Communications and Control Engineering

For further volumes:
www.springer.com/series/61

Roberto Tempo • Giuseppe Calafiore •
Fabrizio Dabbene

Randomized Algorithms for Analysis and Control of Uncertain Systems

With Applications

Second Edition

Springer

Roberto Tempo
CNR - IEIIT
Politecnico di Torino
Turin, Italy

Fabrizio Dabbene
CNR - IEIIT
Politecnico di Torino
Turin, Italy

Giuseppe Calafiore
Dip. Automatica e Informatica
Politecnico di Torino
Turin, Italy

ISSN 0178-5354 Communications and Control Engineering
ISBN 978-1-4471-4609-4 ISBN 978-1-4471-4610-0 (eBook)
DOI 10.1007/978-1-4471-4610-0
Springer London Heidelberg New York Dordrecht

Library of Congress Control Number: 2012951683

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

It follows that the Scientist, like the Pilgrim, must wend a straight and narrow path between the Pitfalls of Oversimplification and the Morass of Overcomplication.

Richard Bellman, 1957

to Chicchi and Giulia for their remarkable endurance
R.T.
to my daughter Charlotte
G.C.
to my lovely kids Francesca and Stefano, and to Paoletta, forever no matter what
F.D.

Foreword

The topic of randomized algorithms has had a long history in computer science. See [290] for one of the most popular texts on this topic. Almost as soon as the first NP-hard or NP-complete problems were discovered, the research community began to realize that problems that are difficult in the worst-case need not always be so difficult on average. On the flip side, while assessing the performance of an algorithm, if we do not insist that the algorithm must *always* return *precisely* the right answer, and are instead prepared to settle for an algorithm that returns *nearly the right answer most of the time*, then some problems for which "exact" polynomial-time algorithms are not known turn out to be tractable in this weaker notion of what constitutes a "solution." As an example, the problem of counting the number of satisfying assignments of a Boolean formula in disjunctive normal form (DNF) can be "solved" in polynomial time in this sense; see [288], Sect. 10.2.

Sometime during the 1990s, the systems and control community started taking an interest in the computational complexity of various algorithms that arose in connection with stability analysis, robustness analysis, synthesis of robust controllers, and other such quintessentially "control" problems. Somewhat to their surprise, researchers found that many problems in analysis and synthesis were in fact NP-hard if not undecidable. Right around that time the first papers on addressing such NP-hard problems using randomized algorithms started to appear in the literature. A parallel though initially unrelated development in the world of machine learning was to use powerful results from empirical process theory to quantity the "rate" at which an algorithm will learn to do a task. Usually this theory is referred to as statistical learning theory, to distinguish it from computational learning theory in which one is also concerned with the running time of the algorithm itself.

The authors of the present monograph are gracious enough to credit me with having initiated the application of statistical learning theory to the design of systems affected by uncertainty [405, 408]. As it turned out, in almost all problems of controller synthesis it is not necessary to worry about the actual execution time of the algorithm to compute the controller; hence statistical learning theory was indeed the right setting for studying such problems. In the world of controller synthesis, the analog of the notion of an algorithm that returns more or less the right answer most

of the time is a controller that stabilizes (or achieves nearly optimal performance for) most of the set of uncertain plants. With this relaxation of the requirements on a controller, most if not all of the problems previously shown to be NP-hard now turned out to be tractable in this relaxed setting. Indeed, the application of randomized algorithms to the synthesis of controllers for uncertain systems is by now a well-developed subject, as the authors point out in the book; moreover, it can be confidently asserted that the theoretical foundations of the randomized algorithms were provided by statistical learning theory.

Having perhaps obtained its initial impetus from the robust controller synthesis problem, the randomized approach soon developed into a subject on its own right, with its own formalisms and conventions. Soon there were new abstractions that were motivated by statistical learning theory in the traditional sense, but are not strictly tied to it. An example of this is the so-called "scenario approach." In this approach, one chooses a set of "scenarios" with which a controller must cope; but the scenarios need not represent randomly sampled instances of uncertain plants. By adopting this more general framework, the theory becomes cleaner, and the precise role of each assumption in determining the performance (e.g. the rate of convergence) of an algorithm becomes much clearer.

When it was first published in 2005, the first edition of this book was among the first to collect in one place a significant body of results based on the randomized approach. Since that time, the subject has become more mature, as mentioned above. Hence the authors have taken the opportunity to expand the book, adopting a more general set of problem formulations, and in some sense moving away from controller design as the main motativating problem. Though controller design still plays a prominent role in the book, there are several other applications discussed therein. One important change in the book is that bibliography has nearly doubled in size. A serious reader will find a wealth of references that will serve as a pointer to practically all of the relevant literature in the field. Just as with the first edition, I have no hesitation in asserting that the book will remain a valuable addition to everyone's bookshelf.

Hyderabad, India M. Vidyasagar
June 2012

Foreword to the First Edition

The subject of control system synthesis, and in particular robust control, has had a long and rich history. Since the 1980s, the topic of robust control has been on a sound mathematical foundation. The principal aim of robust control is to ensure that the performance of a control system is satisfactory, or nearly optimal, even when the system to be controlled is itself not known precisely. To put it another way, the objective of robust control is to assure satisfactory performance even when there is "uncertainty" about the system to be controlled.

During the two past two decades, a great deal of thought has gone into modeling the "plant uncertainty." Originally the uncertainty was purely "deterministic," and was captured by the assumption that the "true" system belonged to some sphere centered around a nominal plant model. This nominal plant model was then used as the basis for designing a robust controller. Over time, it became clear that such an approach would often lead to rather conservative designs. The reason is that in this model of uncertainty, every plant in the sphere of uncertainty is deemed to be equally likely to occur, and the controller is therefore obliged to guarantee satisfactory performance for every plant within this sphere of uncertainty. As a result, the controller design will trade off optimal performance at the nominal plant condition to assure satisfactory performance at off-nominal plant conditions.

To avoid this type of overly conservative design, a recent approach has been to assign some notion of probability to the plant uncertainty. Thus, instead of assuring satisfactory performance at every single possible plant, the aim of controller design becomes one of maximizing the expected value of the performance of the controller. With this reformulation, there is reason to believe that the resulting designs will often be much less conservative than those based on deterministic uncertainty models.

A parallel theme has its beginnings in the early 1990s, and is the notion of the complexity of controller design. The tremendous advances in robust control synthesis theory in the 1980s led to very neat-looking problem formulations, based on very advanced concepts from functional analysis, in particular, the theory of Hardy spaces. As the research community began to apply these methods to large-sized practical problems, some researchers began to study the rate at which the computational complexity of robust control synthesis methods grew as a function of the

problem size. Somewhat to everyone's surprise, it was soon established that several problems of practical interest were in fact NP-hard. Thus, if one makes the reasonable assumption that P $\neq$ NP, then there do not exist polynomial-time algorithms for solving many reasonable-looking problems in robust control.

In the mainstream computer science literature, for the past several years researchers have been using the notion of randomization as a means of tackling difficult computational problems. Thus far there has not been any instance of a problem that is intractable using deterministic algorithms, but which becomes tractable when a randomized algorithm is used. However, there are several problems (for example, sorting) whose computational complexity reduces significantly when a randomized algorithm is used instead of a deterministic algorithm. When the idea of randomization is applied to control-theoretic problems, however, there appear to be some NP-hard problems that do indeed become tractable, provided one is willing to accept a somewhat diluted notion of what constitutes a "solution" to the problem at hand.

With all these streams of thought floating around the research community, it is an appropriate time for a book such as this. The central theme of the present work is the application of randomized algorithms to various problems in control system analysis and synthesis. The authors review practically all the important developments in robustness analysis and robust controller synthesis, and show how randomized algorithms can be used effectively in these problems. The treatment is completely self-contained, in that the relevant notions from elementary probability theory are introduced from first principles, and in addition, many advanced results from probability theory and from statistical learning theory are also presented. A unique feature of the book is that it provides a comprehensive treatment of the issue of sample generation. Many papers in this area simply assume that independent identically distributed (iid) samples generated according to a specific distribution are available, and do not bother themselves about the difficulty of generating these samples. The trade-off between the nonstandardness of the distribution and the difficulty of generating iid samples is clearly brought out here. If one wishes to apply randomization to practical problems, the issue of sample generation becomes very significant. At the same time, many of the results presented here on sample generation are not readily accessible to the control theory community. Thus the authors render a signal service to the research community by discussing the topic at the length they do. In addition to traditional problems in robust controller synthesis, the book also contains applications of the theory to network traffic analysis, and the stability of a flexible structure.

All in all, the present book is a very timely contribution to the literature. I have no hesitation in asserting that it will remain a widely cited reference work for many years.

Hyderabad, India M. Vidyasagar
June 2004

Preface to the Second Edition

Since the first edition of the book "Randomized Algorithms for Analysis and Control of Uncertain Systems" appeared in print in 2005, many new significant developments have been obtained in the area of probabilistic and randomized methods for control, in particular on the topics of sequential methods, the scenario approach and statistical learning techniques. Therefore, Chaps. 9, 10, 11, 12 and 13 have been rewritten to describe the most recent results and achievements in these areas.

Furthermore, in 2005 the development of randomized algorithms for systems and control applications was in its infancy. This area has now reached a mature stage and several new applications in very diverse areas within and outside engineering are described in Chap. 19, including the computation of PageRank in the Google search engine and control design of UAVs (unmanned aerial vehicles). The revised title of the book reflects this important addition. We believe that in the future many further applications will be successfully handled by means of probabilistic methods and randomized algorithms.

Torino, Italy
July 2012

Roberto Tempo
Giuseppe Calafiore
Fabrizio Dabbene

Acknowledgements

This book has been written with substantial help from many friends and colleagues. In particular, we are grateful to B. Ross Barmish, Yasumasa Fujisaki, Hideaki Ishii, Constantino Lagoa, Harald Niederreiter, Yasuaki Oishi, Carsten Scherer and Valery Ugrinovskii for suggesting several improvements on preliminary versions, as well as for pointing out various inaccuracies.

Some sections of this book have been utilized for a NATO lecture series delivered during spring 2008 at the University of Strathclyde, UK, University of Pamplona, Spain and Case Western Reserve University, Cleveland. In 2009, the book for used for teaching a Wintercourse DISC (Dutch Institute of Systems and Control) at Delft University of Technology and Technical University of Eindhoven, The Netherlands, and for a special topic graduate course in Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. In 2011, part of this book was taught as a graduate course at the Université Catholique de Louvain, Louvain la Neuve, Belgium. We warmly thank Tamer Başar, Michel Gevers, Paul Van den Hof and Paul Van Dooren for the invitations to teach at their respective institutions and for the exciting discussions.

We are pleased to thank the support of National Research Council (CNR) of Italy, and to acknowledge funding from HYCON2 Network of Excellence of the European Union Seventh Framework Programme, and from PRIN 2008 of Italian Ministry of Education, Universities and Research (MIUR).

Acknowledgments to the First Edition

This book has been written with substantial help from many friends and colleagues. In particular, we are grateful to B. Ross Barmish, Yasumasa Fujisaki, Constantino Lagoa, Harald Niederreiter, Yasuaki Oishi, Carsten Scherer and Valery Ugrinovskii for suggesting many improvements on preliminary versions, as well as for pointing out various inaccuracies and errors. We are also grateful to Tansu Alpcan and Hideaki Ishii for their careful reading of Sects. 19.4 and 19.6.

During the spring semester of the academic year 2002, part of this book was taught as a special-topic graduate course at CSL, University of Illinois at Urbana-Champaign, and during the fall semester of the same year at Politecnico di Milano, Italy. We warmly thank Tamer Başar and Patrizio Colaneri for the invitations to teach at their respective institutions and for the insightful discussions. Seminars on parts of this book were presented at the EECS Department, University of California at Berkeley, during the spring term 2003. We thank Laurent El Ghaoui for his invitation, as well as Elijah Polak and Pravin Varaiya for stimulating discussions. Some parts of this book have been utilized for a NATO lecture series delivered during spring 2003 in various countries, and in particular at Università di Bologna, Forlì, Italy, Escola Superior de Tecnologia de Setúbal, Portugal, and University of Southern California, Los Angeles. We thank Constantine Houpis for the direction and supervision of these events.

We are pleased to thank the National Research Council (CNR) of Italy for generously supporting for various years the research reported here, and to acknowledge funding from the Italian Ministry of Education, Universities and Research (MIUR) through an FIRB research grant.

Torino, Italy
June 2004

Roberto Tempo
Giuseppe Calafiore
Fabrizio Dabbene

Contents

Chapter 1
Overview

Don't assume the worst-case scenario. It's emotionally draining and probably won't happen anyway.

Anonymous

1.1 Probabilistic and Randomized Methods

The main objective of this book is to introduce the reader to the fundamentals of the area of probabilistic and randomized methods for analysis and design of *uncertain systems*. The take off point of this research is the observation that many quantities of interest in engineering, which are generally very difficult to compute exactly, can be easily approximated by means of randomization.

The presence of uncertainty in the system description has always been a critical issue in control theory and applications. The earliest attempts to deal with uncertainty were based on a *stochastic* approach, that led to great achievements in classical optimal control theory. In this theory, uncertainty is considered only in the form of exogenous disturbances having a stochastic characterization, while the plant dynamics are assumed to be exactly known. On the other hand, the *worst-case* setting, which has later emerged as a successful alternative to the previous paradigm, explicitly considers bounded uncertainty in the plant description. This setting is based on the "concern" that the uncertainty may be very malicious, and the idea is to guard against the worst-case scenario, even if it may be unlikely to occur. However, the fact that the worst-case setting may be too pessimistic, together with research results pointing out the computational hardness of this approach, motivated the need for further explorations towards new paradigms.

The contribution of this book is then in the direction of proposing a new paradigm for control analysis and design, based on a *rapprochement* between the classical stochastic approach and the modern worst-case approach. Indeed, in our setting we shall assume that the uncertainty is confined in a set (as in the worst-case approach) but, in addition to this information, we consider it as a random variable with given multivariate probability distribution. A typical example is a vector of uncertain parameters uniformly distributed inside a ball of fixed radius.

We address the interplay between stochastic (soft) and worst-case (hard) performance bounds for control system design in a rigorous fashion, with the goal to derive

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_1,

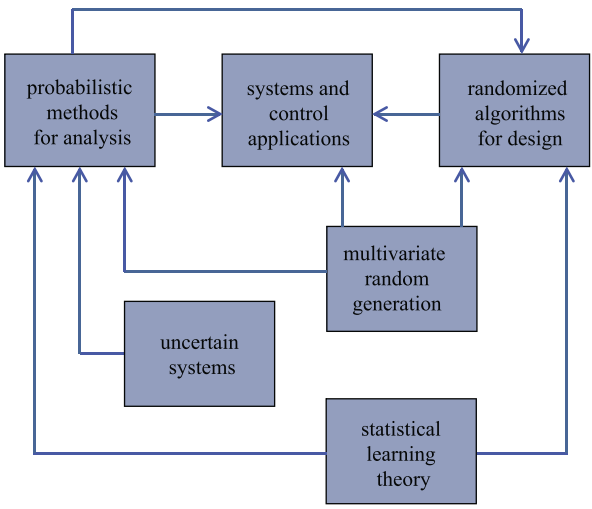

Fig. 1.1 Structure of the book

useful computational tools. The algorithms derived in this context are based on uncertainty randomization and are usually called *randomized algorithms*. These algorithms have been used successfully in, e.g., computer science, computational geometry and optimization. In these areas, several problems dealing with binary-valued functions have been efficiently solved using randomization, such as data structuring, search trees, graphs, agent coordination and Byzantine agreement problems. The derived algorithms are generally called *Las Vegas* randomized algorithms.

The randomized algorithms for control systems are necessarily of different type because we not only need to estimate some fixed quantity, but actually need to optimize over some design parameters (e.g., the controller's parameters), a context to which classical *Monte Carlo* methods cannot be directly applied. Therefore, a novel methodology is developed to derive technical tools which address convex and non-convex control design problems by means of sequential and non-sequential randomized algorithms. These tools are then successfully utilized to study several *systems and control applications*. We show that randomization is indeed a powerful tool in dealing with many interesting applications in various areas of research within and outside control engineering.

We now describe the structure of the book which can be roughly divided into six parts, see the block diagram shown in Fig. 1.1 which explains various interconnections between these parts.

1.2 Structure of the Book

Chapter 2 deals with basic elements of probability theory and introduces the notions of random variables and matrices used in the rest of the book. Classical univariate and multivariate densities are also listed.

- **Uncertain systems**
  Chapter 3: Uncertain Linear Systems
  Chapter 4: Linear Robust Control Design
  Chapter 5: Limits of the Robustness Paradigm

This first part of the book contains an introduction to robust control and discusses the limits of the worst-case paradigm. This part could be used for teaching a graduate course on the topic of uncertain systems, and it may be skipped by the reader familiar with these topics. Chapters 3 and 4 present a rather general and "dry" summary of the key results regarding robustness analysis and design. In Chap. 3, after introducing norms, balls and signals, the standard M–Δ model for describing linear time-invariant systems is studied. The small gain theorem (in various forms), μ theory and its connections with real parametric uncertainty, and the computation of robustness margins constitute the backbone of the chapter.

Chapter 4 deals with $\mathcal{H}_\infty$ and $\mathcal{H}_2$ design methods following a classical approach based on linear matrix inequalities. Special attention is devoted to linear quadratic Gaussian, linear quadratic regulator and guaranteed-cost control of uncertain systems.

In Chap. 5, the main limitations of classical robust control are outlined. First, a summary of concepts and results on computational complexity is presented and a number of NP-hard problems within systems and control are listed. Second, the issue of conservatism in the robustness margin computation is discussed. Third, a classical example regarding discontinuity of the robustness margin is revisited. This chapter provides a launching point for the probabilistic methods discussed next.

- **Probabilistic methods for analysis**
  Chapter 6: Probabilistic Methods for Uncertain Systems
  Chapter 7: Monte Carlo Methods

This part discusses probabilistic techniques for analysis of uncertain systems, Monte Carlo and quasi-Monte Carlo methods. In Chap. 6, the key ideas of probabilistic methods for systems and control are discussed. Basic concepts such as the so-called "good set" and "bad set" are introduced and three different problems, which are the probabilistic counterparts of standard robustness problems, are presented. This chapter also includes many specific examples showing that these problems can sometimes be solved in closed form without resorting to randomization.

The first part of Chap. 7 deals with Monte Carlo methods and provides a general overview of classical methods for both integration and optimization. The laws of large numbers for empirical mean, empirical probability and empirical maximum computation are reported. The second part of the chapter concentrates on quasi-Monte Carlo, which is a deterministic version of Monte Carlo methods. In this case, deterministic sequences for integration and optimization, together with specific error bounds, are discussed.

- **Statistical learning theory**
  Chapter 8: Probability Inequalities
  Chapter 9: Statistical Learning Theory

These two chapters address the crucial issue of finite-time convergence of randomized algorithms and in particular discuss probability inequalities, sample complexity and statistical learning theory. In the first part of Chap. 8, classical probability inequalities, such as Markov and Chebychev, are studied. Extensions to deviation inequalities are subsequently considered, deriving the Hoeffding inequality. These inequalities are then used to derive the sample complexity obtaining Chernoff and related bounds.

Chapter 9 deals with statistical learning theory. These results include the well-known Vapnik–Chervonenkis and Pollard results regarding uniform convergence of empirical means for binary and continuous-valued functions. We also discuss how these results may be exploited to derive the related sample complexity. The chapter includes useful bounds on the binomial distribution that may be used for computing the sample complexity.

- **Randomized algorithms for design**

  Chapter 10: Randomized Algorithms in Systems and Control
  Chapter 11: Sequential Algorithms for Probabilistic Design
  Chapter 12: Scenario Approach for Probabilistic Design
  Chapter 13: Learning-Based Control Design

In this part of the book, we move on to control design of uncertain systems with probabilistic techniques. Chapter 10 formally defines randomized algorithms of Monte Carlo and Las Vegas type. A clear distinction between analysis and synthesis is made. For analysis, we provide a connection with the Monte Carlo methods previously addressed in Chap. 7 and we state the algorithms for the solution of the probabilistic problems introduced in Chap. 6. For control synthesis, three different paradigms are discussed having the objective of studying feasibility and optimization for convex and nonconvex design problems. The chapter ends with a formal definition of efficient randomized algorithms.

The main point of Chap. 11 is the development of iterative stochastic algorithms under a convexity assumption in the design parameters. In particular, using the standard setting of linear matrix inequalities, we analyze sequential algorithms consisting of a probabilistic oracle and a deterministic update rule. Finite-time convergence results and the sample complexity of the probabilistic oracle are studied. Three update rules are analyzed: gradient iterations, ellipsoid method and cutting plane techniques. The differences with classical asymptotic methods studied in the stochastic approximation literature are also discussed.

Chapter 12 studies a non-sequential methodology for dealing with design in a probabilistic setting. In the scenario approach, the design problem is solved by means of a one-shot convex optimization involving a finite number of sampled uncertainty instances, named the scenarios. The results obtained include explicit formulae for the number of scenarios required by the randomized algorithm. The subsequent problem of "discarded constraints" is then analyzed and put in relation with chance-contrained optimization.

Chapter 13 addresses nonconvex optimization in the presence of uncertainty using a setting similar to the scenario approach, but in this case the objective is to

compute only a local solution of the optimization problem. For design with binary constraints given by Boolean functions, we compute the sample complexity, which provides the number of constraints entering into the optimization problem. Furthermore, we present a sequential algorithm for the solution of nonconvex semi-infinite feasibility and optimization problems. This algorithm is closely related to some results on statistical learning theory previously presented in Chap. 9.

- **Multivariate random generation**
  Chapter 14: Random Number and Variate Generation
  Chapter 15: Statistical Theory of Radial Random Vectors
  Chapter 16: Vector Randomization Methods
  Chapter 17: Statistical Theory of Radial Random Matrices
  Chapter 18: Matrix Randomization Methods

The main objective of this part of the book is the development of suitable sampling schemes for the different uncertainty structures analyzed in Chaps. 3 and 4. To this end, we study random number and variate generations, statistical theory of random vectors and matrices, and related algorithms. This requires the development of specific techniques for multivariate generation of independent and identically distributed vector and matrix samples within various sets of interest in control. These techniques are non-asymptotic (contrary to other methods based on Markov chains) and the idea is that the multivariate sample generation is based on simple algebraic transformations of a univariate random number generator.

Chapters 15 and 17 address statistical properties of random vectors and matrices respectively. They are quite technical, especially the latter, which is focused on random matrices. The reader interested in specific randomized algorithms for sampling within various norm-bounded sets may skip these chapters and concentrate instead on Chaps. 16 and 18.

Chapter 14 deals with the topic of random number and variate generation. This chapter begins with an overview of classical linear and nonlinear congruential methods and includes results regarding random variate transformations. Extensions to multivariate problems, as well as rejection methods and techniques based on the conditional density method, are also analyzed. Finally, a brief account of asymptotic techniques, including the so-called Markov chain Monte Carlo method, is given.

Chapter 15 is focused on statistical properties of radial random vectors. In particular, some general results for radially symmetric density functions are presented. Chapter 16 studies specific algorithms which make use of the theoretical results of the previous chapter for random sample generation within ℓ_p norm balls. In particular, efficient algorithms (which do not require rejection) based on the so-called generalized Gamma density are developed.

Chapter 17 is focused on the statistical properties of random matrices. Various norms are considered, but specific attention is devoted to the spectral norm, owing to its interest in control. In this chapter methods based on the singular value decomposition (SVD) of real and complex random matrices are studied. The key point is to compute the distributions of the SVD factors of a random matrix. This provides significant extensions of the results currently available in the theory of random matrices.

In Chap. 18 specific randomized algorithms for real and complex matrices are constructed by means of the conditional density method. One of the main points of this chapter is to develop algebraic tools for the closed-form computation of the marginal density, which is required in the application of this method.

- **Systems and control applications**
  Chapter 19: Applications of randomized algorithms

This chapter shows that randomized algorithms are indeed very useful tools in many areas of application. This chapter is divided into two parts. In the first part, we present a brief overview of some areas where randomized algorithms have been successfully utilized: systems biology, aerospace control, control of hard disk drives, high-speed networks, quantized, switched and hybrid systems, model predictive control, fault detection and isolation, embedded and electric circuits, structural design, linear parameter varying (LPV) systems, automotive and driver assistance systems. In the second part of this chapter, we study in more details a subset of the mentioned applications, including the computation of PageRank in the Google search engine and control design of unmanned aerial vehicles (UAVs). The chapter ends with a brief description of the Toolbox RACT (Randomized Algorithms Control Toolbox).

The Appendix includes some technical results regarding transformations between random matrices, Jacobians of transformations and the Selberg and Dyson–Mehta integrals.

Chapter 2
Elements of Probability Theory

In this chapter, we formally review some basic concepts of probability theory. Most of this material is standard and available in classical references, such as [108, 189, 319]; more advanced material on multivariate statistical analysis can be found in [22]. The definitions introduced here are instrumental to the study of randomized algorithms presented in subsequent chapters.

2.1 Probability, Random Variables and Random Matrices

2.1.1 Probability Space

Given a sample space Ω and a σ-algebra $\mathcal{S}$ of subsets S of Ω (the events), a probability $\text{PR}\{S\}$ is a real-valued function on $\mathcal{S}$ satisfying:

1. $\text{PR}\{S\} \in [0, 1]$;
2. $\text{PR}\{\Omega\} = 1$;
3. If the events S_i are mutually exclusive (i.e., $S_i \cap S_k = \emptyset$ for $i \neq k$), then

$$\text{PR}\left\{\bigcup_{i\in\mathcal{I}} S_i\right\} = \sum_{i\in\mathcal{I}} \text{PR}\{S_i\}$$

where $\mathcal{I}$ is a countable[1] set of positive integers.

The triple $(\Omega, \mathcal{S}, \text{PR}\{S\})$ is called a *probability space*.

A *discrete probability space* is a probability space where Ω is countable. In this case, $\mathcal{S}$ is given by subsets of Ω and the probability $\text{PR} : \Omega \to [0, 1]$ is such that

$$\sum_{\omega\in\Omega} \text{PR}\{w\} = 1.$$

[1] By countable we mean finite (possibly empty) or countably infinite.

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_2,

2.1.2 Real and Complex Random Variables

We denote with $\mathbb{R}$ and $\mathbb{C}$ the real and complex field respectively. The symbol $\mathbb{F}$ is also used to indicate either $\mathbb{R}$ or $\mathbb{C}$. A function $f : \Omega \to \mathbb{R}$ is said to be measurable with respect to a σ-algebra $\mathcal{S}$ of subsets of Ω if $f^{-1}(A) \in \mathcal{S}$ for every Borel set $A \subseteq \mathbb{R}$.

A *real random variable* $\mathbf{x}$ defined on a probability space $(\Omega, \mathcal{S}, \Pr\{S\})$ is a measurable function mapping Ω into $\mathcal{Y} \subseteq \mathbb{R}$, and this is indicated with the shorthand notation $\mathbf{x} \in \mathcal{Y}$. The set $\mathcal{Y}$ is called the range or *support* of the random variable $\mathbf{x}$. A *complex random variable* $\mathbf{x} \in \mathbb{C}$ is a sum $\mathbf{x} = \mathbf{x}_{\mathbb{R}} + j\mathbf{x}_{\mathbb{I}}$, where $\mathbf{x}_{\mathbb{R}} \in \mathbb{R}$ and $\mathbf{x}_{\mathbb{I}} \in \mathbb{R}$ are real random variables, and $j \doteq \sqrt{-1}$. If the random variable $\mathbf{x}$ maps the sample space Ω into a subset $[a, b] \subset \mathbb{R}$, we write $\mathbf{x} \in [a, b]$. If Ω is a discrete probability space, then $\mathbf{x}$ is a *discrete* random variable mapping Ω into a countable set.

Distribution and Density Functions The (cumulative) *distribution function* (cdf) of a random variable $\mathbf{x}$ is defined as

$$F_{\mathbf{x}}(x) \doteq \Pr\{\mathbf{x} \leq x\}.$$

The function $F_{\mathbf{x}}(x)$ is nondecreasing, right continuous (i.e., $F_{\mathbf{x}}(x) = \lim_{z \to x_+} F_{\mathbf{x}}(z)$), and $F_{\mathbf{x}}(x) \to 0$ for $x \to -\infty$, $F_{\mathbf{x}}(x) \to 1$ for $x \to \infty$. Associated with the concept of distribution function, we define the α *percentile* of a random variable

$$x_\alpha = \inf\big\{x : F_{\mathbf{x}}(x) \geq \alpha\big\}.$$

For random variables of continuous type, if there exists a Lebesgue measurable function $f_{\mathbf{x}}(x) \geq 0$ such that

$$F_{\mathbf{x}}(x) = \int_{-\infty}^{x} f_{\mathbf{x}}(x)\,\mathrm{d}x$$

then the cdf $F_{\mathbf{x}}(x)$ is said to be absolutely continuous, and

$$f_{\mathbf{x}}(x) = \frac{\mathrm{d}F_{\mathbf{x}}(x)}{\mathrm{d}x}$$

holds except possibly for a set of measure zero. The function $f_{\mathbf{x}}(x)$ is called the probability *density function* (pdf) of the random variable $\mathbf{x}$.

For discrete random variables, the cdf is a staircase function, i.e. $F_{\mathbf{x}}(x)$ is constant except at a countable number of points $x_1, x_2, \ldots$ having no finite limit point. The total probability is hence distributed among the "mass" points $x_1, x_2, \ldots$ at which the "jumps" of size

$$f_{\mathbf{x}}(x_i) \doteq \lim_{\epsilon \to 0} F_{\mathbf{x}}(x_i + \epsilon) - F_{\mathbf{x}}(x_i - \epsilon) = \Pr\{\mathbf{x} = x_i\}$$

occur. The function $f_{\mathbf{x}}(x_i)$ is called the *mass density* of the discrete random variable $\mathbf{x}$. The definition of random variables is extended to real and complex random matrices in the next section.

2.1.3 Real and Complex Random Matrices

Given n random variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$, their *joint distribution* is defined as

$$F_{\mathbf{x}_1,\ldots,\mathbf{x}_n}(x_1, \ldots, x_n) \doteq \text{PR}\{\mathbf{x}_1 \leq x_1, \ldots, \mathbf{x}_n \leq x_n\}.$$

When the above distribution is absolutely continuous, we can define the joint density function $f_{\mathbf{x}_1,\ldots,\mathbf{x}_n}(x_1, \ldots, x_n)$

$$f_{\mathbf{x}_1,\ldots,\mathbf{x}_n}(x_1, \ldots, x_n) \doteq \frac{\partial^n F_{\mathbf{x}_1,\ldots,\mathbf{x}_n}(x_1, \ldots, x_n)}{\partial x_1 \cdots \partial x_n}.$$

The random variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are said to be *independent* if

$$F_{\mathbf{x}_1,\ldots,\mathbf{x}_n}(x_1, \ldots, x_n) = \prod_{i=1}^{n} F_{\mathbf{x}_i}(x_i)$$

where $F_{\mathbf{x}_i}(x_i) = \text{PR}\{\mathbf{x}_i \leq x_i\}$.

A real random matrix $\mathbf{X} \in \mathbb{R}^{n,m}$ is a measurable function $\mathbf{X} : \Omega \to \mathcal{Y} \subseteq \mathbb{R}^{n,m}$. That is, the entries of $\mathbf{X}$ are real random variables $[\mathbf{X}]_{i,k}$ for $i = 1, \ldots, n$ and $k = 1, \ldots, m$. A complex random matrix $\mathbf{X} \in \mathbb{C}^{n,m}$ is defined as the sum $\mathbf{X} = \mathbf{X}_\mathbb{R} + j\mathbf{X}_\mathbb{I}$, where $\mathbf{X}_\mathbb{R}$ and $\mathbf{X}_\mathbb{I}$ are real random matrices. A random matrix is *discrete* if its entries are discrete random variables.

The distribution function $F_\mathbf{X}(X)$ of a real random matrix $\mathbf{X}$ is the joint cdf of the entries of $\mathbf{X}$. If $\mathbf{X}$ is a complex random matrix, then its cdf is the joint cdf of $\mathbf{X}_\mathbb{R}$ and $\mathbf{X}_\mathbb{I}$. The pdf $f_\mathbf{X}(X)$ of a real or complex random matrix is analogously defined as the joint pdf of the real and imaginary parts of its entries. The notation $\mathbf{X} \sim f_\mathbf{X}(X)$ means that $\mathbf{X}$ is a random matrix with probability density function $f_\mathbf{X}(X)$.

Let $\mathbf{X} \in \mathbb{F}^{n,m}$ be a real or complex random matrix (of continuous type) with pdf $f_\mathbf{X}(X)$ and support $\mathcal{Y} \subseteq \mathbb{F}^{n,m}$. Then, if $Y \subseteq \mathcal{Y}$, we have

$$\text{PR}\{\mathbf{X} \in Y\} = \int_Y f_\mathbf{X}(X)\, \mathrm{d}X.$$

Clearly, $\text{PR}\{\mathbf{X} \in \mathcal{Y}\} = \int_\mathcal{Y} f_\mathbf{X}(X)\, \mathrm{d}X = 1$. When needed, to further emphasize that the probability is relative to the random matrix $\mathbf{X}$, we explicitly write $\text{PR}_\mathbf{X}\{\mathbf{X} \in Y\}$.

2.1.4 Expected Value and Covariance

Let $\mathbf{X} \in \mathcal{Y} \subseteq \mathbb{F}^{n,m}$ be a random matrix and let $J : \mathbb{F}^{n,m} \to \mathbb{R}^{p,q}$ be a Lebesgue measurable function. The *expected value* of the random matrix $J(\mathbf{X})$ is defined as

$$\text{E}_\mathbf{X}\big(J(\mathbf{X})\big) \doteq \int_\mathcal{Y} J(X) f_\mathbf{X}(X)\, \mathrm{d}X$$

where $\mathcal{Y}$ is the support of $\mathbf{X}$. We make use of the symbol $\text{E}_\mathbf{X}(J(\mathbf{X}))$ to emphasize the fact that the expected value is taken with respect to $\mathbf{X}$. The suffix is omitted when clear from the context.

If $\mathbf{X} \in \mathbb{F}^{n,m}$ is a discrete random matrix with countable support $\mathcal{Y} = \{X_1, X_2, \ldots\}$, $X_i \in \mathbb{F}^{n,m}$ and $Y \subseteq \mathcal{Y}$, then

$$\Pr\{\mathbf{X} \in Y\} = \sum_{X_i \in Y} f_{\mathbf{X}}(X_i) = \sum_{X_i \in Y} \Pr\{\mathbf{X} = X_i\}.$$

The expected value of $J(\mathbf{X})$ is defined as

$$\mathrm{E}\big(J(\mathbf{X})\big) \doteq \sum_{X_i \in \mathcal{Y}} J(X_i) f_{\mathbf{X}}(X_i).$$

The expected value of $\mathbf{X} \in \mathbb{R}^{n,m}$ is usually called the *mean*. The *covariance matrix* of $\mathbf{x} \in \mathbb{R}^n$ is defined as

$$\mathrm{Cov}(\mathbf{x}) \doteq \mathrm{E}_{\mathbf{x}}\Big(\big(\mathbf{x} - \mathrm{E}_{\mathbf{x}}(\mathbf{x})\big)^T \big(\mathbf{x} - \mathrm{E}_{\mathbf{x}}(\mathbf{x})\big)\Big)$$

where X^T denotes the transpose of X. The covariance of $\mathbf{x} \in \mathbb{R}$ is called the *variance* and is given by

$$\mathrm{Var}(\mathbf{x}) \doteq \mathrm{E}_{\mathbf{x}}\Big(\big(\mathbf{x} - \mathrm{E}_{\mathbf{x}}(\mathbf{x})\big)^2\Big).$$

The square root of the variance $(\mathrm{Var}(\mathbf{x}))^{1/2}$ is called the *standard deviation*.

2.2 Marginal and Conditional Densities

Consider a random vector $\mathbf{x} = [\mathbf{x}_1 \cdots \mathbf{x}_n]^T \in \mathbb{R}^n$ with joint density function

$$f_{\mathbf{x}}(x) = f_{\mathbf{x}_1,\ldots,\mathbf{x}_n}(x_1, \ldots, x_n).$$

The *marginal density* of the first i components of the random vector $\mathbf{x} = [\mathbf{x}_1 \cdots \mathbf{x}_n]^T$ is defined as

$$f_{\mathbf{x}_1,\ldots,\mathbf{x}_i}(x_1, \ldots, x_i) \doteq \int \cdots \int f_{\mathbf{x}}(x_1, \ldots, x_n)\, \mathrm{d}x_{i+1} \cdots \mathrm{d}x_n. \tag{2.1}$$

The *conditional density* $f_{\mathbf{x}_i|x_1,\ldots,x_{i-1}}(x_i|x_1, \ldots, x_{i-1})$ of the random variable $\mathbf{x}_i$ conditioned to the event $\mathbf{x}_1 = x_1, \ldots, \mathbf{x}_{i-1} = x_{i-1}$ is given by the ratio of marginal densities

$$f_{\mathbf{x}_i|x_1,\ldots,x_{i-1}}(x_i|x_1, \ldots, x_{i-1}) \doteq \frac{f_{\mathbf{x}_1,\ldots,\mathbf{x}_i}(x_1, \ldots, x_i)}{f_{\mathbf{x}_1,\ldots,\mathbf{x}_{i-1}}(x_1, \ldots, x_{i-1})}. \tag{2.2}$$

2.3 Univariate and Multivariate Density Functions

We next present a list of classical univariate and multivariate density functions. The reader is referred to Chap. 14 for numerical methods for generating random variables with the mentioned densities.

Binomial Density The binomial density with parameters n, p is defined as

$$\mathrm{b}_{n,p}(x) \doteq \binom{n}{x} p^x (1-p)^{n-x}, \quad x \in \{0, 1, \ldots, n\} \tag{2.3}$$

where $\binom{n}{x}$ indicates the binomial coefficient $\binom{n}{x} = \frac{n!}{x!(n-x)!}$. The binomial distribution is denoted as

$$\mathrm{B}_{n,p}(x) \doteq \sum_{k=0}^{x} \binom{n}{k} p^k (1-p)^{n-k}, \quad x \in \{0, 1, \ldots, n\}. \tag{2.4}$$

Normal Density The normal (Gaussian) density with mean $\bar{x} \in \mathbb{R}$ and variance $\sigma^2 \in \mathbb{R}$ is defined as

$$\mathcal{N}_{\bar{x},\sigma^2}(x) \doteq \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}(x-\bar{x})^2/\sigma^2}, \quad x \in \mathbb{R}. \tag{2.5}$$

Multivariate Normal Density The multivariate normal density with mean $\bar{x} \in \mathbb{R}^n$ and symmetric positive definite covariance matrix $W \in \mathbb{S}^n$, $W \succ 0$, is defined as

$$\mathcal{N}_{\bar{x},W}(x) \doteq (2\pi)^{-n/2} |W|^{-1/2} \mathrm{e}^{-\frac{1}{2}(x-\bar{x})^T W^{-1}(x-\bar{x})}, \quad x \in \mathbb{R}^n. \tag{2.6}$$

Uniform Density The uniform density on the interval $[a, b]$ is defined as

$$\mathcal{U}_{[a,b]}(x) \doteq \begin{cases} \frac{1}{b-a} & \text{if } x \in [a, b]; \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

Uniform Density over a Set Let S be a Lebesgue measurable set of nonzero volume (see Sect. 3.1.3 for a precise definition of volume). The uniform density over S is defined as

$$\mathcal{U}_S(X) \doteq \begin{cases} \frac{1}{\mathrm{Vol}(S)} & \text{if } X \in S; \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

If instead S is a finite discrete set, i.e. it consists of a finite number of elements $S = \{X_1, X_2, \ldots, X_N\}$, then the uniform density over S is defined as

$$\mathcal{U}_S(X) \doteq \begin{cases} \frac{1}{\mathrm{Card}(S)} & \text{if } X \in S; \\ 0 & \text{otherwise} \end{cases}$$

where $\mathrm{Card}\,(S)$ is the cardinality of S.

Chi-Square Density The unilateral chi-square density with $n > 0$ degrees of freedom is defined as

$$\chi_n^2(x) \doteq \frac{1}{\Gamma(n/2)2^{n/2}} x^{n/2-1} \mathrm{e}^{-x/2}, \quad x \in \mathbb{R}_+ \tag{2.9}$$

where $\Gamma(\cdot)$ is the Gamma function

$$\Gamma(x) \doteq \int_0^\infty \xi^{x-1} \mathrm{e}^{-\xi}\, \mathrm{d}\xi, \quad x > 0.$$

Weibull Density The Weibull density with parameter $a > 0$ is defined as

$$W_a(x) \doteq a x^{a-1} \mathrm{e}^{-x^a}, \quad x \in \mathbb{R}. \tag{2.10}$$

Laplace Density The unilateral Laplace (or exponential) density with parameter $\lambda > 0$ is defined as

$$L_\lambda(x) \doteq \lambda \mathrm{e}^{-\lambda x}, \quad x \in \mathbb{R}_+. \tag{2.11}$$

Gamma Density The unilateral Gamma density with parameters $a > 0$, $b > 0$ is defined as

$$G_{a,b}(x) \doteq \frac{1}{\Gamma(a) b^a} x^{a-1} \mathrm{e}^{-x/b}, \quad x \in \mathbb{R}_+. \tag{2.12}$$

Generalized Gamma Density The unilateral generalized Gamma density with parameters $a > 0$, $c > 0$ is defined as

$$\overline{G}_{a,c}(x) \doteq \frac{c}{\Gamma(a)} x^{ca-1} \mathrm{e}^{-x^c}, \quad x \in \mathbb{R}_+. \tag{2.13}$$

2.4 Convergence of Random Variables

We now recall the formal definitions of convergence almost everywhere (or almost sure convergence), convergence in the mean square sense and convergence in probability. Other convergence concepts not discussed here include vague convergence, convergence of moments and convergence in distribution, see e.g. [108].

Definition 2.1 (Convergence almost everywhere) A sequence of random variables $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$ converges *almost everywhere* (a.e.) (or with probability one) to the random variable $\mathbf{x}$ if

$$\mathrm{Pr}\left\{ \lim_{N \to \infty} \mathbf{x}^{(N)} = \mathbf{x} \right\} = 1.$$

Definition 2.2 (Convergence in the mean square sense) A sequence of random variables $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$ converges *in the mean square sense* to the random variable $\mathbf{x}$ if

$$\lim_{N \to \infty} \mathrm{E}\left(\left| \mathbf{x} - \mathbf{x}^{(N)} \right|^2 \right) = 0.$$

Definition 2.3 (Convergence in probability) A sequence of random variables $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}, \ldots$ converges *in probability* to the random variable $\mathbf{x}$ if, for any $\epsilon > 0$, we have

$$\lim_{N \to \infty} \mathrm{Pr}\left\{ \left| \mathbf{x} - \mathbf{x}^{(N)} \right| > \epsilon \right\} = 0.$$

Convergence a.e. and convergence in the mean square sense both imply convergence in probability, while there is no implicative relationship between convergence a.e. and convergence in the mean square sense.

Chapter 3
Uncertain Linear Systems

This chapter presents a summary of some classical results regarding robustness analysis of linear systems. Synthesis problems are subsequently presented in Chap. 4. In these two chapters, we concentrate on linear, continuous and time-invariant systems and assume that the reader is familiar with the basics of linear algebra and systems and control theory, see e.g. [101, 335]. We do not attempt to provide a comprehensive treatment of robust control, which is discussed in depth for instance in [110, 121, 149, 184, 340, 357, 422]. Advanced material may be also found in the special issues [245, 338], and specific references are listed in [141].

3.1 Norms, Balls and Volumes

3.1.1 Vector Norms and Balls

Let $x \in \mathbb{F}^n$, where $\mathbb{F}$ is either the real or the complex field, then the ℓ_p norm of vector x is defined as

$$\|x\|_p \doteq \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}, \quad p \in [1, \infty) \tag{3.1}$$

and the ℓ_∞ norm of x is

$$\|x\|_\infty \doteq \max_i |x_i|.$$

The ℓ_2 norm is usually called the Euclidean norm. We define the ball of radius ρ in the ℓ_p norm as

$$\mathcal{B}_{\|\cdot\|_p}(\rho, \mathbb{F}^n) \doteq \{x \in \mathbb{F}^n : \|x\|_p \leq \rho\} \tag{3.2}$$

and its boundary as

$$\partial\mathcal{B}_{\|\cdot\|_p}(\rho, \mathbb{F}^n) \doteq \{x \in \mathbb{F}^n : \|x\|_p = \rho\}. \tag{3.3}$$

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_3,

When clear from the context, we simply write $\mathcal{B}_{\|\cdot\|_p}(\rho)$ and $\partial\mathcal{B}_{\|\cdot\|_p}(\rho)$ to denote $\mathcal{B}_{\|\cdot\|_p}(\rho, \mathbb{F}^n)$ and $\partial\mathcal{B}_{\|\cdot\|_p}(\rho, \mathbb{F}^n)$, respectively. Moreover, for balls of unit radius, we write $\mathcal{B}_{\|\cdot\|_p}(\mathbb{F}^n)$ and $\partial\mathcal{B}_{\|\cdot\|_p}(\mathbb{F}^n)$, or in brief as $\mathcal{B}_{\|\cdot\|_p}$ and $\partial\mathcal{B}_{\|\cdot\|_p}$.

We introduce further the weighted ℓ_2 norm of a real vector $x \in \mathbb{R}^n$. For a symmetric, positive definite matrix $W \succ 0$, the weighted ℓ_2 norm, denoted by ℓ_2^W, is defined as

$$\|x\|_2^W \doteq \left(x^T W^{-1} x\right)^{1/2}. \tag{3.4}$$

Clearly, if we compute the Cholesky decomposition $W^{-1} = R^T R$, then we have $\|x\|_2^W = \|Rx\|_2$. The ball of radius ρ in the ℓ_2^W norm is

$$\mathcal{B}_{\|\cdot\|_2^W}\left(\rho, \mathbb{R}^n\right) \doteq \left\{x \in \mathbb{R}^n : \|x\|_2^W \leq \rho\right\}. \tag{3.5}$$

This ball is an ellipsoid in the standard ℓ_2 metric. In fact, if we denote the ellipsoid of center $\bar{x}$ and shape matrix $W \succ 0$ as

$$\mathcal{E}(\bar{x}, W) \doteq \left\{x \in \mathbb{R}^n : (x - \bar{x})^T W^{-1} (x - \bar{x}) \leq 1\right\} \tag{3.6}$$

then $\mathcal{B}_{\|\cdot\|_2^W}(\rho, \mathbb{R}^n) = \mathcal{E}(0, \rho^2 W)$.

3.1.2 Matrix Norms and Balls

Two different classes of norms can be introduced when dealing with matrix variables: the so-called Hilbert–Schmidt norms, based on the isomorphism between the matrix space $\mathbb{F}^{n,m}$ and the vector space $\mathbb{F}^{nm}$, and the induced norms, where the matrix is viewed as an operator between vector spaces.

Hilbert–Schmidt Matrix Norms The (generalized) Hilbert–Schmidt ℓ_p norm of a matrix $X \in \mathbb{F}^{n,m}$ is defined as (see, e.g., [207])

$$\begin{aligned} \|X\|_p &\doteq \left(\sum_{i=1}^{n}\sum_{k=1}^{m} \left|[X]_{i,k}\right|^p\right)^{1/p}, \quad p \in [0, \infty); \\ \|X\|_\infty &\doteq \max_{i,k} \left|[X]_{i,k}\right| \end{aligned} \tag{3.7}$$

where $[X]_{i,k}$ is the (i, k) entry of matrix X. We remark that for $p = 2$ the Hilbert–Schmidt ℓ_p norm corresponds to the well-known Frobenius matrix norm

$$\|X\|_2 = \sqrt{\mathrm{Tr}\, X X^*}$$

where Tr denotes the trace and X^* is the conjugate transpose of X. Given a matrix $X \in \mathbb{F}^{n,m}$, we introduce the column vectorization operator

$$\mathrm{vec}(X) \doteq \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_m \end{bmatrix} \tag{3.8}$$

where $\xi_1, \ldots, \xi_m$ are the columns of X. Then, using (3.7) the Hilbert–Schmidt ℓ_p norm of X can be written as

$$\|X\|_p = \big\|\mathrm{vec}(X)\big\|_p.$$

In analogy to vectors, we denote the ℓ_p Hilbert–Schmidt norm ball in $\mathbb{F}^{n,m}$ of radius ρ as

$$\mathcal{B}_{\|\cdot\|_p}\big(\rho, \mathbb{F}^{n,m}\big) \doteq \big\{X \in \mathbb{F}^{n,m} : \|X\|_p \le \rho\big\}.$$

When clear from the context, we write $\mathcal{B}_{\|\cdot\|_p}(\rho)$ to denote $\mathcal{B}_{\|\cdot\|_p}(\rho, \mathbb{F}^{n,m})$ and $\mathcal{B}_{\|\cdot\|_p}(\mathbb{F}^{n,m})$ or $\mathcal{B}_{\|\cdot\|_p}$ for unit radius balls.

Induced Matrix Norms The ℓ_p induced norm of a matrix $X \in \mathbb{F}^{n,m}$ is defined as

$$|||X|||_p \doteq \max_{\|\xi\|_p=1} \|X\xi\|_p, \quad \xi \in \mathbb{F}^m. \tag{3.9}$$

The ℓ_1 induced norm of a matrix $X \in \mathbb{F}^{n,m}$ turns out to be the maximum of the ℓ_1 norms of its columns, that is

$$|||X|||_1 = \max_{i=1,\ldots,m} \|\xi_i\|_1 \tag{3.10}$$

where $\xi_1, \ldots, \xi_m$ are the columns of X. Similarly, the ℓ_∞ induced norm is equal to the maximum of the ℓ_1 norms of the rows of X, i.e.

$$|||X|||_\infty = \max_{i=1,\ldots,n} \|\eta_i\|_1$$

where $\eta_1^T, \ldots, \eta_n^T$ are the rows of X.

The ℓ_2 induced norm of a matrix is called the *spectral norm* and is related to the *singular value decomposition* (SVD), see for instance [207]. The SVD of a matrix $X \in \mathbb{F}^{n,m}$, $m \ge n$, is given by

$$X = U\Sigma V^*$$

where $\Sigma = \mathrm{diag}([\sigma_1 \ \cdots \ \sigma_n])$, with $\sigma_1 \ge \cdots \ge \sigma_n \ge 0$, $U \in \mathbb{F}^{n,n}$ is unitary, and $V \in \mathbb{F}^{m,n}$ has orthonormal columns.

The elements of Σ are called the *singular values* of X, and Σ is called the singular values matrix. The maximum singular value σ_1 of X is denoted by $\bar{\sigma}(X)$. The ℓ_2 induced norm of a matrix X is equal to

$$|||X|||_2 = \bar{\sigma}(X). \tag{3.11}$$

The ℓ_p induced norm ball of radius ρ in $\mathbb{F}^{n,m}$ is denoted by

$$\mathcal{B}_{|||\cdot|||_p}\big(\rho, \mathbb{F}^{n,m}\big) \doteq \big\{X \in \mathbb{F}^{n,m} : |||X|||_p \le \rho\big\}. \tag{3.12}$$

For simplicity, we denote a ball of radius ρ in the spectral norm as

$$\mathcal{B}_\sigma\big(\rho, \mathbb{F}^{n,m}\big) \doteq \mathcal{B}_{|||\cdot|||_2}\big(\rho, \mathbb{F}^{n,m}\big). \tag{3.13}$$

When clear from the context, we write $\mathcal{B}_{|||\cdot|||_p}(\rho)$ and $\mathcal{B}_\sigma(\rho)$ to denote the balls $\mathcal{B}_{|||\cdot|||_p}(\rho, \mathbb{F}^{n,m})$ and $\mathcal{B}_\sigma(\rho, \mathbb{F}^{n,m})$ respectively. Similarly, $\mathcal{B}_{|||\cdot|||_p}(\mathbb{F}^{n,m})$ or $\mathcal{B}_{\|\cdot\|_p}$, and $\mathcal{B}_\sigma(\mathbb{F}^{n,m})$ or $\mathcal{B}_\sigma$ denote unit radius balls.

3.1.3 Volumes

Consider the field $\mathbb{F}^{n,m}$. The dimension d of $\mathbb{F}^{n,m}$ is $d = nm$ if $\mathbb{F} \equiv \mathbb{R}$, and $d = 2nm$ if $\mathbb{F} \equiv \mathbb{C}$. Let $S \subset \mathbb{F}^{n,m}$ be a Lebesgue measurable set and let $\mu_d(\cdot)$ denote the d-dimensional Lebesgue measure, then the *volume* of S is defined as

$$\mathrm{Vol}(S) \doteq \int_S \mathrm{d}\mu_d(X). \tag{3.14}$$

Similarly, we indicate by $\mathrm{Surf}(S)$ the *surface* of S, that is the $(d-1)$-dimensional Lebesgue measure of S. In particular, for norm balls, volume and surface measures depend on the ball radius according to the relations (see for instance [39])

$$\begin{aligned}\mathrm{Vol}\big(\mathcal{B}_{\|\cdot\|_p}(\rho)\big) &= \mathrm{Vol}(\mathcal{B}_{\|\cdot\|_p})\rho^d;\\ \mathrm{Surf}\big(\mathcal{B}_{\|\cdot\|_p}(\rho)\big) &= \mathrm{Surf}(\mathcal{B}_{\|\cdot\|_p})\rho^{d-1}.\end{aligned}$$

3.2 Signals

In this section, we briefly introduce some concepts related to signals, see e.g. [247] for a comprehensive treatment of this topic.

3.2.1 Deterministic Signals

A *deterministic signal* $v(t) : \mathbb{R} \to \mathbb{R}^n$ is a Lebesgue measurable function of the time variable $t \in \mathbb{R}$. The set

$$\mathcal{V}^+ = \big\{v(t) \in \mathbb{R}^n : v \text{ is Lebesgue measurable, } v(t) = 0 \text{ for all } t < 0\big\}$$

is the linear space of *causal* signals. For $p \in [1, \infty)$, the infinite-horizon $\mathcal{L}_p^+$ space is defined as the space of signals $v \in \mathcal{V}^+$ such that the integral

$$\left(\int_0^\infty \big\|v(t)\big\|_p^p \,\mathrm{d}t\right)^{1/p} \tag{3.15}$$

exists and is bounded. In this case, (3.15) defines a signal norm, which is denoted by $\|v\|_p$. For $p = \infty$, we have $\|v\|_\infty \doteq \operatorname{ess\,sup}_t v(t)$.

For the important special case $p = 2$, $\mathcal{L}_2^+$ is a Hilbert space, equipped with the standard inner product

$$\langle x, y\rangle = \int_0^\infty y^T(t)x(t)\,\mathrm{d}t$$

where $x, y \in \mathcal{L}_2^+$. Signals in $\mathcal{L}_2^+$ are therefore causal signals with finite total energy. These are typically *transient* signals which decay to zero as $t \to \infty$.

We now discuss some fundamental results related to the Laplace transform of signals in $\mathcal{L}_2^+$. The $\mathcal{H}_2^n$ space (see Definition 3.3) is the space of functions of complex variable $g(s) : \mathbb{C} \to \mathbb{C}^n$ which are analytic[1] in $\mathrm{Re}(s) > 0$ and for which the integral

$$\left(\frac{1}{2\pi} \int_{-\infty}^{\infty} g^*(j\omega) g(j\omega) \, \mathrm{d}\omega \right)^{1/2} \tag{3.16}$$

exists and is bounded. In this case, (3.16) defines a norm denoted by $\|g\|_2$. Define further the unilateral Laplace transform of the signal $v \in \mathcal{V}^+$ as

$$\zeta(s) = \mathfrak{L}(v) \doteq \int_0^{\infty} v(t) \mathrm{e}^{-st} \, \mathrm{d}t$$

and the inverse Laplace transform

$$v(t) = \mathfrak{L}^{-1}(\zeta) \doteq \lim_{\omega \to \infty} \frac{1}{2\pi j} \int_{c-j\omega}^{c+j\omega} \zeta(s) \mathrm{e}^{st} \, \mathrm{d}s.$$

Then, if $v \in \mathcal{L}_2^+$, its Laplace transform is in $\mathcal{H}_2^n$. Conversely, by the Paley–Wiener theorem, see e.g. [149], for any $\zeta \in \mathcal{H}_2^n$ there exists a causal signal $v \in \mathcal{L}_2^+$ such that $\zeta = \mathfrak{L}(v)$. Notice also that $\mathcal{H}_2^n$ is a Hilbert space, equipped with the inner product

$$\langle g, h \rangle = \frac{1}{2\pi} \int_{-\infty}^{\infty} g^*(j\omega) h(j\omega) \, \mathrm{d}\omega$$

for $g, h \in \mathcal{H}_2^n$. Finally, we recall the Parseval identity, see e.g. [184], which relates the inner product of the signals $v, w \in \mathcal{L}_2^+$ to the inner product of their Laplace transforms

$$\langle v, w \rangle = \big\langle \mathfrak{L}(v), \mathfrak{L}(w) \big\rangle.$$

3.2.2 Stochastic Signals

The performance specifications of control systems are sometimes expressed in terms of *stochastic*, rather than deterministic, signals. In this section, we summarize some basic definitions related to stochastic signals. For formal definitions of random variables and matrices and their statistics, the reader can refer to Chap. 2 and to [138, 319] for further details on stochastic processes.

Denote with $\mathbf{v}(t)$ a zero-mean, stationary stochastic process. The *autocorrelation* of $\mathbf{v}(t)$ is defined as

$$\mathrm{R}_{\mathbf{v},\mathbf{v}}(\tau) \doteq \mathrm{E}_{\mathbf{v}} \big(\mathbf{v}(t) \mathbf{v}^T(t+\tau) \big)$$

[1] Let $S \subset \mathbb{C}$ be an open set. A function $f : S \to \mathbb{C}$ is said to be *analytic* at a point $s_0 \in S$ if it is differentiable for all points in some neighborhood of s_0. The function is analytic in S if it is analytic for all $s \in S$. A matrix-valued function is analytic if every element of the matrix is analytic.

where $\mathrm{E}_{\mathbf{v}}(\cdot)$ denotes the expectation with respect to the stochastic process. The power spectral density (psd) $\Phi_{\mathbf{v},\mathbf{v}}(\omega)$ of $\mathbf{v}$ is defined as the Fourier transform of $\mathrm{R}_{\mathbf{v},\mathbf{v}}(\tau)$. A frequently used measure of a stationary stochastic signal is its root-mean-square (rms) value

$$\|\mathbf{v}\|_{\mathrm{rms}}^2 = \mathrm{E}_{\mathbf{v}}\big(\mathbf{v}^T(t)\mathbf{v}(t)\big) = \mathrm{Tr}\,\mathrm{R}_{\mathbf{v},\mathbf{v}}(0).$$

The rms value measures the *average power* of the stochastic signal, and it is a steady-state measure of the behavior of the signal, i.e. it is not affected by transients. By the Parseval identity, the average power can alternatively be computed as an integral over frequency of the power spectral density

$$\|\mathbf{v}\|_{\mathrm{rms}}^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{Tr}\,\Phi_{\mathbf{v},\mathbf{v}}(\omega)\,\mathrm{d}\omega.$$

If the process $\mathbf{v}(t)$ is *ergodic*, then its moments can be equivalently computed as time-domain averages of a single realization $v(t)$ of the process. With probability one, the rms norm is given by

$$\|\mathbf{v}\|_{\mathrm{rms}}^2 = \lim_{T\to\infty}\frac{1}{T}\int_0^T v^T(t)v(t)\,\mathrm{d}t.$$

3.3 Linear Time-Invariant Systems

Consider a linear time-invariant (LTI), proper system described in standard state space form

$$\begin{aligned} \dot{x} &= Ax + Bw; \\ z &= Cx + Dw \end{aligned} \tag{3.17}$$

where $A \in \mathbb{R}^{n_s,n_s}$, $B \in \mathbb{R}^{n_s,q}$, $C \in \mathbb{R}^{p,n_s}$, $D \in \mathbb{R}^{p,q}$. This state space system is *stable*, or *Hurwitz*, if $\mathrm{Re}(\lambda_i(A)) < 0$, $i = 1,\ldots,n_s$, where $\lambda_i(A)$ denote the eigenvalues of A.

Assuming $x(0) = 0$, system (3.17) defines a proper linear operator $\mathcal{G}$ mapping the input signal space into the output signal space. In the space of Laplace transforms, the operator $\mathcal{G}$ is represented by the *transfer-function matrix*, or simply *transfer matrix*

$$G(s) = C(sI - A)^{-1}B + D.$$

The system (3.17) is indicated compactly by means of the matrix quadruple

$$\Omega_G \doteq \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right].$$

The operator $\mathcal{G}$ related to system (3.17) is *stable* if and only if it maps $\mathcal{L}_2^+$ into $\mathcal{L}_2^+$. A necessary and sufficient stability condition for $\mathcal{G}$ is that its transfer matrix $G(s)$ has all its poles in the open left-half plane.

Definition 3.1 ($\mathcal{RH}_\infty$ space) The space $\mathcal{RH}_\infty^{p,q}$ is defined as the space of proper, rational functions with real coefficients $G : \mathbb{C} \to \mathbb{C}^{p,q}$ that are analytic in the open right-half plane.

From this definition, it follows that the operator $\mathcal{G}$ is stable if and only if its transfer matrix $G(s)$ belongs to $\mathcal{RH}_\infty$.

Assuming $\mathcal{G}$ stable, since $\mathcal{G}$ maps $\mathcal{L}_2^+$ into $\mathcal{L}_2^+$, it is natural to define its $\mathcal{L}_2^+$-gain as

$$\|\mathcal{G}\|_{\mathcal{L}_2^+ \to \mathcal{L}_2^+} \doteq \sup_{0 \neq w \in \mathcal{L}_2^+} \frac{\|\mathcal{G}w\|_2}{\|w\|_2}.$$

If $\mathcal{G}$ is represented in the frequency domain by the transfer matrix $G(s)$, then it can be shown that its $\mathcal{L}_2^+$-gain coincides with the so-called $\mathcal{H}_\infty$ norm of $G(s)$, defined as

$$\big\|G(s)\big\|_\infty \doteq \operatorname*{ess\,sup}_{\omega \in \mathbb{R}} \bar{\sigma}\big(G(j\omega)\big) \tag{3.18}$$

where $\bar{\sigma}(G(j\omega))$ denotes the largest singular value of $G(j\omega)$, i.e.

$$\big\|G(s)\big\|_\infty = \|\mathcal{G}\|_{\mathcal{L}_2^+ \to \mathcal{L}_2^+}. \tag{3.19}$$

In a more general setting, one can define the space of functions $G : \mathbb{C} \to \mathbb{C}^{p,q}$ (not necessarily rational), for which the norm (3.18) is bounded.

Definition 3.2 ($\mathcal{H}_\infty$ space) The space $\mathcal{H}_\infty^{p,q}$ is defined as the space of functions $G : \mathbb{C} \to \mathbb{C}^{p,q}$ that are analytic and bounded in the open right-half plane.

From this definition it follows immediately that $\mathcal{RH}_\infty \subset \mathcal{H}_\infty$.

Remark 3.1 ($\mathcal{H}_\infty$ norm interpretations) The $\mathcal{H}_\infty$ norm of a stable system may be interpreted from (3.19) as the maximum energy gain of the system. In the case of stochastic signals, it has an alternative interpretation as the rms gain of the system, i.e. it denotes the maximum average power amplification from input to output. We also remark that the $\mathcal{H}_\infty$ norm is *submultiplicative*, i.e.

$$\|GH\|_\infty \leq \|G\|_\infty \|H\|_\infty.$$

For stable single-input single-output (SISO) systems, (3.18) indicates that the value of the $\mathcal{H}_\infty$ norm coincides with the peak of the magnitude of the Bode plot of the transfer function of the system.

Another frequently used measure of a system "gain" is the $\mathcal{H}_2$ norm. This norm and the corresponding linear space of transfer matrices are now defined.

Definition 3.3 ($\mathcal{H}_2$ and $\mathcal{RH}_2$ spaces) The space $\mathcal{H}_2^{p,q}$ is defined as the space of functions $G : \mathbb{C} \to \mathbb{C}^{p,q}$ that are analytic in the open right-half plane and such that the integral

$$\left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \operatorname{Tr} G^*(j\omega) G(j\omega)\, \mathrm{d}\omega\right)^{1/2} \tag{3.20}$$

exists and is bounded. In this case, (3.20) defines the $\mathcal{H}_2$ norm of G, which is denoted by $\|G\|_2$. The space $\mathcal{RH}_2^{p,q}$ is then defined as

$$\mathcal{RH}_2^{p,q} \doteq \left\{ G \in \mathcal{H}_2^{p,q} : G \text{ is real rational} \right\}.$$

Notice that, according to the above definition, a rational transfer matrix $G(s)$ belongs to $\mathcal{RH}_2$ if and only if it is stable and strictly proper.

Remark 3.2 ($\mathcal{H}_2$ norm interpretations) The $\mathcal{H}_2$ norm of a stable system has two interpretations. First, we notice that $\|G(s)\|_2^2$ can be computed in the time domain using the Parseval identity

$$\|G\|_2^2 = \int_0^\infty \operatorname{Tr} g^T(t) g(t)\, \mathrm{d}t$$

where $g(t) = \mathfrak{L}^{-1}(G(s))$ is the impulse response matrix. The $\mathcal{H}_2$ norm can hence be interpreted as the energy of the impulse response of the system.

Secondly, the $\mathcal{H}_2$ norm can be viewed as a measure of the average power of the steady-state output, when the system is driven by white noise input, see for instance [67]. In fact, when a stochastic signal $\mathbf{w}$ with power spectral density $\Phi_{\mathbf{w},\mathbf{w}}(\omega)$ enters a stable and strictly proper system with transfer matrix G, then the output $\mathbf{z}$ has spectral density given by

$$\Phi_{\mathbf{z},\mathbf{z}}(\omega) = G(j\omega)\Phi_{\mathbf{w},\mathbf{w}}(\omega)G^*(j\omega)$$

and the average output power is $\|\mathbf{z}\|_{\text{rms}}$. When $\mathbf{w}$ is white noise, then $\Phi_{\mathbf{w},\mathbf{w}}(\omega) = I$, and $\|\mathbf{z}\|_{\text{rms}} = \|G\|_2$.

3.4 Linear Matrix Inequalities

Many of the analysis and design specifications for control systems may be expressed in the form of satisfaction of a positive (or negative) definiteness condition for a matrix function which depends affinely on the decision variables of the problem. Such matrix "inequalities" are commonly known under the name of linear matrix inequalities (LMIs), and are now briefly defined.

Let $x \in \mathbb{R}^m$ be a vector of decision variables. An LMI condition on x is a matrix inequality of the form

$$F(x) \succ 0 \tag{3.21}$$

with

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \tag{3.22}$$

and where $F_i \in \mathbb{S}^n$, $i = 0, 1, \ldots, m$ are given symmetric matrices. Inequality (3.21) is called a *strict* matrix inequality, because strict positive definiteness is required

by the condition. Nonstrict LMIs are defined analogously, by requiring only positive semidefiniteness of matrix $F(x)$, and are indicated with the notation $F(x) \succeq 0$. The *feasible set* of the LMI (3.21) is defined as the set of x that satisfy the matrix inequality

$$\mathcal{X} = \{x \in \mathbb{R}^m : F(x) \succ 0\}.$$

The most notable feature of LMIs is that the feasible set $\mathcal{X} \in \mathbb{R}^m$ is a *convex set*, meaning that for all $x_1, x_2 \in \mathcal{X}$ and all $\lambda \in [0, 1]$ it holds that

$$\lambda x_1 + (1 - \lambda)x_2 \in \mathcal{X}.$$

This fact can be easily understood by noticing that the condition $F(x) \succ 0$ is equivalent to the condition

$$\xi^T F(x)\xi > 0, \quad \text{for all non-zero } \xi \in \mathbb{R}^n.$$

Indeed, for any given non-zero $\xi \in \mathbb{R}^n$, the set $\{x : \xi^T F(x)\xi > 0\}$ is an open half-space, hence a convex set, and $\mathcal{X}$ is the (infinite) intersection of such half-spaces. LMI conditions are often used as constraints in optimization problems. In particular, mathematical programs having linear objective and an LMI constraint

$$\min_{x \in \mathbb{R}^m} c^T x \quad \text{subject to} \quad F(x) \succ 0$$

are known as semidefinite programs (SDPs), see e.g. [385, 400]. Clearly, SDPs are convex optimization problems, and encompass linear, as well as convex quadratic and conic programs.

The representation of control analysis and design problems by means of SDPs has had enormous success in recent years, owing to the availability of efficient numerical algorithms (interior point algorithms in particular, see [299]) for the solution of SDPs. We refer the reader to [68] for an introduction to LMIs and SDPs in systems and control. The LMI representation for control problems is extensively used in subsequent chapters.

Finally, we remark that in applications we often encounter LMIs where the decision variables are in matrix rather than in vector form as in the standard representation of (3.21) and (3.22). The first and most notable example is the Lyapunov inequality

$$AX + XA^T \prec 0 \tag{3.23}$$

where $A \in \mathbb{R}^{n,n}$ is a given matrix, and $X \in \mathbb{S}^n$ is the decision matrix. Such LMIs in matrix variables can, however, be converted in the standard form (3.22) by introducing a vector x containing the free variables of X and exploiting the linearity of the representation. For example, the LMI (3.23) is rewritten in standard form by first introducing vector $x \in \mathbb{R}^m$, $m = n(n-1)/2$, containing the free elements of the symmetric matrix X. Then, one writes $X = \sum_{i=1}^m x_i S_i$, where $S_i \in \mathbb{R}^{n,n}$ represents an element of the standard basis of symmetric matrices, and therefore (3.23) takes the standard form

$$F_0 + \sum_{i=1}^m x_i F_i \succ 0$$

with $F_0 = 0_{n,n}$, $F_i = -(AS_i + S_i A^T)$, $i = 1, \ldots, m$.

3.5 Computing $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Norms

Let $G(s) = C(sI - A)^{-1}B \in \mathcal{RH}_2^{p,q}$ be a strictly proper transfer matrix, and assume that A is stable. Then, we have

$$\|G\|_2^2 = \operatorname{Tr} C W_c C^T$$

where W_c is the controllability Gramian of the system. The controllability Gramian is positive semidefinite, $W_c \succeq 0$, and it is the unique solution of the Lyapunov equation

$$AW_c + W_c A^T + BB^T = 0.$$

Equivalently, in the dual formulation we obtain

$$\|G\|_2^2 = \operatorname{Tr} B^T W_o B$$

where the observability Gramian $W_o \succeq 0$ is the unique solution of the Lyapunov equation

$$A^T W_o + W_o A + C^T C = 0.$$

For the monotonicity property of the Lyapunov equation, we can also express the $\mathcal{H}_2$ norm in terms of a Lyapunov *inequality*. This characterization in terms of LMIs is stated in the next lemma, see for instance [346].

Lemma 3.1 ($\mathcal{H}_2$ norm characterization) *Let $G(s) = C(sI - A)^{-1}B + D$ and $\gamma > 0$. The following three statements are equivalent*:

1. *A is stable and $\|G(s)\|_2^2 < \gamma$;*
2. *$D = 0$, and there exist $S \succ 0$ such that*

$$\begin{aligned} AS + SA^T + BB^T &\prec 0; \\ \operatorname{Tr} CSC^T &< \gamma; \end{aligned}$$

3. *$D = 0$, and there exist $P \succ 0$ and $Q \succ 0$ such that*

$$\begin{aligned} \begin{bmatrix} PA + A^T P & PB \\ B^T P & -I \end{bmatrix} &\prec 0; \\ \begin{bmatrix} P & C^T \\ C & Q \end{bmatrix} &\succ 0; \\ \operatorname{Tr} Q &< \gamma. \end{aligned}$$

The next well-known lemma, often denoted as *bounded real lemma*, gives a characterization of the $\mathcal{H}_\infty$ norm of a system.

Lemma 3.2 (Bounded real lemma) *Let $G(s) = C(sI - A)^{-1}B + D$ and $\gamma > 0$. The following two statements are equivalent*:

1. *A is stable and $\|G(s)\|_\infty < \gamma$;*

2. *There exist* $P \succ 0$ *such that*

$$\begin{bmatrix} PA + A^T P & PB & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} \prec 0. \tag{3.24}$$

A detailed proof of the bounded real lemma in this form may be found in [344]. There is also a nonstrict characterization of the $\mathcal{H}_\infty$ norm, given in the next lemma, see [344].

Lemma 3.3 (Nonstrict bounded real lemma) *Let* $G(s) = C(sI - A)^{-1}B + D$, *with* A *stable and* (A, B) *controllable*,[2] *and let* $\gamma \geq 0$. *The following two statements are equivalent*:

1. $\|G(s)\|_\infty \leq \gamma$;
2. *There exist* $P = P^T$ *such that*

$$\begin{bmatrix} PA + A^T P & PB & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} \preceq 0.$$

From the computational point of view, checking whether the $\mathcal{H}_\infty$ norm is less than γ amounts to solving Eq. (3.24) with respect to P, which is a convex feasibility problem with LMI constraints.

3.6 Modeling Uncertainty of Linear Systems

In this section, we present a general model that is adopted to represent various sources of uncertainty that may affect a dynamic system. In particular, we follow a standard approach based on the so-called M–Δ model, which is frequently used in modern control theory, see e.g. [422], for a systematic discussion on this topic.

In Fig. 3.1, $M \in \mathcal{RH}_\infty^{c,r}$ represents the transfer matrix of the known part of the system, which consists of the extended plant and the controller. In this description, $\Delta \in \mathcal{RH}_\infty^{r_\Delta, c_\Delta}$ encompasses all time-invariant uncertainties acting on the system. This uncertainty is assumed to belong to a block-diagonal *structured set* $\widetilde{\mathbb{D}}$ of the form

$$\widetilde{\mathbb{D}} \doteq \left\{ \Delta \in \mathcal{RH}_\infty^{r_\Delta, c_\Delta} : \Delta = \text{bdiag}(q_1 I_{m_1}, \ldots, q_\ell I_{m_\ell}, \Delta_1, \ldots, \Delta_b) \right\} \tag{3.25}$$

where $q \doteq [q_1 \cdots q_\ell]^T$ represents (real or complex) uncertain parameters q_i, with multiplicity m_i, $i = 1, \ldots, \ell$, and Δ_i, $i = 1, \ldots, b$, denote general full-block stable

[2] (A, B) is controllable if and only if the reachability matrix $R = [B\ AB\ A^2B \cdots A^{n_s-1}B]$ is full rank.

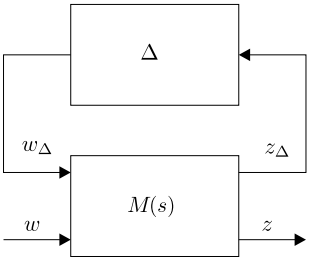

Fig. 3.1 M–Δ model. $M(s)$ is the known part of the system, consisting of the interconnection of plant and controller, Δ represents the uncertain part, w includes noise, disturbances and reference signals and z represents controlled signals and tracking errors

and proper transfer matrices of size $r_i \times c_i$. Moreover, a bound ρ on the magnitude of the uncertainty is imposed. In particular, we assume that $\Delta \in \mathcal{B}_{\widetilde{\mathbb{D}}}(\rho)$, where

$$\mathcal{B}_{\widetilde{\mathbb{D}}} = \mathcal{B}_{\widetilde{\mathbb{D}}}(\rho) \doteq \left\{ \Delta \in \widetilde{\mathbb{D}} : \|q\|_p \le \rho,\ \|\Delta_i\|_\infty \le \rho,\ i = 1, \dots, b \right\} \tag{3.26}$$

and $\|\cdot\|_p$ is an ℓ_p vector norm. Note that, when clear from the context, we drop the dependency on ρ and we simply write $\mathcal{B}_{\widetilde{\mathbb{D}}}$.

Following classical literature on the subject, one can associate to $\widetilde{\mathbb{D}}$ a corresponding *matrix* structure

$$\mathbb{D} \doteq \left\{ \Delta \in \mathbb{F}^{r_\Delta, c_\Delta} : \Delta = \mathrm{bdiag}(q_1 I_{m_1}, \dots, q_\ell I_{m_\ell}, \Delta_1, \dots, \Delta_b) \right\}, \tag{3.27}$$

where Δ_i are (real or complex) matrices. We remark that if q_i in $\widetilde{\mathbb{D}}$ is real (complex), then the corresponding parameter in $\mathbb{D}$ is also real (complex). Similarly, if a full block Δ_i of size $r_i \times c_i$ in $\widetilde{\mathbb{D}}$ is a static real (complex) matrix gain, then the corresponding block in $\mathbb{D}$ is also a real (complex) matrix of size $r_i \times c_i$. If instead a full block $\Delta_i \in \mathcal{RH}_\infty^{r_i, c_i}$ in $\widetilde{\mathbb{D}}$ is a dynamic operator, then the corresponding block in $\mathbb{D}$ is a static complex block of size $r_i \times c_i$. The related norm-bounded set is defined as

$$\mathcal{B}_{\mathbb{D}} = \mathcal{B}_{\mathbb{D}}(\rho) \doteq \left\{ \Delta \in \mathbb{D} : \|q\|_p \le \rho,\ \bar{\sigma}(\Delta_i) \le \rho,\ i = 1, \dots, b \right\}. \tag{3.28}$$

We remark that if no dynamic block appears in (3.25), then the sets $\mathcal{B}_{\widetilde{\mathbb{D}}}$ and $\mathcal{B}_{\mathbb{D}}$ coincide. In the more general situation of dynamic blocks, $\mathcal{B}_{\mathbb{D}}$ may be viewed as a "snapshot" of $\mathcal{B}_{\widetilde{\mathbb{D}}}$ at a fixed frequency.

The signal w in Fig. 3.1 usually represents disturbances of various nature entering the system, such as white (or colored) noise or a deterministic, norm-bounded signal, and z describes errors or other quantities that should be kept small in some sense. The transfer matrix $M(s)$ is partitioned as

$$\begin{bmatrix} z_\Delta(s) \\ z(s) \end{bmatrix} = \begin{bmatrix} M_{11}(s) & M_{12}(s) \\ M_{21}(s) & M_{22}(s) \end{bmatrix} \begin{bmatrix} w_\Delta(s) \\ w(s) \end{bmatrix}$$

so that the transfer matrix of the performance channel $w \to z$ can be expressed in terms of the *upper* linear fractional transformation (LFT)

$$\mathcal{F}_u(M, \Delta) \doteq M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1} M_{12}. \tag{3.29}$$

This LFT is well defined whenever the matrix M_{11} satisfies a *well-posedness* condition, i.e. $(I - M_{11}(\infty)\Delta(\infty))$ is nonsingular for all $\Delta \in \mathcal{B}_{\widetilde{\mathbb{D}}}$. Moreover, the condition $(I - M_{11}\Delta)^{-1} \in \mathcal{RH}_\infty^{c_\Delta, c_\Delta}$ guarantees that the interconnection between $M \in \mathcal{RH}_\infty^{c,r}$ and $\Delta \in \mathcal{RH}_\infty^{r_\Delta, c_\Delta}$ is *internally stable*, see [422].

Two key requirements are typically imposed on the interconnection in Fig. 3.1: (i) the interconnection is well posed and internally stable, for all $\Delta \in \mathcal{B}_{\widetilde{\mathbb{D}}}$; (ii) the influence of the disturbances w on the controlled outputs z is below a desired level for all uncertainties $\Delta \in \mathcal{B}_{\widetilde{\mathbb{D}}}$. In particular, condition (i) is a *robust stability* condition (see details in Sect. 3.7), and condition (ii) typically expresses performance requirements imposed as bounds on the gain of the $w \to z$ channel.

Remark 3.3 (LFT representation lemma) Notice that the LFT representation in (3.29) is general enough to encompass uncertainty entering the transfer matrix in a generic polynomial or rational manner, provided that the transfer matrix has no singularities at zero. This result is known as the LFT representation lemma, see for instance [158]. In [422], constructive rules for building the LFT representation starting from basic algebraic operations on LFTs are given.

In the following, we present some examples, involving different uncertainty configurations, and show how we can express them in the M–Δ framework.

Example 3.1 (Real unstructured uncertainty) Consider a linear time-invariant system expressed in state space form

$$\begin{aligned} \dot{x} &= A(\Delta)x + Bw; \\ z &= Cx + Dw \end{aligned} \tag{3.30}$$

where the matrix $A(\Delta) \in \mathbb{R}^{n_s, n_s}$ depends on the uncertainty $\Delta \in \mathbb{R}^{r_\Delta, c_\Delta}$ in a simple additive form, i.e.

$$A(\Delta) = A + L\Delta R \tag{3.31}$$

with $L \in \mathbb{R}^{n_s, r_\Delta}$ and $R \in \mathbb{R}^{c_\Delta, n_s}$. In this case, since Δ is a full real block, $\mathcal{B}_{\widetilde{\mathbb{D}}} = \mathcal{B}_{\mathbb{D}} = \{\Delta \in \mathbb{R}^{r_\Delta, c_\Delta} : \bar{\sigma}(\Delta) \le \rho\}$.

The uncertainty structure in (3.31) has been extensively studied in the literature, see Sect. 3.7. In particular, the smallest value of ρ such that there exists a value of Δ that makes the system (3.30) unstable is called the *real stability radius* of the system, see Definition 3.4 and Theorem 3.3 for its computation. It can be easily verified that the uncertain system (3.30) may be represented in M–Δ form with

$$M(s) = \begin{bmatrix} R \\ C \end{bmatrix} (sI - A)^{-1} \begin{bmatrix} L & B \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix}.$$

Example 3.2 (Real parametric uncertainty) Consider a system described by the transfer function

$$G(s, q) = \frac{s + 3 + q_1}{s^2 + (2 + q_1)s + 5 + q_2} \tag{3.32}$$

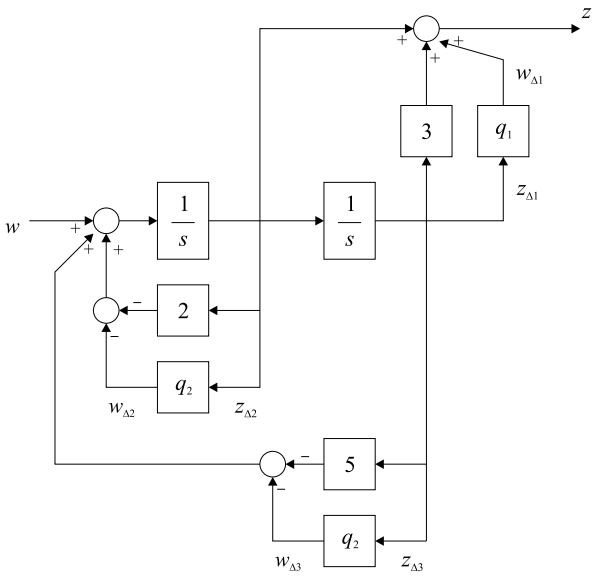

Fig. 3.2 $G(s,q)$ of Example 3.2 expressed in controllable canonical form

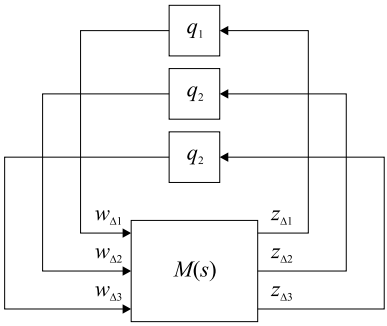

Fig. 3.3 M–Δ representation for Example 3.2

where $q = [q_1\, q_2]^T$ represents a two-dimensional vector of real parametric uncertainty. The vector q is assumed to be bounded in the set $\{q : \|q\|_\infty \leq \rho\}$. An M–Δ representation of the uncertain system in (3.32) may be derived by writing the system in controllable canonical form and *pulling out* the uncertainty, see Fig. 3.2. We therefore obtain

$$M(s) = \begin{bmatrix} 0 & \frac{-1}{s^2+2s+5} & \frac{-1}{s^2+2s+5} & \frac{1}{s^2+2s+5} \\ 0 & \frac{-s}{s^2+2s+5} & \frac{-s}{s^2+2s+5} & \frac{s}{s^2+2s+5} \\ 0 & \frac{-1}{s^2+2s+5} & \frac{-1}{s^2+2s+5} & \frac{1}{s^2+2s+5} \\ 1 & \frac{-(s+3)}{s^2+2s+5} & \frac{-(s+3)}{s^2+2s+5} & \frac{s+3}{s^2+2s+5} \end{bmatrix}$$

and the matrix Δ belongs to the set $\widetilde{\mathbb{D}} \equiv \mathbb{D} = \{\Delta = \text{bdiag}(q_1, q_2 I_2)\}$, with the bound $\|q\|_\infty \leq \rho$. The resulting M–Δ interconnection is shown in Fig. 3.3.

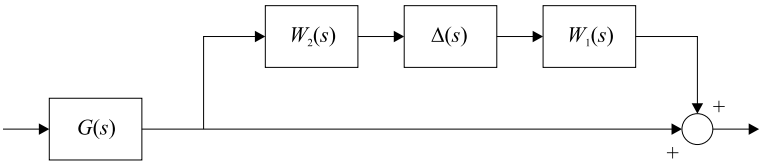

Fig. 3.4 System affected by dynamic multiplicative uncertainty

Example 3.3 (Unmodeled dynamics) Consider a transfer matrix $G(s, \Delta)$ affected by multiplicative uncertainty of the type

$$G(s, \Delta) = \big(I + W_1(s)\Delta(s)W_2(s)\big)G(s) \tag{3.33}$$

where the transfer matrices $W_1(s)$, $W_2(s)$ weight the uncertainty over frequency, as shown in Fig. 3.4.

In this case, $G(s)$ is the nominal model and the uncertainty is constituted by unknown dynamics, expressed in terms of an uncertain stable transfer matrix $\Delta(s)$ that belongs to the set $\mathcal{B}_{\widetilde{\mathbb{D}}} = \{\Delta(s) \in \mathcal{RH}_\infty^{r_\Delta, c_\Delta} : \|\Delta\|_\infty \leq \rho\}$. The multiplicative model (3.33) is immediately rewritten in M–Δ form, letting

$$M(s) = \begin{bmatrix} 0 & W_2(s)G(s) \\ W_1(s) & G(s) \end{bmatrix}.$$

3.7 Robust Stability of M–Δ Configuration

Consider the M–Δ model introduced in the previous section, with $M(s) \in \mathcal{RH}_\infty^{c,r}$ and $\Delta \in \mathcal{B}_{\widetilde{\mathbb{D}}}(\rho)$. For fixed $\rho > 0$, we say that the system is *robustly stable* if it is stable for all uncertainties Δ varying in the set $\mathcal{B}_{\widetilde{\mathbb{D}}}$. More generally, a given property (stability or performance) is robustly satisfied if it holds for all $\Delta \in \mathcal{B}_{\widetilde{\mathbb{D}}}(\rho)$. Consequently, a measure of the "degree" of robustness of a system is given by the largest value of ρ such that the considered property is robustly guaranteed. This measure is generally called the *robustness margin*. When robust stability is of concern, the term *stability radius* is frequently used. The computation of stability radii under various uncertainty structures and for different system descriptions has been one of the main areas of research in the robust control community.

In this section, we report some of the most important results regarding robust stability for different uncertainty configurations. Since the main focus of the section is on stability, to simplify our discussion we refer to systems described by the configuration of Fig. 3.5, where the performance channel $w \to z$ is neglected. In this case, the dimensions of Δ are compatible with M, i.e. $\Delta \in \mathcal{RH}_\infty^{r,c}$.

For historical reasons, we begin our discussion with the case when Δ is an unknown stable transfer matrix bounded in the $\mathcal{H}_\infty$ norm, and show how this problem is equivalent to a "static" problem with a *complex* matrix Δ.

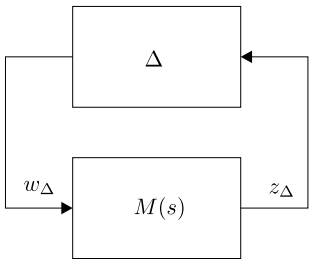

Fig. 3.5 M–Δ configuration for robust stability

3.7.1 Dynamic Uncertainty and Stability Radii

Consider the configuration of Fig. 3.5, in which $M(s) \in \mathcal{RH}_\infty^{c,r}$ and Δ is an unknown transfer matrix $\Delta(s) \in \mathcal{RH}_\infty^{r,c}$. We define the stability radius $r_{\text{LTI}}(M(s))$ under LTI perturbations as the smallest value of $\|\Delta(s)\|_\infty$ such that well-posedness or internal stability are violated.

The radius of stability may be computed by invoking a fundamental result of robustness analysis known as the *small gain theorem*, see e.g. [422].

Theorem 3.1 (Small gain) *Consider the interconnected system in Fig. 3.5, with $M \in \mathcal{RH}_\infty^{c,r}$, and $\rho > 0$. Then, the interconnection is well posed and internally stable for all $\Delta \in \mathcal{RH}_\infty^{r,c}$ with $\|\Delta\|_\infty \leq \rho$ if and only if $\|M\|_\infty < 1/\rho$.*

From this result, it follows immediately that the stability radius under LTI perturbations is given by

$$r_{\text{LTI}}\big(M(s)\big) = \frac{1}{\|M(s)\|_\infty} = \frac{1}{\sup_\omega \bar{\sigma}\big(M(j\omega)\big)}.$$

Remark 3.4 (Extensions of the small gain theorem) It should be observed that the small gain theorem holds for larger classes of uncertainty, see [149]. In fact, this result follows as a special case of the *contraction mapping* theorem in Banach spaces. In particular, it can be reformulated for nonlinear operators from $\mathcal{L}_p^+ \to \mathcal{L}_p^+$, provided that the (Lipschitz) incremental gains of the system and of the uncertainty are used in place of the $\mathcal{H}_\infty$ norm, see also [184]. This fact is useful in robustness analysis, since it allows one to model certain classes of nonlinearities as gain-bounded uncertainties.

Corollary 3.1 *Consider the interconnected system in Fig. 3.5, with $M \in \mathcal{RH}_\infty^{c,r}$, and $\rho > 0$. The following statements are equivalent*:

1. *The interconnection is well posed and internally stable for all $\Delta \in \mathcal{H}_\infty^{r,c}$ with $\|\Delta\|_\infty \leq \rho$;*
2. *The interconnection is well posed and internally stable for all $\Delta \in \mathcal{RH}_\infty^{r,c}$ with $\|\Delta\|_\infty \leq \rho$;*
3. *The interconnection is well posed and internally stable for all matrices $\Delta \in \mathbb{C}^{r,c}$ with $\|\Delta\| \leq \rho$;*
4. *$M \in \mathcal{RH}_\infty^{c,r}$ and $\|M\|_\infty < 1/\rho$.*

An interesting conclusion drawn from this corollary is that checking robust stability of the M–Δ interconnection under purely static uncertainties is necessary and sufficient for robust stability under rather general dynamic perturbations. As a consequence of this fact, when the system is affected by dynamic uncertainty with $\widetilde{\mathbb{D}} = \mathcal{RH}_\infty^{r,c}$, robust stability of the interconnection can be detected by considering the "purely static" matrix structure $\mathbb{D} = \mathbb{C}^{r,c}$.

