal{C}^t} g_t(u_{1:t}) p_t(u_{1:t}|u_{-n_m+1:0}) - \sum_{u_{1:n_m}\in\mathcal{C}^{n_m}} g_{n_m}(u_{1:n_m}) p^{\mathrm{st}}(u_{1:n_m}) \right\| \\
&\quad = \left\| \frac{1}{T}\sum_{t=1}^{T} \sum_{u_{1:t}\in\mathcal{C}^t} \left(g_t(u_{1:t}) - g_{n_m}(u_{t-n_m+1:t})\right) p_t(u_{1:t}|u_{-n_m+1:0}) \right. \\
&\qquad + \frac{1}{T}\sum_{t=1}^{T} \sum_{u_{t-n_m+1:t}\in\mathcal{C}^{n_m}} g_{n_m}(u_{t-n_m+1:t}) \left[p^{(t)}(u_{t-n_m+1:t}|u_{-n_m+1:0}) \right. \\
&\qquad\qquad \left. \left. - p^{\mathrm{st}}(u_{t-n_m+1:t}) \right] \right\| \\
&\quad \leq \frac{1}{T}\sum_{t=1}^{n_m} \sum_{u_{1:t}\in\mathcal{C}^t} \left\| g_t(u_{1:t}) - g_{n_m}(u_{t-n_m+1:t}) \right\| p_t(u_{1:t}|u_{-n_m+1:0}) \\
&\qquad + C\lambda^{n_m} \left| \frac{1}{T}\sum_{t=n_m+1}^{T} \sum_{u_{1:t}\in\mathcal{C}^t} p_t(u_{1:t}|u_{-n_m+1:0}) \right| \\
&\qquad + \left\| \frac{1}{T}\sum_{t=1}^{T} \sum_{u_{t-n_m+1:t}\in\mathcal{C}^{n_m}} g_{n_m}(u_{t-n_m+1:t}) \left[p^{(t)}(u_{t-n_m+1:t}|u_{-n_m+1:0}) \right. \right. \\
&\qquad\qquad \left. \left. - p^{\mathrm{st}}(u_{t-n_m+1:t}) \right] \right\|
\end{aligned}$$

$$\leq \frac{1}{T}\sum_{t=1}^{n_m}\sum_{u_{1:t}\in\mathcal{C}^t} \|g_t(u_{1:t}) - g_{n_m}(u_{t-n_m+1:t})\|\, p_t(u_{1:t}|u_{-n_m+1:0}) + C\lambda^{n_m}\frac{T-n_m}{T} + \frac{1}{T}\sum_{y\in\mathcal{C}^{n_m}} \|g_{n_m}(y)\| \sum_{t=1}^{T}\left|p^{(t)}(y|u_{-n_m+1:0}) - p^{\mathrm{st}}(y)\right|. \quad \text{(D.6)}$$

In the previous steps we have used the exponentially forgetting property of $\{g_t\}_{t\geq 1}$.

Based on the boundedness of g_t*, we conclude that*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{n_m}\sum_{u_{1:t}\in\mathcal{C}^t} \|g_t(u_{1:t}) - g_{n_m}(u_{t-n_m+1:t})\|\, p_t(u_{1:t}|u_{-n_m+1:0}) = 0\,, \quad \text{(D.7)}$$

In addition, note that

$$\lim_{T\to\infty} C\lambda^{n_m}\frac{T-n_m}{T} = C\lambda^{n_m}\,. \quad \text{(D.8)}$$

Finally, it is proven in [58, p. 217] that, for a Markov process satisfying Assumption 2.1, the last sum in t *in the right hand side of the inequality in* (D.6) *converges when* $T\to\infty$*. Hence,*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{y\in\mathcal{C}^{n_m}} \|g_{n_m}(y)\| \sum_{t=1}^{T}\left|p^{(t)}(y|u_{-n_m+1:0}) - p^{\mathrm{st}}(y)\right| = 0\,. \quad \text{(D.9)}$$

Based on Equations (D.7)*,* (D.8) *and* (D.9)*, we conclude that* (D.5) *holds.* ■

Lemma D.1 can be extended to periodic Markov processes:

Lemma D.2 *Consider the same conditions as in Lemma D.1, with the exception that* $\{u_t\}_{t\geq -n_m+1}$ *is an irreducible, periodic Markov process with period* $d\geq 2$ *and cyclic partition of* $\mathcal{C}^{n_m}$ *given by* $\{\mathcal{X}_i\}_{i=1}^{d}$*. Then the result of Lemma D.1 holds with* p^{st} *defined as the sum (scaled by* d^{-1}*) of the stationary probability mass functions of the Markov processes defined over each* $\mathcal{X}_i$*, for all* $i\in\{1,\ldots,d\}$*.*

Proof *The result follows from the fact that for the irreducible, and periodic Markov process* $\{u_t\}_{t\geq -n_m+1}$*, it holds that for all* y*,* $u_{-n_m+1:0}\in\mathcal{C}^{n_m}$*,*

$$\lim_{T\to\infty}\left|\frac{1}{T}\sum_{t=1}^{T} p^{(t)}(y|u_{-n_m+1:0}) - p^{\mathrm{st}}(y)\right| = 0\,, \quad \text{(D.10)}$$

with p^{st} *given as the sum (scaled by* d^{-1}*) of the stationary probability mass functions of the Markov processes defined over each* $\mathcal{X}_i$*, for all* $i\in\{1,\ldots,d\}$ *[58, p. 206]. Hence,*

$$\begin{aligned}
&\left\| \frac{1}{T} \sum_{t=1}^{T} \sum_{u_{t-n_m+1:t} \in \mathcal{C}^{n_m}} g_{n_m}(u_{t-n_m+1:t}) \left[p^{(t)}(u_{t-n_m+1:t} | u_{-n_m+1:0}) \; - p^{\mathrm{st}}(u_{t-n_m+1:t}) \right] \right\| \\
&\qquad = \left\| \sum_{y \in \mathcal{C}^{n_m}} g_{n_m}(y) \left[\frac{1}{N} \sum_{t=1}^{N} p^{(t)}(y | u_{-n_m+1:0}) - p^{\mathrm{st}}(y) \right] \right\| \\
&\qquad \leq \sum_{y \in \mathcal{C}^{n_m}} \| g_{n_m}(y) \| \left| \frac{1}{T} \sum_{t=1}^{T} p^{(t)}(y | u_{-n_m+1:0}) - p^{\mathrm{st}}(y) \right| \\
&\qquad \leq \sup_{x \in \mathcal{C}^{n_m}} \| g_{n_m}(x) \| \sum_{y \in \mathcal{C}^{n_m}} \left| \frac{1}{T} \sum_{t=1}^{T} p^{(t)}(y | u_{-n_m+1:0}) - p^{\mathrm{st}}(y) \right| . \quad \text{(D.11)}
\end{aligned}$$

By (D.10)*, the right hand side of the inequality in* (D.11) *goes to zero as* $T \to \infty$*. Hence, expression* (D.5) *also holds for an irreducible and periodic Markov process* $\{u_t\}_{t \geq -n_m+1}$*.* ■

Based on Lemmas D.1 and D.2 we obtain the following result:

Theorem D.1 *Consider the set* $\{p^{(i)}\}_{i=1}^{n_\mathcal{V}}$ *(*$p^{(i)} \colon \mathcal{C}^{n_m} \to \mathbb{R}$*) to which the transition probability mass functions of the irreducible Markov processes* $\{u_t^{(i)}\}_{t \geq -n_m+1}$ *converge as* $T \to \infty$*:*

$$\lim_{T \to \infty} \left| \frac{1}{T} \sum_{t=1}^{T} p^{(i,t)}(y | u_{-n_m+1:0}) - p^{(i)}(y) \right| = 0 \, , \quad \text{(D.12)}$$

for all $i \in \{1, \ldots, n_\mathcal{V}\}$ *and all* y, $u_{-n_m+1:0} \in \mathcal{C}^{n_m}$*. Define*

$$p^{\mathrm{opt}} := \sum_{i=1}^{n_\mathcal{V}} \alpha_i p^{(i)} \, , \quad \text{(D.13)}$$

with $\alpha_i \geq 0$, $\sum_{i=1}^{n_\mathcal{V}} \alpha_i = 1$*, and assume that the transition probability mass function of the irreducible Markov process* $\{u_t^{\mathrm{opt}}\}_{t \geq -n_m+1}$ *converges to* (D.13) *according to either* (2.8) *or* (D.12)*, for all* y, $u_{-n_m+1:0} \in \mathcal{C}^{n_m}$*. Consider an exponentially forgetting sequence of functions* $\{g_t\}_{t \geq 1}$ *satisfying the conditions stated in Lemma D.2. Then,*

$$\left\| \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{u_{1:t} \in \mathcal{C}^t} g_t(u_{1:t}) p_t^{\mathrm{opt}}(u_{1:t} | u_{-n_m+1:0}) \; - \sum_{i=1}^{n_\mathcal{V}} \alpha_i \, \overline{g}_{n_m}^{(i)} \right\| \leq 2C\lambda^{n_m} \, , \quad \text{(D.14)}$$

where

$$\overline{g}_{n_m}^{(i)} := \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{u_{1:t} \in \mathcal{C}^t} g_t(u_{1:t}) p_t^{(i)}(u_{1:t} | u_{-n_m+1:0}) \, , \quad \text{(D.15)}$$

and p_t^{opt}*,* $p_t^{(i)}$ *given by* (D.4) *for all* $i \in \{1, \ldots, n_\mathcal{V}\}$*.*