A related robustness problem arises in the assessment of robust stability for systems described in state space description. Assume for instance that the uncertain linear system is of the form

$$\begin{aligned} \dot{x} &= A(\Delta)x + Bw; \\ z &= Cx + Dw \end{aligned} \tag{3.34}$$

where $A(\Delta) = A + L\Delta R$, $A \in \mathbb{R}^{n_s,n_s}$ is stable and $\Delta \in \mathbb{F}^{r,c}$ is a full block either real (as in Example 3.1) or complex.

Definition 3.4 (Real and complex stability radii) Consider the uncertain system (3.34), with

$$A(\Delta) = A + L\Delta R \tag{3.35}$$

where $A \in \mathbb{R}^{n_s,n_s}$ is stable, $L \in \mathbb{R}^{n_s,r}$, $R \in \mathbb{R}^{c,n_s}$ and $\Delta \in \mathbb{F}^{r,c}$. Then, the *stability radius* of the triple A, L, R is defined as

$$r_{\mathbb{F}}(A, L, R) \doteq \inf\left\{\bar{\sigma}(\Delta) : \Delta \in \mathbb{F}^{r,c} \text{ and } A(\Delta) \text{ is unstable}\right\}. \tag{3.36}$$

In particular, for $\Delta \in \mathbb{R}^{r,c}$, $r_{\mathbb{R}}(A, L, R)$ is called the *real stability radius*. Similarly, for $\Delta \in \mathbb{C}^{r,c}$, $r_{\mathbb{C}}(A, L, R)$ is called the *complex stability radius*.[3]

The next theorem, which is a direct consequence of the small gain theorem, relates the complex stability radius to the computation of the $\mathcal{H}_\infty$ norm of a certain operator.

Theorem 3.2 (Complex stability radius) *Let* $A(\Delta) = A + L\Delta R$, *where* $A \in \mathbb{R}^{n_s,n_s}$ *is stable*, $L \in \mathbb{R}^{n_s,r}$, $R \in \mathbb{R}^{c,n_s}$ *and* $\Delta \in \mathbb{C}^{r,c}$. *Then*, *the* complex stability radius *is given by*

$$r_{\mathbb{C}}(A, L, R) = \frac{1}{\|M\|_\infty}$$

where $M(s) = R(sI - A)^{-1}L$.

Proof Since A is stable, then $M \in \mathcal{RH}_\infty^{c,r}$. Therefore, by the small gain theorem, the closed-loop interconnection of M with $\Delta \in \mathbb{C}^{r,c}$ is stable for all $\|\Delta\| \leq \rho$ if and only if $\|M\|_\infty < 1/\rho$. The statement then follows, noticing that the dynamic matrix of this interconnection is indeed $A + L\Delta R$. □

[3] Real and complex stability radii can also be defined in more general form for complex matrices A, L, R, see for instance [202].

When the uncertain matrix $A(\Delta)$ is expressed in the form (3.35), but the uncertainty Δ is real, the above result is clearly conservative. In this case, a formula for computing exactly the real stability radius is derived in [331].

Theorem 3.3 (Real stability radius) *Let $A(\Delta) = A + L\Delta R$, where $A \in \mathbb{R}^{n_s,n_s}$ is stable, $L \in \mathbb{R}^{n_s,r}$, $R \in \mathbb{R}^{c,n_s}$ and $\Delta \in \mathbb{R}^{r,c}$. Then, the* real stability radius *is given by*

$$r_{\mathbb{R}}(A, L, R) = \left\{ \sup_{\omega} \inf_{\gamma \in (0,1]} \sigma_2 \left(\begin{bmatrix} \mathrm{Re}(M(j\omega)) & -\gamma\omega\,\mathrm{Im}(M(j\omega)) \\ \gamma^{-1}\omega\,\mathrm{Im}(M(j\omega)) & \mathrm{Re}(M(j\omega)) \end{bmatrix} \right) \right\}^{-1} \tag{3.37}$$

where $M(s) = R(sI - A)^{-1}L$, and $\sigma_2(\cdot)$ denotes the second largest singular value.

We remark that the minimization over γ in (3.37) can be easily performed since the function to be optimized is unimodal with respect to $\gamma \in (0, 1]$, see further details in [331].

3.7.2 Structured Singular Value and μ Analysis

In this section, we consider the general case when the matrix Δ in Fig. 3.5 belongs to the structured set defined in (3.25)

$$\widetilde{\mathbb{D}} = \left\{ \Delta \in \mathcal{RH}_\infty^{r,c} : \Delta = \mathrm{bdiag}(q_1 I_{m_1}, \ldots, q_\ell I_{m_\ell}, \Delta_1, \ldots, \Delta_b) \right\}$$

where $q_i \in \mathbb{F}$, $i = 1, \ldots, \ell$ and $\Delta_i \in \mathcal{RH}_\infty^{r_i,c_i}$, $i = 1, \ldots, b$. To the operator structure $\widetilde{\mathbb{D}}$, we associate the matrix structure (3.27)

$$\mathbb{D} = \left\{ \Delta \in \mathbb{F}^{r,c} : \Delta = \mathrm{bdiag}(q_1 I_{m_1}, \ldots, q_\ell I_{m_\ell}, \Delta_1, \ldots, \Delta_b) \right\}.$$

Letting $M \in \mathcal{RH}_\infty^{c,r}$, we consider the complex matrix $M(j\omega) \in \mathbb{C}^{c,r}$ obtained by evaluating the transfer matrix $M(s)$ for $s = j\omega$, with $\omega \in \mathbb{R}_+$. In this setting, the *multivariable stability margin* of $M(j\omega)$ for a system with diagonal perturbations $\mathbb{D}$ is discussed in [336], and its inverse, the *structured singular value* of $M(j\omega)$ with respect to $\mathbb{D}$, is studied in [143]. We now formally define the structured singular value $\mu_{\mathbb{D}}(M(j\omega))$.

Definition 3.5 (Structured singular value) Let $M(j\omega) \in \mathbb{C}^{c,r}$, the *structured singular value* of $M(j\omega)$ with respect to $\mathbb{D}$ is defined as

$$\mu_{\mathbb{D}}\big(M(j\omega)\big) \doteq \frac{1}{\min\{\bar{\sigma}(\Delta) : \det(I - M(j\omega)\Delta) = 0,\ \Delta \in \mathbb{D}\}} \tag{3.38}$$

unless no $\Delta \in \mathbb{D}$ makes $I - M(j\omega)\Delta$ singular, in which case $\mu_{\mathbb{D}}(M(j\omega)) \doteq 0$.

An alternative expression when Δ is complex, is given in [422]

$$\mu_{\mathbb{D}}\big(M(j\omega)\big) = \max_{\Delta \in \mathcal{B}_{\mathbb{D}}(1)} \rho_\lambda\big(M(j\omega)\Delta\big)$$

where $\rho_\lambda(\cdot)$ denotes the spectral radius, i.e. the maximum modulus of the eigenvalues.

The theorem stated next is a fundamental result for robust stability under structured perturbations and constitutes a generalization of the small gain theorem.

Theorem 3.4 (Small μ) *Consider the interconnected system in Fig. 3.5, with $M \in \mathcal{RH}_\infty^{c,r}$, and $\rho > 0$. The interconnection is well posed and internally stable for all $\Delta \in \widetilde{\mathbb{D}}$ with $\|\Delta\|_\infty < \rho$ if and only if*

$$\sup_{\omega \in \mathbb{R}} \mu_{\mathbb{D}}\big(M(j\omega)\big) \leq \frac{1}{\rho}. \tag{3.39}$$

Remark 3.5 (Equivalence between dynamic and static perturbations) From this result, we see that checking robust stability under dynamic perturbations is equivalent to checking robust stability against *purely static* perturbations, since only the static perturbations set $\mathbb{D}$ enters in condition (3.39).

Lemma 3.4 *Let $M(s) = C(sI - A)^{-1}B + D \in \mathcal{RH}_\infty^{c,r}$. Then, the interconnection in Fig. 3.5 is well posed and internally stable for all $\Delta \in \widetilde{\mathbb{D}}$ with $\|\Delta\|_\infty < \rho$ if and only if the two conditions*:

1. *$(I - D\Delta)$ is nonsingular*;
2. *$A + B\Delta(I - D\Delta)^{-1}C$ is stable*

hold for all $\Delta \in \mathbb{D}$ with $\bar{\sigma}(\Delta) < \rho$.

Proof From Theorem 3.4, it follows that the interconnection is well posed and robustly stable for all $\Delta \in \widetilde{\mathbb{D}}$ with $\|\Delta\|_\infty < \rho$ if and only if it is well posed and robustly stable for all $\Delta \in \mathbb{D}$ with $\bar{\sigma}(\Delta) < \rho$. Then, closing the loop on M with $w_\Delta = \Delta z_\Delta$ with $\Delta \in \mathbb{D}$, we have

$$\begin{aligned} \dot{x} &= Ax + B\Delta z_\Delta; \\ (I - D\Delta)z_\Delta &= Cx. \end{aligned}$$

If $(I - D\Delta)$ is nonsingular, then the dynamic matrix of the closed loop is $A + B\Delta(I - D\Delta)^{-1}C$ and the result follows immediately. □

The previous lemma permits us to extend the notion of stability radius to the case of structured static perturbations.

Definition 3.6 (Stability radius under structured perturbations) Let

$$A(\Delta) = A + B\Delta(I - D\Delta)^{-1}C$$

where $A \in \mathbb{R}^{n_s,n_s}$ is stable, $B \in \mathbb{R}^{n_s,r}$, $C \in \mathbb{R}^{c,n_s}$, $D \in \mathbb{R}^{c,r}$ and $\Delta \in \mathbb{D}$. Then, the *stability radius under structured perturbations* is defined as

$$r_{\mathbb{D}}(A, B, C, D) \doteq \inf\big\{\bar{\sigma}(\Delta), \Delta \in \mathbb{D} : A(\Delta) \text{ is unstable or } I - D\Delta \text{ is singular}\big\}. \tag{3.40}$$

As a direct consequence of Theorem 3.4 and Lemma 3.4, we have the result presented next.

Theorem 3.5 (Stability radius under structured perturbations) *Let* $A(\Delta) = A + B\Delta(I - D\Delta)^{-1}C$, *where* $A \in \mathbb{R}^{n_s,n_s}$ *is stable*, $B \in \mathbb{R}^{n_s,r}$, $C \in \mathbb{R}^{c,n_s}$, $D \in \mathbb{R}^{c,r}$ *and* $\Delta \in \mathbb{D}$. *Moreover, let* $M(s) = C(sI - A)^{-1}B + D$, $M \in \mathcal{RH}_\infty^{c,r}$. *Then, the* stability radius under structured perturbations *is given by*

$$r_{\mathbb{D}}(M) = r_{\mathbb{D}}(A, B, C, D) = \frac{1}{\sup_{\omega\in\mathbb{R}} \mu_{\mathbb{D}}(M(j\omega))}.$$

3.7.3 Computation of Bounds on $\mu_{\mathbb{D}}$

The computation of $\mu_{\mathbb{D}}$ under general uncertainty structures $\mathbb{D}$ is a difficult non-convex problem and many results have appeared in the literature in this sense. The interested reader is addressed to Sect. 5.1 and to the survey paper [61] focused on computational complexity in systems and control. For these reasons, research on computation of the structured singular value mainly concentrated on establishing upper and lower bounds on $\mu_{\mathbb{D}}$. In particular, we now consider the purely complex uncertainty structure

$$\mathbb{D} = \left\{\Delta \in \mathbb{C}^{r,c} : \Delta = \text{bdiag}(q_1 I_{m_1}, \ldots, q_\ell I_{m_\ell}, \Delta_1, \ldots, \Delta_b)\right\}$$

consisting of complex repeated scalars $q_i \in \mathbb{C}$ and square full complex blocks $\Delta_i \in \mathbb{C}^{r_i,r_i}$. In addition, let $M \in \mathbb{C}^{r,r}$, where $r = \sum_{i=1}^{\ell} m_i + \sum_{i=1}^{b} r_i$ and introduce the two scalings sets

$$\mathcal{V} \doteq \left\{V \in \mathbb{D} : VV^* = I_r\right\}$$

and

$$\begin{aligned}\mathcal{D} \doteq \big\{D : D = \text{bdiag}(D_1, \ldots, D_\ell, d_1 I_{r_1}, \ldots, d_{b-1} I_{r_{b-1}}, I_{r_b}),\\ D_i \in \mathbb{C}^{m_i,m_i},\ D_i \succ 0,\ d_i \in \mathbb{R},\ d_i > 0\big\}.\end{aligned}$$

Then, the following bounds hold, see e.g. [422]

$$\max_{V\in\mathcal{V}} \rho_\lambda(VM) \le \mu_{\mathbb{D}}(M) \le \inf_{D\in\mathcal{D}} \bar{\sigma}\left(DMD^{-1}\right).$$

The lower bound in this equation is actually an equality, but unfortunately no efficient algorithm with guaranteed global convergence is currently available for its computation. In contrast, the upper bound can be computed efficiently by solving a convex optimization problem (therefore achieving the global optimum), but the bound coincides with $\mu_{\mathbb{D}}(M)$ only for special uncertainty structures for which $2\ell + b \le 3$, see [314] for details and also see [143] for proof of the case $\ell = 0, b = 3$. For $2\ell + b > 3$ the gap between $\mu_{\mathbb{D}}$ and its upper bound can be arbitrarily large, but computational practice shows that this gap often remains quite small.

Similar upper bounds can also be constructed for the more general uncertainty structure (3.27). These upper bounds are still convex with respect to suitably selected scaling matrices and, therefore, can be efficiently computed. However, in general there is no guarantee that they are close to the actual $\mu_{\mathbb{D}}$.

3.7.4 Rank-One μ Problem and Kharitonov Theory

There are special cases in which $\mu_{\mathbb{D}}$ can also be efficiently computed for general uncertainty structures $\mathbb{D}$. One such case is the so-called *rank-one* μ problem, where $M \in \mathbb{C}^{r,r}$ is a rank-one matrix

$$M = uv^*, \quad u, v \in \mathbb{C}^r.$$

Then, it has been shown in [417] that under rather general uncertainty structures, which may include real or complex repeated scalars, and full complex blocks, $\mu_{\mathbb{D}}$ and its upper bound actually coincide.

The interest in the rank-one μ problem also resides in the fact that it provides a connection to Kharitonov-type results for uncertain polynomials discussed in the next section. To explain this connection more precisely, we assume that the transfer matrix $M(s)$ is of the form

$$M(s) = u(s)v^T(s)$$

where $u(s), v(s) \in \mathcal{RH}_\infty^{r,1}$ and that the structured set $\mathbb{D}$ is given by

$$\mathbb{D} = \left\{ \Delta \in \mathbb{R}^{r,r} : \Delta = \operatorname{diag}\left([q_1 \ \cdots \ q_r]\right),\ q_i \in \mathbb{R},\ i = 1, \ldots, r \right\}. \tag{3.41}$$

Subsequently, we define

$$D(s,q) \doteq \det\left(I + M(s)\Delta\right) = 1 + \sum_{i=1}^{r} u_i(s)v_i(s)q_i$$

where $u_i(s)$ and $v_i(s)$, $i = 1, \ldots, r$, are rational functions of the form

$$u_i(s) = n_{ui}(s)/d_{ui}(s), \qquad v_i(s) = n_{vi}(s)/d_{vi}(s)$$

and $n_{ui}(s)$, $d_{ui}(s)$, $n_{vi}(s)$, $d_{vi}(s)$ are coprime polynomials in s. The assumption $u(s), v(s) \in \mathcal{RH}_\infty^{r,1}$ implies that $d_{ui}(s)$, $d_{vi}(s)$ are Hurwitz polynomials, i.e. all their roots lie in the open left half plane.

Then, assuming a bound $\|q\|_\infty \leq 1$ on the uncertainty q, checking robust stability amounts to verifying if $D(j\omega, q) \neq 0$ for all $\omega \in \mathbb{R}$ and all q within this bound. We notice that

$$D(s,q) = \frac{1}{p_0(s)}\left(p_0(s) + \sum_{i=1}^{r} p_i(s)q_i \right)$$

where

$$p_0(s) = \prod_{i=1}^{r} d_{ui}(s) d_{vi}(s);$$

$$p_i(s) = n_{ui}(s) n_{vi}(s) \prod_{k \neq i}^{r} d_{uk}(s) d_{vk}(s), \quad i = 1, \ldots, r.$$

Since $p_0(s)$ is Hurwitz, $p_0(j\omega) \neq 0$ for all $\omega \in \mathbb{R}$. Then, $D(j\omega, q) \neq 0$ if and only if $p(s, q) \neq 0$, where the affine polynomial $p(s, q)$ is given by

$$p(s, q) = p_0(s) + \sum_{i=1}^{r} p_i(s) q_i.$$

Therefore, robust stability is guaranteed if and only if

$$p(j\omega, q) \neq 0$$

for all $\omega \in \mathbb{R}$ and $\|q\|_\infty \leq 1$. Now, since $p(j\omega, 0) \neq 0$, by simple continuity arguments, this condition is satisfied if and only if no root of $p(s, q)$ crosses the imaginary axis for $\|q\|_\infty \leq 1$, i.e. if and only if $p(s, q)$ is Hurwitz for all q. Conversely, one can show that for any polynomial $p(s, q)$ affine in q, there exists a rational matrix of the form $M(s) = u(s)v^T(s)$ with $u(s)$ and $v(s)$ rational, such that $p(s, q) = \det(I + M(s)\Delta)$, with Δ in (3.41).

3.8 Robustness Analysis with Parametric Uncertainty

In this section, we consider systems affected by real parametric uncertainty of the type

$$q \doteq [q_1 \ \cdots \ q_\ell]^T$$

where each q_i, $i = 1, \ldots, \ell$, is bounded in the interval $[q_i^-, q_i^+]$. That is, the uncertainty vector q is assumed to belong to the set

$$\mathcal{B}_q \doteq \left\{ q \in \mathbb{R}^\ell : q_i \in \left[q_i^-, q_i^+ \right], \ i = 1, \ldots, \ell \right\}. \tag{3.42}$$

The set $\mathcal{B}_q$ is a hyperrectangle whose vertices $q^1, \ldots, q^{2^\ell}$ are obtained considering the 2^ℓ combinations of either $q_i = q_i^-$ or $q_i = q_i^+$, for $i = 1, \ldots, \ell$. As described in Example 3.1, systems affected by bounded parametric uncertainty can be represented in the general M–Δ form. In this case, it can be easily shown that the uncertainty Δ in the M–Δ representation depends on the vector

$$\bar{q} \doteq [\bar{q}_1 \ \cdots \ \bar{q}_\ell]^T$$

where

$$\bar{q}_i = \frac{2}{q_i^+ - q_i^-} q_i - \frac{q_i^+ + q_i^-}{q_i^+ - q_i^-}, \quad i = 1, \ldots, \ell.$$

Consequently, the uncertainty Δ is bounded in the set

$$\mathcal{B}_{\mathbb{D}} = \left\{\Delta \in \mathbb{D} : \|\Delta\|_\infty \leq 1\right\}$$

where the structure $\mathbb{D}$ has the simple form

$$\mathbb{D} = \left\{\Delta \in \mathbb{R}^{r,c} : \Delta = \text{bdiag}(\bar{q}_1 I_{m_1}, \ldots, \bar{q}_\ell I_{m_\ell}),\ \bar{q} \in \mathbb{R}^\ell\right\}.$$

We notice that once the uncertain system has been rewritten in the M–Δ form, the robustness results presented in Sect. 3.7 can be applied directly.

In parallel with the μ analysis approach, in the 1980s and early 1990s a framework for studying stability of systems affected by real parametric uncertainty was developed independently. This framework does not necessarily require rewriting the system in M–Δ form and is based upon a direct representation of the SISO uncertain plant in the transfer function form

$$G(s,q) \doteq \frac{N_G(s,q)}{D_G(s,q)}$$

where $N_G(s,q)$ and $D_G(s,q)$ are the numerator and denominator plant polynomials whose coefficients depend on the uncertainty q. Subsequently, for a given controller

$$K(s) \doteq \frac{N_K(s)}{D_K(s)}$$

with numerator and denominator controller polynomials $N_K(s)$ and $D_K(s)$, we construct the closed-loop polynomial

$$\begin{aligned} p(s,q) &= N_K(s)N_G(s,q) + D_K(s)D_G(s,q) \\ &= a_0(q) + a_1(q)s + \cdots + a_n(q)s^n \end{aligned}$$

where the coefficients $a_i(q)$ of $p(s,q)$ are functions of q. An illustration of the parametric approach, borrowed from [383], is given in the next example.

Example 3.4 (DC electric motor with uncertain parameters) Consider the system in Fig. 3.6, representing an armature-controlled DC electric motor with independent excitation. The voltage to angle transfer function $G(s) = \Theta(s)/V(s)$ is given by

$$G(s) = \frac{K_m}{LJs^3 + (RJ + BL)s^2 + (K_m^2 + RB)s}$$

where L is the armature inductance, R is the armature resistance, K_m is the motor electromotive force–speed constant, J is the moment of inertia and B is the mechanical friction. Then, taking a unitary feedback controller $K(s) = 1$, we write the closed-loop polynomial, obtaining

$$p(s) = K_m + \left(K_m^2 + RB\right)s + (RJ + BL)s^2 + LJs^3.$$

Clearly, the values of some of the motor parameters may be uncertain. For example, the moment of inertia and the mechanical friction are functions of the load. Therefore, if the load is not fixed, then the values of J and B are not precisely known. Similarly, the armature resistance R is a parameter that can be measured very accurately but which is subject to temperature variations, and the motor constant K_m is a

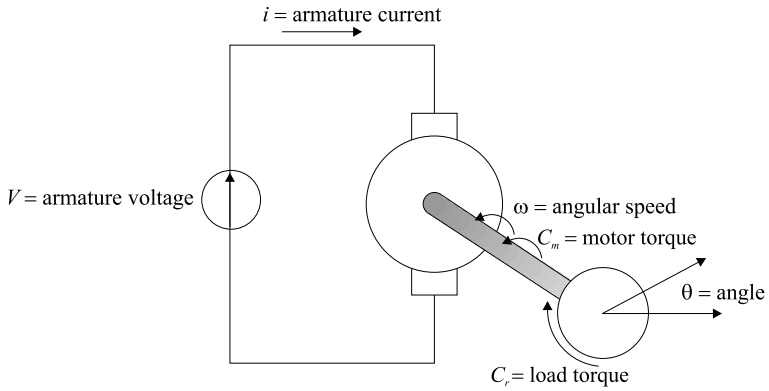

Fig. 3.6 DC-electric motor of Example 3.4

function of the field magnetic flow, which may vary. To summarize, it is reasonable to say that the motor parameters, or a subset of them, may be unknown but bounded within given intervals. More precisely, we can identify uncertain parameters

$$q_1 = L, \quad q_2 = R, \quad q_3 = K_m, \quad q_4 = J, \quad q_5 = B$$

and specify a given range of variation $[q_i^-, q_i^+]$ for each $q_i, i = 1, \ldots, 5$. Then, instead of $G(s)$, we write

$$G(s,q) = \frac{q_3}{q_1 q_4 s^3 + (q_2 q_4 + q_1 q_5)s^2 + (q_3^2 + q_2 q_5)s}$$

and the closed-loop uncertain polynomial becomes

$$p(s,q) = q_3 + \big(q_3^2 + q_2 q_5\big)s + (q_2 q_4 + q_1 q_5)s^2 + q_1 q_4 s^3. \tag{3.43}$$

However, not necessarily all motor parameters are uncertain. For example, we may assume that the armature inductance L, the armature resistance R and the constant K_m are fixed and that the moment of inertia J and the mechanical friction B are unknown. Then, we take $q_1 = J$ and $q_2 = B$ as uncertain parameters. In this case, the closed-loop polynomial has an *affine uncertainty* structure

$$p(s,q) = K_m + \big(K_m^2 + Rq_2\big)s + (Rq_1 + Lq_2)s^2 + Lq_1 s^3.$$

On the other hand, if L, K_m and B are fixed, and R and J are uncertain, then we identify q_1 and q_2 with R and J respectively. In this case, the closed-loop polynomial coefficients are no longer affine functions of the uncertainties, instead they are *multiaffine*[4] functions of q

$$p(s,q) = K_m + \big(K_m^2 + Bq_1\big)s + (q_1 q_2 + BL)s^2 + Lq_2 s^3.$$

In the general case when all motor parameters are uncertain, the polynomial $p(s,q)$ in (3.43) has a *polynomial uncertainty* structure.

[4] A function $f : \mathbb{R}^\ell \to \mathbb{R}$ is said to be multiaffine if the following condition holds: if all components $q_1, \ldots, q_\ell$ except one are fixed, then f is affine. For example, $f(q) = 3q_1q_2q_3 - 6q_1q_3 + 4q_2q_3 + 2q_1 - 2q_2 + q_3 - 1$ is multiaffine.

The study of stability of polynomials with various uncertainty structures is one of the main goals of *parametric stability*. The first result we present is the celebrated Kharitonov theorem [235] on the stability of interval polynomials, i.e. polynomials $p(s,q)$ whose coefficients $a_i(q) = q_i$ are independent and bounded in given intervals $[q_i^-, q_i^+]$.

Theorem 3.6 (Kharitonov) *Consider the interval polynomial family* $\mathcal{P}$

$$\mathcal{P} = \left\{ p(s,q) = q_0 + q_1 s + \cdots + q_n s^n : q_i \in \left[q_i^-, q_i^+\right],\ i = 0, 1, \ldots, n \right\}$$

and the four fixed Kharitonov polynomials

$$\begin{aligned}
p_1(s) &\doteq q_0^- + q_1^- s + q_2^+ s^2 + q_3^+ s^3 + q_4^- s^4 + q_5^- s^5 + q_6^+ s^6 + \cdots; \\
p_2(s) &\doteq q_0^+ + q_1^+ s + q_2^- s^2 + q_3^- s^3 + q_4^+ s^4 + q_5^+ s^5 + q_6^- s^6 + \cdots; \\
p_3(s) &\doteq q_0^+ + q_1^- s + q_2^- s^2 + q_3^+ s^3 + q_4^+ s^4 + q_5^- s^5 + q_6^- s^6 + \cdots; \\
p_4(s) &\doteq q_0^- + q_1^+ s + q_2^+ s^2 + q_3^- s^3 + q_4^- s^4 + q_5^+ s^5 + q_6^+ s^6 + \cdots.
\end{aligned}$$

Then, the interval polynomial family $\mathcal{P}$ *is Hurwitz if and only if the four Kharitonov polynomials are Hurwitz.*

Remark 3.6 (Proof and extensions) This theorem was originally proved by Kharitonov [235] using arguments based on the Hermite–Bieler theorem, see e.g. [175]. A simpler proof based on the so-called value set approach can be found in [286]. We observe that the original statement of the Kharitonov theorem has an invariant degree assumption on the interval family; this hypothesis has been subsequently removed in [199, 222, 413].

The Kharitonov theorem is computationally attractive since it requires checking stability of only four "extreme" polynomials, regardless of the degree of the polynomial. However, it is based on the hypothesis that the polynomial coefficients vary within independent intervals. This hypothesis is generally not satisfied by generic uncertain systems, and therefore the Kharitonov theorem is usually applied by first overbounding the uncertainty with independent intervals at the expense of conservatism, see Chap. 5.

A more general result, which takes into account the dependence of the polynomial coefficients on the uncertain parameters is known as the *edge theorem* [46], is discussed next. In the edge theorem, it is assumed that the coefficients of the polynomial are *affine* functions of the uncertainty vector $q = [q_1 \ \cdots \ q_\ell]^T$ bounded in the hyperrectangle $\mathcal{B}_q$ defined in (3.42). The polynomial family is therefore expressed as

$$\mathcal{P} = \left\{ \begin{array}{l} p(s,q) = a_0(q) + a_1(q)s + \cdots + a_{n-1}(q)s^{n-1} + s^n : \\ \qquad a_i(q) = a_{i0} + \displaystyle\sum_{k=1}^{\ell} a_{ik} q_k,\ q \in \mathcal{B}_q,\ i = 0, 1, \ldots, n-1 \end{array} \right\}. \tag{3.44}$$

The family $\mathcal{P}$ is often called a *polytope of polynomials*, whose vertices are the polynomials $p^i(s)$ corresponding to the 2^ℓ vertices of the hyperrectangle $\mathcal{B}_q$. We define

$$\left[p^i, p^k\right] \doteq \left\{p(s) = \lambda p^i(s) + (1-\lambda)p^k(s),\ \lambda \in [0,1]\right\}$$

as the convex combination of the two vertex polynomials $p^i(s)$ and $p^k(s)$. Then, $[p^i, p^k]$ is said to be an *edge* of the polytope $\mathcal{P}$ if $p^i(s), p^k(s)$ are such that, for any polynomials $p_a(s), p_b(s) \in \mathcal{P}$ with $p_a(s), p_b(s) \notin [p^i, p^k]$, it follows that $[p_a, p_b] \cap [p^i, p^k] = \emptyset$.

The edge theorem is due to [46]. A simplified version of this result in the special case of Hurwitz stability is reported below.

Theorem 3.7 (Edge theorem) *Consider the polytope of polynomials $\mathcal{P}$ defined in* (3.44). *The family $\mathcal{P}$ is Hurwitz if and only if all edges of $\mathcal{P}$ are Hurwitz.*

Remark 3.7 (Edge theorem for $\mathcal{D}$-stability) The edge theorem can be stated in more general form for various root confinement regions of the complex plane, which include the unit disc (Schur stability) and other simply connected sets $\mathcal{D}$. In this case, we deal with the so-called $\mathcal{D}$-stability.

Remark 3.8 (Edge redundancy) Notice that every polynomial belonging to an edge of $\mathcal{P}$ is obtained from one of the $\ell 2^{\ell-1}$ edges of the hyperrectangle $\mathcal{B}_q$, but not necessarily vice versa, [42]. When applying the edge theorem, it is often easier to work with the edges of $\mathcal{B}_q$ instead of the edges of $\mathcal{P}$. In doing this, we accept the possibility of redundancy in the robustness test.

In the context of real parametric uncertainty, for an uncertain polynomial $p(s,q)$, the robustness margin discussed at the beginning of Sect. 3.7 is formally defined as

$$r_q \doteq \inf\left\{\rho : p(s,q) \text{ not Hurwitz for some } q \in \mathcal{B}_q(\rho)\right\}$$

where

$$\mathcal{B}_q(\rho) \doteq \left\{q \in \mathbb{R}^\ell : q_i \in \left[\rho q_i^-, \rho q_i^+\right],\ i = 1, \ldots, \ell\right\}. \tag{3.45}$$

Remark 3.9 (Interval matrices) An obvious question that was studied in depth in the area of parametric robustness is whether the stability results for interval and polytopic polynomials can be extended to uncertain matrices. In particular, a natural generalization of the interval polynomial framework leads to the notion of an *interval matrix*, i.e. a matrix whose entries are bounded in given intervals. Formally, a family $\mathcal{A}$ of interval matrices is defined as

$$\mathcal{A} \doteq \left\{A \in \mathbb{R}^{n,n} : [A]_{i,k} \in \left[a_{ik}^-, a_{ik}^+\right],\ i,k = 1, \ldots, n\right\}. \tag{3.46}$$

Unfortunately, it has been shown that extensions of Kharitonov-like results to stability of interval matrices fail, see for instance [41]. Moreover, this problem has been shown to be computationally intractable, see Sect. 5.1.

Finally, we notice that one of the objectives of parametric stability is to establish "extreme point" and "edge-like" results for special classes of uncertain polynomials and uncertain feedback systems. An important tool for analyzing stability of uncertain systems in the frequency domain is the so-called *value set* (or "template," see [208, 209] for specific discussions on this concept) and efficient algorithms have been developed for its construction. The literature on parametric stability is very broad and the interested reader is redirected, for instance, to the books [41, 58, 136] and the surveys [42, 383].

Chapter 4
Linear Robust Control Design

This chapter continues the study of robustness of uncertain systems initiated in Chap. 3. In particular, we now focus on robust synthesis, i.e., we discuss the problem of designing a controller K such that the interconnection in Fig. 4.1 achieves robust stability with respect to uncertainty and (nominal or robust) performance in the $w \to z$ channel.

4.1 $\mathcal{H}_\infty$ Design

The first problem that we study is the design of a controller such that the interconnection shown in Fig. 4.2 is robustly stable for $\widetilde{\mathbb{D}} = \mathcal{RH}_\infty^{r,c}$, $\Delta \in \mathcal{B}_{\widetilde{\mathbb{D}}}$, where $\widetilde{\mathbb{D}}$ and $\mathcal{B}_{\widetilde{\mathbb{D}}}$ are defined in (3.25) and (3.26). This yields the classical $\mathcal{H}_\infty$ design problem discussed in this section.

In particular, we consider an extended plant G with state space representation

$$\begin{aligned} \dot{x} &= Ax + B_1 w_\Delta + B_2 u; \\ z_\Delta &= C_1 x + D_{11} w_\Delta + D_{12} u; \\ y &= C_2 x + D_{21} w_\Delta \end{aligned} \tag{4.1}$$

where $A \in \mathbb{R}^{n_s,n_s}$, $B_1 \in \mathbb{R}^{n_s,r}$, $B_2 \in \mathbb{R}^{n_s,n_i}$, $C_1 \in \mathbb{R}^{c,n_s}$, $D_{11} \in \mathbb{R}^{c,r}$, $D_{12} \in \mathbb{R}^{c,n_i}$, $C_2 \in \mathbb{R}^{n_o,n_s}$, $D_{21} \in \mathbb{R}^{n_o,r}$. In this section, we make the following standing assumptions

$$\begin{aligned} &(A, B_2) \quad \text{stabilizable;} \\ &(A, C_2) \quad \text{detectable.} \end{aligned} \tag{4.2}$$

The small gain theorem implies that the well posedness and internal stability of the interconnection in Fig. 4.2 for all $\Delta \in \mathcal{B}_{\widetilde{\mathbb{D}}}$ are equivalent to the existence of a controller K such that the closed loop between G and K is well posed and internally

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_4,

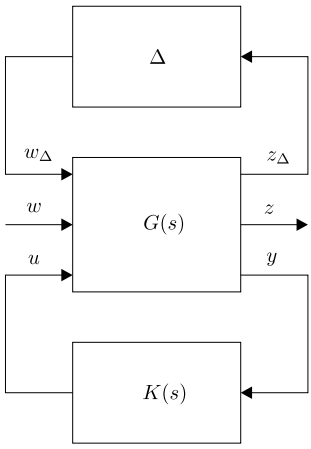

Fig. 4.1 Interconnection for control design guaranteeing robust stability and (nominal or robust) performance. $G(s)$ is the known part of the system, Δ represents the uncertain part, and $K(s)$ is the controller to be designed. The signal w includes noise, disturbances and reference signals, z represents controlled signals and tracking errors, u is the control signal, and y is the measured output

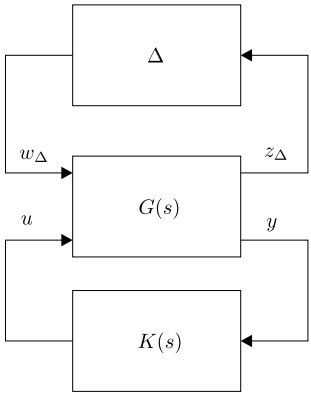

Fig. 4.2 Interconnection for control design guaranteeing robust stability

stable, and the transfer matrix T_{w_Δ, z_Δ} of the channel $w_\Delta \to z_\Delta$ satisfies the bound $\|T_{w_\Delta, z_\Delta}\|_\infty < 1/\rho$. Partitioning the transfer matrix G as

$$\begin{bmatrix} z_\Delta \\ y \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} w_\Delta \\ u \end{bmatrix}$$

we write the transfer matrix T_{w_Δ, z_Δ} in terms of the *lower* linear fractional transformation

$$T_{w_\Delta, z_\Delta} = \mathcal{F}_l(G, K) \doteq G_{11} + G_{12} K (I - G_{22} K)^{-1} G_{21}. \tag{4.3}$$

Letting $\gamma = 1/\rho$, the (γ-suboptimal) $\mathcal{H}_\infty$ robust design problem is now reformulated.

Problem 4.1 (γ-suboptimal $\mathcal{H}_\infty$ design) *For fixed $\gamma > 0$, find, if possible, a controller K such that the closed loop between G and K is well posed and internally stable, and $\|T_{w_\Delta, z_\Delta}\|_\infty < \gamma$.*

In the following, for simplicity, we consider full-order controllers, i.e. controllers of the same order n_s of the plant. Let the controller K, with state space representation

$$\begin{aligned}\dot{x}_K &= A_K x_K + B_K y;\\ u_K &= C_K x_K + D_K y\end{aligned}$$

be described by the quadruple

$$\Omega_K \doteq \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right]. \tag{4.4}$$

Then, the closed-loop transfer matrix $T_{w_\Delta, z_\Delta} = \mathcal{F}_l(G, K)$ has state space representation given by the block matrix

$$\begin{aligned}\Omega_T &\doteq \left[\begin{array}{c|c} A_{cl} & B_{cl} \\ \hline C_{cl} & D_{cl} \end{array}\right]\\ &= \left[\begin{array}{cc|c} A & 0 & B_1 \\ 0 & 0 & 0 \\ \hline C_1 & 0 & D_{11} \end{array}\right] + \left[\begin{array}{cc} 0 & B_2 \\ I & 0 \\ \hline 0 & D_{12} \end{array}\right] \Omega_K \left[\begin{array}{cc|c} 0 & I & 0 \\ C_2 & 0 & D_{21} \end{array}\right].\end{aligned} \tag{4.5}$$

From the bounded real lemma (Lemma 3.2), the closed loop is stable and $\|T_{w_\Delta, z_\Delta}\|_\infty < \gamma$ if and only if there exist a positive definite matrix $P_{cl} \succ 0$ such that the linear matrix inequality

$$\begin{aligned}&\begin{bmatrix} 0 & 0 & 0 \\ 0 & -\gamma I & 0 \\ 0 & 0 & -\gamma I \end{bmatrix} + \begin{bmatrix} P_{cl} & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \Omega_T \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}\\ &\quad + \begin{bmatrix} I & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix} \Omega_T^T \begin{bmatrix} P_{cl} & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \prec 0\end{aligned}$$

holds. This matrix inequality is rewritten more compactly as

$$Z(P_{cl}) + \Phi(P_{cl}) \Omega_K \Psi + \Psi^T \Omega_K^T \Phi^T(P_{cl}) \prec 0 \tag{4.6}$$

where

$$\begin{aligned}Z(P_{cl}) &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\gamma I & 0 \\ 0 & 0 & -\gamma I \end{bmatrix} + \begin{bmatrix} P_{cl} & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \left[\begin{array}{cc|c} A & 0 & B_1 \\ 0 & 0 & 0 \\ \hline C_1 & 0 & D_{11} \end{array}\right] \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}\\ &\quad + \left(\begin{bmatrix} P_{cl} & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \left[\begin{array}{cc|c} A & 0 & B_1 \\ 0 & 0 & 0 \\ \hline C_1 & 0 & D_{11} \end{array}\right] \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \right)^T;\\ \Phi(P_{cl}) &= \begin{bmatrix} P_{cl} & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \left[\begin{array}{cc} 0 & B_2 \\ I & 0 \\ \hline 0 & D_{12} \end{array}\right]; \qquad \Psi = \left[\begin{array}{cc|c} 0 & I & 0 \\ C_2 & 0 & D_{21} \end{array}\right] \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}.\end{aligned}$$

Notice that the matrix inequality (4.6) is *not* jointly linear in P_{cl} and Ω_K. However, (4.6) is an LMI in the controller matrix Ω_K for fixed P_{cl}. A way to overcome

this difficulty has been proposed in [173, 217]. Here, we adopt the representation of [173]. First, we partition P_{cl} and P_{cl}^{-1} as

$$P_{cl} = \begin{bmatrix} S & N \\ N^T & \# \end{bmatrix}, \qquad P_{cl}^{-1} = \begin{bmatrix} R & M \\ M^T & \# \end{bmatrix} \tag{4.7}$$

with $M, N \in \mathbb{R}^{n_s, n_s}$, $R, S \in \mathbb{S}^{n_s}$ and # means "it doesn't matter." Then, the variable Ω_K is eliminated from condition (4.6) using suitable projections. This leads to the existence theorem stated below, see [173].

Theorem 4.1 ($\mathcal{H}_\infty$ LMI solvability conditions) *Consider the extended plant (4.1) with assumptions (4.2). Let N_{12} and N_{21} be orthogonal bases of the null spaces of $[B_2^T \quad D_{12}^T]$ and $[C_2 \quad D_{21}]$ respectively. Then, there exists a controller matrix Ω_K such that the closed loop is internally stable and $\|T_{w_\Delta, z_\Delta}\|_\infty < \gamma$ if and only if there exist symmetric matrices $R, S \in \mathbb{S}^{n_s}$ such that the following system of LMIs is feasible*

$$\left[\begin{array}{c|c} N_{12} & 0 \\ \hline 0 & I \end{array}\right]^T \left[\begin{array}{cc|c} AR + RA^T & RC_1^T & B_1 \\ C_1 R & -\gamma I & D_{11} \\ \hline B_1^T & D_{11}^T & -\gamma I \end{array}\right] \left[\begin{array}{c|c} N_{12} & 0 \\ \hline 0 & I \end{array}\right] \prec 0; \tag{4.8}$$

$$\left[\begin{array}{c|c} N_{21} & 0 \\ \hline 0 & I \end{array}\right]^T \left[\begin{array}{cc|c} A^T S + SA & SB_1 & C_1^T \\ B_1^T S & -\gamma I & D_{11}^T \\ \hline C_1 & D_{11} & -\gamma I \end{array}\right] \left[\begin{array}{c|c} N_{21} & 0 \\ \hline 0 & I \end{array}\right] \prec 0; \tag{4.9}$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \succ 0. \tag{4.10}$$

We now make a few comments regarding the use of this result for constructing an $\mathcal{H}_\infty$ stabilizing controller, see [173]. Once a feasible pair R, S for (4.8)–(4.10) is found, a full-order controller can be determined as follows:

1. Compute via SVD two invertible matrices $M, N \in \mathbb{R}^{n_s, n_s}$ such that $MN^T = I - RS$;
2. The matrix $P_{cl} \succ 0$ in the bounded real lemma is uniquely determined as the solution of the linear equations

$$P_{cl} \begin{bmatrix} R & I \\ M^T & 0 \end{bmatrix} = \begin{bmatrix} I & S \\ 0 & N^T \end{bmatrix};$$

3. For P_{cl} determined as above, any Ω_K that is feasible for the controller LMI (4.6) yields a stabilizing γ-suboptimal controller for the system (4.1).

Remark 4.1 (Reduced-order $\mathcal{H}_\infty$ controller) The LMI condition (4.10) in Theorem 4.1 can be relaxed to nonstrict inequality. The resulting possible rank drop can be exploited to determine a controller of reduced order $k < n_s$. In particular, there

exists a γ-suboptimal controller of order $k < n_s$ if and only if (4.8), (4.9) and the inequality

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \succeq 0$$

hold for some R, S which further satisfy

$$\operatorname{rank}(I - RS) \leq k.$$

We also remark that in the third step of the above procedure for determining the controller matrix Ω_K it is not necessary to solve numerically the controller LMI, since analytic formulas are given in [172, 217].

4.1.1 Regular $\mathcal{H}_\infty$ Problem

In the so-called regular $\mathcal{H}_\infty$ problem discussed in classical references such as [144], the following standard simplifying assumptions are made

$$D_{11} = 0, \qquad D_{12}^T \begin{bmatrix} C_1 & D_{12} \end{bmatrix} = \begin{bmatrix} 0 & I \end{bmatrix}, \qquad D_{21} \begin{bmatrix} B_1^T & D_{21}^T \end{bmatrix} = \begin{bmatrix} 0 & I \end{bmatrix}. \tag{4.11}$$

With these assumptions, it may be easily verified[1] that, letting $X \doteq \gamma R^{-1}$ and $Y \doteq \gamma S^{-1}$, and using the Schur complement rule, conditions (4.8)–(4.10) are equivalent to

$$A^T X + XA + X\big(\gamma^{-2} B_1 B_1^T - B_2 B_2^T\big)X + C_1^T C_1 \prec 0; \tag{4.12}$$

$$AY + YA^T + Y\big(\gamma^{-2} C_1^T C_1 - C_2^T C_2\big)Y + B_1 B_1^T \prec 0; \tag{4.13}$$

$$X \succ 0, \qquad Y \succ 0, \qquad \rho_\lambda(XY) < \gamma^2. \tag{4.14}$$

The left-hand sides of inequalities (4.12) and (4.13) coincide with the expressions arising in the standard Riccati-based $\mathcal{H}_\infty$ formulas [144]. In particular, the connection with the algebraic Riccati equations (AREs) is detailed in the next lemma.

Lemma 4.1 (ARE based $\mathcal{H}_\infty$ solution) *Consider the extended plant* (4.1) *satisfying the following regularity conditions*:

1. $D_{11} = 0$, $D_{12}^T[C_1 \ D_{12}] = [0 \ I]$, $D_{21}[B_1^T \ D_{21}^T] = [0 \ I]$;
2. $\begin{bmatrix} A - j\omega I & B_2 \\ C_1 & D_{12} \end{bmatrix}$ *has full column rank for all* ω;
3. $\begin{bmatrix} A - j\omega I & B_1 \\ C_2 & D_{21} \end{bmatrix}$ *has full row rank for all* ω.

[1] To this end, take $N_{21} = \begin{bmatrix} I & 0 \\ -D_{21}^T C_2 & D_{21}^{T\perp} \end{bmatrix}$, where $D_{21}^{T\perp}$ is the orthogonal complement of D_{21}^T, i.e. $D_{21} D_{21}^{T\perp} = 0$ and $D_{21}^{T\perp T} D_{21}^{T\perp} = I$. Since $D_{21} B_1^T = 0$ we may write $B_1^T = D_{21}^{T\perp} Z$ for some matrix Z and, therefore, $B_1 D_{21}^{T\perp} D_{21}^{T\perp T} B_1^T = B_1 B_1^T$. Everything follows in a similar way for the first inequality, choosing $N_{12} = \begin{bmatrix} I & 0 \\ -B_2^T D_{12} & D_{12}^T \end{bmatrix}$.

Suppose that the two algebraic Riccati inequalities (ARIs)

$$A^T X + XA + X\left(\gamma^{-2} B_1 B_1^T - B_2 B_2^T\right) X + C_1^T C_1 \prec 0; \tag{4.15}$$

$$AY + YA^T + Y\left(\gamma^{-2} C_1^T C_1 - C_2^T C_2\right) Y + B_1 B_1^T \prec 0 \tag{4.16}$$

admit positive definite solutions $X_0, Y_0 \succ 0$. *Then, the corresponding AREs*

$$A^T X + XA + X\left(\gamma^{-2} B_1 B_1^T - B_2 B_2^T\right) X + C_1^T C_1 = 0; \tag{4.17}$$

$$AY + YA^T + Y\left(\gamma^{-2} C_1^T C_1 - C_2^T C_2\right) Y + B_1 B_1^T = 0 \tag{4.18}$$

have stabilizing solutions[2] X_∞, Y_∞ *satisfying*

$$0 \preceq X_\infty \prec X_0;$$

$$0 \preceq Y_\infty \prec Y_0.$$

Moreover, if $\rho_\lambda(X_0 Y_0) < \gamma^2$, *then* $\rho_\lambda(X_\infty Y_\infty) < \gamma^2$.

Conversely, if the AREs (4.17) *and* (4.18) *admit stabilizing solutions* $X_\infty \succeq 0$, $Y_\infty \succeq 0$ *satisfying* $\rho_\lambda(X_\infty Y_\infty) < \gamma^2$, *then there exist feasible solutions* $X_0, Y_0 \succ 0$ *of the ARIs* (4.15) *and* (4.16) *such that* $\rho_\lambda(X_0 Y_0) < \gamma^2$.

Remark 4.2 (Controller formulas) Under the simplifying assumptions (4.11), for any pair X, Y which is feasible for (4.12)–(4.14), a γ-suboptimal controller may be constructed as

$$A_K = A + \left(\gamma^{-2} B_1 B_1^T - B_2 B_2^T\right) X + \left(\gamma^{-2} YX - I\right)^{-1} Y C_2^T C_2;$$

$$B_K = -\left(\gamma^{-2} YX - I\right)^{-1} Y C_2^T;$$

$$C_K = -B_2^T X;$$

$$D_K = 0.$$

Moreover, if $X_\infty \succeq 0$, $Y_\infty \succeq 0$ satisfying $\rho_\lambda(X_\infty Y_\infty) < \gamma^2$ are stabilizing solutions of the AREs (4.17) and (4.18), then substituting formally $X = X_\infty$, $Y = Y_\infty$ in the above controller formulas, we obtain the expression of the so-called *central* controller given in [144]. Riccati-based solutions for the general $\mathcal{H}_\infty$ problem, without simplifying assumptions, are given in [347, 348] and in [339].

4.1.2 Alternative LMI Solution for $\mathcal{H}_\infty$ Design

An alternative approach to determine synthesis LMIs for the $\mathcal{H}_\infty$ problem is based on a systematic technique for transforming analysis LMI conditions into synthesis LMIs using a suitable linearizing change of controller variables and a congruence transformation, see [277, 346]. The next lemma is based on [346].

[2] $X \in \mathbb{S}^n$ is a stabilizing solution of the ARE $A^T X + XA + XRX + Q = 0$ if it satisfies the equation and $A + RX$ is stable.

Lemma 4.2 (Linearizing change of variables) *Let* P_{cl}, P_{cl}^{-1} *be partitioned as in* (4.7), *where* $R, S \in \mathbb{S}^n$ *and* $M, N \in \mathbb{R}^{n_s, n_s}$. *Let*

$$\begin{aligned}
\widehat{A} &\doteq NA_KM^T + NB_KC_2R + SB_2C_KM^T + S(A + B_2D_KC_2)R;\\
\widehat{B} &\doteq NB_K + SB_2D_K;\\
\widehat{C} &\doteq C_KM^T + D_KC_2R;\\
\widehat{D} &\doteq D_K
\end{aligned} \tag{4.19}$$

and

$$\Pi_1 \doteq \begin{bmatrix} R & I \\ M^T & 0 \end{bmatrix}, \qquad \Pi_2 \doteq \begin{bmatrix} I & S \\ 0 & N^T \end{bmatrix}.$$

Then, it holds that

$$\begin{aligned}
P_{cl}\Pi_1 &= \Pi_2;\\
\Pi_1^T P_{cl}A_{cl}\Pi_1 = \Pi_2^T A_{cl}\Pi_1 &= \begin{bmatrix} AR + B_2\widehat{C} & A + B_2\widehat{D}C_2 \\ \widehat{A} & SA + \widehat{B}C_2 \end{bmatrix};\\
\Pi_1^T P_{cl}B_{cl} = \Pi_2^T B_{cl} &= \begin{bmatrix} B_1 + B_2\widehat{D}D_{21} \\ SB_1 + \widehat{B}D_{21} \end{bmatrix};\\
C_{cl}\Pi_1 &= \begin{bmatrix} C_1R + D_{12}\widehat{C} & C_1 + D_{12}\widehat{D}C_2 \end{bmatrix};\\
\Pi_1^T P_{cl}\Pi_1 = \Pi_1^T\Pi_2 &= \begin{bmatrix} R & I \\ I & S \end{bmatrix}.
\end{aligned}$$

Applying the bounded real lemma to the closed-loop system Ω_T in (4.5), and performing the congruence transformation with $\mathrm{bdiag}(\Pi_1, I, I)$, we obtain a result which provides alternative solvability conditions in the modified controller variables. This is stated in the next theorem, see [346] for proof.

Theorem 4.2 (Alternative $\mathcal{H}_\infty$ LMI solution) *Consider the extended plant* (4.1) *with assumptions* (4.2). *Then, there exists a controller matrix* Ω_K *such that the closed loop is internally stable and* $\|T_{w_\Delta, z_\Delta}\|_\infty < \gamma$ *if and only if there exist symmetric matrices* $R, S \in \mathbb{S}^{n_s}$ *and matrices* $\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}$ *such that the following system of LMIs is feasible*[3]

$$\begin{bmatrix}
AR + RA^T + B_2\widehat{C} + \widehat{C}^TB_2^T & \widehat{A}^T + A + B_2\widehat{D}C_2 & B_1 + B_2\widehat{D}D_{21} & * \\
* & A^TS + SA + \widehat{B}C_2 + C_2^T\widehat{B}^T & SB_1 + \widehat{B}D_{21} & * \\
* & * & -\gamma I & * \\
C_1R + D_{12}\widehat{C} & C_1 + D_{12}\widehat{D}C_2 & D_{11} + D_{12}\widehat{D}D_{21} & -\gamma I
\end{bmatrix} \prec 0; \tag{4.20}$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \succ 0. \tag{4.21}$$

[3] The asterisks indicate elements whose values are easily inferred by symmetry.

We now discuss this result and show how a γ-suboptimal $\mathcal{H}_\infty$ controller can be obtained. For any matrices $\widehat{A}$, $\widehat{B}$, $\widehat{C}$, $\widehat{D}$, R, S satisfying (4.20) and (4.21), we can recover a γ-suboptimal $\mathcal{H}_\infty$ controller by inverting the change of variables of Lemma 4.2 as follows:

1. Compute via SVD two square and invertible matrices $M, N \in \mathbb{R}^{n_s,n_s}$ such that $MN^T = I - RS$. We remark that this is always possible for full-order controller design due to the constraint (4.21);
2. The controller matrices are uniquely determined as

$$\begin{aligned} D_K &= \widehat{D}; \\ C_K &= (\widehat{C} - D_K C_2 R) M^{-T}; \\ B_K &= N^{-1}(\widehat{B} - S B_2 D_K); \\ A_K &= N^{-1}\big(\widehat{A} - N B_K C_2 R - S B_2 C_K M^T - S(A + B_2 D_K C_2) R\big) M^{-T} \end{aligned} \tag{4.22}$$

where $M^{-T} = (M^{-1})^T = (M^T)^{-1}$.

4.1.3 μ Synthesis

In this section, we consider the problem of designing a controller such that the interconnection in Fig. 4.2 is well posed and robustly stable for structured uncertainty of the form

$$\widetilde{\mathbb{D}} \doteq \left\{ \Delta \in \mathcal{RH}_\infty^{r,c} : \Delta = \text{bdiag}(\Delta_1, \ldots, \Delta_b),\ \Delta_i \in \mathcal{RH}_\infty^{r_i,r_i} \right\}$$

with the bound $\|\Delta\|_\infty < \rho$. From the small μ theorem applied to the closed-loop system $\mathcal{F}_l(G, K)$, the problem amounts to determining a controller $K(s)$ such that

$$\sup_{\omega \in \mathbb{R}} \mu_{\mathbb{D}}\big(\mathcal{F}_l\big(G(j\omega), K(j\omega)\big)\big) \leq \frac{1}{\rho}$$

where $\mathbb{D}$ is the *purely complex* uncertainty structure

$$\mathbb{D} = \left\{ \Delta \in \mathbb{C}^{r,c} : \Delta = \text{bdiag}(\Delta_1, \ldots, \Delta_b),\ \Delta_i \in \mathbb{C}^{r_i,r_i} \right\}.$$

Unfortunately, this problem cannot be solved efficiently in general. Indeed, even evaluating $\mu_{\mathbb{D}}$ for *fixed* K is computationally difficult, as discussed in Sect. 3.7.3. However, in practice, the problem can be tackled using a suboptimal iterative approach, generally denoted as D–K iteration, see for instance [38, 106, 422].

D–K Iteration for μ Synthesis

1. Let $\widehat{K}(s)$ be any stabilizing controller for the system (usually, $\widehat{K}(s)$ is computed solving a standard $\mathcal{H}_\infty$ problem), and let $\{\omega_1, \ldots, \omega_N\}$ be a suitable frequency grid;

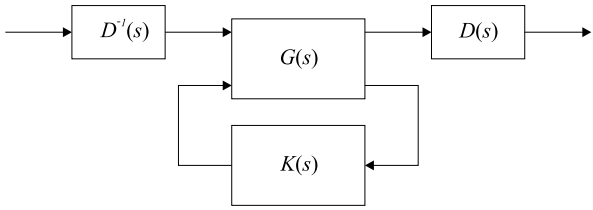

Fig. 4.3 Configuration for μ synthesis via D–K iteration

2. Fix $K(s) = \widehat{K}(s)$. Determine the sequence of scaling matrices of the form
$$\widetilde{D}(\omega_i) \doteq \text{bdiag}\big(\widetilde{d}_1(\omega_i) I_{r_1}, \ldots, \widetilde{d}_{b-1}(\omega_i) I_{r_{b-1}}, I_{r_b}\big), \quad i = 1, \ldots, N$$
with $\widetilde{d}_k(\omega_i) > 0$ for $k = 1, \ldots, b-1$, such that
$$\bar{\sigma}\big(\widetilde{D}(\omega_i) \mathcal{F}_l\big(G(j\omega_i), \hat{K}(j\omega_i)\big) \widetilde{D}^{-1}(\omega_i)\big), \quad i = 1, \ldots, N$$
is minimized;
3. Find scalar transfer functions $d_k(s) \in \mathcal{RH}_\infty$ such that $d_k^{-1}(s) \in \mathcal{RH}_\infty$ and $|d_k(j\omega_i)| \simeq \widetilde{d}_k(j\omega_i)$, for $i = 1, \ldots, N$ and $k = 1, \ldots, b-1$, and construct the transfer matrix
$$D(s) \doteq \text{bdiag}\big(d_1(s) I_{r_1}, \ldots, d_{b-1}(s) I_{r_{b-1}}, I_{r_b}\big);$$
4. Consider the configuration in Fig. 4.3, solve the $\mathcal{H}_\infty$ minimization problem
$$\widehat{K} = \arg\min \big\| D(s) \mathcal{F}_l\big(G(s), K(s)\big) D^{-1}(s) \big\|_\infty$$
and repeat the iterations from step 2.

The D–K iteration is terminated when either an $\mathcal{H}_\infty$ norm smaller than $1/\rho$ is achieved in step 4, or no significant improvement is obtained with respect to the previous iteration.

Remark 4.3 (Convergence of D–K iteration) The D–K iteration is not guaranteed to converge to a global optimum (nor to a local one), but can be a useful tool for many practical design problems.

For fixed K, the D-subproblem (step 2) is a convex optimization problem that may be efficiently solved. Similarly, for fixed D, the K-subproblem (step 4) is a standard $\mathcal{H}_\infty$ design, which can be directly solved by means of the Riccati or LMI approach discussed in the previous sections. However, the problem is not jointly convex in D and K. The rational approximations required in step 3 may be performed using interpolation theory, see e.g. [416], but this approach usually results in high-order transfer functions. This is clearly not desirable, since it increases the order of the scaled plant, and therefore of the controller. For this reason, a frequently used method is based on graphical matching by means of low-order transfer functions.

We finally remark that the D–K iteration may still be applied for more general uncertainty structures, involving for instance repeated scalar blocks, provided that suitable scalings are used in the evaluation of the upper bound of $\mu_{\mathbb{D}}$, see e.g. Sect. 3.7.3.

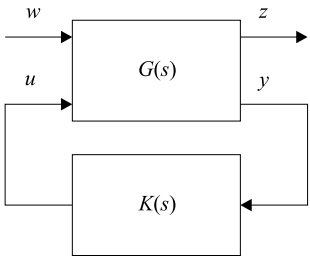

Fig. 4.4 Interconnection for nominal $\mathcal{H}_2$ design

4.2 $\mathcal{H}_2$ Design

In this section, we discuss the problem of designing a controller K such that its interconnection with the *nominal* model G, see Fig. 4.4, provides closed-loop stability and attains a level γ of $\mathcal{H}_2$ performance on the $w \to z$ channel.

Let G be represented by

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 u; \\ z &= C_1 x + D_{11} w + D_{12} u; \\ y &= C_2 x + D_{21} w \end{aligned} \tag{4.23}$$

where $A \in \mathbb{R}^{n_s,n_s}$, $B_1 \in \mathbb{R}^{n_s,q}$, $B_2 \in \mathbb{R}^{n_s,n_i}$, $C_1 \in \mathbb{R}^{p,n_s}$, $D_{11} \in \mathbb{R}^{p,q}$, $D_{12} \in \mathbb{R}^{p,n_i}$, $C_2 \in \mathbb{R}^{n_o,n_s}$, $D_{21} \in \mathbb{R}^{n_o,q}$. In this section, we make the following standing assumptions

$$\begin{aligned} &(A, B_2) \quad \text{stabilizable;} \\ &(A, C_2) \quad \text{detectable.} \end{aligned} \tag{4.24}$$

Following the same derivation as in the beginning of Sect. 4.1, we obtain the closed-loop system $T_{w,z}$ described by the block matrix

$$\Omega_T = \left[\begin{array}{c|c} A_{cl} & B_{cl} \\ \hline C_{cl} & D_{cl} \end{array}\right]$$

given in (4.5). Then, for fixed $\gamma > 0$ the $\mathcal{H}_2$ performance design problem is formulated.

Problem 4.2 (γ-suboptimal $\mathcal{H}_2$ design) *For fixed $\gamma > 0$, find a controller K such that the closed loop between G and K is well posed and internally stable, and $\|T_{w,z}\|_2^2 < \gamma$.*

To obtain the $\mathcal{H}_2$ synthesis conditions we proceed as follows: first, we apply Lemma 3.1 to the closed-loop matrices $A_{cl}, B_{cl}, C_{cl}, D_{cl}$. Then, we introduce the change of controller variables of Lemma 4.2, and perform the congruence transformation with bdiag(Π_1, I), obtaining the result summarized in the next theorem, see [346].

Theorem 4.3 ($\mathcal{H}_2$ LMI solution) *Consider the extended plant* (4.23) *with assumptions* (4.24)*. Then, there exists a controller matrix* Ω_K *such that the closed loop is internally stable and* $\|T_{w,z}\|_2^2 < \gamma$ *if and only if there exist symmetric matrices* $R, S \in \mathbb{S}^{n_s}$ *and matrices* $Q, \widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}$ *such that the following system of LMIs is feasible*

$$\begin{bmatrix} AR + RA^T + B_2\widehat{C} + \widehat{C}^T B_2^T & \widehat{A}^T + A + B_2\widehat{D}C_2 & B_1 + B_2\widehat{D}D_{21} \\ * & A^T S + SA + \widehat{B}C_2 + C_2^T\widehat{B}^T & SB_1 + \widehat{B}D_{21} \\ * & * & -I \end{bmatrix} \prec 0;$$

$$\begin{bmatrix} R & I & (C_1R + D_{12}\widehat{C})^T \\ I & S & (C_1 + D_{12}\widehat{D}C_2)^T \\ (C_1R + D_{12}\widehat{C}) & (C_1 + D_{12}\widehat{D}C_2) & Q \end{bmatrix} \succ 0;$$

$$D_{11} + D_{12}\widehat{D}D_{21} = 0;$$

$$\operatorname{Tr} Q < \gamma.$$

For any matrices $\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}, R, S$ satisfying the LMI solvability conditions of this theorem, we recover a γ-suboptimal $\mathcal{H}_2$ controller by inverting the change of variables in (4.19) using (4.22).

Under the standard regularity assumptions introduced in Sect. 4.1.1, the solution of the $\mathcal{H}_2$ problem can be alternatively stated in terms of two *decoupled* AREs, as summarized in the next lemma, see e.g. [422].

Lemma 4.3 (ARE based $\mathcal{H}_2$ solution) *Consider the extended plant* (4.23) *satisfying the following regularity assumptions*:

1. $D_{11} = 0$;
2. $R_1 \doteq D_{12}^T D_{12} \succ 0$ *and* $R_2 \doteq D_{21} D_{21}^T \succ 0$;
3. $\left[\begin{smallmatrix} A - j\omega I & B_2 \\ C_1 & D_{12} \end{smallmatrix}\right]$ *has full column rank for all* ω;
4. $\left[\begin{smallmatrix} A - j\omega I & B_1 \\ C_2 & D_{21} \end{smallmatrix}\right]$ *has full row rank for all* ω.

Define

$$A_x \doteq A - B_2 R_1^{-1} D_{12}^T C_1, \qquad A_y \doteq A - B_1 D_{21}^T R_2^{-1} C_2$$

and let $X, Y \succeq 0$ *be stabilizing solutions of the two AREs*

$$X A_x + A_x^T X - X B_2 R_1^{-1} B_2^T X + C_1^T \big(I - D_{12} R_1^{-1} D_{12}^T\big) C_1 = 0; \tag{4.25}$$

$$Y A_y^T + A_y Y - Y C_2^T R_2^{-1} C_2 Y + B_1 \big(I - D_{21}^T R_2^{-1} D_{21}\big) B_1^T = 0. \tag{4.26}$$

Then, the controller

$$B_K = \big(B_1 D_{21}^T + Y C_2^T\big) R_2^{-1}; \tag{4.27}$$

$$C_K = -R_1^{-1}\big(D_{12}^T C_1 + B_2^T X\big); \tag{4.28}$$

$$A_K = A + B_2 C_K - B_K C_2; \tag{4.29}$$

$$D_K = 0 \tag{4.30}$$

stabilizes the closed-loop system and minimizes the $\mathcal{H}_2$ norm of the $w \to z$ channel. Moreover, with this controller we have

$$\|T_{w,z}\|_2^2 = \operatorname{Tr} B_1^T X B_1 + \operatorname{Tr} C_K Y C_K^T R_1.$$

Remark 4.4 (LQG control) The $\mathcal{H}_2$ control problem is equivalent to classical linear quadratic Gaussian (LQG) control. The connection is established assuming that the exogenous signal $w(t)$ is white Gaussian noise with $\mathrm{E}(\mathbf{w}(t)) = 0$, $\mathrm{E}(\mathbf{w}(t)\mathbf{w}^T(t+\tau)) = I\delta(\tau)$, where $\delta(\cdot)$ is the Dirac impulse function. In this case, the objective is to determine a control law that stabilizes the closed loop and minimizes the average output power

$$\mathrm{E}\left(\lim_{T\to\infty} \frac{1}{T}\int_0^T \|z\|^2 \,\mathrm{d}t\right) = \|\mathbf{z}\|_{\mathrm{rms}}^2 = \|T_{w,z}\|_2^2.$$

An important special case that falls out directly from the $\mathcal{H}_2$ problem is the classical *linear quadratic regulator* (LQR) problem, which is discussed next.

4.2.1 Linear Quadratic Regulator

In the traditional linear quadratic regulator problem, see for instance [21, 246], we consider the system description

$$\dot{x} = Ax + B_2 u \tag{4.31}$$

with initial state $x(0) = x_0 \in \mathbb{R}^{n_s}$ given but arbitrary. We look for a state feedback control law of the form $u = -Kx$, where $K \in \mathbb{R}^{n_i, n_s}$ such that the closed loop is stable and the following cost is minimized

$$c = \int_0^\infty \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^T Q \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \mathrm{d}t \tag{4.32}$$

where

$$Q \doteq \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{xu}^T & Q_{uu} \end{bmatrix} \succeq 0. \tag{4.33}$$

Since Q is positive semidefinite, it can be factored as

$$Q = \begin{bmatrix} C_1^T \\ D_{12}^T \end{bmatrix} \begin{bmatrix} C_1 & D_{12} \end{bmatrix}.$$

Therefore, the above cost c is equivalent to $\|z\|_2^2$, where z is the output associated with (4.31)

$$z = C_1 x + D_{12} u. \tag{4.34}$$

We now state a result analogous to Lemma 4.3 for the LQR problem.

Lemma 4.4 (Extended LQR) *Consider the plant* (4.31), (4.34) *satisfying the following hypotheses*:

1. (A, B_2) *is stabilizable*;
2. $\left[\begin{smallmatrix} A-j\omega I & B_2 \\ C_1 & D_{12} \end{smallmatrix}\right]$ *has full column rank for all* ω;
3. $[D_{12}\ D_{12}^{\perp}]^T[D_{12}\ D_{12}^{\perp}] = \left[\begin{smallmatrix} Q_{uu} & 0 \\ 0 & I \end{smallmatrix}\right] \succ 0$.

Then, there exists a unique optimal control minimizing the cost function given in (4.32)

$$K = Q_{uu}^{-1}\left(B_2^T X + D_{12}^T C_1\right)$$

where $X \succeq 0$ *is the stabilizing solution of the ARE*

$$\begin{aligned}\left(A - Q_{uu}^{-1} D_{12}^T C_1\right)^T X + X\left(A - Q_{uu}^{-1} D_{12}^T C_1\right) - X B_2 Q_{uu}^{-1} B_2^T X \\ + C_1^T D_{12}^{\perp} D_{12}^{\perp T} C_1 = 0.\end{aligned} \tag{4.35}$$

Moreover, the optimal cost is $c = x_0^T X x_0$.

Remark 4.5 (Standard LQR) We notice that in the standard LQR it is often assumed that $Q_{xu} = 0$, i.e. $C_1^T D_{12} = 0$. This assumption is made without loss of generality, since there always exists a coordinate transformation that brings a general LQR problem in standard form. In this situation, the cost function is

$$c = \int_0^{\infty} \left(x^T(t) Q_{xx} x(t) + u^T(t) Q_{uu} u(t)\right) \mathrm{d}t \tag{4.36}$$

with $Q_{xx} \succeq 0$ and $Q_{uu} \succ 0$. Then, the ARE in (4.35) is simplified as

$$A^T X + X A - X B_2 Q_{uu}^{-1} B_2^T X + Q_{xx} = 0. \tag{4.37}$$

Furthermore, the unique optimal control law is given by

$$u(t) = -Q_{uu}^{-1} B_2^T X x(t) \tag{4.38}$$

where $X \succ 0$ is the corresponding stabilizing solution of (4.37).

In the next section we study the related problems of quadratic stabilizability and guaranteed-cost linear quadratic control.

4.2.2 Quadratic Stabilizability and Guaranteed-Cost

We next consider the case when the system (4.31) described in state space form is affected by uncertainty. In particular, we study

$$\dot{x} = A(\Delta)x + B_2 u, \quad x(0) = x_0 \in \mathbb{R}^{n_s} \tag{4.39}$$

where Δ belongs to the uncertainty set $\mathcal{B}_{\mathbb{D}}$ defined in (3.28). This system is said to be *quadratically stable* if there exists $P \succ 0$ such that the Lyapunov inequality

$$A(\Delta)P + P A^T(\Delta) \prec 0 \tag{4.40}$$

is satisfied for all $\Delta \in \mathcal{B}_{\mathbb{D}}$. Then, we address the problem of finding a control law $u = -Kx$, $K \in \mathbb{R}^{n_i, n_s}$, such that the closed-loop is quadratically stable. That is, we seek $P \succ 0$ and K such that the inequality

$$\big(A(\Delta) + B_2 K\big)P + P\big(A(\Delta) + B_2 K\big)^T \prec 0$$

is satisfied for all $\Delta \in \mathcal{B}_{\mathbb{D}}$. This matrix inequality is not jointly linear in P and K. However, introducing the new matrix variable $Y = KP$ as in [68] we obtain the following *robust LMI* in the variables P, Y

$$A(\Delta)P + PA^T(\Delta) + B_2 Y + Y^T B_2^T \prec 0, \quad \text{for all } \Delta \in \mathcal{B}_{\mathbb{D}}, \tag{4.41}$$

where the matrix $Y \in \mathbb{R}^{n_i, n_s}$ is not necessarily symmetric. We say that the system (4.39) is *quadratically stabilizable* if there exist matrices $P \succ 0$ and Y such that the above LMI holds for all $\Delta \in \mathcal{B}_{\mathbb{D}}$, that is if it holds *robustly* with respect to Δ. Robust LMIs are discussed in the next section. Relations between quadratic stabilization and $\mathcal{H}_\infty$ design can be found in a number of papers, including [234].

Remark 4.6 (Quadratic stability/stabilizability for interval matrices) In general, the matrix inequalities (4.40) or (4.41) are to be solved for all $\Delta \in \mathcal{B}_{\mathbb{D}}$, i.e. we have to satisfy an infinite number of LMIs simultaneously. In many important cases, owing to convexity properties, this problem can be reduced to the solution of a *finite* number of LMIs. For example, let $\mathcal{A}$ be an $n_s \times n_s$ interval matrix family, defined as in (3.46). In this case, the problem of quadratic stability, i.e. finding a *common solution* $P \succ 0$ satisfying (4.40) for all $[A]_{i,k}$ in the intervals $[a_{ik}^-, a_{ik}^+]$, $i, k = 1, \ldots, n_s$, is equivalent to finding $P \succ 0$ that satisfies

$$A^\ell P + P\big(A^\ell\big)^T \prec 0, \quad \ell = 1, \ldots, 2^{n_s^2}$$

where A^ℓ is a vertex matrix obtained setting all the entries of the interval matrix to either lower a_{ik}^- or upper a_{ik}^+ bounds, see e.g. [206]. Extreme point results of this kind also hold for more general classes than interval matrices, see e.g. [41]. A major computational issue, however, is that the number of LMIs which should be simultaneously solved is exponential in n_s. It has been recently shown in [11, 90] that the number of vertex systems may be reduced to 2^{2n_s}.

Quadratic stability and LQR control may be linked together in the so-called *guaranteed-cost* LQR control described below. Consider the system (4.39) and the standard quadratic cost

$$c(\Delta) = \int_0^\infty \big(x^T(t) Q_{xx} x(t) + u^T(t) Q_{uu} u(t)\big)\, \mathrm{d}t$$

with $Q_{xx} \succeq 0$ and $Q_{uu} \succ 0$. Then, for $\gamma > 0$ the objective is to find a state feedback law of the form

$$u(t) = -Q_{uu}^{-1} B_2^T P^{-1} x(t) \tag{4.42}$$

where the design matrix variable $P \succ 0$ is chosen so that the closed-loop system is quadratically stable and the cost

$$c(\Delta) \le \gamma^{-1} x_0^T P^{-1} x_0 \tag{4.43}$$

is guaranteed for all $\Delta \in \mathcal{B}_{\mathbb{D}}$.

The guaranteed-cost control problem can be easily formulated in terms of LMIs, see e.g. [68]. We present here an alternative characterization involving a quadratic matrix inequality (QMI).

Lemma 4.5 *Let $P \succ 0$ be a solution that simultaneously satisfies the QMIs*

$$A(\Delta)P + PA^T(\Delta) - 2B_2 Q_{uu}^{-1} B_2^T + \gamma\left(B_2 Q_{uu}^{-1} B_2^T + P Q_{xx} P\right) \prec 0 \qquad (4.44)$$

for all $\Delta \in \mathcal{B}_{\mathbb{D}}$. Then, the control law

$$u(t) = -Q_{uu}^{-1} B_2^T P^{-1} x(t)$$

quadratically stabilizes the system (4.31) and the cost

$$c(\Delta) \leq \gamma^{-1} x_0^T P^{-1} x_0$$

is guaranteed for all $\Delta \in \mathcal{B}_{\mathbb{D}}$.

The proof of this result is standard and it is not reported here; see e.g. [243, 322] for statements and proofs of similar results.

Remark 4.7 (Special cases of guaranteed-cost control) We notice that if $\gamma \to 0$, then the constraint (4.43) on the cost $c(\Delta)$ vanishes and the guaranteed-cost control reduces to a special case of quadratic stabilizability with the specific choice of a control law of the form (4.42). On the other hand, by setting $\gamma = 2$ and $Q_{xx} = 0$ in Eq. (4.44), we recover the quadratic stability setup.

4.3 Robust LMIs

As seen in the previous sections, several robust control problems can be formulated as the robust solution of general uncertain LMIs of the form

$$F(\Delta, \theta) \preceq 0, \quad \Delta \in \mathcal{B}_{\mathbb{D}} \qquad (4.45)$$

where the LMI matrices $F_i(\Delta) \in \mathbb{S}^n, i = 0, 1, \ldots, n_\theta$, are known functions of the uncertainty Δ. That is

$$F(\Delta, \theta) \doteq F_0(\Delta) + \sum_{i=1}^{n_\theta} \theta_i F_i(\Delta)$$

and $\Delta \in \mathcal{B}_{\mathbb{D}}$ and $\theta \in \mathbb{R}^{n_\theta}$.

Remark 4.8 (Robust optimization) In the optimization area, mathematical programming problems subject to an infinite number of constraints, such as (4.45), are usually referred to as *semi-infinite programs*, see e.g. [333]. In the last years, a new paradigm in this area termed *robust optimization*, has emerged. Pioneering studies in this field are [50, 158]. In particular, [158] deals with optimization problems subject to constraints of type (4.45), which are termed robust semidefinite programs. Determining an exact solution for these problems is computationally intractable in the general case.

4.4 Historical Notes and Discussion

In the early 1970s, a set-theoretic description of a plant family, often called unknown-but-bounded model [351], emerged as a novel paradigm for estimation and Kalman filtering. A few years later, researchers realized some drawbacks of optimal control, such as the lack of guaranteed margins of linear quadratic Gaussian (LQG) [142]. Subsequently, in the early 1980s, a successful alternative to the existing classical approach for control has been developed. In this new setting, the design objective is to determine feedback controllers that are guaranteed against all possible uncertainty realizations, i.e. *worst-case* (or *robust*) controllers, see [337] for an historical account on the history of robust control. In other words, a controller is designed with the aim of guaranteeing a specified performance for all plants that are compatible with the uncertainty description.

A major stepping stone in the robustness direction was the formulation in 1981 by Zames of the $\mathcal{H}_\infty$ problem [420]. In the subsequent fifteen years, the research in robust control evolved in various directions, each based on diverse problem formulations and mathematical tools. Even though several subareas played a major role within robustness, we feel that $\mathcal{H}_\infty$ deserves the credit for its centrality and also for its connections with classical optimal control. However, other successful methods to handle uncertainty have been developed. In particular, we recall the methods based on the structured singular value, also known as μ theory [314], the approach dealing with systems affected by parametric uncertainty, or Kharitonov theory [41, 58], the optimization-based methods based on linear matrix inequalities [68], the ℓ_1 optimal control theory [121] and the so-called quantitative feedback theory (QFT) [208, 209].

In the late 1980s, robust control became a well-known discipline so that the technical results and the algorithms developed were successfully used in various industrial applications, including aerospace, chemical, electrical and mechanical engineering. Moreover, the impact of robust control theory has begun to spread to other fields than engineering, such as economics [191].[4]

A few years later, in the early 1990s, researchers in robust control realized more fully some of its theoretical limitations, which can be roughly summarized as the issues of conservatism and computational complexity. In fact, when compared with classical stochastic methods, the worst-case paradigm may lead to problems whose exact solution cannot be determined in polynomial time, see e.g. [61]. Therefore, relaxation techniques are typically introduced so that the resulting problem can be solved with numerically efficient algorithms. Clearly, this entails a compromise between tightness of the solution and numerical complexity, as discussed in Chap. 5.

The brief presentation of robustness analysis and control techniques given in this and the previous chapter is necessarily incomplete. The $\mathcal{H}_\infty$ theory alone has been the subject of a variety of studies, each addressing different viewpoints. Quoting

[4] T.J. Sargent was awarded the Nobel Memorial Prize in Economic Sciences in 2011 together with C.A. Sims "for their empirical research on cause and effect in the macroeconomy."

Lanzon, Anderson and Bombois (see [253] and its previous versions), "Some control theorists may say that interpolation theory [416] is the essence of $\mathcal{H}_\infty$ control, whereas others may assert that unitary dilation [144] is the fundamental underlying idea of $\mathcal{H}_\infty$ control. Also, J-spectral factorization [236] is a well-known framework of $\mathcal{H}_\infty$ control. A substantial number of researchers may take differential game theory [34] as the most salient feature of $\mathcal{H}_\infty$ control and others may assert that the bounded real lemma is the most fundamental building block with its connections to LMI techniques. All these opinions contain some truth since some techniques may expose certain features of the problem that are hidden (to some degree) when using different viewpoints." The approach chosen in this chapter mainly focuses on LMI techniques, since this appears computationally attractive and quite flexible. For instance, multiobjective design with mixed $\mathcal{H}_2/\mathcal{H}_\infty$ performance specifications may be easily implemented in the LMI framework [345, 346, 375]. Also, robustness analysis based on *integral quadratic constraints* (IQCs) relies on LMI optimization [219, 280], as well as most of the recent results based on parameter-dependent Lyapunov functions [156].

In this chapter, we also presented well-known results related to *nominal* $\mathcal{H}_2$ control design. The issue of robust (or guaranteed-cost) $\mathcal{H}_2$ control is not included in this introductory exposition. The interested reader is referred for instance to the survey [315] and to [373, 415]. We remark that recently there has been a growing interest in solutions to robust control problems based on polynomially-constrained optimization. The interest towards these techniques has been revitalized by the introduction of approaches based on sum-of-squares (SOS) relaxations [105, 320], and other results of real algebraic geometry; see e.g. [254, 255]. In [349], a matrix-version of SOS representation is given to construct relaxations for computing upper bounds of robust optimization problem when uncertain parameters are constrained by polynomial matrix inequalities. These approaches systematically build an hierarchy of convex relaxations of the robust optimization problem. Such "lifted" problems provide a guaranteed—and hence conservative—solution to the original problem. The nice feature of these methods, that makes them preferable to other conservative approaches, is that under mild assumptions one can prove asymptotic convergence of the solutions of the convex relaxations to the solution of the original robust optimization problem as one increases the order of the relaxation.

Chapter 5
Limits of the Robustness Paradigm

In this chapter we discuss some limits of the classical robustness paradigm. In particular, we study complexity issues, conservatism and discontinuity problems. As outlined in Chaps. 3 and 4, stability and performance of control systems affected by bounded perturbations have been studied in depth in recent years. The attention of researchers and control engineers concentrated on specific descriptions of the uncertainty structures and related computational tools. Following the notation established in Sect. 3.6, we denote by Δ the uncertainty affecting a linear time-invariant system $M(s)$. In particular, Δ is generally assumed to belong to the structured set

$$\mathbb{D} = \left\{\Delta \in \mathbb{F}^{r,c} : \Delta = \text{bdiag}(q_1 I_{m_1}, \ldots, q_\ell I_{m_\ell}, \Delta_1, \ldots, \Delta_b)\right\}$$

where $q_1, \ldots, q_\ell$ represent real or complex uncertain parameters, possibly repeated with multiplicity $m_1, \ldots, m_\ell$, respectively, and $\Delta_1, \ldots, \Delta_b$ represent full blocks of appropriate dimensions. The structured matrix Δ is assumed to be bounded by a quantity ρ. That is, Δ belongs to the structured norm bounded set

$$\mathcal{B}_{\mathbb{D}}(\rho) = \left\{\Delta \in \mathbb{D} : \|q\|_p \leq \rho, \bar{\sigma}(\Delta_i) \leq \rho,\ i = 1, \ldots, b\right\}.$$

Then, we consider the family of uncertain systems obtained when the uncertainty Δ varies over $\mathcal{B}_{\mathbb{D}}(\rho)$ and we say that a certain property, e.g. stability or performance, is *robustly satisfied* if it is satisfied for all members of the family. As discussed in Sect. 3.7, the largest value of ρ so that stability or performance holds for all Δ is called the robustness margin or stability radius

$$r_{\mathbb{D}}(M) = \sup\left\{\rho : \text{stability (or performance) holds for all } \Delta \in \mathcal{B}_{\mathbb{D}}(\rho)\right\}.$$

The main objective of robustness analysis is to develop efficient algorithms for computing $r_{\mathbb{D}}(M)$ for various uncertainty structures $\mathbb{D}$. In Chap. 4, we presented some classical results for special classes of $\mathbb{D}$. In general, however, the problem of computing $r_{\mathbb{D}}(M)$ is known to be difficult from the computational point of view. This issue is addressed in the next section.

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_5,

5.1 Computational Complexity

The first critical issue of the classical robustness paradigm is computational complexity. In particular, various robust control problems have been shown to fall into the category of the so-called "intractable" problems, which are practically unsolvable if the number of variables becomes sufficiently large. These problems are generally denoted as *NP-hard*. In this section, we present introductory material on computational complexity and then discuss some robust control problems which fall into the category of NP-hard problems. The reader interested in more advanced material regarding decidability concepts and NP-completeness may refer to standard references such as [7, 176], and for specific results on continuous computational complexity to [388]. The material presented here is based on the overview [61] on computational complexity results in systems and control.

5.1.1 Decidable and Undecidable Problems

First, we study decision problems, i.e. problems for which the answer is either "yes" or "no." An example of a decision problem is checking whether a given matrix with integer entries is nonsingular. This question can be answered, for instance, by computing the determinant of the matrix and verifying if it is zero or not. This nonsingularity problem is known to be *decidable*, that is there exists an algorithm which always halts with the right "yes/no" answer. Unfortunately, there are many *undecidable* problems for which there is no algorithm that always halts with the right answer. We now provide an example of a decidable problem, see for instance [61].

Example 5.1 (Decidable problem) Consider a set of m multivariate polynomials $p_1(x), \ldots, p_m(x)$ with rational coefficients in n real variables $x = [x_1 \cdots x_n]^T$. The problem is to decide whether a solution x exists satisfying m polynomial equalities and inequalities of the form

$$\begin{aligned} p_i(x_1, \ldots, x_n) &= 0 \quad i = 1, \ldots, m_i; \\ p_k(x_1, \ldots, x_n) &> 0 \quad k = 1, \ldots, m_k; \\ p_\ell(x_1, \ldots, x_n) &\geq 0 \quad \ell = 1, \ldots, m_\ell \end{aligned} \tag{5.1}$$

where $m_i + m_k + m_\ell = m$. This problem is decidable.

A variation on this example is the following: to decide if there exist $x_1, \ldots, x_q \in \mathbb{R}$ such that (5.1) is satisfied for all $x_{q+1}, \ldots, x_n \in \mathbb{R}$. This problem is also decidable since it can be reduced to the previous one with the so-called Tarski–Seidenberg *quantifier elimination* (QE) method, see [352, 379]. On the other hand, there are many problems which are undecidable. Probably, the most famous one is the Hilbert's tenth problem on Diophantine equations, which is stated next.

Example 5.2 (Undecidable problem) Given an integer coefficient polynomial $p(x)$ in the variables $x_1, \ldots, x_n$, the goal is to decide if there exists an integer root. This problem has been shown to be undecidable in [278], see also [123] for an elementary exposition.

5.1.2 Time Complexity

Assuming that an algorithm halts, we define its *running time* as the sum of the "costs" of each instruction. In the so-called *random access machine* (RAM) model, each arithmetic operation involves a single instruction which is assumed to have a unit cost. More realistically, in the *bit* model, the cost of an arithmetic operation is given by the sum of the bit length of the integers involved in the operation, i.e. the running time of the algorithm depends on the size of the problem instance.

Since the running time may be different for different instances of the same size, we define the running time $T(n)$ as the worst-case running time over all instances of size n. Formally, we say that an algorithm runs in *polynomial time* if there exists an integer k such that its worst-case running time is[1]

$$T(n) = O\left(n^k\right).$$

We define P as the class of all decision problems having polynomial-time algorithms. In practice, the class P consists of all problems that are efficiently solvable. Notice that the notion of time complexity is associated with a specific algorithm and not with the problem itself. In other words, for the same problem, algorithms with different complexity may be derived.

An alternative definition of polynomial-time algorithms is related to the notion of "average" running time. An interesting example in this direction is the simplex method for solving linear programming problems. The complexity of the simplex method has been shown to be not polynomial-time in the worst case. However, it is known to have polynomial average running time. We will not discuss average complexity issues further, but we refer to [288, 350]. An example of a problem which is solvable in polynomial time is stability of a continuous-time system, which is presented next.

Example 5.3 (Polynomial-time test for Hurwitz stability) Given a monic polynomial of order n

$$p(s) = a_0 + a_1 s + a_2 s^2 + \cdots + s^n$$

we would like to ascertain if $p(s)$ is Hurwitz. This question can be easily answered using the Routh table. In particular, in [313] it is shown that the number of arithmetic

[1] The notation $O(\cdot)$ means the following: for functions $f, g : \mathbb{R}^+ \to \mathbb{R}^+$, we write $f(n) = O(g(n))$ if there exist positive numbers n_0 and c such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

operations (additions, multiplications and divisions) required by the Routh test is given by

$$\frac{n^2-1}{4} \quad \text{for } n \text{ odd} \quad \text{and} \quad \frac{n^2}{4} \quad \text{for } n \text{ even.}$$

Therefore, we conclude that Hurwitz stability has a polynomial-time solution and, moreover, that the number of operations required is $O(n^2)$. In [313], a detailed complexity analysis of tabular and determinant methods for continuous and discrete-time stability is also reported.

5.1.3 NP-Completeness and NP-Hardness

Unfortunately, for many decidable problems of interest, no polynomial-time algorithm is known. Many of these problems belong to a specific class of *nondeterministic polynomial time*, denoted as NP. A decision problem is said to belong to NP if the validity of a "yes" instance can be verified in polynomial time. Clearly, the class NP includes the class P. An example of a problem in the class NP is the so-called *satisfiability* problem (SAT) recalled next.

Example 5.4 (SAT problem) Consider the case of n binary variables $x_1, \ldots, x_n \in \{0, 1\}$ and a set of m linear equality and (strict) inequality constraints

$$\begin{aligned} c_i(x_1, \ldots, x_n) &= 0 \quad i = 1, \ldots, m_i; \\ c_k(x_1, \ldots, x_n) &> 0 \quad k = 1, \ldots, m_k \end{aligned} \tag{5.2}$$

with $m_i + m_k = m$. Notice that the sums and products in the constraints (5.2) can be interpreted as Boolean operations. The problem is to determine if a binary solution exists. Clearly, if we have a solution, then the "yes" instance can be verified in polynomial time, since it suffices to check if the conditions (5.2) are satisfied.

It has been shown, [112], that SAT is the hardest problem in the class NP. That is, every problem in the class NP can be reduced to SAT in polynomial time. More precisely, let $\mathcal{R}$ be any problem in NP. Then, given an instance $\mathcal{I}$ of $\mathcal{R}$, there exists a polynomial-time algorithm which provides an "equivalent" instance $\mathcal{I}'$ of SAT. The "equivalence" statement means that $\mathcal{I}$ is a "yes" instance of $\mathcal{R}$ if and only if $\mathcal{I}'$ is a "yes" instance of SAT. This means that, if a polynomial-time algorithm for SAT were found then every problem in NP would be solvable in polynomial time. For this reason, the SAT problem belongs to the class of *NP-complete* problems. In words, a problem is NP-complete if it is at least as difficult as any other problem in NP. NP-completeness has been established for many problems, see e.g. [176].

To elaborate further, suppose that a polynomial-time algorithm for SAT is known. In this case, any problem in NP could be solved in polynomial time, by first reducing it to SAT and then using the polynomial-time algorithm for SAT. The immediate consequence of this fact would be that the classes NP and P coincide. Unfortunately,

it is not known if a polynomial-time algorithm for SAT exists and, therefore, it is not known if P = NP, even though it is widely believed that P ≠ NP. The question of whether P is equal to NP is actually one of the main outstanding problems in theoretical computer science.

On the other hand, if a problem $\mathcal{R}$ is at least as hard as SAT (i.e. the SAT problem can be mapped into $\mathcal{R}$ in polynomial time), then we say that $\mathcal{R}$ is *NP-hard*. Therefore, a decision problem is NP-complete if and only if it is NP-hard and belongs to NP. In other words, if we could find an algorithm which runs in polynomial time for solving an NP-hard problem, we would have a polynomial-time solution for SAT. For this reason, NP-hardness is generally interpreted as "the problem is intractable."

We notice that there exist classes of problems which are harder than NP. For instance, consider the class EXP ⊃ NP ⊃ P of problems that can be solved in exponential time, i.e. with $O(2^{n^k})$ operations, for some integer k, where n is the problem size. Then, the hardest problems in this class are called EXP-complete. These problems are provably exponential time, i.e. P ≠ EXP, see [318].