Proof *From Lemma D.2 we conclude for each Markov process* $\{u_t^{(i)}\}$ *that*

$$\lim_{T\to\infty}\left\|\frac{1}{T}\sum_{t=1}^{T}\sum_{u_{1:t}\in\mathcal{C}^t} g_t(u_{1:t})p_t^{(i)}(u_{1:t}|u_{-n_m+1:0}) - \sum_{u_{1:n_m}\in\mathcal{C}^{n_m}} g_{n_m}(u_{1:n_m})p^{(i)}(u_{1:n_m})\right\| \leq C\lambda^{n_m}\,. \quad \text{(D.16)}$$

In the same line, for the Markov process $\{u_t^{\text{opt}}\}_{t\geq -n_m+1}$ *we conclude from Lemma D.2 (if the Markov process is aperiodic) or Lemma D.2 (if the Markov process is periodic) that*

$$\lim_{T\to\infty}\left\|\sum_{i=1}^{n_{\mathcal{V}}}\alpha_i\,\overline{g}_T^{(i)} - \sum_{u_{1:n_m}\in\mathcal{C}^{n_m}} g_{n_m}(u_{1:n_m})p^{\text{opt}}(u_{1:n_m})\right\| \leq C\lambda^{n_m}\,, \quad \text{(D.17)}$$

where

$$\overline{g}_T^{(i)} := \frac{1}{T}\sum_{t=1}^{T}\sum_{u_{1:t}\in\mathcal{C}^t} g_t(u_{1:t})p_t^{(i)}(u_{1:t}|u_{-n_m+1:0})\,. \quad \text{(D.18)}$$

Due to the continuity of the norm and the properties of the Markov processes $\{u_t^{(i)}\}_{t\geq -n_m+1}$ *and* $\{u_t^{\text{opt}}\}_{t\geq -n_m+1}$, (D.17) *leads to*

$$\left\|\sum_{i=1}^{n_{\mathcal{V}}}\alpha_i\,\overline{g}_{n_m}^{(i)} - \sum_{u_{1:n_m}\in\mathcal{C}^{n_m}} g_{n_m}(u_{1:n_m})p^{\text{opt}}(u_{1:n_m})\right\| \leq C\lambda^{n_m}\,. \quad \text{(D.19)}$$

On the other hand, if we replace p^{opt} *for* $p^{(i)}$ *in* (D.16) *and using the continuity of the norm, we obtain*

$$\left\|\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{u_{1:t}\in\mathcal{C}^t} g_t(u_{1:t})p_t^{\text{opt}}(u_{1:t}|u_{-n_m+1:0}) - \sum_{u_{1:n_m}\in\mathcal{C}^{n_m}} g_{n_m}(u_{1:n_m})p^{\text{opt}}(u_{1:n_m})\right\| \leq C\lambda^{n_m}\,. \quad \text{(D.20)}$$

Combining (D.19) *and* (D.20) *gives* (D.14)*, which concludes the proof.* ■

Remark D.2 *Theorem D.1 establishes a bound on the difference between the convex combination of the quantities computed for every* v_i ($v_i \in \mathcal{V}_{\mathcal{C}}$) *and the one computed using* (D.1)*. Moreover, the theorem states that the difference can be made arbitrarily small by choosing* n_m *sufficiently large.*

Finally, the following theorem establishes the convergence of the quantities computed using $u^{(i)}_{1:T}$ (where $u^{(i)}_{1:T}$ is a periodic sequence generated from $v_i \in \mathcal{V}_{\mathcal{C}}$) as $T \to \infty$:

Theorem D.2 *If $\{u_t\}$, $u_t \in \mathcal{C}$, has period $\mathbf{T_u}$ satisfying $|u_t| \leq K$ for some $K \geq 0$, and $\{g_t\}_{t\geq 1}$, $g_t \colon \mathcal{C}^t \to \mathbb{R}^{n\times m}$, is an exponentially forgetting sequence of functions, then*

$$\begin{aligned}\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} g_t(u_{1:t}) &= \lim_{t\to\infty} \frac{1}{\mathbf{T_u}} \sum_{j=1}^{\mathbf{T_u}} g_t(u^{(j)}_{1:t}) \\ &= \lim_{t\to\infty} \int g_t(u_{1:t})\, d\mathbf{P}(u_{1:t})\,,\end{aligned} \tag{D.21}$$

where $\mathbf{P}$ is the probability measure of a Markov chain generating $\{u_t\}$ (a uniform probability distribution on the set of possible values of $u_{1:T}$), and $u^{(j)}_{1:t}$ is obtained from $\{u_t\}$ after $j-1$ time shift units.

Proof *Given $\varepsilon > 0$, take S as a multiple of $\mathbf{T_u}$ such that $C\lambda^S < \varepsilon$. Then, for every $t \geq S$,*

$$\|g_t(u_{1:t}) - g_S(u_{t-S+1:t})\| < C\lambda^S < \varepsilon\,. \tag{D.22}$$

On the other hand, since $\{u_t\}$ is periodic of period T, $u_{t-S+1:t}$ takes only a finite number of values for $t \geq S$ (at most S), we have that for $T = mS + n$ (with m, n positive integers, $|n| \leq S$):

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} g_t(u_{1:t}) &= \frac{(m-1)S}{T}\frac{1}{(m-1)S}\left[\sum_{t=1}^{S} g_t(u_{1:t}) + \sum_{t=S+1}^{mS} g_t(u_{1:t}) \right.\\ &\qquad \left. + \sum_{t=mS+1}^{T} g_t(u_{1:t})\right] \\ &= \frac{(m-1)S}{T}\frac{1}{(m-1)S}\sum_{t=S+1}^{mS}\left[g_S(u_{t-S+1:t}) + \eta_t\right] + \frac{1}{T}\left[\sum_{t=1}^{S} g_t(u_{1:t}) \right.\\ &\qquad \left. + \sum_{t=mS+1}^{T} g_t(u_{1:t})\right] \\ &= \frac{(m-1)S}{T}\lim_{t\to\infty}\frac{1}{\mathbf{T_u}}\sum_{j=1}^{\mathbf{T_u}} g_t(u^{(j)}_{1:t}) + \frac{1}{T}\sum_{t=S+1}^{mS}\left[\mu_t + \eta_t\right] \\ &\qquad + \frac{1}{T}\left[\sum_{t=1}^{S} g_t(u_{1:t}) + \sum_{t=mS+1}^{T} g_t(u_{1:t})\right]\,,\end{aligned} \tag{D.23}$$

where (μ_t, η_t) satisfies $\|\mu_t\| \leq \varepsilon$, $\|\eta_t\| \leq \varepsilon$ for all $t \geq S+1$. Thus, the norm of the second term in (D.23) *is bounded by 2ε. Moreover, the third term in* (D.23) *tends*

to 0 *as* $T \to \infty$ *(since it consists of a sum of a most* $2S$ *terms). Finally, we note that* $u_{1:t}$ *has only* $\mathbf{T_u}$ *possible values, as* $\{u_t\}$ *is periodic with period* $\mathbf{T_u}$*. Therefore,*

$$\left\| \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} g_t(u_{1:t}) - \lim_{t\to\infty} \frac{1}{\mathbf{T_u}} \sum_{j=1}^{\mathbf{T_u}} g_t(u_{1:t}^{(j)}) \right\| \leq 2\varepsilon \,, \tag{D.24}$$

and since ε *was arbitrary, we conclude that*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} g_t(u_{1:t}) = \lim_{t\to\infty} \frac{1}{\mathbf{T_u}} \sum_{j=1}^{\mathbf{T_u}} g_t(u_{1:t}^{(j)}) \,. \tag{D.25}$$

The last equality in (D.21) *follows since* $\mathbf{P}$ *assigns equal probability to* $\mathbf{T_u}$ *different sequences (corresponding to the possible sequences obtained by shifting* $\{u_t\}$*).* ■

Appendix E

On the power spectrum and the Fisher information matrix

This appendix derives the expressions for the nonnegativity constraint on the input power spectrum, as well as the Fisher information matrix for wide sense stationary processes. The material in this appendix is used in the numerical examples of Chapters 8 and 9.