In recent years, a number of systems and control problems have been shown to be NP-hard. In the next subsection, we describe some of them.

5.1.4 Some NP-Hard Problems in Systems and Control

In Sect. 3.7, we defined the structured singular value $\mu_{\mathbb{D}}$ of a matrix $M \in \mathbb{F}^{c,r}$ with respect to the uncertainty structure $\mathbb{D}$

$$\mu_{\mathbb{D}}(M) = \frac{1}{\min\{\bar{\sigma}(\Delta) : \det(I - M\Delta) = 0,\ \Delta \in \mathbb{D}\}}$$

unless no $\Delta \in \mathbb{D}$ makes $I - M\Delta$ singular, in which case $\mu_{\mathbb{D}}(M) = 0$. As previously discussed, the computation of $\mu_{\mathbb{D}}$ for general uncertainty structures is a difficult problem. More precisely, it has been shown that the problem of deciding whether $\mu_{\mathbb{D}}(M) \geq 1$ is NP-hard in the following cases:

1. *Real μ problem.* M is a real matrix and
$$\mathbb{D} = \left\{\Delta : \Delta = \text{bdiag}(q_1 I_{m_1}, \ldots, q_\ell I_{m_\ell}),\ q_i \in \mathbb{R}\right\};$$
2. *Mixed μ problem.* M is a complex matrix and
$$\mathbb{D} = \left\{\Delta : \Delta = \text{bdiag}(q_1 I_{m_1}, \ldots, q_\ell I_{m_\ell}),\ q_i \in \mathbb{F}\right\};$$
3. *Purely complex μ problem.* M is a complex matrix and
$$\mathbb{D} = \left\{\Delta : \Delta = \text{bdiag}(q_1 I_{m_1}, \ldots, q_\ell I_{m_\ell}),\ q_i \in \mathbb{C}\right\}.$$

The first two results are proven in [71], and the third is stated in [386]. Subsequent specific results regarding the complexity of approximating μ are not discussed here but are reported in [61].

We now turn our attention to the class of $n \times n$ interval matrices defined in (3.46), where each entry $[A]_{i,k}$ of the matrix is bounded in the interval $[a_{ik}^-, a_{ik}^+]$, for all

$i, k = 1, \ldots, n$. For this class, we are interested in establishing the complexity of various problems, including Hurwitz stability and nonsingularity. Clearly, stability can be detected by studying the characteristic polynomial. In the special case of interval matrices in controllability (observability) canonical form, with perturbations entering only in the last column (row), the characteristic polynomial is an interval polynomial with coefficients lying in independent intervals. Then, stability can be easily established through the Kharitonov theorem, see Sect. 3.8, which requires checking the stability of four specific polynomials. As discussed in Example 5.3, this test can be performed in $O(n^2)$ operations. However, similar results cannot be derived for stability of general interval matrices, and this problem is NP-hard. Similar negative complexity results have been established for different problems. More precisely, for the class of interval matrices, the following problems have been shown to be NP-hard:

1. *Stability.* Decide if all matrices in this class are Hurwitz;
2. *Nonsingularity.* Decide if all matrices in this class are nonsingular;
3. *Largest singular value.* Decide if all matrices in this class have spectral norm less than one;
4. *Positive definiteness.* Decide if all symmetric matrices in this class are positive definite.

These results have been established independently in [295, 323]. Related results for the nonsingularity problem are given in [115, 131].

Next, we turn our attention to the counterpart of stability of a family of matrices, namely the "existence" problem. It has been shown that establishing if there exists a Hurwitz matrix in the class of interval matrices is NP-hard. This problem is closely related to the well-known static output feedback problem: given state space matrices A, B and C, we are interested in determining if there exists a static feedback K such that $A + BKC$ is Hurwitz. Static output feedback has been shown to be NP-hard when lower and upper bounds on the entries of K are given, see e.g. [61]. This problem has also been reformulated in terms of decision methods, see [20]. A specific instance of the static output feedback problem is the so-called "one-in-a-box" problem, see [118]. More generally, reformulations of "difficult" control problems in terms of multivariate polynomial inequalities for the use of quantifier elimination (QE) methods have been addressed in various papers, see e.g. [139, 140]. However, such reformulations do not lead to a simplification, since QE problems have exponential complexity. In recent years, computationally more efficient approaches to optimization problems under multivariate polynomial inequalities have been developed in the context of sum-of-squares and moment-based techniques, see the discussion in Sect. 4.4.

To conclude this section on complexity in systems and control, we mention that branch-and-bound techniques are often used to solve problems which are apparently intractable, see e.g. [37, 127, 301]. These techniques seem to work reasonably well in practice, because the algorithms may run in polynomial time "on average," even though they are exponential time in the worst-case.

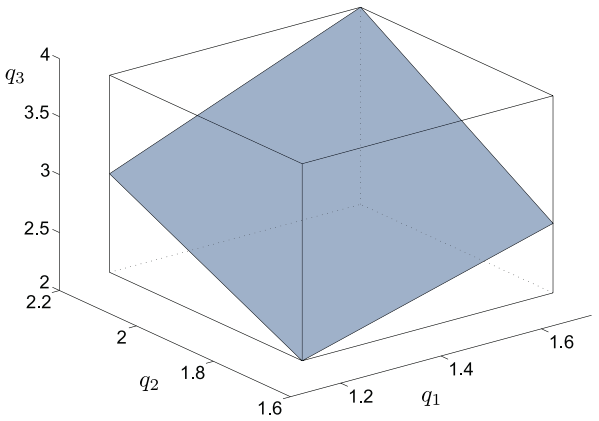

Fig. 5.1 Overbounding of the multiaffine uncertainty of Example 5.5

5.2 Conservatism of Robustness Margin

For real parametric uncertainty entering affinely into a control system, it is well known that the robustness margin can be computed exactly (modulo round-off errors). However, in real-world problems, the plant may be affected by nonlinear uncertainty. In many cases, this nonlinear uncertainty can be embedded into an affine structure by replacing the original family by a "larger" one. This process has the advantage of handling more general robustness problems, but it has the obvious drawback of giving only an approximate but guaranteed solution. Clearly, the quality of the approximation depends on the specific problem under consideration. This issue is analyzed in the next example.

Example 5.5 (Parameter overbounding and relaxation) To illustrate the overbounding methodology, consider the DC electric motor of Example 3.4 with two uncertain parameters $q_1 = R$ and $q_2 = J$. In this case, the closed-loop polynomial has the following multiaffine form

$$p(s,q) = K_m + \big(K_m^2 + Bq_1\big)s + (q_1q_2 + BL)s^2 + Lq_2s^3.$$

To overbound $p(s,q)$ with affine uncertainty, we set $q_3 = q_1q_2$. Given bounds $[q_1^-, q_1^+]$ and $[q_2^-, q_2^+]$ for q_1 and q_2, the range of variation $[q_3^-, q_3^+]$ for q_3 can be immediately computed

$$\begin{aligned} q_3^- &= \min\big\{q_1^-q_2^-, q_1^-q_2^+, q_1^+q_2^-, q_1^+q_2^+\big\};\\ q_3^+ &= \max\big\{q_1^-q_2^-, q_1^-q_2^+, q_1^+q_2^-, q_1^+q_2^+\big\}. \end{aligned}$$

Clearly, the new uncertain polynomial

$$\widetilde{p}(s,q) = K_m + \big(K_m^2 + Bq_1\big)s + (q_3 + BL)s^2 + Lq_2s^3$$

has three uncertain parameters q_1, q_2, q_3 entering affinely into $\widetilde{p}(s,q)$. Since $q_3 = q_1q_2$, this new parameter is not independent of q_1 and q_2, and not all values of $[q_3^-, q_3^+]$ are physically realizable. However, since we neglect this constraint and

assume that the parameters q_i are independent, this relaxation technique leads to an overbounding of $p(s,q)$ with $\widetilde{p}(s,q)$. In other words, if stability is guaranteed for $\widetilde{p}(s,q)$, then it is also guaranteed for $p(s,q)$, but the converse may not be true. Figure 5.1 illustrates the overbounding procedure for $q_1^- = 1.2$, $q_1^+ = 1.7$, $q_2^- = 1.7$, $q_2^+ = 2.2$, $q_3^- = 2.04$ and $q_3^+ = 3.74$. In this figure, the blue square represents the set of all physically realizable uncertainties, and the three-dimensional box shows its overbounding. The conservatism of the approach is evident.

To generalize the discussion in this example, we restate the overbounding problem for parametric uncertainty as follows: given an uncertain polynomial $p(s,q)$ with nonlinear uncertainty structure and a hyperrectangle $\mathcal{B}_q$, find a new uncertain polynomial $\widetilde{p}(s,q)$ with affine uncertainty structure and a new hyperrectangle $\widetilde{\mathcal{B}}_q$. In general, there is no systematic methodology to construct an "optimal" overbounding. The most natural way may be to compute an interval overbounding for each coefficient of the polynomial, i.e. an interval polynomial overbounding. To illustrate, letting $a_i(q)$, $i = 0, 1, \ldots, n-1$, denote the coefficients of $p(s,q)$, the lower and upper bounds are given by

$$a_i^- = \min_{q \in \mathcal{B}_q} a_i(q); \qquad a_i^+ = \max_{q \in \mathcal{B}_q} a_i(q).$$

If $a_i(q)$ are affine or multiaffine functions, these minimizations and maximizations can be easily performed. In fact, we have that

$$a_i^- = \min_{q \in \mathcal{B}_q} a_i(q) = \min_{k=1,\ldots,2^\ell} a_i\big(q^k\big);$$
$$a_i^+ = \max_{q \in \mathcal{B}_q} a_i(q) = \max_{k=1,\ldots,2^\ell} a_i\big(q^k\big)$$

where $q^1, \ldots, q^{2^\ell}$ are the vertices of the hyperrectangle $\mathcal{B}_q$. However, this is not true for more general uncertainty structures, for example if $a_i(q)$ are polynomial functions of q. In this case, a tight interval polynomial overbounding may be difficult to construct.

A technique frequently used to avoid the possible conservatism introduced by overbounding and relaxation techniques is the so-called *gridding approach*. Suppose we wish to establish if a polynomial $p(s,q)$ is Hurwitz when its coefficients are, for example, polynomial functions of $q \in \mathcal{B}_q$. In this case, we can take N equispaced points

$$q_i = q_i^- + k\frac{q_i^+ - q_i^-}{N-1}, \quad k = 0, 1, \ldots, N-1$$

for each parameter q_i, $i = 1, \ldots, \ell$, and check stability for every point in the grid. This procedure gives an "indication" of the robust stability of the system, which becomes more and more precise as the number of grid point increases. Clearly, the answer obtained by checking stability at the grid points does not provide any guarantee for the entire set $\mathcal{B}_q$. More importantly, the total number of grid points is $N_{\text{grid}} = N^\ell$, which is exponential. This exponential growth is often referred to

as the *curse of dimensionality*.[2] This fact clearly illustrates that the two issues of computational complexity and conservatism are indeed closely related and represent one of the main trade-offs in robustness analysis and design.

A similar situation arises in uncertain systems with nonparametric uncertainty described in M–Δ form. For example, for systems affected by more than one full real block or more than three full complex blocks, the computation of $\mu_{\mathbb{D}}$ (and therefore of the robustness margin) can be generally performed only at the expense of some conservatism. In fact, in these cases, only upper and lower bounds of $\mu_{\mathbb{D}}$ can be computed, see Sect. 3.7.3, but it may be difficult to estimate the degree of conservatism introduced. This issue is further addressed in the next example.

Example 5.6 (Conservatism in robustness margin computation) We consider an example concerning a five-mass spring–damper model with parametric uncertainty on the stiffness and damping parameters and dynamic uncertainty due to unmodeled dynamics analyzed in Sect. 19.5.

This flexible structure may be modeled as an M–Δ configuration, with $M(s) = C(sI - A)^{-1}B$. The matrix Δ is structured and consists of two repeated real parameters q_1, q_2 and one transfer matrix $\Delta_1 \in \mathcal{RH}_\infty^{4,4}$, i.e.

$$\Delta \in \widetilde{\mathbb{D}} = \left\{\Delta : \Delta = \text{bdiag}(q_1 I_5, q_2 I_5, \Delta_1)\right\}.$$

For this M–Δ system, $\mu_{\mathbb{D}}(M(j\omega))$ cannot be computed exactly, and the derivation of its upper and lower bounds requires the use of branch-and-bound techniques, which are computationally very expensive. Another approach, similar to the overbounding method previously studied, is to neglect the structure of Δ "lumping together" the uncertainty in one single block. In this case, an upper bound of the stability radius is immediately given by the small gain theorem studied in Sect. 3.7, which requires the computation of the $\mathcal{H}_\infty$ norm of $M(s)$. That is, we obtain the bound

$$r_{\mathbb{D}}\big(M(s)\big) \geq r_{\text{LTI}}\big(M(s)\big) = \frac{1}{\|M(s)\|_\infty}.$$

More generally, a wide class of robust synthesis problems can be recast in the form of nonlinear and nonconvex optimization problems, such as *bilinear matrix inequalities* (BMIs), see, e.g., [182], but such a reformulation does not solve the problem of efficiently deriving a controller. In fact, algorithms for finding global optimal solutions to BMI problems, based for instance on branch-and-bound techniques, have in general very high computational complexity, see comments in Sect. 5.1. On the other hand, also in this case, methods based on relaxations and overbounding have been proposed, at the expense of introducing a certain degree of conservatism.

[2] The frequently used terminology "curse of dimensionality" has been coined by Bellman in 1957 in his classical book on dynamic programming [49].

5.3 Discontinuity of Robustness Margin

There is another drawback inherent to the robustness paradigm that may arise even in the simple case when affine parametric uncertainty enters into the system. This is due to the fact that the robustness margin need not be a continuous function of the problem data. To show this phenomenon, we revisit a classical example [43] regarding the robustness margin of a system affected by linear parametric uncertainty.

Example 5.7 (Discontinuity of robustness margin) Consider a SISO plant of the form

$$G_\kappa(s,q) = \frac{N_\kappa(s,q)}{D_\kappa(s,q)}$$

where the numerator $N_\kappa(s,q)$ and denominator $D_\kappa(s,q)$ polynomials are given by

$$\begin{aligned} N_\kappa(s,q) &= 4\kappa^2 + 10\kappa^2 q_1; \\ D_\kappa(s,q) &= \kappa^2 + (20 + 8\kappa + 20\kappa q_1 - 20q_2)s + (44 + 2\kappa + 10q_1 - 40q_2)s^2 \\ &\quad + (20 - 20q_2)s^3 + s^4, \end{aligned}$$

where κ is a given parameter. The uncertain parameters vector $q = [q_1 q_2]^T$ varies within the hyperrectangle of radius ρ

$$\mathcal{B}_q(\rho) = \left\{ q \in \mathbb{R}^2 : \|q\|_\infty \leq \rho \right\}.$$

Taking a unit feedback $K(s) = 1$, we study robustness of the closed-loop polynomial

$$p_\kappa(s,q) = N_\kappa(s,q) + D_\kappa(s,q)$$

for "large" variations of the parameters q_1 and q_2 within $\mathcal{B}_q(\rho)$ and infinitesimal perturbations of κ. From this point of view, κ is considered as *problem data*, and not an uncertain parameter. Then, this example studies continuity properties of the robustness margin $r_q(\kappa)$ for small variations of κ.

More precisely, we study two cases. First, we consider the case where κ is fixed and set to $\kappa = \bar{\kappa} = 3 + 2\sqrt{2}$. Second, we consider a small perturbation $\epsilon > 0$ around κ, i.e.

$$\kappa(\epsilon) = \bar{\kappa} - \epsilon = 3 + 2\sqrt{2} - \epsilon.$$

Clearly, $\kappa(\epsilon) \to \bar{\kappa}$ as $\epsilon \to 0^+$. Then, we derive the closed-loop polynomials corresponding to these two cases, obtaining

$$\begin{aligned} p_{\bar{\kappa}}(s,q) &= \left(5\bar{\kappa}^2 + 10\bar{\kappa}^2 q_1\right) + (20 + 8\bar{\kappa} + 20\bar{\kappa} q_1 - 20q_2)s \\ &\quad + (44 + 2\bar{\kappa} + 10q_1 - 40q_2)s^2 + (20 - 20q_2)s^3 + s^4 \end{aligned} \tag{5.3}$$

and

$$p_{\kappa(\epsilon)}(s,q) = p_{\bar{\kappa}}(s,q) - 2\bar{\kappa}\epsilon(2 + 5q_1).$$

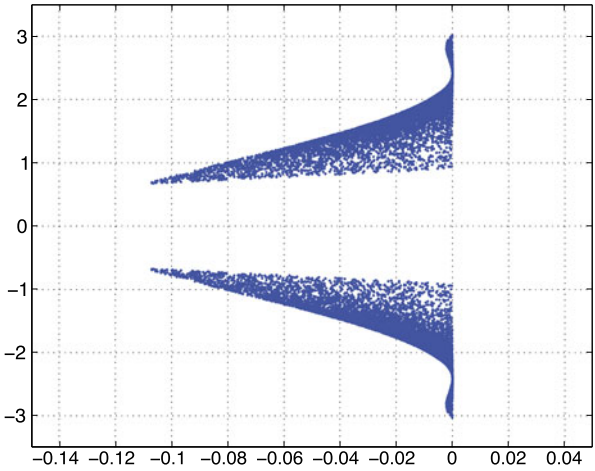

Fig. 5.2 Subset of the roots of $p_{\bar{\kappa}}(s,q)$, $q \in \mathcal{B}_q(\rho)$, $\rho = 0.417$, for Example 5.7

Subsequently, it can be verified that

$$0.417 \approx 1 - \frac{\bar{\kappa}}{10} = \lim_{\epsilon \to 0^+} r_q\big(\kappa(\epsilon)\big) > r_q(\bar{\kappa}) = \frac{7 - \bar{\kappa}}{5} \approx 0.234.$$

Various robustness interpretations may be given, and we redirect the interested reader to the original example in [43] and to further discussions in [41]. Here, we only mention that this example illustrates the phenomenon called the "false sense of security" of the robustness margin. That is, the closed-loop polynomial $p_{\kappa(\epsilon)}(s,q)$ is destabilized by an uncertainty approximately of size 0.234, and the robustness margin corresponding to the polynomial $p_{\bar{\kappa}}(s,q)$ is given by 0.417. In other words, infinitesimal variations around $\bar{\kappa}$ (which can be caused, for example, by numerical round-off), may lead to an overestimate of the robustness margin of about 78 %. This discontinuity phenomenon can also be illustrated by means of a plot of the roots of $p_{\bar{\kappa}}(s,q)$ when q ranges in a hyperrectangle of radius $\rho = 0.417$. It can be easily seen that some roots lie on the imaginary axis, see Fig. 5.2, which depicts a subset of the roots.

Chapter 6
Probabilistic Methods for Uncertain Systems

In this chapter, we introduce a *probabilistic approach* for analysis and design of uncertain systems. As pointed out in Chap. 5, many pessimistic results on the complexity-theoretic barriers of classical robust control have stimulated research in the direction of finding alternative approaches. One of these approaches, which constitutes the main subject treated in this book, proceeds by first shifting the meaning of robustness from its usual deterministic sense to a probabilistic one. In this respect, we shall claim that a certain system property is "almost robustly" satisfied if it holds for "most" of the instances of the uncertainty. In other words, we accept the risk of this property being violated by a set of uncertainties having small probability measure. Such systems may be viewed as being *practically robust*.

6.1 Performance Function for Uncertain Systems

In the robustness analysis framework discussed in Chap. 3, the main objective is to guarantee that a certain system property is attained for all uncertainties Δ bounded within a specified set $\mathcal{B}_{\mathbb{D}}$. To this end, it is useful to define a *performance function* (which is assumed to be measurable)

$$J(\Delta) : \mathbb{D} \to \mathbb{R}$$

where $\mathbb{D}$ is the uncertainty structured set defined in (3.27), and an associated performance level γ. In general, the function $J(\Delta)$ can take into account the simultaneous attainment of various performance requirements.

In the framework of robust synthesis studied in Chap. 4, the performance function depends also on some "design" parameters $\theta \in \mathbb{R}^{n_\theta}$ (e.g. the parameters of the controller), and takes the form

$$J(\Delta, \theta) : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \mathbb{R}.$$

Two examples showing specific performance functions are now given.

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_6,

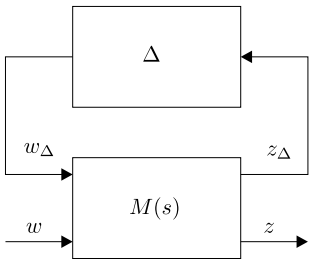

Fig. 6.1 M–Δ configuration of Example 6.1

Example 6.1 (Robust stability) Consider the feedback interconnection shown in Fig. 6.1, where $M(s)$ is a given transfer function and the uncertainty Δ belongs to the structured set $\mathbb{D}$ defined in (3.27). We are interested in the internal stability of this interconnection when the uncertainty $\Delta \in \mathbb{D}$ is such that $\|\Delta\|_\infty < \rho$. Consider a state space realization of the transfer function $M(s)$

$$M(s) = C(sI - A)^{-1}B + D,$$

where $A \in \mathbb{R}^{n_s,n_s}$ is stable, B, C, D are real matrices of suitable dimensions and $\Delta \in \mathcal{B}_\mathbb{D}$.

We assume that the well-posedness condition on D holds, i.e. $(I - D\Delta)$ is non-singular for all Δ. Then, from Lemma 3.4, internal stability of the system is equivalent to the matrix $A + B\Delta(I - D\Delta)^{-1}C$ being stable for all Δ belonging to the set $\mathbb{D}$ defined in (3.27) and with the bound $\bar{\sigma}(\Delta) < \rho$. Then, we choose the performance function $J(\Delta) : \mathbb{D} \to \mathbb{R}$ for robust stability of the M–Δ interconnection as

$$J(\Delta) = \begin{cases} 0 & \text{if } A + B\Delta(I - D\Delta)^{-1}C \text{ is stable;} \\ 1 & \text{otherwise.} \end{cases} \tag{6.1}$$

Setting the performance level for instance to $\gamma = 1/2$, robust stability is equivalent to checking if

$$J(\Delta) \leq \gamma$$

for all $\Delta \in \mathbb{D}$, $\bar{\sigma}(\Delta) < \rho$. If this check is satisfied for all Δ, we say that robust stability is guaranteed.

Example 6.2 ($\mathcal{H}_\infty$ performance) Consider again the feedback interconnection in Fig. 6.1. We now study the $\mathcal{H}_\infty$ norm of the transfer matrix between the disturbances w and the error z, defined in (3.29) as the upper linear fractional transformation $\mathcal{F}_u(M, \Delta)$. Given a level γ, we are interested in checking if the $\mathcal{H}_\infty$ performance is smaller than γ for all uncertainties Δ belonging to the structured bounded set $\mathcal{B}_\mathbb{D}$ defined in (3.28). Assuming that $\mathcal{F}_u(M, \Delta)$ is stable and strictly proper for all Δ, we set the performance function to

$$J(\Delta) = \left\|\mathcal{F}_u(M, \Delta)\right\|_\infty.$$

Then, if $J(\Delta) \leq \gamma$ for all Δ, we conclude that the performance level γ is robustly achieved.

This example can be generalized to the robust satisfaction of a generic *control system property*, as now stated.

Problem 6.1 (Robust performance verification) *Let Δ be bounded in the set $\mathcal{B}_{\mathbb{D}}$ defined in* (3.28). *Given a performance function $J(\Delta) : \mathbb{D} \to \mathbb{R}$ and associated level γ, check whether*

$$J(\Delta) \leq \gamma$$

for all $\Delta \in \mathcal{B}_{\mathbb{D}}$.

An associated problem is related to the computation of the so-called worst-case performance. That is, we are interested in evaluating the optimal guaranteed level of performance γ_{wc} such that $J(\Delta) \leq \gamma_{\text{wc}}$ for all $\Delta \in \mathcal{B}_{\mathbb{D}}$. This amounts to evaluating the supremum of $J(\Delta)$ when Δ ranges in the set $\mathcal{B}_{\mathbb{D}}$. This is shown in the next example.

Example 6.3 (Worst-case $\mathcal{H}_\infty$ performance) We now revisit Example 6.2. Suppose we are interested in determining the smallest level γ such that the $\mathcal{H}_\infty$ norm of the transfer matrix between the disturbances w and the error z is less than γ. As in Example 6.2, we set the performance function to

$$J(\Delta) = \left\| \mathcal{F}_u(M, \Delta) \right\|_\infty.$$

Then, the worst-case $\mathcal{H}_\infty$ performance is given by the supremum of $J(\Delta)$ computed with respect to $\Delta \in \mathcal{B}_{\mathbb{D}}$.

This example is now generalized to formally state the worst-case performance problem.

Problem 6.2 (Worst-case performance) *Let Δ be bounded in the set $\mathcal{B}_{\mathbb{D}}$ defined in* (3.28). *Given a performance function $J(\Delta) : \mathbb{D} \to \mathbb{R}$, compute*

$$\gamma_{\text{wc}} \doteq \sup_{\Delta \in \mathcal{B}_{\mathbb{D}}} J(\Delta).$$

A related problem, also discussed in Chap. 3, is to compute the largest uncertainty radius $r_{\mathbb{D}} = r_{\mathbb{D}}(M)$ such that robust performance is guaranteed for all uncertainties $\Delta \in \mathbb{D}$ with $\|\Delta\| < r_{\mathbb{D}}$.

Example 6.4 (Robust stability radius) Consider the setting of Example 6.1 regarding robust stability of the M–Δ interconnection. Suppose we are interested in evaluating the maximum radius $r_{\mathbb{D}}$ such that the system is robustly stable for all $\Delta \in \mathcal{B}_{\mathbb{D}}(r_{\mathbb{D}})$. From Theorem 3.5, it follows that

$$r_{\mathbb{D}}(M) = \frac{1}{\sup_{\omega \in \mathbb{R}} \mu_{\mathbb{D}}(M(j\omega))}$$

where $\mu_{\mathbb{D}}(M(j\omega))$ is the structured singular value defined in Sect. 3.7.2. Choosing the performance function $J(\Delta)$ as in (6.1) and setting $\gamma = 1/2$, this problem can be rewritten as

$$r_{\mathbb{D}} = \sup\{\rho : J(\Delta) \leq \gamma \text{ for all } \Delta \in \mathcal{B}_{\mathbb{D}}(\rho)\}$$

or equivalently as

$$r_{\mathbb{D}} = \inf\{\rho : J(\Delta) > \gamma \text{ for some } \Delta \in \mathcal{B}_{\mathbb{D}}(\rho)\}.$$

We recall that the computation of $\mu_{\mathbb{D}}(M(j\omega))$, and consequently $r_{\mathbb{D}}(M)$, is NP-hard and only upper and lower bounds of it are in general available.

In the following, we formally define the robust performance radius, which is a generalization of Definition 3.6.

Problem 6.3 (Robust performance radius) *Given a performance function* $J(\Delta) : \mathbb{D} \to \mathbb{R}$ *and associated level* γ, *compute*

$$r_{\mathbb{D}} \doteq \inf\{\rho : J(\Delta) > \gamma \text{ for some } \Delta \in \mathcal{B}_{\mathbb{D}}(\rho)\}$$

where the set $\mathcal{B}_{\mathbb{D}}(\rho)$ *is defined in* (3.28).

Remark 6.1 (Relationships between the robustness problems) The three robustness problems previously stated are closely related. In particular, if Problem 6.1 is solvable for fixed γ and ρ, then Problem 6.2 can be solved via a one-dimensional γ-iteration and Problem 6.3 can be solved via a one-dimensional ρ-iteration.

6.2 Good and Bad Sets

We now study the performance problems previously discussed by introducing two sets, denoted as the *good set* and the *bad set*, see also [84, 85]. These are subsets of $\mathcal{B}_{\mathbb{D}}$ and represent, respectively, the collection of all Δ which satisfy or violate the control system property under attention. These sets are constructed so that their union coincides with the uncertainty set $\mathcal{B}_{\mathbb{D}}$ and their intersection is empty. Formally, we define

$$\begin{aligned} \mathcal{B}_G &\doteq \{\Delta \in \mathcal{B}_{\mathbb{D}} : J(\Delta) \leq \gamma\}; \\ \mathcal{B}_B &\doteq \{\Delta \in \mathcal{B}_{\mathbb{D}} : J(\Delta) > \gamma\}. \end{aligned} \tag{6.2}$$

In the case of purely parametric uncertainty, we usually consider the uncertainty set $\mathcal{B}_q$ defined in (3.45) instead of $\mathcal{B}_{\mathbb{D}}$. Hence, the sets $\mathcal{B}_G$ and $\mathcal{B}_B$ take the form

$$\begin{aligned} \mathcal{B}_G &= \{\Delta \in \mathcal{B}_q : J(\Delta) \leq \gamma\}; \\ \mathcal{B}_B &= \{\Delta \in \mathcal{B}_q : J(\Delta) > \gamma\}. \end{aligned}$$

The set $\mathcal{B}_B$ is sometimes referred to as the *violation set*, see e.g. [85], since it represents the subset of uncertainties for which performance is violated. Robustness of a

Table 6.1 Routh table for Example 6.5

s^4	1	$6+3q_2$	$2+q_1+q_2$
s^3	$4+q_2$	$5+q_1+3q_2$	
s^2	$a_1(q)/(4+q_2)$	$2+q_1+q_2$	
s	$a_2(q)/a_1(q)$		
1	$2+q_1+q_2$		

control system is therefore equivalent to the case when the good set coincides with $\mathcal{B}_{\mathbb{D}}$ and the bad set is empty. We now present two examples showing the computation of $\mathcal{B}_G$ and $\mathcal{B}_B$.

Example 6.5 (Continuous-time stability of a fourth-order system) In this example, we consider stability of a fourth-order continuous-time system affected by a vector of two real uncertain parameters. In particular, we study a closed-loop monic polynomial of the form

$$p(s,q) = 2+q_1+q_2+(5+q_1+3q_2)s+(6+3q_2)s^2+(4+q_2)s^3+s^4.$$

In this case the structured set $\mathbb{D}$ coincides with $\mathbb{R}^2$ and the bounding set is the hyperrectangle

$$\mathcal{B}_q(\rho) = \left\{q \in \mathbb{R}^2 : \|q\|_\infty \leq \rho\right\}$$

with $\rho = 1.8$. Then, we introduce the performance function

$$J(q) = \begin{cases} 0 & \text{if } p(s,q) \text{ is Hurwitz;} \\ 1 & \text{otherwise} \end{cases} \tag{6.3}$$

and set $\gamma = 1/2$. The good set coincides with the set of Hurwitz stable polynomials

$$\mathcal{B}_G = \left\{q \in \mathcal{B}_q(\rho) : p(s,q) \neq 0 \text{ for all } \mathrm{Re}(s) \geq 0\right\}.$$

In order to obtain a closed-form characterization of the set $\mathcal{B}_G$, we construct the Routh table, shown in Table 6.1, which leads to the system of polynomial inequalities in q_1 and q_2

$$\begin{cases} 4+q_2 > 0; \\ a_1(q) = 19-q_1+15q_2+3q_2^2 > 0; \\ a_2(q) = (9+q_1+4q_2)\left(7-q_1+8q_2+2q_2^2\right) > 0; \\ 2+q_1+q_2 > 0. \end{cases}$$

These inequalities lead to the curves delimiting the Hurwitz region in parameter space shown in Fig. 6.2.

Example 6.6 (Robust $\mathcal{H}_2$ performance) Consider a continuous-time system expressed in state space form

$$\begin{aligned} \dot{x} &= A(q)x + Bw; \\ y &= Cx \end{aligned}$$

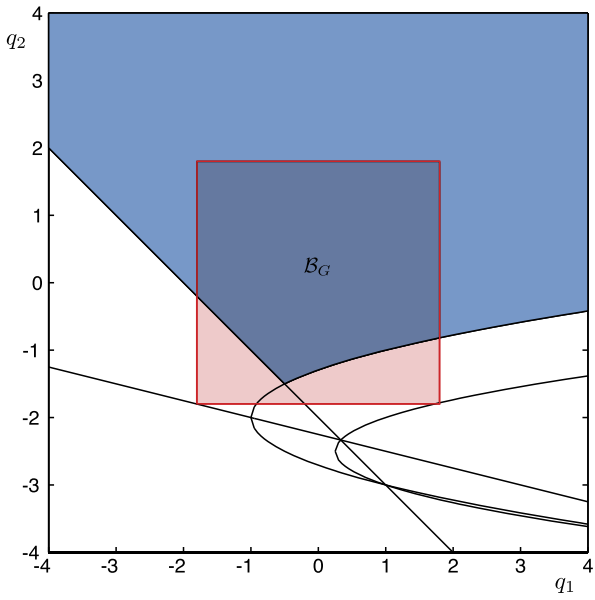

Fig. 6.2 Set of stable polynomials $\mathcal{B}_G$ for Example 6.5. The *light-blue region* indicates the Hurwitz region in parameter space

with

$$A(q) = \begin{bmatrix} -2+q_1 & q_1 q_2 \\ 0 & -4+q_2 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

The uncertainty is confined in the hyperrectangle

$$\mathcal{B}_q = \left\{ q \in \mathbb{R}^2 : \|q\|_\infty \leq 1 \right\}.$$

We are interested in checking if the squared $\mathcal{H}_2$ norm of the transfer function $G(s,q) = C(sI - A(q))^{-1}B$ between the disturbance w and the output z is less then $\gamma = 0.3$. Since the matrix $A(q)$ is upper triangular, it is easy to check that the system is stable for all values of the uncertainty, and therefore $G(s,q) \in \mathcal{H}_2$ for all $q \in \mathcal{B}_q$. Then, letting $J(q) = \|G(s,q)\|_2^2$ and $\gamma = 0.3$, we define the good set as

$$\mathcal{B}_G = \left\{ q \in \mathcal{B}_q : J(q) \leq \gamma \right\} = \left\{ q \in \mathcal{B}_q : \left\| G(s,q) \right\|_2^2 \leq 0.3 \right\}.$$

We can compute $\|G(s,q)\|_2^2 = \operatorname{Tr} C W_c C^T$, where the controllability Gramian $W_c \succeq 0$ is the solution of the Lyapunov equation

$$A(q) W_c + W_c A^T(q) + B B^T = 0.$$

For this simple case, straightforward but lengthy manipulations lead to

$$J(q) = -\frac{1}{2} \frac{q_1^2 q_2^2 - 2q_1 q_2(-4+q_2) + (-2+q_1)(-4+q_2) + (-4+q_2)^2}{(-4+q_2)(-2+q_1)^2 + (-2+q_1)(-4+q_2)^2}. \quad (6.4)$$

The level curve $J(q) = 0.3$ and the good set are depicted in Fig. 6.3.

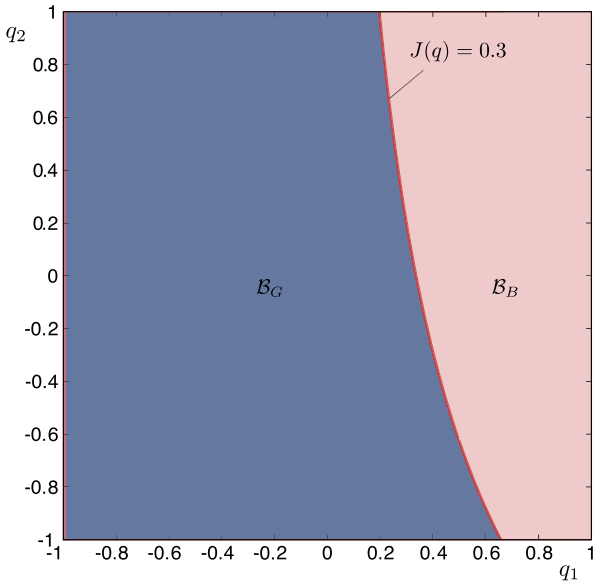

Fig. 6.3 Good and bad sets for Example 6.6

6.3 Probabilistic Analysis of Uncertain Systems

In classical robustness analysis, one of the main objectives, discussed in Problem 6.1, is to check if a given system property is satisfied for all possible values of the uncertainty. That is, for a given performance level γ, we would like to guarantee that

$$J(\Delta) \leq \gamma$$

for all $\Delta \in \mathcal{B}_{\mathbb{D}}$. This is equivalent to require that the sets $\mathcal{B}_G = \{\Delta \in \mathcal{B}_{\mathbb{D}} : J(\Delta) \leq \gamma\}$ and $\mathcal{B}_{\mathbb{D}}$ coincide.

In a probabilistic setting, a measure of robustness can instead be related to the relative volume of the set $\mathcal{B}_G$. In words, we require the volume of the good set to be "sufficiently large", i.e., that the ratio

$$\text{Vol}(\mathcal{B}_G)/\text{Vol}(\mathcal{B}_{\mathbb{D}}) \tag{6.5}$$

be "close" to one. More generally, we may assume that Δ is a random uncertainty with given density function, and we evaluate the degree of robustness as the probability measure of $\mathcal{B}_G$. We make the following fundamental standing assumption on the random nature of the uncertainty throughout the book.

Assumption 6.1 (Random uncertainty) *The uncertainty $\boldsymbol{\Delta}$ is a random matrix with density function $f_{\boldsymbol{\Delta}}(\Delta)$ and support $\mathcal{B}_{\mathbb{D}}$.*

Remark 6.2 (Existence of the pdf) To simplify the derivations and improve readability, the results in this book are derived under the hypothesis that the probability

density $f_{\boldsymbol{\Delta}}(\Delta)$ of $\boldsymbol{\Delta}$ exists. However, most of the results hold under the less restrictive assumption that $\boldsymbol{\Delta}$ only admits a distribution.

We recall that the performance function $J(\Delta) : \mathbb{D} \to \mathbb{R}$ is assumed to be measurable. With these assumptions, probabilistic robustness of a control system is stated in terms of the probability that the desired performance is satisfied. In other words, if Δ is a random matrix, then the volume in (6.5) becomes a "weighted" volume, where the weight is the given probability density function $f_{\boldsymbol{\Delta}}(\Delta)$. That is, the key quantity to be computed is the probability of performance satisfaction

$$\text{PR}\big\{J(\boldsymbol{\Delta}) \le \gamma\big\} = \int_{\mathcal{B}_G} f_{\boldsymbol{\Delta}}(\Delta)\,\mathrm{d}\Delta.$$

Clearly, if $f_{\boldsymbol{\Delta}}(\Delta)$ is the uniform density over $\mathcal{B}_{\mathbb{D}}$, then this probability is indeed the (normalized) volume of the good set

$$\text{PR}\big\{J(\boldsymbol{\Delta}) \le \gamma\big\} = \frac{\text{Vol}(\mathcal{B}_G)}{\text{Vol}(\mathcal{B}_{\mathbb{D}})}.$$

We are now ready to formulate the probabilistic counterpart of Problem 6.1.

Problem 6.4 (Probabilistic performance verification) *Given a performance function $J(\Delta)$ with associated level γ and a density function $f_{\boldsymbol{\Delta}}(\Delta)$ with support $\mathcal{B}_{\mathbb{D}}$, compute the* probability of performance

$$p(\gamma) \doteq \text{PR}\big\{J(\boldsymbol{\Delta}) \le \gamma\big\}. \tag{6.6}$$

The probability of performance $p(\gamma)$ measures the probability that a level of performance γ is achieved when $\boldsymbol{\Delta} \sim f_{\boldsymbol{\Delta}}(\Delta)$. We remark that this probability is in general difficult to compute either analytically or numerically, since it basically requires the evaluation of a multidimensional integral. However, in some special cases it can be evaluated in closed form, as shown in the following examples.

Example 6.7 (Probability of stability) We now revisit Example 6.5 regarding stability of a fourth-order system with closed-loop polynomial

$$p(s,q) = 2 + q_1 + q_2 + (5 + q_1 + 3q_2)s + (6 + 3q_2)s^2 + (4 + q_2)s^3 + s^4.$$

We now assume that q is a random vector with uniform distribution in the set $\mathcal{B}_q(\rho)$ with $\rho = 1.8$, i.e. $\mathbf{q} \sim \mathcal{U}_{\mathcal{B}_q(1.8)}$. Then, we set $J(q)$ as in (6.3) and $\gamma = 1/2$. In this case, the volume of the good set can be computed by integrating the equations defining the Hurwitz region derived in Example 6.5, obtaining

$$\text{Vol}(\mathcal{B}_G) = 10.026.$$

The probability of stability is then immediately given by

$$p(\gamma) = \text{PR}_{\mathbf{q}}\big\{p(s,\mathbf{q}) \text{ Hurwitz}\big\} = \frac{\text{Vol}(\mathcal{B}_G)}{\text{Vol}(\mathcal{B}_q(1.8))} = \frac{10.026}{12.96} = 0.7736.$$

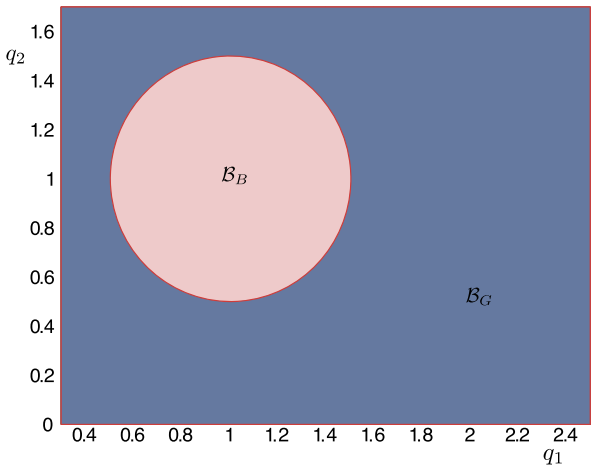

Fig. 6.4 Good and bad sets for Example 6.9

Example 6.8 (Probability of $\mathcal{H}_2$ performance) We revisit Example 6.6 regarding robust $\mathcal{H}_2$ performance of a continuous-time system expressed in state space form. We now assume that the uncertainty is random with

$$\mathbf{q} \sim \mathcal{U}_{\mathcal{B}_q}$$

and we aim at computing the probability of performance

$$p(\gamma) = \mathrm{PR}_{\mathbf{q}}\big\{J(\mathbf{q}) \le \gamma\big\} = \mathrm{PR}_{\mathbf{q}}\big\{\big\|G(s,\mathbf{q})\big\|_2^2 \le 0.3\big\}.$$

Using the expression of $J(q)$ obtained in (6.4), we compute the probability of performance integrating in closed form the level function $J(q) = 0.3$, obtaining the value $p(\gamma) = 0.6791$.

Example 6.9 (Probability of stability versus guaranteed stability) This example is due to Truxal [391], and it has been subsequently reconsidered in [5]. We study stability of a third-order continuous-time system affected by a vector of uncertainties q bounded in the set

$$\mathcal{B}_q = \big\{q \in \mathbb{R}^2 : 0.3 \le q_1 \le 2.5;\ 0 \le q_2 \le 1.7\big\}.$$

The closed-loop polynomial is a bilinear function on the uncertainty and is given by

$$p(s,q) = 1 + \alpha^2 + 6q_1 + 6q_2 + 2q_1q_2 + (q_1 + q_2 + 3)s + (q_1 + q_2 + 1)s^2 + s^3$$

where α varies in the interval $[0, 0.7]$. It can be easily verified that the set of unstable polynomials, the bad set, is a disk in parameter space

$$\mathcal{B}_B = \big\{q \in \mathcal{B}_q : (q_1 - 1)^2 + (q_2 - 1)^2 \le \alpha^2\big\}$$

with volume $\mathrm{Vol}(\mathcal{B}_B) = \pi\alpha^2$. The sets $\mathcal{B}_G$ and $\mathcal{B}_B$ are displayed in Fig. 6.4 for $\alpha = 0.5$.

Suppose now that the uncertainty is random with uniform density function over the set $\mathcal{B}_q$. Then, the probability of stability is

$$\text{Pr}_{\mathbf{q}}\{p(s,\mathbf{q}) \text{ Hurwitz}\} = 1 - \frac{\pi\alpha^2}{3.74}.$$

We notice that by taking $\alpha = 0$ the set of unstable polynomials becomes a singleton centered at $q_1 = 1$, $q_2 = 1$. In this case, the probability of stability is equal to one.

Example 6.9 shows some interesting features of the probabilistic approach. In fact, taking the parameter α equal to zero, we obtain that the system is stable with probability one. Of course, the system in *not* robustly stable, even though stability is violated only in a set of measure zero. This clearly shows the differences between the two settings.

Example 6.10 (Probability of Schur stability) In this example, we study Schur stability[1] of a discrete-time system affected by a vector q of parametric uncertainty bounded in the hyperrectangle $\mathcal{B}_q$. The closed-loop polynomial is a monic interval polynomial of order n

$$p(z,q) = q_0 + q_1 z + q_2 z^2 + \cdots + q_{n-1} z^{n-1} + z^n.$$

The set of coefficients $q \in \mathbb{R}^n$ leading to Schur stable polynomials (Schur region) is defined as

$$\mathcal{S}_n \doteq \{q \in \mathbb{R}^n : p(z,q) \text{ Schur}\}. \tag{6.7}$$

Consider now the case when $\mathbf{q}$ is a random vector with uniform density over $\mathcal{B}_q$ and introduce the performance function

$$J(q) = \begin{cases} 0 & \text{if } p(z,q) \text{ is Schur;} \\ 1 & \text{otherwise.} \end{cases} \tag{6.8}$$

Setting $\gamma = 1/2$, the probability that the discrete-time system is stable is

$$p(\gamma) = \text{Pr}_{\mathbf{q}}\{J(\mathbf{q}) \le \gamma\} = \text{Pr}_{\mathbf{q}}\{p(z,\mathbf{q}) \text{ Schur}\}.$$

That is, we define

$$\mathcal{B}_G = \{q \in \mathcal{B}_q : p(z,q) \text{ Schur}\} = \mathcal{B}_q \cap \mathcal{S}_n.$$

The volume of Schur stable polynomials

$$\text{Vol}(\mathcal{S}_n) = \int_{\mathcal{S}_n} \mathrm{d}q$$

can be explicitly computed by the recursive formulae given in [161]

$$\text{Vol}(\mathcal{S}_{n+1}) = \frac{\text{Vol}(\mathcal{S}_n)^2}{\text{Vol}(\mathcal{S}_{n-1})} \quad \text{for } n \text{ odd;} \tag{6.9}$$

$$\text{Vol}(\mathcal{S}_{n+1}) = \frac{n\text{Vol}(\mathcal{S}_n)\text{Vol}(\mathcal{S}_{n-1})}{(n+1)\text{Vol}(\mathcal{S}_{n-2})} \quad \text{for } n \text{ even} \tag{6.10}$$

where $\text{Vol}(\mathcal{S}_1) = 2$, $\text{Vol}(\mathcal{S}_2) = 4$ and $\text{Vol}(\mathcal{S}_3) = 16/3$.

[1] A polynomial $p(z)$ is Schur stable if $p(z) \neq 0$ for all $|z| \ge 1$.

Next, we remark that a polytope in coefficient space which is guaranteed to contain the Schur region $\mathcal{S}_n$ can be computed using the classical necessary conditions for Schur stability, see e.g. [273]

$$\begin{aligned} 0 &< p(1,q) < 2^n; \\ 0 &< (-1)^n p(-1,q) < 2^n; \\ |q_0| &< 1. \end{aligned} \tag{6.11}$$

Then, if this polytope is contained in the set $\mathcal{B}_q$, we conclude that condition $\mathcal{S}_n \subseteq \mathcal{B}_q$ is satisfied. In this case, the good set coincides with $\mathcal{S}_n$ and the probability of stability is immediately given in closed form as

$$p(\gamma) = \frac{\text{Vol}(\mathcal{S}_n)}{\text{Vol}(\mathcal{B}_q)}. \tag{6.12}$$

To illustrate, consider a fourth-order polynomial $p(z,q)$, whose coefficients lie in the hyperrectangle

$$\mathcal{B}_q = \left\{ q \in \mathbb{R}^4 : -1 \le q_0 \le 3;\ -5 \le q_1 \le 5;\ -3 \le q_2 \le 6;\ -4 \le q_3 \le 4 \right\}.$$

Using conditions (6.11), it can be checked that $\mathcal{S}_n \subseteq \mathcal{B}_q$ holds. Then, the probability of stability is given by

$$p(\gamma) = \frac{\text{Vol}(\mathcal{S}_4)}{\text{Vol}(\mathcal{B}_q)} = \frac{64/9}{2880} = 0.0025.$$

However, if $\mathcal{S}_n \not\subseteq \mathcal{B}_q$, Eq. (6.12) is no longer valid and other methods should be devised for computing $p(\gamma)$ exactly.

We now consider the second problem presented Sect. 6.1, related to the computation of the worst-case performance, and introduce its probabilistic counterpart.

Problem 6.5 (Probabilistic worst-case performance) *Given a performance function $J(\Delta)$, a density function $f_{\boldsymbol{\Delta}}(\Delta)$ with support $\mathcal{B}_{\mathbb{D}}$ and a probability level $\epsilon \in (0,1)$, compute $\bar{\gamma}$ such that*

$$\bar{\gamma} \le \gamma_{\text{wc}} = \sup_{\Delta \in \mathcal{B}_{\mathbb{D}}} J(\Delta)$$

and

$$\text{Pr}\left\{ J(\boldsymbol{\Delta}) \le \bar{\gamma} \right\} \ge 1 - \epsilon. \tag{6.13}$$

Remark 6.3 (Good and bad sets interpretation of Problem 6.5) Equation (6.13) can be interpreted in terms of good and bad sets. That is, if we define $\mathcal{B}_G = \{\Delta \in \mathcal{B}_{\mathbb{D}} : J(\Delta) \le \bar{\gamma}\}$, we write this equation as

$$\text{Pr}\{\boldsymbol{\Delta} \in \mathcal{B}_G\} \ge 1 - \epsilon.$$

In terms of a bad set, defining $\mathcal{B}_B = \{\Delta \in \mathcal{B}_{\mathbb{D}} : J(\Delta) > \bar{\gamma}\}$, we have

$$\text{Pr}\{\boldsymbol{\Delta} \in \mathcal{B}_B\} < \epsilon.$$

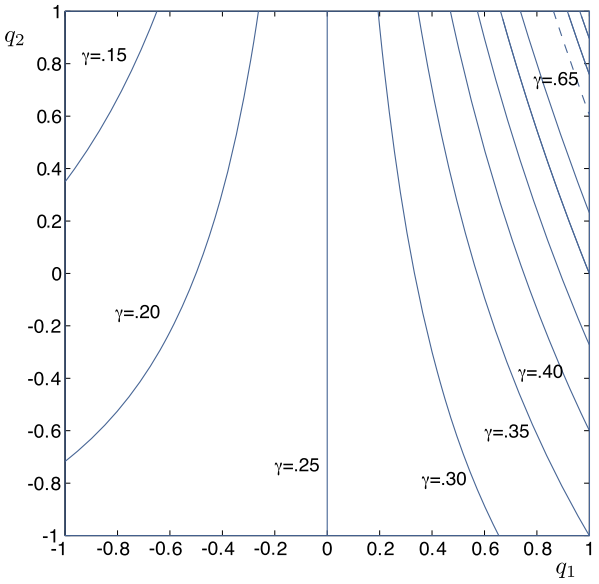

Fig. 6.5 Level curves $J(q) = \gamma$ for different values of γ for Example 6.11. The *dashed line* corresponds to $\gamma = 0.65$

In other words, we accept a ϵ-risk that the performance is violated, i.e.

$$\text{PR}\left\{J(\boldsymbol{\Delta}) > \bar{\gamma}\right\} < \epsilon.$$

The probability $\text{PR}\{J(\boldsymbol{\Delta}) > \bar{\gamma}\}$ is sometimes called the *probability of violation*, see e.g. [79, 85] and Sect. 10.4. For uniform densities, this amounts to requiring that

$$\text{Vol}(\mathcal{B}_B) < \epsilon \text{Vol}(\mathcal{B}_{\mathbb{D}}).$$

Computation of the probabilistic worst-case performance is illustrated in the next example.

Example 6.11 (Probabilistic worst-case $\mathcal{H}_2$ performance) Consider the state space representation of Example 6.6 and suppose we are interested in computing a level of performance $\bar{\gamma}$ such that

$$\text{PR}_{\mathbf{q}}\left\{J(\mathbf{q}) \leq \bar{\gamma}\right\} \geq 0.99.$$

To this aim, we repeat the procedure outlined in Example 6.8, and compute the corresponding probability of performance for different values of γ. In Fig. 6.5 we report the level curves $J(q) = \gamma$, which were used to compute $p(\gamma)$. Using this procedure, we obtained that the level $\gamma = 0.65$ guarantees a probability $\text{PR}_{\mathbf{q}}\{J(\mathbf{q}) \leq 0.65\} = 0.9931$, which is greater than the desired level $1 - \epsilon = 0.99$.

We now turn our attention to the third problem introduced in Sect. 6.1 and state its probabilistic version.

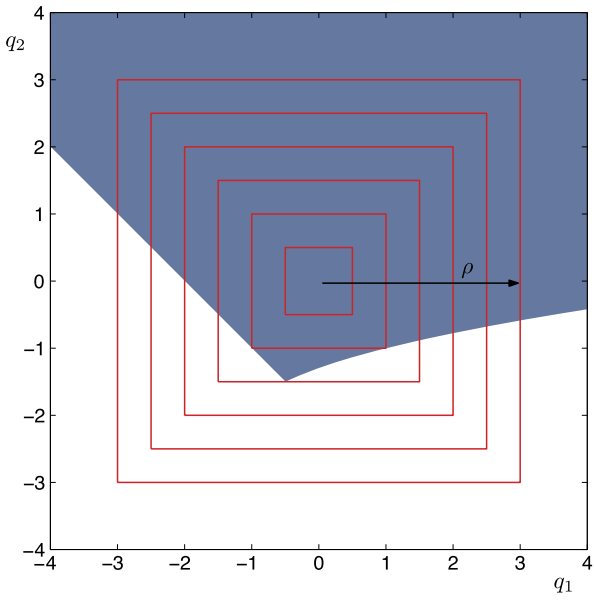

Fig. 6.6 The set $\mathcal{B}_q(\rho)$, $\rho \in [0.5, 3]$ for Example 6.12. The *blue region* indicates the Hurwitz region

Problem 6.6 (Probabilistic performance radius) *Given a performance function* $J(\Delta)$ *with associated level* γ *and a probability level* $\epsilon \in (0, 1)$, *compute* $\bar{r}(\epsilon)$ *as*

$$\bar{r}(\epsilon) \doteq \inf\big\{\rho : \text{PR}\big\{J(\boldsymbol{\Delta}) \leq \gamma\big\} < 1 - \epsilon,\ \boldsymbol{\Delta} \in \mathcal{B}_{\mathbb{D}}(\rho),\ \boldsymbol{\Delta} \sim f_{\boldsymbol{\Delta}}(\Delta)\big\}.$$

We now further elaborate on this problem: for given radius ρ, we assume that $\boldsymbol{\Delta}$ is a random matrix with support $\mathcal{B}_{\mathbb{D}}(\rho)$ and with given density function $f_{\boldsymbol{\Delta}}(\Delta)$, also depending on ρ. Then, we define the *performance degradation function* as

$$\text{degrad}(\rho) \doteq \text{PR}\big\{J(\boldsymbol{\Delta}) \leq \gamma\big\}, \quad \boldsymbol{\Delta} \in \mathcal{B}_{\mathbb{D}}(\rho),\ \boldsymbol{\Delta} \sim f_{\boldsymbol{\Delta}}(\Delta). \tag{6.14}$$

Then, we conclude that the probabilistic performance radius becomes

$$\bar{r}(\epsilon) \doteq \inf\big\{\rho : \text{degrad}(\rho) < 1 - \epsilon\big\}.$$

We now present an example showing the computation of the performance degradation function.

Example 6.12 (Performance degradation function) We revisit again Example 6.5, regarding stability of a fourth-order system. We now assume that q is a random vector with uniform distribution in the set $\mathcal{B}_q(\rho)$. Setting $J(q)$ and γ as before, we evaluate the probability of stability when ρ varies in the interval $[0.5, 3]$, as depicted in Fig. 6.6.

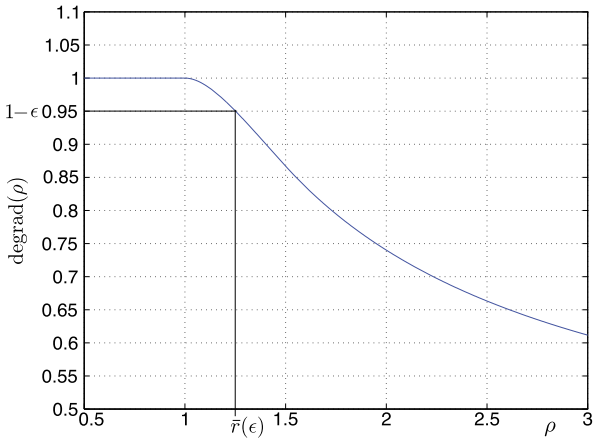

Fig. 6.7 Performance degradation function for Example 6.12

As in the previous case, the volume of the good set can be computed in closed form by integrating the equations defining the Hurwitz region

$$\mathrm{Vol}\big(\mathcal{B}_G(\rho)\big) = \begin{cases} 4\rho^2 & \text{if } \rho \le 1; \\ \frac{2}{3}\rho^3 - 3\rho^2 - (\alpha(\rho) - 13)\rho - \alpha(\rho) - \frac{16}{3} & \text{if } 1 < \rho \le 1.5; \\ \frac{3}{2}\rho^2 + 4\rho + \frac{7}{24} - \frac{9}{2}\alpha(\rho)^3 & \text{if } 1.5 < \rho \le 3 \end{cases}$$

where $\alpha(\rho) = \frac{1}{3}\sqrt{2\rho + 2}$. Then, the performance degradation function is immediately obtained as

$$\mathrm{degrad}(\rho) = \frac{\mathrm{Vol}(\mathcal{B}_G(\rho))}{\mathrm{Vol}(\mathcal{B}_q(\rho))}$$

which is shown in Fig. 6.7. Fixing a level of probability $\epsilon = 0.05$, from this figure we obtain that the probabilistic stability radius is given by

$$\bar{r}(0.05) = 1.2514.$$

Finally, we remark that, in this example, the performance degradation function is monotonically decreasing.

As a final tutorial example, we report an illustrative problem of $\mathcal{H}_\infty$ performance first introduced in [85].

Example 6.13 (Probabilistic $\mathcal{H}_\infty$ performance) Consider the linear system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -a_0(q) & -a_1(q) \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w; \quad z = [\,1 \quad 0\,]x \tag{6.15}$$

with parameters

$$a_0(q) = \bar{a}_0 + q_0, \qquad a_1(q) = \bar{a}_1 + q_1$$

and uncertainty vector $q \doteq [q_0 \;\; q_1]^T$ that belongs to the set

$$\mathcal{B}_q(\rho) = \big\{ q \in \mathbb{R}^2 : \|q\|_\infty \le \rho \big\}$$

for some positive ρ (i.e. $|q_0| \le \rho$, $|q_1| \le \rho$), and nominal values $\bar{a}_0 = 1$, $\bar{a}_1 = 0.8$. Suppose that we are interested in computing the peak of the modulus of the frequency response on the w–z channel. When the system is stable, this peak is given by the $\mathcal{H}_\infty$ norm of the transfer function $G(s,q)$ of this channel, $\|G(s,q)\|_\infty = \sup_\omega |G(j\omega,q)|$, see Sect. 3.3. Given a level $\gamma \ge 1$, a (deterministic) robustness analysis problem requires, for instance, to verify whether the system is stable and the performance level $\|G(s,q)\|_\infty \le \gamma$ is guaranteed for all $q \in \mathcal{B}_q(\rho)$. That is, this is equivalent to verify if the specification "$G(s,q)$ *is stable and* $\|G(s,q)\|_\infty \le \gamma$" is satisfied for all $q \in \mathcal{B}_q(\rho)$. To this end, we introduce the performance function

$$J(q) = \begin{cases} \infty & \text{if } G(s,q) \text{ is unstable;} \\ \|G(s,q)\|_\infty & \text{otherwise.} \end{cases} \tag{6.16}$$

Let for instance $\gamma = \sqrt{2}$. Since (6.15) is in companion form, it can be immediately seen that the closed-loop transfer function is of interval form

$$G(s,q) = \frac{1}{s^2 + [1-\rho, 1+\rho]s + [0.8-\rho, 0.8+\rho]},$$

so that vertex-type conditions can be used to determine the Bode envelope, and thus to evaluate the worst-case $\mathcal{H}_\infty$ norm, see e.g. [41]. Moreover, in this simple example, the system is of second order, and thus stability and performance can be assessed by direct calculation. It is immediate to verify that $J(q) \le \gamma$ is satisfied for all $q \in \mathcal{B}_q(\rho)$ if and only if

$$\rho < 0.8, \tag{6.17}$$

$$\frac{(0.8-\rho)^2}{2-\sqrt{2}} > 1+\rho \quad \Rightarrow \quad \rho < \bar{\rho} \doteq 0.025, \tag{6.18}$$

where (6.17) implies robust stability, while (6.18) implies $\|G(s,q)\|_\infty \le \gamma$, see Fig. 6.8, where $\bar{\rho}$ is the *deterministic radius of performance* for the uncertain system (6.15).

Notice that, from the point of view of worst-case approach, it is not possible to go beyond the uncertainty level $\bar{\rho}$, since after this level the system ceases to satisfy the required robust performance specification. In this sense, worst-case analysis provides a yes/no answer, and gives no information on the system behavior for uncertainty level beyond $\bar{\rho}$.

Probabilistic analysis can nicely complement the information provided by the deterministic approach, or can be a valid alternative when the deterministic approach cannot be applied. To this end, assume for instance that $\mathbf{q}_0$, $\mathbf{q}_1$ are random variables, independent and uniformly distributed in the interval $[-\rho, \rho]$. Then, for $\rho > \bar{\rho}$, some uncertainty realizations violate the given specifications. However, in practice, a violation might be tolerated, if it happens "rarely." To quantify how rare is the violation, we use probabilities. More precisely, with reference to Fig. 6.9, we admit a bad set where performance is violated.

In the simple example at hand, the degradation function can be computed explicitly for various values of ρ, and the curve shown in Fig. 6.10 can be traced. We can

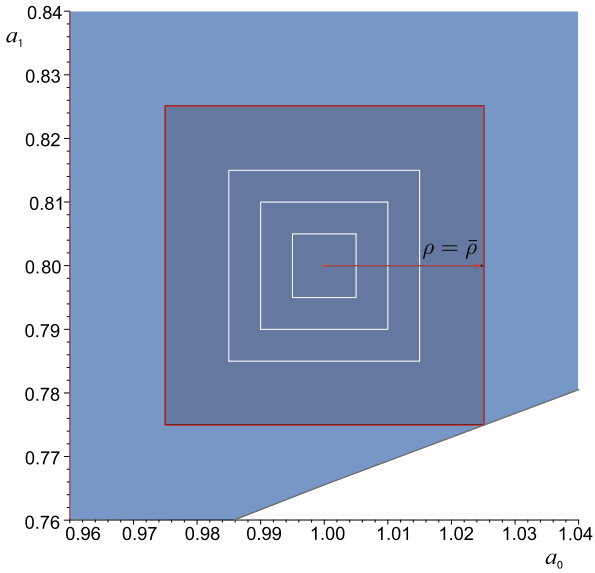

Fig. 6.8 The *light-blue region* contains coefficients values for which the system in Example 6.13 is stable and $\|G(s,q)\|_\infty \leq \sqrt{2}$ for all $q \in \mathcal{B}_q(\rho)$. The square of radius $\bar{\rho}$ is the maximal admissible uncertainty set around the nominal parameters values $\bar{a}_0, \bar{a}_1$

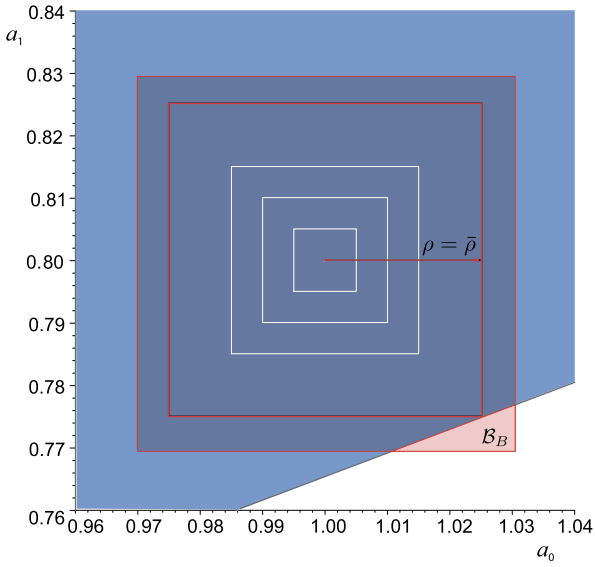

Fig. 6.9 The uncertainty radius in Example 6.13 is extended beyond $\bar{\rho}$ and a subset of $\mathcal{B}_q(\rho)$ violates the performance specifications (*red area*)

infer useful information from this curve: for instance, we see that if a 5 % probability of violation is allowed, then the tolerable uncertainty radius can be increased by about 54 % with respect to the deterministic radius $\bar{\rho}$.

Example 6.14 (Probabilistic stability radius) We revisit Example 5.7 regarding discontinuity of the robustness margin and study Hurwitz stability for $\bar{\kappa} = 3 + 2\sqrt{2}$ of the closed-loop polynomial $p_{\bar{\kappa}}(s,q)$ in (5.3). In Example 5.7, the stability radius

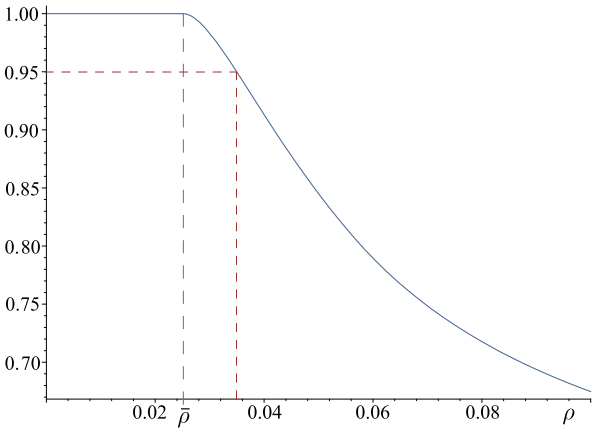

Fig. 6.10 Degradation of the reliability level as a function of the uncertainty radius ρ in Example 6.13

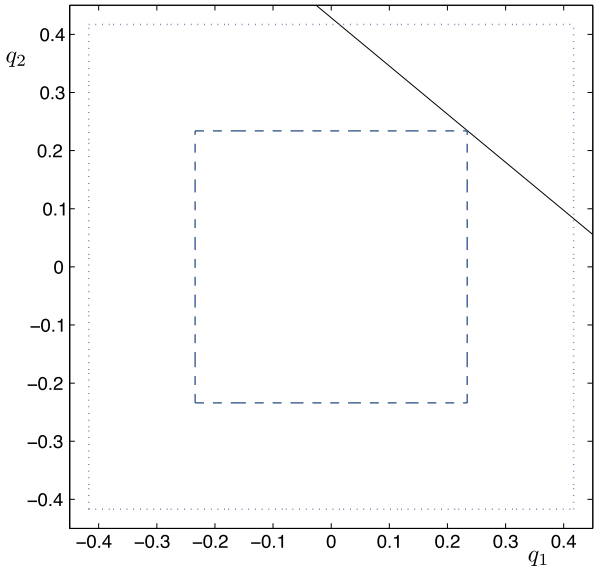

Fig. 6.11 Set of unstable polynomials in the box of radius $\rho = 0.417$ (*dotted line*), and box of radius $\rho = r_q(\bar{\kappa})$ (*dashed line*)

of the polynomial is computed, obtaining $r_q(\bar{\kappa}) = 0.234$. Following a probabilistic approach, we now take the uncertain parameters q_1 and q_2 as random variables with uniform probability density on $\mathcal{B}_q$ and study stability degradation as ρ varies between $\rho = 0.234$ and $\rho = 0.417$.

Using the Routh table, and by means of lengthy computations, we conclude that the stability boundary in parameter space is given by the line segment

$$\left\{ q \in \mathcal{B}_q : \frac{1+\sqrt{2}}{2} q_1 + q_2 = \frac{4-\sqrt{2}}{5} \right\}.$$

This line is tangent to the box of radius $\rho = r_q(\bar{\kappa})$, see the plot of Fig. 6.11.

It follows that the set of unstable polynomials $\mathcal{B}_B$ corresponding to this line is a set of measure zero. Therefore, the ratio $\text{Vol}(\mathcal{B}_B)/\text{Vol}(\mathcal{B}_q)$ is equal to zero and the probability degradation function remains equal to one up to $\rho = 0.417$, i.e.

$$\text{degrad}(\rho) = 1, \quad \text{for all } \rho \in [0, 0.417].$$

Notice that for $\kappa = \bar{\kappa} - \epsilon$ the diagonal line segment in Fig. 6.11 "disappears," giving rise to a discontinuity in the (deterministic) stability radius. This discontinuity is not present in its probabilistic counterpart.

Remark 6.4 (Relationships between probabilistic problems) Similarly to the robustness problems studied in Sect. 6.1 (see Remark 6.1), the three probabilistic problems stated in this section are closely related. In particular, if Problem 6.4 is solvable for fixed γ and ρ, then Problem 6.5 can be solved via a one-dimensional γ iteration and Problem 6.6 can be solved via a one-dimensional ρ iteration. However, the probabilistic interpretation of these problems and the specific results obtained may be different.

Remark 6.5 (Closed-form computation) The examples reported in this chapter show the closed-form computation of the various probabilistic quantities under study. Obviously, these exact computations are limited to special cases. The solution of these problems requires the evaluation of multidimensional integrals, which is in general a very difficult task. In subsequent chapters we develop randomized algorithms based on uncertainty sampling to obtain estimates of the required probabilities, up to a certain level of confidence.

6.4 Distribution-Free Robustness

In the probabilistic setting described in this chapter, the density function of the uncertainty $\boldsymbol{\Delta}$ is assumed to be known. If this is not the case, then clearly the probability of performance depends on the specific choice of $f_{\boldsymbol{\Delta}}(\Delta)$. For example, in the previous section we have shown that the probability $p(\gamma) = \Pr\{J(\boldsymbol{\Delta}) \leq \gamma\}$ coincides with the ratio of volumes $\text{Vol}(\mathcal{B}_G)/\text{Vol}(\mathcal{B}_{\mathbb{D}})$ if the pdf is uniform. For a different pdf, the probability $p(\gamma)$ may be dramatically different. In other words, without some reasoning attached to the selection of the probability density, the probability of performance obtained may be meaningless. Generally, the probability density function may be estimated directly from available data or prior information, but if this prior information is not available, then the selection of the distribution should be performed with great care.

To address this problem further we now consider an example.

Example 6.15 (Probabilistic stability for various distributions) We now continue Example 6.9. Consider now the case when $\alpha = 0.1$ and the density function $f_{\mathbf{q}}(q)$ is a truncated Gaussian pdf with expected value $\text{E}(\mathbf{q}) = [1\ 1]^T$, covariance matrix

Table 6.2 Probability of stability and constant C for decreasing values of σ

σ	C	$\text{PR}\{p(s,\mathbf{q})$ Hurwitz$\}$
2	0.2955	0.9907
0.4	1.0199	0.9664
0.2	3.9789	0.8824
0.1	15.916	0.6065
0.01	1591.6	$\to 0$

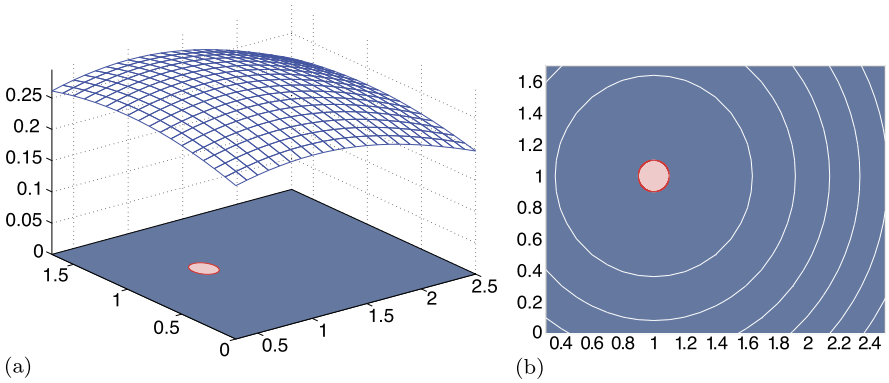

Fig. 6.12 (**a**) Gaussian pdf centered in $[1\ 1]^T$ with $\sigma = 2$. (**b**) Level curves of the distribution. The disk of radius 0.1 represents the subset of unstable parameters $\mathcal{B}_B$

$\text{Cov}(\mathbf{q}) = \text{diag}([\sigma^2\, \sigma^2])$ and support $\mathcal{B}_q = \{q \in \mathbb{R}^2 : 0.3 \le q_1 \le 2.5;\ 0 \le q_2 \le 1.7\}$. That is, we write

$$f_{\mathbf{q}}(q) = \begin{cases} C\mathrm{e}^{-\frac{(q_1-1)^2+(q_2-1)^2}{2\sigma^2}} & \text{if } q \in \mathcal{B}_q; \\ 0 & \text{otherwise} \end{cases}$$

where C is a normalizing constant obtained by imposing $\int_{\mathcal{B}_q} f_{\mathbf{q}}(q)\,\mathrm{d}q = 1$. In this example we compute in closed form the probability of stability by solving explicitly the multiple integral required to compute the probability

$$\text{PR}_{\mathbf{q}}\big\{p(s,\mathbf{q}) \text{ Hurwitz}\big\} = 1 - \int_{\mathcal{B}_B} f_{\mathbf{q}}(q)\,\mathrm{d}q = 1 - 2\pi\sigma^2\big(1 - \mathrm{e}^{-\frac{\alpha^2}{2\sigma^2}}\big)$$

where $\mathcal{B}_B = \{q \in \mathcal{B}_q : (q_1-1)^2 + (q_2-1)^2 \le \alpha^2\}$.

Table 6.2 shows how different values of σ lead to very different values of the probability of stability. This behavior is also shown in Figs. 6.12 and 6.13. When the value of σ decreases, the Gaussian pdf shrinks around the bad set. In the limit case $\sigma \to 0$, the probability of stability approaches zero, whereas for $\sigma \to \infty$ the Gaussian distribution tends to the uniform distribution and the probability of stability approaches the ratio $\text{Vol}(\mathcal{B}_G)/\text{Vol}(\mathcal{B}_q(\rho))$. The conclusion is that if the density

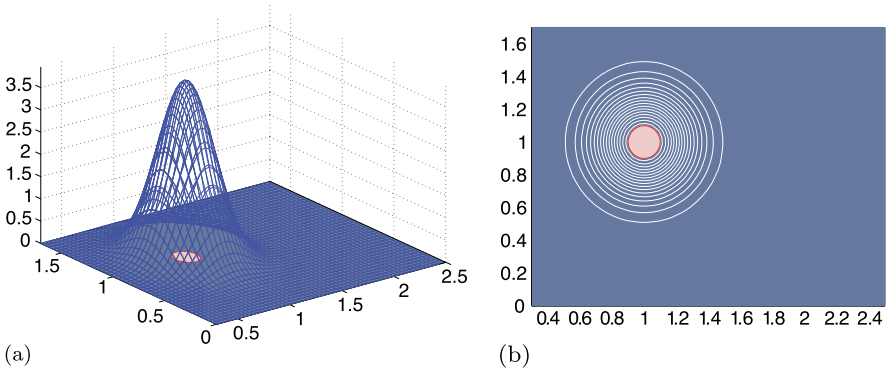

Fig. 6.13 (**a**) Gaussian pdf centered in $[1\,1]^T$ with $\sigma = 0.2$. (**b**) Level curves of the distribution. The disk of radius 0.1 represents the subset of unstable parameters $\mathcal{B}_B$

function is chosen without any specific guideline, then the probability of stability may vary arbitrarily between the extreme values zero and one.

Motivated by these considerations, in [44] the problem of distribution-free robustness is studied. In other words, the objective of this line of research is to determine the worst-case distribution in a certain class of probability measures. More precisely, let $\mathbf{q}$ be an ℓ-dimensional real vector of independent random variables with support

$$\mathcal{B}_q = \left\{\mathbf{q} \in \mathbb{R}^\ell : \|\mathbf{q}\|_\infty \leq 1\right\}$$

and $\mathcal{B}_G \subset \mathbb{R}^\ell$ be a closed, convex and centrally symmetric set. Then, in [44] it is proven that

$$\min_{f_\mathbf{q} \in \mathcal{F}} \int_{\mathcal{B}_G} f_\mathbf{q}(q)\,\mathrm{d}q = \int_{\mathcal{B}_G} \mathcal{U}_{\mathcal{B}_q}\,\mathrm{d}q$$

where $\mathcal{U}_{\mathcal{B}_q}$ is the uniform probability density function with support $\mathcal{B}_q$ and $\mathcal{F}$ is the set of probability density functions satisfying two conditions:

1. The cdf $F_\mathbf{q}(q)$ is absolutely continuous, so that the density function

$$f_\mathbf{q}(q) = \prod_{i=1}^{\ell} f_{\mathbf{q}_i}(q_i)$$

   is well defined;
2. The marginal density functions $f_{\mathbf{q}_i}(q_i)$ are nonincreasing and centrally symmetric.

This result is generally denoted as the *uniformity principle*. In [44], applications to robustness analysis of affine polynomial families are also shown, taking $\mathcal{B}_G$ as the so-called value set. However, the fact that $\mathcal{B}_G$ needs to be convex and centrally symmetric may be a critical requirement that is generally not satisfied for the good

and bad sets. The convexity assumption has been partially removed in [251] by introducing the concept of *unirectangular sets*, while [248] considers the case of nonsymmetric distributions. In the paper [36], further worst-case properties of the uniform distribution are proved. Subsequent research along this direction has been performed in various papers. The interested reader may also refer to the survey paper [249], which is focused on this particular line of research.

6.5 Historical Notes on Probabilistic Methods

From the historical point of view, probabilistic methods for robustness made some early appearances in the 1980s, but they did not receive adequate attention in the systems and control literature at that time. In particular, the notion of "probability of instability," which is crucial for probabilistic robustness, was first introduced in the context of flight control in 1980 by Stengel [366]. Similar ideas were subsequently revisited in Stengel's book on stochastic optimal control [367] in 1986. In 1989, the paper titled "Probabilistic robust controller design" [137] was published in the Proceedings of the IEEE Conference on Decision and Control. This is presumably the first paper with a title containing both words "probabilistic" and "robust." Stengel and co-workers further pursued the line of research on stochastic robustness, publishing several interesting results. In particular, those papers explored various techniques, mainly based on Monte Carlo, for the computation of the probability of instability, and related performance concepts, with specific attention to flight dynamics applications within aerospace engineering. However, the absence of newly developed mathematical tools basically limited these attempts to merge probability and robustness to *analysis* problems. We recall that in the context of system identification and estimation in the presence of noisy measurements, a parallel line of research is the so-called *rapprochement viewpoint*, see e.g. [178, 306].

A few years later, in 1996 the papers [233, 381] (developed independently by Khargonekar and Tikku and by Tempo, Bai and Dabbene) proposed an approach based on explicit sample size bounds, thus refuelling enthusiasm on randomized techniques. The study of statistical learning theory and its application to control conducted by Vidyasagar [404, 405] provided additional impetus and also exposed researchers to a different viewpoint based on the solid mathematical foundations of statistical learning theory. This formulation led to the development of randomized algorithms for control system *design*. Subsequently, research on randomized algorithms and probabilistic methods evolved significantly in particular on the topics of sequential methods, the scenario approach, statistical learning techniques for control, and on specific applications. At this stage, it is premature to provide an historical viewpoint on these more recent developments.

Chapter 7
Monte Carlo Methods

In this chapter we discuss Monte Carlo (MC) and quasi-Monte Carlo (QMC) methods. The Monte Carlo method has been widely used for simulations of various physical and mathematical systems and has a very long history that officially began in 1949 with the seminal paper of Metropolis and Ulam [284], see also [283]. Such algorithms were used in the Manhattan Project. The name Monte Carlo probably originated from the famous casino in Monaco and reflects the random and repetitive nature of the process, which is similar to gambling in casinos. We refer e.g. to [177] for historical remarks and introductory material. The quasi-Monte Carlo method is more recent and may be regarded as a deterministic version of MC, see e.g. [303, 305]. A formal definition of a Monte Carlo randomized algorithm is given in Chap. 10.

7.1 Probability and Expected Value Estimation

In this section, we discuss randomized techniques for probability and expected value estimation. In Chap. 6 we introduced probabilistic versions of classical robustness problems, which are based on the computation of the probability of performance. That is, for a given performance level γ, the objective is to estimate the probability

$$p(\gamma) = \mathrm{Pr}_{\boldsymbol{\Delta}}\{J(\boldsymbol{\Delta}) \leq \gamma\} = \int_{\mathcal{B}_G} f_{\boldsymbol{\Delta}}(\Delta)\,\mathrm{d}\Delta.$$

The evaluation of this probability requires the solution of multiple integrals. Hence, its exact computation is very difficult in general, and only in special cases $p(\gamma)$ can be obtained in closed form, see the examples given in the previous chapters.

A classical tool for the numerical evaluation of multiple integrals is the Monte Carlo method. To estimate the probability $p(\gamma)$ with this approach, we generate N independent identically distributed (iid) random samples within the set $\mathcal{B}_{\mathbb{D}}$

$$\boldsymbol{\Delta}^{(1\ldots N)} \doteq \{\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(N)}\} \tag{7.1}$$

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_7,

according to the given density function $f_{\boldsymbol{\Delta}}(\Delta)$, where $\boldsymbol{\Delta}^{(1\ldots N)}$ is called a *multisample* of $\boldsymbol{\Delta}$ of cardinality N. Then, we evaluate

$$J\big(\boldsymbol{\Delta}^{(1)}\big), \ldots, J\big(\boldsymbol{\Delta}^{(N)}\big).$$

A *Monte Carlo estimate* of $p(\gamma)$ is given by

$$\widehat{\mathbf{p}}_N(\gamma) = \frac{\mathbf{N}_G}{N}$$

where $\mathbf{N}_G$ is the number of "good" samples such that $J(\boldsymbol{\Delta}^{(i)}) \leq \gamma$. More formally, we define the indicator function associated with the good set

$$\mathbb{I}_{\mathcal{B}_G}(\Delta) = \begin{cases} 1 & \text{if } \Delta \in \mathcal{B}_G; \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{B}_G = \{\Delta \in \mathcal{B}_{\mathbb{D}} : J(\Delta) \leq \gamma\}$ is defined in (6.2). Then, we write

$$\widehat{\mathbf{p}}_N(\gamma) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{\mathcal{B}_G}\big(\boldsymbol{\Delta}^{(i)}\big). \tag{7.2}$$

That is, the probability $p(\gamma)$ is estimated by means of the empirical mean of the good set indicator function (see further discussions in Remark 7.1). The estimate $\widehat{\mathbf{p}}_N(\gamma)$ is a random variable usually referred to as the *empirical probability*. The weak and strong *laws of large numbers* presented next guarantee asymptotic convergence in probability and with probability one, respectively, of the empirical probability $\widehat{\mathbf{p}}_N(\gamma)$ to $p(\gamma)$.

Theorem 7.1 (Laws of large numbers for empirical probability) *For any* $\epsilon \in (0, 1)$, *the* weak law of large numbers *states that*

$$\lim_{N\to\infty} \Pr\big\{\big|p(\gamma) - \widehat{\mathbf{p}}_N(\gamma)\big| > \epsilon\big\} = 0. \tag{7.3}$$

The strong law of large numbers *guarantees that*

$$\lim_{N\to\infty} \widehat{\mathbf{p}}_N(\gamma) = p(\gamma) \tag{7.4}$$

with probability one (*a.e.*).

The weak law (7.3) is classical, see e.g. [319], and follows directly from the Bernoulli bound presented in Chap. 8. The strong law (7.4) is a consequence of the Borel–Cantelli lemma, which gives a sufficient condition for a.e. convergence, see e.g. [406] for a proof and a discussion on the subject.

The MC approach can be readily used in the more general situation where the estimation of the expected value is of concern. That is, given a performance function $J(\Delta)$ and a density function $f_{\boldsymbol{\Delta}}(\Delta)$ with support $\mathcal{B}_{\mathbb{D}}$, we aim at estimating

$$\mathrm{E}_{\boldsymbol{\Delta}}\big(J(\boldsymbol{\Delta})\big) = \int_{\mathcal{B}_{\mathbb{D}}} J(\Delta) f_{\boldsymbol{\Delta}}(\Delta)\, \mathrm{d}\Delta.$$

In this case, we take the multisample $\boldsymbol{\Delta}^{(1\dots N)}$ defined in (7.1) and compute the so-called *empirical mean*

$$\widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big) \doteq \frac{1}{N}\sum_{i=1}^{N} J\big(\boldsymbol{\Delta}^{(i)}\big). \tag{7.5}$$

Subsequently, when clear from the context, the empirical mean will be denoted by $\widehat{\mathbf{E}}_N$. The two laws of large numbers are now stated for empirical mean estimation.

Theorem 7.2 (Laws of large numbers for empirical mean) *For any $\epsilon > 0$, we have*

$$\lim_{N\to\infty} \Pr\Big\{\big|\mathrm{E}_{\boldsymbol{\Delta}}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big)\big| > \epsilon\Big\} = 0.$$

Moreover, the empirical mean converges a.e. to the expected value, that is

$$\lim_{N\to\infty} \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big) = \mathrm{E}_{\boldsymbol{\Delta}}\big(J(\boldsymbol{\Delta})\big).$$

Remark 7.1 (Probability versus expected value) The probability estimation problem can be seen as a special case of expected value estimation. Indeed, if we define

$$\widetilde{J}(\Delta) \doteq \mathbb{I}_{\mathcal{B}_G}(\Delta)$$

then it follows that

$$\mathrm{E}_{\boldsymbol{\Delta}}\big(\widetilde{J}(\boldsymbol{\Delta})\big) = \int_{\mathcal{B}_{\mathbb{D}}} \mathbb{I}_{\mathcal{B}_G}(\Delta) f_{\boldsymbol{\Delta}}(\Delta)\,\mathrm{d}\Delta = \int_{\mathcal{B}_G} f_{\boldsymbol{\Delta}}(\Delta)\,\mathrm{d}\Delta = \Pr_{\boldsymbol{\Delta}}\big\{J(\boldsymbol{\Delta}) \le \gamma\big\}.$$

The asymptotic convergence of $\widehat{\mathbf{E}}_N(J(\boldsymbol{\Delta}))$ to the expected value $\mathrm{E}(J(\boldsymbol{\Delta}))$ is guaranteed by the laws of large numbers. For finite sample size N, it is of great interest to compute the expected value of the squared difference between $\mathrm{E}(J(\boldsymbol{\Delta}))$ and the empirical mean $\widehat{\mathbf{E}}_N(J(\boldsymbol{\Delta}))$. More precisely, in the next theorem, which is a classical result in the Monte Carlo literature, see e.g. [303], we explicitly compute

$$\begin{aligned}\mathrm{Var}(\widehat{\mathbf{E}}_N) &\doteq \mathrm{E}_{\boldsymbol{\Delta}^{(1\dots N)}}\Big(\big(\mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big)^2\Big)\\ &= \int_{\mathcal{B}_{\mathbb{D}}} \cdots \int_{\mathcal{B}_{\mathbb{D}}} \left(\mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \frac{1}{N}\sum_{i=1}^{N} J\big(\Delta^{(i)}\big)\right)^2 \prod_{k=1}^{N} f_{\boldsymbol{\Delta}}\big(\Delta^{(k)}\big)\,\mathrm{d}\Delta^{(k)}.\end{aligned}$$

The variance $\mathrm{Var}(\widehat{\mathbf{E}}_N)$ is clearly a measure of the "goodness" of the approximation error of MC methods.

Theorem 7.3 *If the variance $\mathrm{Var}(J(\boldsymbol{\Delta}))$ is finite, then for any $N \ge 1$, we have*

$$\mathrm{E}_{\boldsymbol{\Delta}^{(1\dots N)}}\Big(\big(\mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big)^2\Big) = \frac{\mathrm{Var}(J(\boldsymbol{\Delta}))}{N}. \tag{7.6}$$

Proof First, we define $h(\Delta) \doteq \mathrm{E}(J(\boldsymbol{\Delta})) - J(\Delta)$. Then, we have

$$\int_{\mathcal{B}_{\mathbb{D}}} h(\Delta) f_{\boldsymbol{\Delta}}(\Delta)\,\mathrm{d}\Delta = 0$$

and

$$\mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N = \frac{1}{N}\sum_{i=1}^{N} h\big(\boldsymbol{\Delta}^{(i)}\big).$$

Hence, we write

$$\begin{aligned}
&\mathrm{E}_{\boldsymbol{\Delta}^{(1\ldots N)}}\Big(\big(\mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big)^2\Big) \\
&\quad= \int_{\mathcal{B}_{\mathbb{D}}}\int_{\mathcal{B}_{\mathbb{D}}}\cdots\int_{\mathcal{B}_{\mathbb{D}}}\left(\frac{1}{N}\sum_{i=1}^{N} h\big(\Delta^{(i)}\big)\right)^2 \prod_{k=1}^{N} f_{\boldsymbol{\Delta}}\big(\Delta^{(k)}\big)\,\mathrm{d}\Delta^{(k)} \\
&\quad= \frac{1}{N^2}\sum_{i=1}^{N}\int_{\mathcal{B}_{\mathbb{D}}}\int_{\mathcal{B}_{\mathbb{D}}}\cdots\int_{\mathcal{B}_{\mathbb{D}}} h\big(\Delta^{(i)}\big)^2 \prod_{k=1}^{N} f_{\boldsymbol{\Delta}}\big(\Delta^{(k)}\big)\,\mathrm{d}\Delta^{(k)} \\
&\qquad+ \frac{2}{N^2}\sum_{i=1}^{N}\sum_{k>i}^{N}\int_{\mathcal{B}_{\mathbb{D}}}\int_{\mathcal{B}_{\mathbb{D}}}\cdots\int_{\mathcal{B}_{\mathbb{D}}} h\big(\Delta^{(i)}\big)h\big(\Delta^{(k)}\big)\prod_{k=1}^{N} f_{\boldsymbol{\Delta}}\big(\Delta^{(k)}\big)\,\mathrm{d}\Delta^{(k)} \\
&\quad= \frac{1}{N}\int_{\mathcal{B}_{\mathbb{D}}} h(\Delta)^2 f_{\boldsymbol{\Delta}}(\Delta)\,\mathrm{d}\Delta = \frac{\mathrm{Var}(J(\boldsymbol{\Delta}))}{N}.
\end{aligned}$$

□

Remark 7.2 (Breaking the curse of dimensionality) As a consequence of this theorem, we obtain that the average absolute value of the approximation error of the MC method is given by $\sigma_J/\sqrt{N}$, where $\sigma_J = \sqrt{\mathrm{Var}(J(\boldsymbol{\Delta}))}$ is the standard deviation. Assuming that the variance $\mathrm{Var}(J(\boldsymbol{\Delta}))$ is known, then the number of samples necessary to guarantee a given error can be established *a priori*. That is, we compute

$$N \geq \frac{\mathrm{Var}(J(\boldsymbol{\Delta}))}{\mathrm{Var}(\widehat{\mathbf{E}}_N)}. \tag{7.7}$$

Unfortunately, since $\mathrm{Var}(J(\boldsymbol{\Delta}))$ is generally unknown, Eq. (7.6) can only be used to conclude that the error is of the order $O(N^{-1/2})$. An important consequence of this discussion is that the mean square error of the Monte Carlo estimate is *independent* of the problem dimension. This is the reason why Monte Carlo methods are said to *break the curse of dimensionality*. This issue is further discussed in Sect. 10.5 when dealing with computational complexity of randomized algorithms.