E.1 Imposing nonnegative constraint on the power spectrum

When designing the input spectrum Φ_u, an important constraint is that $\Phi_u(\omega)$ must be a positive definite matrix for all $\omega \in [-\pi, \pi]$. This constraint can be imposed by the use of the following lemma:

Lemma E.1 *(Kalman-Yakubovich-Popov) Let $\{A, B, C, D\}$ be a controllable state space realization of $\sum_{\tau=0}^{m} c_\tau e^{j\omega\tau}$. Then, there exists a $Q = Q^\top$ such that*

$$\begin{bmatrix} Q - A^\top Q A & -A^\top Q B \\ -B^\top Q A & -B^\top Q B \end{bmatrix} + \begin{bmatrix} 0 & C^\top \\ C & D + D^\top \end{bmatrix} \succeq 0\,, \tag{E.1}$$

if and only if

$$\Phi_u(\omega) = \sum_{\tau=-m}^{m} c_{|\tau|} e^{j\omega\tau} \succeq 0\,, \text{ for all } \omega \in [-\pi, \pi]\,. \tag{E.2}$$

Proof *Follows from the positive real lemma [20, Section 2.7.2].* ■

E.2 Fisher information matrix for wide sense stationary processes

An expression required in the numerical examples of Chapters 8-9 is the Fisher information matrix as a function of the input power spectrum Φ_u. This quantity has been discussed in the literature for MIMO linear models [2, 12]. Here, we make use of [2, Theorem 4] for computing the Fisher information matrix. In the following lemma, for a matrix

$$A = \begin{bmatrix} a^{(1)} & \cdots & a^{(m)} \end{bmatrix} \in \mathbb{C}^{n \times m}, \tag{E.3}$$

where $a^{(i)} \in \mathbb{C}^{n \times 1}$ for all $i \in \{1, \ldots, m\}$, we define

$$\mathrm{vec}\{A\} := \begin{bmatrix} a^{(1)} \\ \vdots \\ a^{(m)} \end{bmatrix}. \tag{E.4}$$

Lemma E.2 *Consider the model* (9.3) *on page 161, where* $\{u_t\}$ *is a wide sense stationary process with power spectrum* Φ_u*, and* $\{e_t\}$ *is Gaussian white noise of zero mean and covariance matrix* Σ_e*. Assume that* $\{u_t\}$ *and* $\{e_t\}$ *are mutually independent. Then, the Fisher information matrix* (9.8) *on page 161 is an affine function of the power spectrum* Φ_u*, and is given by*

$$\overline{\mathcal{I}}_F(\theta_0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \overline{\Gamma(e^{j\omega}, \theta_0)}^{\top} \left\{ \Phi_{\mathcal{X}}(\omega) \otimes \Phi_v^{-1}(\omega, \theta_0) \right\} \Gamma(e^{j\omega}, \theta_0) d\omega, \tag{E.5}$$

where

$$\Gamma(e^{j\omega}, \theta_0) := \nabla_\theta^\top \mathcal{Z}_\theta(e^{j\omega})\big|_{\theta=\theta_0}, \tag{E.6}$$

$$\tag{E.7}$$

$$\mathcal{Z}_\theta(e^{j\omega}) := \begin{bmatrix} \mathrm{vec}\{G_\theta(e^{j\omega})\} \\ \mathrm{vec}\{H_\theta(e^{j\omega})\} \end{bmatrix}, \tag{E.8}$$

$$\tag{E.9}$$

$$\Phi_{\mathcal{X}}(\omega) := \begin{bmatrix} \Phi_u(\omega) & 0 \\ 0 & \Sigma_e \end{bmatrix}, \tag{E.10}$$

$$\tag{E.11}$$

$$\Phi_v(\omega, \theta_0) := H_{\theta_0}(e^{j\omega}) \, \Sigma_e \, \overline{H_{\theta_0}(e^{j\omega})}^{\top}, \tag{E.12}$$

and $A \otimes B$ *corresponds to the Kronecker product between the matrices* A *and* B*.*

Proof *We refer to [2, Theorem 4] for a proof of this lemma.* ■

Appendix F

Proof of Theorem 9.1

Here we present the proof of Theorem 9.1, which follows the proof of [39, Theorem 1.4.1] with minor modifications.

We say that $\{v(\gamma_{n_k}), v(\gamma_{n_k+1}), \ldots, v(\gamma_{m_k})\}$ crosses an interval $[\delta_1, \delta_2]$, if $n_k < m_k$, $v(\gamma_{n_k}) \le \delta_1$, $v(\gamma_{m_k}) \ge \delta_2$, and

$$\delta_1 < v(\gamma_i) < \delta_2\,, \text{ for all } n_k < i < m_k\,.$$

We first prove that the number of truncations in (9.32) may happen at most for a finite number of steps. Assume the reverse: there are infinitely many truncations occurring in (9.32). Since $v(\mu) < \min\{v(\delta), v(M)\}$ by Condition 9.5, there is an interval $[\delta_1, \delta_2]$ such that

$$[\delta_1, \delta_2] \subset (v(\mu), \min\{v(\delta), v(M)\})\,,\ \delta_1\delta_2 > 0\,, \tag{F.1}$$

and there are infinitely many $\{v(\gamma_{n_k}), \ldots, v(\gamma_{m_k})\}$, $k = 1, 2, \ldots$, that cross $[\delta_1, \delta_2]$.

Since $\{\gamma_k\}$ is bounded, we may extract a convergent subsequence from $\{\gamma_{n_k}\}$. We denote the extracted convergent subsequence still by $\{\gamma_{n_k}\}$: $\gamma_{n_k} \underset{k\to\infty}{\longrightarrow} \overline{\gamma}$, $\overline{\gamma} \in \Gamma_\delta^M$. It is clear that

$$v(\overline{\gamma}) = \lim_{k\to\infty} v(\gamma_{n_k}) = \delta_1\,. \tag{F.2}$$

Since the limit of $\{\gamma_{n_k}\}$ is located in the open set Γ_δ^M, there is an $\varepsilon > 0$ such that

$$\delta + \varepsilon < \gamma_{n_k} < M - \varepsilon \tag{F.3}$$

for all sufficiently large k.

To continue, we define

$$g(\gamma) := -1 + \gamma_0 \mathcal{R}_{\hat{\Theta},\Theta_0}\left(J(u_{1:T}^{\gamma}, \hat{\theta}_T, \theta_0)\right)\,, \tag{F.4}$$

$$y_{k+1} := g(\gamma_k) + \gamma_0 \boldsymbol{\nu}_{k+1|\gamma_k}\,, \tag{F.5}$$

Since $\{g(\gamma_k)\}$ is bounded (which follows from Condition 9.7) and the boundedness of $\{\gamma_k\}$, using (9.32) we have

$$\left|\sum_{i=n_k}^{m+1} a_i g(\gamma_i) + \gamma_0 \sum_{i=n_k}^{m+1} a_i \boldsymbol{\nu}_{i+1|\gamma_i}\right| \leq \frac{\varepsilon}{2}\,, \text{ for all } n_k \leq m \leq m(n_k, T_a)\,, \tag{F.6}$$

if T_a is small enough and k is large enough.

Expression (F.6) together with (F.3) implies that

$$\delta + \frac{\varepsilon}{2} < |\gamma_{n_k}| + \left|\sum_{i=n_k}^{m+1} a_i (g(\gamma_i) + \gamma_0 \boldsymbol{\nu}_{i+1|\gamma_i})\right| < M - \frac{\varepsilon}{2}\,, \quad \text{for all } n_k \leq m \leq m(n_k, T_a)\,. \tag{F.7}$$

Therefore, the magnitude of γ_{m+1},

$$\gamma_{m+1} = \gamma_m + a_m y_{m+1}\,, \quad m \geq n_k, \quad n_k \leq m \leq m(n_k, T_a)\,, \tag{F.8}$$

cannot reach neither the upper nor the lower bounds M and δ, respectively. In other words, the algorithm (9.32) corresponds to an untruncated stochastic approximation algorithm for m: $\{n_k \leq m \leq m(n_k, T_a)\}$ for small T_a and large k.

By the mean value theorem there exists z such that $\gamma_{n_k} < z < \gamma_{m(n_k,T_a)+1}$ and

$$\begin{aligned} v(\gamma_{m(n_k,T)+1}) - v(\gamma_{n_k}) = {} & (\gamma_{m(n_k,T)+1} - \gamma_{n_k}) \nabla_\gamma v(\overline{\gamma}) \\ & + (\gamma_{m(n_k,T)+1} - \gamma_{n_k})(\nabla_\gamma v(z) - \nabla_\gamma v(\overline{\gamma}))\,. \end{aligned} \tag{F.9}$$

We note that by (9.41) the left-hand side of (F.6) is of order $\mathcal{O}(T_a)$ for all sufficiently large k since $|g(\gamma_i)|$ is bounded. From this follows that i) for small enough $T_a > 0$ and large enough k

$$|v(\gamma_{m+1}) - v(\gamma_{n_k})| < \delta_2 - \delta_1, \quad v(\gamma_{m+1}) < \delta_2, \quad \text{for all } n_k \leq m \leq m(n_k, T_a)\,, \tag{F.10}$$

and hence $m(n_k, T_a) + 1 < m_k$, and ii) the last term in (F.9) is of $o(T_a)$ since $\nabla_\gamma v(z) - \nabla_\gamma v(\overline{\gamma}) \to 0$ as $T_a \to 0$. From (F.8) and (F.9) it follows that

$$\begin{aligned} v(\gamma_{m(n_k,T_a)+1}) - v(\gamma_{n_k}) &= \sum_{i=n_k}^{m(n_k,T_a)} a_i y_{i+1} \nabla_\gamma v(\overline{\gamma}) + o(T_a) \\ &= \sum_{i=n_k}^{m(n_k,T_a)} a_i g(\gamma_i) \nabla_\gamma v(\gamma_i) \\ &\quad + \sum_{i=n_k}^{m(n_k,T_a)} a_i g(\gamma_i)(\nabla_\gamma v(\overline{\gamma}) - \nabla_\gamma v(\gamma_i)) \end{aligned}$$

$$+\gamma_0 \sum_{i=n_k}^{m(n_k,T_a)} a_i \nabla_\gamma v(\overline{\gamma}) \boldsymbol{\nu}_{i+1|\gamma_i} + o(T_a)\,. \quad \text{(F.11)}$$

Since $\delta_1\delta_2 > 0$, the interval $[\delta_1, \delta_2]$ does not contain the origin. Noticing that $v(\gamma_m) \in [\delta_1, \delta_2]$ for all m: $\{n_k \le m \le m(n_k, T_a)\}$, we find $v(\gamma_m) \neq 0$, and that there is $\beta > 0$ such that

$$|\gamma_m - \gamma^\star| > \beta, \quad \text{for all } n_k \le m \le m(n_k, T_a) \quad \text{(F.12)}$$

for sufficiently small $T_a > 0$ and all large enough k. Then by Condition 9.5 there is $\eta > 0$ such that

$$g(\gamma_m)\nabla_\gamma v(\gamma_m) \le -\eta, \quad \text{for all } n_k \le m \le m(n_k, T_a) \quad \text{(F.13)}$$

for all large k and small enough T_a. As previously mentioned, $|\nabla_\gamma v(\overline{\gamma}) - \nabla_\gamma v(\gamma_i)| \underset{T_a \to 0}{\longrightarrow} 0$, from (F.11) we have

$$v(\gamma_{m(n_k,T_a)+1}) - v(\gamma_{n_k}) \le -\eta T_a + o(T_a) + o(1) \le -\frac{\eta}{2} T_a\,, \quad \text{(F.14)}$$

for sufficiently large k and small enough T_a, where $o(1)$ denotes a magnitude tending to zero as $k \to \infty$.

By considering (F.2), from (F.14) we have

$$v(\gamma_{m(n_k,T_a)+1}) < \delta_1 \quad \text{(F.15)}$$

for large k. However, it has been shown that

$$m(n_k, T_a) + 1 < m_k, \text{ i.e., } v(\gamma_{m(n_k,T_a)+1}) > \delta_1\,. \quad \text{(F.16)}$$

The resulting contradiction shows that the number of truncations in (9.32) can only be finite.

We have proved that starting from some large k_0, the algorithm (9.32) develops as

$$\gamma_{k+1} = \gamma_k + a_k y_{k+1} \quad k \ge k_0\,, \quad \text{(F.17)}$$

and $\{\gamma_k\}$ is bounded.

The next step in the proof is to show that $v(\gamma_k)$ converges.

Assume that it were not true. Then we would have

$$-\infty < \liminf_{k\to\infty} v(\gamma_k) < \limsup_{k\to\infty} v(\gamma_k) < \infty\,. \quad \text{(F.18)}$$

Then there would exist an interval $[\delta_1, \delta_2]$ not containing the origin and

$$\{v(\gamma_{n_k}), \ldots, v(\gamma_{m_k})\} \quad \text{(F.19)}$$

would cross $[\delta_1, \delta_2]$ for infinitely many k.

Following the previous steps, we assume without loss of generality that $\gamma_{n_k} \underset{k\to\infty}{\longrightarrow} \overline{\gamma}$. By the previous steps, we arrive at (F.11) and (F.14) for large k, resulting in a contradiction. Hence, $v(\gamma_k)$ tends to a finite limit as $k \to \infty$.

The final part of the proof corresponds to establish that $\gamma_k \underset{k\to\infty}{\longrightarrow} \gamma^\star$. To this end, we assume the reverse: there is a subsequence $x_{n_k} \underset{k\to\infty}{\longrightarrow} \overline{\gamma} \neq \gamma^\star$. Then there is a $\beta > 0$ such that $|\gamma_{n_k} - \gamma^\star| > \beta$ for all sufficiently large k. We still have (F.9), (F.11) and (F.14).

By letting $k \to \infty$ in (F.14) and based on the convergence of $v(\gamma_k)$, we arrive at the contradictory inequality

$$0 \leq -\frac{\eta}{2} T_a \,. \tag{F.20}$$

Therefore, $\gamma_k \underset{k\to\infty}{\longrightarrow} \gamma^\star$. ■

Appendix G

Inequalities for application oriented input design

This appendix provides complementary material to understand the formulations employed in Chapter 9, Section 9.5. The material in this appendix is partially taken from [8, 63].

G.1 Ellipsoidal approximation

Consider the application set

$$\Theta_{\text{app}}(\gamma) = \left\{ \theta \in \Theta : V_{\text{app}}(\theta, \theta_0) \leq \frac{1}{\gamma} \right\}, \tag{G.1}$$

where it is assumed that $\theta_0 \in \Theta$ is given.

To derive the ellipsoidal approximation of the application set (G.1), we employ the second order Taylor expansion of $V_{\text{app}}(\theta, \theta_0)$ at $\theta = \theta_0$:

$$\begin{aligned} V_{\text{app}}(\theta) &\approx V_{\text{app}}(\theta_0) + \nabla_\theta V_{\text{app}}(\theta, \theta_0)|_{\theta=\theta_0}^\top (\theta - \theta_0) \\ &\qquad + \frac{1}{2} (\theta - \theta_0)^\top \nabla_\theta^2 V_{\text{app}}(\theta, \theta_0)\big|_{\theta=\theta_0} (\theta - \theta_0) \\ &= \frac{1}{2} (\theta - \theta_0)^\top \nabla_\theta^2 V_{\text{app}}(\theta, \theta_0)\big|_{\theta=\theta_0} (\theta - \theta_0) . \end{aligned} \tag{G.2}$$

Thus, the ellipsoidal approximation of the application set (G.1) is given by

$$\Theta_{\text{app}}^{\mathcal{E}}(\gamma) := \left\{ \theta \in \Theta : (\theta - \theta_0)^\top \nabla_\theta^2 V_{\text{app}}(\theta, \theta_0)\big|_{\theta=\theta_0} (\theta - \theta_0) \leq \frac{2}{\gamma} \right\}. \tag{G.3}$$

To continue, we note that the set constraint $\Theta_{\text{id}}(\alpha) \subseteq \Theta_{\text{app}}^{\mathcal{E}}(\gamma)$ is fulfilled if for every $\theta \in \Theta$ satisfying

$$\frac{1}{\chi_\alpha^2(n_\theta)} (\theta - \theta_0)^\top T \overline{\mathcal{I}}_F(\theta_0) (\theta - \theta_0) \leq 1 , \tag{G.4}$$

it holds that

$$\frac{\gamma}{2}(\theta-\theta_0)^\top \nabla_\theta^2 V_{\rm app}(\theta,\theta_0)\big|_{\theta=\theta_0}(\theta-\theta_0)\le 1\,. \tag{G.5}$$

Finally, the desired LMI constraint is obtained from the following lemma [97, Lemma 3.1]:

Lemma G.1 *Let* $\overline{\mathcal{I}}_F(\theta_0)\succeq 0$ *and* $\nabla_\theta^2 V_{\rm app}(\theta,\theta_0)\big|_{\theta=\theta_0}\succeq 0$. *Then, the following statements are equivalent:*

(i) (G.4) *implies* (G.5).