Remark 7.3 (Probability of rare events) A technique to improve the quality of Monte Carlo estimation consists in reducing the variance $\mathrm{Var}(J(\boldsymbol{\Delta}))$ by shaping the density function $f_{\boldsymbol{\Delta}}(\Delta)$ in a suitable way. This leads to the methods of stratified and importance sampling, see e.g. [177, 334]. These techniques have been developed to progressively shift the sampling distribution towards the failure region, so as to gain information from rare events more efficiently. In particular, if the probability of interest is itself very small (i.e. it is a rare event probability), then we would need

on average many samples before a "failure" occurs, that is before a random sample $\boldsymbol{\Delta}^{(i)}$ is found such that $J(\boldsymbol{\Delta}^{(i)}) > \gamma$. To show this fact, let $V = \Pr\{J(\boldsymbol{\Delta}) > \gamma\}$ be small (rare failure), let $\boldsymbol{\Delta}^{(1)}, \boldsymbol{\Delta}^{(2)}, \ldots$ be a sequence of random samples and define the random variable

$$\mathbf{n} \doteq \inf\{i = 1, 2, \ldots : J(\boldsymbol{\Delta}^{(i)}) > \gamma\}.$$

Then, we compute the expected value of $\mathbf{n}$ obtaining

$$\begin{aligned} \mathrm{E}(\mathbf{n}) &= \Pr\{J(\boldsymbol{\Delta}^{(1)}) > \gamma\} + 2\Pr\{J(\boldsymbol{\Delta}^{(1)}) \leq \gamma, J(\boldsymbol{\Delta}^{(2)}) > \gamma\} + \cdots \\ &\quad + k\Pr\{J(\boldsymbol{\Delta}^{(1)}) \leq \gamma, \ldots, J(\boldsymbol{\Delta}^{(k-1)}) \leq \gamma, J(\boldsymbol{\Delta}^{(k)}) > \gamma\} + \cdots \\ &= \sum_{k=0}^{\infty} k(1-V)^{k-1}V = \frac{V}{1-V}\sum_{k=0}^{\infty} k(1-V)^{k} = \frac{1}{V}. \end{aligned}$$

We conclude that finding small probabilities thus requires information from the samples of rare events corresponding to failures and, on average, $1/V$ samples are needed before a failure is detected.

Methods for computing small failure probabilities for certain classes of dynamical systems subject to stochastic excitation include the "subset simulation" approach [30, 107], which is based on the idea of factoring the failure probability in the product of larger conditional failure probabilities that can be estimated with lower computational effort. Hence, we are replacing the problem in the original probability space by a sequence of simulations of more frequent events in the conditional probability spaces.

Remark 7.4 (Random sample generation) One of the key issues regarding the application of MC techniques in systems and control is the availability of efficient algorithms for the generation of the multisample (7.1) according to a given density function over the support $\mathcal{B}_{\mathbb{D}}$. This problem is fully addressed in Chaps. 16 and 18, where algorithms for generating samples of random vectors and matrices in the structured set $\mathcal{B}_{\mathbb{D}}$ are presented. These algorithms reduce the problem to the univariate generation of uniform samples in the interval $[0, 1]$, which is the standard random number generation problem discussed in Chap. 14.

The techniques and the convergence results presented in this section are at the basis of Monte Carlo methods for computation of multiple integrals, which are discussed next.

7.2 Monte Carlo Methods for Integration

Monte Carlo methods address the general problem of computing numerically the multiple integral

$$\int_{\mathcal{Y}} g(x)\,\mathrm{d}x \tag{7.8}$$

of a multivariate measurable function $g(x) : \mathbb{R}^n \to \mathbb{R}$ with domain $\mathcal{Y} \subset \mathbb{R}^n$. The main idea is to transform this integral into an expected value computation problem. This can be done by factorizing the function $g(x)$ into the product of two terms $J(x)$ and $f_{\mathbf{x}}(x)$ such that $f_{\mathbf{x}}(x)$ is a probability density function with support $\mathcal{Y}$, and

$$g(x) = J(x) f_{\mathbf{x}}(x). \tag{7.9}$$

With this choice, the integral (7.8) can be viewed as the expected value of $J(\mathbf{x})$ with respect to the density function $f_{\mathbf{x}}(x)$, i.e.

$$\int_{\mathcal{Y}} g(x)\,\mathrm{d}x = \int_{\mathcal{Y}} J(x) f_{\mathbf{x}}(x)\,\mathrm{d}x = \mathrm{E}_{\mathbf{x}}\big(J(\mathbf{x})\big). \tag{7.10}$$

An MC estimate of the integral (7.8) is then immediately obtained via the techniques described in the previous section. That is, we approximate the expected value in (7.10) with the empirical mean

$$\widehat{\mathbf{E}}_N\big(J(\mathbf{x})\big) = \frac{1}{N}\sum_{i=1}^{N} J\big(\mathbf{x}^{(i)}\big)$$

where the multisample

$$\mathbf{x}^{(1\ldots N)} \doteq \big\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\big\} \tag{7.11}$$

of cardinality N is generated according to the pdf $f_{\mathbf{x}}(x)$ with support $\mathcal{Y}$. Recalling Theorem 7.3, we immediately obtain that the variance of the estimate is equal to

$$\mathrm{Var}\big(\widehat{\mathbf{E}}_N\big(J(\mathbf{x})\big)\big) = \frac{\mathrm{Var}(J(\mathbf{x}))}{N}.$$

Various techniques have been developed to reduce this error for finite and fixed sample size, such as the importance sampling techniques discussed in Remark 7.3. To conclude this section, in the following example we compare the complexity of integration based on the MC method with that obtained with a trapezoidal rule of integration.

Example 7.1 (Trapezoidal rule for integration) The computation of multiple integrals can be performed using the multidimensional trapezoidal rule for integration of functions with continuous and bounded second partial derivatives. For example, consider the integral

$$\int_{\mathcal{Y}} g(x)\,\mathrm{d}x \tag{7.12}$$

where $g(x) : \mathbb{R}^n \to \mathbb{R}$ is a twice-differentiable function and the integration domain $\mathcal{Y}$ is the unit cube in $\mathbb{R}^n$. In this case, we construct a trapezoidal approximation of (7.12) based on N grid points for each x_i

$$\sum_{i_1=1}^{N} \cdots \sum_{i_n=1}^{N} w_{i_1} \cdots w_{i_n} g\big(\widetilde{x}(i_1, \ldots, i_n)\big)$$

where

$$\widetilde{x}(i_1,\ldots,i_n)=\left[\left(\frac{i_1-1}{N-1}\right)\quad\cdots\quad\left(\frac{i_n-1}{N-1}\right)\right]^T$$

and the weights are given by

$$w_1=\frac{1}{2N-2},\quad w_2=\frac{1}{N-1},\quad w_3=\frac{1}{N-1},\quad\ldots,$$
$$w_{N-1}=\frac{1}{N-1},\quad w_N=\frac{1}{2N-2}.$$

It is well known that the deterministic error given by the trapezoidal rule is of the order $O(N^{-2/n})$, see for instance [303]. However, in Theorem 7.3, we have shown that the average error of MC algorithms is of the order $O(N^{-1/2})$, where N is the number of random samples. Hence, comparing these errors we conclude that Monte Carlo methods improve upon classical deterministic algorithms based on the trapezoidal rule for integration for $n>4$. Obviously, it should be noticed that the Monte Carlo mean square error given in Theorem 7.3 is of probabilistic, and not deterministic, nature.

7.3 Monte Carlo Methods for Optimization

In this section we briefly discuss the application of Monte Carlo techniques in optimization problems, see [334] for a survey of the literature. In particular, consider a bounded multivariate function $g(x):\mathcal{Y}\to\mathbb{R}$ and suppose we are interested in evaluating

$$g^*=g\big(x^*\big)=\sup_{x\in\mathcal{Y}}g(x). \tag{7.13}$$

A simple algorithm for estimating the optimal value of $g(x)$ has been proposed in [74] using the so-called *nonadaptive random search algorithm*. First, we draw N iid points $\mathbf{x}^{(i)}$, $i=1,\ldots,N$ in $\mathcal{Y}$ according to a given density $f_{\mathbf{x}}(x)$. Then, an approximation of the maximum in (7.13) is given by the *empirical maximum*

$$\widehat{\mathbf{g}}_N=\max_{i=1,\ldots,N}g\big(\mathbf{x}^{(i)}\big).$$

The next theorem studies asymptotic convergence of the estimate $\widehat{\mathbf{g}}_N$ to g^*.

Theorem 7.4 (Laws of large numbers for empirical maximum) *Assume that the density function $f_{\mathbf{x}}(x)$ assigns a nonzero probability to every neighborhood of x^*, and $g(x)$ is continuous at x^*. Then, for any $\epsilon>0$*

$$\lim_{N\to\infty}\Pr\big\{g^*-\widehat{\mathbf{g}}_N>\epsilon\big\}=0.$$

Moreover, the empirical maximum converges a.e. to the true maximum

$$\lim_{N\to\infty}\widehat{\mathbf{g}}_N=g^*.$$

More sophisticated Monte Carlo algorithms for global optimization have been widely studied in the literature. We recall here the *multistart random search*, which performs a series of gradient descents starting from random generating initial points, and the *simulated annealing* algorithm, for further details see for instance [421].

7.4 Quasi-Monte Carlo Methods

The quasi-Monte Carlo method is a deterministic version of Monte Carlo with the primary goal of obtaining guaranteed (instead of probabilistic) errors. Some motivations for studying QMC methods may also come from the difficulty of generating "truly" random samples, see Chap. 14 for further discussions and the classical reference [303] for a complete treatment of the subject.

In the quasi-Monte Carlo method, *deterministic points* chosen according to some optimality criterion are used instead of random samples generated according to a given probability density function. For integration of a multivariate function, the most frequently used criterion is the so-called *discrepancy*, which is a measure of how the sample set is "evenly distributed" within the integration domain, which is usually taken as the unit cube in $\mathbb{R}^n$. The problem therefore is to find specific sequences of points which minimize the discrepancy, or upper bounds on it. There are many such sequences, including Sobol', Halton, and Niederreiter. For integrands with a sufficiently low "degree of regularity," these sequences guarantee a deterministic error bound for integration of $O(N^{-1}(\log N)^n)$. For fixed dimension n, and for large N, this error is therefore smaller than the mean square error of the MC estimate, which is $O(N^{-1/2})$. QMC methods are also used for optimization problems. In this case, the optimality criterion used is the *dispersion*, which is a measure of denseness, rather than equidistribution, as in the case of discrepancy. In this section, which is largely based on [303], we study both integration and optimization problems and the related sequences.

Various applications of QMC methods and numerical comparisons with MC methods have been developed, in particular in the areas of path integrals for mathematical finance, see for instance [389] and references therein, planning algorithms [256], and congestion control of communication networks [17].

7.4.1 Discrepancy and Error Bounds for Integration

In this section we consider the integration of a (measurable) multivariate function $g(x) : \mathbb{R}^n \to \mathbb{R}$

$$\int_{[0,1]^n} g(x)\,\mathrm{d}x$$

with domain

$$[0,1]^n \doteq \left\{x \in \mathbb{R}^n : x_i \in [0,1],\ i = 1, \ldots, n\right\}.$$

This integral is approximated by the sum

$$\frac{1}{N}\sum_{i=1}^{N} g\left(x^{(i)}\right)$$

where $x^{(i)} \in [0,1]^n$, $i = 1, \ldots, N$, are now *deterministic* vector points. The deterministic multisample of cardinality N

$$x^{(1\ldots N)} \doteq \left\{x^{(1)}, \ldots, x^{(N)}\right\} \tag{7.14}$$

is usually called a *point set*. Intuitively, these points should be "evenly distributed," so that the irregularity of their distribution within the unit cube is minimized. This intuitive concept leads to the definition of discrepancy.

Definition 7.1 (Discrepancy) Let $\mathcal{S}$ be a nonempty family of subsets of $[0,1]^n$. The discrepancy $D_N(\mathcal{S}, x^{(1\ldots N)})$ of a point set $x^{(1\ldots N)}$ of cardinality N with respect to $\mathcal{S}$ is defined as

$$D_N\left(\mathcal{S}, x^{(1\ldots N)}\right) \doteq \sup_{S\in\mathcal{S}} \left| \frac{\sum_{i=1}^{N} \mathbb{I}_S(x^{(i)})}{N} - \mathrm{Vol}(S) \right| \tag{7.15}$$

where $\mathbb{I}_S(x)$ is the indicator function of S and $\mathrm{Vol}(S)$ is its volume.

We remark that the discrepancy is a nonnegative quantity and it is upper bounded by one. Next, we define the star discrepancy $D_N^*(x^{(1\ldots N)})$ and the extreme discrepancy $D_N^e(x^{(1\ldots N)})$, obtained by considering specific choices of the family $\mathcal{S}$.

Definition 7.2 (Star discrepancy) Let $\mathcal{S}^*$ be the family of all subintervals of the semi-open unit cube[1] $[0,1)^n$ of the form $\{x \in \mathbb{R}^n : x_i \in [0, v_i),\ i = 1, \ldots, n\}$. Then, the star discrepancy $D_N^*(x^{(1\ldots N)})$ is defined as

$$D_N^*\left(x^{(1\ldots N)}\right) \doteq D_N\left(\mathcal{S}^*, x^{(1\ldots N)}\right).$$

Definition 7.3 (Extreme discrepancy) Let $\mathcal{S}^e$ be the family of all subintervals of the semi-open unit cube $[0,1)^n$ of the form $\{x \in \mathbb{R}^n : x_i \in [u_i, v_i),\ i = 1, \ldots, n\}$. Then, the extreme discrepancy $D_N^e(x^{(1\ldots N)})$ is defined as

$$D_N^e\left(x^{(1\ldots N)}\right) = D_N\left(\mathcal{S}^e, x^{(1\ldots N)}\right).$$

For any $x^{(1\ldots N)}$ in the unit cube, it can be shown that the extreme and the star discrepancies are related as follows

$$D_N^*\left(x^{(1\ldots N)}\right) \le D_N^e\left(x^{(1\ldots N)}\right) \le 2^n D_N^*\left(x^{(1\ldots N)}\right).$$

The definition of extreme discrepancy will be used later in Sect. 7.4.4 when studying the connections with dispersion.

[1] The semi-open unit cube is defined as $[0,1)^n \doteq \{x \in \mathbb{R}^n : x_i \in [0,1),\ i = 1, \ldots, n\}$.

By means of the star discrepancy, we can establish error bounds on the integration error

$$\left| \int_{[0,1]^n} g(x)\,\mathrm{d}x - \frac{1}{N}\sum_{i=1}^{N} g\big(x^{(i)}\big). \right| \tag{7.16}$$

A classical result in this direction, often called the *Koksma–Hlawka inequality* [203, 239], can be stated in terms of the total variation. Namely, for functions with continuous partial derivatives on $[0,1]^n$, the total variation of g in the sense of Vitali is defined as

$$V^{(n)}(g) = \int_0^1 \cdots \int_0^1 \left| \frac{\partial^n g}{\partial x_1 \cdots \partial x_n} \right| \mathrm{d}x_1 \cdots \mathrm{d}x_n.$$

Then, if g has bounded variation on $[0,1]^n$ in the sense of Vitali, and the restriction of g to each k-dimensional face of $[0,1]^n$, for $k = 1, \ldots, n$, is of bounded variation in the sense of Vitali, then g is said to be of bounded variation on $[0,1]^n$ in the sense of Hardy and Krause. This concept is an n-dimensional extension of the scalar variation in the interval $[0,1]$.

Theorem 7.5 (Koksma–Hlawka inequality) *Assume that $g : \mathbb{R}^n \to \mathbb{R}$ has bounded variation $V^{(n)}(g)$ on $[0,1]^n$ in the sense of Hardy and Krause. Then, for any $x^{(1\ldots N)}$ with $x^{(i)} \in [0,1)^n$, $i = 1, \ldots, N$, we have*

$$\left| \int_{[0,1]^n} g(x)\,\mathrm{d}x - \frac{1}{N}\sum_{i=1}^{N} g\big(x^{(i)}\big) \right| \le V^{(n)}(g) D_N^*\big(x^{(1\ldots N)}\big). \tag{7.17}$$

From this theorem it follows that, for given variation $V^{(n)}(g)$, the integration error in (7.17) is minimized if the point set $x^{(1\ldots N)}$ is selected so that the star discrepancy $D_N^*(x^{(1\ldots N)})$ is minimized.

Next, we present two results stated in [303] that give a precise characterization of the star and extreme discrepancies in the special case $n = 1$.

Theorem 7.6 *If $n = 1$ and $0 \le x^{(1)} \le \cdots \le x^{(N)} \le 1$, then*

$$D_N^*\big(x^{(1\ldots N)}\big) = \frac{1}{2N} + \max_{1\le i\le N} \left| x^{(i)} - \frac{2i-1}{2N} \right|.$$

Theorem 7.7 *If $n = 1$ and $0 \le x^{(1)} \le \cdots \le x^{(N)} \le 1$, then*

$$D_N^e\big(x^{(1\ldots N)}\big) = \frac{1}{N} + \max_{1\le i\le N} \left(\frac{i}{N} - x^{(i)} \right) - \min_{1\le i\le N} \left(\frac{i}{N} - x^{(i)} \right).$$

From these results, it can be easily verified that the two inequalities

$$D_N^*\big(x^{(1\ldots N)}\big) \ge \frac{1}{2N} \quad \text{and} \quad D_N^e\big(x^{(1\ldots N)}\big) \ge \frac{1}{N}$$

hold for any $N \geq 1$. We remark that equality is attained with the choice

$$x^{(i)} = \frac{2i-1}{2N}. \tag{7.18}$$

In this case, both star and extreme discrepancy are minimized. That is, if one wants to place N points in the interval $[0, 1]$ in order to minimize the error bound for integration given in (7.17), then the "optimal gridding" is the one given in (7.18), which corresponds to the N-panel midpoint integration rule, see e.g. [124]. Notice, however, that these facts may be used only when N is known a priori. A subsequent problem, addressed in the next section, is how to construct *recursive sequences* of points that guarantee low discrepancy.

7.4.2 One-Dimensional Low Discrepancy Sequences

We first study low discrepancy sequences for $n = 1$ on the semi-open interval $[0, 1)$. For an integer $b \geq 2$, we define

$$\mathcal{Z}_b \doteq \{0, 1, \ldots, b-1\}.$$

Every integer $k \geq 0$ has a unique digit expansion in base b

$$k = \sum_{i=0}^{\infty} a_i(k) b^i \tag{7.19}$$

where $a_i(k) \in \mathcal{Z}_b$ for all $i \geq 0$ and $a_i(k) = 0$ for all sufficiently large i. Then, for an integer $b \geq 2$, we define the *radical-reversal* function in base b as

$$\phi_b(k) = \sum_{i=0}^{\infty} a_i(k) b^{-i-1} \tag{7.20}$$

for all integers $k \geq 0$ and where $a_i(k)$ is given by (7.19). We are now ready to define the van der Corput sequence [399].

Definition 7.4 (van der Corput sequence) Let $n = 1$. For an integer $b \geq 2$, the van der Corput sequence in base b is the sequence

$$x^{(1,\ldots)} \doteq x^{(1)}, \quad x^{(2)}, \quad \ldots$$

where $x^{(i)} = \phi_b(i-1)$ for all $i \geq 1$.

Example 7.2 (Binary van der Corput sequence) We study the van der Corput sequence in base $b = 2$. To illustrate, we compute the element $x^{(24)} = \phi_2(23)$. First, we write

$$23 = \sum_{i=0}^{\infty} a_i(23) 2^i.$$

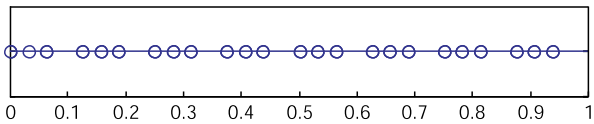

Fig. 7.1 First 24 points of the binary van der Corput sequence

With straightforward computations, we obtain

$$a_0(23) = 1, \quad a_1(23) = 1, \quad a_2(23) = 1, \quad a_2(23) = 0, \quad a_4(23) = 1$$

and $a_i(23) = 0$ for all $i \geq 5$. Then, we have

$$\begin{aligned} x^{(24)} = \phi_2(23) &= \frac{a_0(23)}{2} + \frac{a_1(23)}{4} + \frac{a_2(23)}{8} + \frac{a_3(23)}{16} + \frac{a_4(23)}{32} \\ &= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{32} = 0.9063. \end{aligned}$$

Similarly, we construct the other elements $x^{(1)} = \phi_2(0), x^{(2)} = \phi_2(1), \ldots$ obtaining the sequence

0, 0.5, 0.25, 0.75, 0.125, 0.625, 0.375, 0.875, 0.0625,
0.5625, 0.3125, 0.8125, 0.1875, 0.6875, 0.4375, 0.9375, 0.0313, 0.5313,
0.2813, 0.7813, 0.1563, 0.6563, 0.4063, 0.9063, ...

The first 24 points of this sequence are plotted in Fig. 7.1.

The discrepancy of the van der Corput sequence is of the order of magnitude of $O(N^{-1} \log N)$ for all $N \geq 2$ (see Theorem 7.8). It can be shown [303] that this is the best bound achievable by any sequence of points in $[0, 1]$. This result can be compared with Theorem 7.6, which states that the discrepancy of an N-point set in $[0, 1]$ is of the order $O(N^{-1})$.

In the next section, we study low discrepancy sequences for $n > 1$.

7.4.3 Low Discrepancy Sequences for n > 1

The van der Corput sequence can be extended to any dimension n. This leads to the definition of the so-called Halton sequence [190].

Definition 7.5 (Halton sequence) Let $b_1, \ldots, b_n$ be integers ≥ 2 and let ϕ_{b_i} be defined as in (7.20) for $b = b_i$. The Halton sequence in the bases $b_1, \ldots, b_n$ is the sequence

$$x^{(1,\ldots)} \doteq x^{(1)}, \quad x^{(2)}, \quad \ldots$$

where

$$x^{(i)} = \left[\phi_{b_1}(i-1) \quad \cdots \quad \phi_{b_n}(i-1)\right]^T$$

for all $i \geq 0$.

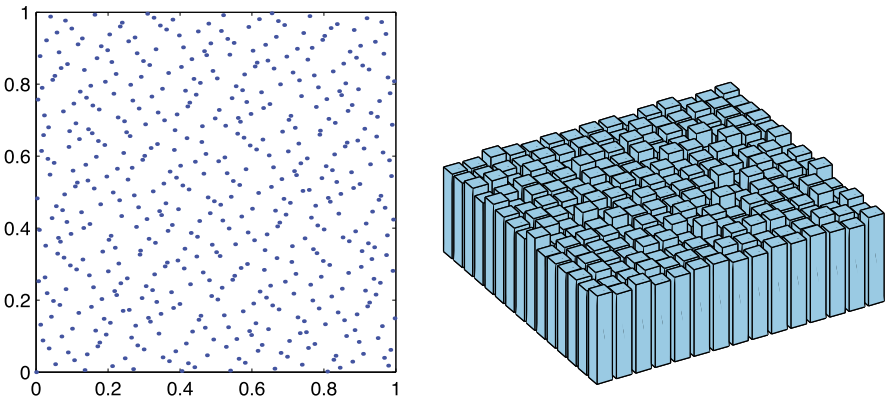

Fig. 7.2 Samples of two-dimensional Halton sequence for $N = 500$ and histogram of the relative frequency for $N = 10{,}000$ (each dimension is partitioned in 15 bins)

Samples of a two-dimensional Halton sequence are shown in Fig. 7.2. We now present a result which gives an upper bound on the star discrepancy of a Halton sequence.

Theorem 7.8 *For all $N \geq 1$, if $x^{(1\ldots N)}$ are the first N points of the Halton sequence in the pairwise relatively prime bases $b_1, \ldots, b_n$, then*

$$D_N^*\left(x^{(1\ldots N)}\right) < \frac{n}{N} + \frac{1}{N}\prod_{i=1}^{n}\left[\frac{b_i - 1}{2\log b_i}\log N + \frac{b_i + 1}{2}\right]. \tag{7.21}$$

Combining this result with Theorem 7.5, we conclude that, for functions with finite variation $V(g)$, the integration error given in (7.16) is of the order $O(N^{-1}(\log N)^n)$. It can be easily verified that, asymptotically, this error is much smaller than $O(N^{-1/2})$, which is that associated with classical Monte Carlo. However, when n is large, the factor $(\log N)^n$ becomes huge, and it takes an impracticably large sample size N before the performance of QMC becomes superior to MC.

Many other low discrepancy sequences are studied in the quasi-Monte Carlo literature. We recall in particular the Sobol' [361], Faure [163] and Niederreiter [302] sequences. The basic idea underlying these methods is to suitably permute the elements of a Halton sequence. In particular, the Sobol' sequence uses only the basis 2, whereas in the Faure sequence the basis is the smallest prime number $b \geq n$. For illustrative purposes, 1,000 points are generated in the unit box for the case of $n = 2$ for Halton, Faure, Sobol' and Niederreiter sequences. These points are shown in Fig. 7.3.

Finally, we would like to recall that discrepancy is not the only optimality criterion used in QMC methods. For example, as previously discussed, the *dispersion* is generally used in the context of optimization.

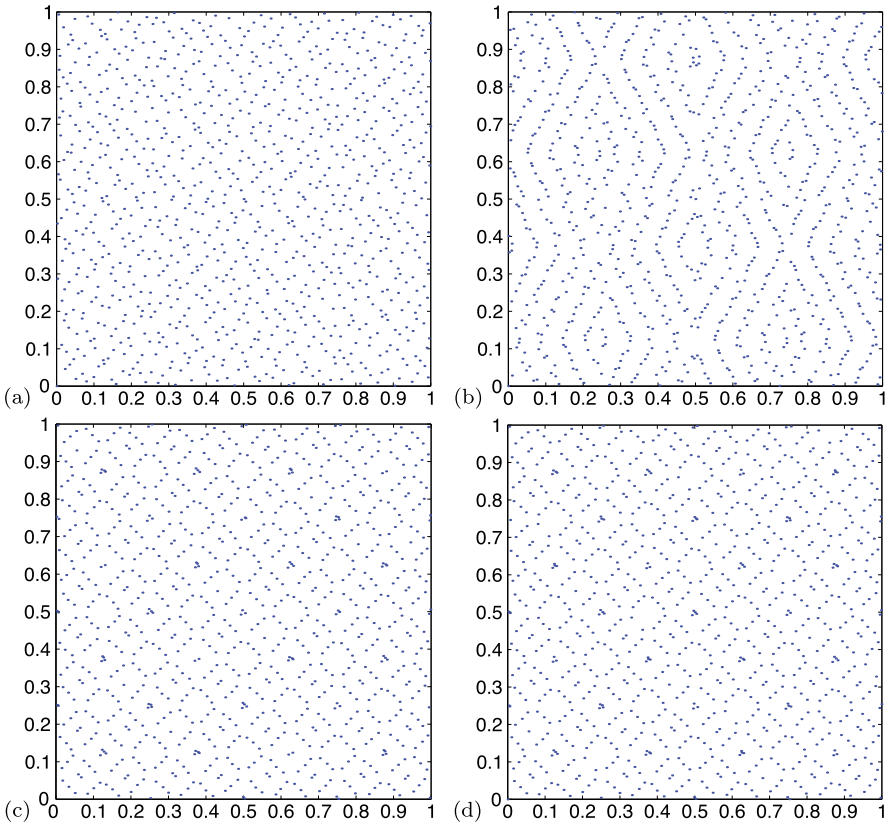

Fig. 7.3 Samples of two-dimensional sequences: (**a**) Halton, (**b**) Sobol', (**c**) Faure and (**d**) Niederreiter for $N = 1{,}000$

7.4.4 Dispersion and Point Sets for Optimization

In this section we study the QMC approach for maximization of a bounded multivariate function $g(x) : \mathbb{R}^n \to \mathbb{R}$ over the unit cube $[0, 1]^n$

$$\sup_{x \in [0,1]^n} g(x).$$

We consider the QMC approximation

$$\max_{i=1,\ldots,N} g\left(x^{(i)}\right)$$

where $x^{(i)}$, $i = 1, \ldots, N$, belong to a deterministic point set $x^{(1\ldots N)}$. Clearly, the approximation error

$$\sup_{x \in [0,1]^n} g(x) - \max_{i=1,\ldots,N} g\left(x^{(i)}\right) \tag{7.22}$$

is related to specific properties of the point set. Hence, we define the dispersion of a point set $x^{(1\ldots N)}$ in the n-dimensional unit cube $[0,1]^n$.

Definition 7.6 (Dispersion) The dispersion $d_N(x^{(1\ldots N)})$ of a point set $x^{(1\ldots N)}$ with cardinality N is defined as

$$d_N\left(x^{(1\ldots N)}\right) = \sup_{x\in[0,1]^n} \min_{1\le i\le N} \left\|x - x^{(i)}\right\|_\infty. \tag{7.23}$$

The next theorem relates the approximation error (7.22) to the dispersion of the point set $x^{(1\ldots N)}$. Let $\varpi(g,r)$ be the *modulus of continuity* of $g(x)$, which is defined for $r \ge 0$ as

$$\varpi(g,r) = \sup_{\substack{x,y\in[0,1]^n \\ \|x-y\|_\infty \le r}} \left|g(x) - g(y)\right|.$$

Theorem 7.9 *Let $g(x) : \mathbb{R}^n \to \mathbb{R}$ be a bounded function on the unit cube $[0,1]^n$. For any point set $x^{(1\ldots N)}$, we have*

$$\sup_{x\in[0,1]^n} g(x) - \max_{i=1,\ldots,N} g\left(x^{(i)}\right) \le \varpi\left(g, d_N\left(x^{(1\ldots N)}\right)\right).$$

From this theorem, we see that point sets with low dispersion are required when dealing with optimization problems. The following result establishes a precise connection between the dispersion and the extreme discrepancy of a point set.

Theorem 7.10 *For any point set $x^{(1\ldots N)}$ of cardinality N we have*

$$d_N\left(x^{(1\ldots N)}\right) \le \left[D_N^e\left(x^{(1\ldots N)}\right)\right]^{1/n}.$$

Therefore, we conclude that low extreme discrepancy implies low dispersion, but the converse is not necessarily true.

We now turn our attention to the computation of the dispersion of a given point set. First, we present a result for the special case $n = 1$.

Theorem 7.11 *If $n = 1$ and $0 \le x^{(1)} \le \cdots \le x^{(N)} \le 1$, then*

$$d_N\left(x^{(1\ldots N)}\right) = \max\left(x^{(1)}, \frac{1}{2}\left(x^{(2)} - x^{(1)}\right), \ldots, \frac{1}{2}\left(x^{(N)} - x^{(N-1)}\right), 1 - x^{(N)}\right).$$

It can be easily verified that the same set of points given in (7.18),

$$x^{(i)} = (2i-1)/(2N),$$

guarantees $d_N(x^{(1\ldots N)}) = 1/(2N)$, which is the minimum value of dispersion for any point set $x^{(1\ldots N)}$ in the unit cube.

For the n-dimensional case, we present a universal lower bound on the dispersion stated in [372], which gives a characterization of the minimum attainable dispersion for any point set.

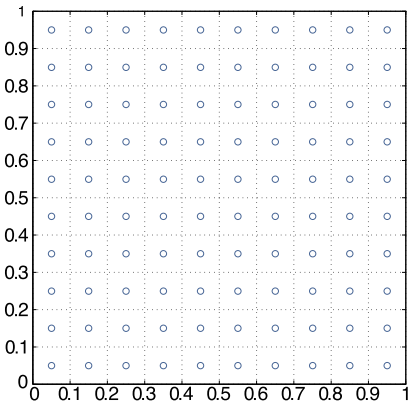

Fig. 7.4 Plot of 100 points chosen according to the Sukharev sampling criterion

Theorem 7.12 (Sukharev inequality) *For any point set* $x^{(1\ldots N)}$ *of cardinality* N, *we have*

$$d_N\left(x^{(1\ldots N)}\right) \geq \frac{1}{2\lfloor N^{1/n} \rfloor}. \tag{7.24}$$

Remark 7.5 (Sukharev sampling criterion) The implications of this result are now briefly discussed. To simplify the discussion, suppose that $N^{1/n}$ is an integer, and suppose we are interested in generating N points in $[0, 1]^n$ with "optimal" dispersion. Then, it can be shown that equality in (7.24) is attained if the points of $x^{(1\ldots N)}$ are placed in a grid with discretization interval $N^{1/n}$ and the first point is shifted of $(N^{1/n})/2$ from the origin. This particular choice of point set is sometimes called the Sukharev sampling criterion, see Fig. 7.4. In addition, we notice that, if we solve Eq. (7.24) for N, we obtain $N \geq (2d_N(x^{(1\ldots N)}))^{-n}$. Hence, the number of points N is an exponential function in n, regardless of how the sample set is generated.

Chapter 8
Probability Inequalities

In this chapter we address the issue of finite sample size, i.e., the so-called *sample complexity*, in probability estimation. That is, we analyze the reliability of the probabilistic estimates introduced in Chap. 7, for *finite* values of N. This issue is crucial in the development of randomized algorithms for uncertain systems and control and makes a clear distinction with the *asymptotic* laws of large numbers preliminarily discussed in Chap. 7.

8.1 Probability Inequalities

This section presents some standard material on probability inequalities, which is the backbone for the sample size bounds subsequently derived in this chapter. The first fundamental result is the Markov inequality.

Markov inequality *Let* $\mathbf{x} \in [0, \infty)$ *be a nonnegative random variable with* $\mathrm{E}(\mathbf{x}) < \infty$. *Then, for any* $\epsilon > 0$, *we have*

$$\mathrm{Pr}\{\mathbf{x} \geq \epsilon\} \leq \frac{\mathrm{E}(\mathbf{x})}{\epsilon}. \tag{8.1}$$

Proof The proof of this result is immediate and follows from the chain of inequalities

$$\mathrm{E}(\mathbf{x}) = \int_0^\infty x f_{\mathbf{x}}(x)\,\mathrm{d}x \geq \int_\epsilon^\infty x f_{\mathbf{x}}(x)\,\mathrm{d}x \geq \epsilon \int_\epsilon^\infty f_{\mathbf{x}}(x)\,\mathrm{d}x = \epsilon \mathrm{Pr}\{\mathbf{x} \geq \epsilon\}. \quad \square$$

Obviously, the Markov inequality, as well as the other inequalities presented in this section, is meaningful only when the right-hand side of (8.1) is not greater than one. We now show that various classical inequalities can be derived from the Markov inequality. To this end, let a and $m > 0$ be two real numbers and observe that the

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_8,

random variable $|\mathbf{x} - a|^m$ is nonnegative. Then, applying the Markov inequality to this random variable, we obtain

$$\Pr\left\{|\mathbf{x} - a|^m \geq \epsilon^m\right\} \leq \frac{\mathrm{E}(|\mathbf{x} - a|^m)}{\epsilon^m}.$$

Taking $a = \mathrm{E}(\mathbf{x})$, we immediately derive to the so-called Bienaymé inequality

$$\Pr\left\{\left|\mathbf{x} - \mathrm{E}(\mathbf{x})\right| \geq \epsilon\right\} \leq \frac{\mathrm{E}(|\mathbf{x} - \mathrm{E}(\mathbf{x})|^m)}{\epsilon^m}. \tag{8.2}$$

The well-known Chebychev inequality is a special case of Bienaymé inequality, obtained for $m = 2$.

Chebychev inequality *Let* $\mathbf{x}$ *be a random variable with* $\mathrm{Var}(\mathbf{x}) < \infty$. *Then, for any* $\epsilon > 0$, *we have*

$$\Pr\left\{\left|\mathbf{x} - \mathrm{E}(\mathbf{x})\right| \geq \epsilon\right\} \leq \frac{\mathrm{Var}(\mathbf{x})}{\epsilon^2}. \tag{8.3}$$

We remark that, while in the Markov inequality $\mathbf{x}$ is a nonnegative random variable, in Bienaymé and Chebychev inequalities there is no sign restriction. However, these latter inequalities hold only when $\mathbf{x}$ has bounded variance or bounded moment of order m.

Remark 8.1 (Historical remarks) The problems concerning the computation of probability inequalities given moments of different order of a random variable have a long history and a rich literature. For example, moment problems have been analyzed in the early works of Stieltjes [370, 371]. An elementary introduction to Chebychev and Markov inequalities is given in [319]. The interested reader may also refer to the original paper of Chebychev [100] and to the thesis of his student Markov [274]. Multivariate generalizations of these inequalities are studied in [276, 387] and in [329], which also contains an historical overview of the topic. Additional related results on large deviation methods can be found in [130].

We now analyze other less well known inequalities. In all these inequalities we assume $\epsilon > 0$. The first one we study was derived by Uspensky [397]

$$\Pr\left\{\mathbf{x} \geq (1 + \epsilon)\mathrm{E}(\mathbf{x})\right\} \leq \frac{\mathrm{Var}(\mathbf{x})}{\mathrm{Var}(\mathbf{x}) + \epsilon^2 \mathrm{E}(\mathbf{x})^2}.$$

This inequality always improves upon the so-called right-sided Chebychev inequality

$$\Pr\left\{\mathbf{x} \geq (1 + \epsilon)\mathrm{E}(\mathbf{x})\right\} \leq \frac{\mathrm{Var}(\mathbf{x})}{\epsilon^2 \mathrm{E}(\mathbf{x})^2}.$$

For completeness, we also state the left-sided Chebychev inequality

$$\Pr\left\{\mathbf{x} \leq (1 - \epsilon)\mathrm{E}(\mathbf{x})\right\} \leq \frac{\mathrm{Var}(\mathbf{x})}{\mathrm{Var}(\mathbf{x}) + \epsilon^2 \mathrm{E}(\mathbf{x})^2}.$$

Other inequalities are proved in [329] for the case when moments up to third order are given. These inequalities are "tight" in a certain sense and have been derived via a general convex optimization reformulation, see [56]. For instance, for a nonnegative random variable $\mathbf{x}$, it is shown that

$$\Pr\{|\mathbf{x} - \mathrm{E}(\mathbf{x})| \geq \epsilon \mathrm{E}(\mathbf{x})\} \leq \min\left(1, 1 + 27\frac{\alpha^2 + \beta^4 - \epsilon^2}{4 + 3(1+3\epsilon^2) + 2(1+3\epsilon^2)^{\frac{3}{2}}}\right)$$

where

$$\alpha = \frac{\mathrm{E}(\mathbf{x})\mathrm{E}(\mathbf{x}^3) - \mathrm{E}(\mathbf{x}^2)}{\mathrm{E}(\mathbf{x})^4} \quad \text{and} \quad \beta = \frac{\mathrm{Var}(\mathbf{x})}{\mathrm{E}(\mathbf{x})^2}.$$

In the next section we study applications of the previous inequalities (and in particular of the Markov inequality) to the problem of bounding the probability of deviation from the mean of sums of random variables.

8.2 Deviation Inequalities for Sums of Random Variables

We here focus our attention on inequalities for the tail probabilities of the *sum of random variables*. That is, we consider N independent random variables $\mathbf{x}_1, \ldots, \mathbf{x}_N$, define the new random variable

$$\mathbf{s}_N \doteq \sum_{i=1}^{N} \mathbf{x}_i$$

and aim to compute bounds on the probability $\Pr\{|\mathbf{s}_N - \mathrm{E}(\mathbf{s}_N)| \geq \epsilon\}$.

A first simple inequality for sums of random variables may be directly derived from the Chebychev inequality, obtaining

$$\Pr\{|\mathbf{s}_N - \mathrm{E}(\mathbf{s}_N)| \geq \epsilon\} \leq \frac{\mathrm{Var}(\mathbf{s}_N)}{\epsilon^2} = \frac{\sum_{i=1}^{N} \mathrm{Var}(\mathbf{x}_i)}{\epsilon^2}. \tag{8.4}$$

A tighter classical inequality, which holds for the case of bounded random variables, is due to Hoeffding [205]. Before stating and proving the Hoeffding inequality, we need to introduce a lemma.

Lemma 8.1 *Let $\mathbf{x} \in [a, b]$ be a random variable with $\mathrm{E}(\mathbf{x}) = 0$. Then, for any $\lambda > 0$*

$$\mathrm{E}(\mathrm{e}^{\lambda \mathbf{x}}) \leq \mathrm{e}^{\lambda^2 (b-a)^2/8}. \tag{8.5}$$

Proof Since $\mathbf{x} \in [a, b]$, we write it as a convex combination of a and b, namely $\mathbf{x} = \eta b + (1 - \eta)a$, where $\eta = (\mathbf{x} - a)/(b - a)$. So, by convexity of $\mathrm{e}^{\lambda \mathbf{x}}$ we have

$$\mathrm{e}^{\lambda \mathbf{x}} \leq \frac{\mathbf{x} - a}{b - a}\mathrm{e}^{\lambda b} + \frac{b - \mathbf{x}}{b - a}\mathrm{e}^{\lambda a}.$$

Taking expectation of both sides and using the fact that $\mathrm{E}(\mathbf{x}) = 0$, we get

$$\mathrm{E}(\mathrm{e}^{\lambda \mathbf{x}}) \leq -\frac{a}{b - a}\mathrm{e}^{\lambda b} + \frac{b}{b - a}\mathrm{e}^{\lambda a} = (1 - p + p\mathrm{e}^{\lambda(b-a)})\mathrm{e}^{-p\lambda(b-a)}$$

where $p = -a/(b-a)$. Next, defining the function

$$L(u) \doteq -pu + \log\bigl(1 - p + p\mathrm{e}^{u}\bigr)$$

we have, for $u = \lambda(b-a)$

$$\mathrm{E}\bigl(\mathrm{e}^{\lambda \mathbf{x}}\bigr) \leq \mathrm{e}^{L(u)}.$$

The first and second derivatives of $L(u)$ are given by

$$L'(u) = -p + \frac{p\mathrm{e}^{u}}{1 - p + p\mathrm{e}^{u}};$$
$$L''(u) = \frac{p(1-p)\mathrm{e}^{u}}{(1 - p + p\mathrm{e}^{u})^2} \leq \frac{1}{4} \quad \text{for all } u > 0.$$

Therefore, using Taylor expansion, we have that for some $\xi \in (0, u)$

$$L(u) = L(0) + uL'(0) + \frac{u^2}{2}L''(\xi) = \frac{u^2}{2}L''(\xi) \leq \frac{u^2}{8} = \frac{\lambda^2(b-a)^2}{8}$$

which proves the lemma. □

We now state the Hoeffding inequality.

Hoeffding inequality *Let $\mathbf{x}_1, \ldots, \mathbf{x}_N$ be independent bounded random variables with $\mathbf{x}_i \in [a_i, b_i]$. Then, for any $\epsilon > 0$, we have*

$$\mathrm{PR}\bigl\{\mathbf{s}_N - \mathrm{E}(\mathbf{s}_N) \geq \epsilon\bigr\} \leq \mathrm{e}^{-2\epsilon^2/\sum_{i=1}^{N}(b_i - a_i)^2} \tag{8.6}$$

and

$$\mathrm{PR}\bigl\{\mathbf{s}_N - \mathrm{E}(\mathbf{s}_N) \leq -\epsilon\bigr\} \leq \mathrm{e}^{-2\epsilon^2/\sum_{i=1}^{N}(b_i - a_i)^2}. \tag{8.7}$$

Proof The inequality is derived using the Chernoff bounding method. That is, for any random variable $\mathbf{x}$, we write the Markov inequality for the random variable $\mathrm{e}^{\lambda \mathbf{x}}$ with $\lambda > 0$, obtaining

$$\mathrm{PR}\bigl\{\mathrm{e}^{\lambda \mathbf{x}} \geq \alpha\bigr\} \leq \frac{\mathrm{E}(\mathrm{e}^{\lambda \mathbf{x}})}{\alpha}$$

for any $\alpha > 0$. Then, taking $\alpha = \mathrm{e}^{\lambda \epsilon}$ and replacing $\mathbf{x}$ with $\mathbf{x} - \mathrm{E}(\mathbf{x})$, we write

$$\mathrm{PR}\bigl\{\mathbf{x} - \mathrm{E}(\mathbf{x}) \geq \epsilon\bigr\} \leq \mathrm{e}^{-\lambda \epsilon}\mathrm{E}\bigl(\mathrm{e}^{\lambda(\mathbf{x} - \mathrm{E}(\mathbf{x}))}\bigr). \tag{8.8}$$

Applying this bound to the random variable $\mathbf{s}_N$, due to the independence of the random variables $\mathbf{x}_i$, we obtain

$$\begin{aligned}\mathrm{PR}\bigl\{\mathbf{s}_N - \mathrm{E}(\mathbf{s}_N) \geq \epsilon\bigr\} &\leq \mathrm{e}^{-\lambda \epsilon}\mathrm{E}\bigl(\mathrm{e}^{\lambda \sum_{i=1}^{N}(\mathbf{x}_i - \mathrm{E}(\mathbf{x}_i))}\bigr) \\ &= \mathrm{e}^{-\lambda \epsilon}\prod_{i=1}^{N}\mathrm{E}\bigl(\mathrm{e}^{\lambda(\mathbf{x}_i - \mathrm{E}(\mathbf{x}_i))}\bigr).\end{aligned} \tag{8.9}$$

To complete the proof, we apply the result of Lemma 8.1 in combination with (8.9), obtaining

$$\Pr\{\mathbf{s}_N - \mathrm{E}(\mathbf{s}_N) \geq \epsilon\} \leq \mathrm{e}^{-\lambda\epsilon} \prod_{i=1}^{N} \mathrm{E}\left(\mathrm{e}^{\lambda(\mathbf{x}_i - \mathrm{E}(\mathbf{x}_i))}\right)$$
$$\leq \mathrm{e}^{-\lambda\epsilon} \prod_{i=1}^{N} \mathrm{e}^{\lambda^2 (b_i - a_i)^2/8} = \mathrm{e}^{-\lambda\epsilon} \mathrm{e}^{\lambda^2 \sum_{i=1}^{N} (b_i - a_i)^2/8}.$$

Inequality (8.6) is obtained by selecting λ such that the exponent is minimized

$$\lambda = \frac{4\epsilon}{\sum_{i=1}^{N} (b_i - a_i)^2}.$$

Inequality (8.7) follows from similar derivations. □

The Hoeffding inequality takes a simpler form in the case when the random variables $\mathbf{x}_i$ are independent and bounded in the same interval $[a, b]$. In this case, combining (8.6) and (8.7), we derive the inequality presented next.

Two-sided Hoeffding inequality *Let* $\mathbf{x}_1, \ldots, \mathbf{x}_N$ *be independent random variables such that* $\mathbf{x}_i \in [a, b]$. *Then, for any* $\epsilon > 0$, *we have*

$$\Pr\left\{\left|\mathbf{s}_N - \mathrm{E}(\mathbf{s}_N)\right| \geq \epsilon\right\} \leq 2\mathrm{e}^{-2\epsilon^2/(N(b-a)^2)}. \tag{8.10}$$

Finally, we state without proof another classical inequality due to Bernstein [55], see additional details and extensions in [52].

Bernstein inequality *Let* $\mathbf{x}_1, \ldots, \mathbf{x}_N$ *be independent random variables with* $\mathbf{x}_i \in [-a, a]$, $\mathrm{E}(\mathbf{x}_i) = 0$ *and* $\mathrm{Var}(\mathbf{x}_i) < \infty$. *Then, for any* $\epsilon > 0$, *we have*

$$\Pr\{\mathbf{s}_N \geq \epsilon\} \leq \mathrm{e}^{-\epsilon^2/(2N\sigma^2 + 2a\epsilon/3)} \tag{8.11}$$

where $\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} \mathrm{Var}(\mathbf{x}_i)$.

Remark 8.2 (Concentration inequalities) More general functions (other than sums) of independent random variables may also be bounded using so-called "concentration" inequalities, such as the Efron–Stein inequality [155] or Talagrand inequality, but this topic goes beyond the scope of this discussion. We address the reader to [66, 377] and the references therein for further details on these issues.

8.3 Sample Complexity for Probability Estimation

In this section we specialize the previous inequalities to derive the sample complexity for the randomized algorithms presented in Chap. 10. In fact, these inequalities are the key tools for determining the minimum number of samples needed to compute the reliability of the estimate

$$\widehat{\mathbf{p}}_N(\gamma) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{\mathcal{B}_G}\left(\boldsymbol{\Delta}^{(i)}\right)$$

of the probability of performance introduced in Chap. 6

$$p(\gamma) = \Pr_{\boldsymbol{\Delta}}\{J(\boldsymbol{\Delta}) \leq \gamma\}.$$

This reliability is measured in terms of the "closeness" of $\widehat{\mathbf{p}}_N(\gamma)$ to the true probability $p(\gamma)$. That is, given $\epsilon \in (0, 1)$, we wish to assure that the event

$$\left|\widehat{\mathbf{p}}_N(\gamma) - p(\gamma)\right| < \epsilon$$

holds with high probability. Since $\widehat{\mathbf{p}}_N(\gamma)$ is estimated via random sampling, it is itself a random variable which depends on the multisample $\boldsymbol{\Delta}^{(1\ldots N)}$. Therefore, for given $\delta \in (0, 1)$, we require that

$$\Pr_{\boldsymbol{\Delta}^{(1\ldots N)}}\left\{\left|\widehat{\mathbf{p}}_N(\gamma) - p(\gamma)\right| < \epsilon\right\} > 1 - \delta. \tag{8.12}$$

The problem is then finding the minimal N such that (8.12) is satisfied for fixed *accuracy* $\epsilon \in (0, 1)$ and *confidence* $\delta \in (0, 1)$.

The first bound on the sample complexity was derived in 1713 by Jacob Bernoulli, see [54], and is reported for historical reasons and for its simplicity.

Bernoulli bound *For any* $\epsilon \in (0, 1)$ *and* $\delta \in (0, 1)$, *if*

$$N \geq \frac{1}{4\epsilon^2\delta} \tag{8.13}$$

then, with probability greater than $1 - \delta$, *we have* $|\widehat{\mathbf{p}}_N(\gamma) - p(\gamma)| < \epsilon$.

Proof This result is proved by means of the Chebychev inequality. The empirical probability $\widehat{\mathbf{p}}_N(\gamma)$ is a random variable binomially distributed, with expected value $\mathrm{E}(\widehat{\mathbf{p}}_N(\gamma)) = p(\gamma)$ and variance $\mathrm{Var}(\widehat{\mathbf{p}}_N(\gamma)) = p(\gamma)(1 - p(\gamma))/N$. Substituting these values in (8.3) for $\mathbf{x} = \widehat{\mathbf{p}}_N(\gamma)$, we obtain

$$\Pr\left\{\left|\widehat{\mathbf{p}}_N(\gamma) - p(\gamma)\right| \geq \epsilon\right\} \leq \frac{p(\gamma)(1 - p(\gamma))}{N\epsilon^2} \leq \frac{1}{4N\epsilon^2}$$

for all $p(\gamma) \in (0, 1)$. The bound (8.13) then follows immediately from this inequality. □

A significant improvement on the previous bound is given by the classical Chernoff bound [104].

Chernoff bound *For any* $\epsilon \in (0, 1)$ *and* $\delta \in (0, 1)$, *if*

$$N \geq \frac{1}{2\epsilon^2}\log\frac{2}{\delta} \tag{8.14}$$

then, with probability greater than $1 - \delta$, *we have* $|\widehat{\mathbf{p}}_N(\gamma) - p(\gamma)| < \epsilon$.

Proof The Chernoff bound follows from direct application of the Hoeffding inequality to the random variables $\mathbf{x}_1, \ldots, \mathbf{x}_N$, defined as

$$\mathbf{x}_i = \mathbb{I}_{\mathcal{B}_G}\left(\boldsymbol{\Delta}^{(i)}\right) = \begin{cases} 1 & \text{if } \boldsymbol{\Delta}^{(i)} \in \mathcal{B}_G; \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, N$. Since $\mathbf{x}_i \in [0, 1]$, letting $\mathbf{s}_N = \sum_{i=1}^{N} \mathbf{x}_i$ and applying inequality (8.10) we get

$$\Pr\left\{\left|\mathbf{s}_N - \mathrm{E}(\mathbf{s}_N)\right| \geq \epsilon\right\} \leq 2\mathrm{e}^{-2\epsilon^2/N}.$$

Now, observing that $\widehat{\mathbf{p}}_N(\gamma) = \mathbf{s}_N/N$ and $\mathrm{E}(\widehat{\mathbf{p}}_N(\gamma)) = p(\gamma)$, we write

$$\Pr\left\{\left|\widehat{\mathbf{p}}_N(\gamma) - p(\gamma)\right| \geq \epsilon\right\} \leq 2\mathrm{e}^{-2N\epsilon^2}$$

from which the desired bound follows immediately. □

Remark 8.3 (Chernoff inequalities) In the proof of the Chernoff bound, applying the two-sided Hoeffding inequality, we obtained the Chernoff inequality

$$\Pr\left\{\left|\widehat{\mathbf{p}}_N(\gamma) - p(\gamma)\right| \geq \epsilon\right\} \leq 2\mathrm{e}^{-2N\epsilon^2} \quad \text{for } \epsilon \in (0, 1).$$

Similarly, applying the one-sided Hoeffding inequalities, we obtain the one-sided Chernoff inequalities

$$\Pr\left\{\widehat{\mathbf{p}}_N(\gamma) - p(\gamma) \geq \epsilon\right\} \leq \mathrm{e}^{-2N\epsilon^2}; \tag{8.15}$$

$$\Pr\left\{\widehat{\mathbf{p}}_N(\gamma) - p(\gamma) \leq -\epsilon\right\} \leq \mathrm{e}^{-2N\epsilon^2} \tag{8.16}$$

for $\epsilon \in (0, 1)$. The above inequalities are often denoted as *additive* Chernoff inequalities to distinguish them from the so-called *multiplicative* Chernoff inequalities, see e.g. [406]. For this reason, the bound (8.14) is often called the *additive Chernoff bound*. The multiplicative Chernoff inequalities take the form

$$\Pr\left\{\widehat{\mathbf{p}}_N(\gamma) - p(\gamma) \geq \epsilon\, p(\gamma)\right\} \leq \mathrm{e}^{-p(\gamma)N\epsilon^2/3}; \tag{8.17}$$

$$\Pr\left\{\widehat{\mathbf{p}}_N(\gamma) - p(\gamma) \leq -\epsilon\, p(\gamma)\right\} \leq \mathrm{e}^{-p(\gamma)N\epsilon^2/2} \tag{8.18}$$

for $\epsilon \in (0, 1)$. Contrary to those in the additive form, we observe that these inequalities are not symmetric. The associated bounds are the so-called *multiplicative Chernoff bounds*. We refer to [406] for additional discussions on these topics.

From the one-sided Chernoff inequality we immediately derive the one-sided Chernoff bound.

One-sided Chernoff bound *For any* $\epsilon \in (0, 1)$ *and* $\delta \in (0, 1)$, *if*

$$N \geq \frac{1}{2\epsilon^2} \log \frac{1}{\delta} \tag{8.19}$$

then, with probability greater than $1 - \delta$, *we have* $\widehat{\mathbf{p}}_N(\gamma) - p(\gamma) < \epsilon$.

We remark that the Chernoff bound largely improves upon the Bernoulli bound. In particular, whereas the sample size in Bernoulli depends on $1/\delta$, the Chernoff bound is a function of $\log(2/\delta)$. However, in both cases, the dependence on ϵ is unchanged and it is inversely proportional to ϵ^2. We conclude that confidence is "cheaper" than accuracy. Table 8.1 shows a comparison between these bounds for several values of ϵ and δ.

Table 8.1 Comparison of the sample size obtained with Bernoulli and Chernoff bounds for different values of ϵ and δ

ϵ	$1-\delta$	Bernoulli	Chernoff
0.05	0.95	2000	738
	0.99	1.00×10^4	1060
	0.995	2.00×10^4	1199
	0.999	1.00×10^5	1521
0.01	0.95	5.00×10^4	1.84×10^4
	0.99	2.50×10^5	2.65×10^4
	0.995	5.00×10^5	3.00×10^4
	0.999	2.50×10^6	3.80×10^4
0.005	0.95	2.00×10^5	7.38×10^4
	0.99	1.00×10^6	1.06×10^5
	0.995	2.00×10^6	1.20×10^5
	0.999	1.00×10^7	1.52×10^5
0.001	0.95	5.00×10^6	1.84×10^6
	0.99	2.50×10^7	2.65×10^6
	0.995	5.00×10^7	3.00×10^6
	0.999	2.50×10^8	3.80×10^6

We observe that the bounds discussed in these sections can be computed *a priori* and are explicit. That is, given ϵ and δ one can find directly the minimum value of N without generating the samples $\boldsymbol{\Delta}^{(1\dots N)}$ and evaluating $J(\boldsymbol{\Delta}^{(i)})$ for $i=1,\dots,N$. On the other hand, when computing the classical *confidence intervals*, see e.g. [111], the sample size obtained is not explicit. More precisely, for given $\delta\in(0,1)$, the lower and upper confidence intervals $\mathbf{p}_L$ and $\mathbf{p}_U$ are such that

$$\mathrm{PR}_{\boldsymbol{\Delta}^{(1\dots N)}}\left\{\mathbf{p}_L\le p(\gamma)\le\mathbf{p}_U\right\}>1-\delta.$$

The evaluation of this probability requires the solution with respect to $\mathbf{p}_L$ and $\mathbf{p}_U$ of equations of the type

$$\sum_{k=\mathbf{N}_G}^{N}\binom{N}{k}{\mathbf{p}_L}^k(1-\mathbf{p}_L)^{N-k}=\delta_L; \tag{8.20}$$

$$\sum_{k=0}^{\mathbf{N}_G}\binom{N}{k}{\mathbf{p}_U}^k(1-\mathbf{p}_U)^{N-k}=\delta_U \tag{8.21}$$

with $\delta_L+\delta_U=\delta$, where $\mathbf{N}_G$ is the number of samples such that $J(\boldsymbol{\Delta}^{(i)})\le\gamma$. Clearly, the probabilities $\mathbf{p}_L$ and $\mathbf{p}_U$ are random variables which can be computed only *a posteriori*, once the value of $\mathbf{N}_G$ is known. Moreover, an explicit solution of the previous equations is not available, so that standard tables or numerical methods are generally used; see e.g. [332]. Figure 8.1 shows the confidence intervals

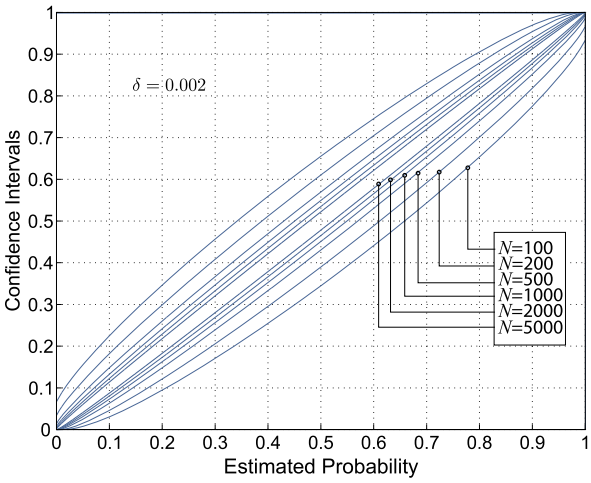

Fig. 8.1 Confidence intervals for $\delta = 0.002$

for $\delta = 0.002$ and various values of N. The figure should be interpreted as follows: if, for instance, $N = 1{,}000$ and $N_G = 700$, then the estimated probability is $\widehat{p}_N(\gamma) = N_G/N = 0.7$ and the values $p_U = 0.74$, $p_L = 0.65$ can be obtained from the plot.

8.4 Sample Complexity for Estimation of Extrema

The sample complexity considered in the previous section applies to the estimation of probabilities, which are essentially expectations of random variables defined by means of indicator functions. In this section, we deal instead with sample bounds that apply to the estimation of extrema (as opposed to expectation) of a random variable. Computing the extremum of a function is directly related to the issue of assessing the worst-case performance of a system, as defined in Problem 6.2. In particular, we consider the problem of computing a probabilistic estimate of the worst-case performance

$$\gamma_{\text{wc}} = \sup_{\Delta \in \mathcal{B}_{\mathbb{D}}} J(\Delta).$$

To this end, we introduce a random sampling scheme, generating N iid samples of $\boldsymbol{\Delta}$ according to $f_{\boldsymbol{\Delta}}(\Delta)$ with support $\mathcal{B}_{\mathbb{D}}$, and define the empirical maximum

$$\widehat{\boldsymbol{\gamma}}_N = \max_{i=1,\ldots,N} J\left(\boldsymbol{\Delta}^{(i)}\right).$$

For this specific problem, a bound on the sample size, derived in [233] and [382], is now given.

Theorem 8.1 (Sample size bound for worst-case performance) *For any* $\epsilon \in (0, 1)$ *and* $\delta \in (0, 1)$, *if*

$$N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}} \tag{8.22}$$

then, with probability greater than $1 - \delta$, *we have*

$$\Pr_{\boldsymbol{\Delta}}\left\{J(\boldsymbol{\Delta}) \leq \widehat{\boldsymbol{\gamma}}_N\right\} \geq 1 - \epsilon.$$

That is

$$\Pr_{\boldsymbol{\Delta}^{(1\ldots N)}}\left\{\Pr_{\boldsymbol{\Delta}}\left\{J(\boldsymbol{\Delta}) \leq \widehat{\boldsymbol{\gamma}}_N\right\} \geq 1 - \epsilon\right\} > 1 - \delta.$$

Proof Let α be the minimum value in the interval $[\inf_{\boldsymbol{\Delta} \in \mathcal{B}_{\mathbb{D}}} J(\boldsymbol{\Delta}), \gamma_{\text{wc}}]$ such that $F_{J(\boldsymbol{\Delta})}(\alpha) \geq 1 - \epsilon$, where $F_{J(\boldsymbol{\Delta})}(\cdot)$ is the distribution function of the random variable $J(\boldsymbol{\Delta})$. Notice that α always exists, since $F_{J(\boldsymbol{\Delta})}(\cdot)$ is right continuous. Now, we have

$$\begin{aligned}\Pr\left\{F_{J(\boldsymbol{\Delta})}(\widehat{\boldsymbol{\gamma}}_N) \geq 1 - \epsilon\right\} &= \Pr\{\widehat{\boldsymbol{\gamma}}_N \geq \alpha\} \\ &= 1 - \Pr\{\widehat{\boldsymbol{\gamma}}_N < \alpha\} = 1 - F_{J(\boldsymbol{\Delta})}\left(\alpha^-\right)^N\end{aligned}$$

where $F_{J(\boldsymbol{\Delta})}(\alpha^-)$ is the limit of $F_{J(\boldsymbol{\Delta})}(\alpha)$ from the left. In addition, observe that $F_{J(\boldsymbol{\Delta})}(\alpha^-) \leq 1 - \epsilon$. Then

$$\Pr\left\{F_{J(\boldsymbol{\Delta})}(\widehat{\boldsymbol{\gamma}}_N) \geq 1 - \epsilon\right\} \geq 1 - (1 - \epsilon)^N.$$

Next, notice that if (8.22) holds, then $(1 - \epsilon)^N \leq \delta$. Thus

$$\Pr\left\{F_{J(\boldsymbol{\Delta})}(\widehat{\boldsymbol{\gamma}}_N) \geq 1 - \epsilon\right\} \geq 1 - \delta.$$

This completes the proof. □

It is shown in [103] that the bound (8.22) is tight if the distribution function is continuous. This is a consequence of the fact that the bound on the sample size is minimized if and only if

$$\sup_{\{\gamma : F_{J(\boldsymbol{\Delta})}(\gamma) \leq 1-\epsilon\}} F_{J(\boldsymbol{\Delta})}(\gamma) = 1 - \epsilon.$$

Comparing the bound (8.22) with the Chernoff bound, it can be observed that in the former case the bound grows as $1/\epsilon$, since $\log(1/(1 - \epsilon)) \approx \epsilon$, whereas in the latter case it grows as $1/\epsilon^2$.

This fact leads to a major reduction in the number of samples needed, as shown in Fig. 8.2, which compares the worst-case bound of Theorem 8.1 with the Chernoff bound for various values of ϵ and δ. This is not surprising, since in the worst-case bound the performance level is selected *a posteriori*, once the samples are generated and $\widehat{\boldsymbol{\gamma}}_N$ is computed, while the Chernoff bound holds for any a priori specified value of γ. Note however that the two bounds apply to different problems. Namely, the Chernoff bound applies to estimation of expected values or probabilities, while the bound in Theorem 8.1 applies to the estimation of extrema.

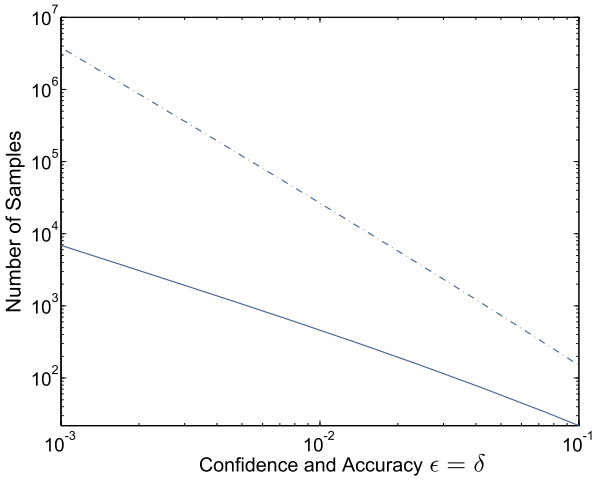

Fig. 8.2 Comparison between the Chernoff bound (*dash-dotted line*) and the worst-case bound (*solid line*)

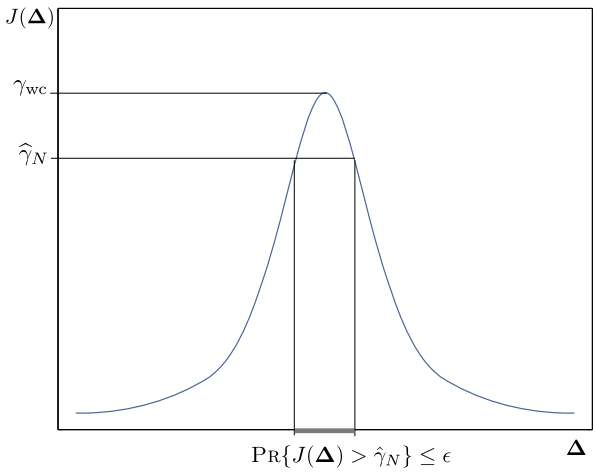

Fig. 8.3 Interpretation of the worst-case bound

Remark 8.4 (Worst-case bound interpretation) In general, there is no assurance that $\widehat{\boldsymbol{\gamma}}_N$ is actually close to the maximum γ_{wc}. As shown in Fig. 8.3, the bound previously discussed only guarantees (in probability) that the performance is less than $\widehat{\boldsymbol{\gamma}}_N$ with high probability $1 - \epsilon$. In other words, the set of points greater than the estimated value has a measure smaller than ϵ, and this is true with a probability at least $1 - \delta$. In turn, this implies that, if the function $J(\Delta)$ is sufficiently smooth, then the estimated and actual maximum may be close. For this reason, care should be exercised when the bound is applied for solving optimization problems with a randomized approach.

Remark 8.5 (Bounds for one level of probability) We notice that [36] investigates the minimal sample size problem with one level of probability, when the performance function J is in the class of all Lipschitz continuous functions $\mathcal{J}_L$. More

precisely, the problem addressed in [36] is to compute the minimum number of samples required to satisfy

$$\mathrm{PR}_{\boldsymbol{\Delta}^{(1\ldots N)}}\left\{\sup_{J\in\mathcal{J}_L}|\gamma_{\mathrm{wc}}-\widehat{\boldsymbol{\gamma}}_N|\leq\epsilon\right\}\geq 1-\delta.$$

In the same paper, it is also shown that the uniform distribution is "optimal" in the sense of minimizing the sample size, but the sample size obtained may be an exponential function of the number of variables and, therefore, computational complexity becomes a critical issue. Contrary to the bounds previously discussed, in this line of research the performance function is not fixed, rather it varies within the class $\mathcal{J}_L$.

In [119], the authors derive a one level of probability bound on the expected number of samples that have to be drawn for approximating the maximum of a linear functional on a convex body with precision α. In particular, it is shown that this number grows exponentially in the dimensionality of the problem, see Sect. 14.4.3 for further details.

8.5 Sample Complexity for the Binomial Tail

This binomial distribution plays a key role in many problems. In our context, for example, it arises when establishing tight bounds on the violation probability in the so-called scenario optimization approach discussed in Chap. 12. Here, we study sample cardinality bounds for the tail of the binomial distribution, see Eq. (2.4) in Sect. 2.3

$$\mathrm{B}_{N,d}(\epsilon)=\sum_{i=0}^{d}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i}$$

where $d\leq N$ is a nonnegative integer and $\epsilon\in(0,1)$ is the accuracy. The objective is to determine a function $\widetilde{N}(\epsilon,d,\delta)$ such that the inequality $\mathrm{B}_{N,d}(\epsilon)\leq\delta$ holds for any $N\geq\widetilde{N}(\epsilon,d,\delta)$, where $\delta\in(0,1)$ is a confidence parameter.

The following technical lemma, stated in [10], provides an upper bound for the binomial distribution $\mathrm{B}_{N,d}(\epsilon)$.

Lemma 8.2 *Let N, d be nonnegative integers with $N\geq d$, and $\epsilon\in(0,1)$. Then,*

$$\mathrm{B}_{N,d}(\epsilon)=\sum_{i=0}^{d}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i}\leq a^d\left(\frac{\epsilon}{a}+1-\epsilon\right)^N$$

for all $a\geq 1$.

We notice that each particular choice of the parameter $a\geq 1$ provides an upper bound for the binomial density $\mathrm{B}_{N,d}(\epsilon)$.

Lemma 8.3 *Let N, d be nonnegative integers with $N \geq d$, $a > 1$, $\epsilon \in (0,1)$ and $\delta \in (0,1)$. If*

$$N \geq \frac{1}{\epsilon}\left(\frac{a}{a-1}\right)\left(\log\frac{1}{\delta} + d\log a\right)$$

then

$$\mathrm{B}_{N,d}(\epsilon) = \sum_{i=0}^{d}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i} \leq \delta.$$

Obviously, the best sample size bound is obtained taking the infimum with respect to $a > 1$. However, a suboptimal value easily follows setting a equal to the Euler number e which provides the sample complexity

$$N \geq \frac{1}{\epsilon}\left(\frac{\mathrm{e}}{\mathrm{e}-1}\right)\left(\log\frac{1}{\delta} + d\right).$$

Chapter 9
Statistical Learning Theory

In this chapter, we provide an overview of *statistical learning theory* and we describe some results regarding uniform convergence of empirical means and related sample complexity. This theory provides a fundamental extension of the probability inequalities studied in Chap. 8 to the case when parameterized *families* of performance functions are considered, instead of a *fixed* function. For an advanced treatment of this topic, the interested reader may refer to [132, 148, 270, 363, 401, 406]. In Chap. 13, we study applications of statistical learning theory to control systems design.

9.1 Deviation Inequalities for Finite Families

In Chap. 8 we studied several deviation-type inequalities for random variables. Here, we begin by reconsidering the two-sided Hoeffding inequality, applied to the general case of expected value estimation. Consider a performance function $J : \mathbb{D} \to [0, 1]$, and its empirical mean

$$\widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big) = \frac{1}{N}\sum_{i=1}^{N} J\big(\boldsymbol{\Delta}^{(i)}\big)$$

computed using a multisample $\boldsymbol{\Delta}^{(1\ldots N)}$ of cardinality N. Then, it immediately follows from (8.10) that the probability of deviation of the empirical mean from the actual one is bounded as

$$\Pr\Big\{\big|\mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big)\big| \geq \epsilon\Big\} \leq 2\mathrm{e}^{-2N\epsilon^2}. \tag{9.1}$$

It should be emphasized that this inequality holds for a *fixed* performance function J. On the other hand, if we wish to consider m performance functions simultaneously, we need to define a *finite class* of functions, consisting of m elements

$$\mathcal{J}_m \doteq \{J_1, \ldots, J_m\}$$

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_9,

where $J_i : \mathbb{D} \to [0, 1]$, $i = 1, \ldots, m$. We now aim at bounding the probability of deviation of the empirical mean from the actual one for all functions in the considered class. This worst-case probability

$$q(\mathcal{J}_m, N, \epsilon) = \Pr\left\{ \sup_{J \in \mathcal{J}_m} \left| \mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big) \right| > \epsilon \right\}$$

can be bounded by repeated application of the inequality (9.1), obtaining

$$q(\mathcal{J}_m, N, \epsilon) \leq 2m\mathrm{e}^{-2N\epsilon^2}. \tag{9.2}$$

Notice that this result is distribution free, i.e. the actual distribution of the data does not play any role in the upper bound. A similar bound on the expected value of the maximal deviation

$$\mathrm{E}_{\boldsymbol{\Delta}^{(1\ldots N)}}\left(\sup_{J \in \mathcal{J}_m} \left| \mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big) \right| \right)$$

can be derived. This bound is reported in the following lemma, whose proof may be found for instance in [270].

Lemma 9.1 *Let $\mathcal{J}_m$ be a family of performance functions of finite cardinality m. Then*

$$\mathrm{E}_{\boldsymbol{\Delta}^{(1\ldots N)}}\left(\sup_{J \in \mathcal{J}_m} \left| \mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big) \right| \right) \leq \sqrt{\frac{\log(2m)}{2N}}.$$

9.2 Vapnik–Chervonenkis Theory

We remark that the inequality (9.2) implies that the probability $q(\mathcal{J}_m, N, \epsilon)$ approaches zero asymptotically as N goes to infinity for all $\epsilon > 0$. This convergence property is therefore satisfied for any family of performance functions $\mathcal{J}_m$ having finite cardinality. The question that naturally arises is whether *infinite* families of performance functions enjoy the same property. In this case, the finite class of functions $\mathcal{J}_m$ is replaced by the infinite family $\mathcal{J}$ of measurable binary-valued functions mapping $\mathbb{R}^n$ into $\{0, 1\}$. Similarly, the probability $q(\mathcal{J}_m, N, \epsilon)$ is replaced by the probability

$$q(\mathcal{J}, N, \epsilon) = \Pr\left\{ \sup_{J \in \mathcal{J}} \left| \mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big) \right| > \epsilon \right\}.$$

This probability is sometimes called *probability of two-sided failure* and the convergence property we are interested in is denoted as *uniform convergence of empirical means* (UCEM), which is now formally introduced.

Definition 9.1 (UCEM) Consider a family $\mathcal{J}$ of measurable binary-valued functions mapping $\mathbb{R}^n$ into $\{0, 1\}$. The class $\mathcal{J}$ enjoys the uniform convergence of empirical means (UCEM) property if $q(\mathcal{J}, N, \epsilon) \to 0$ as $N \to \infty$ for any $\epsilon > 0$.

The problem of establishing the UCEM property and deriving bounding inequalities for the probability of two-sided failure is the focal point of statistical learning theory, which was initiated by Vapnik and Chervonenkis in their seminal paper [402]. In this chapter, we report some of the key results of this theory, with particular attention to derive the sample complexity.

Note that, associated to the family of functions $\mathcal{J}$, we may define the corresponding class $\mathcal{S}_J$, whose elements are the sets

$$S_J \doteq \{\Delta \in \mathcal{B} : J(\Delta) = 1\}, \quad J \in \mathcal{J}$$

where $\mathcal{B}$ is a generic set in $\mathbb{R}^n$. In the following, we use families of functions or families of sets interchangeably, depending on which one is more convenient in the context. In fact, given a family of functions $\mathcal{J}$, we construct the corresponding family of sets $\mathcal{S}_J$ as shown above. Conversely, to a family of measurable sets $\mathcal{S}$ we associate the family of binary-valued functions $\mathcal{J}$ whose elements are

$$J(\Delta) \doteq \mathbb{I}_S(\Delta), \quad S \in \mathcal{S}.$$

Notice that with these definitions we have

$$\Pr_{\boldsymbol{\Delta}}\{\boldsymbol{\Delta} \in S_J\} = \mathrm{E}_{\boldsymbol{\Delta}}\big(J(\boldsymbol{\Delta})\big).$$

Consider a point set $\Delta^{(1\ldots N)} = \{\Delta^{(1)}, \ldots, \Delta^{(N)}\}$ of cardinality N. For a given family $\mathcal{J}$, let

$$\mathbb{N}_{\mathcal{J}}\big(\Delta^{(1\ldots N)}\big) \doteq \mathrm{Card}\big(\Delta^{(1\ldots N)} \cap S_J, S_J \in \mathcal{S}_J\big)$$

be the number of different subsets of $\Delta^{(1\ldots N)}$ obtained intersecting $\Delta^{(1\ldots N)}$ with the elements of $\mathcal{S}_J$. When $\mathbb{N}_{\mathcal{J}}(\Delta^{(1\ldots N)})$ equals the maximum number of possible different subsets of $\Delta^{(1\ldots N)}$, which is 2^N, we say that $\mathcal{S}_J$ *shatters* the set $\Delta^{(1\ldots N)}$. We can hence define the "shatter coefficient" of a set of points $\Delta^{(1\ldots N)}$ as follows.

Definition 9.2 (Shatter coefficient) Let $\mathcal{J}$ be a family of measurable functions $\mathbb{R}^n \to \{0, 1\}$. The *shatter coefficient* of the family $\mathcal{J}$, or equivalently of the family of sets $\mathcal{S}_J$, is defined as

$$\mathbb{S}_{\mathcal{J}}(N) \doteq \max_{\Delta^{(1\ldots N)}} \mathbb{N}_{\mathcal{J}}\big(\Delta^{(1\ldots N)}\big). \tag{9.3}$$

Thus, $\mathbb{S}_{\mathcal{J}}(N)$ is the maximum number of different subsets of any point set $\Delta^{(1\ldots N)}$ of cardinality N that can be obtained by intersecting $\Delta^{(1\ldots N)}$ with elements of $\mathcal{S}_J$.

Example 9.1 (Shatter coefficient of a family of half-spaces in $\mathbb{R}^2$) Consider the family $\mathcal{J}$ of binary-valued functions $J(\Delta)$ mapping $\mathbb{R}^2$ into $\{0, 1\}$ of the form

$$J(\Delta) = \begin{cases} 1 & \text{if } \theta_1^T \Delta + \theta_2 \geq 0; \\ 0 & \text{otherwise} \end{cases}$$

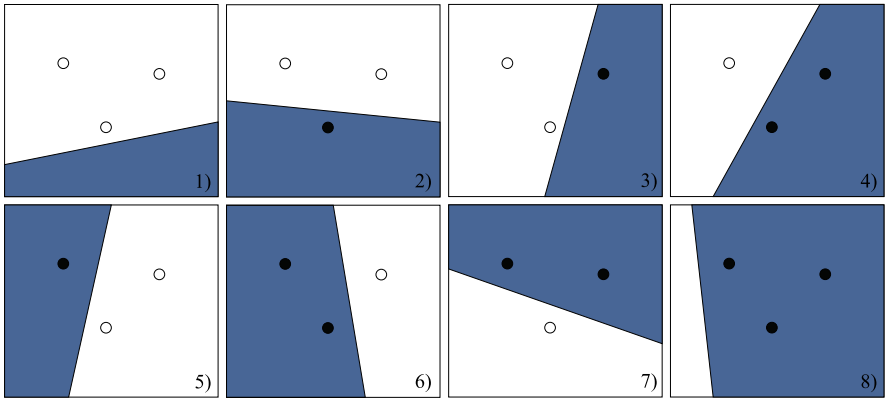

Fig. 9.1 A family of half-spaces shatters a point set of cardinality three

with parameters $\theta_1 \in \mathbb{R}^2$, $\theta_2 \in \mathbb{R}$. We associate to the family $\mathcal{J}$ the family of sets $\mathcal{S}_J$ formed by all possible linear half-spaces of $\mathbb{R}^2$. Then, in Fig. 9.1, we consider a point set $\Delta^{(1,2,3)} = \{\Delta^{(1)}, \Delta^{(2)}, \Delta^{(3)}\}$ of cardinality three and observe that

$$\mathbb{N}_{\mathcal{J}}\big(\Delta^{(1,2,3)}\big) = 8 = 2^3.$$

Therefore, we say that the family $\mathcal{S}_J$ shatters the point set, and $\mathbb{S}_{\mathcal{J}}(3) = 8$. Similarly, in Fig. 9.2 we consider a point set $\Delta^{(1,\ldots,4)}$ of cardinality four and observe that

$$\mathbb{N}_{\mathcal{J}}\big(\Delta^{(1,\ldots,4)}\big) = 14.$$

Since $14 < 2^4$, we conclude that the set considered is *not* shattered by the family. It can indeed be shown that the shatter coefficient of the family is $\mathbb{S}_{\mathcal{J}}(4) = 14$, i.e. it is not possible to find *any* point set of cardinality four that can be shattered by $\mathcal{S}_J$.

The celebrated Vapnik–Chervonenkis inequality [402] states a bound on the probability of two-sided failure for any $\epsilon > 0$

$$q(\mathcal{J}, N, \epsilon) \leq 4\,\mathbb{S}_{\mathcal{J}}(2N)\,\mathrm{e}^{-N\epsilon^2/8}. \tag{9.4}$$

The Vapnik–Chervonenkis inequality therefore provides a bound on the uniform deviation of empirical means in terms of the combinatorial parameter $\mathbb{S}_{\mathcal{J}}(N)$. This parameter can be interpreted as a measure of the "richness" of the class of functions $\mathcal{J}$. An important issue therefore is the computation of explicit upper bounds for $\mathbb{S}_{\mathcal{J}}(N)$, so that the probability $q(\mathcal{J}, N, \epsilon)$ can be suitably bounded. To this end, we now define the Vapnik–Chervonenkis dimension VC of a family of binary-valued functions.

Definition 9.3 (VC dimension) Let $\mathcal{J}$ be a family of measurable binary-valued functions mapping $\mathbb{R}^n$ into $\{0, 1\}$. The VC dimension $\mathrm{VC}(\mathcal{J})$ of $\mathcal{J}$ is defined as the largest integer k such that $\mathbb{S}_{\mathcal{J}}(k) = 2^k$. If $\mathbb{S}_{\mathcal{J}}(k) = 2^k$ for all k, then we say that $\mathrm{VC}(\mathcal{J}) = \infty$.

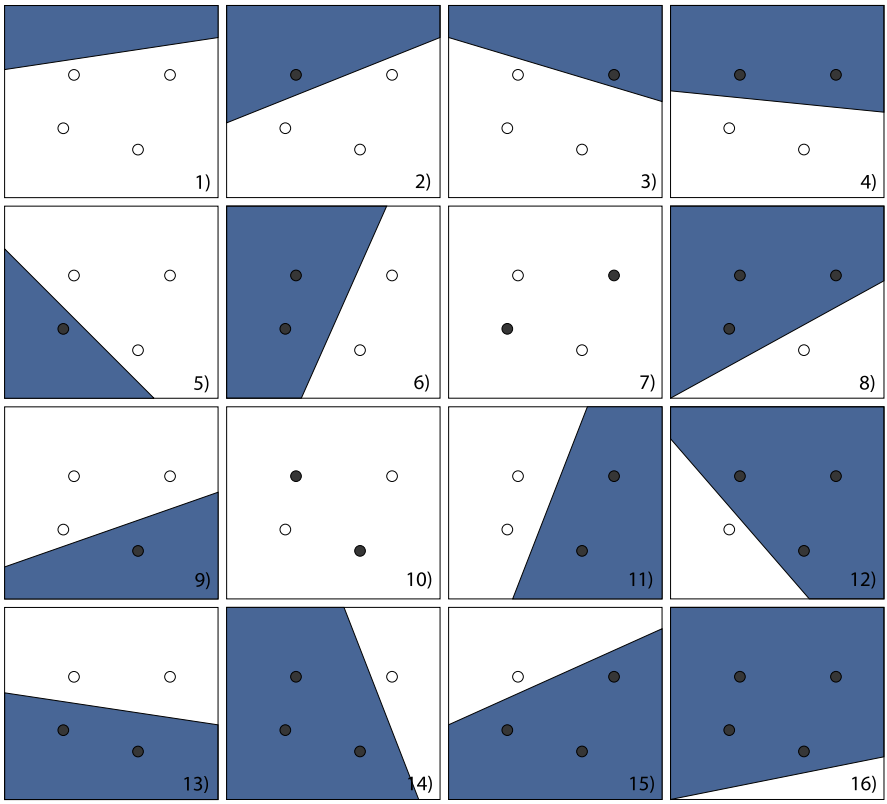

Fig. 9.2 A point set of cardinality four is not shattered by $\mathcal{S}_J$. In fact, the subsets in (7) and (10) cannot be obtained intersecting an element of $\mathcal{S}_J$ with $\Delta^{(1,\ldots,4)}$

In other words, the VC dimension of $\mathcal{J}$ is the largest integer N such that there exists a set of cardinality N that is shattered by $\mathcal{J}$.

Example 9.2 (VC dimension of a family of half-spaces in $\mathbb{R}^2$) In Example 9.1 we considered a family of half-spaces in $\mathbb{R}^2$ and showed that the maximum cardinality of a point set shattered by $\mathcal{S}_J$ is three. We hence conclude that the VC dimension of this family is $\text{VC}(\mathcal{J}) = 3$.

The VC dimension is used to determine a bound on the shatter coefficient, by means of the so-called Sauer lemma, see [343].

Lemma 9.2 (Sauer) *Let $\mathcal{J}$ be a family of measurable binary-valued functions $\mathbb{R}^n \to \{0, 1\}$ and let $\text{VC}(\mathcal{J}) \leq d < \infty$. Then*

$$\mathbb{S}_{\mathcal{J}}(N) \leq \sum_{i=0}^{d} \binom{N}{i}.$$

Moreover, for all $N \geq d$, we have

$$\sum_{i=0}^{d} \binom{N}{i} \leq \left(\frac{N\mathrm{e}}{d}\right)^{d}.$$

One of the direct consequences of this result is that, under the assumption of finite VC dimension, $\mathbb{S}_{\mathcal{J}}(N)$ is bounded by a polynomial function of N. In turn, this implies that

$$\mathbb{S}_{\mathcal{J}}(2N)\mathrm{e}^{-N\epsilon^2/8} \to 0 \quad \text{for } N \to \infty.$$

Bearing in mind the Vapnik–Chervonenkis inequality (9.4), if the VC dimension is bounded, we easily obtain

$$q(\mathcal{J}, N, \epsilon) \to 0 \quad \text{for } N \to \infty.$$

Furthermore, combining Sauer lemma and the inequality (9.4), we obtain the following fundamental result stated in [402].

Theorem 9.1 (Vapnik–Chervonenkis) *Let $\mathcal{J}$ be a family of measurable binary-valued functions mapping $\mathbb{R}^n$ into $\{0, 1\}$. Suppose $\mathcal{J}$ has finite VC dimension $\mathrm{VC}(\mathcal{J}) \leq d < \infty$. Then, $\mathcal{J}$ has the UCEM property. In particular, for any $\epsilon > 0$ and $N \geq d$*

$$q(\mathcal{J}, N, \epsilon) \leq 4\left(\frac{2\mathrm{e}N}{d}\right)^{d} \mathrm{e}^{-N\epsilon^2/8}.$$

Conversely, if $\mathcal{J}$ has the UCEM property, then its VC dimension is finite.

As discussed in [9], there exist other results in the literature that allow one to reduce by a factor close to 8 the exponent. Basically, these results differ from Eq. (9.4) in the exponent $-N\epsilon^2/8$ which is replaced by less conservative ones, with almost no increase in the other constants appearing in the bounds. For example, the exponent $-N\epsilon^2$ can be found in [321] and [401]. To obtain more significant improvements, we now introduce other notions of probability of failure. This requires the introduction a *level* parameter $\beta \in [0, 1)$, which allows broadening the class of problems that are studied. First, we define the probability of *one-sided constrained failure* as follows

$$\begin{aligned} &p(\mathcal{J}, N, \epsilon, \beta) \\ &\quad = \Pr\left\{\exists J \in \mathcal{J} : \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big) \leq \beta \text{ and } \mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big) > \epsilon\right\}. \end{aligned} \tag{9.5}$$

Notice that taking $\beta > 0$ allows one to consider probabilistic constraints of the form $\widehat{\mathbf{E}}_N(J(\boldsymbol{\Delta})) \leq \beta$, which will turn useful in the learning-theory approach to control discussed in Chap. 13. Then, we define the probability of *relative difference failure* as

$$r(\mathcal{J}, N, \epsilon) = \Pr\left\{\sup_{J \in \mathcal{J}} \frac{\mathrm{E}(J(\boldsymbol{\Delta})) - \widehat{\mathbf{E}}_N(J(\boldsymbol{\Delta}))}{\sqrt{\mathrm{E}(J(\boldsymbol{\Delta}))}} > \epsilon\right\}.$$

The next theorem summarizes the results [401] and [9], and shows the specific relations between these probabilities.

Theorem 9.2 *Let $\mathcal{J}$ be a family of measurable binary-valued functions mapping $\mathbb{R}^n$ into $\{0,1\}$. Suppose $\mathcal{J}$ has finite VC dimension $\mathrm{VC}(\mathcal{J}) \leq d < \infty$. Then, for any $\epsilon \in (0,1)$, $\beta \in [0,1)$ and $N \geq d$*

$$p(\mathcal{J}, N, \epsilon, \beta) \leq q(\mathcal{J}, N, \epsilon) < 4\mathrm{e}^{2\epsilon}\left(\frac{2\mathrm{e}N}{d}\right)^d \mathrm{e}^{-N\epsilon^2};$$

$$p(\mathcal{J}, N, \epsilon, \beta) \leq r\left(\mathcal{J}, N, \frac{\epsilon}{\sqrt{\epsilon+\beta}}\right);$$

$$r(\mathcal{J}, N, \epsilon) < 4\left(\frac{2\mathrm{e}N}{d}\right)^d \mathrm{e}^{-N\epsilon^2/4}.$$

To conclude this section, we present an extension given in [270] to infinite families of Lemma 9.1 on the expectation of maximal deviations.

Lemma 9.3 *Let $\mathcal{J}$ be a family of measurable binary-valued functions mapping $\mathbb{R}^n$ into $\{0,1\}$. Suppose $\mathcal{J}$ has finite VC dimension $\mathrm{VC}(\mathcal{J}) \leq d < \infty$. Then, for any $\epsilon \in (0,1)$ and $N \geq d$*

$$\mathrm{E}_{\boldsymbol{\Delta}^{(1\ldots N)}}\left(\sup_{J\in\mathcal{J}}\left|\mathrm{E}\big(J(\boldsymbol{\Delta})\big) - \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta})\big)\right|\right) \leq 2\sqrt{\frac{d\log(N+1)+\log 2}{N}}.$$

9.3 Sample Complexity for the Probability of Failure

In this section, we focus on the derivation of explicit bounds on the number of samples required to guarantee that the probability of failure is bounded by given confidence $\delta \in (0,1)$. The first result in this direction for statistical learning theory has been proved in [405] and is stated next.