(ii)

$$\frac{T}{\chi_\alpha^2(n_\theta)}\overline{\mathcal{I}}_F(\theta_0)\succeq\frac{\gamma}{2}\nabla_\theta^2 V_{\rm app}(\theta,\theta_0)\big|_{\theta=\theta_0}\,. \tag{G.6}$$

Proof *Follows immediately from [97, Lemma 3.1].* ■

Based on Lemma G.1, we obtain the LMI constraint in problem (9.49).

G.2 Worst case approach for application oriented input design

An alternative to the ellipsoidal approximation is to consider an scenario formulation for the set constraint $\Theta_{\rm id}(\alpha)\subseteq\Theta_{\rm app}(\gamma)$. To this end, we assume that $\theta_0\in\Theta$ is given. Then, we can rewrite the application set (G.1) as

$$\Theta_{\rm app}(\gamma)=\left\{\theta\in\Theta:\gamma\,\chi_\alpha^2(n_\theta)V_{\rm app}(\theta,\theta_0)\le\chi_\alpha^2(n_\theta)\right\}\,, \tag{G.7}$$

where we have multiplied the inequality in (G.1) by $\gamma\chi_\alpha^2(n_\theta)$.

Based on (G.7), we have that $\Theta_{\rm id}(\alpha)\subseteq\Theta_{\rm app}(\gamma)$ is fulfilled if for every $\theta\in\Theta$ satisfying

$$(\theta-\theta_0)^\top T\,\overline{\mathcal{I}}_F(\theta_0)\,(\theta-\theta_0)\le\chi_\alpha^2(n_\theta)\,, \tag{G.8}$$

it also holds that

$$\frac{\gamma\,\chi_\alpha^2(n_\theta)}{2}(\theta-\theta_0)^\top\nabla_\theta^2 V_{\rm app}(\theta,\theta_0)\big|_{\theta=\theta_0}(\theta-\theta_0)\le\chi_\alpha^2(n_\theta)\,, \tag{G.9}$$

The previous statement holds if the inequality

$$\gamma\,\chi_\alpha^2(n_\theta)V_{\rm app}(\theta,\theta_0)\le(\theta-\theta_0)^\top T\,\overline{\mathcal{I}}_F(\theta_0)\,(\theta-\theta_0)\,, \tag{G.10}$$

holds for all $\theta\in\Theta_{\rm app}(\gamma)$. The inequality (G.10) is satisfied for all $\theta\in\Theta_{\rm app}(\gamma)$ if

$$\max_{\theta\in\Theta_{\rm app}(\gamma)}\gamma\,\chi_\alpha^2(n_\theta)V_{\rm app}(\theta,\theta_0)-(\theta-\theta_0)^\top T\,\overline{\mathcal{I}}_F(\theta_0)\,(\theta-\theta_0)\le 0\,, \tag{G.11}$$

holds. Finally, the risk coherent inequality in (9.51) follows by putting the risk coherent measure $\mathcal{R}_{\Theta_0}$ into (G.11). We refer to [8] for more details on the scenario approach in application oriented input design.

Bibliography

[1] J.C. Agüero and G.C. Goodwin. Choosing between open- and closed-loop experiments in linear system identification. *IEEE Transactions on Automatic Control*, 52(8):1475–1480, August 2007.

[2] J.C. Agüero, C.R. Rojas, H. Hjalmarsson, and G.C. Goodwin. Accuracy of linear multiple-input multiple-output (MIMO) models obtained by maximum likelihood estimation. *Automatica*, 48(4):632–637, 2012.

[3] T. Alamo, R. Tempo, and E.F. Camacho. Improved sample size bounds for probabilistic robust control design: a packet-based strategy. In *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, United States, December 2007.

[4] T. Alamo, R. Tempo, and E.F. Camacho. Statistical learning theory: a pack-based strategy for uncertain feasibility and optimization problems. In *V.D. Blondel, S.P. Boyd, and H. Kimura Editors: Recent Advances in Learning and Control*, pages 1–14. Springer-Verlag, 2008.

[5] B.D.O. Anderson and J.B. Moore. *Optimal filtering*. Prentice Hall, Englewood Cliffs, 1979.

[6] D.A. Anderson, J.C. Tannehill, and R.H. Pletcher. *Computational fluid mechanics and heat transfer*. Hemisphere Publishing, 1984.

[7] C. Andrieu, N. De Freitas, and A. Doucet. Sequential MCMC for Bayesian model selection. In *Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics*, pages 130–134, Caesarea, Israel, June 1999.

[8] M. Annergren. *Application-oriented input design and optimization methods involving ADMM*. Ph.D. thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Sweden, 2016.

[9] P. Artzner, F. Delbaen, J. Eber, and D. Heath. Thinking Coherently. *Risk*, 10:68–71, 1997.

[10] P. Artzner, F. Delbaen, J. Eber, and D. Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

[11] K.J. Åström and B. Wittenmark. *Computer-controlled systems: theory and design, 3rd edition.* Prentice Hall, Englewood Cliffs, 1997.

[12] M. Barenthin Syberg. *Complexity issues, validation and input design for control in system identification.* Ph.D. thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Sweden, 2008.

[13] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[14] J.O. Berger. *Statistical decision theory and Bayesian analysis.* Springer, 2013.

[15] D. Bertsimas and A. Thiele. Robust and data-driven optimization: Modern decision-making under uncertainty. *INFORMS Tutorials in Operations Research*, pages 95–122, 2006.

[16] P. Billingsley. *Convergence of probability measures, 2nd edition.* John Wiley & Sons, 1999.

[17] F. Black and R. Litterman. Global portfolio optimization. *Financial analysts journal*, 48(5):28–43, 1992.

[18] X. Bombois, G. Scorletti, M. Gevers, P.M.J. Van den Hof, and R. Hildebrand. Least costly identification experiment for control. *Automatica*, 42(10):1651–1662, 2006.

[19] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov chain on a graph. *SIAM Review*, 46(4):667–689, October 2004.

[20] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM studies in applied and numerical mathematics, 1994.

[21] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[22] P. Boyle. *Gaussian processes for regression and optimisation.* Ph.D. thesis, Victoria University Wellington, Wellington, New Zealand, 2007.

[23] M.W. Braun, R. Ortiz-Mojica, and D.E. Rivera. Application of minimum crest factor multisinusoidal signals for "plant-friendly" identification of nonlinear process systems. *Control Engineering Practice*, 10(3):301–313, 2002.

[24] C. Brighenti. *On input design for system identification: input design using Markov chains.* Master thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2009.

[25] C. Brighenti, B. Wahlberg, and C.R. Rojas. Input design using Markov chains for system identification. In *Proceedings of the joint 48th Conference on Decision and Control and 28th Chinese Conference*, pages 1557–1562, Shanghai, China, December 2009.

[26] E. Brochu, V.M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical report, Department of Computer Science, University of British Columbia, Vancouver, Canada, 2009. Technical Report UBC TR-2009-23.

[27] G. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

[28] G.C. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[29] M.C. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.

[30] O. Cappé, S.J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[31] O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer, 2005.

[32] O. Cappé, C.P. Robert, and T. Rydén. Reversible jump, birth-and-death and more general continuous time Markov chain Monte Carlo samplers. *Journal of the Royal Statistical Society. Series B*, 65(3):679–700, 2003.

[33] B.P. Carlin and T.A. Louis. *Bayes and empirical Bayes methods for data analysis*. Chapman and Hall, London, 1996.

[34] C.K. Carter and R. Kohn. On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.

[35] G. Casella and R.L. Berger. *Statistical inference*. Duxbury Pacific Grove, 2002.

[36] M. Casini, A. Garulli, and A. Vicino. Input design for worst-case system identification with uniformly quantized measurements. In *15th IFAC Symposium on System Identification (SYSID)*, Saint-Malo, France, 2009.