Theorem 9.3 *Let $\mathcal{J}$ be a family of measurable binary-valued functions mapping $\mathbb{R}^n$ into $\{0,1\}$. Suppose $\mathcal{J}$ has finite VC dimension $\mathrm{VC}(\mathcal{J}) \leq d < \infty$. For any $\epsilon \in (0,1)$ and $\delta \in (0,1)$, if*

$$N \geq \max\left\{\frac{16}{\epsilon^2}\log\frac{4}{\delta}, \frac{32d}{\epsilon^2}\log\frac{32\mathrm{e}}{\epsilon^2}\right\}$$

then $q(\mathcal{J}, N, \epsilon) \leq \delta$.

This theorem is based on "inverting" Sauer lemma. More generally, the problem addressed in the next result, see [9] for proof, can be stated as follows: given $\delta \in (0,1)$, and real constants a, b, and c, we aim at computing $N \geq d$ satisfying the

inequality

$$a\left(\frac{ceN}{d}\right)^d e^{-bN} < \delta.$$

Lemma 9.4 *Suppose that $a \geq 1$, $b \in (0, 1]$, $c \geq 1$, $d \geq 1$ and $\delta \in (0, 1)$ are given. Then, $N \geq d$ and*

$$a\left(\frac{ceN}{d}\right)^d e^{-bN} < \delta$$

provided that

$$N \geq \inf_{\mu>1} \frac{\mu}{b(\mu-1)}\left(\log\frac{a}{\delta} + d\log\frac{c\mu}{b}\right).$$

Using this lemma, an alternative explicit bound to that presented in Theorem 9.3 can be obtained. From the Vapnik–Chervonenkis theorem, to achieve a probability of two-sided failure smaller than δ, it suffices to choose N such that

$$4\left(\frac{2eN}{d}\right)^d e^{-N\epsilon^2/8} < \delta.$$

Taking $a = 4$, $b = \epsilon^2/8$, $c = 2$ and applying Theorem 9.4, we easily obtain the bound

$$N \geq \inf_{\mu>1} \frac{8}{\epsilon^2}\left(\frac{\mu}{\mu-1}\right)\left(\log\frac{4}{\delta} + d\log\frac{16\mu}{\epsilon^2}\right).$$

Clearly, a more conservative bound may be computed if, instead of evaluating the infimum with respect to $\mu > 1$, a suboptimal value of μ is chosen. For example, setting μ equal to 2e, we conclude that it suffices to take

$$N \geq \frac{9.81}{\epsilon^2}\left(\log\frac{4}{\delta} + d\log\frac{32e}{\epsilon^2}\right).$$

We remark that this bound improves upon Theorem 9.3. Using Theorems 9.2 and 9.4, in the next theorem we provide further improvements.

Theorem 9.4 *Let $\mathcal{J}$ be a family of measurable binary-valued functions mapping $\mathbb{R}^n$ into $\{0, 1\}$. Suppose $\mathcal{J}$ has finite VC dimension $\mathrm{VC}(\mathcal{J}) \leq d < \infty$. For any $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$, if*

$$N \geq \frac{1.2}{\epsilon^2}\left(\log\frac{4e^{2\epsilon}}{\delta} + d\log\frac{12}{\epsilon^2}\right)$$

then $q(\mathcal{J}, N, \epsilon) < \delta$.

The Vapnik–Chervonenkis theory provides both necessary and sufficient conditions for the UCEM property to hold. Therefore, improvements may be obtained only in the derivation of less conservative estimates for the sample-complexity given

accuracy, confidence and the VC dimension. In particular, Theorem 9.4 is an improvement with respect to Theorem 9.3. Note however that the obtained sample size bound still grows with $\frac{1}{\epsilon^2}\log\frac{1}{\epsilon^2}$. This dependence with respect to ϵ makes the bound of practical interest only for relatively large values of the accuracy parameter $\epsilon \in (0, 1)$.

Next, we concentrate on the probability of one-sided constrained failure introduced in (9.5). In particular, we show that when the parameter β is chosen close to zero, manageable sample size bounds are obtained for reasonable values of accuracy and confidence parameters ϵ and δ. This constitutes a computational improvement that significantly reduces the required sample size. Notice that it makes sense to choose β close to zero because in most applications one desires to have a small probability of violation. This is summarized in the next result [9].

Theorem 9.5 *Let $\mathcal{J}$ be a family of measurable binary-valued functions mapping $\mathbb{R}^n$ into $\{0, 1\}$. Suppose $\mathcal{J}$ has finite VC dimension $\mathrm{VC}(\mathcal{J}) \leq d < \infty$. For any $\epsilon \in (0, 1)$, $\delta \in (0, 1)$ and $\beta \in [0, 1)$, if*

$$N \geq \frac{5(\beta+\epsilon)}{\epsilon^2}\left(\log\frac{4}{\delta} + d\log\frac{40(\beta+\epsilon)}{\epsilon^2}\right) \tag{9.6}$$

then $p(\mathcal{J}, N, \epsilon, \beta) < \delta$.

In some applications, a reasonable choice of the level parameter is $\beta = \epsilon$. In this case, taking $\beta = \epsilon$ in Theorem 9.5 one immediately obtains the bound

$$N \geq \frac{10}{\epsilon}\left(\log\frac{4}{\delta} + d\log\frac{80}{\epsilon}\right)$$

which grows with $\frac{1}{\epsilon}\log\frac{1}{\epsilon}$. Another reasonable choice of the level parameter β is to set it to zero. In this case, we easily obtain the bound

$$N \geq \frac{5}{\epsilon}\left(\log\frac{4}{\delta} + d\log\frac{40}{\epsilon}\right).$$

It is worth remarking that a similar bound for the particular case $\beta = 0$ can be obtained using the notion of "version space" presented in [321]. More generally, we can introduce a parametrization of the form $\beta = \epsilon^\ell$, where ℓ is any nonnegative scalar, which relates β and ϵ. With this parametrization the sample size bound grows as $\frac{\epsilon^{\ell-1}+1}{\epsilon}\log\frac{\epsilon^{\ell-1}+1}{\epsilon}$.

9.4 Bounding the VC Dimension

From the previous results, it appears that it is crucial to assess whether a given family $\mathcal{J}$ has finite VC dimension and, in this case, to determine an upper bound d on it. We next report without proof some known results on the computation of the VC dimension for special classes of sets. First, we state a simple result regarding the VC dimension of hyperrectangles, see e.g. [270].

Lemma 9.5 *If $\mathcal{S}$ is the class of all rectangles in $\mathbb{R}^d$, then* $\text{VC}(\mathcal{S}) = 2d$.

The following result on linear functional spaces is stated in [145].

Lemma 9.6 *Let $\mathcal{G}$ be an m-dimensional vector space of real-valued functions $g : \mathbb{R}^d \to \mathbb{R}$. Define the class of sets*

$$\mathcal{S} = \big\{\big\{x \in \mathbb{R}^d : g(x) \geq 0\big\} : g \in \mathcal{G}\big\}.$$

Then, $\text{VC}(\mathcal{S}) \leq m$.

From this lemma we deduce the following corollary.

Corollary 9.1

1. *If $\mathcal{S}$ is the class of all linear half-spaces $\{x : a^T x \geq b\}$, $a \in \mathbb{R}^m$, $b \in \mathbb{R}$, then* $\text{VC}(\mathcal{S}) \leq m + 1$;
2. *If $\mathcal{S}$ is the class of all closed balls $\{x : \|x - a\|_2 \leq r\}$, $a \in \mathbb{R}^m$, $r \in \mathbb{R}_+$, then* $\text{VC}(\mathcal{S}) \leq m + 2$;
3. *If $\mathcal{S}$ is the class of all ellipsoids in $\mathbb{R}^m$ centered at the origin $\mathcal{E}(0, W)$, $W = W^T \succ 0$, then* $\text{VC}(\mathcal{S}) \leq m(m + 1)/2 + 1$.

Using a result in [114], one can show that for linear half-spaces the VC dimension actually equals $m + 1$. Similarly, in [146] it is proved that for the class of closed balls the VC dimension equals $m + 1$.

Next, we report a general result that is useful to establish upper bounds on the VC dimension of families of functions that arise as unions, intersections or other Boolean functions operating on functions belonging to families with known VC dimension.

Lemma 9.7 *Let $\mathcal{J}_1, \ldots, \mathcal{J}_m$ be m families consisting of measurable functions mapping $\mathbb{R}^n$ into $\{0, 1\}$ and let ϕ be a given Boolean function $\phi : \{0, 1\}^m \to \{0, 1\}$. Consider the class $\mathcal{J}_\phi$*

$$\mathcal{J}_\phi \doteq \big\{\phi(J_1, \ldots, J_m) : J_i \in \mathcal{J}_i, i = 1, \ldots, m\big\}.$$

Then, we have

$$\text{VC}(\mathcal{J}_\phi) \leq 2m \log(\mathrm{e}\, m) \max_{i=1,\ldots,m} \big\{\text{VC}(\mathcal{J}_i)\big\}.$$

This lemma is proved for instance in [363]. Results along these lines may also be found in [147, 325, 406, 412]. In particular, the following lemma deals with the case of Boolean closures of polynomial classes.

Lemma 9.8 *Let $\mathcal{J}_1, \ldots, \mathcal{J}_m$ be m families consisting of measurable functions mapping $\mathcal{B}$ into $\{0, 1\}$ and let ϕ be a given Boolean function $\phi : \{0, 1\}^m \to \{0, 1\}$. Suppose further that each class $\mathcal{J}_i$ consists of functions $J_i(\Delta, \theta)$ that are polynomials*

in the parameter $\theta \in \mathbb{R}^\ell$ *of maximum degree* α *in the variables* θ_i, $i = 1, \ldots, \ell$. *Consider the class* $\mathcal{J}_\phi$

$$\mathcal{J}_\phi \doteq \left\{\phi(J_1, \ldots, J_m) : J_i \in \mathcal{J}_i,\ i = 1, \ldots, m\right\}.$$

Then

$$\text{VC}(\mathcal{J}_\phi) \leq 2\ell \log_2(4e\alpha m).$$

This result is proven in a more general form in [227], and it is given in the form stated above in [406]. In Chap. 13 we provide examples showing the explicit computation of the VC dimension for control problems.

9.5 Pollard Theory

The Vapnik–Chervonenkis theory has been extended from binary-valued functions to families of continuous-valued functions bounded in the interval $[0, 1]$. In this case, the analogous concept of the VC dimension is the Pollard (or pseudo) dimension P of the class of functions. To define the P dimension, consider a family of functions $\mathcal{J}$ mapping $\mathbb{R}^n$ into the closed interval $[0, 1]$. Then, a point set $\Delta^{(1\ldots N)} = \{\Delta^{(1)}, \ldots, \Delta^{(N)}\}$ is said to be P-shattered by $\mathcal{J}$ if there exists a vector

$$v = [v_1 \ldots v_N]^T, \quad v_i \in [0, 1],\ i = 1, \ldots, N$$

such that for every binary vector

$$b = [b_1 \ldots b_N]^T, \quad b_i \in \{0, 1\},\ i = 1, \ldots, N$$

there exists a function $J \in \mathcal{J}$ such that

$$\begin{cases} J(\Delta^{(i)}) \geq v_i, & \text{if } b_i = 1; \\ J(\Delta^{(i)}) < v_i, & \text{if } b_i = 0 \end{cases}$$

for all $i = 1, \ldots, N$. Then, the P dimension of $\mathcal{J}$ is defined as follows.

Definition 9.4 (P dimension) Let $\mathcal{J}$ be a family of measurable functions mapping $\mathbb{R}^n$ into $[0, 1]$. The P dimension of $\mathcal{J}$, denoted as P-DIM($\mathcal{J}$), is the largest integer N such that there exists a set of cardinality N that is P-shattered by $\mathcal{J}$.

Note that all the relevant concepts defined in the previous sections for binary-valued functions, such as the empirical mean, the probability of two-sided failure $q(\mathcal{J}, N, \epsilon)$, and the UCEM property, can be readily re-defined to the case of $J : \mathbb{R}^n \to [0, 1]$. With this understanding, we now state a result, due to [324], which is the analog of Theorem 9.1 for the case of continuous-valued functions with finite P dimension.

Theorem 9.6 (Pollard) *Let $\mathcal{J}$ be a family of measurable functions mapping $\mathbb{R}^n$ into* $[0, 1]$. *Suppose $\mathcal{J}$ has finite P dimension* $\text{P-DIM}(\mathcal{J}) \leq d < \infty$. *Then, $\mathcal{J}$ has the UCEM property. In particular, for any $\epsilon > 0$*

$$q(\mathcal{J}, N, \epsilon) \leq 8\left(\frac{16\mathrm{e}}{\epsilon}\log\left(\frac{16\mathrm{e}}{\epsilon}\right)\right)^{d} \mathrm{e}^{-N\epsilon^2/32}.$$

Remark 9.1 (Relation between VC and P dimensions) If $\mathcal{J}$ is a family of measurable functions mapping $\mathbb{R}^n$ into $[0, 1]$, and every function in $\mathcal{J}$ is actually binary-valued, then it can be easily verified that $\text{VC}(\mathcal{J}) = \text{P-DIM}(\mathcal{J})$. More generally, the VC and P dimensions are related as follows: given a family $\mathcal{J}$ of measurable functions mapping $\mathbb{R}^n$ into $[0, 1]$, define the associated family of binary functions $\bar{\mathcal{J}}$ whose elements are the functions $\bar{J}(\Delta, c) = \mathbb{I}_{J(\Delta)\geq c}(\Delta)$, for $c \in [0, 1]$, $J \in \mathcal{J}$. Then

$$\text{VC}(\bar{\mathcal{J}}) = \text{P-DIM}(\mathcal{J}).$$

This relationship is explicitly proved in [272], and it is also reported in [406].

Similarly to Theorem 9.3, the Pollard bound can be inverted to derive explicit sample complexity bounds. In particular, the following result is given in [406].

Theorem 9.7 *Let $\mathcal{J}$ be a family of measurable functions mapping $\mathbb{R}^n$ into* $[0, 1]$. *Suppose $\mathcal{J}$ has finite P dimension* $\text{P-DIM}(\mathcal{J}) \leq d < \infty$. *For any $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$, if*

$$N \geq \frac{32}{\epsilon^2}\left[\log\frac{8}{\delta} + d\left(\log\frac{16\mathrm{e}}{\epsilon} + \log\log\frac{16\mathrm{e}}{\epsilon}\right)\right],$$

then $q(\mathcal{J}, N, \epsilon) \leq \delta$.

Finally, we state a result on the pseudo dimension of the composition of functions $(h \circ J)(\Delta) \doteq h(J(\Delta))$, due to [196].

Lemma 9.9 *Let $\mathcal{J}$ be a family of measurable functions mapping $\mathbb{R}^n$ into* $[0, 1]$, *and let h be a fixed nondecreasing measurable function $h : \mathbb{R} \to \mathbb{R}$. Then, we have*

$$\text{P-DIM}\big(\{(h \circ J) : J \in \mathcal{J}\}\big) \leq \text{P-DIM}(\mathcal{J}).$$

Chapter 10
Randomized Algorithms in Systems and Control

In Chap. 6 we introduced the notion of *probabilistic performance* of a control system, as a counterpart to classical worst-case performance. It should be observed that, except for some special cases which include the examples shown in that chapter, this probabilistic shift does not imply a simplification of the analysis or design problem. Indeed, even establishing if a given system satisfies a probabilistic constraint may be computationally very hard, since it requires the exact evaluation of a multidimensional probability integral. At this point *randomization* comes into play: the probabilistic performance may be *estimated* by randomly sampling the uncertainty, and tail inequalities are used to bound the estimation error. Since the estimated probability is itself a random quantity, this method always entails a certain *risk of failure*, i.e. there exists a nonzero probability of making an erroneous estimation.

In this chapter we first introduce the formal definitions of Monte Carlo and Las Vegas randomized algorithms. Then, we overview algorithms for analysis of uncertain systems, which are also based on the Monte Carlo methods studied in Chap. 7. For control design, the subsequent Chaps. 11, 12 and 13 discuss in detail feasibility and optimization of various convex and nonconvex control problems which are introduced here. Crucial steps for implementing these algorithms are the determination of an appropriate sample size N, discussed in Chaps. 8 and 9, and the construction of efficient algorithms for random sampling the uncertainty $\boldsymbol{\Delta}$ within the structured set $\mathcal{B}_{\mathbb{D}}$, see Chaps. 14, 16 and 18.

10.1 Preliminaries

As we shall see in the following, a randomized algorithm for probabilistic analysis is an algorithm that, based on random extractions of uncertainty samples, returns an estimated probability of satisfaction of the performance specifications. The estimate provided by the randomized algorithm should be within an a priori specified *accuracy* $\epsilon \in (0, 1)$ from the true value, with high *confidence* $1 - \delta$, $\delta \in (0, 1)$. That is, the algorithm may indeed fail to return an approximately correct estimate, but with probability at most δ.

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_10,

A similar situation also arises in the more difficult case of *synthesis*. The complication stems from the fact that the probability to be estimated is no longer fixed, but it is instead a function of some unknown controller or, more generally, design parameters θ. In this case, a randomized algorithm should return a design, represented by a vector $\theta \in \mathbb{R}^{n_\theta}$, which guarantees the desired performance with an a priori specified accuracy $\epsilon \in (0, 1)$. As in the analysis case, this algorithm may fail with probability at most δ.

We highlight some points that are specific to the randomized approach that we intend to pursue. First, we aim to establish probabilistic statements with emphasis on finite sample bounds, as opposed to asymptotic results arising in other branches of probability theory. In particular, as discussed in Chap. 8, the sample complexity of a randomized algorithm is the minimum number of uncertainty samples N (sample size) that need to be drawn in order to achieve the desired levels of accuracy and confidence.

A conceptual distinction with respect to the common use of randomized algorithms in other fields, such as optimization, is that the randomization process is (whenever possible) applied only to the actual uncertainties present in the system, and not to other deterministic decision parameters. Therefore, randomness is not artificially introduced into the problem by the solution technique. Instead, the "natural" randomness due to the presence of stochastic uncertainty in the plant is exploited in the probabilistic solution. Furthermore, we point out that, in principle, design under probabilistic constraints can be viewed as a *chance constrained* optimization problem, see for instance [330, 396]. However, chance constrained problems are generally hard to solve exactly, and therefore randomization provides a viable approach for their solution. Finally, we mention the "mixed deterministic/randomized" approach studied in [118, 171].

10.2 Randomized Algorithms: Definitions

In agreement with classical notions in computer science [288, 290], a *randomized algorithm* (RA) is formally defined as an algorithm that *makes random choices* during its execution to produce a result. This implies that, even for the same input, the algorithm might produce different results at different runs and moreover the results may be incorrect. Therefore, statements regarding properties of the algorithms are probabilistic and these algorithms are sometimes called probably approximately correct (PAC).

Next, we formally introduce the so-called Monte Carlo randomized algorithms. We note that most of the probabilistic results studied in systems and control are based on this type of algorithms.

Definition 10.1 (Monte Carlo randomized algorithm) A Monte Carlo randomized algorithm (MCRA) is a randomized algorithm that may produce an incorrect result, but the probability of such an incorrect result is bounded.

In general, for an MCRA, the results as well as the running times would be different from one run to another since the algorithm is based on random sampling which is a random variable. As a consequence, the computational complexity of such an algorithm is usually measured in terms of its expected running times. An MCRA is said to be efficient if the expected running time is of polynomial order in the problem size (at the end of this chapter we discuss more precisely the meaning of efficient RA).

One simple way to reduce the probability of erroneous results is to run the algorithm repeatedly with independent randomized samples at each time. Therefore, it is often possible to make the error probability arbitrarily small at the expense of increasing the running time. In [384], the two classes of one-sided and two-sided Monte Carlo randomized algorithms are studied, and specific examples in control are given.

We now introduce another class of randomized algorithms known as Las Vegas randomized algorithms.

Definition 10.2 (Las Vegas randomized algorithm) A Las Vegas randomized algorithm (LVRA) is a randomized algorithm which always gives the correct answer. The only difference from one run to another is the running time.

Because of randomization, the running time of a LVRA is random (similarly to MCRA) and may be different in each execution. Hence, it is of interest to study the expected running time of the algorithm. It is noted that the expectation is with respect to the random samples generated during the execution of the algorithm and not to the input of the algorithm (i.e. regarding the inputs to the problem, no assumption on their distribution is made). Furthermore, if the expected running time is of polynomial order in the problem size, the LVRA is said to be efficient.

Remark 10.1 (Las Vegas randomized algorithm) In Chap. 7 we briefly discussed the history of Monte Carlo methods. Regarding Las Vegas, this notion was apparently introduced in 1979 in [33] by the computer scientist Babai. The reason for the name is not clear, but its relation to gambling is obvious. However, such algorithms already existed before this name appeared. An example is the Randomized Quick Sort (RQS) algorithm, which is a well-known Las Vegas type algorithm for sorting real numbers, see [204, 237]. RQS is implemented in a C library of the Unix operating system, see [53]. Extensions to sorting matrices instead of numbers are given in [214] for the special case of Lyapunov equations. More detailed discussions about LVRA for systems and control are presented in [384].

10.3 Randomized Algorithms for Probabilistic Analysis

We consider the classical $M-\Delta$ model, which is frequently used in modern control theory, see Chap. 3. In Fig. 10.1, $M(s)$ represents the transfer matrix of the known

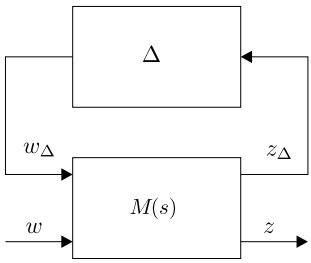

Fig. 10.1 Interconnection for analysis

part of the system, which consists of the extended plant and the controller; w includes noise, disturbances and reference signals, z represents controlled signals and tracking errors. According to Assumption 6.1, the uncertainty $\boldsymbol{\Delta}$ is a random matrix with support $\mathcal{B}_{\mathbb{D}}$. Associated to the M–Δ configuration, in Chap. 6, we defined a *performance function for analysis* (which is assumed to be measurable)

$$J(\Delta) : \mathbb{D} \to \mathbb{R}$$

where $\mathbb{D}$ is the uncertainty structured set defined in (3.27), and an associated performance level γ. The function $J(\Delta)$ takes into account various performance requirements.

Using the M–Δ configuration with random uncertainty, the two basic types of randomized algorithms for performance analysis presented next address Problems 6.4 and 6.5 regarding "Probabilistic performance verification" and "Probabilistic worst-case performance" introduced in Chap. 6. First, we specify the characteristics that an RA for probabilistic performance verification should comply with.

Definition 10.3 (RA for probabilistic performance verification) Let $\boldsymbol{\Delta}$ be random with density $f_{\boldsymbol{\Delta}}(\Delta)$ having support $\mathcal{B}_{\mathbb{D}}$, and let $\epsilon \in (0, 1)$, $\delta \in (0, 1)$ be assigned probability levels. Given a performance function $J(\Delta) : \mathbb{D} \to \mathbb{R}$ and associated level γ, the RA should return with probability at least $1 - \delta$ an estimate $\widehat{\mathbf{p}}_N(\gamma)$ of the probability of performance

$$p(\gamma) = \Pr\left\{ J(\boldsymbol{\Delta}) \leq \gamma \right\}$$

that is within ϵ from $p(\gamma)$. The estimate $\widehat{\mathbf{p}}_N(\gamma)$ should be constructed based on a finite number N of random samples of $\boldsymbol{\Delta} \in \mathcal{B}_{\mathbb{D}}$.

The probability $1 - p(\gamma)$ is sometimes called *probability of violation*. This concept is central in the design case, see Sect. 10.4. Note that the concept of *approximate constraint satisfaction* has been introduced in [45].

Notice that a simple RA for performance verification is directly constructed by means of the Monte Carlo method presented in Chap. 7. This is summarized in the following meta-algorithm.

Algorithm 10.1 (Probabilistic performance verification) *Given* $\epsilon, \delta \in (0,1)$ *and* γ, *this RA returns with probability at least* $1-\delta$ *an estimate* $\widehat{\mathbf{p}}_N(\gamma)$ *such that*

$$\left|p(\gamma) - \widehat{\mathbf{p}}_N(\gamma)\right| < \epsilon.$$

1. Choose integer N satisfying (10.1)
2. Draw N samples $\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(N)}$ according to $f_{\boldsymbol{\Delta}}$;
3. Return the empirical probability

$$\widehat{\mathbf{p}}_N(\gamma) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_{\mathcal{B}_G}\left(\boldsymbol{\Delta}^{(i)}\right)$$

where $\mathbb{I}_{\mathcal{B}_G}(\cdot)$ is the indicator function of the set $\mathcal{B}_G = \{\Delta : J(\Delta) \leq \gamma\}$.

We now comment on step 1 of this algorithm. In Chap. 7 we stated the laws of large numbers for empirical probability, which guarantee the asymptotic convergence $\widehat{\mathbf{p}}_N(\gamma) \to p(\gamma)$ for $N \to \infty$. However, to use Algorithm 10.1 we need estimates based on a *finite* number of samples. This topic has been fully addressed in Chap. 8, in which lower bounds for N are derived. In particular, if N in Algorithm 10.1 is chosen according to the well-known Chernoff bound (8.14), that is

$$N \geq \frac{1}{2\epsilon^2}\log\frac{2}{\delta} \tag{10.1}$$

then this RA satisfies the requirements of Definition 10.3.

Example 10.1 (RA for probability of Schur stability) We revisit Example 6.10 regarding Schur stability of a discrete-time system affected by parametric uncertainty $q \in \mathcal{B}_q$. In particular, we consider the same fourth-order monic polynomial

$$p(z,q) = q_0 + q_1 z + q_2 z^2 + q_3 z^3 + z^4$$

but we now take a different hyperrectangle given by

$$\mathcal{B}_q = \left\{q \in \mathbb{R}^4 : \|q\|_\infty \leq 0.5\right\}.$$

In this case, conditions (6.11) do not guarantee that $\mathcal{S}_n \subseteq \mathcal{B}_q$, and the formulae (6.9) and (6.10) cannot be used. Therefore, we estimate the probability of stability using Algorithm 10.1. In particular, we select the performance function (6.8) and set $\gamma = 1/2$. Then, we assign probability levels $\epsilon = 0.005$ and $\delta = 0.001$ and, by means of the Chernoff bound, we determine the sample size

$$N \geq 1.52 \times 10^5.$$

With this (or larger) sample size we guarantee that with probability at least $1-\delta = 0.999$

$$\left|p(\gamma) - \widehat{\mathbf{p}}_N(\gamma)\right| \leq 0.005.$$

In particular, we applied the RA, using $N = 155{,}000$ iid samples of the random vector $\mathbf{q} \in \mathcal{B}_q$ uniformly distributed within $\mathcal{B}_q$, obtaining an estimated probability

$$\widehat{p}_N = 0.9906.$$

In words, we can say with 99.9% confidence that at least 98.56% of the uncertain polynomials are Schur stable.

Example 10.2 (RA for probability of $\mathcal{H}_\infty$ performance) In this example, we consider a continuous-time SISO system affected by parametric uncertainty, originally presented in [41]. The transfer function of the system is

$$G(s,q) = \frac{0.5q_1q_2s + 10^{-5}q_1}{(10^{-5} + 0.005q_2)s^2 + (0.00102 + 0.5q_2)s + (2\times 10^{-5} + 0.5q_1^2)}$$

where the uncertainty q is bounded in the hyperrectangle

$$\mathcal{B}_q = \left\{q \in \mathbb{R}^2 : 0.2 \le q_1 \le 0.6,\ 10^{-5} \le q_2 \le 3\times 10^{-5}\right\}.$$

Assume further that $\mathbf{q}$ is random, with uniform distribution over the set $\mathcal{B}_q$. We are interested in evaluating the probability of attaining an $\mathcal{H}_\infty$ performance level no larger than $\gamma = 0.003$. We apply a randomized approach, setting the performance function to $J(\mathbf{q}) = \|G(s,\mathbf{q})\|_\infty$. Then, we assign probability levels $\epsilon = 0.01$ and $\delta = 0.001$ and apply Algorithm 10.1. By means of the Chernoff bound, we determine the sample size

$$N \ge 3.801\times 10^4$$

which guarantees that with probability at least $1 - \delta = 0.999$

$$\left|p(\gamma) - \widehat{\mathbf{p}}_N(\gamma)\right| \le 0.01.$$

To apply the RA, we generated $N = 40{,}000$ iid samples of $\mathbf{q} \in \mathcal{B}_q$, uniformly distributed within $\mathcal{B}_q$ and obtained the empirical mean

$$\widehat{p}_N = 0.3482.$$

In practice, we may conclude with 99.9% confidence that the $\mathcal{H}_\infty$ performance level is below $\gamma = 0.003$ for at least 34.8% of the uncertain plants.

Similarly to Algorithm 10.1, an RA for probabilistic worst-case performance should determine a performance level $\widehat{\boldsymbol{\gamma}}_N$ which is guaranteed for most of the uncertainty instances. This is summarized in the next definition.

Definition 10.4 (RA for probabilistic worst-case performance) Let $\boldsymbol{\Delta}$ be random with density $f_{\boldsymbol{\Delta}}(\Delta)$ having support $\mathcal{B}_{\mathbb{D}}$, and let $\epsilon \in (0,1)$, $\delta \in (0,1)$ be assigned probability levels. Given a performance function $J(\Delta) : \mathbb{D} \to \mathbb{R}$, the RA should return with probability at least $1 - \delta$ a performance level $\widehat{\boldsymbol{\gamma}}_N \le \sup_{\Delta\in\mathcal{B}_{\mathbb{D}}} J(\Delta)$ such that

$$\Pr\left\{J(\boldsymbol{\Delta}) \le \widehat{\boldsymbol{\gamma}}_N\right\} \ge 1 - \epsilon.$$

The performance level $\widehat{\boldsymbol{\gamma}}_N$ should be constructed based on a finite number N of random samples of $\boldsymbol{\Delta} \in \mathcal{B}_{\mathbb{D}}$.

An RA for probabilistic worst-case performance is described in the following meta-algorithm.

Algorithm 10.2 (Probabilistic worst-case performance) *Given* $\epsilon, \delta \in (0,1)$, *this RA returns with probability at least* $1-\delta$ *a level* $\widehat{\boldsymbol{\gamma}}_N \leq \sup_{\Delta \in \mathcal{B}_{\mathbb{D}}} J(\Delta)$ *such that*

$$\Pr\left\{ J(\boldsymbol{\Delta}) \leq \widehat{\boldsymbol{\gamma}}_N \right\} \geq 1-\epsilon.$$

1. Choose integer N satisfying (10.2)
2. Draw N samples $\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(N)}$ according to $f_{\boldsymbol{\Delta}}$;
3. Return the empirical maximum

$$\widehat{\boldsymbol{\gamma}}_N = \max_{i=1,\ldots,N} J\left(\boldsymbol{\Delta}^{(i)}\right).$$

In Chap. 8, a finite sample size bound is computed for this RA. In particular, Theorem 8.1, derived in [382], shows that if N in Algorithm 10.2 is chosen according to

$$N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}} \tag{10.2}$$

then this RA satisfies the requirements of Definition 10.4.

Example 10.3 (Randomized algorithm for worst-case $\mathcal{H}_\infty$ performance) We revisit Example 10.2 and study $\mathcal{H}_\infty$ performance of $G(s,q)$. In particular, we are interested in evaluating a performance level $\widehat{\boldsymbol{\gamma}}_N$ which is guaranteed with high probability $1-\epsilon = 0.9999$. To this aim, we apply Algorithm 10.2. Setting a confidence level $\delta = 0.0001$, we determine the sample size by means of the bound given in Theorem 8.1, obtaining

$$N \geq 9.210 \times 10^4.$$

To apply the RA, we hence generated $N = 100{,}000$ iid samples uniformly distributed within $\mathcal{B}_q$ and obtained the empirical maximum

$$\widehat{\gamma}_N = 0.0087.$$

Then, with confidence greater than $1-\delta = 0.9999$, we may conclude that the performance of the uncertain system is below $\widehat{\gamma}_N$ for at least 99.99 % of the uncertain plants.

10.4 Randomized Algorithms for Probabilistic Design

As discussed in Chap. 4, for a fixed plant $G(s)$, the synthesis problem is to design a controller $K(s,\theta)$ such that the interconnection in Fig. 10.2 achieves stability and performance in the presence of uncertainty, where $\theta \in \mathbb{R}^{n_\theta}$ denotes the vector of design parameters (e.g. the free parameters of the controller). The signals w and z have the same meaning of Fig. 10.1, and u, y represent inputs and outputs, respectively. In the probabilistic version of the synthesis problem the uncertainty $\boldsymbol{\Delta}$ is a random matrix with structure $\mathbb{D}$ and support $\mathcal{B}_{\mathbb{D}}$.

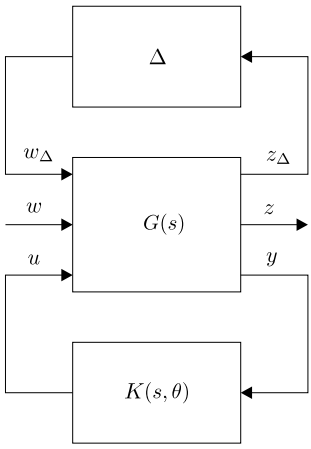

Fig. 10.2 Interconnection for control design

In this section we study RAs for probabilistic controller synthesis. These techniques are based on the interplay of random sampling in the uncertainty space, and deterministic optimization in the design parameter space. Formally, we define a performance function that takes into account the design and performance constraints related to the uncertainty system. These constraints are rewritten in the form of the inequality

$$J(\Delta, \theta) \leq \gamma \tag{10.3}$$

where $J(\Delta, \theta) : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$ is a scalar-valued function denoted as *performance function for design*. Therefore, the performance function for analysis $J(\Delta)$ is a special case of $J(\Delta, \theta)$ for fixed controller parameters θ. We now define the probability of violation and the reliability of the design.

Definition 10.5 (Probability of violation and reliability) Let $\epsilon \in (0, 1)$ be a given accuracy and $\gamma > 0$ a performance level. Given $\theta \in \mathbb{R}^{n_\theta}$, we define the *probability of violation* for the design θ as

$$V(\theta) \doteq \Pr\{J(\boldsymbol{\Delta}, \theta) > \gamma\}. \tag{10.4}$$

Then, the *reliability* of the design θ is defined as

$$R(\theta) \doteq 1 - V(\theta). \tag{10.5}$$

From the previous definition, it follows immediately that θ is a *probabilistic reliable design* if $V(\theta) \leq \epsilon$ or, equivalently, $R(\theta) \geq 1 - \epsilon$ for given accuracy $\epsilon \in (0, 1)$. Most of the results available in the literature for finding a probabilistic reliable design have been derived under the assumption that the function $J(\Delta, \theta)$ is *convex* in θ for all $\Delta \in \mathcal{B}_{\mathbb{D}}$. This assumption is now formally stated and is used in Chaps. 11 and 12. Chapter 13 discusses instead the nonconvex case, which is handled using the statistical learning techniques introduced in Chap. 9.

Assumption 10.1 (Convexity) *The performance function $J(\Delta, \theta)$ is convex in θ for any fixed value of $\Delta \in \mathcal{B}_{\mathbb{D}}$.*

Notice that this assumption requires convexity with respect to the design variable θ, while the only assumption which is needed on the dependence of $J(\Delta, \theta)$ with respect to Δ is measurability.

Remark 10.2 (Performance functions for LMIs) A standard example of convex function J arises when considering performance requirements expressed as uncertain linear matrix inequality (LMI) conditions, as discussed in Sect. 4.3. We recall that a robust LMI feasibility problem is expressed in the form

$$\text{Find } \theta \text{ such that } F(\Delta, \theta) \preceq 0 \text{ for all } \Delta \in \mathcal{B}_{\mathbb{D}}.$$

This problem is rewritten in the scalar-function framework by setting, for instance,

$$J(\Delta, \theta) \doteq \lambda_{\max}\big(F(\Delta, \theta)\big), \quad \gamma \equiv 0, \tag{10.6}$$

where $\lambda_{\max}$ denotes the largest eigenvalue. The following alternative choice in the LMI case was proposed in [86, 328]

$$J(\Delta, \theta) = \big\| \big[F(\Delta, \theta)\big]^+ \big\|_2,$$

where $\|\cdot\|_2$ is the Frobenius norm and $[A]^+$ denotes the projection on the cone $\mathbb{S}^n_+$ of positive semidefinite matrices, i.e.

$$[A]^+ \doteq \arg\min_{X \in \mathbb{S}^n_+} \|A - X\|_2. \tag{10.7}$$

This projection can be computed explicitly, for example, by means of the real Schur decomposition, see [356]. We write

$$[A]^+ = \begin{cases} 0 & \text{if } A \preceq 0; \\ U_p \Lambda_p U_p^T & \text{otherwise,} \end{cases}$$

where the matrices U_p and Λ_p are taken from the real Schur decomposition of A. That is,

$$A = [\, U_p \quad U_n \,] \begin{bmatrix} \Lambda_p & 0 \\ 0 & \Lambda_n \end{bmatrix} \begin{bmatrix} U_n^T \\ U_p^T \end{bmatrix}$$

where Λ_p and Λ_n are diagonal matrices containing, respectively, the nonnegative and the strictly negative eigenvalues of A, and $[U_p \; U_n]$ is an orthogonal matrix.

Remark 10.3 (Multiobjective design) We consider scalar-valued performance functions without loss of generality since a multiobjective design problem subject to multiple constraints

$$J_1(\Delta, \theta) \le \gamma_1, \quad \ldots, \quad J_\ell(\Delta, \theta) \le \gamma_\ell$$

can be easily reduced to a single scalar-valued function by setting

$$J(\Delta, \theta) = \max_{i=1,\ldots,\ell} J_i(\Delta, \theta).$$

Notice that if the functions $J_i(\Delta, \theta)$ are convex in θ, then also the point-wise maximum $J(\Delta, \theta)$ is convex in θ.

We now define the two design problems that we aim to solve using randomized algorithms. The first one is a feasibility problem with constraints expressed in probability.

Problem 10.1 (Find a reliable design) *Let Assumption 10.1 be satisfied. Given a performance function* $J(\Delta, \theta) : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$, *a density* $f_{\boldsymbol{\Delta}}(\boldsymbol{\Delta})$ *with support* $\mathcal{B}_{\mathbb{D}}$ *and accuracy* $\epsilon \in (0, 1)$, *compute* $\theta \in \mathbb{R}^{n_\theta}$ *such that*

$$V(\theta) = \Pr\left\{ J(\boldsymbol{\Delta}, \theta) > \gamma \right\} \leq \epsilon. \tag{10.8}$$

The second problem relates to the optimization of a linear function of the design parameter θ under a probability constraint.

Problem 10.2 (Optimize a reliable design) *Let Assumption 10.1 be satisfied. Given a performance function* $J(\Delta, \theta) : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$, *a density* $f_{\boldsymbol{\Delta}}(\boldsymbol{\Delta})$ *with support* $\mathcal{B}_{\mathbb{D}}$ *and accuracy* $\epsilon \in (0, 1)$, *solve the optimization problem*

$$\begin{aligned} &\min_{\theta} \; c^T \theta \\ &\text{subject to } V(\theta) = \Pr\left\{ J(\boldsymbol{\Delta}, \theta) > \gamma \right\} \leq \epsilon. \end{aligned} \tag{10.9}$$

We remark that Problems 10.1 and 10.2 are generally nonconvex (even when the convexity Assumption 10.1 is satisfied) and therefore are numerically hard to solve. The randomized algorithms we discuss in the subsequent chapters provide a numerically viable way to compute probabilistic approximate solutions to these problems. In particular, under the convexity assumption, in Chaps. 11 and 12 we present RAs for probabilistic feasibility and optimization, respectively. In Chap. 13 we study RAs for probabilistic optimization without the convexity Assumption 10.1 and for general nonlinear measurable objective functions

$$c(\theta) : \mathbb{R}^{n_\theta} \to (-\infty, \infty).$$

In this case, the obtained results are largely based on the statistical learning techniques analyzed in Chap. 9. To this end, we introduce a *binary performance function*

$$J(\Delta, \theta) : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \{0, 1\}$$

which measures the performance of the controlled plant, and we set $\gamma = 1/2$. That is $J(\Delta, \theta) = 0$ means that the performance is verified, and $J(\Delta, \theta) = 1$ that the performance is violated. The violation function is defined as follows

$$V(\theta) \doteq \Pr\left\{ J(\boldsymbol{\Delta}, \theta) = 1 \right\}. \tag{10.10}$$

Then, we define the nonconvex counterpart of Problem 10.2, where the nonconvexity refers to the dependence of J with respect to θ.

Problem 10.3 (Optimize a nonconvex reliable design) *Given a binary performance function* $J(\Delta, \theta) : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \{0, 1\}$, *a density* $f_{\boldsymbol{\Delta}}(\boldsymbol{\Delta})$ *with support* $\mathcal{B}_{\mathbb{D}}$ *and accuracy* $\epsilon \in (0, 1)$, *compute a* local *minimum of the optimization problem*

$$\begin{aligned} &\min_{\theta} \; c(\theta) \\ &\text{subject to } V(\theta) = \Pr\left\{ J(\boldsymbol{\Delta}, \theta) = 1 \right\} \leq \epsilon. \end{aligned} \tag{10.11}$$

A probabilistic synthesis philosophy follows from the previous discussions and aims at designing controllers that satisfy the performance specification for "most" values of the uncertainties, i.e. having small probability of violation. To address Problems 10.1, 10.2 and 10.3, we specify the features that an RA for reliable design should satisfy.

Definition 10.6 (RA for reliable design) Let $\boldsymbol{\Delta}$ be random with density $f_{\boldsymbol{\Delta}}(\boldsymbol{\Delta})$ having support $\mathcal{B}_{\mathbb{D}}$, and let $\epsilon \in (0, 1), \delta \in (0, 1)$ be assigned probability levels. Given a performance function $J(\Delta, \theta) : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$ and associated level γ, the RA should return with probability at least $1 - \delta$ a design vector $\widehat{\boldsymbol{\theta}}_N \in \mathbb{R}^{n_\theta}$ such that

$$V(\widehat{\boldsymbol{\theta}}_N) \leq \epsilon. \tag{10.12}$$

The controller parameter $\widehat{\boldsymbol{\theta}}_N$ should be constructed based on a finite number N of random samples of $\boldsymbol{\Delta} \in \mathcal{B}_{\mathbb{D}}$.

10.5 Computational Complexity

The computational complexity of an RA is due to three main sources: the computational cost of generating random samples of $\boldsymbol{\Delta}$ according to the density $f_{\boldsymbol{\Delta}}(\Delta)$ having support $\mathcal{B}_{\mathbb{D}}$, the computational cost of evaluating the performance $J(\boldsymbol{\Delta}^{(i)})$ for $\boldsymbol{\Delta}^{(i)}$, and the minimum number N of samples required to attain the desired probabilistic levels. A formal definition of an *efficient* randomized algorithm is presented next.

Definition 10.7 (Efficient RA) An RA is said to be *efficient* if:

1. Random sample generation of $\boldsymbol{\Delta}^{(i)}$ can be performed in polynomial time;
2. Performance function can be evaluated in polynomial time for $\boldsymbol{\Delta}^{(i)}$;
3. Sample size is polynomial in the problem size and probabilistic levels.

The random sample generation depends on the type and structure of the set in which randomization is performed. This issue is discussed in detail in Chaps. 16 and 18. More precisely, it is shown that in many cases of practical interest in systems and control, namely vectors and block-structured matrices with norm bound, uniform sample generation can be performed efficiently in polynomial time.

Concerning the second issue above, we remark that, in the majority of cases arising in control problems, the performance function $J(\boldsymbol{\Delta}^{(i)})$ can be efficiently

evaluated for $\boldsymbol{\Delta}^{(i)}$. For example, stability tests, as well as other performance tests based on the solution of linear matrix inequalities, can be performed in polynomial time.

The third source of complexity is indeed the most critical one for a randomized algorithm, and it is discussed in detail in Chaps. 8 and 9. In particular, for analysis, explicit bounds on the sample complexity are available. These bounds depend polynomially on the probabilistic levels, and are actually *independent* of the problem size (dimension and structure of the uncertainty, and type of performance function). For synthesis RAs, similar bounds are shown which are polynomial in the probabilistic levels and in the size and structure of the problem, see previous discussions and Chaps. 11, 12 and 13.

Chapter 11
Sequential Methods for Probabilistic Design

In this chapter we study sequential algorithms for control design of uncertain systems with probabilistic techniques. We introduce a unifying theoretical framework that encompasses most of the sequential algorithms for feasibility that appeared in the literature for the solution of Problem 10.1. In particular, under convexity Assumption 10.1, we develop stochastic approximation algorithms that return a reliable design according to the Definition 10.6. First, we state the definition of ν-feasibility.

Definition 11.1 (ν-feasibility) For given $\nu > 0$ and $\gamma > 0$, we say that Problem 10.1 is ν-feasible if the solution set

$$\mathcal{S} = \left\{\theta \in \mathbb{R}^{n_\theta} : J(\Delta, \theta) \leq \gamma \text{ for all } \Delta \in \mathcal{B}_{\mathbb{D}}\right\} \tag{11.1}$$

contains a full-dimensional ball $\mathcal{B}(\nu)$ of radius ν, called the ν-feasibility ball.

The algorithms presented in the literature for finding a probabilistic feasible solution follow a general iterative scheme, which consists of successive randomization steps to handle uncertainty and optimization steps to update the design parameters. In particular, a careful analysis shows that these algorithms share two fundamental ingredients:

1. A *probabilistic oracle*, which has the purpose of checking if the probability of violation, see Definition 10.5, of the current candidate solution is smaller than a given accuracy $\epsilon \in (0, 1)$;
2. An *update rule*, which exploits the convexity of the problem for constructing a new candidate solution based on the oracle outcome.

With these ingredients in mind, these iterative schemes can be recast in the form of a general sequential algorithm, as depicted in Fig. 11.1. We now make a few comments regarding this algorithm. The probabilistic oracle randomly checks if a candidate solution $\boldsymbol{\theta}^{(k)}$ is reliable for given accuracy ϵ, i.e. $R(\boldsymbol{\theta}^{(k)}) \geq 1 - \epsilon$. If this condition, called `eps-reliable`, holds, then the candidate solution is returned. Otherwise, if the probabilistic oracle returns the unfeasibility condition `unfeas`,

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_11,

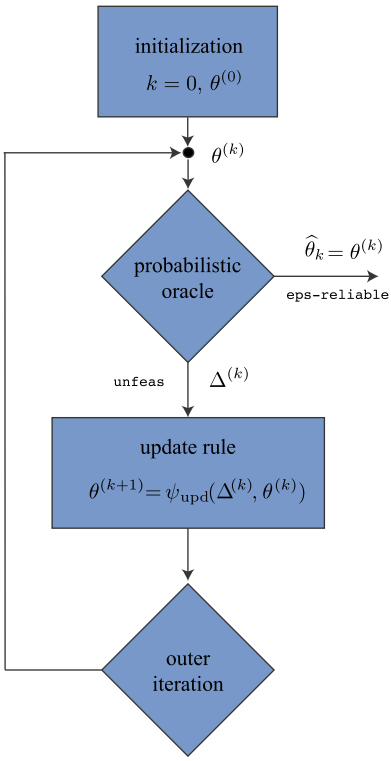

Fig. 11.1 Scheme for sequential probabilistic design

then it also provides a *violation certificate*, which is a realization $\boldsymbol{\Delta}^{(k)}$ of the random uncertainty $\boldsymbol{\Delta}$ such that $J(\boldsymbol{\Delta}^{(k)}, \boldsymbol{\theta}^{(k)}) > \gamma$. We remark that the oracle, due to its probabilistic nature, may possibly declare `eps-reliable` a solution $\boldsymbol{\theta}^{(k)}$ for which $V(\boldsymbol{\theta}^{(k)}) \leq \epsilon$ is not satisfied, i.e. the violation is larger than ϵ. However, this event may only occur with arbitrarily low and prespecified probability. The probabilistic oracle is described in full details in Sect. 11.1 and the update rule $\psi_{\mathrm{upd}}(\boldsymbol{\Delta}^{(k)}, \boldsymbol{\theta}^{(k)})$ is discussed in Sect. 11.3. Notice that the methods proposed in the literature differ only regarding the update rule. This update is applied to the current design parameter $\boldsymbol{\theta}^{(k)}$, when this parameter leads to a performance violation in the probabilistic oracle, to determine a new candidate design $\boldsymbol{\theta}^{(k+1)}$ based on the random sample $\boldsymbol{\Delta}^{(i)}$.

11.1 Probabilistic Oracle

The probabilistic oracle constitutes the randomized part of the algorithm and its role is to check feasibility of the current solution, based on random samples of the uncertainty. More precisely, a multisample of cardinality N_k

$$\boldsymbol{\Delta}^{(1\ldots N_k)} = \left\{\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(N_k)}\right\}$$

is drawn according to the density $f_{\boldsymbol{\Delta}}(\boldsymbol{\Delta})$ with support $\mathcal{B}_{\mathbb{D}}$. Then, we say that the candidate design $\boldsymbol{\theta}^{(k)}$ is `eps-reliable` if

$$J\left(\boldsymbol{\Delta}^{(i)}, \boldsymbol{\theta}^{(k)}\right) \leq \gamma, \quad i = 1, 2, \ldots, N_k.$$

This leads to the following randomized scheme for the probabilistic oracle.

Algorithm 11.1 (Probabilistic oracle) *Given* $\epsilon \in (0, 1)$, N_k *and* $\boldsymbol{\theta}^{(k)}$, *this RA returns either* `sol=eps-reliable`, *or* `sol=unfeas` *and violation certificate* $\boldsymbol{\Delta}^{(k)}$.

1. For $i = 1$ to N_k do.
   - ▷ Draw a random sample $\boldsymbol{\Delta}^{(i)}$ according to $f_{\boldsymbol{\Delta}}$;
2. Randomized test.
   - ▷ If $J(\boldsymbol{\Delta}^{(i)}, \boldsymbol{\theta}^{(k)}) > \gamma$ then;
     - set `sol=unfeas` and $\boldsymbol{\Delta}^{(k)} = \boldsymbol{\Delta}^{(i)}$;
     - return `sol`, $\boldsymbol{\Delta}^{(k)}$ and Exit;
   - ▷ End if;
3. End for.

Notice that at step k the feasibility of the candidate solution $\boldsymbol{\theta}^{(k)}$ is verified with respect to a finite number N_k of random samples. If the test is passed, the solution is considered probabilistic feasible, and labeled `eps-reliable`; otherwise, the uncertainty realization $\boldsymbol{\Delta}^{(k)}$ for which the randomized test failed is returned as a violation certificate. The sample size N_k depends on k, and has to be chosen to guarantee the desired probabilistic properties of the solution. A subsequent result states that, with high probability, the solution $\boldsymbol{\theta}^{(k)}$ is indeed a probabilistic feasible solution.

To this end, we define the *probability of misclassification* as the probability that the oracle labels `eps-reliable` a *bad* solution, i.e. a solution for which the violation $V(\boldsymbol{\theta}^{(k)}) > \epsilon$ occurs. Formally, we define the event

$$\text{Misclass} \doteq \left\{\text{oracle labels } \boldsymbol{\theta}^{(k)} \texttt{ eps-reliable} \text{ and } \left\{V\left(\boldsymbol{\theta}^{(k)}\right) > \epsilon\right\}\right\}.$$

Then, we state a theorem regarding the probability of misclassification, see [88].

Theorem 11.1 (Probability of misclassification of the oracle) *Let* $\epsilon \in (0, 1)$ *be a given accuracy and let* $\boldsymbol{\theta}^{(k)}$ *be a query point. Then, the probability of misclassification of the oracle is less than* $(1 - \epsilon)^{N_k}$, *i.e.*

$$\text{PR}_{\boldsymbol{\Delta}^{(1\ldots N_k)}}\{\text{Misclass}\} \leq (1 - \epsilon)^{N_k}.$$

Remark 11.1 (Probability of success) We notice that, by appropriate choice of the number of iterations N_k, we can make the probability of misclassification of the probabilistic oracle as close as desired to zero. In particular, to achieve a desired success probability $1-\delta$, where $\delta \in (0,1)$ is a given confidence, we need a number of iterations in the oracle given by

$$N_k \geq N_{\text{oracle}}(\epsilon, \delta) \doteq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}}. \tag{11.2}$$

11.2 Unified Analysis of Sequential Schemes

In this section we present a unifying view for studying the probabilistic behavior of this scheme, which is formally stated in Algorithm 11.2.

Algorithm 11.2 (Sequential probabilistic design) *Given* $\epsilon, \delta \in (0,1)$ *and* $N_{\text{out}}(\nu)$, *this RA returns either* `sol=eps-reliable` *and a design vector* $\widehat{\boldsymbol{\theta}}_k$, *or* `sol=unfeas`.

1. Initialization.
   - ▷ Choose $\theta^{(0)} \in \mathbb{R}^{n_\theta}$, set $k=0$ and `sol=unfeas`;
2. Outer iteration.
   - ▷ While `sol=unfeas` and $k < N_{\text{out}}(\nu)$ do;
     - – Choose integer N_k according to (11.3);
     - – Obtain `sol` and $\boldsymbol{\Delta}^{(k)}$ by Algorithm 11.1 (Probabilistic oracle);
     - – If `sol=eps-reliable` then Return $\widehat{\boldsymbol{\theta}}_k = \boldsymbol{\theta}^{(k)}$ and Exit;
     - – Else Update $\boldsymbol{\theta}^{(k+1)}$ as in Algorithm 11.3 or 11.4 (Update rule);
     - – End if;
     - – Set $k = k+1$;
   - ▷ End while

To discuss Algorithm 11.2, first notice that, contrary to the probabilistic oracle, the update step is completely *deterministic* and does not involve randomization. To clarify this point, consider again the sequential scheme in Fig. 11.1, and suppose that an *exact oracle* were available, that is a deterministic oracle which would be able to discern exactly feasibility of a candidate solution $\boldsymbol{\theta}^{(k)}$. Such an exact oracle would return `exact-feas` whenever $J(\Delta, \boldsymbol{\theta}^{(k)}) \leq \gamma$ holds for all $\Delta \in \mathcal{B}_{\mathbb{D}}$, and `unfeas` otherwise. In this case, Algorithm 11.2 would be completely deterministic, and its convergence properties could be analyzed in a non-probabilistic setting. Unfortunately, to verify if the inequality $J(\Delta, \boldsymbol{\theta}^{(k)}) \leq \gamma$ is satisfied for all $\Delta \in \mathcal{B}_{\mathbb{D}}$ is in general very hard, and such an oracle can be rarely constructed in practice.

However, its introduction is of conceptual importance, since it allows us to formally unify all the randomized sequential algorithms previously presented in the literature. We now introduce an assumption regarding convergence with an exact oracle.

Assumption 11.1 (Outer convergence of exact oracle) *We assume that the update rule in Algorithm 11.2 is such that the following implication holds true*:

- ***if*** *the problem is ν-feasible (see Definition 11.1) and an exact oracle is available*
- ***then*** *Algorithm 11.2 converges in a finite number of outer iterations and this number is bounded from above by a known function $N_{\text{out}}(\nu)$.*

We notice that $N_{\text{out}}(\nu)$ may also depend on other specific parameters, such as the parameters entering into the update rule.

The probabilistic properties of Algorithm 11.2 are formally derived in the next theorem, which constitutes a slight improvement upon original results first appeared in [310] and then in [170] and [88]. The proof follows the same lines of the proof of Theorem 5.3 in [119].

Theorem 11.2 (Probabilistic properties of Algorithm 11.2) *Let convexity Assumption 10.1 and feasibility Assumption 11.1 be satisfied. Let $\epsilon, \delta \in (0, 1)$ be given probability levels, and assume that at step k of Algorithm 11.2 the probabilistic oracle is invoked with the sample size*

$$N_k \geq N_{\text{oracle}}(\epsilon, \delta) + \frac{\alpha \log k + \log \zeta(\alpha)}{\log \frac{1}{1-\epsilon}}, \quad \alpha > 1 \tag{11.3}$$

where $N_{\text{oracle}}(\epsilon, \delta)$ is given in (11.2) and $\zeta(\cdot)$ is the Riemann zeta function. Then, the following statements hold:

1. *The probability that Algorithm 11.2 terminates at some outer iteration $k < N_{\text{out}}(\nu)$ returning a design vector $\widehat{\boldsymbol{\theta}}_k$ such that*

$$V(\widehat{\boldsymbol{\theta}}_k) = \Pr\left\{ J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_k) > \gamma \right\} > \epsilon$$

   is less than δ;
2. *If Algorithm 11.2 reaches the number of outer iterations $N_{\text{out}}(\nu)$, then the problem is not ν-feasible.*

Remark 11.2 (Number of outer iterations) If Algorithm 11.2 exits at some outer iteration k, then based on Theorem 11.2, we may declare that the solution $\widehat{\boldsymbol{\theta}}_k$ is reliable with accuracy ϵ and confidence δ, and this corresponds to a *successful exit* of the algorithm. On the other hand, if the algorithm exits because no solution has been found in $N_{\text{out}}(\nu)$ iterations, then we state *unsuccessful exit*. In this case, we conclude with certainty that the problem is not ν-feasible. Notice that, in practice, one may not know in advance whether the problem is ν-feasible or not. The described randomized schemes can be used also if the problem is not ν-feasible, or even if no feasible solution exists at all, i.e. the set $\mathcal{S}$ is empty. The point is that *if*

Algorithm 11.2 terminates at some outer iteration $k < N_{\text{out}}(\nu)$, then the returned solution $\widehat{\boldsymbol{\theta}}_k$ is reliable with accuracy ϵ, unless a rare event of probability smaller than δ occurred. The probability δ has, of course, to be set to a suitable low value to make the a priori chance of occurrence of this event negligible in practice. The introduction of the $N_{\text{out}}(\nu)$ limit in the number of iterations has the purpose of guaranteeing finite-time termination of the sequential scheme in Algorithm 11.2, in cases when the oracle cannot find a probabilistic solution. If we do not assume ν-feasibility, we cannot determine $N_{\text{out}}(\nu)$, but we can still use the algorithm letting it run indefinitely, without a priori guarantee of finite termination.

Notice that if we set $\alpha = 2$ in (11.3), then we recover the bound derived in [310]. If we choose, for instance, $\alpha = 1.11$, we obtain the bound

$$N_k \geq N_{\text{oracle}}(\epsilon, \delta) + \frac{1.11 \log k + 2.27}{\log \frac{1}{1-\epsilon}}, \tag{11.4}$$

which improves upon the bound in [310] for $k > 3$. Notice that any N_k which satisfies (11.3) also satisfies the bound (11.2). It is also important to remark that the sample size N_k in (11.3) is independent of the number of uncertain and design parameters, see the discussion in Chap. 10 regarding polynomial time randomized algorithms.

11.3 Update Rules

Various update rules have been proposed in the literature, among which we recall gradient, ellipsoid and cutting plane. The update rules we discuss in this section assume the availability of a subgradient $\partial_k(\theta)$ of the performance function $J(\Delta, \theta)$ at $\Delta^{(k)}$. Notice that, if $J(\Delta, \theta)$ is differentiable at θ, then $\partial_k(\theta)$ is simply the gradient of J, i.e. $\partial_k(\theta) = \nabla_\theta J(\Delta^{(k)}, \theta)$.

Remark 11.3 (Subgradient for LMIs) We consider a set of uncertain LMIs as in Sect. 4.3

$$F(\Delta, \theta) \preceq 0, \quad \Delta \in \mathcal{B}_{\mathbb{D}}$$

where

$$F(\Delta, \theta) = F_0(\Delta) + \sum_{i=1}^{n_\theta} \theta_i F_i(\Delta)$$

and $F_i(\Delta) \in \mathbb{S}^n$, $i = 0, 1, \ldots, n_\theta$, $\Delta \in \mathcal{B}_{\mathbb{D}}$ and $\theta \in \mathbb{R}^{n_\theta}$. Then, according to (10.6) in Remark 10.2, we set $J(\Delta^{(k)}, \theta) = \lambda_{\max}(F(\Delta^{(k)}, \theta))$ and $\gamma \equiv 0$, where $\lambda_{\max}$ denotes the largest eigenvalue of $F(\Delta^{(k)}, \theta)$. A subgradient of $J(\Delta^{(k)}, \theta)$ at $\theta = \theta^{(k)}$ is given by, see [86] for details,

$$\partial_k(\theta^{(k)}) = \begin{bmatrix} \xi_{\max}^T F_1(\Delta^{(k)}) \xi_{\max} & \cdots & \xi_{\max}^T F_{n_\theta}(\Delta^{(k)}) \xi_{\max} \end{bmatrix}^T$$

where $\xi_{\max}$ is a unit norm eigenvector associated to the largest eigenvalue of $F(\Delta^{(k)}, \theta^{(k)})$.

11.3.1 Subgradient Update

The first update rule proposed in the literature, see [86, 328], is also the simplest one, and is based on a (sub)gradient technique. The main distinguishing feature of the method lies in the particular choice of the stepsize $\boldsymbol{\eta}^{(k)}$

$$\boldsymbol{\eta}^{(k)} = \frac{J(\boldsymbol{\Delta}^{(k)}, \boldsymbol{\theta}^{(k)})}{\|\partial_k(\boldsymbol{\theta}^{(k)})\|} + \nu \tag{11.5}$$

where ν is the given feasibility radius. This update rule is summarized in the next algorithm.

Algorithm 11.3 (Update rule—subgradient method) *Given* $\boldsymbol{\theta}^{(k)}$, $\boldsymbol{\Delta}^{(k)}$, *this RA returns* $\boldsymbol{\theta}^{(k+1)}$.

1. Subgradient computation.
   - ▷ Compute the subgradient $\partial_k(\theta)$ of $J(\boldsymbol{\Delta}^{(k)}, \theta)$;
2. Stepsize computation.
   - ▷ Compute the stepsize according to (11.5);
3. Update.
   - ▷ Set $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \boldsymbol{\eta}^{(k)} \frac{\partial_k(\boldsymbol{\theta}^{(k)})}{\|\partial_k(\boldsymbol{\theta}^{(k)})\|}$.

This update rule satisfies Assumption 11.1. In particular, the stepsize (11.5) guarantees finite-time convergence of the deterministic version of the algorithm with an exact oracle in a number of steps bounded by

$$N_{\text{out}}(\nu) = \frac{\Omega^2}{\nu^2},$$

where Ω is an a priori upper bound on the distance of the initial solution $\theta^{(0)}$ from the center of the ν-feasibility ball $\mathcal{B}(\nu)$, see Definition 11.1.

Remark 11.4 (Stochastic approximation and gradient methods) We observe that sequential algorithms based on gradient updates have been used in a deterministic setting for adaptive control and unknown but bounded identification, see e.g. [35, 65]. In particular, these iterative algorithms can be traced back to the Kaczmarz projection method [220] for solving linear equations, which was studied in 1937, and to the methods given in [6, 291] for studying linear inequalities, which were presented in 1954. A finite-time version of the latter method was proposed later, see [65] and references therein. A similar approach for solving convex inequalities can also be found in [165, 326]. Finally, if compared with the method of alternating projections, see [356], we observe that the gradient update rule does not require the computation

of projections. From the numerical point of view, this is a crucial advantage, since this operation is generally difficult to perform.

More generally, stochastic approximation algorithms have been widely studied in the optimization literature, see e.g. [102, 244] for detailed discussions on this topic. However, these algorithms do not have a finite termination criterion, and the subgradient is generally computed numerically, contrary to the closed-form expression stated in Remark 11.3. For convex problems, we also recall the stochastic optimization methods proposed in [297], and more recently discussed in [300].

In the randomized literature, various gradient-based algorithms have been proposed, including [168, 328, 395] which are focused on the design of linear parameter systems (LPV), guaranteed-cost quadratic regulator problems, and stochastic jump systems. Furthermore, the paper [86] studies feasible and unfeasible LMIs and [117] deals with system identification.

11.3.2 Localization Methods

More sophisticated randomized algorithms, that still guarantee probabilistic properties of the ensuing solution, and provide improved convergence rates, have been proposed in the literature. These techniques fall in the class of the so-called *localization methods*. In these methods, the update rule is based on the computation of a center of a suitably defined localization set $\mathcal{L}_k$. The set $\mathcal{L}_k$ is guaranteed to contain at each step the feasible set $\mathcal{S}$ defined in (11.1), that is $\mathcal{S} \subseteq \mathcal{L}_k$, and is constructed based on the violation certificate $\boldsymbol{\Delta}^{(k)}$ returned by the probabilistic oracle. In particular, the uncertainty realization $\boldsymbol{\Delta}^{(k)}$ is used to construct a separating hyperplane $h_k \doteq \{\xi \in \mathbb{R}^{n_\theta} : a_k^T \xi = b_k\}$ having the property that $a_k^T \boldsymbol{\theta}^{(k)} \geq b_k$ and $a_k^T \theta \leq b_k$ for all $\theta \in \mathcal{S}$. Specifically, if $\partial_k(\boldsymbol{\theta}^{(k)})$ is a subgradient of $J(\boldsymbol{\Delta}^{(k)}, \theta)$ at $\theta = \boldsymbol{\theta}^{(k)}$, then a separating hyperplane may be obtained as

$$
\begin{aligned}
a_k &= \partial_k\big(\boldsymbol{\theta}^{(k)}\big); \\
b_k &= \partial_k^T\big(\boldsymbol{\theta}^{(k)}\big)\boldsymbol{\theta}^{(k)} - J\big(\boldsymbol{\Delta}^{(k)}, \boldsymbol{\theta}^{(k)}\big) + \gamma.
\end{aligned}
$$

The separating hyperplane h_k indicates that the half-space $\{\theta : a_k^T \theta > b_k\}$ cannot contain a feasible point and can therefore be eliminated (cut) in subsequent steps of the algorithm. In this case, we know that $\mathcal{S} \subseteq \mathcal{L}_k \cap \mathcal{H}_k$, where

$$
\mathcal{H}_k \doteq \big\{\theta : a_k^T \theta \leq b_k\big\}
$$

and the algorithm constructs an updated localization set $\mathcal{L}_{k+1}$ such that $\mathcal{L}_{k+1} \supseteq \mathcal{L}_k \cap \mathcal{H}_k$. A new query point $\boldsymbol{\theta}^{(k+1)} \in \mathcal{L}_{k+1}$ is then computed, and the process is repeated. This is summarized in the following scheme.

Algorithm 11.4 (Update rule—localization methods) *Given* $\boldsymbol{\theta}^{(k)}$, $\boldsymbol{\Delta}^{(k)}$, *this RA returns* $\boldsymbol{\theta}^{(k+1)}$.

1. Subgradient computation.
   - ▷ Compute the subgradient $\partial_k(\theta)$ of $J(\boldsymbol{\Delta}^{(k)}, \theta)$;
2. Half-space construction.
   - ▷ Compute the half-space $\mathcal{H}_k$ using the subgradient;
3. Update.
   - ▷ Update the localization set $\mathcal{L}_{k+1} \supseteq \mathcal{L}_k \cap \mathcal{H}_k$;
   - ▷ Return $\boldsymbol{\theta}^{(k+1)} = \texttt{center}(\mathcal{L}_k)$.

The convergence of these methods hinges upon the fact that each time a cut is performed, the localization set shrinks by a certain factor. Intuitively, this guarantees that eventually, for sufficiently large k, either we terminate by finding a feasible point, or we declare that the problem is not ν-feasible. Regarding $\texttt{center}(\mathcal{L}_k)$, different methods follow from different choices of the shape and description of the localization sets. In particular, in the probabilistic ellipsoid algorithm the localization set is an ellipsoid and $\texttt{center}(\mathcal{L}_k)$ is the center of the ellipsoid. In the probabilistic cutting plane methods, the localization set is a polytope, and $\texttt{center}(\mathcal{L}_k)$ is a center of this polytope (usually, the analytic center). These two classes of localization methods are discussed in the next sections.

11.3.3 Probabilistic Ellipsoid Algorithm

This is the first randomized localization scheme proposed in the literature, see [224]. The method represents a probabilistic extension of the classical ellipsoid algorithm, see Fig. 11.2, originally studied in [355, 418]. An ellipsoid is described by means of a center $\theta \in \mathbb{R}^n$ and a symmetric positive definite shape matrix $W \in \mathbb{S}^n$,

$$\mathcal{E}(\theta, W) = \left\{x \in \mathbb{R}^n : (x-\theta)^T W^{-1}(x-\theta) \leq 1\right\}.$$

We assume that an initial ellipsoid $\mathcal{E}^{(0)} = \mathcal{E}(\theta^{(0)}, W^{(0)})$ is given, such that $\mathcal{S} \subseteq \mathcal{E}^{(0)}$. The advantage of this method is that the information associated to the localization set at step k is captured by two parameters only: the center of the ellipsoid and the shape matrix. This allows us to write the update rule in a simple closed-form. If $\partial_k(\boldsymbol{\theta}^{(k)})$ is a subgradient of $J(\boldsymbol{\Delta}^{(k)}, \theta)$ at $\theta = \boldsymbol{\theta}^{(k)}$, then the new centers and shape matrices can be obtained as follows

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \frac{1}{n_\theta + 1} \frac{\mathbf{W}^{(k+1)} \partial_k(\boldsymbol{\theta}^{(k)})}{\sqrt{\partial_k^T(\boldsymbol{\theta}^{(k)}) \mathbf{W}^{(k)} \partial_k(\boldsymbol{\theta}^{(k)})}};$$

$$\mathbf{W}^{(k+1)} = \frac{{n_\theta}^2}{{n_\theta}^2 - 1} \left(\mathbf{W}^{(k)} - \frac{2}{n_\theta + 1} \frac{\mathbf{W}^{(k)} \partial_k(\boldsymbol{\theta}^{(k)}) \partial_k^T(\boldsymbol{\theta}^{(k)}) \mathbf{W}^{(k)}}{\partial_k^T(\boldsymbol{\theta}^{(k)}) \mathbf{W}^{(k)} \partial_k^T(\boldsymbol{\theta}^{(k)})} \right).$$

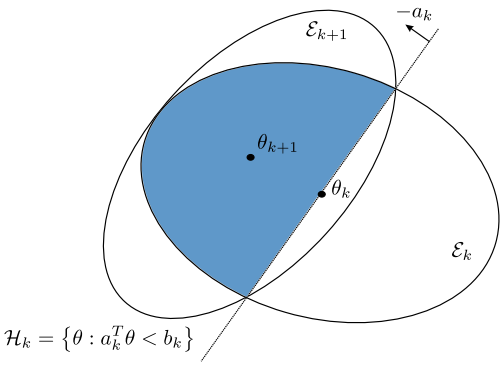

Fig. 11.2 Update step of ellipsoid method

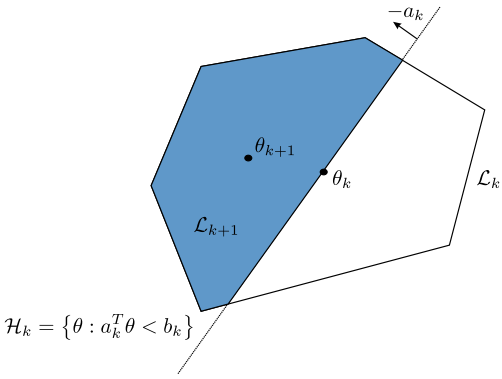

Fig. 11.3 Update steps of cutting-plane method

For this algorithm, the results in [224] imply that Theorem 11.2 holds with

$$N_{\text{out}}(\nu) = 2n_\theta \log \frac{\text{Vol}(\mathcal{E}^{(0)})}{\text{Vol}(\mathcal{B}(\nu))}$$

where $\text{Vol}(\mathcal{E}^{(0)})$ and $\text{Vol}(\mathcal{B}(\nu))$ are the volumes of the initial ellipsoid $\mathcal{E}^{(0)}$ and of the ν-feasibility ball $\mathcal{B}(\nu)$.

11.3.4 Probabilistic Cutting Plane Techniques

Randomized localization schemes based on cutting planes have been proposed in [88]. In probabilistic cutting plane methods, the localization set is described by means of a polytope $\mathcal{P}_k$. In the update phase, a new polytope $\mathcal{P}_k$ is constructed as the intersection of the current localization set $\mathcal{P}_k$ and the cutting plane $\mathcal{H}_k$. Then, a new query point is computed as a center of the polytope, see Fig. 11.3.

Various cutting plane methods may be developed based on different strategies in the outer phase for updating the localization set and constructing the new query point. This is a very delicate phase, since the numerical complexity and convergence

of the method critically rely on these choices. Detailed discussions of the various methods fall outside the scope of this section, so that we only mention some of the most widely known, and we refer the reader to the specific literature, see, e.g. [181, 287] for further details. Currently, the most popular family of cutting plane methods is that based on analytic centers [181, 298]. In this case, the query point is computed as the analytic center of the polytope $\mathcal{P}_k$, or a suitably pruned version of it, see [29]. The analytic center is defined as the unique minimizer of the logarithmic barrier function

$$\phi(\theta) = -\sum_i \log\big(b_i - a_i^T \theta\big),$$

where $a_i \in \mathbb{R}^{n_\theta}$, $b_i \in \mathbb{R}$ are the parameters of the hyperplanes defining the polytope $\mathcal{P}_k$. In the probabilistic version of the analytic center cutting plane method analyzed in [88], an explicit bound is obtained

$$N_{\text{out}}(\nu) = \max\left\{50 n_\theta, 13.87 n_\theta^2, 8 n_\theta^2 \left(\frac{\Omega}{\nu}\right)^{2.1}\right\}$$

where Ω is the radius of a hypercube known to contain $\mathcal{S}$ and ν is the ν-feasibility radius.

Remark 11.5 (Complexity of different randomized schemes) If we define the quantity $h = \Omega/\nu$, being Ω the radius of a ball inscribing the set $\mathcal{S}$, then it can be seen that $N_{\text{out}}(\nu)$ grows as $O(h^2)$ for the gradient rule, and as $O(n_\theta^2 \log(\sqrt{n_\theta} h))$ for the ellipsoid method. For one of the best-known cutting-plane methods detailed in [29], the convergence is instead of the order of $O(n_\theta \log^2(h))$. We also remark that all the update rules satisfy Assumption 11.1.

We now present an example, taken from [120], showing an application of Algorithm 11.2. A similar example studying a gradient update rule is given in [328].

Example 11.1 (LQR design of a lateral motion of an aircraft) We consider a multivariable example given in [21] (see also the original paper [393] for a slightly different model and set of data) which studies the design of a controller for the lateral motion of an aircraft. The state space equation consists of four states and two inputs and is given by

$$\begin{aligned}\dot{x}(t) = &\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & L_p & L_\beta & L_r \\ g/V & 0 & Y_\beta & -1 \\ N_{\dot\beta}(g/V) & N_p & N_\beta + N_{\dot\beta} Y_\beta & N_r - N_{\dot\beta} \end{bmatrix} x(t) \\ &+ \begin{bmatrix} 0 & 0 \\ 0 & L_{\delta_a} \\ Y_{\delta_r} & 0 \\ N_{\delta_r} + N_{\dot\beta} Y_{\delta_r} & N_{\delta_a} \end{bmatrix} u(t) \end{aligned} \tag{11.6}$$

Table 11.1 Uncertain parameters q and nominal values $\bar{q}$

L_p	L_β	L_r	g/V	Y_β	$N_{\dot{\beta}}$	N_p	N_β	N_r	L_{δ_a}	Y_{δ_r}	N_{δ_r}	N_{δ_a}
−2.93	−4.75	0.78	0.086	−0.11	0.1	−0.042	2.601	−0.29	−3.91	0.035	−2.5335	0.31

where x_1 is the bank angle, x_2 its derivative, x_3 is the sideslip angle, x_4 the yaw rate, u_1 the rudder deflection and u_2 the aileron deflection.

We consider the case when the thirteen aircraft parameters entering into the state and input matrices of Eq. (11.6) are uncertain. Hence, we consider the system

$$\dot{x}(t) = A(q)x(t) + B_2(q)u(t),$$

where we introduced the uncertainty vector $q = [q_1 \ \cdots \ q_\ell]^T$ with $\ell = 13$. The vector q is assumed to vary in the hyperrectangle centered at the nominal value $\bar{q}$, and radius ρ i.e.

$$\mathcal{B}_q(\rho) = \left\{ q \in \mathbb{R}^\ell : q_i \in \left[(1-\rho)\bar{q}_i, (1+\rho)\bar{q}_i \right], \ i = 1, \ldots, \ell \right\}. \tag{11.7}$$

In particular, we consider the case when each parameter q_i is perturbed by a relative uncertainty equal to $\rho = 10\,\%$ around its nominal value $\bar{q}_i$, as reported in Table 11.1. For simplicity, we write in the sequel $\mathcal{B}_q = \mathcal{B}_q(0.1)$.

We are interested in designing a state feedback controller $u = Kx$ that stabilizes the system guaranteeing a desired decay rate $\alpha > 0$, which is equivalent to having all closed loop eigenvalues with real part smaller than $-\alpha$. A sufficient condition, see Sect. 4.2.2, for the existence of a stabilizing controller requires finding a symmetric positive definite matrix $P \in \mathbb{R}^{4,4}$ and a matrix $Y \in \mathbb{R}^{2,4}$ such that the quadratic performance criterion is satisfied for all values of $q \in \mathcal{B}_q$,

$$F(q, P, Y) = A(q)P + PA^T(q) + B_2(q)Y + Y^T B_2^T(q) + 2\alpha P \preceq 0. \tag{11.8}$$

Further, if we find common solutions $P \succ 0$ and Y that simultaneously satisfy this linear matrix inequality for all $q \in \mathcal{B}_q$, then a control gain K which guarantees the desired decay rate can be recovered as $K = YP^{-1}$.

Next, we assume that $\mathbf{q}$ is a random vector uniformly distributed with $\mathcal{B}_q$. Then, we look for $P \succ 0$, Y that satisfies the probability constraint

$$\text{PR}\left\{ \mathbf{q} \in \mathcal{B}_q : F(\mathbf{q}, P, Y) \preceq 0 \right\} \geq 1 - \epsilon.$$

If we introduce the performance function

$$J(\mathbf{q}, P, Y) = \lambda_{\max}\big(F(\mathbf{q}, P, Y)\big), \quad \gamma \equiv 0$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of its argument, then we rewrite the probability constraint as

$$\text{PR}\left\{ \mathbf{q} \in \mathcal{B}_q : J(\mathbf{q}, P, Y) \leq 0 \right\} \geq 1 - \epsilon. \tag{11.9}$$

The results of the simulations are now described. First, we set $\alpha = 0.5$ and look for a probabilistic feasible solution $\theta = \{P, Y\}$ to the uncertain LMIs

$$\begin{aligned} P &\succeq \beta I; \\ A(\mathbf{q})P + PA^T(\mathbf{q}) + B_2(\mathbf{q})Y + Y^T B_2^T(\mathbf{q}) + 2\alpha P &\preceq 0 \end{aligned} \tag{11.10}$$

where $\mathbf{q} \in \mathcal{B}_q$ and $\beta = 0.01$. We apply Algorithm 11.2 with ellipsoid update rule and probability levels $\epsilon = 0.01$, $\delta = 10^{-6}$. With this setting, the algorithm is guaranteed to return (with 99.9999 % probability) a solution P, Y such that (11.10) holds with 99 % probability.

The algorithm was run with initial random solution $\theta^{(0)}$ and terminated after $N = 28$ outer iterations returning the solution

$$\widehat{P}_N = \begin{bmatrix} 0.3075 & -0.3164 & -0.0973 & -0.0188 \\ -0.3164 & 0.5822 & -0.0703 & -0.0993 \\ -0.0973 & -0.0703 & 0.2277 & 0.2661 \\ -0.0188 & -0.0993 & 0.2661 & 0.7100 \end{bmatrix}; \tag{11.11}$$

$$\widehat{Y}_N = \begin{bmatrix} -0.0191 & -0.0920 & 0.0803 & 0.4496 \\ 0.2733 & 0.4325 & -0.3821 & -0.2032 \end{bmatrix}. \tag{11.12}$$

This solution is deemed probabilistically feasible by the oracle after checking (11.10) for $N_k = 2,089$ uncertainty samples. Then, the probabilistic controller is constructed as

$$\widehat{K}_N = \begin{bmatrix} -2.9781 & -1.9139 & -3.2831 & 1.5169 \\ 7.3922 & 5.1010 & 4.1401 & -0.9284 \end{bmatrix}.$$

Subsequently, we perform a posteriori analysis of the solution that has been obtained. First, we proceed with worst-case analysis. To this end, we notice that the entries of the state matrices

$$A(q) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & q_1 & q_2 & q_3 \\ q_4 & 0 & q_5 & -1 \\ q_4 q_6 & q_7 & q_8 + q_5 q_6 & q_9 - q_6 \end{bmatrix}, \qquad B_2(q) = \begin{bmatrix} 0 & 0 \\ 0 & q_{10} \\ q_{11} & 0 \\ q_{12} + q_6 q_{11} & q_{13} \end{bmatrix}$$

depend multiaffinely on the uncertainty q. In this case, to detect quadratic performance of an uncertain system affected by multiaffine uncertainty, it suffices to check the simultaneous satisfaction of the uncertain constraint (11.8) for a specific subset of the vertices of the hyperrectangle $\mathcal{B}_q$, see Sect. 4.2.2 for details. Then, computing the largest radius ρ of $\mathcal{B}_q(\rho)$ amounts to solving a one-dimensional problem in the variable ρ and, for each value of ρ, to verify if the LMIs (11.10) are satisfied for the vertices of $\mathcal{B}_q(\rho)$. Performing this worst-case analysis for the design matrices $\widehat{P}_N$ and $\widehat{Y}_N$, we obtain

$$\rho_{\text{wc}} \approx 0.12.$$

We conclude that the controller derived for the aircraft model is stable and attains quadratic performance for all values of $q \in \mathcal{B}_q(\rho)$, with $\rho \in [0, \rho_{\text{wc}}]$.

We now perform a probabilistic analysis, which consists of a Monte Carlo experiment which is based on random extractions of uncertainty samples. In this case, Algorithm 11.2 can be used. Then, we construct the performance degradation function, which shows how the probability of quadratic stability decreases as a function

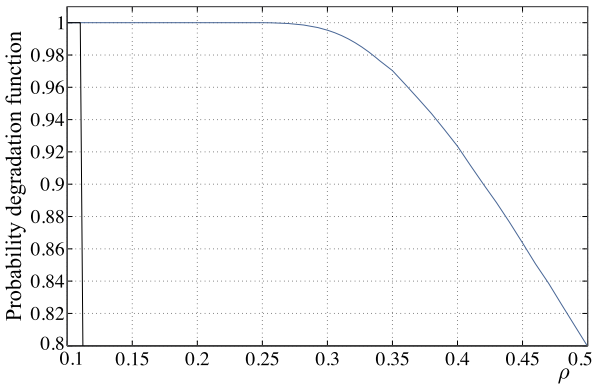

Fig. 11.4 Performance degradation function

of the radius ρ. This plot may be compared with the worst-case radius of performance ρ_{wc} that was previously computed. For instance, taking $\epsilon = 0.005$, $\delta = 10^{-6}$, by means of the Chernoff bound (8.14) we obtain $N = 290,174$. Then, we estimated the performance degradation function for 100 equispaced values of ρ in the range $[0.12, 0.5]$. For each grid point the estimated probability of performance is computed by means of Algorithm 11.2. For each value of ρ, the accuracy of this estimate satisfies the inequality with probability at least $1 - \delta$. The obtained results showing the estimated probability together with the deterministic radius ρ_{wc} are given in Fig. 11.4.

From this plot we observe, for instance, that if a 2 % loss of probabilistic performance may be tolerated, then the performance margin may be increased by approximately 270 % with respect to its deterministic counterpart. In fact, for $\rho = 0.34$, the estimated probability of performance is 0.98. In addition, we notice that the estimated probability is equal to one for values of the radius up to $\rho \approx 0.26$.

In Fig. 11.5 we plot the closed-loop eigenvalues for $\rho = 0.34$.

Example 11.2 (Gain scheduling control) We present an example regarding a gradient update rule for a control problem that is expressed in the form of an LMI feasibility condition. Consider a linear system depending on a time-varying parameter $q(t) = [q_1(t) \cdots q_\ell(t)]^T$

$$\dot{x} = A\big(q(t)\big)x + B_2 u.$$

For fixed t, the parameter vector q is constrained in the box $\mathcal{B}_q = \{q \in \mathbb{R}^\ell : |q_i| \leq \rho_i, i = 1, \ldots, \ell\}$, where ρ_i denote the allowed limits of variations of the parameters. We assume that the parameter $q(t)$ can be measured on-line, and hence this information is available to the controller. This setup is indeed a special case of a more general class of control problems usually known as linear parameter varying (LPV) problems, see Sect. 19.1. Here, we specifically consider a state feedback controller of the form

$$u = K\big(q(t)\big)x$$

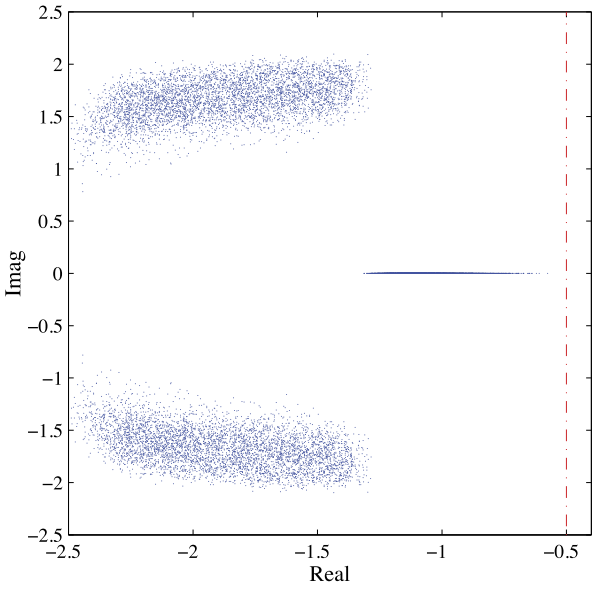

Fig. 11.5 Closed-loop eigenvalues for $\rho = 0.34$

where we set

$$K\big(q(t)\big) = K_0 + \sum_{i=1}^{\ell} K_i q_i(t).$$

The control objective is to determine the matrix gains K_i, $i = 0, 1, \ldots, \ell$, such that the controlled system has a guaranteed exponential decay rate $\alpha > 0$. Defining the closed loop matrix

$$A_{cl}\big(q(t)\big) = A\big(q(t)\big) + B_2 K\big(q(t)\big)$$

the control objective is met if there exists a symmetric matrix $P \succ 0$ such that the linear matrix inequality

$$A_{cl}(q)P + PA_{cl}^T(q) + 2\alpha P \prec 0 \tag{11.13}$$

holds for all $q \in \mathcal{B}_q$. We notice that, while there is no actual "uncertainty" in the plant (since the parameter $q(t)$ is measured on-line), the resulting design condition (11.13) is a robust matrix inequality, i.e. a condition that should be satisfied for all possible values of a "formal" uncertain parameter $q \in \mathcal{B}_q$, see Sect. 4.2.2 for details. In particular, introducing the new variables

$$Y_i = K_i P$$

for $i = 0, 1, \ldots, \ell$, the control design problem is expressed as a robust LMI feasibility problem. That is, we determine $P = P^T \in \mathbb{R}^{4,4}$, $Y_i = K_i P \in \mathbb{R}^{2,4}$, for $i = 0, 1, \ldots, \ell$, such that

$$
\begin{aligned}
&P \succ 0 \\
&0 \succ A(q)P + PA^T(q) + 2\alpha P \\
&\qquad + B_2 Y_0 + Y_0^T B_2^T + \sum_{i=1}^{\ell} q_i B_2 Y_i + \sum_{i=1}^{\ell} q_i Y_i^T B_2^T
\end{aligned}
\tag{11.14}
$$

for all $q \in \mathcal{B}_q$. Equations (11.14) represent a robust LMI condition of the form $F(q,\theta) \preceq 0$, where the design variable θ contains the free entries of P and Y_i, $i = 0, 1, \ldots, \ell$, and q represents parametric uncertainty.

For numerical simulations, we considered the same data as in Example 11.1 but in this case we set to their nominal values the parameters $L_{\delta_a}, Y_{\delta_r}, N_{\delta_r}, N_{\delta_a}$ in Table 11.1 and we consider the input matrix B_2 fixed

$$
B_2 = \begin{bmatrix} 0 & 0 \\ 0 & -3.91 \\ 0.035 & 0 \\ -2.53 & 0.31 \end{bmatrix}.
$$

Assume that the values of the $\ell = 9$ random uncertainties $\mathbf{q}$ of the plant can be measured on-line, and that the bounds ρ_i are equal to 15 % of the nominal values given in Table 11.1. Setting the desired decay rate to $\alpha = 0.5$, we applied Algorithm 11.2 to determine a probabilistically feasible solution for the system of LMIs (11.14). The probability levels were set to $\epsilon = 0.1$ and $\delta = 0.01$, and we selected feasibility radius equal to 0.05. The probability distribution of the parameters $\mathbf{q}$ was set to the uniform distribution over $\mathcal{B}_q$. Since the LMIs (11.14) are homogeneous in P, we added a condition number constraint on P of the form $I \preceq P \preceq \beta P$, with $\beta = 1{,}000$. Algorithm 11.2 converged in $N = 48$ outer iterations to the solution

$$
\widehat{P}_N = \begin{bmatrix} 1.2675 & -1.1846 & -0.0142 & 0.1013 \\ -1.1846 & 8.3174 & 0.2765 & -0.5450 \\ -0.0142 & 0.2765 & 1.2810 & 0.5769 \\ 0.1013 & -0.5450 & 0.5769 & 2.4329 \end{bmatrix};
$$

$$
\widehat{K}_{0N} = \begin{bmatrix} 0.0023 & 0.0768 & -1.0892 & 0.5657 \\ 1.3496 & -0.1164 & -0.9659 & 0.1213 \end{bmatrix};
$$

$$
\widehat{K}_{1N} = \begin{bmatrix} 0.0826 & 0.0353 & -0.1685 & 0.2000 \\ 0.3875 & 0.0808 & -0.0610 & 0.0226 \end{bmatrix};
$$

$$
\widehat{K}_{2N} = \begin{bmatrix} -0.0127 & -0.0090 & 0.0359 & -0.0975 \\ 0.1151 & 0.0074 & 0.2536 & -0.0489 \end{bmatrix};
$$

$$
\widehat{K}_{3N} = \begin{bmatrix} 0.0207 & 0.0039 & 0.0023 & 0.0039 \\ 0.0291 & 0.0075 & -0.0301 & 0.0107 \end{bmatrix};
$$

$$
\widehat{K}_{4N} = \begin{bmatrix} 0.0011 & 0.0005 & -0.0026 & 0.0026 \\ 0.0042 & 0.0009 & -0.0016 & 0.0004 \end{bmatrix};
$$

$$
\widehat{K}_{5N} = \begin{bmatrix} -0.0016 & -0.0006 & 0.0039 & -0.0006 \\ -0.0014 & -0.0004 & 0.0018 & -0.0018 \end{bmatrix};
$$

$$\widehat{K}_{6N} = \begin{bmatrix} -0.1134 & -0.0029 & -0.1616 & 0.2304 \\ 0.0544 & 0.0162 & -0.0916 & -0.0388 \end{bmatrix};$$

$$\widehat{K}_{7N} = \begin{bmatrix} 0.0023 & 0.0002 & 0.0025 & -0.0021 \\ 0.0004 & 0.0001 & -0.0004 & 0.0007 \end{bmatrix};$$

$$\widehat{K}_{8N} = \begin{bmatrix} -0.1134 & -0.0029 & -0.1616 & 0.2304 \\ 0.0544 & 0.0162 & -0.0916 & -0.0388 \end{bmatrix};$$

$$\widehat{K}_{9N} = \begin{bmatrix} -0.0024 & -0.0001 & -0.0026 & 0.0083 \\ 0.0012 & -0.0001 & 0.0021 & -0.0035 \end{bmatrix}.$$

This solution was tested a posteriori using the Monte Carlo method, showing that it actually satisfies the design LMIs with estimated probability higher than 0.998.