[37] M. Casini, A. Garulli, and A. Vicino. Input design in worst-case system identification using binary sensors. *IEEE Transactions on Automatic Control*, 56(5):1186–1191, 2011.

[38] M. Casini, A. Garulli, and A. Vicino. Input design in worst-case system identification with quantized measurements. *Automatica*, 48(5):2297–3007, 2012.

[39] H.F. Chen. *Stochastic approximation and its applications*, volume 64. Kluwer Academic Publishers, 2002.

[40] N. Chen, Z. Qian, I.T. Nabney, and X. Meng. Wind power forecasts using Gaussian processes and numerical weather prediction. *IEEE Transactions on Power Systems*, 29(2):656–665, 2014.

[41] T. Chen, H. Ohlsson, and L. Ljung. On the estimation of transfer functions, regularizations and Gaussian processes−revisited. *Automatica*, 48(8):1525–1535, 2012.

[42] Y. Chow and M. Pavone. Stochastic optimal control with dynamic, time-consistent risk constraints. In *Proceedings of the American Control Conference*, pages 390–395, Washington D.C., United States, June 2013.

[43] Y. Chow and M. Pavone. A framework for time-consistent, risk-averse model predictive control: Theory and algorithms. In *Proceedings of the American Control Conference*, pages 4204–4211, Portland, United States, June 2014.

[44] G. Claeskens and N.L. Hjort. *Model selection and model averaging*. Cambridge University Press, 2008.

[45] D.R. Cox. *Planning of experiments*. Wiley, 1958.

[46] H. Cramér. *On the mathematical theory of risk*. Centraltryckeriet, 1930.

[47] H. Cramér. *Mathematical methods of statistics*. Princeton University Press, 1946.

[48] J. Dahlin and F. Lindsten. Particle filter-based Gaussian process optimisation for parameter inference. In *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, August 2014.

[49] J. Dahlin, F. Lindsten, T.B. Schön, and A. Wills. Hierarchical Bayesian ARX models for robust inference. In *Proceedings of the 16th IFAC Symposium on System Identification*, Brussels, Belgium, 2012.

[50] J. Dahlin and T.B. Schön. Getting started with particle Metropolis-Hastings for inference in nonlinear dynamical models. *arXiv preprint arXiv:1511.01707*, 2015.

[51] A.C. Davison and D.V. Hinkley. *Bootstrap methods and their application*. Cambridge University Press, 1997.

[52] N.G. de Bruijn and P. Erdös. A combinatorial problem. *Koninklijke Nederlandse Akademie v. Wetenschappen*, 49:758–764, 1946.

[53] A. De Cock, M. Gevers, and J. Schoukens. A preliminary study on optimal input design for nonlinear systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4931–4936, Florence, Italy, December 2013.

[54] A. De Cock, M. Gevers, and J. Schoukens. D-optimal input design for nonlinear FIR-type systems: A dispersion based approach. *Automatica*, 73:88–100, 2016.

[55] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B*, 68(3):411–436, 2006.

[56] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

[57] P.A.M. Dirac. *The principles of quantum mechanics, 4th edition.* Oxford university press, 1981.

[58] J.L. Doob. *Stochastic processes.* New York Wiley, 1953.

[59] R. Douc, A. Garivier, E. Moulines, and J. Olsson. Sequential Monte Carlo smoothing for general state space hidden Markov models. *Annals of Applied Probability*, 21(6):2109–2145, 2011.

[60] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice.* Springer, 2001.

[61] A. Doucet and A. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *The Oxford Handbook of Nonlinear Filtering.* Oxford University Press, 2011.

[62] R. Durrett. *Probability: theory and examples, 4th edition.* Cambridge university press, 2010.

[63] A. Ebadat. *On application oriented experiment design for closed-loop system identification.* Licentiate thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Sweden, 2015.

[64] L. El Ghaoui, M. Oks, and F. Oustry. Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Operations research*, 51(4):543–556, 2003.

[65] L. Favre and J. Galeano. Mean-modified value-at-risk optimization with hedge funds. *The Journal of Alternative Investments*, 5(2):21–25, 2002.

[66] V.V. Fedorov. *Theory of optimal experiments.* Academic Press, New York, 1972.

[67] R.A. Fisher. On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41(8):155–160, 1912.

[68] R.A. Fisher. Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22(5):700–725, 1925.

[69] M. Forgione, X. Bombois, P.M.J. Van den Hof, and H. Hjalmarsson. Experiment design for parameter estimation in nonlinear systems based on multilevel excitation. In *Proceedings of the 13th European Control Conference*, Strasbourg, France, June 2014.

[70] U. Forssell and L. Ljung. Identification for control: some results on optimal experiment design. In *Proceedings of the 37th Conference on Decision and Control*, pages 3384–3389, Tampa, United States, December 1998.

[71] U. Forssell and L. Ljung. Closed-loop identification revisited. *Automatica*, 35(7):1215–1241, 1999.

[72] U. Forssell and L. Ljung. Some results on optimal experiment design. *Automatica*, 36(5):749–756, 2000.

[73] E. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, 59(4):1569–1585, 2011.

[74] A.A. Gaivoronski and G. Pflug. Value-at-risk in portfolio optimization: properties and computational approach. *Journal of risk*, 7(2):1–31, 2005.

[75] M. Galrinho. *Least squares methods for system identification of structured models*. Licentiate thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2016.

[76] R. Garnett, M.A. Osborne, and S.J. Roberts. Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 209–219, Stockholm, Sweden, April 2010.

[77] K.F. Gauss. *Theoria motus corporum celestium,* English translation: Theory of the Motion of the Heavenly Bodies. Dover, 1963.

[78] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741, 1984.

[79] L. Gerencsér and H. Hjalmarsson. Adaptive input design in system identification. In *Proceedings of the 44th Conference on Decision and Control, and European Control Conference*, pages 4988–4993, Seville, Spain, December 2005.

[80] L. Gerencsér, H. Hjalmarsson, and J. Mårtensson. Identification of ARX systems with non−stationary inputs−asymptotic analysis with application to adaptive input design. *Automatica*, 45(3):623–633, 2009.

[81] M. Gevers. Identification for control: from the early achievements to the revival of experiment design. *European Journal of Control*, 11:1–18, 2005.

[82] M. Gevers and L. Ljung. Optimal experiment designs with respect to the intended model application. *Automatica*, 22(5):543–554, 1986.

[83] F. Giri and E.W. Bai. *Block-oriented nonlinear system identification*, volume 1. Springer, 2010.

[84] B.I. Godoy, G.C. Goodwin, J.C. Agüero, D. Marelli, and T. Wigren. On identification of FIR systems having quantized output data. *Automatica*, 46(9):1905–1915, 2011.

[85] B.I. Godoy, P.E. Valenzuela, C.R. Rojas, J.C. Agüero, and B. Ninness. A novel input design approach for systems with quantized output data. In *Proceedings of the 13th European Control Conference*, pages 1049–1054, Strasbourg, France, June 2014.

[86] G.C. Goodwin, J.C. Murdoch, and R.L. Payne. Optimal test signal design for linear SISO system identification. *International Journal of Control*, 17(1):45–55, 1973.

[87] G.C. Goodwin and R.L. Payne. *Dynamic system identification: experiment design and data analysis*. Academic Press, 1977.

[88] R.B. Gopaluni, T.B. Schön, and A.G. Wills. Input design for nonlinear stochastic dynamic systems - A particle filter approach. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, August 2011.

[89] M.C. Grant and S.P. Boyd. *The CVX users' guide*. CVX Research, Inc., 2nd. edition, October 2014.

[90] M. Green and D. Limebeer. *Linear robust control*. Dover, 2012.

[91] P.J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[92] F. Gustafsson and R. Karlsson. Statistical results for system identification based on quantized observations. *Automatica*, 45(12):2794–2801, 2009.

[93] D.I. Hastie and P.J. Green. Model choice using reversible jump Markov chain Monte Carlo. *Statistica Neerlandica*, 66(3):309–338, 2012.

[94] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[95] R. Hildebrand and M. Gevers. Identification for control: Optimal input design with respect to a worst-case ν-gap cost function. *SIAM Journal of Control Optimization*, 41(5):1586–1608, 2003.

[96] H. Hjalmarsson. From experiment design to closed-loop control. *Automatica*, 41:393–438, 2005.

[97] H. Hjalmarsson. System identification of complex and structured systems. *European Journal of Control*, 15(3):275–310, 2009.

[98] H. Hjalmarsson and F. Gustafsson. Composite modeling of transfer functions. *IEEE Transactions on Automatic Control*, 40(5):820–832, 1995.

[99] H. Hjalmarsson and J. Mårtensson. Optimal input design for identification of non-linear systems: Learning from the linear case. In *Proceedings of the American Control Conference*, pages 1572–1576, New York, United States, 2007.

[100] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, 1991.

[101] F. Hutter, H.H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the International Conference on Learning and Intelligent Optimization*, pages 507–523, Rome, Italy, January 2011.

[102] H. Jansson and H. Hjalmarsson. Input design via LMIs admitting frequency-wise model specifications in confidence regions. *IEEE Transactions on Automatic Control*, 50(10):1534–1549, 2005.

[103] D.B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975.

[104] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[105] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[106] T. Katayama. *Subspace methods for system identification*. Springer, 2006.

[107] J. Kiefer. General equivalence theory for optimum designs (approximate theory). *The Annals of Statistics*, 2(5):849–879, 1974.

[108] G. Kitagawa and S. Sato. Monte Carlo smoothing and self-organising state-space model. In A. Doucet, N. de Fretias, and N. Gordon, editors, *Sequential Monte Carlo methods in practice*, pages 177–195. Springer, 2001.

[109] J. Kleinberg and E. Tardos. *Algorithm design*. Pearson Education, 2006.

[110] A. Kong, J.S. Liu, and W.H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

[111] P. Krokhmal, J. Palmquist, and S. Uryasev. Portfolio optimization with conditional value-at-risk objective and constraints. *Journal of risk*, 4:43–68, 2002.

[112] C.A. Larsson. *Application-oriented experiment design for industrial model predictive control*. Ph.D. thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Sweden, 2014.

[113] C.A. Larsson, M. Annergren, H. Hjalmarsson, C.R. Rojas, X. Bombois, A. Mesbah, and P.E. Modén. Model predictive control with integrated experiment design for output error systems. In *Proceedings of the European Control Conference*, pages 3790–3795, Zurich, Switzerland, July 2013.

[114] C.A. Larsson, E. Geerardyn, and J. Schoukens. Robust input design for resonant systems under limited a priori information. In *Proceedings of the 16th IFAC Symposium on System Identification*, pages 1611–1616, Brussels, Belgium, July 2012.

[115] C.A. Larsson, H. Hjalmarsson, and C.R. Rojas. On optimal input design for nonlinear FIR-type systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 7220–7225, Atlanta, United States, December 2010.

[116] C.A. Larsson, C.R. Rojas, and H. Hjalmarsson. MPC oriented experiment design. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, August 2011.

[117] K. Lindqvist and H. Hjalmarsson. Optimal input design using linear matrix inequalities. In *Proceedings of the 12th IFAC Symposium on System Identification*, Santa Barbara, California, United States, July 2000.

[118] K. Lindqvist and H. Hjalmarsson. Identification for control: Adaptive input design using convex optimization. In *Proceedings of the 40th Conference on Decision and Control*, pages 4326–4331, Orlando, United States, December 2001.

[119] F. Lindsten, M.I. Jordan, and T.B. Schön. Particle Gibbs with ancestor sampling. *Journal of Machine Learning Research*, 15(1):2145–2184, 2014.

[120] F. Lindsten and T.B. Schön. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143, 2013.

[121] D.J. Lizotte, T. Wang, M.H. Bowling, and D. Schuurmans. Automatic gait optimization with Gaussian Process regression. In *Proceedings of the 20th International Joint Conference on Artificial Inteligence*, volume 7, pages 944–949, Hyderabad, India, January 2007.

[122] L. Ljung. *System Identification. Theory for the User, 2nd edition*. Prentice-Hall, Upper Saddle River, 1999.

[123] L. Ljung. Prediction error estimation methods. Technical report, Linköping University, Department of Electrical Engineering, October 2001.

[124] T.A. Louis. Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 44(2):226–233, 1982.

[125] K. Mahata, J. Schoukens, and A. De Cock. Design of Gaussian inputs for Wiener model identification. In *Proceedings of the 17th IFAC Symposium on System Identification*, Beijing, China, October 2015.

[126] K. Mahata, J. Schoukens, and A. De Cock. Information matrix and D-optimal design with Gaussian inputs for Wiener model identification. *Automatica*, 69:65–77, 2016.

[127] R. Marchant and F. Ramos. Bayesian optimisation for intelligent environmental monitoring. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2242–2249, Vilamoura, Portugal, October 2012.

[128] J. Mårtensson and H. Hjalmarsson. Robust input design using sum of squares constraints. In *Proceedings of the 14th IFAC Symposium on System Identification*, pages 1352–1357, Newcastle, Australia, March 2006.

[129] R. Martinez-Cantin, N. de Freitas, A. Doucet, and J.A. Castellanos. Active policy learning for robot planning and exploration under uncertainty. In *Proceedings of Robotics: Science and Systems*, pages 321–328, Atlanta, United States, June 2007.

[130] J. Max. Quantization for minimum distortion. *IRE Transactions on Information Theory*, 46(12):3423–3426, 1998.

[131] G.J. McLachlan and T. Krishnan. *The EM algorithm and extensions, 2nd edition*. John Wiley & Sons, 2008.

[132] R.K. Mehra. Optimal input signals for parameter estimation in dynamic systems – survey and new results. *IEEE Transactions on Automatic Control*, 19(6):753–768, December 1974.

[133] I. Meilijson. A fast improvement to the EM algorithm on its own terms. *Journal of the Royal Statistical Society: Series B*, pages 127–138, 1989.

[134] S.P. Meyn and R.L. Tweedie. *Markov chains and stochastic stability, 2nd edition*. Cambridge Univeristy Press, 2008.

[135] E.A. Nadaraya. On estimating regression. *Theory of Probability & its Applications*, 9(1):141–142, 1964.

[136] O. Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer, 2013.

[137] B. Ninness and S. Henriksen. Bayesian system identification via Markov chain Monte Carlo techniques. *Automatica*, 46(1):40–51, 2010.

[138] J. Olsson, O. Cappé, R. Douc, and E. Moulines. Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179, 2008.

[139] M.A. Osborne, R. Garnett, and S.J. Roberts. Gaussian processes for global optimization. In 3*rd International Conference on Learning and Intelligent Optimization*, pages 1–15, 2009.

[140] M.A. Osborne, S.J. Roberts, A. Rogers, S.D. Ramchurn, and N.R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *Proceedings of the* 7*th Conference on Information Processing in Sensor Networks*, pages 109–120, St. Louis, United States, April 2008.

[141] A. Papoulis. *Probability, random variables and stochastic processes, 2nd edition*. McGraw-Hill, 1987.

[142] A. Pázman and L. Pronzato. Quantile and probability-level criteria for nonlinear experimental design. In *J. López-Fidalgo, J.M. Rodríguez-Díaz, and B. Torsney Editors: mODa 8-Advances in Model-Oriented Design and Analysis*, pages 157–164. Springer, 2007.

[143] A.F. Perold. Large-scale portfolio optimization. *Management science*, 30(10):1143–1160, 1984.

[144] V. Peterka. Bayesian system identification. *Automatica*, 17(1):41–53, 1981.

[145] G.Ch. Pflug. Some remarks on the value-at-risk and the conditional value-at-risk. In *S.P. Uryasev Editor: Probabilistic constrained optimization*, pages 272–281. Springer, 2000.

[146] G. Pillonetto and G. De Nicolao. A new kernel−based approach for linear system identification. *Automatica*, 46(1):81–93, 2010.

[147] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682, 2014.

[148] M.K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

[149] K. Postek, D. Den Hertog, and B. Melenberg. Tractable counterparts of distributionally robust constraints on risk measures. *CentER Discussion Paper Series No. 2014-031*, 2014.

[150] G. Prando, A. Chiuso, and G. Pillonetto. Bayesian and regularization approaches to multivariable linear system identification: the role of rank penalties. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 1482–1487, Los Angeles, United States, Dec. 2014.

[151] L. Pronzato. Adaptive optimization and D-optimum experimental design. *Annals of Statistics*, 28(6):1743–1761, 2000.

[152] L. Pronzato and A. Pázman. Design of experiments in nonlinear models. *Lecture Notes in Statistics*, 212, 2013.

[153] L. Pronzato and É. Walter. Robust experiment design via stochastic approximation. *Mathematical Biosciences*, 75(1):103–120, 1985.

[154] C.E. Rasmussen and H. Nickish. *Gaussian process regression and classification toolbox, version 3.6*, July 2015.