11.4 Sequential Methods for Optimization

The sequential randomized methods discussed in the previous sections are aimed at solving feasibility problems. However, these techniques can also be adapted to address optimization problems, by suitably embedding them into bisection or similar techniques. The basic idea is to fix an initial objective level γ and reformulate the optimization Problem 10.1 into feasibility epigraphic format as

$$\text{Find } \theta \text{ such that:} \quad c^T\theta \leq \gamma, V(\theta) \leq \epsilon,$$

and iteratively adjust the level γ until it cannot be further reduced while maintaining feasibility. Sequential randomized algorithm for optimization which use stochastic bisection have been introduced in [169, 410], where an improvement based on iterative decrease of the objective level is proposed. An alternative approach based on an ellipsoidal algorithm with deep cuts has been recently proposed in [28].

Chapter 12
Scenario Approach to Probabilistic Design

This chapter presents the so-called *scenario approach* for probabilistic design, as an alternative to the sequential methods analyzed in Chap. 11. As extensively discussed in this book, the prime motivation for studying robustness problems in engineering comes from the fact that the actual system (a "plant," or in general a problem involving physical data) upon which the engineer should act, is realistically not fixed and certain, but rather it entails some level of uncertainty. For instance, the characterization of the model of some industrial plant G depends on the value of physical parameters. If measurements of these parameters are performed, say, on different days or under different operating conditions, it is likely that we will end up not with a single plant G but with a finite collection $\mathcal{G}_N = \{G^{(1)}, \ldots, G^{(N)}\}$ of possible plants, each corresponding to a different realization (scenario) of the uncertain parameters upon which the plant depends. If the task of the problem solver is to devise a once-and-for-all fixed policy that performs well on the actual (unknown) plant, a sensible strategy would be to design this policy such that it performs well on all the collected scenarios $\mathcal{G}_N$. This is of course a well-established technique which is widely used in practical problems, and it is for instance the standard way in which uncertainty is dealt with in difficult financial planning problems, such as multi-stage stochastic portfolio optimization, see e.g. [122].

While simple and effective in practice, the scenario approach also raises interesting theoretical questions. First, it is clear that a design that performs well for given scenarios may fail in the standard deterministic worst-case sense, unless the considered scenarios $\mathcal{G}_N$ actually contain all possible realizations of the uncertain parameters. Then, it becomes natural to ask what is the relation between robustness in the scenario sense and the probabilistic approach discussed extensively in this book. It turns out that a design based on scenarios actually guarantees a specified level of probabilistic robustness, provided that the number N of scenarios is chosen properly. In this sense, a design based on a number of randomly chosen scenarios fits exactly in the definitions of randomized algorithm for probabilistic design given in Chap. 10. The scenario approach has been pioneered in [79, 87] and further developed in [95] and [78, 96]. A special case where constraints are affine in θ has also been studied earlier, in the context of system

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_12,

identification, in [94, 311], based on the idea of leave-one-out estimation presented in [401].

12.1 Three Design Paradigms

With the notation set in the previous chapters, we let $\boldsymbol{\Delta} \in \mathcal{B}_{\mathbb{D}}$ be the random uncertainty acting on the system, and we let $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ be the design vector, which includes controller parameters, as well as possibly other additional variables. The set Θ represents the domain of the optimization variables, as well as all the constraints in the problem that do not depend on the uncertain parameters. We shall next assume that Θ is a convex and compact set.[1] We let further $J(\boldsymbol{\Delta}, \theta)$ be a performance function for the closed-loop system, and γ an assigned performance level. We again make a standing assumption on the convexity of $J(\boldsymbol{\Delta}, \theta)$, see Assumption 10.1.

In this context, a typical design problem is to determine a design parameter $\theta \in \Theta$ such that

$$J(\boldsymbol{\Delta}, \theta) \leq \gamma$$

holds "robustly" with respect to the possible realizations of $\boldsymbol{\Delta}$. We remain voluntarily vague as to the meaning of robustness, since we now define three different ways in which this can be intended.

The first paradigm is of course the deterministic worst-case one, in which we seek θ such that

$$J(\boldsymbol{\Delta}, \theta) \leq \gamma, \quad \text{for all } \boldsymbol{\Delta} \in \mathcal{B}_{\mathbb{D}}.$$

To add some generality to this problem, we next consider a situation where an optimization can be performed over the design parameters $\theta \in \Theta$. This gives rise to the *worst-case design problem* defined below.

Problem 12.1 (Worst-case design) *Let Assumption 10.1 be satisfied. The worst-case design problem is to determine $\theta \in \Theta$ that solves*

$$\begin{aligned} &\min_{\theta} \; c^T \theta \\ &\text{subject to } J(\boldsymbol{\Delta}, \theta) \leq \gamma \quad \text{for all } \boldsymbol{\Delta} \in \mathcal{B}_{\mathbb{D}}. \end{aligned} \tag{12.1}$$

Clearly, the feasibility problem, when one is only interested in determining a solution θ that satisfies the constraints, is simply recovered as a special case of the above optimization problem, introducing a slack variable and rewriting the constraint in epigraphic form.

[1] For problems where the domain of optimization is $\mathbb{R}^{n_\theta}$, we shall just assume that Θ is some "very large box" representing maximum amplitude limits on the decision variables. This can always be done in practice, without modifying the nature of the optimization problem at hand.

A second paradigm is the probabilistic one: if $\boldsymbol{\Delta}$ is a random variable with assigned probability distribution over $\mathcal{B}_{\mathbb{D}}$, then the probabilistic design objective is to determine a parameter θ such that, for $\epsilon \in (0, 1)$

$$\Pr\left\{ J(\boldsymbol{\Delta}, \theta) \leq \gamma \right\} \geq 1 - \epsilon$$

where $1 - \epsilon$ is a given desired level of accuracy. Again, considering the more general situation in which we optimize over the design parameters, we can state a *probabilistic design problem*, which coincides with the "reliable design optimization problem" defined in Problem 10.2, and which is reported again below for clarity of exposition.

Problem 12.2 (Probabilistic design) *Let Assumption 10.1 be satisfied. The probabilistic design problem is to determine $\theta \in \Theta$ that solves*

$$\min_{\theta} \; c^T \theta \tag{12.2}$$

$$\text{subject to } \Pr\left\{ J(\boldsymbol{\Delta}, \theta) \leq \gamma \right\} \geq 1 - \epsilon \tag{12.3}$$

for some assigned probability level $\epsilon \in (0, 1)$.

Finally, we define the scenario approach: let $\boldsymbol{\Delta}^{(1,\dots,N)}$ denote a multisample $\boldsymbol{\Delta}^{(1)}, \dots, \boldsymbol{\Delta}^{(N)}$ of independent samples of $\boldsymbol{\Delta}$ extracted according to some probability distribution. Since $\boldsymbol{\Delta}^{(i)} \in \mathcal{B}_{\mathbb{D}}$, $i = 1, \dots, N$, we have that $\boldsymbol{\Delta}^{(1,\dots,N)} \in \mathcal{B}_{\mathbb{D}}^N$, where $\mathcal{B}_{\mathbb{D}}^N$ is the Cartesian product $\mathcal{B}_{\mathbb{D}} \times \cdots \times \mathcal{B}_{\mathbb{D}}$ (N times). Then, $\boldsymbol{\Delta}^{(1,\dots,N)} \in \mathcal{B}_{\mathbb{D}}^N$ represents the randomly selected scenarios from the uncertain system, and we define the following *scenario design problem*.

Problem 12.3 (Scenario design) *Let Assumption 10.1 be satisfied. For randomly extracted scenarios $\boldsymbol{\Delta}^{(1,\dots,N)}$, the scenario design problem is to determine $\theta \in \Theta$ that solves*

$$\begin{aligned} &\min_{\theta} \; c^T \theta \\ &\text{subject to } J\left(\boldsymbol{\Delta}^{(i)}, \theta\right) \leq \gamma, \quad i = 1, \dots, N. \end{aligned} \tag{12.4}$$

12.1.1 Advantages of Scenario Design

The scenario design has a distinctive advantage over its two competitors: it is computationally tractable. Indeed, we already know (see, e.g., Chap. 5) that worst-case design problems are, in general, computationally hard. Even under the convexity assumption (Assumption 10.1), the optimization Problem 12.1 resulting from the worst-case design approach usually entails a infinite number of constraints. This class of problems goes under the name of robust convex programs, which are known to be NP-hard, [50, 158], see also Sect. 4.3.

It is worth stressing that the probabilistic design approach does *not* alleviate the computational complexity issue. In fact, even under the convexity assumption, Problem 12.2 is extremely hard to solve exactly in general. This is due to the fact that the probability in the so-called "chance constraint" (12.3) is hard to compute explicitly and, more precisely, the restriction imposed on θ by (12.3) is in general *nonconvex*, even though $J(\Delta, \theta)$ is convex in θ, see for instance [51, 330, 396].

Contrary to the two situations above, the scenario design defined in Problem 12.3 is a standard convex program, with a finite number of constraints, and as such it is usually computationally tractable. For instance, in control problems where the performance function is expressed (for each given Δ) as linear matrix inequality constraints on the θ parameter, the scenario design simply amounts to solving a standard semidefinite program.

In addition to being efficiently computable, the solution $\widehat{\boldsymbol{\theta}}_N$ resulting from a scenario optimal design has an additional fundamental property: with high probability, it also satisfies the probability constraint (12.3). In other words, if the number N of scenarios is chosen properly, then the optimal solution returned by the scenario design is (with high probability) also robust in the probabilistic sense required by constraint (12.3). In this sense, the scenario approach provides a reliable design, according to Definition 10.6.

12.2 Scenario Design

In this section we analyze in further detail the properties of the solution of Problem 12.3. The multisample $\boldsymbol{\Delta}^{(1,\dots,N)} \in \mathcal{B}_{\mathbb{D}}^N$ contains the collection of iid random uncertainties $\boldsymbol{\Delta}^{(1)}, \dots, \boldsymbol{\Delta}^{(N)}$, and events related to $\boldsymbol{\Delta}^{(1,\dots,N)}$ are measured by the product probability $\mathrm{Pr}_{\boldsymbol{\Delta}^{(1,\dots,N)}}$. Clearly, in the absence of further assumptions, it may happen that for some extraction $\boldsymbol{\Delta}^{(1,\dots,N)}$ the scenario problem (12.4) is unfeasible. On the contrary, for any extraction for which the problem is feasible, the problem attains an optimal solution (due to compactness of Θ) and, for simplicity we assume that such a solution is unique. We thus denote with $\mathcal{B}_{\mathbb{D}}^{*N}$ the subset of $\mathcal{B}_{\mathbb{D}}^N$ containing those multiextractions that lead to a feasible scenario problem (and hence to an optimal solution). Notice that problem (12.4) is certainly feasible whenever the worst-case problem (12.1) is feasible, since the former involves a subset of the constraints of the latter; in such a special case we have that $\mathcal{B}_{\mathbb{D}}^{*N} = \mathcal{B}_{\mathbb{D}}^N$.

Let us denote with $\widehat{\boldsymbol{\theta}}_N$ the optimal solution of (12.4), whenever it exists. The optimal solution $\widehat{\boldsymbol{\theta}}_N$ is a random variable, since it depends on the sampled random scenarios $\boldsymbol{\Delta}^{(1)}, \dots, \boldsymbol{\Delta}^{(N)}$. We thus define the constraint violation probability for the scenario solution as

$$V(\widehat{\boldsymbol{\theta}}_N) = \begin{cases} \mathrm{Pr}\{J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_N) > \gamma\}, & \text{if } \boldsymbol{\Delta}^{(1,\dots,N)} \in \mathcal{B}_{\mathbb{D}}^{*N}; \\ 1, & \text{otherwise.} \end{cases} \tag{12.5}$$

Note that $V(\widehat{\boldsymbol{\theta}}_N)$ is itself random variable with support in the interval $[0, 1]$, and events related to $V(\widehat{\boldsymbol{\theta}}_N)$ are measured by the product probability $\mathrm{PR}_{\boldsymbol{\Delta}^{(1,\dots,N)}}$. Equivalently, we may define the *reliability* for a scenario solution as $R(\widehat{\boldsymbol{\theta}}_N) = 1 - V(\widehat{\boldsymbol{\theta}}_N)$, according to the notation set in Definition 10.5, and we write

$$R(\widehat{\boldsymbol{\theta}}_N) = \mathrm{PR}\left\{J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_N) \leq \gamma\right\} \cdot \mathbb{I}_{\mathcal{B}^{*N}_{\mathbb{D}}}\left(\boldsymbol{\Delta}^{(1,\dots,N)}\right),$$

where $\mathbb{I}_{\mathcal{B}^{*N}_{\mathbb{D}}}$ is the indicator function of the set $\mathcal{B}^{*N}_{\mathbb{D}}$. Notice that these definitions of $V(\widehat{\boldsymbol{\theta}}_N)$, $R(\widehat{\boldsymbol{\theta}}_N)$ coincide with those reported in Definition 10.5, if Problem 12.4 is feasible with probability one.

The following key result from [78] (see Corollary 3.4 in this reference) establishes the connection between the scenario approach and the probabilistic approach to robust design.

Theorem 12.1 (Scenario optimization) *Let Assumption 10.1 be satisfied, let Θ be convex and compact, and assume that Problem 12.3, when feasible, attains a unique optimal solution. Let $\epsilon \in (0, 1)$ be a given probability level and let $N \geq n_\theta + 1$. Then, it holds that*

$$\mathrm{PR}_{\boldsymbol{\Delta}^{(1,\dots,N)}}\left\{\left\{V(\widehat{\boldsymbol{\theta}}_N) > \epsilon\right\} \cap \mathcal{B}^{*N}_{\mathbb{D}}\right\} \leq \mathrm{B}_{N,\epsilon}(n_\theta) \tag{12.6}$$

where $\mathrm{B}_{N,\epsilon}(n_\theta)$ is the binomial distribution defined in (2.4), i.e.,

$$\mathrm{B}_{N,\epsilon}(n_\theta) = \sum_{k=0}^{n_\theta} \binom{N}{k} \epsilon^k (1-\epsilon)^{N-k}. \tag{12.7}$$

We remark that the right-hand-side term in (12.6) goes to zero rapidly as N increases. This means that, for suitably large number N of scenarios, the probability of getting a solution which is "bad" in terms of violation probability (i.e. having violation larger than the assigned ϵ) is extremely low. Equation (12.6) can also be "inverted" so to make explicit the dependence of N on the required probability levels, as specified in the next corollary (see Corollary 5.1 in [78]); see also [10] and Sect. 8.5 for refined bounds which show that the constant 2 in Eq. (12.8) can be reduced to $\mathrm{e}/(\mathrm{e}-1) \approx 1.58$.

Corollary 12.1 (Scenario optimization) *Let the assumptions in Theorem 12.1 be satisfied, and let $\epsilon, \delta \in (0, 1)$ be given probability levels. If N is an integer such that*

$$N \geq \frac{2}{\epsilon}\left(\log\frac{1}{\delta} + n_\theta\right) \tag{12.8}$$

then it holds that

$$\mathrm{PR}_{\boldsymbol{\Delta}^{(1,\dots,N)}}\left\{\left\{V(\widehat{\boldsymbol{\theta}}_N) > \epsilon\right\} \cap \mathcal{B}^{*N}_{\mathbb{D}}\right\} \leq \delta.$$

This result states that if the number of scenarios is selected according to the bound (12.8), then the optimal solution returned by the scenario design has, with high probability $1 - \delta$, a guaranteed level of accuracy $1 - \epsilon$. Both Theorem 12.1

and Corollary 12.1 hold with slightly refined bounds under the additional assumption that the scenario problem is feasible with probability one, that is when $\Pr_{\boldsymbol{\Delta}^{(1,\dots,N)}}\{\mathcal{B}_{\mathbb{D}}^{*N} \cap \mathcal{B}_{\mathbb{D}}^{N}\} = 1$. In such a case (12.6) and (12.8) hold substituting $n_\theta - 1$ in place of n_θ, see [78]. Moreover, the bound in (12.6), with $n_\theta - 1$ in place of n_θ, holds with equality if the problem is feasible with probability one and an additional technical assumption (fully supported problem) holds, see [95]. We summarize the scenario-based design approach in the following algorithm.

Algorithm 12.1 (Scenario RA for probabilistic design) *Let the hypotheses of Theorem 12.1 be satisfied, let $\epsilon, \delta \in (0, 1)$ be given, and let N be an integer satisfying (12.8). Then, with probability at least $1 - \delta$, the following RA is either unfeasible, or it is feasible and it returns a decision variable $\widehat{\boldsymbol{\theta}}_N \in \Theta$ such that*

$$\Pr\left\{ J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_N) \leq \gamma \right\} \geq 1 - \epsilon.$$

1. Generate N random iid scenarios $\boldsymbol{\Delta}^{(1)}, \dots, \boldsymbol{\Delta}^{(N)}$;
2. Solve (12.4) with the given scenarios;
3. If feasible, return $\widehat{\boldsymbol{\theta}}_N$.

Remark 12.1 (Sample complexity and a priori vs. a posteriori probabilities) From Corollary 12.1, we conclude that the sample complexity, that is the number of constraints needed in optimization problem (12.4), is inversely proportional to the probability level ϵ, while it only increases with the logarithm of δ^{-1}. This means that level δ can be set to a very low value, say $\delta = 10^{-9}$, without increasing too much the number of required scenarios. For such low values of δ, we have that the occurrence of the undesirable event $\{V(\widehat{\boldsymbol{\theta}}_N) > \epsilon\} \cap \mathcal{B}_{\mathbb{D}}^{*N}\}$ is immaterial to all practical purposes. In other words, we may claim a priori that, with practical certainty, either the problem is unfeasible, or the solution will have small violation probability, i.e. $V(\widehat{\boldsymbol{\theta}}_N) \leq \epsilon$.

It is important to stress the difference between a-priori and a-posteriori levels of probability. Specifically, the a-priori levels in Theorem 12.1 are guaranteed *before* we run any actual optimization experiment. In contrast, once we run the optimization and hence have in our hands a *fixed* candidate design $\widehat{\theta}_N$, we can (and should) test this solution a posteriori, using a Monte Carlo test. This analysis gives us an estimate of the actual (a-posteriori) probability levels attached to the design. The a-posteriori test does not involve the solution of any optimization problem and can, therefore, be performed using a very large sample size, see Chap. 8.

We now study an example, taken from [120], showing an application of the scenario approach.

Example 12.1 (Scenario design) We revisit Example 11.1 which studies LQR design for the lateral motion of an aircraft subject to 13 uncertain parameters $q \in \mathcal{B}_q$.

In particular, we consider the problem of determining a probabilistic solution $\theta = \{P, Y\}$ of the uncertain convex optimization problem

$$\min_{P,Y} \operatorname{Tr} P \quad \text{subject to} \quad F(q, P, Y) \preceq 0 \quad \text{and} \quad P \succeq \beta I,\ q \in \mathcal{B}_q$$

where

$$F(q, P, Y) = A(q)P + PA^T(q) + B_2(q)Y + Y^T B_2^T(q) + 2\alpha P,$$

and $\alpha = 0.5$, $\beta = 0.01$. In this example we have $n_\theta = 18$ decision variables (corresponding to the free entries of P and Y). Hence, setting $\epsilon = 0.01$ and $\delta = 10^{-6}$ the bound (12.8) provides a sample size $N = 6,364$, which means that we need to solve a convex optimization problem with this number of LMI constraints and 18 design variables. Notice that there is no specific compact domain Θ for the decision variables in this example, hence we may simply assume that Θ is some "box" of large side length (this has no practical impact on the optimization problem). Algorithm 12.1 returned the solution

$$\widehat{P}_N = \begin{bmatrix} 0.1445 & -0.0728 & 0.0035 & 0.0085 \\ -0.0728 & 0.2192 & -0.0078 & -0.0174 \\ 0.0035 & -0.0078 & 0.1375 & 0.0604 \\ 0.0085 & -0.0174 & 0.0604 & 0.1975 \end{bmatrix};$$

$$\widehat{Y}_N = \begin{bmatrix} 0.0109 & 7.2929 & 0.0439 & 0.6087 \\ 0.0908 & 3.4846 & -0.0565 & -3.9182 \end{bmatrix}$$

from which we obtain the probabilistic controller

$$\widehat{K}_N = \begin{bmatrix} 20.0816 & 40.3852 & -0.4946 & 5.9234 \\ 10.7941 & 18.1058 & 9.8937 & -21.7363 \end{bmatrix}.$$

Example 12.2 (Fixed-order robust control design) The example reported below is an adaptation of a fixed-order robust controller design problem originally presented in [228]. Consider a plant described by the uncertain transfer function

$$G(s, q) = 2(1 + q_1)\frac{s^2 + 1.5(1 + q_2)s + 1}{(s - (2 + q_3))(s + (1 + q_4))(s + 0.236)}$$

where $q = [q_1 \ \ q_2 \ \ q_3 \ \ q_4]^T$ collects the uncertainty terms acting respectively on the DC-gain, the numerator damping, and the pole locations of the plant. In this example, we assume

$$\mathcal{B}_q = \big\{q : |q_1| \le 0.01,\ |q_2| \le 0.01,\ |q_3| \le 0.02,\ |q_4| \le 0.01\big\}.$$

The above uncertain plant can be rewritten in the form

$$G(s, q) = \frac{N_G(s, q)}{D_G(s, q)} = \frac{b_0(q) + b_1(q)s + b_2(q)s^2}{a_0(q) + a_1(q)s + a_2(q)s^2 + s^3}$$

where

$$b_0(q) = 2(1+q_1);$$
$$b_1(q) = 3(1+q_1)(1+q_2);$$
$$b_2(q) = 2(1+q_1);$$
$$a_0(q) = -0.236(2+q_3)(1+q_4);$$
$$a_1(q) = -(2+q_3)(1+q_4) + 0.236(q_4 - q_3) - 0.236;$$
$$a_2(q) = q_4 - q_3 - 0.764.$$

Now define the following target stable interval polynomial family

$$\mathcal{P} = \left\{ p(s) : p(s) = c_0 + c_1 s + c_2 s^2 + c_3 s^3 + s^4,\ c_i \in \left[c_i^-, c_i^+\right],\ i = 0, 1, \ldots, 3 \right\}$$

with

$$c^- \doteq \begin{bmatrix} c_0^- \\ c_1^- \\ c_2^- \\ c_3^- \end{bmatrix} = \begin{bmatrix} 38.25 \\ 57 \\ 31.25 \\ 6 \end{bmatrix}, \qquad c^+ \doteq \begin{bmatrix} c_0^+ \\ c_1^+ \\ c_2^+ \\ c_3^+ \end{bmatrix} = \begin{bmatrix} 54.25 \\ 77 \\ 45.25 \\ 14 \end{bmatrix}.$$

The robust synthesis problem we consider is to determine (if one exists) a first-order controller

$$K(s, \theta) = \frac{N_K(s)}{D_K(s)} = \frac{\theta_1 + \theta_2 s}{\theta_3 + s}$$

depending on the design parameter $\theta \doteq [\theta_1\ \theta_2\ \theta_3]^T$, such that the closed-loop polynomial of the system

$$\begin{aligned} p_{cl}(s, q) &= N_G(s, q) N_K(s) + D_G(s, q) D_K(s) \\ &= \big(b_0(q)\theta_1 + a_0(q)\theta_3\big) + \big(b_1(q)\theta_1 + b_0(q)\theta_2 + a_1(q)\theta_3 + a_0(q)\big)s \\ &\quad + \big(b_2(q)\theta_1 + b_1(q)\theta_2 + a_2(q)\theta_3 + a_1(q)\big)s^2 \\ &\quad + \big(b_2(q)\theta_2 + \theta_3 + a_2(q)\big)s^3 + s^4 \end{aligned}$$

belongs to $\mathcal{P}$, for all $q \in \mathcal{B}_q$. Then, defining

$$A(q) \doteq \begin{bmatrix} b_0(q) & 0 & a_0(q) \\ b_1(q) & b_0(q) & a_1(q) \\ b_2(q) & b_1(q) & a_2(q) \\ 0 & b_2(q) & 1 \end{bmatrix}, \qquad d(q) \doteq \begin{bmatrix} 0 \\ a_0(q) \\ a_1(q) \\ a_2(q) \end{bmatrix}$$

the robust synthesis conditions are satisfied if and only if

$$c^- \leq A(q)\theta + d(q) \leq c^+ \quad \text{for all } q \in \mathcal{B}_q. \tag{12.9}$$

To these linear constraints, we associate a linear objective vector $c^T \doteq [0\ 1\ 0]$, which amounts to seeking the robustly stabilizing controller having the smallest

high-frequency gain. We thus obtain the robust linear program

$$\min_{\theta} \; c^T\theta, \quad \text{subject to (12.9)}.$$

Solving this robust linear program corresponds to determining a worst-case design for the system, see Problem 12.1. Notice, however, that the numerical solution of this problem is not "easy," since the coefficients $a_i(q), b_i(q)$ do not lie in independent intervals and depend in a nonlinear way on q. Therefore, the approach of [228] cannot be directly applied in this case.

We hence proceed via the scenario approach: assuming a uniform density over $\mathcal{B}_q$ and fixing accuracy parameter to $\epsilon = 0.01$, and the confidence parameter to $\delta = 10^{-9}$, one can check that for $N = 2{,}903$ it holds that $\mathrm{B}_{N,\epsilon}(n_\theta) \leq \delta$, hence the desired a-priori probabilistic levels are attained by the scenario design. Then, $N = 2,903$ iid scenarios $\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(N)}$ are generated, and the scenario problem

$$\begin{aligned} &\min_{\theta \in \Theta} \; c^T\theta \\ &\quad \text{subject to } c^- \leq A\big(\mathbf{q}^{(i)}\big)\theta + d\big(\mathbf{q}^{(i)}\big) \leq c^+, \quad i = 1, \ldots, N \end{aligned}$$

is formed (assuming that Θ is a given very large box). The numerical solution of one instance of the above scenario linear program yielded the solution

$$\widehat{\theta}_N = [29.3881 \quad -2.1052 \quad 11.0379]^T$$

and hence the controller

$$K(s, \widehat{\theta}_N) = \frac{29.3881 - 2.1052s}{11.0379 + s}.$$

Once we have solved the synthesis problem, we can proceed to a Monte Carlo test in order to obtain an a-posteriori estimate of the probability of constraint violation for the computed solution. As discussed in Remark 12.1, we can use a much larger sample size for this a posteriori analysis, since no numerical optimization is involved in the process. Setting for instance $\epsilon = 0.001$, and $\delta = 0.00001$, from the Chernoff bound we obtain that the test should be run using at least $N = 6.103 \times 10^6$ samples. This test yielded an estimated probability of feasibility of about 0.997.

12.3 Scenario Optimization with Violated Constraints

In this section we discuss an important generalization of the basic scenario technique: scenario optimization with violated constraints. The idea of this technique is that one first collects N randomly generated scenarios (like in the basic scenario approach) and then purposely discards $r < N - n_\theta$ of them. The rationale behind this procedure is that scenarios correspond to constraints in the optimization problem, hence discarding a suitably selected set of scenarios *improves* the objective of the

optimization problem. Clearly, one can intuitively observe that discarding too many scenarios improves the objective but provides a solution with poor violation probability. The key issue here is indeed to find a suitable tradeoff between N and r so to improve the objective while preserving a desired level of probabilistic feasibility for the solution. To formalize the discussion, we define a *rule* R_r that selects which $N - r$ scenarios are retained in the optimization; similarly, $\overline{R}_r$ denotes the complement of this rule, indicating the r scenarios (constraints) that are violated by the optimal solution of the scenario problem with the R_r selected constraints. This rule is generic, with the only restriction of being invariant to permutations of the scenarios. That is, the rule depends on the set of extracted scenarios, and not on the order in which the scenarios are collected. For example, the rule can be defined as follows: remove from the observed multisample the r constraints that yield the best overall improvement in the objective. This would be a globally optimal constraint removal strategy (which can indeed be computationally hard to implement). However, we stress that the results below hold for any rule, hence also for some suboptimal one (e.g., remove successively the one constraint that gives the best improvement in the objective, until r constraints are removed).

The scenario problem with violated constraints amounts to solving a random optimization problem similar to (12.4), but considering only the scenarios selected by the rule. We thus consider the problem

$$\begin{aligned} &\min_{\theta\in\Theta} \; c^T\theta \\ &\quad \text{subject to } J\left(\boldsymbol{\Delta}^{(i)},\theta\right)\le\gamma, \quad i\in R_r, \end{aligned} \tag{12.10}$$

and we denote with $\mathcal{B}_{\mathbb{D}}^{*(N-r)}$ the subset of $\mathcal{B}_{\mathbb{D}}^{N}$ where problem (12.10) is feasible and, for $\boldsymbol{\Delta}^{(1,\dots,N)}\in\mathcal{B}_{\mathbb{D}}^{*(N-r)}$, we denote with $\widehat{\boldsymbol{\theta}}_{N-r}$ the resulting optimal solution. The violation probability for this solution is defined as

$$V(\widehat{\boldsymbol{\theta}}_{N-r}) = \begin{cases} \Pr\{J(\boldsymbol{\Delta},\widehat{\boldsymbol{\theta}}_{N-r})>\gamma\}, & \text{if } \boldsymbol{\Delta}^{(1,\dots,N)}\in\mathcal{B}_{\mathbb{D}}^{*(N-r)}; \\ 1, & \text{otherwise.} \end{cases} \tag{12.11}$$

The following key result from [78] (see Corollary 4.2 in this reference) establishes the connection between the scenario approach and the probabilistic approach to robust design, see also [96] for a similar result holding under an additional hypothesis of feasibility in all realizations.

Theorem 12.2 (Scenario optimization with violated constraints) *Let Assumption 10.1 be satisfied, let Θ be convex and compact, and assume that problem (12.10), when feasible, attains a unique optimal solution. Let $\epsilon\in(0,1)$ be a given probability level, and let N, r be integers such that $N - r \ge n_\theta + 1$. Then, it holds that*

$$\Pr_{\boldsymbol{\Delta}^{(1,\dots,N)}}\left\{\left\{V(\widehat{\boldsymbol{\theta}}_{N-r})>\epsilon\right\}\cap\mathcal{B}_{\mathbb{D}}^{*(N-r)}\right\}\le\widetilde{\mathrm{B}}_{N,r,\epsilon}(r+n_\theta), \tag{12.12}$$

where

$$\widetilde{\mathrm{B}}_{N,r,\epsilon}(r+n_\theta)\doteq\binom{r+n_\theta}{r}\mathrm{B}_{N,\epsilon}(r+n_\theta), \tag{12.13}$$

being $\mathrm{B}_{N,\epsilon}$ *the binomial distribution given in* (2.4) *and* $V(\cdot)$ *the constraint violation probability defined in* (12.11).

Following the same approach as in Corollary 12.1, we can "invert" bound (12.12) to derive an explicit lower bound for the number of scenarios N. Specifically, given $\delta \in (0, 1)$, we have that the left-hand-side of Eq. (12.12) is no larger than δ, provided that

$$N \geq \frac{2}{\epsilon} \log \frac{1}{\delta} + \frac{4}{\epsilon}(r + n_\theta), \tag{12.14}$$

see Corollary 5.1 in [78].

We next report two explicit algorithms for scenario optimization with violated constraints. Algorithm 12.2 uses a globally-optimal constraints removal rule, and it is formulated as a mixed-integer optimization problem (as such, it may be hard to solve numerically). Algorithm 12.3 uses instead a sub-optimal rule for constraints removal, based on local sensitivity of the objective, as measured by Lagrange dual variables, and it results in a sequence of efficiently solvable convex optimization problems. In this second algorithm, the constraints are removed iteratively in batches of n_{rem} at a time. At each iteration, the constraints to be removed are chosen among the active ones as those with highest values of the Lagrange multipliers. It is well known that the Lagrange multiplier λ_i represents the sensitivity of the objective value to variations of the i-th constraint. Hence, this suboptimal strategy aims at removing those constraints that, at each iteration, provide locally the largest improvement of the objective value.

Algorithm 12.2 (Scenario with optimal constraint removal) *Let the hypotheses of Theorem* 12.2 *be satisfied, let* $\epsilon, \delta \in (0, 1)$ *be given, and let* N, r *be integers satisfying* (12.14). *Then, with probability at least* $1 - \delta$, *the following RA is either unfeasible, or it returns* $\widehat{\boldsymbol{\theta}}_{N-r} \in \Theta$ *such that*

$$\Pr\left\{ J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_{N-r}) \leq \gamma \right\} \geq 1 - \epsilon.$$

1. Generate N random iid scenarios $\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(N)}$;
2. Solve the following mixed-integer optimization problem (for large M)

$$\begin{aligned} \min_{\theta \in \Theta, s \in \{0,1\}^N} \quad & c^T \theta \\ \text{subject to } & J\left(\boldsymbol{\Delta}^{(i)}, \theta\right) \leq \gamma + M s_i, \quad i = 1, \ldots, N, \\ & \sum_{i=1}^{N} s_i = r, \end{aligned}$$

   and let θ^* be the optimal solution (if it exists);
3. If feasible, return $\widehat{\theta}_{N-r} = \theta^*$.

Algorithm 12.3 (Scenario with suboptimal constraint removal) *Let the hypotheses of Theorem 12.2 be satisfied, let* $\epsilon, \delta \in (0,1)$ *be given, and let* N, r *be integers satisfying* (12.14)*. Then, with probability at least* $1-\delta$*, the following RA is either unfeasible, or it returns* $\widehat{\boldsymbol{\theta}}_{N-r} \in \Theta$ *such that*

$$\Pr\left\{J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_{N-r}) \leq \gamma\right\} \geq 1-\epsilon.$$

1. Let $C = \{1, \ldots, N\}$, $\texttt{rmvd} = 0$, $t = r$, n_{rmvd} integer such that $n_{\text{rmvd}} \leq n_\theta$;
2. Generate N random iid scenarios $\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(N)}$;
3. Solve the following convex optimization problem and its Lagrange dual

$$\min_{\theta \in \Theta} c^T \theta$$
$$\text{subject to } J\left(\boldsymbol{\Delta}^{(i)}, \theta\right) \leq \gamma, \quad i \in C,$$

   and let θ^* be its optimal solution (if it exists), and λ_i, $i \in C$, the optimal Lagrange multipliers associated with the constraints;
4. If $t = 0$, then return $\widehat{\theta}_{N-r} = \theta^*$, or `unfeas`, if the problem is unfeasible;
5. Let $t = \min(r - \texttt{rmvd}, n_{\text{rmvd}})$, and let D be the set of indices in C corresponding to the t largest λ_i's;
6. Let $C = C \setminus D$; $\texttt{rmvd} = \texttt{rmvd} + t$; Goto 3.

12.3.1 Relations with Chance-Constrained Design

In this section we briefly discuss how the optimal objective value obtained from a scenario approach with violated constraints may approximate arbitrarily well the optimal value of the chance-constrained Problem 12.2. To setup this comparison, we next assume that, for $\epsilon \in (0,1)$:

1. The optimal value of the chance-constrained problem in (12.2) is denoted by $\eta^*_{\text{ccp}}(\epsilon)$;
2. Integers N, r are chosen so that $\widetilde{\text{B}}_{N,r,\epsilon}(r+n_\theta) < 1$, where $\widetilde{\text{B}}_{N,r,\epsilon}$ is defined in (12.13);
3. The optimal value of the scenario problem with *globally optimal constraint removal rule* described in Algorithm 12.2 is denoted by $\boldsymbol{\eta}^*_{\text{sce}}$.

Under this setup, it is stated in Theorem 6.2 of [78] that, for any $\epsilon_1 < \epsilon$, it holds that

$$\Pr_{\boldsymbol{\Delta}^{(1,\ldots,N)}}\left\{\eta^*_{\text{ccp}}(\epsilon) \leq \boldsymbol{\eta}^*_{\text{sce}} \leq \eta^*_{\text{ccp}}(\epsilon_1)\right\} \geq \text{B}_{N,\epsilon_1}(r) - \widetilde{\text{B}}_{N,r,\epsilon}(r+n_\theta). \tag{12.15}$$

Furthermore, it is shown in [78] that one can *always* find (sufficiently large) N and r values such that the right-hand side of Eq. (12.15) is larger than $1-\delta$, for any pre-specified $\delta \in (0,1)$. This means that one can always find values for N and r such that the optimal value of a scenario problem with r violated constraints (optimally chosen) is "sandwiched" between the optimal chance-constrained problems

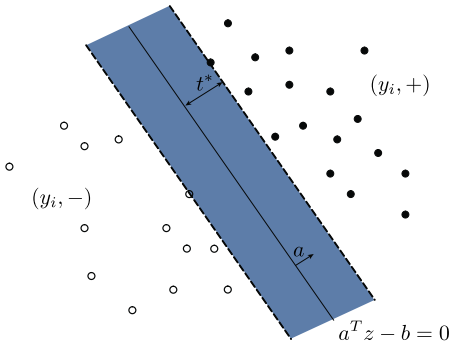

Fig. 12.1 Linear separation with margin

values $\eta^*_{\text{ccp}}(\epsilon)$, $\eta^*_{\text{ccp}}(\epsilon_1)$, with arbitrarily high probability. Since ϵ_1 can be chosen arbitrarily close to ϵ, this also means that the optimal scenario value may approximate arbitrarily well the value of the chance-constrained problem, with high probability.

Example 12.3 (Linear discrimination with soft margin) As an example of application of scenario optimization with violated constraints, we consider a classical problem of data classification using linear separation surfaces, which is commonly encountered, for instance, in the context of support vector machines with linear kernel, see, e.g., [116, 401].

Let $\mathbf{q} \doteq (\mathbf{y}, \mathbf{l})$ represent a random datum-label pair coming from a possibly unknown distribution, where for each realization $y \in \mathbb{R}^n$ is the datum and $\mathbf{l} \in \{-1, 1\}$ is a corresponding label. Suppose N observations are available from this distribution: $\mathbf{q}^{(i)} \doteq (\mathbf{y}_i, \mathbf{l}_i)$, $i = 1, \ldots, N$: the problem is to separate the points with label $l_i = +1$ from those with label $l_i = -1$, by means of an affine classifying function $f(y) = a^T y - b$, so to maximize the *margin* of separation between the two classes. This problem can formally be written as a convex program with N linear constraints and one second-order-cone constraint

$$\max_{t,b,\|a\|\leq 1} \quad t \tag{12.16}$$

$$\text{subject to } \mathbf{l}_i \left(a^T \mathbf{y}_i - b\right) \geq t, \quad i = 1, \ldots, N. \tag{12.17}$$

If the optimal value t^* of this program is positive, then the observed points are linearly separable, and it can be proved that $\|a\| = 1$ at the optimum, see, e.g., Sect. 8.6 in [70]. The geometric interpretation is the following: $a^T y_i - b$ represents the Euclidean distance of point y_i from the separating hyperplane $\mathcal{H} = \{z : a^T z - b = 0\}$. Hence, (12.17) imposes that all points with label $l_i = +1$ are at signed distance at least t from $\mathcal{H}$ and that all points with label $l_i = +1$ are at signed distance at least $-t$ from $\mathcal{H}$. Problem (12.16) thus determines the thickest slab separating the two data sets, where $2t^*$ is the maximal thickness of the slab, see Fig. 12.1.

Notice that it may happen that the two labeled data sets are not linearly separable. To deal with these situations, a standard approach used in linear support vector classifiers is to introduce slack variables $v_i \geq 0$ accounting for constraint violations (thus replacing (12.17) with $\mathbf{l}_i (a^T \mathbf{y}_i - b) \geq t - v_i$), and then augment the objective

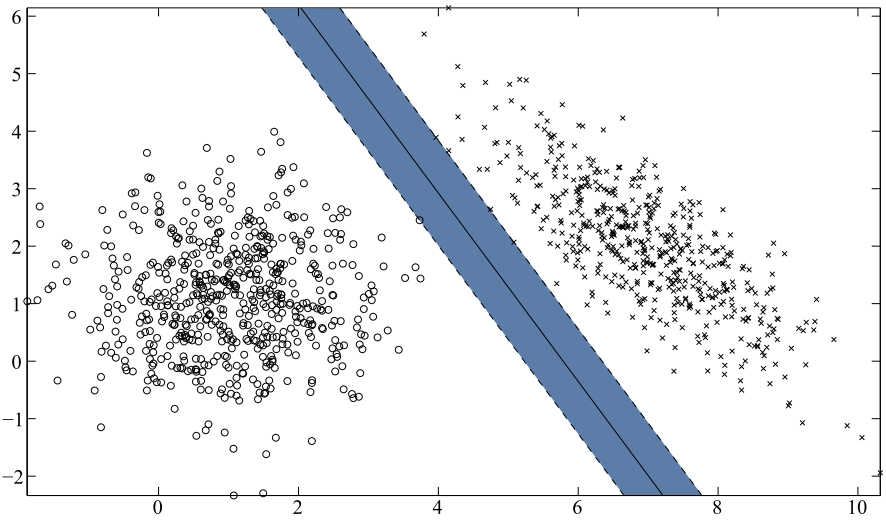

Fig. 12.2 Separation with margin $t_0^* = 0.4753$ (no a posteriori removed constraints)

of the optimization problem by subtracting a term proportional to the sum of violations. Intuitively, a support vector classifier determines a tradeoff between the "soft" margin of separation (width of the slab) and the number of observed points that fall inside the slab (shaded area in Fig. 12.1).

A fundamental question that arises at this point is the following one: once a classifier has been computed, what is the probability with which it will correctly classify a new unseen datum? This probability measures the *generalization* capability of the classifier. It turns out that the scenario theory is able to provide a sound answer to the previous question: consider (12.16) as an instance of a random convex program with N sampled constraints, and use a constraints removal procedure as defined in Sect. 12.3 to select $r \geq 0$ constraints to be violated. Clearly, as r increases the width of the soft separation margin increases (since we allow more and more points to fall inside the slab), while the generalization capability of the classifier degrades. However, we can precisely control this latter quantity via the constraint violation probability ϵ of the scenario theory. There are different ways to proceed, depending on which quantities are assigned a priori. For instance, we can fix the desired generalization probability $1 - \epsilon$ and a small δ, and then use Eq. (12.12) to determine a pair (N, r) such that the violation probability on the left-hand side is no larger than δ. For a numerical example, we generated the data as follows

$$\begin{cases} \mathbf{y} = \begin{bmatrix} 7 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \mathbf{w}, & \mathbf{l} = 1, \text{ with probability } 0.5; \\ \mathbf{y} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{w}, & \mathbf{l} = -1, \text{ with probability } 0.5, \end{cases}$$

where $\mathbf{w}$ is a standard Normal vector. The number of decision variables is $n_\theta = n + 2 = 4$. Setting $\epsilon = 0.01$ and $\delta = 10^{-6}$, we have that for $N = 1,128$ and $r = 15$

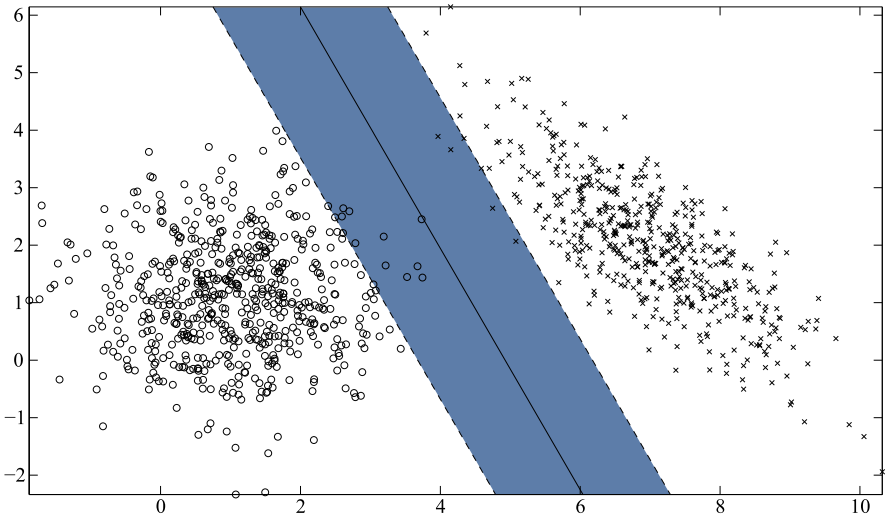

Fig. 12.3 Separation with margin $t_{15}^* = 1.1280$ ($r = 15$ a posteriori removed constraints)

the left-hand side of (12.12) is smaller than δ. Solving one instance of the scenario problem with $r = 0$, we obtained exact separation with an optimal margin $t_0^* = 0.4753$, see Fig. 12.2. Then, we removed $r = 15$ constraints via the suboptimal constraints removal Algorithm 12.3 with $n_{\text{rem}} = 1$, which resulted in an increased margin of $t_{15}^* = 1.1280$, see Fig. 12.3. In both cases, we are guaranteed (with high a priory confidence $1 - \delta$) that the resulting classifier will provide separation with margin t^* on a new observation, with probability at least $1 - \epsilon$.

Chapter 13
Learning-Based Probabilistic Design

In Chap. 9 we provided an overview of some key results of *statistical learning theory*, see [401, 406]. In this chapter we discuss their specific application to the design of systems affected by uncertainty. This line of research was initiated in [404, 405, 408], see also subsequent developments in [9, 240, 241].

We study a family of uncertain plants $\{G(s, \Delta) : \Delta \in \mathcal{B}_{\mathbb{D}}\}$ and a family of parameterized controllers of fixed order $\{K(s, \theta) : \theta \in \mathbb{R}^{n_\theta}\}$, where θ represents the vector of controller parameters. Consider a *binary* measurable performance function

$$J(\Delta, \theta) : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \{0, 1\}$$

which measures the performance of the controlled plant for given uncertainty Δ and controller parameters θ. Clearly, as θ varies over $\mathbb{R}^{n_\theta}$, $J(\cdot, \theta)$ spans an infinite family $\mathcal{J}$ of performance functions. As discussed in Chap. 10, we fix $\gamma = 1/2$, and we say that performance is satisfied when $J(\Delta, \theta) = 0$ and it is violated when $J(\Delta, \theta) = 1$.

We now formally define the worst-case design problem for nonconvex performance functions.

Problem 13.1 (Worst-case nonconvex design) *Given a binary measurable performance function* $J(\Delta, \theta) : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \{0, 1\}$ *and a bounded measurable function* $c : \mathbb{R}^{n_\theta} \to (-\infty, \infty)$, *compute a local minimum of the optimization problem*

$$\begin{aligned} &\min_{\theta} \ c(\theta) \\ &\text{subject to } J(\Delta, \theta) = 0, \quad \text{for all } \Delta \in \mathcal{B}_{\mathbb{D}}. \end{aligned} \tag{13.1}$$

We now provide three examples illustrating specific problems that can be reformulated in this form.

Example 13.1 (Design with an infinite number of inequality constraints) We consider a multiobjective control design problem which is subject to a set of uncertain constraints. Suppose that, given functions $f_i : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$, $i = 1, \ldots, m$, one desires to design $\theta \in \mathbb{R}^{n_\theta}$ such that the semi-infinite set of constraints

$$f_i(\Delta, \theta) \le 0, \quad i = 1, \ldots, m$$

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_13,

is satisfied for all $\Delta \in \mathcal{B}_{\mathbb{D}}$. If this problem is feasible, we may choose a specific solution within the feasible set to optimize a given criterion. This requires to solve a semi-infinite optimization problem of the form

$$\begin{aligned} &\min_{\theta} \ c(\theta) \\ &\quad \text{subject to } f_i(\Delta, \theta) \leq 0, \quad i = 1, \ldots, m, \ \text{for all } \Delta \in \mathcal{B}_{\mathbb{D}}. \end{aligned}$$

Then, design with an infinite number of inequality constraints can be reformulated into Eq. (13.1), provided that the binary performance function $J : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \{0, 1\}$ is defined as

$$J(\Delta, \theta) = \begin{cases} 0 & \text{if } f_i(\Delta, \theta) \leq 0, \ i = 1, \ldots, m; \\ 1 & \text{otherwise.} \end{cases}$$

Example 13.2 (Min-max problem with uncertainty) We study a min-max problem which is analyzed in the theory of differential games [34]. Consider a measurable function $g : \mathbb{D} \times \mathbb{R}^{n_\xi} \to (-\infty, \infty)$ and the min-max design problem

$$\min_{\xi \in \Xi} \sup_{\Delta \in \mathcal{B}_{\mathbb{D}}} g(\Delta, \xi)$$

where $\Xi \subset \mathbb{R}^{n_\xi}$ is a bounded set. The binary function $J : \mathbb{D} \times \mathbb{R}^{n_\theta} \to \{0, 1\}$ is defined as follows: given $\Delta \in \mathcal{B}_{\mathbb{D}}$ define $\theta = \{(\xi, \eta) : \xi \in \mathbb{R}^{n_\xi}, \eta \in (-\infty, \infty)\}$, with $n_\theta = n_\xi + 1$, then

$$J(\Delta, \theta) = \begin{cases} 0 & \text{if } g(\Delta, \xi) \leq \eta; \\ 1 & \text{otherwise} \end{cases}$$

and the objective function is defined as $c(\theta) = \eta$. Then, the min-max problem above can be immediately reformulated in the form of Eq. (13.1). We observe that this optimization problem is always feasible because g is a bounded function.

Example 13.3 (Static output feedback) Consider a strictly proper plant of the form

$$\begin{aligned} \dot{x} &= A(q)x + B_2(q)u; \\ y &= C(q)u \end{aligned}$$

where $x \in \mathbb{R}^{n_s}$ is the state, $u \in \mathbb{R}^{n_i}$ is the control input, $y \in \mathbb{R}^{n_o}$ is the measurement output and q represents real parametric uncertainty affecting the matrices $A(q)$, $B_2(q)$ and $C(q)$. The objective is to find, if it exists, a static output feedback law

$$u = Ky$$

which stabilizes the system for all $q \in \mathcal{B}_q$. Equivalently, we seek for a gain matrix $K \in \mathbb{R}^{n_i, n_o}$ such that the closed-loop system

$$A_{cl}(q, K) = A(q) + B_2(q)KC(q)$$

has all its eigenvalues in a specified region of the complex plane for all $q \in \mathcal{B}_q$. More precisely, we look for the optimal value of the gain matrix K that maximizes the convergence rate η to the origin. The decision variable θ consists of K and η, i.e. $\theta = (K, \eta)$ and $n_\theta = n_i n_o + 1$. Then, static output feedback may be reformulated in terms of Problem 13.1 as follows

$$\begin{aligned}&\min_{\eta,K}(-\eta)\\ &\quad\text{subject to } A_{cl}(q,\theta)+\eta I_{n_s} \text{ is Hurwitz for all } q\in\mathcal{B}_q;\\ &\qquad\qquad -\bar{K}_{i\ell}\le K_{i\ell}\le\bar{K}_{i\ell},\quad i=1,\dots,n_i;\ \ell=1,\dots,n_o.\end{aligned}$$

In this case, the performance function is defined as

$$J(q,\theta)=\begin{cases}0 & \text{if } A_{cl}(q,\theta)+\eta I_{n_s} \text{ is Hurwitz;}\\ 1 & \text{otherwise.}\end{cases}$$

As discussed in Chap. 5, the static output feedback problem has been shown to be NP-hard when bounds on the gain matrix are given, even if no uncertainty affects the plant. Convex relaxations leading to sufficient conditions for the existence of a stabilizing gain matrix K have been derived in the literature. For example, in [157] a sufficient condition has been obtained using an LMI approach.

13.1 Sample Complexity of Nonconvex Scenario Design

The binary optimization problem (13.1) is obviously very difficult to solve because the equality constraint $J(\Delta,\theta)=0$ is generally nonconvex in the decision variable θ. Moreover, the set $\mathcal{B}_{\mathbb{D}}$ has infinite cardinality. For these reasons, we develop a randomization approach based on statistical learning theory. We are not making an attempt to find a global minimum (i.e. we are satisfied with a *local minimum*), but rather to reformulate a randomized version of the semi-infinite optimization problem (13.1) which is subject to a *finite* number of constraints. To this end, we use the concept of empirical mean and formally assume that the uncertainty $\boldsymbol{\Delta}\in\mathcal{B}_{\mathbb{D}}$ is random with given pdf and support $\mathcal{B}_{\mathbb{D}}$, see Assumption 6.1. In this approach, no probability density function is introduced in the space of controller parameters, see Remark 13.1.

Since the uncertainty $\boldsymbol{\Delta}$ is random, we draw N samples $\boldsymbol{\Delta}^{(1)},\dots,\boldsymbol{\Delta}^{(N)}$ according to a given pdf $f_{\boldsymbol{\Delta}}$ and we introduce the sampled counterpart of Problem 13.1 as follows.

Problem 13.2 (Nonconvex scenario design) *Given a performance function* $J(\Delta,\theta):\mathbb{D}\times\mathbb{R}^{n_\theta}\to\{0,1\}$, *a density* $f_{\boldsymbol{\Delta}}(\boldsymbol{\Delta})$ *with support* $\mathcal{B}_{\mathbb{D}}$, *compute a* local *minimum* $\widehat{\boldsymbol{\theta}}_N$ *to the nonconvex optimization problem*

$$\begin{aligned}&\min_{\theta}\ c(\theta)\\ &\quad\text{subject to } J\left(\boldsymbol{\Delta}^{(i)},\theta\right)=0,\quad i=1,\dots,N.\end{aligned}\tag{13.2}$$

Note that this is the same spirit of the scenario approach discussed in Chap. 12. To analyze the sample complexity of the nonconvex scenario problem in the context of learning theory, we reformulate the constraints in (13.2) in terms of empirical mean. This is possible since J is a binary function. That is, we form the empirical mean of $J(\boldsymbol{\Delta},\theta)$ as

$$\widehat{\mathbf{E}}_N\left(J(\boldsymbol{\Delta},\theta)\right)=\frac{1}{N}\sum_{i=1}^{N}J\left(\boldsymbol{\Delta}^{(i)},\theta\right).$$

Then, we have

$$J\left(\boldsymbol{\Delta}^{(i)},\theta\right)=0 \text{ for } i=1,\ldots,N \quad \text{if and only if} \quad \widehat{\mathbf{E}}_N\left(J(\boldsymbol{\Delta},\theta)\right)=0.$$

We observe that the empirical mean $\widehat{\mathbf{E}}_N(J(\boldsymbol{\Delta},\theta))$ is a random variable. Since $J(\cdot,\cdot)$ is a binary function, $\widehat{\mathbf{E}}_N(J(\boldsymbol{\Delta},\theta))$ is always within the closed interval $[0,1]$. Hence, we rewrite problem (13.2) as follows

$$\min_{\theta} \; c(\theta) \tag{13.3}$$

$$\text{subject to } \widehat{\mathbf{E}}_N\left(J(\boldsymbol{\Delta},\theta)\right)=0. \tag{13.4}$$

Remark 13.1 (Design approaches based on controller randomization) In the absence of a convexity assumption, Problem 13.2 requires the solution of a nonconvex optimization problem. Previous work [169, 405] focused on the use of randomization in the parameter space to obtain a finite set of random designs. In this way, the so-called near minima are obtained in polynomial time, employing results for finite families of controllers. However, for controller design, as discussed in [241], a simple Monte Carlo optimization scheme can be very misleading because the obtained empirical minimum may be much larger than the true one with probability practically equal to one. Moreover, there is no "physical reason" for considering a pdf in controller space. We observe that the approach proposed in this chapter is not based on controller randomization.

To derive the sample complexity of nonconvex scenario design, we focus on a specific class of performance functions $J(\boldsymbol{\Delta},\theta)$, namely the class of (α,m)-Boolean functions.

Assumption 13.1 ((α,m)-Boolean function) *The function $J:\mathbb{D}\times\mathbb{R}^{n_\theta}\to\{0,1\}$ is a (α,m)-Boolean function. That is, for fixed Δ, it can be written as an expression consisting of Boolean operators involving m polynomials*

$$\beta_1(\theta), \quad \beta_2(\theta), \quad \ldots, \quad \beta_m(\theta)$$

in the variables θ_i, $i=1,\ldots,n_\theta$, and the degree with respect to θ_i of all these polynomials is no larger than $\alpha>0$.

We now formally state a result, see [9] for proof, which establishes the sample complexity for this class of performance functions, so that a solution of the randomized Problem 13.2 complies with the definition of reliable design given in Chap. 10.

Theorem 13.1 (Sample complexity for nonconvex scenario design) *Let Assumption 13.1 be satisfied, and let $\epsilon\in(0,0.14)$ and $\delta\in(0,1)$ be given probability levels. If*

$$N\geq\frac{4.1}{\epsilon}\left(\log\frac{21.64}{\delta}+4.39n_\theta\log_2\left(\frac{8e\alpha m}{\epsilon}\right)\right) \tag{13.5}$$

then, with probability at least $1-\delta$,

1. *either Problem 13.2 is unfeasible and, hence, also Problem 13.1 is unfeasible;*

2. *or Problem 13.2 is feasible, and then any local solution* $\widehat{\boldsymbol{\theta}}_N$ *satisfies the inequality*

$$V(\widehat{\boldsymbol{\theta}}_N) = \Pr\left\{J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_N) = 1\right\} \leq \epsilon.$$

We now comment on this theorem. Obviously, the solution $\widehat{\boldsymbol{\theta}}_N$ obtained solving Problem 13.2 is random, because it is based on the multisample $\boldsymbol{\Delta}^{(1\ldots N)}$. We also observe that $\Pr\{J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_N) = 1\}$ represents indeed the probability that the constraint $J(\boldsymbol{\Delta}, \theta) = 0$ appearing in (13.1) is violated. This is in accordance with the definition of violation probability for binary performance given in (10.11). Hence, the theorem above states that *any* local solution $\widehat{\boldsymbol{\theta}}_N$ of the nonconvex scenario problem (13.2) violates the constraints in (13.1) with a probability at most ϵ, and this event occurs with probability no smaller than $1 - \delta$. We conclude that $\widehat{\boldsymbol{\theta}}_N$ is a reliable design with high probability.

We remark that Theorem 13.1 does not require the explicit computation of the VC dimension, which is already embedded into the constants appearing into (13.5). We also notice that, if $\epsilon \in (0, 0.14)$ and $2\epsilon \mathrm{e} \alpha m \leq 1$, then

$$\log_2 \frac{8\mathrm{e}\alpha m}{\epsilon} \leq \log_2 \frac{4}{\epsilon^2} = 2 \log_2 \frac{2}{\epsilon}.$$

Therefore, in this case the bound provided in Theorem 13.1 yields

$$N \geq \frac{4.1}{\epsilon}\left(\log \frac{21.64}{\delta} + 8.78 n_\theta \log_2 \frac{2}{\epsilon}\right).$$

We conclude that, if ϵ is sufficiently small, we obtain an explicit bound that depends only on the accuracy ϵ, confidence δ and on the number of controller parameters n_θ. More general results in which ϵ is not constrained within the open interval $(0, 0.14)$ can be derived, at the expense of obtaining larger constants appearing in the sample complexity provided in Theorem 13.1.

The following algorithm summarizes the developments of this section, and complies with the definition of randomized algorithm for reliable design given in Definition 10.6.

Algorithm 13.1 (Nonconvex scenario design) *Let J be a (α, m)-Boolean function. Given $\epsilon, \delta \in (0, 1)$, this RA is either unfeasible or it returns with probability at least $1 - \delta$ a design vector $\widehat{\boldsymbol{\theta}}_N \in \mathbb{R}^{n_\theta}$ such that*

$$V(\widehat{\boldsymbol{\theta}}_N) = \Pr\left\{J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_N) = 1\right\} \leq \epsilon.$$

1. Initialization.
   - ▷ Choose integer N satisfying (13.5);
2. Sample generation.
   - ▷ Generate N random iid samples $\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(N)}$;
3. Solve (13.2) with the given samples.
4. Return $\widehat{\boldsymbol{\theta}}_N$.

13.2 Sequential Algorithm for Nonconvex Scenario

Algorithm 13.1 for nonconvex scenario design is based upon the sample complexity bound (13.5). A more sophisticated sequential randomized algorithm for the solution of nonconvex semi-infinite feasibility and optimization problems has been proposed in [9] and is studied in this section. This algorithm is more closely related to the results on statistical learning theory presented in Chap. 9. We recall that other sequential algorithms based on statistical learning theory have been studied in [241] for finite families of controllers.

We consider a family of measurable binary-valued functions $\mathcal{J}$ having finite VC dimension $\mathrm{VC}(\mathcal{J}) \leq d < \infty$. For given level $\beta \in (0, 1)$, we consider the one-sided constraint failure defined in (9.6), and apply the sample complexity results of Theorem 9.5. Then, we provide a strategy that allows one to circumvent the potential conservativeness of this result for small values of the probabilistic levels ϵ and δ. This objective is accomplished by means of a sequence of optimization problems of increasing complexity, where the number of iterations is bounded by a termination parameter $\bar{\kappa}$. As it can be seen in the algorithm below, at each iteration, two sets of samples of cardinality N_k and M_k are generated. The first set, consisting of random samples $\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(N_k)}$, is used to obtain a candidate probabilistic solution $\widehat{\boldsymbol{\theta}}_{N_k}$ to the nonconvex optimization problem. The performance of this candidate solution is then tested using the random validation samples $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(M_k)}$. The cardinality of these sets grows at each iteration k as

$$N_k = \left\lceil \tau_{\Delta}^{k} \left(\frac{\beta + \epsilon}{\epsilon^2} \right) \right\rceil; \tag{13.6}$$

$$M_k = \left\lceil 2\tau_v^k \left(\frac{\beta + \epsilon}{\epsilon^2} \right) \log \frac{2\bar{\kappa}}{\delta} \right\rceil \tag{13.7}$$

where the constants τ_{Δ} and τ_v are chosen such that

$$N_{\bar{\kappa}} = \overline{N}; \qquad M_{\bar{\kappa}} = \max \left\{ \overline{N}, \left\lceil 2 \left(\frac{\beta + \epsilon}{\epsilon^2} \right) \log \frac{2\bar{\kappa}}{\delta} \right\rceil \right\}$$

where $\overline{N}$ is the sample size bound (9.6) provided by Theorem 9.5.

The parameter $\bar{\kappa}$ limits the maximal number of iterations of the algorithm and allows the user to control the sample size of the first iteration of the algorithm. In particular, $\bar{\kappa}$ may be chosen in the interval $[\bar{\kappa}^-, \bar{\kappa}^+]$ where

$$\bar{\kappa}^- = \left\lceil \frac{\log(\overline{N}_t \epsilon^2) - \log(\beta + \epsilon)}{\log 2} \right\rceil \quad \text{and} \quad \bar{\kappa}^+ = 2\bar{\kappa}^-.$$

With this choice, if $\bar{\kappa}$ is equal to $\bar{\kappa}^-$, then τ_{Δ} is close to 2, N_k approximately doubles at each iteration and the sample size of the first iteration is very close to $\frac{\beta+\epsilon}{\epsilon^2}$. The main idea of the algorithm is that if the candidate solution $\widehat{\theta}_{N_k}$ satisfies the inequality

$$\frac{1}{M_k} \sum_{i=1}^{M_k} J\left(\mathbf{v}^{(i)}, \widehat{\theta}_{N_k}\right) \leq \beta + \left(1 - \tau_v^{-\frac{k}{2}}\right)\epsilon$$

at iteration k, then it can be classified as a probabilistic solution with confidence δ, accuracy ϵ, level β as in Theorem 9.5, and no further iterations of the algorithm

are needed. This means that the algorithm may find a solution using a number of samples much smaller than $\overline{N}_t$.

Algorithm 13.2 (Sequential RA for nonconvex scenario) *Given* $\epsilon, \delta \in (0,1)$ *and* $\beta \in [0,1)$, *then*, *with probability at most* δ, *this RA returns a design vector* $\widehat{\boldsymbol{\theta}}_N \in \mathbb{R}^{n_\theta}$ *such that*

$$\widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_N)\big) \le \beta \quad \textit{and} \quad \mathrm{E}\big(J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_N)\big) - \widehat{\mathbf{E}}_N\big(J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_N)\big) > \epsilon. \tag{13.8}$$

1. Initialization.
   - ▹ Choose an integer $\bar{\kappa} \ge 1$, set $k = 0$ and
   $$\overline{N} = \left\lceil \frac{5(\beta+\epsilon)}{\epsilon^2}\left(\log\frac{8}{\delta} + d\log\frac{40(\beta+\epsilon)}{\epsilon^2}\right)\right\rceil;$$
   $$\tau_\Delta = \left(\frac{\overline{N}\epsilon^2}{\beta+\epsilon}\right)^{\frac{1}{\bar{\kappa}}};$$
   $$\tau_v = \max\left\{1, \tau_\Delta\left(2\log\frac{2\bar{\kappa}}{\delta}\right)^{-\frac{1}{\bar{\kappa}}}\right\};$$
2. Iteration.
   - ▹ If $k \ge \bar{\kappa}$ then Exit;
   - ▹ Else, set $k = k+1$ and $N_k = \lceil \tau_\Delta^k (\frac{\beta+\epsilon}{\epsilon^2}) \rceil$;
3. Nonconvex scenario with violated constraints.
   - ▹ Draw N_k iid samples $\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(N_k)}$ according to the pdf $f_{\boldsymbol{\Delta}}$;
   - ▹ Compute (if possible) a local minimum $\widehat{\theta}_{N_k}$ of the problem
   $$\begin{aligned} &\min_\theta \ c(\theta) \\ &\text{subject to } \frac{1}{N_k}\sum_{i=1}^{N_k} J\big(\boldsymbol{\Delta}^{(i)}, \theta\big) \le \beta; \end{aligned} \tag{13.9}$$
   - ▹ If $k = \bar{\kappa}$, then Exit,
4. Validation.
   - ▹ Draw
   $$M_k = \left\lceil 2\tau_v^k\left(\frac{\beta+\epsilon}{\epsilon^2}\right)\log\frac{2\bar{\kappa}}{\delta}\right\rceil$$
   iid validation samples $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(M_k)}$ according to the pdf $f_{\mathbf{v}}$;
   - ▹ If
   $$\frac{1}{M_k}\sum_{i=1}^{M_k} J\big(\mathbf{v}^{(i)}, \widehat{\theta}_{N_k}\big) \le \beta + \big(1 - \tau_v^{-\frac{k}{2}}\big)\epsilon$$
   then Exit;
   - ▹ Else, Goto 2;

We remark that the design vector $\widehat{\boldsymbol{\theta}}_N$ provided by this algorithm guarantees that the probability of one-sided constrained failure studied in Theorem 9.5 is at most δ, see further remarks in Sect. 9.3.

Remark 13.2 (Feasibility) As discussed in [9], Algorithm 13.2 guarantees that, if at iteration k the feasibility step is satisfied for a design vector $\widehat{\boldsymbol{\theta}}_{N_k}$, then with probability no smaller than $1-\delta$

$$V(\widehat{\boldsymbol{\theta}}_N) = \Pr\big\{J(\boldsymbol{\Delta}, \widehat{\boldsymbol{\theta}}_N) = 1\big\} \leq \beta + \epsilon.$$

On the other hand, if the algorithm terminates without providing a feasible solution, then, with probability no smaller than $1-\delta$, there is no $\theta \in \mathbb{R}^{n_\theta}$ such that

$$V(\theta) = \Pr\big\{J(\boldsymbol{\Delta}, \theta) = 1\big\} \leq \beta + \sqrt{\frac{1}{2\overline{N}_t} \log \frac{2}{\delta}}.$$

The proof of these statements is given in [9].

Remark 13.3 (Optimization with violated constraints) Note that, in step 3 of Algorithm 13.2, one has to find a solution θ such that (13.9) holds. Since $J(\Delta, \theta)$ is a binary function, this corresponds to requiring that the solution θ satisfies $J(\boldsymbol{\Delta}^{(i)}, \theta) = 0$ holds for all indices $i = 1, \ldots, N_k$, *except for a subset* of cardinality $\lceil \beta N_k \rceil$. This is in full analogy with the techniques discussed in Sect. 12.3 for the convex case: the optimization problem in step 3 is indeed a *nonconvex scenario problem with violated constraints*, where $r = \lceil \beta N_k \rceil$ constraints of the original problem are violated on purpose. Hence, the techniques for optimal (or suboptimal) constraint removal can be adapted to this setup. For a direct comparison of the VC bounds discussed in this section and the scenario-based bounds, the reader is referred to Sect. 7 of [78].

Example 13.4 (Stabilization of the lateral motion of an aircraft) We revisit Example 11.1 regarding LQR design of the lateral motion of an aircraft. In this case, we consider nine aircraft uncertain parameters entering into the state matrix A, while the input matrix B_2 is fixed. More precisely, these parameters coincide with those previously considered: $L_p = -2.93$, $L_\beta = -4.75$, $L_r = 0.78$, $g/V = 0.086$, $Y_\beta = -0.11$, $N_{\dot{\beta}} = 0.1$, $N_p = -0.042$, $N_\beta = 2.601$ and $N_r = -0.29$. Each nominal parameter is perturbed by a relative uncertainty equal to 15 %. That is, we consider the uncertain system

$$\dot{x} = A(q)x + B_2 u$$

where

$$q = [q_1 \quad \cdots \quad q_9]^T = [L_p \quad L_\beta \quad L_r \quad g/V \quad Y_\beta \quad N_{\dot{\beta}} \quad N_p \quad N_\beta \quad N_r]^T.$$

The objective is to design a state feedback $u = Kx$ such that the closed-loop system $\dot{x} = (A(q) + B_2 K)x$ has all its eigenvalues in a specified region of the complex

plane for all $q \in \mathcal{B}_q$. Moreover, as in [21], we assume a constraint on the magnitude of the entries of the gain matrix K. That is, we take

$$-\overline{K}_{i\ell} \leq K_{i\ell} \leq \overline{K}_{i\ell}$$

for $i = 1, 2$ and $\ell = 1, 2, 3, 4$, where the matrix $\overline{K}$ is given by

$$\overline{K} = \begin{bmatrix} 5 & 0.5 & 5 & 5 \\ 5 & 2 & 20 & 1 \end{bmatrix}.$$

In particular, we look for the optimal value of K that maximizes the convergence rate η to the origin. The decision variable θ consists of K and η, i.e. $\theta = (K, \eta)$ and $n_\theta = 9$, and we aim at solving the optimization problem

$$\begin{aligned} &\min_{\eta, K}(-\eta) \\ &\quad \text{subject to } A(q) + B_2 K + \eta I_4 \text{ is Hurwitz for all } q \in \mathcal{B}_q; \qquad (13.10) \\ &\qquad -\overline{K}_{i\ell} \leq K_{i\ell} \leq \overline{K}_{i\ell}, \quad i = 1, 2;\ \ell = 1, 2, 3, 4. \qquad (13.11) \end{aligned}$$

We notice that checking the constraint (13.10) can be performed using the classical Hurwitz test. Since the dimension of the state matrix is equal to four, this requires testing strict positivity of four Hurwitz determinants $H_i(q, \theta)$, $i = 1, \ldots, 4$ of the closed-loop system $A(q) + B_2 K + \eta I_4$. That is, (13.10) is equivalent to

$$\mathcal{H}_H(q, \theta) = \{H_1(q, \theta) > 0\} \wedge \cdots \wedge \{H_4(q, \theta) > 0\} \quad \text{is positive for all } q \in \mathcal{B}_q$$

where the symbol $\wedge$ denotes the logic operator "and". We observe that each determinant $H_i(q, \theta)$ is a polynomial in θ whose degree is at most $\alpha_i = i(i+1)/2$, see the computations given in [404]. Hence, $\mathcal{H}_H(q, \theta)$ is an $(\alpha, 4)$-Boolean function, with

$$\alpha = \max_{i=1,\ldots,4} \alpha_i = 4(4+1)/2 = 10.$$

Additionally, the constraint (13.11) is rewritten as the Boolean condition

$$\begin{aligned} &\mathcal{H}_K(\theta) \\ &\quad = \{K_{1,1} \geq -\overline{K}_{1,1}\} \wedge \{K_{1,1} \leq \overline{K}_{1,1}\} \wedge \cdots \wedge \{K_{2,4} \geq -\overline{K}_{2,4}\} \wedge \{K_{2,4} \leq \overline{K}_{2,4}\} \end{aligned}$$

which is a $(1, 16)$-Boolean function. Hence, the binary function

$$J(q, \theta) = \begin{cases} 0 & \text{if } \mathcal{H}_H(q, \theta) \text{ is positive and } \mathcal{H}_K(\theta) \text{ is satisfied;} \\ 1 & \text{otherwise} \end{cases}$$

is a $(10, 20)$-Boolean function. A bound on the VC dimension can be immediately computed by means of Lemma 9.8 obtaining that the VC dimension (α, m)-Boolean functions is bounded by $2n_\theta \log_2(4e\alpha m) \leq 200$.[1]

[1] We recall that bounds on the VC dimension have been computed for various control problems. For example, in [408], the VC dimension for static output feedback is given by $2n_i n_o \log_2[2en_s^2(n_s + 1)]$.

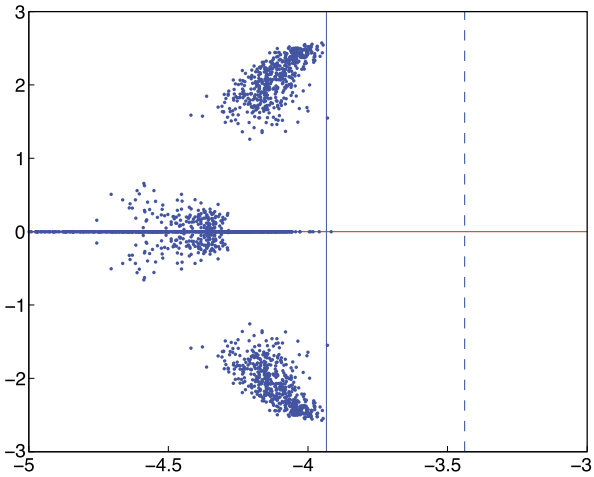

Fig. 13.1 Closed-loop eigenvalues for $\epsilon = 0.01$, $\delta = 10^{-6}$ and $\beta = 0$

In this example, we assume that $\mathbf{q}$ is a random vector with uniform pdf within the set $\mathcal{B}_q$. Then, we set confidence δ, accuracy ϵ and level β to 10^{-6}, 10^{-2} and 0 respectively. In this case, using Theorem 9.5 we obtain a sample size $N \geq 835,176$; since β is equal to zero, and we are studying Boolean binary functions, we also compute the sample size bound $N \geq 310,739$ using Theorem 13.1. For the sake of comparison, the sample size given by Theorem 9.3 is $N \geq 8.75 \times 10^8$.

To circumvent the conservativeness of these bounds, we run Algorithm 13.2 with the exiting parameter $\bar{\kappa}$ equal to 20. The algorithm provided a probabilistic solution at iteration 5. The sample sizes N_k and M_k at the last iteration were equal to 957 and 13,761, respectively. The obtained controller is given by

$$K = \begin{bmatrix} 1.9043 & 0.5000 & -5.0000 & 2.8951 \\ 5.0000 & 1.5080 & 4.4829 & -1.0000 \end{bmatrix}$$

and the corresponding value for γ is 3.93. Therefore, we conclude that, with confidence $1-\delta$, for the obtained controller 99 % of the uncertain plants (notice that $1-\beta-\epsilon = 0.99$) have a rate of convergence greater or equal to 3.93. We notice that some of the controller gains are equal to the entries of the bounding matrix $\overline{K}$, which shows that the constraints of the problem take effect into the obtained solution.

Once the controller was computed, we formulated a (deterministic) sufficient condition for quadratic stability, obtaining a worst-case value for the convergence rate γ. In particular, we obtained the generalized eigenvalue problem

$$\max_{\gamma, P \succ 0} \ \gamma$$

$$\text{subject to } P\big(A(q) + \gamma \mathrm{I} + BK\big) + \big(A(q) + \gamma \mathrm{I} + BK\big)^T P \prec 0 \quad \text{for all } q \in \mathcal{B}_q.$$

Since q enters affinely in the matrix $A(q)$, we reformulate this as a generalized eigenvalue problem subject to 512 constraints, i.e. the number of vertices of the

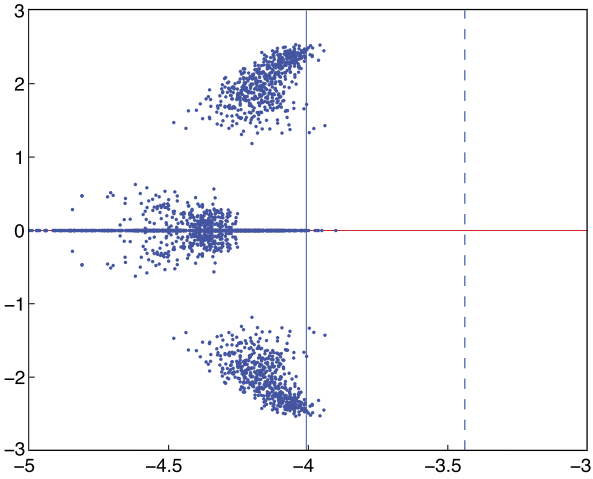

Fig. 13.2 Closed-loop eigenvalues for $\epsilon = 0.01$, $\delta = 10^{-6}$ and $\beta = 0.05$

hyperrectangle $\mathcal{B}_q$ (this number of constraints can be reduced using the results of [11, 90]). Using an LMI solver, the matrix

$$P = \begin{bmatrix} 0.1164 & 0.0202 & -0.0822 & 0.0297 \\ 0.0202 & 0.0048 & 0.0105 & 0.0000 \\ -0.0822 & 0.0105 & 0.9472 & -0.2042 \\ 0.0297 & 0.0000 & -0.2042 & 0.0475 \end{bmatrix}$$

and the value $\gamma = 3.44$ have been computed. This guarantees that the controller K derived with the proposed randomized strategy provides a closed-loop uncertain system which is stable for all $q \in \mathcal{B}_q$. Figure 13.1 shows the eigenvalues of the closed-loop system for 500 elements randomly drawn from $\mathcal{B}_q$.

We also run the algorithm for $\epsilon = 0.01$, $\delta = 10^{-6}$, $\beta = 0.05$ and $\bar{\kappa} = 20$. In this case we obtained a controller that, with probability no smaller than $1 - \delta$, guarantees that 94 % of the uncertain plants (notice that $1 - \beta - \epsilon = 0.94$) have a rate of convergence greater or equal to 4.0. The deterministic sufficient condition provided a worst-case rate of convergence equal to 3.43. Figure 13.2 shows the closed-loop eigenvalues of the system for 500 elements randomly drawn from $\mathcal{B}_q$.

Chapter 14
Random Number and Variate Generation

In this chapter, we discuss various methods for the generation of random samples distributed according to given probability distributions, in both the univariate and multivariate cases. These methods can be traced back to the issue of generating uniform random numbers in the interval [0, 1]. This problem is analyzed in Sect. 14.1, where a summary of the main existing techniques as well as more recent algorithms are reported. Subsequently, we study the problem of univariate random generation. In particular, we present some standard results regarding transformations between random variables and show specific examples for various classical distributions. The second part of the chapter describes techniques for multivariate distributions, focusing in particular on rejection methods, on the recursive conditional densities method, and on asymptotic methods based on Markov chains.

14.1 Random Number Generators

The importance of random numbers in Monte Carlo methods has been discussed in Chap. 7. Good random number generators (RNGs) should provide uniform and independent samples, and should be reproducible and fast. Computer methods for random generation, such as the classical one based on the method of von Neumann [409] for simulating neutron transport, produce only *pseudo-random* sequences, which show cyclicities and correlations. Indeed, RNGs are deterministic algorithms that provide numbers with certain statistical properties. Roughly speaking, these numbers should behave similar to realizations of independent, identically distributed uniform random variables. However, every RNG has its deficiencies, so that no RNG is appropriate for all purposes. For example, several "good" RNGs for stochastic simulation are unsuitable for cryptographic applications, because they produce predictable output streams.

RNGs mainly consist of linear and nonlinear generators. The linear algorithms are well-known and widely available. However, linear RNGs may sometimes be inadequate, since these algorithms may produce lattice structures in every dimension, as shown in Sect. 14.1.1, and this fact may interfere with the simulation

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_14,

problem at hand. For this reason, nonlinear generators have been introduced. In general, these latter methods are computationally slower than linear generators of comparable size, but they allow for the use of larger strings of samples, see for instance [197].

Random number generation constitutes a whole field of study in its own. As a starting point, the reader interested in further understanding these topics may consult the classical reference [238], as well as the special issue [113], the survey paper [259] and the edited volume [198]. However, even though this is a well-established topic, current research is performed with the objective to produce extremely fast and reliable algorithms for various applications. An example of a recent and extremely efficient RNG is the so-called *Mersenne twister* (MT) algorithm [279].

We now describe linear congruential generators (LCGs), which are among the earliest methods for random number generation.

14.1.1 Linear Congruential Generators

Linear congruential generators for uniform distribution in the interval $[0, 1)$, have been proposed by Lehmer [261], and are based on a recursion of the form

$$x^{(i+1)} = ax^{(i)} + c - mk^{(i)}, \quad i = 0, 1, \ldots$$

where the *multiplier* a, the *increment* c and the *modulus* m are nonnegative integers, and $k^{(i)}$ is given by

$$k^{(i)} = \left\lfloor \frac{ax^{(i)} + c}{m} \right\rfloor.$$

This recursion is often written using the notation

$$x^{(i+1)} = \left(ax^{(i)} + c\right) \bmod m, \quad i = 0, 1, \ldots \tag{14.1}$$

This linear congruential generator is denoted as $\mathrm{LCG}(a, c, m, x^{(0)})$. The modulus m is chosen as a "large" positive integer. In particular, this value is generally set to the word length of the machine, for example $m = 2^{32}$. The multiplier $a \in \{1, \ldots, m\}$ is selected so that the greatest common divisor of (a, m) is one and $c \in \{0, 1, \ldots, m-1\}$. If the increment c is set to zero, the RNG is called "multiplicative congruential generator," otherwise it is called "mixed congruential generator."

Given an initial value $x^{(0)} \in \{0, 1, \ldots, m-1\}$, called the *seed*, one generates a sequence $x^{(0)}, x^{(1)}, \ldots, x^{(m-1)}$ according to the recursion (14.1); this sequence is generally called a Lehmer sequence. Notice that $x^{(i)} \in \{0, 1, \ldots, m-1\}$ for all i; therefore, numbers $y^{(i)}$ in the interval $[0, 1)$ can be subsequently obtained by taking

$$y^{(i)} = \frac{x^{(i)}}{m} \in [0, 1), \quad i = 0, 1, \ldots.$$

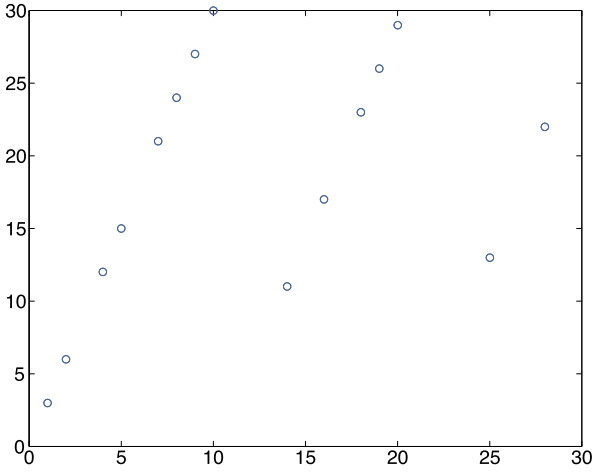

Fig. 14.1 Lattice structure of the two-dimensional vectors (14.3) obtained from points generated by LCG(3, 0, 31, 5)

We remark that the sequences $x^{(0)}, x^{(1)}, \ldots$ and $y^{(0)}, y^{(1)}, \ldots$ are both periodic with the same period, which is no greater than m. For example, taking $m = 31$, $a = 3$, $c = 0$ and $x^{(0)} = 5$, using (14.1), we obtain the sequence

$$5\ 15\ 14\ 11\ 2\ 6\ 18\ 23\ 7\ 21\ 1\ 3\ \ 9\ \ 27\ 19\ 26\ 16\ 17\ 20\ 29\ 25\ 13\ 8\ 24\ 10\ 30\ 28\ 22\ 4\ 12\ 5\ 15\ 14\ 11\ 2\ 6\ 18\ 23\ 7\ 21\ 1\ 3\ \cdots \tag{14.2}$$

It can be easily observed that the period in this case is 30.

A critical issue regarding LCGs is that multidimensional samples built using successive outcomes of the generator lie on a lattice. That is, if we consider the d-dimensional vectors $w^{(k)} = [x^{(k)}\ x^{(k+1)}\ \cdots\ x^{(k+d-1)}]^T$ for different values of k, and study the distribution of the points $w^{(k)}$ in $[0, 1)^d$, we observe that the generated points are of the form

$$w^{(k)} = \sum_{i=1}^{d} z_i^{(k)} v_i$$

where $\{z_1^{(k)}, \ldots, z_d^{(k)}\}$ are integers, and $\{v_1, \ldots, v_d\}$ is a set of linearly independent vectors $v_i \in \mathbb{R}^d$, $i = 1, \ldots, d$, which constitutes a basis of the lattice. For instance, in the LCG(3, 0, 31, 5) sequence (14.2), the couples obtained considering the nonoverlapping vectors

$$w^{(1)} = \begin{bmatrix} 5 \\ 15 \end{bmatrix}, \quad w^{(3)} = \begin{bmatrix} 14 \\ 11 \end{bmatrix}, \quad w^{(5)} = \begin{bmatrix} 2 \\ 6 \end{bmatrix}, \quad w^{(7)} = \begin{bmatrix} 18 \\ 23 \end{bmatrix}, \quad \ldots \tag{14.3}$$

lie on three lines, see Fig. 14.1. The IBM random generator RANDU, which was used for a number of years, is a striking example of a generator providing a lattice structure in the distribution. RANDU is LCG($2^{16} + 3, 0, 2^{31}, x^{(0)}$), see further discussions in [177, 197]. This discussion pinpoints some of the limits of LCGs and motivates the study of other more sophisticated generators.

14.1.2 Random Number Generators

We now briefly describe some standard linear and nonlinear generators. This description is at introductory level, and the interested reader may consult more specific references such as [177, 304].

Multiple Recursive Generators A simple generalization of the multiplicative congruential generator is given by

$$x^{(i+1)} = \left(a_1 x^{(i)} + a_2 x^{(i-1)} + \cdots + a_n x^{(i-n-1)}\right) \bmod m, \quad i = 0, 1, \ldots$$

where $a_1, a_2, \ldots, a_n$ are given multipliers. One of the advantages of this generator is to exhibit a longer period than the multiplicative congruential generator. Further statistical properties and specific recommendations regarding the selection of multipliers are given in [260].

Lagged Fibonacci Generators It is well known that a Fibonacci sequence is given by

$$x^{(i+1)} = x^{(i)} + x^{(i-1)}, \quad i = 0, 1, \ldots.$$

In principle, we can use a Fibonacci sequence as an RNG, with the simple modification

$$x^{(i+1)} = \left(x^{(i)} + x^{(i-1)}\right) \bmod m, \quad i = 0, 1, \ldots.$$

However, this generator does not have nice statistical properties. A simple way to modify and improve this algorithm is to introduce the so-called lagged Fibonacci congruential generator. That is, we introduce the terms $x^{(i-\ell)}$ and $x^{(i-k)}$, where $k > \ell$, obtaining

$$x^{(i+1)} = \left(x^{(i-\ell)} + x^{(i-k)}\right) \bmod m, \quad i = 0, 1, \ldots.$$

Clearly, in this sequence a set of seeds, rather than a single seed, needs to be specified. In [18] it is shown that if the initial sequence, ℓ, k and m are carefully selected, then this RNG "performs well." In particular, if m is a prime number and $k > \ell$, in this paper it is shown that the period of the sequence is at most $m^k - 1$.

Nonlinear Congruential Generators Knuth [238] suggested a simple generalization of the linear congruential generator proposing the nonlinear congruential generator

$$x^{(i+1)} = \left(d\left(x^{(i)}\right)^2 + a x^{(i)} + c\right) \bmod m, \quad i = 0, 1, \ldots.$$

In general, higher order polynomials could also be used, but the advantages of this further extension and the rules for selecting the order of the polynomial are unclear. A special case of this nonlinear congruential generator has been studied in [62, 63]

$$x^{(i+1)} = \left(d\left(x^{(i)}\right)^2\right) \bmod m, \quad i = 0, 1, \ldots$$

where m is the product of two "large" distinct prime numbers, and the output is the least significant bit of $x^{(i+1)}$, or its k least significant bits. This generator, known as the Blum Blum Shub (BBS) generator, has interesting theoretical properties that make it suitable for applications in cryptography.

Remark 14.1 (RNG for cryptography) It should be noted that not any generator is appropriate for cryptographic applications, as shown in [48, 166]. In particular, linear congruential generators are not suitable, since it is possible to recover their parameters in polynomial time, given a sufficiently long observation of the output stream. Unlike LCGs, the BBS generator has very strong cryptographic properties, which relate the quality of the generator to the difficulty of the "integer factorization problem;" that is, computing the prime factors of a very large integer. When the prime factors of m are chosen appropriately, and $O(\log\log m)$ bits of each $x^{(i)}$ are considered as the output of the BBS recursion, then, for large values of m, distinguishing the output bits from random numbers becomes at least as difficult as factoring m. Since integer factorization is largely believed not to be polynomial-time solvable, then BBS with large m has an output free from any nonrandom patterns that could be discovered with a reasonable amount of calculations.

On the other hand, BBS does not seem the preferred choice for stochastic simulations, since the required nonlinear operations cannot be performed with high computational efficiency. However, LCGs do have high computational efficiency, as do the shift register generators discussed in the following paragraph.

Feedback Shift Register Generators A generator that returns binary numbers $x^{(i)} \in \{0, 1\}$ is studied in [380]. This *binary generator* is of the form

$$x^{(i+1)} = \left(c_p x^{(i-p)} + c_{p-1} x^{(i-p+1)} + \cdots + c_1 x^{(i-1)}\right) \bmod 2, \quad i = 0, 1, \ldots$$

where all variables take binary values $\{0, 1\}$ and p is the order of the recursion. The name of this generator follows from the fact that recursive operations of this form can be performed in a feedback shift register. Further properties of the feedback shift register generators are discussed in [238].

The Mersenne twister generator [279] is itself a twisted generalized shift feedback register generator. The "twist" is a transformation which assures equidistribution of the generated numbers in 623 dimensions, while LCGs can at best manage reasonable distribution in five dimensions. MT was proved to have a period as large as $2^{19937} - 1$, which, incidentally, explains the origin of the name: the number $2^{19937} - 1$ is a Mersenne prime. Unlike BBS, the MT algorithm in its native form is not suitable for cryptography. For many other applications, such as stochastic simulation, however, it is becoming the random number generator of preferred choice.

14.2 Nonuniform Random Variables

In the previous section we introduced several standard methods for generating pseudo-random numbers, which can be considered uniformly distributed over the interval $[0, 1)$. Starting from these basic uniform generators, many different distributions can be obtained by means of suitable functional transformations or other operations.

In this section, we study general operations on random variables and analyze some well-known univariate random generation methods. These results can be used for constructing sample generators according to various distributions, as shown in the examples presented in this section. First, we present a fundamental result that indicates how the probability density is changed by a functional operation on a random variable, see e.g. [334] for proof.

Theorem 14.1 (Functions of scalar random variables) *Let* $\mathbf{x} \in \mathbb{R}$ *be a random variable with distribution function* $F_{\mathbf{x}}(x)$ *and pdf* $f_{\mathbf{x}}(x)$, *and let* $\mathbf{y}$ *be related to* $\mathbf{x}$ *by a strictly monotone and absolutely continuous transformation* $\mathbf{y} = g(\mathbf{x})$. *Let* $h(\cdot) \doteq g^{-1}(\cdot)$. *Then, the random variable* $\mathbf{y}$ *has distribution function*

$$F_{\mathbf{y}}(y) = \begin{cases} F_{\mathbf{x}}(h(y)) & \text{if } g(x) \text{ is increasing;} \\ 1 - F_{\mathbf{x}}(h(y)) & \text{if } g(x) \text{ is decreasing} \end{cases}$$

and density

$$f_{\mathbf{y}}(y) = f_{\mathbf{x}}\big(h(y)\big) \left| \frac{\mathrm{d}h(y)}{\mathrm{d}y} \right|$$

for almost all y.

The transformation rule of Theorem 14.1 also has a multivariate extension, which is stated in Sect. 14.3. Some standard applications of this result are presented in the following examples.

Example 14.1 (Linear transformation) The simpler transformation on a random variable $\mathbf{x}$ is the linear transformation $\mathbf{y} = a\mathbf{x} + b$, $a > 0$. If $\mathbf{x}$ has distribution function $F_{\mathbf{x}}(x)$ then

$$F_{\mathbf{y}}(y) = F_{\mathbf{x}}\left(\frac{y-b}{a}\right).$$

If the corresponding density $f_{\mathbf{x}}(x)$ exists, then we also have

$$f_{\mathbf{y}}(y) = \frac{1}{a} f_{\mathbf{x}}\left(\frac{y-b}{a}\right).$$

Example 14.2 (Linear transformation of the Gamma density) A density widely used in statistics is the unilateral Gamma density with parameters a, b, defined in (2.12) as

$$G_{a,b}(x) = \frac{1}{\Gamma(a) b^a} x^{a-1} \mathrm{e}^{-x/b}, \quad x \geq 0.$$

If $\mathbf{x}$ is distributed according to $G_{a,b}$, then the random variable $\mathbf{y} = c\mathbf{x}$, $c > 0$, obtained by linear transformation, is distributed according to $G_{a,cb}$.

Example 14.3 (Power transformation) If a random variable $\mathbf{x} \geq 0$ has distribution function $F_{\mathbf{x}}(x)$ and density $f_{\mathbf{x}}(x)$, then the variable $\mathbf{y} = \mathbf{x}^{\lambda}$, $\lambda > 0$, has distribution

$$F_{\mathbf{y}}(y) = F_{\mathbf{x}}\big(y^{1/\lambda}\big)$$

and density

$$f_{\mathbf{y}}(y) = \frac{1}{\lambda} y^{\frac{1}{\lambda}-1} f_{\mathbf{x}}\big(y^{1/\lambda}\big).$$

Example 14.4 (Generalized Gamma density via power transformation) The (unilateral) generalized Gamma density with parameters a, c, see (2.13), is given by

$$\overline{G}_{a,c}(x) = \frac{c}{\Gamma(a)} x^{ca-1} \mathrm{e}^{-x^c}, \quad x \geq 0.$$

If $\mathbf{x} \sim G_{a,1}$, then the random variable $\mathbf{y} = \mathbf{x}^{1/c}$ obtained by power transformation has density function $f_{\mathbf{y}}(y) = \overline{G}_{a,c}(y)$.

Example 14.5 (Weibull density via power transformation) A random variable with Weibull density (2.10)

$$W_a(x) = a x^{a-1} \mathrm{e}^{-x^a}$$

can be obtained from a random variable distributed according to an exponential density via power transformation. In fact, if $\mathbf{x} \sim \mathrm{e}^{-x}$, $x \geq 0$, then $\mathbf{y} = \mathbf{x}^{1/a}$, $a > 0$, has density $f_{\mathbf{y}}(y) = W_a(y)$.

Example 14.6 (Logarithmic transformation) If a random variable $\mathbf{x} \geq 0$ has distribution function $F_{\mathbf{x}}(x)$ and density $f_{\mathbf{x}}(x)$, then the variable $\mathbf{y} = -\frac{1}{\lambda} \log \mathbf{x}$, $\lambda > 0$, has distribution

$$F_{\mathbf{y}}(y) = 1 - F_{\mathbf{x}}\big(\mathrm{e}^{-\lambda y}\big)$$

and density

$$f_{\mathbf{y}}(y) = \lambda \mathrm{e}^{-\lambda y} f_{\mathbf{x}}\big(\mathrm{e}^{-\lambda y}\big).$$

For instance, if $\mathbf{x}$ is uniform in $[0, 1]$, then $\mathbf{y} = -\frac{1}{\lambda} \log \mathbf{x}$ has the unilateral Laplace (exponential) density (2.11), i.e.

$$f_{\mathbf{y}}(y) = \lambda \mathrm{e}^{-\lambda y}, \quad y \geq 0.$$

A useful consequence of Theorem 14.1 is a standard method for generating a univariate random variable with a given distribution function. This method is known as the *inversion method*, see e.g. [133, 319], and is stated next.

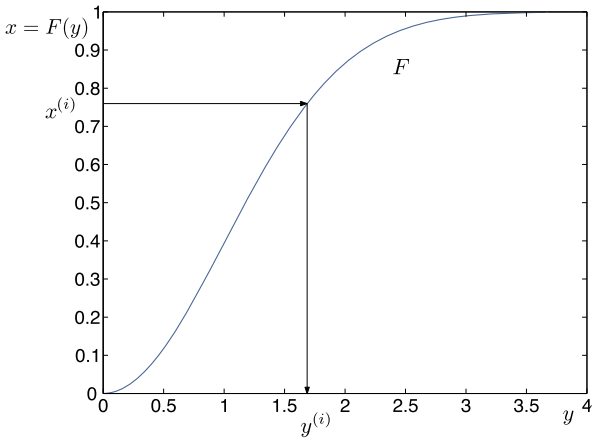

Fig. 14.2 Graphical interpretation of the inversion method

Corollary 14.1 (Inversion method) *Let* $\mathbf{x} \in \mathbb{R}$ *be a random variable with uniform distribution in the interval* $[0, 1]$. *Let* F *be a continuous distribution function on* $\mathbb{R}$ *with inverse* F^{-1} *defined by*

$$F^{-1}(x) = \inf\{y : F(y) = x,\ 0 \le x \le 1\}.$$

Then, the random variable $\mathbf{y} = F^{-1}(\mathbf{x})$ *has distribution function* F. *Also, if a random variable* $\mathbf{y}$ *has distribution function* F, *then the random variable* $\mathbf{x} = F(\mathbf{y})$ *is uniformly distributed on* $[0, 1]$.

Proof The statement immediately follows from Theorem 14.1, taking $y = g(x) \doteq F^{-1}(x)$, i.e. $x = h(y) \doteq F(y)$, and noticing that for the uniform distribution $F(h(y)) = h(y)$. □

A plot showing the idea behind the inversion method is shown in Fig. 14.2: uniform samples of $\mathbf{x}$ in the vertical axis are mapped into samples of $\mathbf{y}$ having distribution F.

Corollary 14.1 can be used to generate samples of a univariate random variables with an arbitrary continuous distribution function, provided that its inverse is explicitly known, or readily computable. Clearly, the numerical efficiency of the method relies on how fast the inverse can be numerically computed. Implementation refinements of the above method are discussed for instance in [133]. An application of the inversion method for generation of samples according to a polynomial density is presented next.

Example 14.7 (Generation from a polynomial density) The inversion method is useful for the generation of samples according to a generic univariate polynomial density over the interval $[0, c]$. Let

$$f_{\mathbf{y}}(y) = p(y) = \sum_{k=0}^{n} a_k y^k$$

be a polynomial density with support $[0, c]$. Notice that the distribution function of $\mathbf{y}$ can be easily computed as

$$F_{\mathbf{y}}(y) = \int_0^y p(y)\,\mathrm{d}y = \sum_{k=0}^{n} \frac{a_k}{k+1} y^{k+1}.$$

The condition that the polynomial $p(y)$ is a density function requires that $p(y) \geq 0$ for all $y \in [0, c]$, and that

$$\sum_{k=0}^{n} \frac{a_k}{k+1} c^{k+1} = 1.$$

A simple algorithm can hence be used for polynomial sample generation.

Algorithm 14.1 (Generation from a polynomial density) *This algorithm returns a random variable* $\mathbf{y}$ *distributed according to the polynomial density* $f_{\mathbf{y}}(y) = \sum_{k=0}^{n} a_k y^k$ *with support* $[0, c]$.

1. Generate a random variable $\mathbf{x} \sim \mathcal{U}_{[0,1]}$;
2. Compute the unique root $\mathbf{y}$ in $[0, c]$ of the polynomial
$$\sum_{k=0}^{n} \frac{a_k}{k+1} y^{k+1} - \mathbf{x} = 0;$$
3. Return $\mathbf{y}$.

In step 2, the numerical computation of the root can be performed, up to a given accuracy, using some standard method such as bisection or Newton–Raphson. We also remark that more efficient methods for generating samples from polynomial densities exist, see for instance the method in [8], based on finite mixtures.

In the next section, we briefly discuss two classical tests, the chi-square and the Kolmogorov–Smirnov (KS) test, that are used in statistics to assess whether a given batch of sample data comes from a specific distribution, see e.g. [238].

14.2.1 Statistical Tests for Pseudo-Random Numbers

Chi-Square Test The chi-square goodness-of-fit test, see e.g. [360], is used to decide if a batch of sampled data comes from a specific distribution $F^0(x)$. An attractive feature of this test is that it can be applied to any univariate distribution (continuous or discrete) for which the cumulative distribution function can be calculated. The chi-square test is applied to binned data, but this is actually not a restriction, since for nonbinned data one can simply calculate a histogram or frequency table

before applying the test. However, the value of the chi-square test statistic depends on how the data is binned. Another disadvantage is that this test requires sufficiently large sample size for its approximations to be valid. This method is briefly described below.

Let $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ be a multisample of size N drawn from an unknown cdf $F_{\mathbf{x}}(x)$, and let $F^0(x)$ be a completely specified candidate cdf. We wish to test the null hypothesis

$$H_0 : F_{\mathbf{x}}(x) = F^0(x), \quad \text{for all } x$$

against the alternative

$$H_1 : \exists x : F_{\mathbf{x}}(x) \neq F^0(x).$$

In the chi-square test, the data is divided into n bins, i.e. into n intervals $[x_i^-, x_i^+]$. Then, we introduce the sum known as *Pearson's test statistic*

$$\widehat{\mathbf{y}} = \sum_{i=1}^{n} \frac{(\widehat{\mathbf{p}}_i - p_i)^2}{p_i} \tag{14.4}$$

where $\widehat{\mathbf{p}}_i$ and p_i are the empirical probability and the true probability associated with the ith bin. In particular, p_i is given by

$$p_i = F^0\big(x_i^+\big) - F^0\big(x_i^-\big).$$

Clearly, the value of $\widehat{\mathbf{y}}$ tends to be small when H_0 is true, and large when H_0 is false. For large sample size, the distribution of $\widehat{\mathbf{y}}$ is approximately chi-square with $n-1$ degrees of freedom, see (2.9). Therefore, under the H_0 hypothesis, we expect

$$\text{PR}\{\widehat{\mathbf{y}} > x_{1-\alpha}\} = \alpha$$

where $\alpha \in (0, 1)$ is the significance level, and $x_{1-\alpha}$ is the $1-\alpha$ percentile of the chi-square distribution.

The chi-square test then goes as follows: given samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$, and a candidate cdf $F^0(x)$, compute $\widehat{\mathbf{y}}$ from (14.4). Select a significance level $\alpha \in (0, 1)$ and compute the chi-square percentile[1] $x_{1-\alpha}$. If $\widehat{\mathbf{y}} \leq x_{1-\alpha}$, then the test is passed, and the cdf F^0 is a good fit for the data distribution.

Kolmogorov–Smirnov Test The Kolmogorov–Smirnov test, see e.g. [319], is an alternative to the chi-square goodness-of-fit test. Given a multisample $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ of size N drawn from an unknown cdf $F_{\mathbf{x}}(x)$, the empirical distribution function is defined as

$$\widehat{\mathbf{F}}_N(x) \doteq \frac{\widehat{\mathbf{k}}(x)}{N}$$

where $\widehat{\mathbf{k}}(x)$ is the number of sample points $\mathbf{x}^{(i)}$ which are smaller than x. This is a step function that increases by $1/N$ at the value of each data point. The KS test is

[1] Chi-square percentile tables are available in standard statistics books, e.g. [319].

based on the maximum distance between the empirical distribution and a candidate distribution $F^0(x)$. Formally, the random quantity

$$\widehat{\mathbf{y}} = \sup_x \left| \widehat{\mathbf{F}}_N(x) - F^0(x) \right| \tag{14.5}$$

measures how far $\widehat{\mathbf{F}}_N(x)$ deviates from $F^0(x)$, and is called the Kolmogorov–Smirnov one-sample statistic. For large sample size N, it holds that

$$\Pr\left\{ \sqrt{N}\, \widehat{\mathbf{y}} \leq x \right\} \simeq H(x) \tag{14.6}$$

where

$$H(x) = 1 - 2 \sum_{k=1}^{\infty} (-1)^{k-1} \mathrm{e}^{-2k^2 x^2}.$$

The function $H(x)$ is tabulated, and the approximation (14.6) is practically good for $N > 35$.

The KS test goes as follows: given samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$, and a candidate cdf $F^0(x)$, compute $\widehat{\mathbf{y}}$ using equation (14.5). Select a significance level $\alpha \in (0, 1)$ and compute the $1 - \alpha$ percentile of $H(x)$. If $\sqrt{N}\, \widehat{\mathbf{y}} \leq x_{1-\alpha}$, then the test is passed, and the cdf $F^0(x)$ is a good fit for the empirical distribution. In the case when $F^0(x)$ is close to the true underlying cdf $F_{\mathbf{x}}(x)$, the probability of failing the test is smaller than α.

An attractive feature of this test is that the distribution of the KS test statistic $\widehat{\mathbf{y}}$ does not depend on the underlying cdf being tested. Another advantage is that it is an exact test. Despite these advantages, the KS test has several limitations: (1) it only applies to continuous distributions; (2) it tends to be more sensitive near the center of the distribution than at the tails; (3) the distribution should be fully specified. Owing to these limitations, many analysts prefer to use other, more sophisticated tests, such as the Anderson–Darling goodness-of-fit test [23].

14.3 Methods for Multivariate Random Generation

In this section, we discuss some standard methods for generating random samples from multivariate densities. In particular, we discuss rejection-based methods, and the method based on conditional densities.

We first present a multivariate extension of Theorem 14.1, see for instance [133]. More precisely, the following theorem can be used for obtaining a random vector $\mathbf{y} \in \mathbb{R}^n$ with desired density $f_{\mathbf{y}}(y)$, starting from a random vector $\mathbf{x} \in \mathbb{R}^n$ with pdf $f_{\mathbf{x}}(x)$. This tool is based on the functional transformation $\mathbf{y} = g(\mathbf{x})$ and it can be

used whenever the inverse $\mathbf{x} = h(\mathbf{y}) \doteq g^{-1}(\mathbf{y})$ exists. The transformation rule makes use of the Jacobian of the function $x = h(y)$, defined as

$$J(x \to y) \doteq \begin{vmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_2}{\partial y_1} & \cdots & \frac{\partial x_n}{\partial y_1} \\ \frac{\partial x_1}{\partial y_2} & \frac{\partial x_2}{\partial y_2} & \cdots & \frac{\partial x_n}{\partial y_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial x_1}{\partial y_n} & \frac{\partial x_2}{\partial y_n} & \cdots & \frac{\partial x_n}{\partial y_n} \end{vmatrix} \tag{14.7}$$

where $\frac{\partial x_i}{\partial y_\ell} \doteq \frac{\partial h_i(y)}{\partial y_\ell}$ are the partial derivatives, and $|\cdot|$ denotes the absolute value of the determinant. The notation $J(x \to y)$ means that the Jacobian is computed taking the derivatives of x with respect to y. It also helps recalling that the Jacobian $J(x \to y)$ is used when determining the pdf of $\mathbf{y}$ given the pdf of $\mathbf{x}$, see Theorem 14.2 Various rules for computation of Jacobians are given in Appendix A.2.

Theorem 14.2 (Functions of random vectors) *Let $\mathbf{x} \in \mathbb{R}^n$ be a random vector with density $f_{\mathbf{x}}(x_1, \ldots, x_n)$ continuous on the support $\mathcal{Y}_x \subseteq \mathbb{R}^n$, and let $\mathbf{y} = g(\mathbf{x})$, where $g : \mathcal{Y}_x \to \mathcal{Y}_y$, $\mathcal{Y}_y \subseteq \mathbb{R}^n$, is a one-to-one and onto mapping, so that the inverse $\mathbf{x} = h(\mathbf{y}) \doteq g^{-1}(\mathbf{y})$ is well-defined. Then, if the partial derivatives $\frac{\partial x_i}{\partial y_\ell} \doteq \frac{\partial h_i(y)}{\partial y_\ell}$ exist and are continuous on $\mathcal{Y}_y$, the random vector $\mathbf{y}$ has density*

$$f_{\mathbf{y}}(y) = f_{\mathbf{x}}\big(h(y)\big) J(x \to y), \quad y \in \mathcal{Y}_y. \tag{14.8}$$

An extension of this theorem to transformations between random matrices is presented in Appendix A.1. A classical example of application of this result is the Box–Muller method for generating normal samples in $\mathbb{R}^2$, starting from uniform samples.

Example 14.8 (Box–Muller method for normal densities in $\mathbb{R}^2$) Consider the transformation $y = g(x)$, defined by

$$\begin{aligned} y_1 &= \sqrt{-2 \log x_1} \cos(2\pi x_2); \\ y_2 &= \sqrt{-2 \log x_1} \sin(2\pi x_2) \end{aligned}$$

mapping $x \in \mathcal{Y}_x = [0, 1]^2$ into $y \in \mathcal{Y}_y = \mathbb{R}^2$. Then, if $\mathbf{x}$ is uniform in $\mathcal{Y}_x$, the transformed random variables $\mathbf{y}_1, \mathbf{y}_2$ have independent normal densities. This fact is easily verified using Theorem 14.2. Indeed, the inverse mapping $x = h(y)$ is

$$\begin{aligned} x_1 &= \mathrm{e}^{-\frac{R^2}{2}}; \\ x_2 &= \frac{1}{2\pi} \arcsin \frac{y_2}{R}, \quad R \doteq \sqrt{y_1^2 + y_2^2}. \end{aligned}$$

The Jacobian of this transformation is given by

$$J(x \to y) = \begin{vmatrix} -y_1 \mathrm{e}^{-\frac{R^2}{2}} & -y_2 \mathrm{e}^{-\frac{R^2}{2}} \\ -\frac{1}{2\pi} \frac{y_2}{R^2} & \frac{1}{2\pi} \frac{y_1}{R^2} \end{vmatrix} = \frac{1}{2\pi} \mathrm{e}^{-\frac{R^2}{2}}.$$

Since $f_{\mathbf{x}}(h(y)) = 1$ for all $y \in \mathbb{R}^2$, from (14.8) we have

$$f_{\mathbf{y}}(y) = f_{\mathbf{x}}\big(h(y)\big) J(x \to y) = \frac{1}{2\pi} \mathrm{e}^{-\frac{R^2}{2}} = \left(\frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{y_1^2}{2}}\right)\left(\frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{y_2^2}{2}}\right).$$

14.3.1 Rejection Methods

We study two classes of rejection methods: the first one is based on the concept of rejection from a "dominating density." The second is for uniform generation in given sets and it is based on rejection from a bounding set. The two methods are obviously related, and a critical issue in both cases is their numerical efficiency.

Rejection from Dominating Density We now discuss the standard version of a rejection algorithm for sample generation from a multivariate density $f_{\mathbf{x}}(x)$, $\mathbf{x} \in \mathbb{R}^n$. A basic result on multivariate densities, see for instance [133], is the following.

Theorem 14.3 *Let $f_{\mathbf{x}}(x) : \mathbb{R}^n \to \mathbb{R}$ be a density function, and consider the set $S \subset \mathbb{R}^{n+1}$ defined as*

$$S = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} : x \in \mathbb{R}^n,\ u \in \mathbb{R},\ 0 \le u \le f_{\mathbf{x}}(x) \right\}.$$

If the random vector $\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}$ is uniformly distributed in S, then $\mathbf{x}$ has density function $f_{\mathbf{x}}(x)$ on $\mathbb{R}^n$. On the other hand, if $\mathbf{x} \in \mathbb{R}^n$ has pdf $g_{\mathbf{x}}(x)$ and $\mathbf{w} \in \mathbb{R}$ is uniformly distributed on the interval $[0, 1]$, then the random vector $\begin{bmatrix} \mathbf{x} \\ \eta \mathbf{w} g_{\mathbf{x}}(x) \end{bmatrix}$ is uniformly distributed in the set

$$S_d = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} : x \in \mathbb{R}^n,\ u \in \mathbb{R},\ 0 \le u \le \eta g_{\mathbf{x}}(x),\ \eta > 0 \right\}.$$

Using this theorem, we now present the basic rejection scheme. Let $f_{\mathbf{x}}(x)$ be a density function on $\mathbb{R}^n$ and let $g_{\mathbf{x}}(x)$ be a *dominating density* for $f_{\mathbf{x}}(x)$, i.e. a density such that

$$f_{\mathbf{x}}(x) \le \eta g_{\mathbf{x}}(x) \tag{14.9}$$

for some constant $\eta \ge 1$. Random samples from $f_{\mathbf{x}}(x)$ can be obtained using the following algorithm.

Algorithm 14.2 (Rejection from a dominating density) *Given a density function $f_{\mathbf{x}}(x)$ and a dominating density $g_{\mathbf{x}}(x)$ satisfying (14.9), this algorithm returns a random vector $\mathbf{x} \in \mathbb{R}^n$ with pdf $f_{\mathbf{x}}(x)$.*

1. Generate a random vector $\mathbf{x} \in \mathbb{R}^n$ with pdf $g_{\mathbf{x}}(x)$;
2. Generate a random variable $\mathbf{w} \in \mathbb{R}$ uniform in $[0, 1]$;
3. If $\eta \mathbf{w} g_{\mathbf{x}}(\mathbf{x}) \le f_{\mathbf{x}}(\mathbf{x})$ return $\mathbf{x}$ else goto 1.

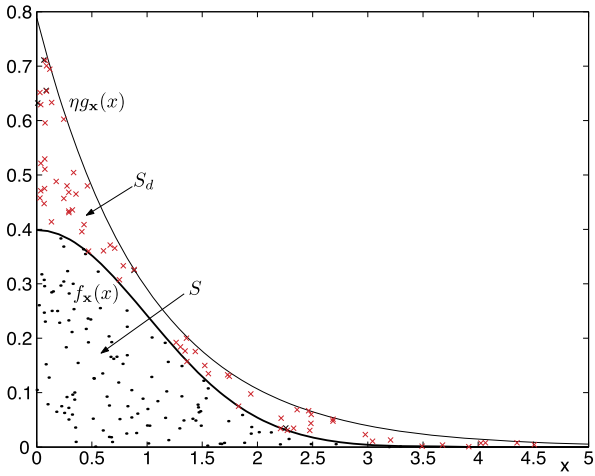

Fig. 14.3 Rejection from a dominating density

The interpretation of the rejection algorithm is given in Fig. 14.3. First, random samples

$$\begin{bmatrix} \mathbf{x}^{(i)} \\ \eta \mathbf{w}^{(i)} g_{\mathbf{x}}(\mathbf{x}^{(i)}) \end{bmatrix}$$

are generated uniformly inside the dominating set S_d. These uniform samples are obtained by generating $\mathbf{x}^{(i)}$ according to $g_{\mathbf{x}}(x)$, and $\mathbf{w}^{(i)}$ uniform in $[0, 1]$. Then, samples lying outside the set S (red crosses) are rejected. The remaining ones (black dots) are uniform in S and therefore, by Theorem 14.3, their projection onto the x space is distributed according to $f_{\mathbf{x}}(x)$.

It is worth noting that two features are essential for a rejection algorithm to work properly: first, the sample generation according to the dominating density $g_{\mathbf{x}}(x)$ should be feasible and simple. Second, the value of η should be known and not too large.

Remark 14.2 (Rejection rate) A critical parameter assessing the efficiency of a rejection algorithm is the *rejection rate*, defined as the expected value of the number of samples that have to be drawn from the dominating density $g_{\mathbf{x}}(x)$ in order to find one "good" sample. For Algorithm 14.2, the rejection rate coincides with the value of the constant η. The variance of the rejection rate is also related to η and it is given by $\eta(\eta - 1)$, see for instance [133].

Example 14.9 (A Gamma generator based on rejection) An efficient algorithm for generation of samples distributed according to $G_{a,1}$, $a \in (0, 1)$, can be derived via rejection from the Weibull density $W_a(x)$ given in (2.10). In fact

$$G_{a,1}(x) \leq \eta W_a(x), \quad x \geq 0$$

where the rejection constant η is given by

$$\eta = \frac{\mathrm{e}^{(1-a)a^{a/(1-a)}}}{\Gamma(1+a)} < \mathrm{e}.$$

Various other Gamma generators are discussed in detail in [133]. For instance, Gamma generators for $a > 1$ may be obtained via rejection from a so-called Burr XII density.

Uniform Densities in Bounded Sets Via Rejection Rejection methods may also be used to generate uniform samples in generic bounded domains of $\mathbb{R}^n$. In this case, these methods are often called "hit-or-miss" and they are related to the techniques based on importance sampling, where the samples are chosen according to their relevance, see e.g. [177, 334]. The basic idea of rejection methods for uniform densities in bounded domains is now recalled.

Let S and S_d be bounded sets, such that $S \subseteq S_d$. Suppose that a uniform generator in the "dominating set" S_d is available, then to generate uniform samples in S we proceed according to the following algorithm.

Algorithm 14.3 (Rejection from a dominating set) *Given a set S and a dominating set $S_d \supseteq S$, this algorithm returns a random vector* $\mathbf{x}$, *uniformly distributed in S.*

1. Generate a random vector $\mathbf{x} \in S_d$;
2. If $\mathbf{x} \in S$ return $\mathbf{x}$ else goto 1.

The rejection rate η of this method is given by the ratio of the volumes of the two sets

$$\eta = \frac{\mathrm{Vol}(S_d)}{\mathrm{Vol}(S)}. \tag{14.10}$$

The effectiveness of this method is clearly related to three basic factors: first, the efficiency of the uniform generator in S_d, second the numerical evaluation of the membership of the sample $\mathbf{x}^{(i)}$ in S; and third, the ratio of volumes η, which is in general the most critical factor. Notice that, in principle, it is often possible to bound the set S with a suitable n-dimensional hyperrectangle, in which uniform generation is straightforward. However, this usually results in a dramatically large value of the rejection rate η, as shown in the following simple example.

Example 14.10 (Uniform generation in a sphere via rejection) Suppose one is interested in generating uniform samples in the Euclidean sphere of unit radius $S = \mathcal{B}_{\|\cdot\|_2} = \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$. Clearly, the sphere is bounded by the n-dimensional

hypercube $S_d = [-1, 1]^n$. The volume of the hypercube is 2^n, while the volume of the sphere is given by the formula

$$\mathrm{Vol}\big(\mathcal{B}_{\|\cdot\|_2}(\mathbb{R}^n)\big) = \frac{2\pi^{n/2}}{n\Gamma(n/2)}.$$

The rejection rate grows with respect to the dimension n as

$$\eta(n) = \left(\frac{2}{\sqrt{\pi}}\right)^n \Gamma(n/2+1).$$

For small n, we have $\eta(1) = 1$, $\eta(2) = 1.2732$, $\eta(3) = 1.9099$, whereas for large n the method is not viable since we have for instance $\eta(20) = 4.06\times10^7$, $\eta(30) = 4.90\times10^{13}$. This "curse of dimensionality" turns out to be a common problem for set rejection methods, as discussed further in Chap. 15.

14.3.2 Conditional Density Method

The conditional density method is another standard method for random generation with multivariate distributions [133]. This is a recursive method in which the individual entries of the multivariate samples are generated according to their conditional probability density. In particular, let $f_{\mathbf{x}}(x)$ be the joint pdf of a vector of random variables $\mathbf{x} = [\mathbf{x}_1 \cdots \mathbf{x}_n]^T$. This pdf can be written as

$$f_{\mathbf{x}}(x_1, \ldots, x_n) = f_{\mathbf{x}_1}(x_1) f_{\mathbf{x}_2|x_1}(x_2|x_1) \cdots f_{\mathbf{x}_n|x_1\cdots x_{n-1}}(x_n|x_1 \cdots x_{n-1})$$

where $f_{\mathbf{x}_i|x_1,\ldots,x_{i-1}}(x_i|x_1, \ldots, x_{i-1})$ are the conditional densities. These densities are defined, see (2.2), as the ratio of marginal densities

$$f_{\mathbf{x}_i|x_1,\ldots,x_{i-1}}(x_i|x_1, \ldots, x_{i-1}) = \frac{f_{\mathbf{x}_1,\ldots,\mathbf{x}_i}(x_1, \ldots, x_i)}{f_{\mathbf{x}_1,\ldots,\mathbf{x}_{i-1}}(x_1, \ldots, x_{i-1})}.$$

In turn, the marginal densities $f_{\mathbf{x}_1,\ldots,\mathbf{x}_i}(x_1, \ldots, x_i)$ are given by

$$f_{\mathbf{x}_1,\ldots,\mathbf{x}_i}(x_1, \ldots, x_i) = \int \cdots \int f_{\mathbf{x}}(x_1, \ldots, x_n)\, \mathrm{d}x_{i+1} \cdots \mathrm{d}x_n.$$

A random vector $\mathbf{x}$ with density $f_{\mathbf{x}}(x)$ can therefore be obtained by generating sequentially the $\mathbf{x}_i$, $i = 1, \ldots, n$, where $\mathbf{x}_i$ is distributed according to the univariate density $f_{\mathbf{x}_i|x_1,\ldots,x_{i-1}}(x_i)$.

The basic idea of this method, therefore, is to generate the first random variable according to $f_{\mathbf{x}_1}(x_1)$, then generate the next one conditional on the first one, and so forth. In other words, the conditional density method reduces an n-dimensional generation problem to n one-dimensional problems. The main difficulty in its application arises from the fact that the computation of the marginal densities is necessary. This is often a very difficult task, since it requires the computation of multiple integrals, see for instance [134]. The conditional density method is one of the basic tools used in Chap. 18 for generating random matrices uniformly distributed in norm bounded sets.

14.4 Asymptotic Methods Based on Markov Chains

In this section, we briefly account for a large body of literature on sampling methods generally known as Markov chain Monte Carlo (MCMC). The basic idea behind these methods is to obtain the desired distribution by simulating a random walk on a graph. The sequence of randomly visited nodes on the graph constitutes a Markov chain, and the output distribution is the stationary distribution of the Markov chain (when such a distribution exists). Since the stationary distribution is only achieved after a certain "burn-in" period of time, these methods are *asymptotic*. This makes a substantial difference with the "exact" nonasymptotic methods previously studied in this chapter. The main difficulty is actually to prove bounds on the burn-in period (or mixing time), after which one has the guarantee that the chain has reached steady state. Some of these issues are discussed in the following sections.

14.4.1 Random Walks on Graphs

In this section, we discuss discrete distributions obtained as asymptotic distributions (among which the uniform distribution is a particular case) of random walks on the nodes of *undirected* graphs. The exposition here is based on the surveys [64, 267], to which the reader is referred to for a more extensive treatment and pointers to the existing literature.

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an undirected connected graph[2] with vertex (or node) set $\mathcal{V}$ of cardinality n, and edge set $\mathcal{E}$ of cardinality m. The *degree* $d(v)$ of a vertex $v \in \mathcal{V}$ is defined as the number of edges incident on v; if every vertex has degree d, then the graph is said d-regular. We now define a random walk on the nodes of $\mathcal{G}(\mathcal{V}, \mathcal{E})$, with initial state $\mathbf{v}_0$, as a sequence of random variables $\mathbf{v}_0, \mathbf{v}_1, \ldots$ taking values in $\mathcal{V}$, such that for $i, j \in \mathcal{V}$ and $k \geq 0$

$$p_{ij} \doteq \text{PR}\{\mathbf{v}_{k+1} = j | \mathbf{v}_k = i\} = \begin{cases} \frac{1}{d(i)} & \text{if } i, j \in \mathcal{E}; \\ 0 & \text{otherwise.} \end{cases} \tag{14.11}$$

This means that, if at the kth step we are at node $v_k = i$, we move to some neighboring node with probability $1/d(i)$. Clearly, the sequence of random nodes $\mathbf{v}_0, \mathbf{v}_1, \ldots$ is a Markov chain. Denoting with $\pi(k)$ the probability distribution of the nodes at time k (i.e. $\pi_i(k) \doteq \text{PR}\{\mathbf{v}_k = i\}$, for $i = 1, \ldots, n$), we have the recursion

$$\pi(k+1) = P^T \pi(k)$$

where P is the transition probability matrix having p_{ij} in the ith row and jth column. A standard result from the elementary theory of Markov chains is that the chain admits a stationary distribution π, which solves $\pi = P^T \pi$, and that this distribution is unique, if the graph is connected (we refer the reader to [285] for a treatment of the

[2] An undirected graph is connected if there exists a path between all pairs of its vertices $i, j \in \mathcal{V}$.

convergence properties of Markov chains). For the assigned transition probabilities, the stationary distribution is given by

$$\pi_i = \frac{d(i)}{2m}, \quad i = 1, \ldots, n.$$

If the graph is d-regular, then this stationary distribution is the uniform distribution, since $d(i) = d$, for all nodes. Notice that for d-regular graphs the corresponding Markov chain is symmetric, meaning that the probability of moving to state j, given that the chain is at node i, is the same as the probability of moving to state i, given that the chain is at node j. Therefore, for d-regular graphs the transition matrix P is symmetric.

An important property of nonbipartite[3] (but not necessarily d-regular) graphs is that the distribution of $\pi(k)$ tends to the stationary distribution as $k \to \infty$, regardless of the initial distribution $\pi(0)$. Therefore, for a d-regular nonbipartite graph with transition probabilities given by (14.11), the distribution $\pi(k)$ tends to the uniform distribution $\pi_i = 1/n$, for $i = 1, \ldots, n$.

A natural question at this point is how many steps one has to wait before the distribution of $\mathbf{v}_k$ is close to the stationary one, which leads to the fundamental notion of the *mixing rate* of the Markov chain. The mixing rate is a measure of how fast the random walk converges to its stationary distribution. If the graph is nonbipartite, then $p_{ij}(k) = d(j)/(2m)$ for $k \to \infty$ and the mixing rate $\lambda > 0$ is defined as

$$\lambda = \limsup_{k \to \infty} \max_{i,j} \left| p_{ij}(k) - \frac{d(j)}{2m} \right|^{1/k}.$$

For a random walk starting at node i, we can express the deviation from stationarity in terms of the total variation distance $|\pi_j(k) - \pi_j|$. This quantity is bounded as follows, see [267]

$$\left|\pi_j(k) - \pi_j\right| \leq \lambda^k \sqrt{\frac{d(j)}{d(i)}}.$$

The next theorem establishes a key result for the mixing rate of a random walk on a graph.

Theorem 14.4 *The mixing rate of a random walk on a nonbipartite graph is given by*

$$\lambda = \max\left\{|\lambda_2|, |\lambda_n|\right\}$$

where $\lambda_1 = 1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1$ *are the eigenvalues of* $M \doteq D^{-1/2} P D^{1/2}$, $D = \operatorname{diag}([1/d(1) \ \cdots \ 1/d(n)])$.

[3] A bipartite graph is a graph whose nodes can be partitioned into two subsets, with no edge joining nodes that are in the same subset. A node in one of the subsets may be joined to all, some, or none of the nodes in the other. A bipartite graph is usually shown with the two subsets as top and bottom rows of nodes, or with the two subsets as left and right columns of nodes.

Very often the matrix M is positive semidefinite (or the graph can be transformed into one for which M is semidefinite, by adding self-loops to the nodes), in which case the mixing rate is simply given by $\lambda = \lambda_2$.

Unfortunately, however, in the applications where the random walk method is of interest, the "size" n of the underlying graph is exponentially large, and the estimation of the mixing rate via eigenvalue computation is very difficult. This is the case for instance in the "card deck shuffling" problem, where we wish to sample uniformly over all permutations of 52 elements, or we are willing to sample uniformly in a convex body[4] $K \subset \mathbb{R}^n$ (with large n), by defining a random walk on a fine lattice inside K, see for instance [152]. For this reason, a different approach is usually taken in the literature for proving the "fast mixing" property of a random walk. In this approach, further concepts such as the *conductance* of a graph and isoperimetric inequalities are introduced, which are outside the scope of this chapter. The interested reader is referred to [64, 218, 267] and to Sect. 19.2 for results regarding random walks on directed graphs with applications to the PageRank computation.

Metropolis Random Walk The so-called Metropolis random walk [283] is a modification of the simple random walk described in the previous section. This modification is introduced in order to make the random walk converge asymptotically to an arbitrary desired probability distribution. Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a d-regular graph, let $f : \mathcal{V} \to \mathbb{R}_+$, and let $\mathbf{v}_0$ be the initial node. Suppose that at time k we are at node $\mathbf{v}_k$, the modified random walk goes as follows: (i) select a random neighbor $\mathbf{u}$ of $\mathbf{v}_k$; (ii) if $f(u) \geq f(v_k)$, then move to $\mathbf{u}$, i.e. $\mathbf{v}_{k+1} = \mathbf{u}$; else move to $\mathbf{u}$ with probability $f(u)/f(v_k)$, or stay in $\mathbf{v}_k$ with probability $1 - f(u)/f(v_k)$. Clearly, this modified random walk is still a Markov chain. The fundamental fact that can be proven is that this chain admits the stationary distribution

$$\widetilde{\pi}_i = \frac{f(i)}{\sum_{j \in \mathcal{V}} f(j)}, \quad i = 1, \ldots, n.$$

Unlike the simple random walk, in general it is difficult to estimate the mixing rate of the Metropolis walk. Relevant results in this direction include [27, 135].

14.4.2 Methods for Continuous Distributions

The idea of constructing random walks on graphs can be extended to the case when the node set of the graph is a "continuous," or dense set. In this section, we discuss some well-known methods to generate (asymptotically) random samples distributed according to a desired continuous distribution.

[4] A convex body is a closed, bounded, convex set of nonzero volume.

Metropolis–Hastings Sampler Suppose that the problem is to sample from a multivariate probability density $f(x)$, $x \in \mathbb{R}^n$, and we can evaluate $f(x)$ (up to a constant), but have no means to generate a sample directly. As discussed previously, the rejection method is an option if a dominating density can be found with a computable bound η of moderate magnitude, which is unlikely if the dimension n is high. We remark that rejection, inversion and the density transformation methods all produce independent realizations from $f(x)$. If these methods are inefficient, or difficult to implement, then we can adopt the weaker goal of generating a *dependent* sequence with marginal distribution equal to (or converging to) $f(x)$, using some variant of the random walk techniques describer earlier. It should be pointed out that this necessitates giving up independence of the samples, since the successive outcomes of a Markov chain (even when the chain is in steady state) are *not* independent.

The Metropolis–Hastings (MH) algorithm [192] builds the random walk (with continuous node space) according to the following rules. Suppose we have a transition density $g(y|x)$, such that

$$\Pr\{\mathbf{x}_{k+1} \in \mathcal{Y} | \mathbf{x}_k = x\} = \int_{\mathcal{Y}} g(y|x)\, \mathrm{d}y.$$

This density is sometimes called a "proposal density" and plays a role similar to the dominating density in the standard rejection method. Define the Metropolis–Hastings ratio by

$$\eta(x, y) \doteq \begin{cases} \min\left(\frac{f(y)g(x|y)}{f(x)g(y|x)}, 1\right) & \text{if } f(x)g(y|x) > 0; \\ 1 & \text{otherwise.} \end{cases} \tag{14.12}$$

Notice that only density evaluations up to a constant are required, since unknown normalizing constants cancel out when forming the above ratio. Suppose that at time k the state is at $\mathbf{x}_k$, then we choose the next state according to the following procedure: (i) draw a random sample $\mathbf{y}$ from the proposal density $g(y|x_k)$; (ii) with probability $\eta(x_k, y)$ move to $\mathbf{y}$, i.e. set $\mathbf{x}_{k+1} = \mathbf{y}$, otherwise stay put, i.e. set $\mathbf{x}_{k+1} = \mathbf{x}_k$.

It can be proved that the resulting Markov chain will reach steady state and that the stationary density is the target density $f(x)$. The analysis of the mixing rate of this algorithm is not, however, an easy task. We refer to [282] for a contribution in this direction and to [177] for further references on variations and refinements of the basic MH algorithm. Metropolis–Hastings sampling and general Markov chain Monte Carlo methods for estimation of expectations are presented in [258, 364, 365]. The explicit Metropolis–Hastings sampling algorithm is reported next.

Algorithm 14.4 (Metropolis–Hastings) *Given a target probability density* $f(x)$, *a "proposal density"* $g(y|x)$, *and a "burn-in" period* $T \gg 0$, *this algorithm returns a random variable* $\mathbf{x}$ *that is approximately distributed according to* $f(x)$ (*the approximating density asymptotically converges to* $f(x)$ *as* T *increases*).

1. Select an arbitrary initial state $\mathbf{x}_0$, and set $k = 0$;
2. Generate a candidate state $\mathbf{y}$ according to $g(y|x_k)$;
3. Compute the ratio $\eta(x_k, y)$ in (14.12);
4. Generate $\mathbf{w}$ uniformly in $[0, 1]$;
5. If $\mathbf{w} \leq \eta(x_k, y)$, set $\mathbf{x}_{k+1} = \mathbf{y}$, else set $\mathbf{x}_{k+1} = \mathbf{x}_k$;
6. Set $k = k + 1$;
7. If $k = T$, return $\mathbf{x}_k$ and end, else goto 2.

Remark 14.3 (Properties of Metropolis–Hastings) It is worth underlining that the output of Algorithm 14.4 is one single random sample, approximately distributed according to $f(x)$. Therefore, if another run of the algorithm is executed, and then a third run, and so on, we would obtain a sequence of samples that are all (approximately) distributed according to $f(x)$, and statistically *independent*, since after T steps the Markov chain approximately "forgets" its initial state. However, this procedure would be very inefficient, since at each run the algorithm should execute T burn-in iterations, with T that is typically very large.

Notice that this is not in contradiction with our initial statement, that Markov chain methods produce *dependent* samples, since the actual use of these algorithms is to produce not a single sample at step $k = T$, but the whole sequence of dependent samples that is generated by the chain from step T on. This is obtained by substituting step 7 in Algorithm 14.4 with "If $k \geq T$, return $\mathbf{x}_k$, goto 2; else goto 2."

14.4.3 Uniform Sampling in a Convex Body

Random walk techniques have been successfully adopted to develop polynomial-time algorithms that produce approximately uniform (although not independent) samples in a convex body $K \subset \mathbb{R}^n$. It is further commonly assumed that K contains the unit ball, and that it is contained in a ball of radius r.

In [152] an algorithm based on random walk on a sufficiently dense lattice inside K was originally proposed, and it was proved that the walk mixes in time polynomial in n (notice that this is far from obvious, since the number of nodes in the lattice grows exponentially with n). The motivation of [152] was to design a polynomial-time algorithm to approximate the volume of a convex body, which is an "ancient" and extremely difficult problem. Indeed, several negative results on the computational complexity of this problem have appeared in the literature, stating for instance that any (deterministic) algorithm that approximates the volume within

a factor of $n^{o(n)}$ necessarily takes exponential time [40, 231]. Therefore, the result of [152] has a particular importance, since it gives a breakthrough in the opposite direction, showing that randomization provably helps in solving efficiently those problems that are deterministically hard.

A similar method called "walk with ball steps" may be easily described. Let $\mathbf{x} \in K$ be the current point: (i) we generate a random point $\mathbf{z}$ in the Euclidean ball of radius δ centered in $\mathbf{x}$; (ii) if $\mathbf{z} \in K$, we move to $\mathbf{z}$, else we stay at $\mathbf{x}$. This procedure corresponds to a random walk on the graph whose vertex set is K, with two points x, z connected by an edge if and only if $|x - z| \leq \delta$. It has been proved in [226] that (a slightly modified version of) this walk converges to an approximation of the uniform distribution in K which gets better as δ is reduced, and that the walk mixes in $O(nr^2/\delta^2)$ steps, provided that $\delta < 1/\sqrt{n}$.

Another popular method for generating (asymptotically) uniform samples in K is the so-called hit-and-run (HR) method introduced in [359] and reported next.

Algorithm 14.5 (Hit-and-run) *Given a convex body $K \subset \mathbb{R}^n$ and a "burn-in" period $T \gg 0$, this algorithm returns a random variable* $\mathbf{x}$ *that is approximately uniform within K (the approximating density asymptotically converges to the uniform density in K as T increases).*

1. Select an arbitrary initial state $\mathbf{x}_0$ in the interior of K, and set $k = 0$;
2. Generate a random direction $\mathbf{v} = \mathbf{y}/\|\mathbf{y}\|$, where $\mathbf{y} \sim \mathcal{N}_{0,I_n}$;
3. Compute the extreme points $\mathbf{a}, \mathbf{b}$ (on the boundary of K) of the chord in K through $\mathbf{x}_k$ along direction $\mathbf{v}$;
4. Generate $\mathbf{w}$ uniformly in $[0, 1]$;
5. Set $\mathbf{x}_{k+1} = \mathbf{wa} + (1 - \mathbf{w})\mathbf{b}$;
6. Set $k = k + 1$;
7. If $k = T$, return $\mathbf{x}_k$ and end, else goto 2.

In the HR method, the random walk is constructed as follows: if the current point is $\mathbf{x}$, then we generate the next point by selecting a random line through $\mathbf{x}$ (uniformly over all directions) and choosing the next point uniformly from the segment of the line in K. This walk has the uniform as the stationary distribution, and mixes in $O(n^2r^2)$ steps [268].

The techniques for generating uniform samples in convex bodies have been exploited in the context of convex optimization. In particular, in the work [57, 221] polynomial-time randomized algorithms for solving convex optimization problems are derived. These techniques have been specialized in [119], that focuses on the case when one aims at computing the maximum of a linear function

$$\gamma_{\max} = \max_{x \in K} c^T x, \tag{14.13}$$

over the convex body $K \subset \mathbb{R}^n$. The paper proposes a sequential method based on cutting-plane, which is proved to improve with high probability over [57]. The basic

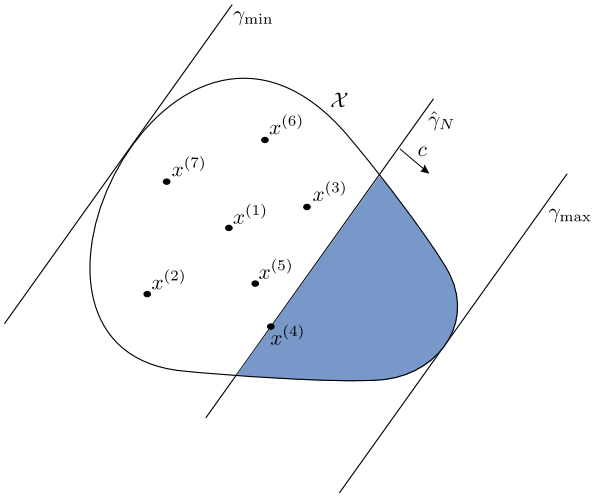

Fig. 14.4 Setup of Theorem 14.5

step of this method consists in an empirical maximum estimation, as discussed in Sect. 8.4. That is, N samples *uniformly* distributed over K

$$\mathbf{x}^{(1)}, \quad \dots, \quad \mathbf{x}^{(N)}$$

are drawn, and the empirical maximum $\widehat{\boldsymbol{\gamma}}_N = \max_{i=1,\dots,N} c^T \mathbf{x}^{(i)}$ is constructed. Then, the following theorem, proved in [119], provides upper and lower bounds on the expected *relative error* between the empirical estimate and the true maximum.

Theorem 14.5 *Let $K \subset \mathbb{R}^n$ be a convex body. Given $c \in \mathbb{R}^n$, it holds that*

$$\frac{1}{nN+1} \leq \mathrm{E}_{\mathbf{x}^{(1\dots N)}} \left(\frac{\gamma_{\max} - \widehat{\boldsymbol{\gamma}}_N}{\gamma_{\max} - \gamma_{\min}} \right) \leq \frac{1}{n} B(N+1, 1/n)$$
$$\leq \left(\frac{1}{N+1} \right)^{\frac{1}{n}}, \tag{14.14}$$

where $\gamma_{\min} = \min_{x \in K} c^T x$ and $B(a,b)$ is the Beta function

$$B(a,b) \doteq \int_0^1 t^{a-1}(1-t)^{b-1}\,\mathrm{d}t.$$

The setup of the theorem is illustrated in Fig. 14.4. We remark that the bound (14.14) is rather different in nature with respect to the bound provided in Sect. 8.4, for the following reasons: (i) it hold *specifically* for uniformly distributed samples, (ii) it provides a one level of probability type statement. In particular, this result can be used to prove that the expected number of samples N necessary for obtaining an estimate with *guaranteed* error less that α, that is such that

$$\mathrm{E}_{\mathbf{x}^{(1\dots N)}} \left(\frac{\gamma_{\max} - \widehat{\boldsymbol{\gamma}}_N}{\gamma_{\max} - \gamma_{\min}} \right) \leq \alpha,$$

grows exponentially in α as $N = \lceil \frac{1}{\alpha^n} \rceil$.

Chapter 15
Statistical Theory of Random Vectors

In this chapter we study the statistical properties of random vectors, for a certain class of symmetric probability distributions. In particular, we introduce the notions of ℓ_p radial and ℓ_2^W radial random vectors, and analyze the properties of these distributions, highlighting the connection among ℓ_p radial symmetry and the uniform distribution in ℓ_p norm balls. These properties will be used in the algorithms for vector sample generation presented in Chap. 16.

15.1 Radially Symmetric Densities

In this section we introduce a class of probability distributions with symmetry properties with respect to ℓ_p norms, and discuss some preliminary concepts that will be used in later parts of the book. We first formally define the notion of radially symmetric density functions (or simply radial densities).

Definition 15.1 (ℓ_p radial density) A random vector $\mathbf{x} \in \mathbb{F}^n$, where $\mathbb{F}$ is either the real or complex field, is ℓ_p radial if its density function can be written as

$$f_{\mathbf{x}}(x) = g(\rho), \quad \rho = \|x\|_p$$

where $g(\rho)$ is called the *defining function* of $\mathbf{x}$.

In other words, for radially symmetric random vectors, the density function is uniquely determined by its radial shape, which is described by the defining function $g(\rho)$. When needed, we use the notation $g_{\mathbf{x}}(\rho)$ to specify the defining function of the random variable $\mathbf{x}$. For given ρ, the level set of the density function is an equal probability set represented by $\partial\mathcal{B}_{\|\cdot\|_p}(\rho)$. Examples of radial densities for real random vectors are, for instance, the normal density and the Laplace density. This is shown in the next example.

Example 15.1 (Radial densities) The classical multivariate normal density with identity covariance matrix and zero mean is an ℓ_2 radial density. In fact, setting

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_15,

$\rho = \|x\|_2$, we write

$$f_{\mathbf{x}}(x) = \frac{1}{\sqrt{(2\pi)^n}} \mathrm{e}^{-\frac{1}{2}x^T x} = \frac{1}{\sqrt{(2\pi)^n}} \mathrm{e}^{-\frac{1}{2}\rho^2} = g(\rho).$$

The multivariate Laplace density with zero mean is an ℓ_1 radial density. Indeed, setting $\rho = \|x\|_1$, we have

$$f_{\mathbf{x}}(x) = \frac{1}{2^n} \mathrm{e}^{-\sum_{i=1}^n |x_i|} = \frac{1}{2^n} \mathrm{e}^{-\rho} = g(\rho).$$

Furthermore, the generalized Gamma density, defined in (2.13), is radial with respect to the ℓ_p norm. The properties of this density are discussed in more detail in Sect. 16.2.

Another notable case of radial density is the uniform density over $\mathcal{B}_{\|\cdot\|_p}$. In fact, for a random vector $\mathbf{x}$ uniformly distributed in the ℓ_p norm ball of unit radius we have that

$$f_{\mathbf{x}}(x) = \mathcal{U}_{\mathcal{B}_{\|\cdot\|_p}}(x) = \begin{cases} \frac{1}{\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_p})} & \text{if } \|x\|_p \leq 1; \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, this density can be expressed as a function of $\rho = \|x\|_p$, i.e. $f_{\mathbf{x}}(x) = g(\rho)$, with defining function

$$g(\rho) = \begin{cases} \frac{1}{\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_p})} & \text{if } \rho \leq 1; \\ 0 & \text{otherwise.} \end{cases} \tag{15.1}$$

15.2 Statistical Properties of ℓ_p Radial Real Vectors

The next theorem provides a fundamental characterization of ℓ_p radial real random vectors, and relates them to the uniform distribution on the surface of the ℓ_p norm ball.

Theorem 15.1 (ℓ_p radial vectors in $\mathbb{R}^n$) *Let $\mathbf{x} \in \mathbb{R}^n$ be factored as $\mathbf{x} = \boldsymbol{\rho}\mathbf{u}$, where $\boldsymbol{\rho} > 0$, and $\mathbf{u} \in \mathbb{R}^n$, $\|\mathbf{u}\|_p = 1$. The following statements are equivalent*:

1. *$\mathbf{x}$ is ℓ_p radial with defining function $g(\rho)$;*
2. *$\boldsymbol{\rho}$, $\mathbf{u}$ are independent, and $\mathbf{u}$ is uniformly distributed on $\partial\mathcal{B}_{\|\cdot\|_p}$.*

Moreover, we have that

$$f_{\boldsymbol{\rho}}(\rho) = \frac{2^n \Gamma^n(1/p)}{p^{n-1}\Gamma(n/p)} g(\rho)\rho^{n-1}; \tag{15.2}$$

$$\begin{aligned} f_{\mathbf{u}_1,\ldots,\mathbf{u}_{n-1}}(u_1,\ldots,u_{n-1}) &= \frac{p^{n-1}\Gamma(n/p)}{2^{n-1}\Gamma^n(1/p)} \psi^{(1-p)/p}(u_1,\ldots,u_{n-1}) \\ &\quad |u_i| < 1, \quad i = 1,\ldots,n-1 \\ &\quad \sum_{i=1}^{n-1} |u_i|^p < 1 \end{aligned} \tag{15.3}$$

where

$$\psi(u_1, \ldots, u_{n-1}) \doteq \left(1 - \sum_{i=1}^{n-1} |u_i|^p\right). \tag{15.4}$$

Proof Consider $x \in \mathbb{R}^n$. If we assume $x_n > 0$, then x can be factored as $x = \rho u$, where ρ is positive, and the vector u lies on the unit surface $\partial \mathcal{B}_{\|\cdot\|_p}$ and is parameterized as

$$u = \left[u_1 \quad u_2 \quad \cdots \quad u_{n-1} \quad \left(1 - \sum_{i=1}^{n-1} |u_i|^p\right)^{1/p} \right]^T.$$

The transformation

$$\begin{aligned} x_i &= \rho u_i, \quad i = 1, \ldots, n-1; \\ x_n &= \rho \left(1 - \sum_{i=1}^{n-1} |u_i|^p\right)^{1/p}, \quad \rho > 0,\ |u_i| < 1,\ \sum_{i=1}^{n-1} |u_i|^p < 1 \end{aligned}$$

from the variables $x_1, \ldots, x_n$, $x_n > 0$, to the variables $u_1, \ldots, u_{n-1}$, ρ is then one-to-one. The Jacobian of this transformation is

$$J(x \to u, \rho) = \rho^{n-1} \left(1 - \sum_{i=1}^{n-1} |u_i|^p\right)^{(1-p)/p}. \tag{15.5}$$

Notice that, if we set $x_n < 0$, similar computations lead to the same Jacobian. Furthermore, the case $x_n = 0$ is neglected, since the event $\mathbf{x}_n = 0$ occurs with zero probability. For each of the two half-spaces (for $x_n > 0$ and for $x_n < 0$), the density $f_{\mathbf{x}}(x)$ restricted to the half-space is given by $2g(\rho)$, since $g(\rho)$ represents the density on the whole space. From the rule of change of variables, see Theorem 14.2, we then have

$$f_{\mathbf{u},\mathbf{r}}(u_1, \ldots, u_{n-1}, \rho) = 2g(\rho)\rho^{n-1} \left(1 - \sum_{i=1}^{n-1} |u_i|^p\right)^{(1-p)/p}. \tag{15.6}$$

Then, it follows that $u_1, \ldots, u_{n-1}$ are independent from ρ. Since, see for instance [362]

$$\int_{\mathcal{D}} \left(1 - \sum_{i=1}^{n-1} |u_i|^p\right)^{(1-p)/p} \mathrm{d}u_1 \cdots \mathrm{d}u_{n-1} = \frac{2^{n-1}\Gamma^n(1/p)}{p^{n-1}\Gamma(n/p)} \tag{15.7}$$

where $\mathcal{D} = \{-1 < u_i < 1,\ i = 1, \ldots, n-1;\ \sum_{i=1}^{n-1} |u_i|^p < 1\}$, we obtain the marginal densities (15.2) and (15.3) by integration of (15.6) with respect to $u_1, \ldots, u_{n-1}$ and ρ respectively. □

Remark 15.1 (Uniform density on $\partial \mathcal{B}_{\|\cdot\|_p}(\mathbb{R}^n)$) The density (15.3) is called the "ℓ_p norm uniform distribution" in [187, 362]. The interested reader is referred to

those papers for further details on ℓ_p radial distributions. The pdf (15.3) is therefore a representation of the uniform distribution on the surface of the unit sphere $\partial\mathcal{B}_{\|\cdot\|_p}(\mathbb{R}^n)$ and is denoted by $\mathcal{U}_{\partial\mathcal{B}_{\|\cdot\|_p}(\mathbb{R}^n)}$. We also notice that Theorem 15.1 can be stated in the following alternative form: if $\mathbf{x} \in \mathbb{R}^n$ is ℓ_p radial, then the random vector $\mathbf{u} = \mathbf{x}/\|\mathbf{x}\|_p$ is uniformly distributed on $\partial\mathcal{B}_{\|\cdot\|_p}$, and $\mathbf{u}$ and $\|\mathbf{x}\|_p$ are independent.

Remark 15.2 (Uniform density on $\partial\mathcal{B}_{\|\cdot\|_2}(\mathbb{R}^n)$) For the case $p = 2$, a commonly used parameterization of u is obtained by means of the generalized polar coordinates

$$\begin{aligned}
u_1 &= \sin\phi_1;\\
u_2 &= \cos\phi_1 \sin\phi_2;\\
u_3 &= \cos\phi_1 \cos\phi_2 \sin\phi_3;\\
&\vdots\\
u_{n-1} &= \cos\phi_1 \cos\phi_2 \cdots \cos\phi_{n-2} \sin\phi_{n-1};\\
u_n &= \pm\cos\phi_1 \cos\phi_2 \cdots \cos\phi_{n-2} \cos\phi_{n-1}
\end{aligned}$$

where $-\pi/2 < \phi_i \leq \pi/2$, for $i = 1, \ldots, n-1$. It is easy to verify that $u_n^2 = 1 - \sum_{i=1}^{n-1} u_i^2$. The Jacobian of the transformation from $u_1, \ldots, u_{n-1}$ to $\phi_1, \ldots, \phi_{n-1}$ is, see for instance [22]

$$J(u_1, \ldots, u_{n-1} \to \phi_1, \ldots, \phi_{n-1}) = \cos^{n-1}\phi_1 \cos^{n-2}\phi_2 \cdots \cos\phi_{n-1}.$$

In these coordinates, the uniform density on $\partial\mathcal{B}_{\|\cdot\|_2}$ is

$$f_{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_{n-1}}(\phi_1, \ldots, \phi_{n-1}) = \frac{\Gamma(n/2)}{\Gamma^n(1/2)} \cos^{n-2}\phi_1 \cos^{n-3}\phi_2 \cdots \cos\phi_{n-2}.$$

Remark 15.3 (ℓ_p norm density and volume of $\mathcal{B}_{\|\cdot\|_p}(r, \mathbb{R}^n)$) The *norm density* of an ℓ_p radial random vector $\mathbf{x} \in \mathbb{R}^n$ is defined as the probability density of its norm $\boldsymbol{\rho} = \|\mathbf{x}\|_p$. The norm density of $\mathbf{x}$ is explicitly given in (15.2).

Notice that if $\mathbf{x}$ is uniformly distributed in $\mathcal{B}_{\|\cdot\|_p}(r, \mathbb{R}^n)$, then its defining function is given by (15.1). Therefore, substituting this $g(\rho)$ into (15.2) and integrating for ρ from 0 to r, we obtain a closed-form expression for the volume of the ℓ_p norm ball of radius r in $\mathbb{R}^n$

$$\mathrm{Vol}\big(\mathcal{B}_{\|\cdot\|_p}(r, \mathbb{R}^n)\big) = 2^n \frac{\Gamma^n(1/p+1)}{\Gamma(n/p+1)} r^n. \tag{15.8}$$

In the next section we study ℓ_p radial complex vectors.

15.3 Statistical Properties of ℓ_p Radial Complex Vectors

We now present an analogous result to Theorem 15.1 for the case of complex ℓ_p radial vectors.

Theorem 15.2 (ℓ_p radial vectors in $\mathbb{C}^n$) *Let* $\mathbf{x} \in \mathbb{C}^n$ *be factored as* $\mathbf{x} = \boldsymbol{\rho}\mathbf{v}$, *where* $\boldsymbol{\rho} > 0$, $\mathbf{v} = \mathrm{e}^{j\boldsymbol{\Phi}}\mathbf{u}$, *with* $\mathbf{u} \in \mathbb{R}^n_+$, $\|\mathbf{u}\|_p = 1$, $\boldsymbol{\Phi} = \mathrm{diag}([\boldsymbol{\varphi}_1 \ \cdots \ \boldsymbol{\varphi}_n])$, $\boldsymbol{\varphi}_i \in [0, 2\pi]$. *The following statements are equivalent*:

1. $\mathbf{x}$ *is* ℓ_p *radial with defining function* $g(\rho)$;
2. $\boldsymbol{\rho}$, $\mathbf{u}$, $\boldsymbol{\Phi}$ *are independent, and* $\mathbf{v}$ *is uniformly distributed on* $\partial\mathcal{B}_{\|\cdot\|_p}$.

Moreover, we have that

$$f_{\boldsymbol{\rho}}(\rho) = \frac{(2\pi)^n \Gamma^n(2/p)}{p^{n-1}\Gamma(2n/p)} g(\rho)\rho^{2n-1}; \tag{15.9}$$

$$f_{\mathbf{u}_1,\ldots,\mathbf{u}_{n-1}}(u_1,\ldots,u_{n-1}) = \frac{p^{n-1}\Gamma(2n/p)}{\Gamma^n(2/p)}\widetilde{\psi}(u_1,\ldots,u_{n-1}); \tag{15.10}$$

$$0 < u_i < 1, \quad i = 1,\ldots,n-1$$

$$\sum_{i=1}^{n-1} |u_i|^p < 1$$

$$f_{\boldsymbol{\Phi}}(\varphi_1,\ldots,\varphi_n) = \frac{1}{(2\pi)^n} \tag{15.11}$$

where $\widetilde{\psi}$ *is defined as*

$$\widetilde{\psi}(u_1,\ldots,u_{n-1}) \doteq \left(\prod_{i=1}^{n-1} u_i\right)\psi^{2/p-1}(u_1,\ldots,u_{n-1})$$

and

$$\psi(u_1,\ldots,u_{n-1}) = \left(1 - \sum_{i=1}^{n-1} |u_i|^p\right).$$

Proof Observe that, owing to the norm constraint, the vector u has $n-1$ free components, and it is expressed as $u = [u_1 \ \cdots \ u_{n-1}\psi^{1/p}(u_1,\ldots,u_{n-1})]^T$, $u_i \in [0,1]$, $i = 1,\ldots,n-1$. Let $a_i = \mathrm{Re}(x_i)$, $b_i = \mathrm{Im}(x_i)$, then the change of variables $a_i + jb_i = \rho\mathrm{e}^{j\varphi_i}u_i$ is one-to-one, and is explicitly written as

$$\begin{aligned}
a_1 &= \rho u_1 \cos\varphi_1;\\
b_1 &= \rho u_1 \sin\varphi_1;\\
a_2 &= \rho u_2 \cos\varphi_2;\\
b_2 &= \rho u_2 \sin\varphi_2;\\
&\vdots\\
a_{n-1} &= \rho u_{n-1}\cos\varphi_{n-1};\\
b_{n-1} &= \rho u_{n-1}\sin\varphi_{n-1};\\
a_n &= \rho\psi^{1/p}(u_1,\ldots,u_{n-1})\cos\varphi_n;\\
b_n &= \rho\psi^{1/p}(u_1,\ldots,u_{n-1})\sin\varphi_n.
\end{aligned}$$

	a_1	b_1	a_2	b_2	$\cdots$	a_{n-1}	b_{n-1}	a_n	b_n
φ_1	$-\rho u_1 S\varphi_1$	$\rho u_1 C\varphi_1$	0	0	$\cdots$	0	0	0	0
u_1	$\rho C\varphi_1$	$\rho S\varphi_1$	0	0	$\cdots$	0	0	$u_1^{p-1} C\varphi_n$	$u_1^{p-1} S\varphi_n$
φ_2	0	0	$-\rho u_2 S\varphi_2$	$\rho u_2 C\varphi_2$	$\cdots$	0	0	0	0
u_2	0	0	$\rho C\varphi_2$	$\rho S\varphi_2$	$\cdots$	0	0	$u_2^{p-1} C\varphi_n$	$u_2^{p-1} S\varphi_n$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
φ_{n-1}	0	0	0	0	$\cdots$	$-\rho u_{n-1} S\varphi_{n-1}$	$\rho u_{n-1} C\varphi_{n-1}$	0	0
u_{n-1}	0	0	0	0	$\cdots$	$\rho C\varphi_{n-1}$	$\rho S\varphi_{n-1}$	$u_{n-1}^{p-1} C\varphi_n$	$u_{n-1}^{p-1} S\varphi_n$
φ_n	0	0	0	0	$\cdots$	0	0	$-\rho S\varphi_n$	$\rho C\varphi_n$
ρ	$u_1 C\varphi_1$	$u_1 S\varphi_1$	$u_2 C\varphi_2$	$u_2 S\varphi_2$	$\cdots$	$u_{n-1} C\varphi_{n-1}$	$u_{n-1} S\varphi_{n-1}$	$C\varphi_n$	$S\varphi_n$

To compute the Jacobian of this transformation, we construct the following scheme of partial derivatives, where $S\varphi_i$ and $C\varphi_i$ stand for $\sin\varphi_i$ and $\cos\varphi_i$.

Using the Schur determinant rule, and exploiting the block diagonal structure of the $2(n-1)\times 2(n-1)$ upper-left block in the above table, we obtain

$$J(x\to\Phi,u,\rho)=\rho^{2n-1}\left(\prod_{i=1}^{n-1}u_i\right)\psi^{2/p-1}(u_1,\ldots,u_{n-1}).$$

Therefore, we have that

$$f_{\boldsymbol{\Phi},\mathbf{u},\boldsymbol{\rho}}(\Phi,u,\rho)=g(\rho)J(x\to\Phi,u,\rho)=\rho^{2n-1}g(\rho)\widetilde{\psi}(u_1,\ldots,u_{n-1}) \tag{15.12}$$

which proves the statistical independence of $\boldsymbol{\Phi},\mathbf{u},\boldsymbol{\rho}$. The marginal density $f_{\boldsymbol{\rho}}(\rho)$ in (15.9) is then obtained by integrating this joint pdf over Φ and u, and using the following facts

$$\int_0^{2\pi}\mathrm{d}\varphi_1\cdots\mathrm{d}\varphi_n=(2\pi)^n;$$
$$\int_{\mathcal{D}}\widetilde{\psi}(u_1,\ldots,u_{n-1})\,\mathrm{d}u_1\cdots\mathrm{d}u_{n-1}=\frac{\Gamma^n(2/p)}{p^{n-1}\Gamma(2n/p)}$$

where $\mathcal{D}=\{0<u_i<1,i=1,\ldots,n-1;\ \sum_{i=1}^{n-1}|u_i|^p<1\}$. The marginal densities (15.10) and (15.11) are obtained in a similar way, integrating with respect to Φ,ρ and with respect to u,ρ respectively. Integrating (15.12) with respect to ρ, we obtain the pdf of $\mathbf{v}$, which is defined as the uniform density over $\partial\mathcal{B}_{\|\cdot\|_p}$, see Remark 15.4. □

Remark 15.4 (Uniform pdf on $\partial\mathcal{B}_{\|\cdot\|_p}(\mathbb{C}^n)$) Similar to the development for the real case in [93, 162, 362], the marginal density $f_{\boldsymbol{\Phi},\mathbf{u}}(\Phi,u)$, obtained by integrating (15.12) with respect to ρ, is defined as the "(complex) ℓ_p norm uniform distribution" on the surface of the unit sphere $\partial\mathcal{B}_{\|\cdot\|_p}(\mathbb{C}^n)$, and it is denoted by $\mathcal{U}_{\partial\mathcal{B}_{\|\cdot\|_p}(\mathbb{C}^n)}$. In

addition, notice that Theorem 15.2 could be stated in the following form: if $\mathbf{x} \in \mathbb{C}^n$ is ℓ_p radial, then $\mathbf{v} = \mathbf{x}/\|\mathbf{x}\|_p$ is uniformly distributed on $\partial\mathcal{B}_{\|\cdot\|_p}$, and $\mathbf{v}$ and $\|\mathbf{x}\|_p$ are independent.

Remark 15.5 (ℓ_p norm density and volume of $\mathcal{B}_{\|\cdot\|_p}(r, \mathbb{C}^n)$) The norm density (see Remark 15.3 for its definition) of an ℓ_p radial random vector $\mathbf{x} \in \mathbb{C}^n$ is explicitly given in (15.9). We also notice that if $\mathbf{x}$ is uniformly distributed in $\mathcal{B}_{\|\cdot\|_p}(r, \mathbb{C}^n)$, its defining function is given by (15.1), and hence substituting this $g(\rho)$ into (15.9) and integrating for ρ from 0 to r, we obtain a closed-form expression for the volume of the ℓ_p norm ball of radius r in $\mathbb{C}^n$

$$\mathrm{Vol}\big(\mathcal{B}_{\|\cdot\|_p}\big(r, \mathbb{C}^n\big)\big) = \pi^n \frac{\Gamma^n(2/p+1)}{\Gamma(2n/p+1)} r^{2n}. \tag{15.13}$$

Finally, we observe that the expressions for the norm density (15.9) and (15.2) can be unified in a single formula that is valid for both the real and complex cases. This is stated in the next lemma.

Lemma 15.1 (ℓ_p norm density) *If $\mathbf{x} \in \mathbb{F}^n$ is ℓ_p radial, then the random variable $\boldsymbol{\rho} = \|\mathbf{x}\|_p$ has density function $f_{\boldsymbol{\rho}}(\rho)$ given by*

$$f_{\boldsymbol{\rho}}(\rho) = \mathrm{Vol}(\mathcal{B}_{\|\cdot\|_p}) d\rho^{d-1} g(\rho) \tag{15.14}$$

where $d = n$ if $\mathbb{F} \equiv \mathbb{R}$ or $d = 2n$ if $\mathbb{F} \equiv \mathbb{C}$.

15.4 ℓ_p Radial Vectors and Uniform Distribution in $\mathcal{B}_{\|\cdot\|_p}$

The results of the previous sections provide a connection between ℓ_p radial distributions and uniform distributions within ℓ_p norm balls, for real and complex random vectors. This connection is analyzed in [82] and it is stated in the next corollary.

Corollary 15.1 (Uniform vectors in $\mathcal{B}_{\|\cdot\|_p}$) *Let $d = n$ if $\mathbb{F}$ is the real field, and $d = 2n$ if $\mathbb{F}$ is the complex field. The following two conditions are equivalent*:

1. $\mathbf{x} \in \mathbb{F}^n$ *is ℓ_p radial, with norm density function* $f_{\boldsymbol{\rho}}(\rho) = \rho^{d-1} d$, $\rho \in [0, 1]$;
2. $\mathbf{x} \in \mathbb{F}^n$ *is uniformly distributed in* $\mathcal{B}_{\|\cdot\|_p}$.

Proof 1.$\rightarrow$ 2. Since $\mathbf{x}$ is ℓ_p radial, its norm density is given by (15.14), then the defining function of $\mathbf{x}$ is

$$f_{\mathbf{x}}(x) = g\big(\|x\|_p\big) = \frac{1}{\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_p})}, \quad \|x\|_p \leq 1$$

which implies that the pdf of $\mathbf{x}$ is constant on its domain, i.e. $\mathbf{x}$ is uniformly distributed in $\mathcal{B}_{\|\cdot\|_p}$.

2.→1. Since $\mathbf{x}$ is uniform in $\mathcal{B}_{\|\cdot\|_p}$ then

$$f_{\mathbf{x}}(x) = \begin{cases} \frac{1}{\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_p})} & \text{if } \|x\|_p \leq 1; \\ 0 & \text{otherwise.} \end{cases}$$

Notice that $f_{\mathbf{x}}(x)$ depends only on $\|x\|_p$. Therefore, $\mathbf{x}$ is ℓ_p radial, with defining function $g(\rho) = f_{\mathbf{x}}(x)$, $\rho = \|x\|_p$. Substituting $g(\rho)$ in (15.14), we obtain the norm density as claimed. □

The next corollary shows how to obtain an ℓ_p radial distribution with given defining function, or with given norm density, starting from any arbitrary ℓ_p radial distribution.

Corollary 15.2 *Let $\mathbf{x} \in \mathbb{F}^n$ be ℓ_p radial and let $\mathbf{z} \in \mathbb{R}^+$ be an independent random variable with density $f_{\mathbf{z}}(z)$. Then, the random vector*

$$\mathbf{y} = \mathbf{z}\frac{\mathbf{x}}{\|\mathbf{x}\|_p}$$

is ℓ_p radial with norm density function $f_{\boldsymbol{\rho}}(\rho) = f_{\mathbf{z}}(\rho)$, $\rho = \|y\|_p$. Moreover, the defining function of y, $g_{\mathbf{y}}(\rho)$, is given by

$$g_{\mathbf{y}}(\rho) = \frac{1}{\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_p})\rho^{d-1}d} f_{\boldsymbol{\rho}}(\rho). \tag{15.15}$$

Proof Clearly, $\mathbf{y}$ is ℓ_p radial, and $\|\mathbf{y}\|_p = \mathbf{z}$. Therefore, the norm density function $f_{\boldsymbol{\rho}}(\rho)$ of $\mathbf{y}$ coincides with the density function $f_{\mathbf{z}}(z)$ of $\mathbf{z}$. Relation (15.15) follows immediately from (15.14). □

The previous corollary may be used to generate ℓ_p radial random vectors with a given defining function. In the next corollary, we specialize this result to uniform distributions.

Corollary 15.3 (Uniform vectors in $\mathcal{B}_{\|\cdot\|_p}(r, \mathbb{F}^n)$) *Let $\mathbf{x} \in \mathbb{F}^n$ be ℓ_p radial and let $\mathbf{w} \in \mathbb{R}$ be an independent random variable uniformly distributed in* $[0, 1]$. *Then*

$$\mathbf{y} = r\mathbf{z}\frac{\mathbf{x}}{\|\mathbf{x}\|_p}, \quad \mathbf{z} = \mathbf{w}^{1/d}$$

where $d = n$ if $\mathbb{F} \equiv \mathbb{R}$ or $d = 2n$ if $\mathbb{F} \equiv \mathbb{C}$, is uniformly distributed in $\mathcal{B}_{\|\cdot\|_p}(r)$.

Proof By the inversion method, see Corollary 14.1, it follows that the distribution of $\mathbf{z}$ is $F_{\mathbf{z}}(z) = z^d$, therefore $f_{\mathbf{z}}(z) = dz^{d-1}$. For $r = 1$, the statement is proved by means of Theorem 15.1. With a rescaling, it is immediate to show that $\mathbf{y}$ is uniformly distributed in $\mathcal{B}_{\|\cdot\|_p}(r)$. □

This result can be interpreted as follows: first, an ℓ_p radial random vector $\mathbf{x}$ is normalized to obtain a uniform distribution on the surface $\partial\mathcal{B}_{\|\cdot\|_p}(r)$ of the set $\mathcal{B}_{\|\cdot\|_p}(r)$,

then each sample is projected into $\mathcal{B}_{\|\cdot\|_p}(r)$ by the random *volumetric factor* $\mathbf{z}$. Therefore, the problem of uniform generation is reduced to that of generation of ℓ_p radially symmetric random vectors. This is discussed in more detail in Chap. 16, where explicit algorithms for uniform generation are provided.

15.5 Statistical Properties of ℓ_2^W Radial Vectors

In this section we introduce the notion of real random vectors with ℓ_2^W radial distribution (or ℓ_2^W radial vectors) and discuss their statistical properties. Then, we study how the uniform distribution on the unit ball in the ℓ_2 norm is transformed by a linear mapping. For further details and references on distributions with ellipsoidal support, the reader is referred to [186].

We recall that the ℓ_2^W norm of a vector $x \in \mathbb{R}^n$ is defined in (3.4) as

$$\|x\|_2^W = \left(x^T W^{-1} x\right)^{1/2}$$

where $W = W^T \succ 0$ is a given weighting matrix. We first introduce the notion of ℓ_2^W radially symmetric density.

Definition 15.2 (ℓ_2^W radial density) A random vector $\mathbf{x} \in \mathbb{R}^n$ is ℓ_2^W radial if its probability density function $f_{\mathbf{x}}(x)$ can be expressed in the form

$$f_{\mathbf{x}}(x) = g(\rho), \quad \rho = \|x\|_2^W.$$

Example 15.2 (Normal density as ℓ_2^W radial density) The multivariate normal density $\mathcal{N}_{0,W}$ is ℓ_2^W radial since

$$\mathcal{N}_{0,W}(x) = (2\pi)^{-n/2} |W|^{-1/2} \mathrm{e}^{-\frac{1}{2} g(\rho)}, \quad \rho = \|x\|_2^W.$$

We now state a well-known result, see e.g. [22], regarding the linear mapping of a vector with multivariate normal density.

Lemma 15.2 *Let* $\mathbf{x} \in \mathbb{R}^n \sim \mathcal{N}_{\bar{x},W}$, *and let* $\mathbf{y} = T\mathbf{x}$, *where* $\mathbf{y} \in \mathbb{R}^m$, *and* T *may be rank deficient. Then,* $\mathbf{y}$ *is also normally distributed, with* $\mathrm{E}(\mathbf{y}) = T\bar{x}$ *and* $\mathrm{Cov}(\mathbf{y}) = TWT^T$.

The following lemma states the relation between linear transformations of ℓ_2 radial vectors and ℓ_2^W radial vectors.

Lemma 15.3 *Let* $\mathbf{x} \in \mathbb{R}^n$ *be* ℓ_2 *radial with defining function* $g_{\mathbf{x}}(\|x\|_2)$, *and let* $\mathbf{y} = T\mathbf{x}$, *with* $T \in \mathbb{R}^{n,n}$ *invertible. Then,* $\mathbf{y}$ *is* ℓ_2^W *radial with* $W = TT^T$, *and has defining function*

$$g_{\mathbf{y}}(\rho) = |W|^{-1/2} g_{\mathbf{x}}(\rho), \quad \rho = \|y\|_2^W.$$

Proof By assumption, the random vector $\mathbf{x}$ is ℓ_2 radial, therefore $f_{\mathbf{x}}(x) = g_{\mathbf{x}}(\|x\|_2)$. Since $x = T^{-1}y$, from the rule of change of variables, see Theorem 14.2, we have

$$f_{\mathbf{y}}(y) = f_{\mathbf{x}}\big(T^{-1}y\big)J(x \to y)$$

where $J(x \to y) = |T|^{-1}$. Letting $W = TT^T$, this expression becomes

$$f_{\mathbf{y}}(y) = g_{\mathbf{x}}\big(\big\|T^{-1}y\big\|_2\big)|W|^{-1/2} = g_{\mathbf{x}}(\rho)|W|^{-1/2}$$

for $\rho = (y^T(TT^T)^{-1}y)^{1/2}$. □

An immediate consequence of this lemma is that if $\mathbf{x}$ is uniform in $\mathcal{B}_{\|\cdot\|_2}$, then the vector $\mathbf{y} = T\mathbf{x}$, with T invertible, is uniformly distributed inside the ellipsoid $\mathcal{E}(0, TT^T)$. The subsequent result states the relation between the defining function $g_{\mathbf{y}}(\rho)$ of an ℓ_2^W radial vector, and its ℓ_2^W norm density $f_{\boldsymbol{\rho}}(\rho)$, defined as the pdf of the random variable $\boldsymbol{\rho} = \|\mathbf{y}\|_2^W$. This result is analogous to Lemma 15.1, which was stated for ℓ_p radial densities.

Lemma 15.4 (ℓ_2^W norm density) *Let $\mathbf{y} \in \mathbb{R}^n$ be ℓ_2^W radial with defining function $g_{\mathbf{y}}(\rho)$, where $\rho = \|y\|_2^W$. Then, the pdf of $\boldsymbol{\rho}$ is called the ℓ_2^W norm density and is given by*

$$f_{\boldsymbol{\rho}}(\rho) = |W|^{1/2}\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_2})n\rho^{n-1}g_{\mathbf{y}}(\rho). \tag{15.16}$$

Proof We first notice that any ℓ_2^W radial vector $\mathbf{y}$ can be factored as $\mathbf{y} = T\mathbf{r}\mathbf{u}$, where T is such that $W = TT^T$, $\boldsymbol{\rho} > 0$, and $\|\mathbf{u}\|_2 = 1$. We then compute the joint pdf in the new variables $\mathbf{u}$, $\boldsymbol{\rho}$, in a way similar to Theorem 15.1

$$f_{\mathbf{u},\boldsymbol{\rho}}(u_1, \ldots, u_{n-1}, \rho) = 2f_{\mathbf{y}}(y)J(y \to u, \rho).$$

Now, introducing the slack variable $x = \rho u$, and applying the chain rule for the computation of Jacobians, see Rule A.1 in Appendix A.2, we have that

$$J(y \to u, \rho) = J(y \to x)J(x \to u, \rho).$$

The first factor is equal to $J(y \to x) = |T| = |W|^{1/2}$, since $y = Tx$. The second factor has been computed in Theorem 15.1, and is equal to

$$J(x \to u, \rho) = \rho^{n-1}\psi^{-1/2}(u_1, \ldots, u_{n-1})$$

where ψ is defined in (15.4). Therefore

$$f_{\mathbf{u},\mathbf{r}}(u_1, \ldots, u_{n-1}, \rho) = 2|W|^{1/2}\rho^{n-1}g_{\mathbf{y}}(\rho)\psi^{-1/2}(u_1, \ldots, u_{n-1})$$

where $\rho = \|y\|_2^W$. Integrating over $u_1, \ldots, u_{n-1}$, and using (15.7), we obtain

$$f_{\boldsymbol{\rho}}(\rho) = 2|W|^{1/2}\frac{\pi^{n/2}}{\Gamma(n/2)}\rho^{n-1}g_{\mathbf{y}}(\rho).$$

The statement of the lemma then follows noticing that

$$\frac{2\pi^{n/2}}{\Gamma(n/2)} = n\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_2}).$$

□

We now extend the previous results to the case when the transformation matrix T is rectangular. In particular, we address the problem of determining the distribution on the image of a linear transformation, when the distribution on the domain is uniform. More precisely, given a random variable $\mathbf{x} \in \mathbb{R}^n \sim \mathcal{U}_{\mathcal{B}_{\|\cdot\|_2}}$, and given a full-rank matrix $T \in \mathbb{R}^{m,n}$, $m \leq n$, we derive the distribution of $\mathbf{y} = T\mathbf{x}$. Of course, when $m = n$ and $|T| \neq 0$, the answer follows from Lemma 15.3, i.e. a nonsingular linear mapping transforms uniform distributions into uniform distributions. The general case when $m < n$ is addressed in the next lemma.

Lemma 15.5 *Let $\mathbf{x} \in \mathbb{R}^n$ be ℓ_2 radial with defining function $g_{\mathbf{x}}(\|x\|_2)$, and let $T \in \mathbb{R}^{m,n}$ be full-rank, with $m < n$. Then, the random vector $\mathbf{y} = T\mathbf{x}$ is ℓ_2^W radial with $W = TT^T$, and in particular the pdf $f_{\mathbf{y}}(y)$ is given by*

$$f_{\mathbf{y}}(y) = g_{\mathbf{y}}(\rho) = |\Sigma|^{-1}\text{Surf}\big(\mathcal{B}_{\|\cdot\|_2}\big(1, \mathbb{R}^{n-m}\big)\big) \int_0^\infty g_{\mathbf{x}}\big(\big(\widetilde{\rho}^2 + \rho^2\big)^{1/2}\big)\widetilde{\rho}^{n-m-1}\mathrm{d}\widetilde{\rho}, \quad \rho = \|y\|_2^W \tag{15.17}$$

where Σ is a diagonal matrix containing the singular values of T.

Proof Consider the singular value decomposition $T = U\Sigma V^T$, where $U \in \mathbb{R}^{m,m}$, $V \in \mathbb{R}^{n,m}$ are orthogonal and $\Sigma \in \mathbb{R}^{m,m}$ is the diagonal matrix of the singular values of T. Take $\widetilde{V} \in \mathbb{R}^{n,n-m}$, such that $\widetilde{V}^T\widetilde{V} = I_{n-m}$, $V^T\widetilde{V} = 0_{m,n-m}$, and define

$$\widetilde{T} = \begin{bmatrix} T \\ \widetilde{V}^T \end{bmatrix} \in \mathbb{R}^{n,n}.$$

Then, $\widetilde{T}$ is invertible, and $\widetilde{T}^{-1} = [V\Sigma^{-1}U^T \ \widetilde{V}]$, $|\widetilde{T}| = |\Sigma|$. Next, consider the change of variables $w = \widetilde{T}x$, where $w^T = [y^T \ \widetilde{y}^T]$. Hence, it follows that

$$f_{\mathbf{w}}(w) \doteq f_{\mathbf{y},\widetilde{\mathbf{y}}}(y, \widetilde{y}) = f_{\mathbf{x}}\big(\widetilde{T}^{-1}w\big)J(x \to w) = g_{\mathbf{x}}\big(\big\|\widetilde{T}^{-1}w\big\|_2\big)|\widetilde{T}|^{-1}.$$

Since $\|\widetilde{T}^{-1}w\|_2^2 = \widetilde{y}^T\widetilde{y} + y^T(TT^T)^{-1}y$, we have that

$$f_{\mathbf{y},\widetilde{\mathbf{y}}}(y, \widetilde{y}) = g_{\mathbf{x}}\big(\big(\widetilde{\rho}^2 + \rho^2\big)^{1/2}\big)|\Sigma|^{-1}$$

where $\widetilde{\rho} = \|\widetilde{y}\|_2$, and $\rho = \|y\|_2^W$, with $W = TT^T$. The marginal density $f_{\mathbf{y}}(y)$ can be derived by integrating $f_{\mathbf{y},\widetilde{\mathbf{y}}}(y, \widetilde{y})$ over $\widetilde{y}$. This integration can be performed using a radial element of volume, obtaining

$$f_{\mathbf{y}}(y) = |\Sigma|^{-1}\text{Surf}\big(\mathcal{B}_{\|\cdot\|_2}\big(\mathbb{R}^{n-m}\big)\big)\int_0^\infty g_{\mathbf{x}}\big(\big(\widetilde{\rho}^2 + \rho^2\big)^{1/2}\big)\widetilde{\rho}^{n-m-1}\mathrm{d}\widetilde{\rho} = g_{\mathbf{y}}(\rho)$$

which proves that $\mathbf{y}$ is ℓ_2^W radial with $W = TT^T$. □

We now explicitly determine the distribution of $\mathbf{y}$, under the assumption that $\mathbf{x}$ is uniformly distributed in the unit ball.

Theorem 15.3 *Let $\mathbf{x} \in \mathbb{R}^n \sim \mathcal{U}_{\mathcal{B}_{\|\cdot\|_2}}$ and let $T \in \mathbb{R}^{m,n}$ be full-rank, with $m < n$. Then, the pdf of the random vector $\mathbf{y} = T\mathbf{x}$ is given by*

$$f_{\mathbf{y}}(y) = |W|^{-1/2} \frac{\Gamma(n/2+1)}{\Gamma^m(1/2)\Gamma((n-m)/2+1)} \left(1-\rho^2\right)^{(n-m)/2} \tag{15.18}$$

where $\rho = \|y\|_2^W$, with $W = TT^T$. Moreover, the ℓ_2^W norm density of $\mathbf{y}$ is given by

$$f_{\boldsymbol{\rho}}(\rho) = \frac{2\Gamma(n/2+1)}{\Gamma(m/2)\Gamma((n-m)/2+1)} \rho^{m-1} \left(1-\rho^2\right)^{(n-m)/2}. \tag{15.19}$$

Proof Using (15.17) of Lemma 15.5, we first observe that $|\Sigma| = |TT^T|^{1/2}$. Then, we introduce the change of variables $s = (\widetilde{\rho}^2 + \rho^2)^{1/2}$, from which $\mathrm{d}\widetilde{\rho} = (s^2 - \rho^2)^{-1/2} s \mathrm{d}s$, and the integral in (15.17) becomes

$$g_{\mathbf{y}}(\rho) = \left|TT^T\right|^{-1/2} \mathrm{Surf}\left(\mathcal{B}_{\|\cdot\|_2}\left(\mathbb{R}^{n-m}\right)\right) \int_{\rho}^{\infty} g_{\mathbf{x}}(s) \left(s^2-\rho^2\right)^{(n-m)/2-1} s \, \mathrm{d}s.$$

Since $\mathbf{x}$ is uniform in $\mathcal{B}_{\|\cdot\|_2}$, we have that

$$g_{\mathbf{x}}(s) = \begin{cases} \frac{1}{\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_2}(\mathbb{R}^n))} & \text{if } s < 1; \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, $g_{\mathbf{y}}(\rho)$ can be written as

$$g_{\mathbf{y}}(\rho) = \left|TT^T\right|^{-1/2} \frac{\mathrm{Surf}(\mathcal{B}_{\|\cdot\|_2}(\mathbb{R}^{n-m}))}{\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_2}(\mathbb{R}^n))} \int_{\rho}^{1} \left(s^2-\rho^2\right)^{(n-m)/2-1} s \, \mathrm{d}s.$$

Using the fact that

$$\int \left(s^2-\rho^2\right)^{q+1/2} s \, \mathrm{d}s = \frac{(s^2-\rho^2)^{q+3/2}}{2q+3}$$

with $q = (n-m)/2 - 3/2$, and substituting the values for $\mathrm{Surf}(\mathcal{B}_{\|\cdot\|_2}(\mathbb{R}^{n-m}))$ and $\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_2}(\mathbb{R}^n))$, we obtain $g_{\mathbf{y}}(\rho)$ as stated in (15.18). Now, substituting the expression of $g_{\mathbf{y}}(\rho)$ into (15.16), we finally derive the ℓ_2^W norm density as given in (15.19). □

Remark 15.6 (Rectangular transformation of uniform densities) An important consequence of Theorem 15.3 is that if a uniform distribution is transformed by a rectangular full-rank linear mapping, then the resulting image density is no longer uniform. A linear (rectangular) transformation therefore changes the nature of the uniform distribution on a ball. In particular, the transformed vector $\mathbf{y} = T\mathbf{x}$ tends to concentrate towards the center of the image of the support set (the ellipsoid $\mathcal{E}(0, TT^T)$), rather than to its surface, see Fig. 15.1 for an illustration of this phenomenon. An extension of Lemma 15.5 and Theorem 15.3 to the complex case is reported in [80].