[155] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for Machine Learning*. MIT press, 2006.

[156] H.E. Rauch, F. Tung, and C.T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, August 1965.

[157] R.S. Risuleo. *System identification with input uncertainties: an EM kernel-based approach*. Licentiate thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2016.

[158] D.E. Rivera, H. Lee, M.W. Braun, and H.D. Mittelmann. Plant-friendly system identification: a challenge for the process industries. In *Proceeding of the 13th IFAC Symposium on System Identification*, pages 917–922, Rotterdam, The Netherlands, August 2003.

[159] C. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer, 2007.

[160] C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer, 2004.

[161] R. Rockafellar. Coherent approaches to risk in optimization under uncertainty. *Tutorials in operations research*, 3:38–61, 2007.

[162] R. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[163] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[164] C.R. Rojas, J.C. Agüero, J.S. Welsh, G.C. Goodwin, and A. Feuer. Robustness in experiment design. *IEEE Transactions on Automatic Control*, 57(4):860–874, 2012.

[165] C.R. Rojas, H. Hjalmarsson, L. Gerencsér, and J. Mårtensson. An adaptive method for consistent estimation of real-valued non-minimum phase zeros in stable LTI systems. *Automatica*, 47(7):1388–1398, 2011.

[166] C.R. Rojas, D. Katselis, H. Hjalmarsson, R. Hildebrand, and M. Bengtsson. Chance constrained input design. In *Proceedings of the 50th Conference on Decision and Control, and European Control Conference*, pages 2957–2962, Orlando, United States, December 2011.

[167] C.R. Rojas, J.S. Welsh, G.C. Goodwin, and A. Feuer. Robust optimal experiment design for system identification. *Automatica*, 43(6):993–1008, June 2007.

[168] W. Rudin. *Principles of Mathematical Analysis, 3rd edition*. McGraw-Hill, 1976.

[169] B. Sánchez, C.R. Rojas, G. Vandersteen, R. Bragos, and J. Schoukens. On the calculation of the Data-optimal multisine excitation power spectrum for broadband impedance spectroscopy measurements. *Measurement Science and Technology*, 23(8):1–15, 2012.

[170] T.B. Schön and F. Lindsten. Learning of dynamical systems - Particle filters and Markov chain methods. (forthcoming), 2015.

[171] T.B. Schön, A. Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, January 2011.

[172] S. Searle, G. Casella, and C. McCulloch. *Variance components*. John Wiley, 1992.

[173] M. Segal and E. Weinstein. A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems. *IEEE Transactions on Information Theory*, 35(3):682–687, 1989.

[174] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[175] Y. Shen, W. Stannat, and K. Obermayer. A unified framework for risk-sensitive Markov control processes. In *Proceedings of the 53rd Conference on Decision and Control*, Los Angeles, United States, December 2014.

[176] J. Sjöberg, H. Hjalmarsson, and L. Ljung. Neural networks in system identification. Technical report, Linköping University, 1994.

[177] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

[178] T. Söderström and P. Stoica. Comparison of some instrumental variable methods: consistency and accuracy aspects. *Automatica*, 17(1):101–115, 1981.

[179] T. Söderström and P. Stoica. *Instrumental variable methods for system identification*. Springer, 1983.

[180] T. Söderström and P. Stoica. *System identification*. Prentice Hall, 1989.

[181] J.C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons, 2005.

[182] H. Suzuki and T. Sugie. On input design for system identification in time domain. In *Proceedings of the European Control Conference*, Kos, Greece, July 2007.

[183] E. Taghavi, F. Lindsten, L. Svensson, and T.B. Schön. Adaptive stopping for fast particle smoothing. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013.

[184] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, June 1972.

[185] R. Tóth. *Modeling and identification of linear parameter-varying systems*, volume 403. Springer, 2010.

[186] P.E. Valenzuela. *Optimal input design for nonlinear dynamical systems: a graph-theory approach*. Licentiate Thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Sweden, 2014.

[187] P.E. Valenzuela, J. Dahlin, C.R. Rojas, and T.B. Schön. A graph/particle-based method for experiment design in nonlinear systems. In *Proceedings of the 19th IFAC World Congress*, pages 1404–1409, Cape Town, South Africa, August 2014.

[188] P.E. Valenzuela, J. Dahlin, C.R. Rojas, and T.B. Schön. On robust input design for nonlinear dynamical models. *Automatica*, 2016. (Accepted).

[189] P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. Optimal input design for dynamic systems: a graph theory approach. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5740–5745, Florence, Italy, December 2013.

[190] P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. A graph theoretical approach to input design for identification of nonlinear dynamical models. *Automatica*, 51:233–242, 2015.

[191] P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. Uncertainty in system identification: learning from the theory of risk. In *Proceedings of the 17th IFAC Symposium on System Identification*, pages 1053–1058, Beijing, China, October 2015.

[192] P. Van Overschee and B. De Moor. *Subspace identification for linear systems: theory, implementation, applications*. Kluwer academic publishers, 1996.

[193] M. Viberg. Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31(12):1835–1851, 1995.

[194] T.L. Vincent, C. Novara, K. Hsu, and K. Poolla. Input design for structured nonlinear system identification. In *15th IFAC Symposium on System Identification*, pages 174–179, Saint-Malo, France, July 2009.

[195] T.L. Vincent, C. Novara, K. Hsu, and K. Poolla. Input design for structured nonlinear system identification. *Automatica*, 46(6):990–998, 2010.

[196] B. Wahlberg, H. Hjalmarson, and M. Annergren. On optimal input design in system identification for control. In *49th IEEE Conference on Decision and Control*, Atlanta, United States, 2010.

[197] B. Wahlberg, H. Hjalmarsson, and P. Stoica. On optimal input signal design for frequency response estimation. In *49th IEEE Conference on Decision and Control*, pages 302–307, Atlanta, United States, 2010.

[198] A. Wald. Note on the consistency of the maximum likelihood estimate. *The Annals of Mathematical Statistics*, 20(4):595–601, 1949.

[199] L.Y. Wang and G.G. Yin. Asymptotically efficient parameter estimation using quantized output observations. *Automatica*, 43(7):1178–1191, 2007.

[200] L.Y. Wang and G.G. Yin. Quantized identification with dependent noise and Fisher information ratio of communication channels. *IEEE Transactions on Automatic Control*, 55(3):674–690, 2010.

[201] L.Y. Wang, G.G. Yin, J-F. Zhang, and Y. Zhao. *System identification with quantized observations*. Springer, 2010.

[202] Z. Wang, B. Shakibi, L. Jin, and N. de Freitas. Bayesian multi-scale optimistic optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 1005–1014, Reykjavik, Iceland, April 2014.

[203] Z. Wang, M. Zoghi, D. Matheson, F. Hutter, and N. de Freitas. Bayesian optimization in high dimensions via random embeddings. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 1778–1784, Beijing, China, August 2013.

[204] J.S. Welsh and C.R. Rojas. A scenario based approach to robust experiment design. In *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009.

[205] H. White. *Estimation, inference and specification analysis*. Cambridge university press, 1996.

[206] P. Whittle. Some general points in the theory of optimal experiment design. *Journal of Royal Statistical Society: Series B*, 1:123–130, 1973.

[207] T. Wigren. Adaptive filtering using quantized output measurements. *IEEE Transactions on Signal Processing*, 46(12):3423–3426, 1998.

[208] A. Wills, T.B. Schön, F. Lindsten, and B. Ninness. Estimation of linear systems using a Gibbs sampler. In *Proceedings of the 16th IFAC Symposium on System Identification*, Brussels, Belgium, 2012.

[209] Z.D. Yuan and L. Ljung. Black−box identification of multivariable transfer functions−asymptotic properties and optimal input design. *International Journal of Control*, 40(2):233–256, 1984.

[210] A. Zaman. Stationarity on finite strings and shift register sequences. *The Annals of Probability*, 11(3):678–684, 1983.

[211] M. Zarrop. *Optimal experiment design for dynamic system identification*. Springer, 1979.

[212] Q. Zhang and A. Benveniste. Wavelet networks. *IEEE Transactions on Neural Networks*, 3(6):889–898, 1992.

[213] Y. Zhao, J-F. Zhang, L.Y. Wang, and G.G. Yin. Identification of Hammerstein systems with quantized observations. *SIAM Journal on Control and Optimization*, 48(7):4352–4376, 2010.

[214] K. Zhou and J.C. Doyle. *Essentials of robust control*. Prentice Hall, Englewood Cliffs, 1998.

[215] K. Zhou, J.C. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, 1996.