Remark 15.7 (Probabilistic predictors) The result of Theorem 15.3 has been proved in [327], using an alternative derivation. This result is then exploited to determine probabilistic confidence ellipsoids for random vectors, and to construct *probabilistic*

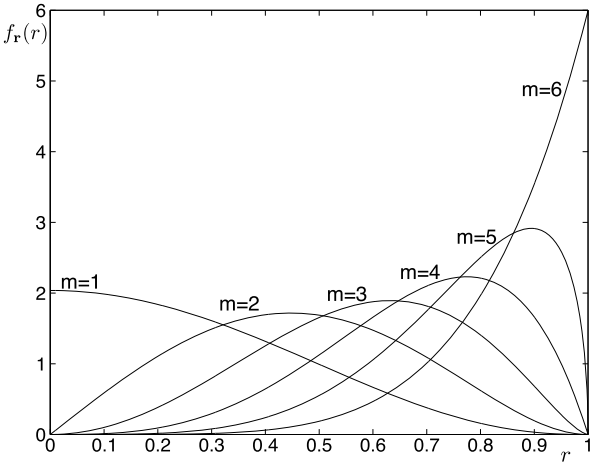

Fig. 15.1 ℓ_2^W norm density of $\mathbf{y} = T\mathbf{x}$, when $\mathbf{x} \sim \mathcal{U}_{\mathcal{B}_{\|\cdot\|_2}}$ and $T \in \mathbb{R}^{m,n}$ is full-rank. The plot shows the case $n = 6$ and $m = 1, \ldots, 6$

predictors of certain sets in $\mathbb{R}^n$. The predictors are subsequently applied to various systems and control problems.

Chapter 16
Vector Randomization Methods

In this chapter we address the issue of real and complex vector randomization in ℓ_p norm balls, according to the uniform distribution. We present efficient algorithms based upon the theoretical developments of Chap. 15 regarding the statistical properties of random vectors. The presentation is partly based on the results reported in [82]. We recall that the uniform density in the ℓ_p norm ball $\mathcal{B}_{\|\cdot\|_p}(r)$ is a special case of the more general ℓ_p radial densities. Hence, random vector generation for ℓ_p radial densities is an immediate extension of uniform generation and follows from the application of Corollary 15.2.

We also observe that the algorithms for vector generation presented in this chapter are based on simple algebraic transformations on samples obtained from the *univariate* generalized Gamma density $\overline{G}_{a,c}$ defined in (2.13). If a generator for $\overline{G}_{a,c}$ is available (such as that presented in Example 14.4), uniform generation in ℓ_p norm balls can be readily performed. We remark that these methods are non-asymptotic, contrary to the techniques introduced in Sect. 14.4. Therefore, the methods discussed in this chapter can be implemented on parallel and distributed architectures, see e.g. [167].

16.1 Rejection Methods for Uniform Vector Generation

We first discuss an application of the rejection technique for uniform generation of n-dimensional vector samples in ℓ_p norm balls. As already observed in Sect. 14.3.1, we expect these methods to be inefficient for large n, due to the curse of dimensionality. These rejection techniques are discussed here mainly to motivate the need for more efficient methods such as those presented in Sect. 16.3 for real vectors and in Sect. 16.4 for complex vectors.

To show the application of the rejection technique, we first construct an outer bounding set for the ball $\mathcal{B}_{\|\cdot\|_p}$. Notice that the norm inequality

$$\|x\|_{p_2} \geq \|x\|_{p_1}, \quad \text{for } p_2 > p_1$$

implies the set inclusion $\mathcal{B}_{\|\cdot\|_{p_1}} \subseteq \mathcal{B}_{\|\cdot\|_{p_2}}$, for $p_2 > p_1$.

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_16,

Table 16.1 Rejection rates for generating samples uniformly in $\mathcal{B}_{\|\cdot\|_p}(\mathbb{R}^n)$ from uniform samples in $\mathcal{B}_{\|\cdot\|_\infty}(\mathbb{R}^n)$

	$p=1$	$p=1.5$	$p=2$
$n=2$	2	1.4610	1.2732
$n=3$	6	2.7185	1.9099
$n=4$	24	6.0412	3.2423
$n=5$	120	15.446	6.0793
$n=10$	3.62×10^{6}	7.21×10^{3}	401.54
$n=20$	2.43×10^{18}	1.15×10^{11}	4.06×10^{7}
$n=30$	2.65×10^{32}	5.24×10^{19}	4.90×10^{13}

Assuming that generation in $\mathcal{B}_{\|\cdot\|_{p_2}}$ is easier to perform than generation in $\mathcal{B}_{\|\cdot\|_{p_1}}$ (see Example 16.1), the idea is therefore to generate samples uniformly in the outer bounding set $\mathcal{B}_{\|\cdot\|_{p_2}}$ and then to reject those which fall outside the set $\mathcal{B}_{\|\cdot\|_{p_1}}$. The rejection rate of such an algorithm is equal to the ratio of volumes, see (14.10), and is given by

$$\eta(n) = \frac{\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_{p_2}}(\mathbb{F}^n))}{\mathrm{Vol}(\mathcal{B}_{\|\cdot\|_{p_1}}(\mathbb{F}^n))}.$$

In the subsequent examples, we compute rejection rates for real and complex balls.

Example 16.1 (Rejection method for real random vectors) Observe that the norm ball $\mathcal{B}_{\|\cdot\|_{p_2}}(\mathbb{R}^n)$ with $p_2=\infty$ contains all the other norm balls. Clearly, generation of uniform samples in $\mathcal{B}_{\|\cdot\|_\infty}(\mathbb{R}^n)$ is straightforward, since

$$\mathcal{B}_{\|\cdot\|_\infty}\big(\mathbb{R}^n\big) = \big\{x\in\mathbb{R}^n : \|x\|_\infty \le 1\big\}$$

is an n-dimensional hypercube whose edges have length equal to two. Hence, a random vector $\mathbf{x}\in\mathbb{R}^n$ uniformly distributed in $\mathcal{B}_{\|\cdot\|_\infty}$ can be immediately obtained generating independently the n components of $\mathbf{x}$, each uniform in the interval $[-1,\ 1]$. Using the volume formulae (15.8), the rejection rate of an algorithm for uniform generation in $\mathcal{B}_{\|\cdot\|_p}(\mathbb{R}^n)$ based on samples in $\mathcal{B}_{\|\cdot\|_\infty}(\mathbb{R}^n)$ is given by

$$\eta(n) = \frac{\Gamma(n/p+1)}{\Gamma^n(1/p+1)}.$$

Table 16.1 presents the rejection rate $\eta(n)$ for different values of n and p.

The values of the rejection rates reported in this table clearly show that this rejection technique may be useful only for small values of n, whereas it becomes extremely inefficient for large values of n.

Example 16.2 (Rejection method for complex random vectors) As with the previous case, a rejection method may be based on the complex hypercube

$$\big\{x\in\mathbb{C}^n : \|x\|_\infty \le 1\big\}.$$

A random vector $\mathbf{x}$ uniformly distributed in this set is obtained by generating independently each component $\mathbf{x}_i$ uniformly in the complex disk of radius one. The

Table 16.2 Rejection rates for generating samples uniformly in $\mathcal{B}_{\|\cdot\|_p}(\mathbb{C}^n)$ from uniform samples in $\mathcal{B}_{\|\cdot\|_\infty}(\mathbb{C}^n)$

	$p=1$	$p=1.5$	$p=2$
$n=2$	6	2.8302	2
$n=3$	90	14.219	6
$n=4$	2.52×10^{3}	106.50	24
$n=5$	1.13×10^{5}	1.08×10^{3}	120
$n=10$	2.38×10^{15}	2.60×10^{9}	3.63×10^{6}
$n=20$	7.78×10^{41}	1.10×10^{26}	2.43×10^{18}
$n=30$	7.75×10^{72}	4.35×10^{45}	2.65×10^{32}

rejection rate of this method can be easily computed by means of the formula for the volume (15.13), obtaining

$$\eta(n)=\frac{\Gamma(2n/p+1)}{\Gamma^n(2/p+1)}.$$

Table 16.2 reports several values of the rejection rate, showing the extreme inefficiency of this method for large values of n.

16.2 Generalized Gamma Density

The main technical tool used in the algorithms for random vector generation presented in this chapter is the generalized Gamma density, defined in (2.13) as

$$\overline{G}_{a,c}(x)=\frac{c}{\Gamma(a)}x^{ca-1}\mathrm{e}^{-x^c},\quad x\geq 0.$$

We notice that $\overline{G}_{a,c}(x)$ is a *unilateral* density, since $x\geq 0$. A *bilateral* generalized Gamma density is defined accordingly as

$$f_{\mathbf{x}}(x)=\frac{c}{2\Gamma(a)}|x|^{ca-1}\mathrm{e}^{-|x|^c}.$$

We recall that the generalized Gamma density coincides with classical density functions, such as Gaussian and Laplace, for a specific choice of the parameters a and c. This is further discussed in the next example.

Example 16.3 (Generalized Gamma density) We illustrate the density $\overline{G}_{a,c}$ for specific values of a, c. First, we set $a=1/p$ and $c=p$. Taking $p=1$, we obtain the unilateral Laplace density with maximum value equal to one

$$f_{\mathbf{x}}(x)=\mathrm{e}^{-x},\quad x\geq 0. \tag{16.1}$$

For $p=2$ we have the unilateral normal density with mean value zero and variance equal to $1/2$

$$f_{\mathbf{x}}(x)=\frac{2}{\sqrt{\pi}}\mathrm{e}^{-x^2},\quad x\geq 0.$$

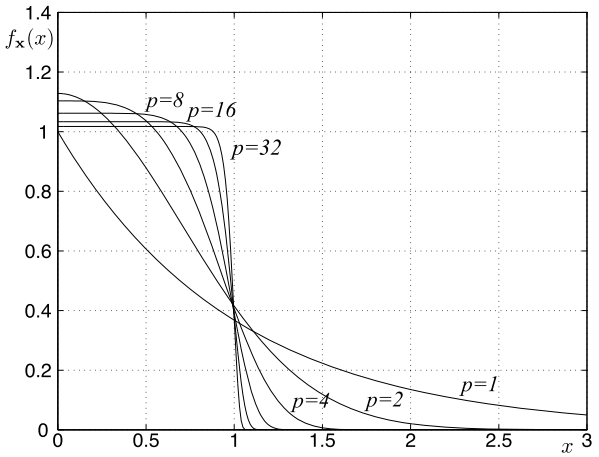

Fig. 16.1 Generalized Gamma density $\overline{G}_{1/p,p}$ for different values of p

In addition, it can be shown that

$$\lim_{p\to\infty} \frac{p}{\Gamma(1/p)} \mathrm{e}^{-x^p} = \begin{cases} 1 & \text{if } x \in (0,1); \\ 0 & \text{if } x > 1. \end{cases}$$

Thus, the generalized Gamma density approaches the uniform density on $[0, 1]$ for large values of p. Figure 16.1 shows a plot of the unilateral generalized Gamma density for various values of p.

To conclude this example, we discuss the relation between $\overline{G}_{a,c}$ and the unilateral Gamma density $G_{a,b}$. The unilateral Gamma density, see (2.12), is

$$G_{a,b}(x) = \frac{1}{\Gamma(a) b^a} x^{a-1} \mathrm{e}^{-x/b}, \quad x \geq 0.$$

In Example 14.4 it is shown that a random variable $\mathbf{x} \sim \overline{G}_{a,c}$ is obtained from a random variable with Gamma density $\mathbf{z} \sim G_{a,1}$ by means of the change of variables $\mathbf{x} = \mathbf{z}^{1/c}$. Hence, it is straightforward to generate samples according to $\overline{G}_{a,c}$ if a random generator for $G_{a,b}$ is available.

Finally, we remark that samples distributed according to a bilateral univariate density $\mathbf{x} \sim f_{\mathbf{x}}(x)$ can be immediately obtained from a unilateral univariate density $\mathbf{z} \sim f_{\mathbf{z}}(z)$ taking $\mathbf{x} = \mathbf{sz}$, where $\mathbf{s}$ is an independent random sign that takes the values ± 1 with equal probability.

16.3 Uniform Sample Generation of Real Vectors

In this section we present an efficient algorithm for uniform generation of samples in $\mathcal{B}_{\|\cdot\|_p}(r, \mathbb{R}^n)$, based on the results of Corollary 15.3. Let each component of $\mathbf{x}$ be

independently distributed according to the (bilateral) generalized Gamma density with parameters $1/p$, p. Then, the joint density $f_{\mathbf{x}}(x)$ is

$$f_{\mathbf{x}}(x) = \prod_{i=1}^{n} \frac{p}{2\Gamma(1/p)} \mathrm{e}^{-|x_i|^p} = \frac{p^n}{2^n \Gamma^n(1/p)} \mathrm{e}^{-\|x\|_p^p}.$$

Hence, it follows immediately that $f_{\mathbf{x}}(x)$ is ℓ_p radial with defining function

$$g(\rho) = \frac{p^n}{2^n \Gamma^n(1/p)} \mathrm{e}^{-\rho^p}, \quad \rho = \|x\|_p.$$

Further, we have from Corollary 15.3 that if $\mathbf{w}$ is uniform in $[0, 1]$ then $\mathbf{y} = r\mathbf{w}^{1/n}\mathbf{x}/\|\mathbf{x}\|_p$ is uniform in $\mathcal{B}_{\|\cdot\|_p}(r, \mathbb{R}^n)$.

The algorithm for uniform sample generation in real ℓ_p balls is summarized next.

Algorithm 16.1 (Uniform generation in real ℓ_p norm ball) *Given n, p and r, this algorithm returns a real random vector $\mathbf{y}$ uniformly distributed in $\mathcal{B}_{\|\cdot\|_p}(r, \mathbb{R}^n)$.*

1. Generate n independent random real scalars $\boldsymbol{\xi}_i \sim \overline{G}_{1/p,p}$;
2. Construct the vector $\mathbf{x} \in \mathbb{R}^n$ of components $\mathbf{x}_i = \mathbf{s}_i \boldsymbol{\xi}_i$, where $\mathbf{s}_i$ are independent random signs;
3. Generate $\mathbf{z} = \mathbf{w}^{1/n}$, where $\mathbf{w}$ is uniform in $[0, 1]$;
4. Return $\mathbf{y} = r\mathbf{z}\frac{\mathbf{x}}{\|\mathbf{x}\|_p}$.

We remark that this algorithm is an extension of the method proposed in [200, 292] for generating random points uniformly distributed on n-dimensional spheres starting from normally distributed real vectors.

Example 16.4 (Uniform generation in $\mathcal{B}_{\|\cdot\|_p}(\mathbb{R}^2)$) For illustrative purposes, in this example we consider the case $n = 2$, $p = 1.5$ and $r = 1$. Figure 16.2 shows the three steps involved in uniform generation:

1. Each vector sample $\mathbf{x}^{(i)}$ is generated according to a generalized Gamma density with parameters $1/p$, p;
2. Each sample is normalized taking $\mathbf{x}^{(i)}/\|\mathbf{x}^{(i)}\|_p$, obtaining a uniform distribution on the contour $\partial\mathcal{B}_{\|\cdot\|_p}$;
3. Each normalized sample is scaled by the *volumetric factor* $\sqrt{\mathbf{w}^{(i)}}$, where $\mathbf{w}^{(i)} \sim \mathcal{U}_{[0,1]}$, which smudges the samples uniformly inside $\mathcal{B}_{\|\cdot\|_p}$.

We remark that Algorithm 16.1 can be also used when $p \in (0, 1)$. However, in this case $\|x\|_p^p = \sum_{i=1}^n |x_i|^p$ is not a norm and the set

$$\mathcal{B}_{\|\cdot\|_p}(r) = \left\{x \in \mathbb{R}^n : \|x\|_p \leq r\right\}$$

is not convex. Figure 16.3 shows 1,000 samples of real two-dimensional vectors uniformly distributed in $\mathcal{B}_{\|\cdot\|_p}$ for $p = 0.7$ and $p = 1$.

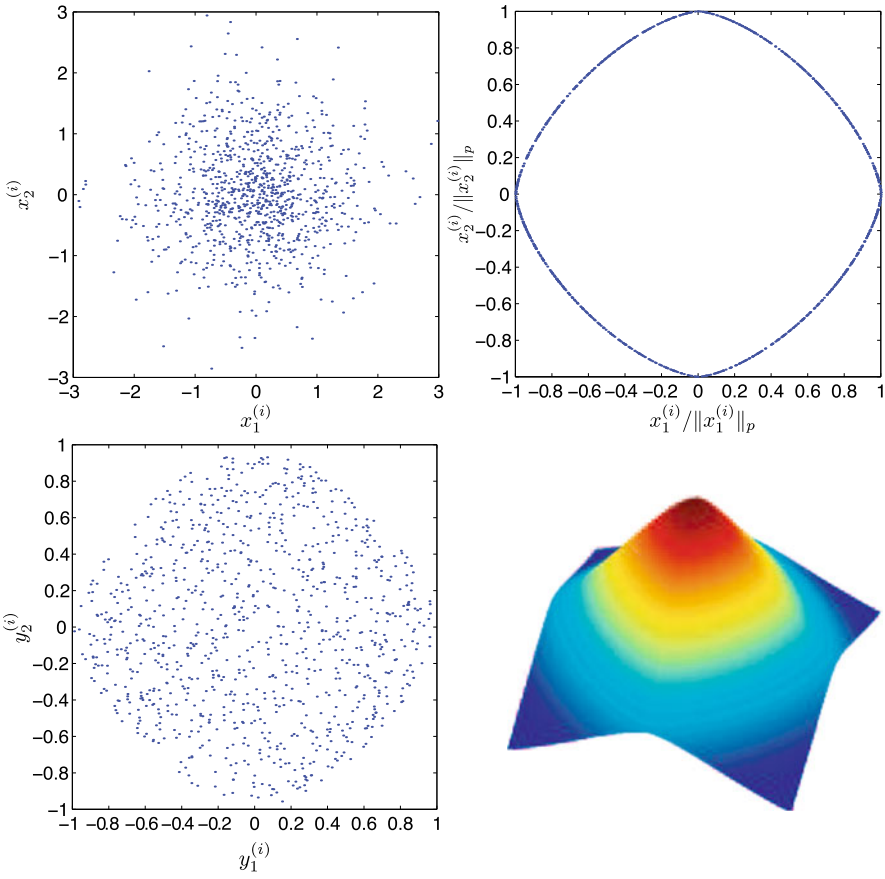

Fig. 16.2 Generation of 1,000 uniform samples in $\mathcal{B}_{\|\cdot\|_p}$, for $p = 1.5$

Next, we discuss sample generation within an ellipsoid

$$\mathcal{E}(\bar{x}, W) = \left\{x \in \mathbb{R}^n : x^T W^{-1} x \leq 1\right\}, \quad W \succ 0,$$

and show that it can be easily performed using the techniques discussed in Sect. 15.5. This is described in the following algorithm.

Algorithm 16.2 (Uniform generation in an ellipsoid) *Given* $n, \bar{x}, W \succ 0$, *this algorithm returns a random vector* $\mathbf{y} \in \mathbb{R}^n$ *uniformly distributed in the ellipsoid* $\mathcal{E}(\bar{x}, W)$.

1. Compute a matrix T such that $W = TT^T$;
2. Generate a random vector $\mathbf{x} \sim \mathcal{U}_{\mathcal{B}_{\|\cdot\|_2}}$ using Algorithm 16.1;
3. Return $\mathbf{y} = T\mathbf{x} + \bar{x}$.

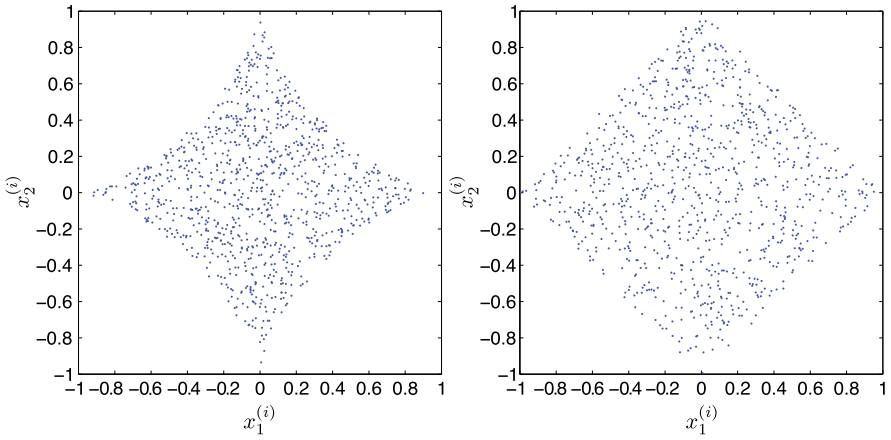

Fig. 16.3 Generation of 1,000 uniform samples in $\mathcal{B}_{\|\cdot\|_p}$, for $p = 0.7$ (*left*) and $p = 1$ (*right*)

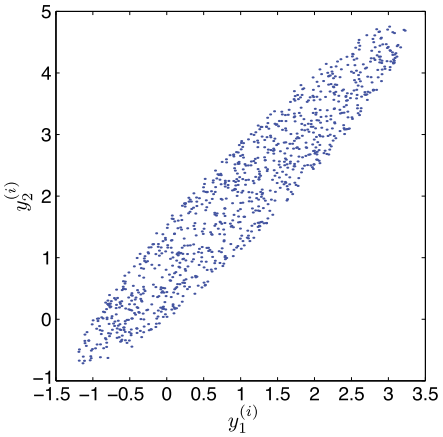

Fig. 16.4 Uniform generation in the ellipse (16.2)

Example 16.5 (Uniform generation in an ellipse) Consider the ellipse

$$\mathcal{E}(\bar{x}, W) = \left\{x \in \mathbb{R}^2 : x^T W^{-1} x \leq 1\right\} \tag{16.2}$$

where

$$\bar{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \qquad W = \begin{bmatrix} 5 & 6 \\ 6 & 8 \end{bmatrix}.$$

Figure 16.4 shows uniform generation of 1,000 random vector samples inside the ellipse $\mathcal{E}(\bar{x}, W)$.

Example 16.6 (Uniform sample generation in a simplex) We describe in this example a simple method for generating uniform points in the standard simplex, based on the generalized Gamma density $\overline{G}_{1,1}$, which corresponds to the unilateral Laplace

density (16.1). The standard unit simplex, also known as *probability simplex*, is defined as

$$\mathrm{PS}(n) \doteq \left\{ x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1,\ x_i \geq 0,\ i = 1, \ldots, n \right\}.$$

Notice that the probability simplex is the intersection between the nonnegative orthant of $\mathbb{R}^n$ and the surface of the unit ball in the ℓ_1 norm, i.e.

$$\mathrm{PS}(n) = \mathbb{R}^n_+ \cap \mathcal{B}_{\|\cdot\|_1}.$$

As discussed in Remark 15.1, if a random vector $\mathbf{x}$ is ℓ_1 radial, then $\mathbf{y} = \mathbf{x}/\|\mathbf{x}\|_1$ is uniformly distributed on the surface of the unit ℓ_1 norm ball. Therefore, the uniform distribution on the unit simplex can be obtained by projecting the $\mathbf{y}$ samples onto the nonnegative orthant. The explicit algorithm is given next.

Algorithm 16.3 (Uniform generation in the unit simplex) *Given n, this algorithm returns a real random vector $\mathbf{y}$ uniformly distributed in* $\mathrm{PS}(n)$.

1. Generate n independent random real scalars $\mathbf{x}_i \sim \overline{G}_{1,1}$;
2. Return $\mathbf{y} = \frac{\mathbf{x}}{\|\mathbf{x}\|_1}$.

Uniform samples in the probability simplex can be used to generate uniform samples in generic simplices. A generic $(k-1)$-dimensional simplex in $\mathbb{R}^n$ is defined as the convex hull of k affinely independent vectors

$$\mathrm{Simplex}(v_1, \ldots, v_k) = \left\{ \xi = x_1 v_1 + \cdots + x_k v_k,\ x = [x_1\ \cdots\ x_k]^T \in \mathrm{PS}(k) \right\}$$

where $v_i \in \mathbb{R}^n$, $i = 1, \ldots, k$, and $[v_2 - v_1\ \ v_3 - v_1\ \cdots\ v_k - v_1]$ is full-rank. Therefore, defining the simplex vertex matrix $V \doteq [v_1\ \cdots\ v_k] \in \mathbb{R}^{n,k}$, it can be verified that if the random vector $\mathbf{x}$ is uniformly distributed in $\mathrm{PS}(k)$, then the random vector

$$\mathbf{y} = V\mathbf{x}$$

is uniformly distributed in $\mathrm{Simplex}(v_1, \ldots, v_k)$.

16.4 Uniform Sample Generation of Complex Vectors

In this section we present an algorithm based on Corollary 15.3 for generating complex vectors uniformly distributed in ℓ_p norm balls. Let

$$\mathbf{x} = [\mathbf{x}_1\ \cdots\ \mathbf{x}_n]^T \in \mathbb{C}^n$$

be a complex random vector. Clearly, each component $\mathbf{x}_i$ of $\mathbf{x}$ is equivalent to a two-dimensional real vector $\mathbf{z}_i = [\mathrm{Re}(\mathbf{x}_i)\ \ \mathrm{Im}(\mathbf{x}_i)]^T \in \mathbb{R}^2$, so that the absolute value of

$\mathbf{x}_i$ coincides with the ℓ_2 norm of $\mathbf{z}_i$. Let the components $\mathbf{z}_i$ be independent ℓ_2 radial vectors, with defining function

$$f_{\mathbf{z}_i}(z_i) = \frac{p}{2\pi\,\Gamma(2/p)} \mathrm{e}^{-\rho_i^p} = g_i(\rho_i), \quad \rho_i = \|z_i\|_2.$$

Therefore, using (15.14) the corresponding norm density function is

$$f_{\boldsymbol{\rho}_i}(\rho_i) = \frac{p}{\Gamma(2/p)} \rho_i \mathrm{e}^{-\rho_i^p}, \quad \rho_i \geq 0.$$

This density is a (unilateral) generalized Gamma density with parameters $(2/p, p)$. Since the components of $\mathbf{x}$ are independent, the density function $f_{\mathbf{x}}(x)$ is obtained as the (bilateral) joint density function

$$f_{\mathbf{x}}(x) = \prod_{i=1}^{n} \frac{p}{2\pi\,\Gamma(2/p)} \mathrm{e}^{-|x_i|^p} = \frac{p^n}{2^n \pi^n \Gamma^n(2/p)} \mathrm{e}^{-\|x\|_p^p}.$$

As in the real case, from this expression it follows that the random vector $\mathbf{x}$ is ℓ_p radial, with defining function

$$g(\rho) = \frac{p^n}{2^n \pi^n \Gamma^n(2/p)} \mathrm{e}^{-\rho^p}, \quad \rho = \|x\|_p$$

and the results of Corollary 15.3 can be applied. The algorithm is given next.

Algorithm 16.4 (Uniform generation in complex ℓ_p norm ball) *Given n, p and r, this algorithm returns a complex random vector $\mathbf{y}$ uniformly distributed in $\mathcal{B}_{\|\cdot\|_p}(r, \mathbb{C}^n)$.*

1. Generate n independent random complex numbers $\mathbf{s}_i = \mathrm{e}^{j\boldsymbol{\theta}}$, where $\boldsymbol{\theta}$ is uniform in $[0, 2\pi]$;
2. Construct the vector $\mathbf{x} \in \mathbb{C}^n$ of components $\mathbf{x}_i = \mathbf{s}_i \boldsymbol{\xi}_i$, where the $\boldsymbol{\xi}_i$ are independent random variables $\boldsymbol{\xi}_i \sim \bar{G}_{2/p,p}$;
3. Generate $\mathbf{z} = \mathbf{w}^{1/(2n)}$, where $\mathbf{w}$ is uniform in $[0, 1]$;
4. Return $\mathbf{y} = r\mathbf{z}\frac{\mathbf{x}}{\|\mathbf{x}\|_p}$.

16.5 Uniform Generation of Stable Polynomials

In this section we briefly review techniques for generating *stable* polynomials of given degree. The analysis is carried out for discrete-time polynomials, since most results are specific to this class: the interested reader is referred to [354] for a more detailed review of these topics.

We recall that the discrete-time monic polynomial of degree n

$$p(z) = q_0 + q_1 z + \cdots + q_{n-1} z^{n-1} + z^n \tag{16.3}$$

with real coefficients $q \in \mathbb{R}^n$ is said to be Schur stable if all its roots $z_k, k = 1, \ldots, n$, lie in the open unit disk on the complex plane. We are interested in generating uniform samples of q in the Schur region defined in (6.7) as follows

$$\mathcal{S}_n = \left\{ q \in \mathbb{R}^n : p(z, q) \text{ Schur} \right\}. \tag{16.4}$$

Note that the Schur region for monic polynomials is bounded (contrary to the Hurwitz region), and this fact facilitates the design of efficient generation schemes. Moreover, in many applications of randomized algorithms to system analysis problems, it is of importance to obtain samples *uniformly* distributed in the coefficient space. Indeed, randomly generated Schur polynomials can be used to represent stable uncertain dynamics affecting the system.

In [354] various techniques for generating random samples inside $\mathcal{S}_n$ are discussed, including rejection, direct root generation, random walks and the so-called *backward discrete Hermite–Biehler* (DHB) method. However, none of these techniques leads to uniform samples of $\mathbf{q} \in \mathcal{S}_n$. To this regard, the most interesting generation scheme discussed in [354] is based on the following parameterization

Lemma 16.1 (Levinson–Durbin parameterization) *Given a polynomial $p(z)$ of the form* (16.3), *define the following reverse-order polynomial*

$$\overline{p}_n(z) \doteq z^n q\left(z^{-1}\right) = q_0 z^n + q_1 z^{n-1} + \cdots + q_{n-1} z + 1. \tag{16.5}$$

Then, any monic Schur stable polynomial[1] $p_n(z)$ *of degree n can be obtained via the following recursion*:

$$\begin{aligned} &p_0(z) = 1, \\ &p_k(z) = z p_{k-1}(z) + t_k \overline{p}_{k-1}(z), \quad |t_k| < 1,\ k = 1, \ldots, n. \end{aligned} \tag{16.6}$$

In the control literature, this parametrization has been introduced in [160], where the coefficients t_k, $k = 1, \ldots, n$, are referred to as *canonical parameters*, and has been later exploited in [308, 309] to derive results on robust stability analysis and design. However, the recursion (16.6) has been known in the signal processing community much earlier, at least since the works of Durbin [150] and Levinson [262].

The lemma states that sweeping the unit cube $\mathcal{B}_{\|\cdot\|_\infty}(\mathbb{R}^n)$ in the space of these parameters yields all stable monic polynomials of degree n, and it can be easily seen that this mapping is one-to-one. Moreover, all stable polynomials of *all* degrees up to n are generated. Hence, Schur stable polynomials can be immediately obtained by generating random samples $\mathbf{t} \in \mathcal{B}_{\|\cdot\|_\infty}(\mathbb{R}^n)$. Clearly, various probability distributions for the coefficients $\mathbf{t}$ lead to different coefficient distributions. In particular, since there exists a one-to-one mapping between the parameters $\mathbf{t}$ and the polynomial coefficients $\mathbf{q}$, we can explicitly determine what density function should be adopted for the $\mathbf{t}$'s in order to obtain a uniform density over $\mathcal{S}_n$. Interestingly, this idea was considered for the first time in the field of signal processing: in particular, in the work [47] the authors explicitly compute the Jacobian of the reverse mapping between $\mathbf{q}$ and $\mathbf{t}$, deriving a recursive expression as reported in the next lemma.

[1] Here, $p_k(z)$ denotes a polynomial of degree k obtained at the kth step of the recursion.

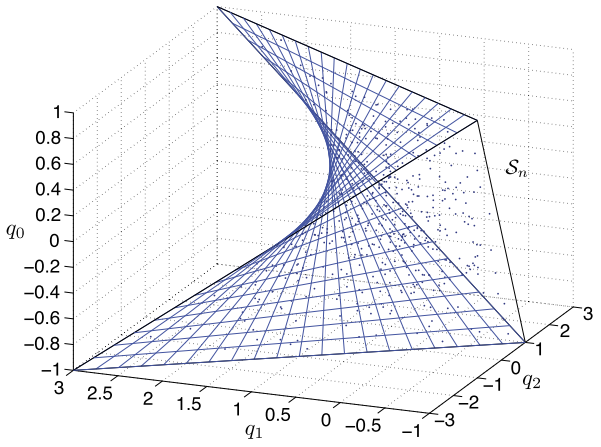

Fig. 16.5 Uniform Schur polynomials of degree three

Lemma 16.2 *If* $\mathbf{t}_1$ *is uniform over the interval* $(0, 1)$ *and, for* $k = 2, \ldots, n$, $\mathbf{t}_k$ *has pdf* $f_{\mathbf{t}_k}(t_k)$ *proportional to*

$$\left|J_k(t_k)\right| = \left|\left(t_k + (-1)^k\right) J_{k-1}(t_k)\right|, \quad J_1 \equiv 1 \tag{16.7}$$

then the coefficients of the polynomial (16.6) *are uniform over* $\mathcal{S}_n$.

In the above lemma, $J_k(t_k)$ represents the Jacobian of the transformation induced by the recursion (16.6), and it is itself given in recursive form. Hence, the lemma provides a direct way of generating Schur stable polynomials with coefficients uniformly distributed in the Schur region $\mathcal{S}_n$. We refer to these polynomials shortly as *uniform Schur stable*. This is summarized in Algorithm 16.5. Note that the generation of $\mathbf{t}_k$ can be performed using Algorithm 14.1 discussed in Chap. 14, since $f_{\mathbf{t}_k}(t_k)$ is a univariate polynomial density. Figure 16.5 depicts the coefficients of 300 uniform Schur stable polynomials inside $\mathcal{S}_3$.

Algorithm 16.5 (Uniform generation of Schur polynomials) *Given* n, *this algorithm returns a uniform Schur stable polynomial* $p_n(z)$.

1. Set $t_1 = 1$, $J_1 = 1$ and $p_0(z) = 1$.
2. For $k = 2$ to n
   - ▷ Construct $J_k(t_k)$ using recursion (16.7);
   - ▷ Generate $\mathbf{t}_k$ according to $f_{\mathbf{t}_k}(t_k) \propto |J_k(t_k)|$;
   - ▷ Construct p_k according to (16.6);
3. End for.

Remark 16.1 (Hurwitz Polynomials) Since the Hurwitz region is unbounded, the case of continuous-time polynomials differs significantly from the discrete-time

one. Hence, for this situation uniform generation does not make sense, since the uniform density is not even defined for unbounded sets. One might think of forcing the coefficients to be bounded in, say, given intervals, generate uniformly the coefficient vector inside the bounded rectangular domain, and then use rejection to obtain uniform distribution for Hurwitz polynomials with bounded coefficients. However, it has been shown in [296] that the probability of picking a Hurwitz polynomial rapidly decreases to zero as the degree of the polynomial grows. A possibility is to use the bilinear transformation [175]

$$s = \frac{z+1}{z-1},$$

between the interior of the unit disk on the complex plane and the open left-half plane to map Schur stable polynomials into Hurwitz ones. The reader is referred to [354], where this method is discussed together with other techniques based on backward Routh table.

Finally, we note that the generation of stable polynomials can be seen as a first step towards generating stable random dynamic uncertainties. On this line, the work [374] presents an algorithm for approximately generating uniform samples in the $\mathcal{H}_\infty$ ball. These random samples can be applied for probabilistic model validation, see e.g. [264, 265, 271].

Chapter 17
Statistical Theory of Random Matrices

In this chapter we study the statistical properties of random matrices whose probability density belongs to the class of matrix radial densities. The results presented in this chapter constitute the theoretical foundations of the algorithms for generation of random matrices uniformly distributed in norm bounded sets presented in Chap. 18.

This chapter has the following structure. In Sect. 17.1 we introduce the notion of ℓ_p induced radial matrix density, which is a direct extension of the concept of radial vector densities introduced in Chap. 15. In Sects. 17.2 and 17.3 we state specific results for ℓ_p induced radial densities for the cases $p = 1, \infty$ and $p = 2$ respectively. For the case $p = 2$, we first study symmetric real matrices, and then extend these results to real and complex rectangular matrices. The contents of Sect. 17.2 are based on concepts of multivariate statistical analysis. The reader interested in this topic may refer to [22] for an introductory presentation. More specific material on the theory of random matrices can be found in [153, 154, 186, 281]. Additional material is available in [180, 210]. An overview of random matrix theory for wireless communication systems is available in [392]. The results presented in this chapter are based on [81, 83].

17.1 Radial Matrix Densities

In Chap. 15 we defined a particular class of vector densities which depend only on the norm of the random vector. This definition is now extended to random matrices. We first discuss Hilbert–Schmidt ℓ_p radial densities.

17.1.1 Hilbert–Schmidt ℓ_p Radial Matrix Densities

Definition 17.1 (Hilbert–Schmidt ℓ_p radial matrix densities) A random matrix $\mathbf{X} \in \mathbb{F}^{n,m}$ is radial in the Hilbert–Schmidt ℓ_p norm if its density function can be written as

$$f_{\mathbf{X}}(X) = g(\rho), \quad \rho = \|X\|_p$$

where $g(\rho)$ is the defining function of $\mathbf{X}$.

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_17,

Using the vectorization operator $\mathrm{vec}(\cdot)$ introduced in (3.8), the Hilbert–Schmidt ℓ_p norm of X can be written as

$$\|X\|_p = \big\|\mathrm{vec}(X)\big\|_p.$$

Then, the statistical properties of an ℓ_p radial matrix $\mathbf{X}$ are equivalent to the properties of the ℓ_p radial random vector $\mathbf{x} = \mathrm{vec}(\mathbf{X})$ studied in Chap. 15.

17.1.2 ℓ_p Induced Radial Matrix Densities

In this chapter we mainly concentrate on the properties of the class of random matrices whose densities depend only on the ℓ_p induced norm, and refer to this class as ℓ_p induced radial densities. A formal definition is given below.

Definition 17.2 (ℓ_p induced radial matrix densities) A random matrix $\mathbf{X} \in \mathbb{F}^{n,m}$ is radial in the ℓ_p induced norm if its density function can be written as

$$f_{\mathbf{X}}(X) = g(\rho), \quad \rho = |||X|||_p$$

where $g(\rho)$ is the defining function of $\mathbf{X}$.

Example 17.1 (Uniform matrices in $\mathcal{B}_{|||\cdot|||_p}$ are ℓ_p induced radial) Consider the definition of uniform density in the ℓ_p induced norm unit ball

$$f_{\mathbf{X}}(X) = \mathcal{U}_{\mathcal{B}_{|||\cdot|||_p}}(X) = \begin{cases} \frac{1}{\mathrm{Vol}(\mathcal{B}_{|||\cdot|||_p})} & \text{if } |||X|||_p \le 1; \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to verify that this pdf depends only on the norm of X. That is,

$$f_{\mathbf{X}}(X) = g(\rho), \quad \rho = |||X|||_p$$

where the defining function $g(\rho)$ is given by

$$g(\rho) = \begin{cases} \frac{1}{\mathrm{Vol}(\mathcal{B}_{|||\cdot|||_p})} & \text{if } \rho \le 1; \\ 0 & \text{otherwise.} \end{cases} \tag{17.1}$$

The fact that the uniform density is ℓ_p induced radial turns out to be crucial for the development of efficient algorithms for random matrix generation in ℓ_p induced norm ball presented in Chap. 18.

17.2 Statistical Properties of ℓ_1 and ℓ_∞ Induced Densities

In this section we study ℓ_1 and ℓ_∞ radial densities for real and complex random matrices.

17.2.1 Real Matrices with ℓ_1/ℓ_∞ Induced Densities

The ℓ_1 induced norm of a given matrix $X \in \mathbb{F}^{n,m}$ is equal to the maximum of the ℓ_1 norms of its columns, see (3.10), that is

$$|||X|||_1 = \max_{i=1,\dots,m} \|\xi_i\|_1$$

where $\xi_1, \dots, \xi_m$ are the columns of X. The pdf of an ℓ_p induced radial matrix $\mathbf{X} \in \mathbb{F}^{n,m}$ can therefore be written as

$$f_{\mathbf{X}}(X) = g(\bar{\rho}), \quad \bar{\rho} \doteq \max_{i=1,\dots,m} \rho_i$$

where $\rho_i \doteq \|\xi_i\|_1$, for $i = 1, \dots, m$.

The following theorem defines a decomposition of an ℓ_1 induced radial real matrix $\mathbf{X}$ in two terms: a normalized matrix $\mathbf{U}$ and a diagonal matrix $\mathbf{R}$ containing the norms of the columns, and provides a closed-form expression for their probability densities. This theorem is the counterpart, for ℓ_1 induced radial matrices, of Theorem 15.1.

Theorem 17.1 (ℓ_1 induced radial matrices in $\mathbb{R}^{n,m}$) *Let the random matrix* $\mathbf{X} \in \mathbb{R}^{n,m}$ *be factored as* $\mathbf{UR}$, *where*

$$\mathbf{U} = [\mathbf{u}_1 \ \cdots \ \mathbf{u}_m], \quad \|\mathbf{u}_i\|_1 = 1;$$
$$\mathbf{R} = \operatorname{diag}\big([\boldsymbol{\rho}_1 \ \cdots \ \boldsymbol{\rho}_m]\big), \quad \boldsymbol{\rho}_i > 0$$

being $\mathbf{u}_i \in \mathbb{R}^n$ *the* i*th column of* $\mathbf{U}$. *The following statements are equivalent*:

1. $\mathbf{X}$ *is* ℓ_1 *induced radial with defining function* $g(\bar{\rho})$, $\bar{\rho} = \max_{i=1,\dots,m} \rho_i$;
2. $\mathbf{U}$ *and* $\mathbf{R}$ *are independent*, *and their densities are given by*

$$f_{\mathbf{U}}(U) = \prod_{i=1}^{m} f_{\mathbf{u}_i}(u_i), \quad f_{\mathbf{u}_i}(u_i) = \mathcal{U}_{\partial \mathcal{B}_{\|\cdot\|_1}(\mathbb{R}^n)};$$
$$f_{\mathbf{R}}(R) = \left[\frac{2^n}{(n-1)!}\right]^m g(\bar{\rho}) \prod_{i=1}^{m} \rho_i^{n-1}.$$

Proof The proof follows the same lines as Theorem 15.1. For each column ξ_i of the matrix variable X we write

$$\xi_i = \rho_i u_i, \quad i = 1, \dots, m.$$

Then, we assume $[X]_{n,i} > 0$, $i = 1, \dots, n$, and observe that in this case the transformation from X to U, R is one-to-one. The Jacobian of this transformation is easily computed as

$$J(X \to U, R) = \prod_{i=1}^{m} J(\xi_i \to u_i, \rho_i) = \prod_{i=1}^{m} \rho_i^{n-1}$$

where the last equality follows from Eq. (15.5) with $p = 1$. From Theorem A.1 (see Appendix A.1) on the transformation of random matrices, we write

$$f_{\mathbf{U},\mathbf{R}}(U, R) = 2^m g(\bar{\rho}) \prod_{i=1}^{m} \rho_i^{n-1}$$

where the factor 2^m is introduced to consider all possible combinations of signs in the terms $[X]_{n,i}$, and thus to remove the condition $[X]_{n,i} > 0$. From this equation, it follows that $\mathbf{U}$ and $\mathbf{R}$ are independent, and that the pdf of each $\mathbf{u}_i$ is a constant that can be computed by setting $p = 1$ in Eq. (15.3). The density $f_{\mathbf{R}}(R)$ is then obtained integrating the joint density $f_{\mathbf{U},\mathbf{R}}(U, R)$ with respect to U. □

The following corollary, based on the results of Theorem 17.1, provides a statistical characterization of real matrices uniformly distributed in the norm ball $\mathcal{B}_{|||\cdot|||_1}$.

Corollary 17.1 (Uniform real matrices in $\mathcal{B}_{|||\cdot|||_1}$) *Let* $\mathbf{X} \in \mathbb{R}^{n,m}$ *and let* $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m$ *be the columns of* $\mathbf{X}$. *The following statements are equivalent*:

1. $\mathbf{X}$ *is uniformly distributed in the set* $\mathcal{B}_{|||\cdot|||_1}(\mathbb{R}^{n,m})$;
2. $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m$ *are independent and uniformly distributed in* $\mathcal{B}_{\|\cdot\|_1}(\mathbb{R}^n)$.

Proof The defining function of $\mathbf{X}$ is given by

$$g(\bar{\rho}) = \begin{cases} \frac{1}{\mathrm{Vol}(\mathcal{B}_{|||\cdot|||_1})} & \text{if } \bar{\rho} \leq 1; \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{\rho} = \max_{i=1,\ldots,m} \rho_i$ and $\rho_i = \|\xi_i\|_1$, for $i = 1, \ldots, m$.

From Theorem 17.1, we immediately obtain

$$f_{\mathbf{U}}(U) = \prod_{i=1}^{m} f_{\mathbf{u}_i}(u_i), \quad f_{\mathbf{u}_i}(u_i) = \mathcal{U}_{\partial\mathcal{B}_{\|\cdot\|_1}(\mathbb{R}^n)};$$

$$f_{\mathbf{R}}(R) = \left[\frac{2^n}{(n-1)!}\right]^m \frac{1}{\mathrm{Vol}(\mathcal{B}_{|||\cdot|||_1})} \prod_{i=1}^{m} \rho_i^{n-1}, \quad \bar{\rho} \leq 1.$$

Since $f_{\mathbf{R}}(R)$ is a density function, we impose

$$\int f_{\mathbf{R}}(R)\,\mathrm{d}R = \int_{\bar{\rho}\leq 1} \left[\frac{2^n}{(n-1)!}\right]^m \frac{1}{\mathrm{Vol}(\mathcal{B}_{|||\cdot|||_1})} \prod_{i=1}^{m} \rho_i^{n-1}\,\mathrm{d}\rho_1 \cdots \mathrm{d}\rho_m = 1$$

obtaining

$$\mathrm{Vol}(\mathcal{B}_{|||\cdot|||_1}) = \left[\frac{2^n}{n!}\right]^m.$$

Moreover, substituting this expression in $f_{\mathbf{R}}(R)$, we have

$$f_{\mathbf{R}}(R) = \prod_{i=1}^{m} n\rho_i^{n-1}, \quad \bar{\rho} \leq 1.$$

The joint density of $\mathbf{R}$, $\mathbf{U}$ is then given by

$$f_{\mathbf{U},\mathbf{R}}(U,R) = f_{\mathbf{R}}(R) f_{\mathbf{U}}(U) = \prod_{i=1}^{m} n\rho_i^{n-1} f_{\mathbf{u}_i}(u_i) = \prod_{i=1}^{m} f_{\mathbf{u}_i,\boldsymbol{\rho}_i}(u_i,\rho_i)$$

being $f_{\mathbf{u}_i}(u_i) = \mathcal{U}_{\partial\mathcal{B}_{\|\cdot\|_1}}$. The statement then follows from Theorem 15.1, observing that $\xi_i = \rho_i u_i$, $i = 1,\dots,m$. □

Remark 17.1 (Volume of the ℓ_1 induced norm ball in $\mathbb{R}^{n,m}$) From the proof of Corollary 17.1, we obtain a closed-form expression for the volume of the ℓ_1 induced norm ball of radius r in $\mathbb{R}^{n,m}$

$$\mathrm{Vol}\big(\mathcal{B}_{|||\cdot|||_1}(r,\mathbb{R}^{n,m})\big) = \left[\frac{2^n}{n!}\right]^m r^{nm}.$$

Remark 17.2 (ℓ_∞ induced radial densities in $\mathbb{R}^{n,m}$) The case of ℓ_∞ induced radial matrices may be treated in a similar way. Indeed, the ℓ_∞ induced norm is the maximum of the ℓ_1 norms of the rows of X, i.e.

$$|||X|||_\infty = \max_{i=1,\dots,n} \|\eta_i\|_1$$

where $\eta_1^T,\dots,\eta_n^T$ are the rows of X.

The statistical properties of a real ℓ_∞ induced radial random matrix $\mathbf{X}$ can be immediately deduced, noticing that $|||\mathbf{X}|||_\infty = |||\mathbf{X}^T|||_1$. Therefore, if a random matrix $\mathbf{X}$ has an ℓ_∞ induced radial density, then its transpose $\mathbf{X}^T$ is ℓ_1 induced radial. In particular, the volume of an ℓ_∞ induced norm ball in $\mathbb{R}^{n,m}$ is given by

$$\mathrm{Vol}\big(\mathcal{B}_{|||\cdot|||_\infty}(r,\mathbb{R}^{n,m})\big) = \left[\frac{2^m}{m!}\right]^n r^{nm}.$$

17.2.2 Complex Matrices with ℓ_1/ℓ_∞ Induced Densities

The previous results for ℓ_1 and ℓ_∞ induced radial real matrices are immediately extended to the complex case. The results are reported without proof.

Theorem 17.2 (ℓ_1 induced radial matrices in $\mathbb{C}^{n,m}$) *Let the random matrix* $\mathbf{X} \in \mathbb{C}^{n,m}$ *be factored as* $\mathbf{UR}$, *where*

$$\mathbf{U} = [\mathbf{u}_1 \ \cdots \ \mathbf{u}_m], \quad \|\mathbf{u}_i\|_1 = 1;$$
$$\mathbf{R} = \mathrm{diag}\big([\boldsymbol{\rho}_1 \ \cdots \ \boldsymbol{\rho}_m]\big), \quad \boldsymbol{\rho}_i > 0$$

being $\mathbf{u}_i \in \mathbb{C}^n$ *the* i*th column of* $\mathbf{U}$. *The following statements are equivalent*:

1. $\mathbf{X}$ *is* ℓ_1 *induced radial with defining function* $g(\bar{\rho})$, $\bar{\rho} \doteq \max_{i=1,\dots,m} \rho_i$;

2. **U** *and* **R** *are independent, and their densities are given by*

$$f_{\mathbf{U}}(U) = \prod_{i=1}^{m} f_{\mathbf{u}_i}(u_i), \quad f_{\mathbf{u}_i}(u_i) = \mathcal{U}_{\partial \mathcal{B}_{\|\cdot\|_1}(\mathbb{C}^n)};$$

$$f_{\mathbf{R}}(R) = \left[\frac{(2\pi)^n}{(2n-1)!}\right]^m g(\bar{\rho}) \prod_{i=1}^{m} \rho_i^{2n-1}.$$

Next, we present a corollary which gives a characterization of uniformly distributed complex matrices in ℓ_1 induced norm balls.

Corollary 17.2 (Uniform complex matrices in $\mathcal{B}_{|||\cdot|||_1}$) *Let* $\mathbf{X} \in \mathbb{C}^{n,m}$ *and let* $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m$ *be the columns of* $\mathbf{X}$. *The following statements are equivalent*:

1. $\mathbf{X}$ *is uniformly distributed in the set* $\mathcal{B}_{|||\cdot|||_1}(\mathbb{C}^{n,m})$;
2. $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m$ *are independent and uniformly distributed in* $\mathcal{B}_{\|\cdot\|_1}(\mathbb{C}^n)$.

Remark 17.3 (ℓ_∞ induced radial densities in $\mathbb{C}^{n,m}$) Similar to the real case, the statistical properties of a complex ℓ_∞ induced radial random matrix $\mathbf{X}$ can be immediately derived. In fact, if a complex random matrix $\mathbf{X}$ has an ℓ_∞ induced radial density, then $\mathbf{X}^*$ is ℓ_1 induced radial.

Remark 17.4 (Volume of the ℓ_1 and ℓ_∞ induced norm balls in $\mathbb{C}^{n,m}$) The volume of the ℓ_1 and ℓ_∞ induced norm balls of radius r in $\mathbb{C}^{n,m}$ are given by

$$\mathrm{Vol}\big(\mathcal{B}_{|||\cdot|||_1}\big(r, \mathbb{C}^{n,m}\big)\big) = \left[\frac{(2\pi)^n}{(2n)!}\right]^m r^{2nm};$$

$$\mathrm{Vol}\big(\mathcal{B}_{|||\cdot|||_\infty}\big(r, \mathbb{C}^{n,m}\big)\big) = \left[\frac{(2\pi)^m}{(2m)!}\right]^n r^{2nm}.$$

Next, we study in detail the important case of ℓ_2 induced radial matrices, also denoted as σ radial matrices.

17.3 Statistical Properties of σ Radial Densities

In this section we consider random matrices with radial distribution with respect to the ℓ_2 induced norm. We recall that the ℓ_2 induced norm of a matrix $X \in \mathbb{F}^{n,m}$ is usually referred to as the spectral norm (or σ norm), which is defined in (3.11) as

$$|||X|||_2 = \bar{\sigma}(X)$$

where $\bar{\sigma}(X)$ is the largest singular value of X. We now define σ radial matrix density functions.

Definition 17.3 (σ radial matrix densities) A random matrix $\mathbf{X} \in \mathbb{F}^{n,m}$ is radial in the ℓ_2 induced norm, or σ radial, if its density can be written as

$$f_{\mathbf{X}}(X) = g(\bar{\sigma}), \quad \bar{\sigma} = \bar{\sigma}(X)$$

where $g(\bar{\sigma})$ is the defining function of $\mathbf{X}$.

17.3.1 Positive Definite Matrices

We first consider the case of a real positive definite matrix $X \succ 0$ and define a *normalized* singular value decomposition (SVD) of X.

Definition 17.4 (Normalized SVD of positive definite matrices) Any positive definite matrix $X \in \mathbb{S}^n$, $X \succ 0$, can be written in the form

$$X = U \Sigma U^T \tag{17.2}$$

where $\Sigma = \operatorname{diag}(\sigma)$, $\sigma = [\sigma_1 \ \cdots \ \sigma_n]^T$, with $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, and $U \in \mathbb{R}^{n,n}$ has orthonormal columns, normalized so that the first nonvanishing component of each column is positive.

Remark 17.5 (The orthogonal group) The set of orthogonal matrices in $\mathbb{R}^{n,n}$ forms a group. This group is generally referred to as the *orthogonal group* and is denoted by

$$\mathcal{G}_{\mathcal{O}}^n \doteq \left\{ U \in \mathbb{R}^{n,n} : U^T U = I \right\}. \tag{17.3}$$

Moreover, the real manifold

$$\mathcal{R}^{m,n} \doteq \left\{ U \in \mathbb{R}^{m,n} : U^T U = I; \ [U]_{1,i} > 0, \ i = 1, \ldots, n \right\} \tag{17.4}$$

represents the set of matrices $U \in \mathbb{R}^{m,n}$, $m \geq n$, whose columns are orthonormal with positive first component.

Remark 17.6 (Haar invariant distributions) In the literature, see e.g. [22], the uniform distribution over the orthogonal group $\mathcal{G}_{\mathcal{O}}^n$ is known as the Haar invariant distribution, which is denoted here as $\mathcal{U}_{\mathcal{G}_{\mathcal{O}}^n}$. Similarly, the uniform distribution over the manifold $\mathcal{R}^{n,n}$ of normalized orthogonal matrices is known as the conditional Haar invariant distribution, denoted as $\mathcal{U}_{\mathcal{R}^{n,n}}$. The Haar invariant distribution is the only distribution with the property that if $\mathbf{U} \in \mathcal{G}_{\mathcal{O}}^n$ is distributed according to Haar, then $Q\mathbf{U} \sim \mathbf{U}$ for any fixed orthogonal matrix Q.

The objective of this section is to relate the density function of the positive definite random matrix $\mathbf{X}$ to the pdfs of its SVD factors $\mathbf{U}$ and $\boldsymbol{\Sigma}$, using the mapping defined in (17.2). However, we notice that this mapping presents some ambiguity, in the sense that X may not be uniquely defined by a U, Σ pair, i.e. the mapping is not one-to-one. This is discussed in the next example.

Example 17.2 (Nonuniqueness of the SVD) Consider the identity matrix $X \equiv I$. In this case, for any $U_1 \in \mathcal{G}^n_{\mathcal{O}}$, $U_2 \in \mathcal{G}^n_{\mathcal{O}}$ such that $U_1 \neq U_2$, we can write the SVD of X either as

$$X = U_1 \Sigma U_1^T, \quad \Sigma = I$$

or as

$$X = U_2 \Sigma U_2^T.$$

Furthermore, consider a positive definite matrix $X \succ 0$ with singular value decomposition $X = U_1 \Sigma U_1^T$, $U_1 \in \mathcal{G}^n_{\mathcal{O}}$. It can be easily seen that

$$X = U_2 \Sigma U_2^T, \quad U_2 = -U_1.$$

Remark 17.7 (One-to-one SVD) The matrices with at least two coincident singular values are not uniquely represented by (17.2). However, this mapping may be made one-to-one by considering strict inequalities in the ordering of the singular values. That is, we consider $\sigma \in \mathcal{D}_\sigma$, where the singular value *ordered domain* $\mathcal{D}_\sigma$ is defined as

$$\mathcal{D}_\sigma \doteq \left\{\sigma \in \mathbb{R}^n : 1 \geq \sigma_1 > \cdots > \sigma_n > 0\right\}. \tag{17.5}$$

For a similar reason, to avoid the possible ambiguities shown in Example 17.2, a normalization condition may be imposed on every *first* element of the columns of U. That is, in (17.2) we fix the signs of the rows of the matrix U, taking $U \in \mathcal{R}^{n,n}$.

We notice that these normalizations do not affect the probabilistic results developed in the following. In fact, we are excluding a set of measure zero from the decomposition (17.2). In other words, for the class of densities under study, the probability of two singular values being equal is zero and the probability of $[\mathbf{U}]_{1,i} = 0$ is also zero. Finally, we remark that these normalizations are in agreement with classical literature on this topic, see e.g. [22], where strict inequalities in the ordering of the eigenvalues of symmetric matrices and a normalization condition on the columns of the eigenvector matrices are considered.

The next result relates the density function of a σ radial positive definite matrix to the densities of its SVD factors Σ and U.

Theorem 17.3 (Positive definite σ radial matrices) *Let the positive definite random matrix* $\mathbf{X} \in \mathbb{S}^n$, $\mathbf{X} \succ 0$, *be factored as* $\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$ *according to Definition* 17.4, *with* $\boldsymbol{\sigma} \in \mathcal{D}_\sigma$ *and* $U \in \mathcal{R}^{n,n}$. *The following statements are equivalent*:

1. $\mathbf{X}$ *is* σ *radial with defining function* $g(\bar{\sigma})$;
2. $\mathbf{U}$ *and* $\boldsymbol{\Sigma}$ *are independent, and their densities are given by*

$$f_{\mathbf{U}}(U) = \mathcal{U}_{\mathcal{R}^{n,n}}; \tag{17.6}$$

$$f_{\boldsymbol{\Sigma}}(\Sigma) = \Upsilon_{\mathbb{S}} g(\bar{\sigma}) \prod_{1 \leq i < k \leq n} (\sigma_i - \sigma_k) \tag{17.7}$$

where the normalization constant $\Upsilon_{\mathbb{S}}$ is

$$\Upsilon_{\mathbb{S}} = \pi^{\frac{n}{4}(n+1)} \prod_{i=1}^{n} \frac{1}{\Gamma(\frac{n-i+1}{2})}. \tag{17.8}$$

Proof Consider the transformation $X = U\Sigma U^T$. The strict inequalities in the ordering of the singular values and the normalization conditions on the columns of U make the mapping between X and U, Σ one-to-one. The joint pdf in the new variables U, Σ may be obtained applying Theorem A.1 (see Appendix A.1) on the transformation of random variables

$$f_{\mathbf{U},\boldsymbol{\Sigma}}(U, \Sigma) = g(\bar{\sigma}) J(X \to U, \Sigma). \tag{17.9}$$

To compute the Jacobian $J(X \to U, \Sigma)$, we make use of the rules stated in Appendix A.2. In particular, using Rule A.2 on the Jacobian of the differentials, we have that $J(X \to U, \Sigma) = J(\mathrm{d}X \to \mathrm{d}U, \mathrm{d}\Sigma)$. The differential of X is given by

$$\mathrm{d}X = \mathrm{d}U\, \Sigma U^T + U\, \mathrm{d}\Sigma\, U^T + U\Sigma\, \mathrm{d}U^T.$$

Multiplying this equation by U^T on the left and by U on the right, we obtain

$$Z \doteq U^T\, \mathrm{d}X U = U^T\, \mathrm{d}U\, \Sigma + \mathrm{d}\Sigma + \Sigma\, \mathrm{d}U^T U. \tag{17.10}$$

Applying the chain rule for Jacobians (Rule A.1), we have

$$J(\mathrm{d}X \to \mathrm{d}U, \mathrm{d}\Sigma) = J(\mathrm{d}X \to Z) J(Z \to \mathrm{d}U, \mathrm{d}\Sigma).$$

Since, by Rule A.4, $J(\mathrm{d}X \to Z) = 1$, it follows that

$$J(X \to U, \Sigma) = J(Z \to \mathrm{d}U, \mathrm{d}\Sigma).$$

Next, we rewrite (17.10) in the form

$$Z = S_u \Sigma + \mathrm{d}\Sigma + \Sigma S_u^T$$

where $S_u \doteq U^T \mathrm{d}U \in \mathbb{R}^{n,n}$. We notice that the Jacobian $J(S_u, \mathrm{d}\Sigma \to \mathrm{d}U, \mathrm{d}\Sigma)$ is equal to one by Rule A.4. Then, applying the chain rule again, we have

$$J(Z \to \mathrm{d}U, \mathrm{d}\Sigma) = J(Z \to S_u, \mathrm{d}\Sigma) J(S_u, \mathrm{d}\Sigma \to \mathrm{d}U, \mathrm{d}\Sigma) = J(Z \to S_u, \mathrm{d}\Sigma).$$

We now concentrate on the evaluation of $J(Z \to S_u, \mathrm{d}\Sigma)$. First, notice that S_u is skew-symmetric. This is easily seen by differentiating the identity $U^T U = I$, obtaining $\mathrm{d}U^T U + U^T \mathrm{d}U = 0$. The matrix Z may therefore be rewritten in the form

$$Z = S_u \Sigma - \Sigma S_u + \mathrm{d}\Sigma.$$

Then, we examine the number of free variables that describe the quantities of interest. The symmetric matrix X is described by means of $\frac{n}{2}(n+1)$ real variables, the orthogonal matrix U by $n_u \doteq \frac{n}{2}(n-1)$ real variables, and Σ by means of its n diagonal entries. The differentials $\mathrm{d}U$ and $\mathrm{d}\Sigma$ are described by the same number of free variables of U and Σ respectively. Therefore, S_u is described by n_u variables. Since $\mathrm{d}\Sigma$ is diagonal, we choose its n free variables as the diagonal entries $\eta_i = \mathrm{d}\sigma_i$,

$1 \le i \le n$. Since S_u is skew-symmetric, we choose the n_u free variables μ_{ik} as the coefficients of the standard orthonormal basis of the space of $n \times n$ skew-symmetric matrices. In particular

$$S_u = \sum_{1 \le i < k \le n} \mu_{ik} B_{ik}^{\mathbb{R}}$$

where $B_{ik}^{\mathbb{R}}$ are the elements of the basis. Denoting by E_{ik} an $n \times n$ matrix having one in position (i, k) and zero otherwise, the elements of the basis are defined as

$$B_{ik}^{\mathbb{R}} \doteq \frac{1}{\sqrt{2}}(E_{ik} - E_{ki}), \quad 1 \le i < k \le n. \tag{17.11}$$

The (i, k) entry of Z may now be expressed as

$$[Z]_{i,k} = \begin{cases} \mu_{ik}(\sigma_k - \sigma_i); & 1 \le i < k \le n; \\ \eta_i; & i = k,\ 1 \le i \le n. \end{cases}$$

To compute the Jacobian $J(Z \to S_u, \mathrm{d}\Sigma)$, we construct the following scheme of partial derivatives

	$[Z]_{i,i}$ $1 \le i \le n$	$[Z]_{i,k}$ $1 \le i < k \le n$
η_i; $1 \le i \le n$	I	0
μ_{ik}; $1 \le i < k \le n$	0	$C - D$

where

$$C = \frac{1}{\sqrt{2}} \operatorname{bdiag}\big(\operatorname{diag}\big([\sigma_2 \ \cdots \ \sigma_n]\big), \operatorname{diag}\big([\sigma_3 \ \cdots \ \sigma_n]\big), \ldots, \sigma_n\big);$$

$$D = \frac{1}{\sqrt{2}} \operatorname{bdiag}(\sigma_1 I_{n-1}, \sigma_2 I_{n-2}, \ldots, \sigma_{n-2} I_2, \sigma_{n-1}).$$

The matrix of partial derivatives is block diagonal and, therefore, its determinant is given by

$$J(Z \to S_u, \mathrm{d}\Sigma) = |C - D| = 2^{\frac{n}{4}(1-n)} \prod_{1 \le i < k \le n} (\sigma_i - \sigma_k).$$

Now, from (17.9) it follows that

$$f_{\mathbf{U},\boldsymbol{\Sigma}}(U, \Sigma) = g(\bar{\sigma}) 2^{\frac{n}{4}(1-n)} \prod_{1 \le i < k \le n} (\sigma_i - \sigma_k). \tag{17.12}$$

From this equation we immediately conclude that $\mathbf{U}$ and $\boldsymbol{\Sigma}$ are statistically independent. It also follows that $f_{\mathbf{U}}(U)$ is constant over its domain, and this proves (17.6). Finally, integrating (17.12) with respect to U we get

$$f_{\boldsymbol{\Sigma}}(\Sigma) = \int f_{\mathbf{U},\boldsymbol{\Sigma}}(U, \Sigma)\, \mathrm{d}U = \Upsilon_{\mathbb{S}} g(\bar{\sigma}) \prod_{1 \le i < k \le n} (\sigma_i - \sigma_k).$$

The constant $\Upsilon_{\mathbb{S}}$ is given by

$$\Upsilon_{\mathbb{S}} = 2^{\frac{n}{4}(1-n)} \int_{\mathcal{R}^{n,n}} \mathrm{d}U$$

where (see for instance [210])

$$\int_{\mathcal{R}^{n,n}} \mathrm{d}U = \frac{1}{2^n} \int_{\mathcal{G}^n_{\mathcal{O}}} \mathrm{d}U = \frac{(8\pi)^{\frac{n}{4}(n-1)}}{2^n} \prod_{i=1}^{n} \frac{\Gamma(\frac{i-1}{2})}{\Gamma(i-1)}.$$

Simple computations finally lead to Eq. (17.8). □

This theorem is closely related to a classical result, given in [22], on the density function of the eigenvalues of a symmetric matrix whose density depends only on its eigenvalues.

Remark 17.8 (Symmetric σ radial matrices) A result analogous to Theorem 17.3 can be easily obtained for symmetric (not necessarily positive definite) σ radial matrices $\mathbf{X} \in \mathbb{S}^n$. In this case we consider the factorization $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{S}\mathbf{U}^T$, where $\mathbf{U}$ and $\boldsymbol{\Sigma}$ are given in Definition 17.4, and $\mathbf{S}$ is a diagonal matrix of signs. In this case, the densities of $\mathbf{U}$ and $\boldsymbol{\Sigma}$ are as in Theorem 17.3 and the random signs in $\mathbf{S}$ are uniform. Therefore, the joint density of the factors is given by

$$f_{\mathbf{U},\boldsymbol{\Sigma},\mathbf{S}}(U, \Sigma, S) = \frac{1}{2^n} f_{\mathbf{U}}(U) f_{\boldsymbol{\Sigma}}(\Sigma)$$

where the constant $1/2^n$ takes into account all possible combinations of signs.

The following corollary, based on the results of Theorem 17.3, provides a characterization of positive definite matrices uniformly distributed in the σ norm ball

$$\mathcal{B}_\sigma\left(\mathbb{S}^n_+\right) \doteq \left\{X \in \mathbb{S}^n,\ X \succ 0 : \bar{\sigma}(X) \leq 1\right\}.$$

Corollary 17.3 (Uniform positive definite matrices in $\mathcal{B}_\sigma(\mathbb{S}^n_+)$) *Let the positive definite random matrix $\mathbf{X} \in \mathbb{S}^n$, $\mathbf{X} \succ 0$, be factored as $\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$ according to Definition 17.4, with $\boldsymbol{\sigma} \in \mathcal{D}_\sigma$ and $U \in \mathcal{R}^{n,n}$. The following statements are equivalent*:

1. $\mathbf{X}$ *is uniformly distributed in* $\mathcal{B}_\sigma(\mathbb{S}^n_+)$;
2. $\mathbf{U}$ *and* $\boldsymbol{\Sigma}$ *are independent, and their densities are given by*

$$\begin{aligned} f_{\mathbf{U}}(U) &= \mathcal{U}_{\mathcal{R}^{n,n}}; \\ f_{\boldsymbol{\Sigma}}(\Sigma) &= K_{\mathbb{S}} \prod_{1 \leq i < k \leq n} (\sigma_i - \sigma_k) \end{aligned} \tag{17.13}$$

where the normalization constant $K_{\mathbb{S}}$ is

$$K_{\mathbb{S}} = \pi^{\frac{n}{2}} \prod_{i=1}^{n} \frac{\Gamma(\frac{n+i}{2} + 1)}{\Gamma(\frac{i}{2})\Gamma^2(\frac{i+1}{2})}.$$

Proof To obtain (17.13) we substitute the defining function (17.1) of the uniform pdf in $\mathcal{B}_\sigma(\mathbb{S}_+^n)$ into Eq. (17.7) of Theorem 17.3. The normalization constant $K_{\mathbb{S}}$ can be computed by imposing

$$\int_{\mathcal{D}_\sigma} f_{\boldsymbol{\Sigma}}(\Sigma)\,\mathrm{d}\Sigma = 1.$$

To solve this integral we use a standard technique (see e.g. [281]) that consists in removing the ordering condition on the singular values, introducing an absolute value sign, and dividing the resulting integral by $n!$. That is, we obtain

$$\int_{\mathcal{D}_\sigma} K_{\mathbb{S}} \prod_{1\le i<k\le n} (\sigma_i - \sigma_k)\,\mathrm{d}\sigma = \frac{1}{n!}\int_0^1 \cdots \int_0^1 K_{\mathbb{S}} \prod_{1\le i<k\le n} |\sigma_i - \sigma_k|\,\mathrm{d}\sigma.$$

The corollary is then proved, noticing that the right-hand side of this equation is a Selberg integral with parameters $\gamma = 1/2$, $\alpha = \beta = 1$, see Appendix A.3. □

Remark 17.9 (Volumes of $\mathcal{B}_\sigma(r, \mathbb{S}_+^n)$ and $\mathcal{B}_\sigma(r, \mathbb{S}^n)$) From the proof of Corollary 17.3, comparing Eqs. (17.7) and (17.13), we derive a closed-form expression for the volume of the σ norm ball of radius r in the space of positive definite matrices

$$\mathrm{Vol}\big(\mathcal{B}_\sigma\big(r, \mathbb{S}_+^n\big)\big) = \frac{\Upsilon_{\mathbb{S}}}{K_{\mathbb{S}}} = \pi^{\frac{n}{4}(n-1)} \prod_{i=1}^{n} \frac{\Gamma^2(\frac{i+1}{2})}{\Gamma(\frac{n+i}{2}+1)} r^{n^2}.$$

Similarly, uniform symmetric matrices in $\mathcal{B}_\sigma(\mathbb{S}^n)$ (not necessarily positive definite) have the same singular values density (17.13), but the volume of the norm ball is in this case given by

$$\mathrm{Vol}\big(\mathcal{B}_\sigma\big(r, \mathbb{S}^n\big)\big) = 2^n \pi^{\frac{n}{4}(n-1)} \prod_{i=1}^{n} \frac{\Gamma^2(\frac{i+1}{2})}{\Gamma(\frac{n+i}{2}+1)} r^{n^2}.$$

Next, we consider the general case of rectangular real matrices with σ radial density.

17.3.2 Real σ Radial Matrix Densities

Consider the normalized singular value decomposition given below.

Definition 17.5 (Normalized SVD of real matrices) Any matrix $X \in \mathbb{R}^{n,m}$ can be written in the form

$$X = U\,\Sigma\,V^T$$

where $\Sigma = \mathrm{diag}(\sigma)$, $\sigma = [\sigma_1 \cdots \sigma_n]^T$, with $\sigma_1 \ge \cdots \ge \sigma_n \ge 0$, $U \in \mathbb{R}^{n,n}$ and $V \in \mathbb{R}^{m,n}$ have orthonormal columns, and V is normalized so that the first non-vanishing component of each column is positive.

We now state a result that relates the pdf of a σ radial real matrix to the pdfs of its SVD factors. In order to make the SVD mapping one-to-one, the next theorem requires the additional conditions $V \in \mathcal{R}^{m,n}$ and $\sigma \in \mathcal{D}_\sigma$ (see Definitions (17.4) and (17.5)). This issue is also discussed in Remark 17.7 for positive definite matrices, and a similar reasoning applies here.

Theorem 17.4 (σ radial matrices in $\mathbb{R}^{n,m}$) *Let the real random matrix* $\mathbf{X} \in \mathbb{R}^{n,m}$, $m \geq n$, *be factored as* $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ *according to Definition* 17.5, *with* $\boldsymbol{\sigma} \in \mathcal{D}_\sigma$ *and* $\mathbf{V} \in \mathcal{R}^{m,n}$. *The following statements are equivalent*:

1. $\mathbf{X}$ *is* σ *radial with defining function* $g(\bar{\sigma})$;
2. $\mathbf{U}$, $\boldsymbol{\Sigma}$ *and* $\mathbf{V}$ *are independent, and their densities are given by*

$$f_{\mathbf{U}}(U) = \mathcal{U}_{\mathcal{G}^n_{\mathcal{O}}}; \tag{17.14}$$

$$f_{\boldsymbol{\Sigma}}(\Sigma) = \Upsilon_{\mathbb{R}} g(\bar{\sigma}) \prod_{i=1}^{n} \sigma_i^{m-n} \prod_{1 \leq i < k \leq n} \left(\sigma_i^2 - \sigma_k^2\right); \tag{17.15}$$

$$f_{\mathbf{V}}(V) = \mathcal{U}_{\mathcal{R}^{m,n}} \tag{17.16}$$

where the normalization constant $\Upsilon_{\mathbb{R}}$ *is*

$$\Upsilon_{\mathbb{R}} = 2^n \pi^{\frac{n}{2}(m+1)} \prod_{i=1}^{n} \frac{1}{\Gamma(\frac{n-i+1}{2})\Gamma(\frac{m-i+1}{2})}. \tag{17.17}$$

Proof This proof is similar to that of Theorem 17.3. The SVD $X = U\Sigma V^T$, with the strict ordering of the singular values $\sigma \in \mathcal{D}_\sigma$ and the normalization condition on the columns of $V \in \mathcal{R}^{m,n}$, is one-to-one. Using Theorem A.1 (see Appendix A.1), the joint pdf of the random matrices $\mathbf{U}$, $\boldsymbol{\Sigma}$, $\mathbf{V}$ is

$$f_{\mathbf{U},\boldsymbol{\Sigma},\mathbf{V}}(U, \Sigma, V) = g(\bar{\sigma}) J(X \to U, \Sigma, V). \tag{17.18}$$

The differential of X is given by

$$\mathrm{d}X = \mathrm{d}U\, \Sigma V^T + U\, \mathrm{d}\Sigma\, V^T + U \Sigma\, \mathrm{d}V^T. \tag{17.19}$$

If $m > n$, then let $V_1 \in \mathbb{R}^{m,m-n}$ be such that $\bar{V} \doteq [V \;\; V_1]$ is orthogonal; otherwise, if $m = n$, then let $\bar{V} = V$. Then, multiplying (17.19) by U^T on the left and by $\bar{V}$ on the right, we obtain

$$Z \doteq U^T\, \mathrm{d}X\, \bar{V} = \left[U^T\, \mathrm{d}U\, \Sigma \quad 0\right] + [\mathrm{d}\Sigma \quad 0] + \Sigma\, \mathrm{d}V^T\, \bar{V}.$$

Proceeding as in the proof of Theorem 17.3, we write

$$\begin{aligned} J(X \to U, \Sigma, V) &= J(\mathrm{d}X \to \mathrm{d}U, \mathrm{d}\Sigma, \mathrm{d}V) \\ &= J(\mathrm{d}X \to Z) J(Z \to \mathrm{d}U, \mathrm{d}\Sigma, \mathrm{d}V) \\ &= J(Z \to \mathrm{d}U, \mathrm{d}\Sigma, \mathrm{d}V) \end{aligned} \tag{17.20}$$

since $J(\mathrm{d}X \to Z) = 1$. Next, we rewrite this equation in the form

$$Z = [S_u \Sigma \quad 0] + [\mathrm{d}\Sigma \quad 0] + \Sigma \bar{S}_v^T$$

where

$$S_u \doteq U^T \, \mathrm{d}U \in \mathbb{R}^{n,n} \quad \text{and} \quad \bar{S}_v \doteq \bar{V}^T \, \mathrm{d}V \in \mathbb{R}^{m,n}.$$

By Rule A.4, the Jacobian $J(S_u, \mathrm{d}\Sigma, \bar{S}_v \to \mathrm{d}U, \mathrm{d}\Sigma, \mathrm{d}V)$ is equal to one and, applying the chain rule, we have

$$\begin{aligned} J(Z \to \mathrm{d}U, \mathrm{d}\Sigma, \mathrm{d}V) &= J(Z \to S_u, \mathrm{d}\Sigma, \bar{S}_v) J(S_u, \mathrm{d}\Sigma, \bar{S}_v \to \mathrm{d}U, \mathrm{d}\Sigma, \mathrm{d}V) \\ &= J(Z \to S_u, \mathrm{d}\Sigma, \bar{S}_v). \end{aligned} \tag{17.21}$$

We now concentrate on the evaluation of $J(Z \to S_u, \mathrm{d}\Sigma, \bar{S}_v)$. We notice that S_u is skew-symmetric and, if $m > n$, $\bar{S}_v$ can be partitioned as

$$\bar{S}_v = \begin{bmatrix} S_v \\ Q^T \end{bmatrix}$$

where $S_v \doteq V^T \mathrm{d}V \in \mathbb{R}^{n,n}$ is skew-symmetric and $Q \doteq \mathrm{d}V^T V_1 \in \mathbb{R}^{n,m-n}$. The matrix Z is finally rewritten in the form

$$Z = [S_u \Sigma - \Sigma S_v + \mathrm{d}\Sigma \quad \Sigma Q].$$

Clearly, if $m = n$, then $\bar{S}_v \equiv S_v$ and $Z = S_u \Sigma - \Sigma S_v + \mathrm{d}\Sigma$.

We now examine the number of free variables that describe the quantities of interest. The matrix X is described by means of nm real variables, the orthogonal matrix U by $n_u \doteq \frac{n}{2}(n-1)$ real variables, and Σ by means of its n diagonal entries; therefore, V is described by the remaining $n_v \doteq nm - \frac{n}{2}(n-1) - n = n(m-n) + \frac{n}{2}(n-1)$ real variables.[1] The differentials $\mathrm{d}U$, $\mathrm{d}V$, and $\mathrm{d}\Sigma$ are described by the same number of free variables as, respectively, U, V, and Σ. Therefore, S_u and $\bar{S}_v^T = [S_v^T \;\; Q]$ are described by n_u and n_v variables respectively. Since $\mathrm{d}\Sigma$ is diagonal, we choose its n free variables as the diagonal entries $\eta_i = \mathrm{d}\sigma_i$, $1 \le i \le n$. Since S_u is skew-symmetric, we choose the n_u free variables μ_{ik} as the coefficients of the standard orthonormal basis of the space of $n \times n$ skew-symmetric matrices. Therefore, using the notation introduced in the proof of Theorem 17.3, we write

$$S_u = \sum_{1 \le i < k \le n} \mu_{ik} B_{ik}^{\mathbb{R}}.$$

Similarly, considering that the matrix $\bar{S}_v = [S_v^T \;\; Q]^T$ is the first (block) column of the $m \times m$ skew-symmetric matrix $\bar{V}^T \mathrm{d}\bar{V}$, we choose $n(n-1)/2$ free variables ν_{ik} such that

$$S_v = \sum_{1 \le i < k \le n} \nu_{ik} B_{ik}^{\mathbb{R}}$$

[1] The number $n_v = nm - \frac{n}{2}(n+1)$ of free variables needed to represent an $m \times n$ matrix $V = [v_1 \cdots v_n]$, $m \ge n$ with orthonormal columns can be constructed as follows: the first column v_1 can be chosen in $m-1$ different ways (m free variables with one norm constraint), v_2 can be chosen in $m-2$ different ways (m free variables with one norm constraint and one orthogonality constraint), …, v_n can be chosen in $m-n$ different ways.

where $B_{ik}^{\mathbb{R}}$ are defined in (17.11). The remaining $n_v - n(n-1)/2 = n(m-n)$ free variables are needed to describe Q. Hence, we write

$$[Q]_{i,k} = \frac{1}{\sqrt{2}} q_{ik}, \quad 1 \le i \le n;\ 1 \le k \le m-n.$$

The (i,k) entry of Z may now be expressed as

$$[Z]_{i,k} = \begin{cases} \mu_{ik}\sigma_k - \nu_{ik}\sigma_i; & 1 \le i < k \le n; \\ -\mu_{ki}\sigma_k + \nu_{ki}\sigma_i; & 1 \le k < i \le n; \\ \eta_i; & i = k,\ 1 \le i \le n; \\ q_{ir}\sigma_i; & r = k-n;\ 1 \le i \le n;\ n < k \le m. \end{cases}$$

To compute the Jacobian $J(S_u, \mathrm{d}\Sigma, \bar{S}_v; Z)$, we construct the following scheme of partial derivatives

	$[Z]_{i,i}$ $1 \le i \le n$	$[Z]_{i,k}$ $1 \le i < k \le n$	$[Z]_{i,k}$ $1 \le k < i \le n$	$[Z]_{i,k}$ $1 \le i \le n;\ n < k \le m$
η_i; $1 \le i \le n$	I	0	0	0
μ_{ik}; $1 \le i < k \le n$	0	C	$-D$	0
ν_{ik}; $1 \le i < k \le n$	0	$-D$	C	0
q_{ik}; $1 \le i \le n$; $1 \le k \le m-m$	0	0	0	F

where

$$C = \frac{1}{\sqrt{2}} \operatorname{bdiag}\big(\operatorname{diag}\big([\sigma_2\ \cdots\ \sigma_n]\big), \operatorname{diag}\big([\sigma_3\ \cdots\ \sigma_n]\big), \ldots, \sigma_n\big);$$

$$D = \frac{1}{\sqrt{2}} \operatorname{bdiag}(\sigma_1 I_{n-1}, \sigma_2 I_{n-2}, \ldots, \sigma_{n-2} I_2, \sigma_{n-1});$$

$$F = \frac{1}{\sqrt{2}} \operatorname{bdiag}(\sigma_1 I_{m-n}, \sigma_2 I_{m-n}, \ldots, \sigma_{n-1} I_{m-n}, \sigma_n I_{m-n}).$$

The matrix of partial derivatives is block diagonal and, therefore, its determinant is given by

$$J(Z \to S_u, \mathrm{d}\Sigma, \bar{S}_v) = |F| \begin{vmatrix} C & -D \\ -D & C \end{vmatrix}.$$

Using the Schur complement, we have

$$J(Z \to S_u, \mathrm{d}\Sigma, \bar{S}_v) = |F| \big|C^2 - D^2\big| = 2^{\frac{n}{2}(1-m)} \prod_{i=1}^{n} \sigma_i^{m-n} \prod_{1 \le i < k \le n} \big(\sigma_i^2 - \sigma_k^2\big).$$

Now, from (17.18), (17.20), and (17.21) it follows that

$$f_{\mathbf{U},\boldsymbol{\Sigma},\mathbf{V}}(U,\Sigma,V) = g(\bar{\sigma})2^{\frac{n}{2}(1-m)}\prod_{i=1}^{n}\sigma_i^{m-n}\prod_{1\le i<k\le n}\left(\sigma_i^2-\sigma_k^2\right). \tag{17.22}$$

From this equation we immediately obtain that $\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V}$ are statistically independent. It also follows that $f_{\mathbf{U}}(U)$ and $f_{\mathbf{V}}(V)$ are constant over their respective domains, which proves (17.14) and (17.16). Finally, integrating (17.22) with respect to U and V we get the marginal density (17.15)

$$\begin{aligned} f_{\boldsymbol{\Sigma}}(\Sigma) &= \int\cdots\int f_{\mathbf{U},\boldsymbol{\Sigma},\mathbf{V}}(U,\Sigma,V)\,\mathrm{d}U\,\mathrm{d}V \\ &= \Upsilon_{\mathbb{R}}\, g(\bar{\sigma})\prod_{i=1}^{n}\sigma_i^{m-n}\prod_{1\le i<k\le n}\left(\sigma_i^2-\sigma_k^2\right) \end{aligned}$$

where $\Upsilon_{\mathbb{R}}$ is a constant computed as

$$\Upsilon_{\mathbb{R}} = 2^{\frac{n}{2}(1-m)}\int_{\mathcal{G}_{\mathcal{O}}^n}\mathrm{d}U\int_{\mathcal{R}^{m,n}}\mathrm{d}V.$$

The above integrals evaluate to (see for instance [210])

$$\begin{aligned} \int_{\mathcal{G}_{\mathcal{O}}^n}\mathrm{d}U &= (8\pi)^{\frac{n}{4}(n-1)}\prod_{i=1}^{n}\frac{\Gamma(\frac{i-1}{2})}{\Gamma(i-1)}; \\ \int_{\mathcal{R}^{m,n}}\mathrm{d}V &= \frac{(8\pi)^{\frac{mn}{2}-\frac{n}{4}(n+1)}}{2^n}\prod_{i=m-n+1}^{m}\frac{\Gamma(\frac{i-1}{2})}{\Gamma(i-1)} \end{aligned}$$

where, for continuity, we take $\frac{\Gamma(\frac{i-1}{2})}{\Gamma(i-1)} = 2$ for $i = 1$. Finally, simple computations lead to Eq. (17.17). □

The following corollary, based on the results of Theorem 17.3, provides a characterization of real matrices uniformly distributed in $\mathcal{B}_\sigma(\mathbb{R}^{n,m})$.

Corollary 17.4 (Uniform real matrices in $\mathcal{B}_\sigma(\mathbb{R}^{n,m})$) *Let the real random matrix* $\mathbf{X}\in\mathbb{R}^{n,m}$, $m\ge n$, *be factored as* $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ *according to Definition* 17.5, *with* $\boldsymbol{\sigma}\in\mathcal{D}_\sigma$ *and* $\mathbf{V}\in\mathcal{R}^{m,n}$. *The following statements are equivalent*:

1. $\mathbf{X}$ *is uniformly distributed in* $\mathcal{B}_\sigma(\mathbb{R}^{n,m})$;
2. $\mathbf{U}$, $\boldsymbol{\Sigma}$ *and* $\mathbf{V}$ *are independent, and their densities are given by*

$$\begin{aligned} f_{\mathbf{U}}(U) &= \mathcal{U}_{\mathcal{G}_{\mathcal{O}}^n}; \\ f_{\boldsymbol{\Sigma}}(\Sigma) &= K_{\mathbb{R}}\prod_{i=1}^{n}\sigma_i^{m-n}\prod_{1\le i<k\le n}\left(\sigma_i^2-\sigma_k^2\right); \\ f_{\mathbf{V}}(V) &= \mathcal{U}_{\mathcal{R}^{m,n}} \end{aligned} \tag{17.23}$$

where the normalization constant $K_{\mathbb{R}}$ is

$$K_{\mathbb{R}} = (4\pi)^{\frac{n}{2}} \prod_{i=1}^{n} \frac{\Gamma(\frac{m+i+1}{2})}{\Gamma(\frac{i}{2})\Gamma(\frac{i+1}{2})\Gamma(\frac{m-n+i}{2})}. \tag{17.24}$$

Proof The proof is similar that of Corollary 17.3. To obtain (17.23) we substitute the defining function (17.1) of the uniform pdf in Eq. (17.15).The normalization constant $K_{\mathbb{R}}$ can be computed by imposing

$$\int f_{\boldsymbol{\Sigma}}(\Sigma)\, \mathrm{d}\Sigma = 1.$$

With some algebraic manipulations, we notice that this integral is a Selberg integral with parameters $\gamma = 1/2$, $\alpha = \frac{m-n+1}{2}$, $\beta = 1$, see Appendix A.3. Therefore, we obtain $K_{\mathbb{R}}$ as given in (17.24). □

Remark 17.10 (Volume of $\mathcal{B}_\sigma(r, \mathbb{R}^{n,m})$) From the proof of Corollary 17.4, comparing Eqs. (17.15) and (17.23), we derive a closed-form expression for the volume of the σ norm real ball of radius r

$$\mathrm{Vol}\big(\mathcal{B}_\sigma\big(r, \mathbb{R}^{n,m}\big)\big) = \pi^{\frac{nm}{2}} \prod_{i=1}^{n} \frac{\Gamma(\frac{i+1}{2})\Gamma(\frac{m-n+i}{2})}{\Gamma(\frac{m+i+1}{2})\Gamma(\frac{m-i+1}{2})} r^{nm}. \tag{17.25}$$

Next, we derive similar results for complex matrices with σ radial density.

17.3.3 Complex σ Radial Matrix Densities

To analyze the complex case, we first introduce the following normalized SVD decomposition.

Definition 17.6 (Normalized SVD of complex matrices) Any matrix $X \in \mathbb{C}^{n,m}$ can be written in the form

$$X = U \Sigma V^*$$

where $\Sigma = \mathrm{diag}(\sigma)$, $\sigma = [\sigma_1 \ \cdots \ \sigma_n]^T$, with $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, $U \in \mathbb{C}^{n,n}$ and $V \in \mathbb{C}^{m,n}$ have orthonormal columns, and V is normalized so that the first non-vanishing component of each column is real and positive.

Remark 17.11 (The unitary group) The set of unitary matrices in $\mathbb{C}^{n,n}$ forms a group, called the *unitary group*, which is denoted by

$$\mathcal{G}_{\mathcal{U}}^{n} \doteq \big\{U \in \mathbb{C}^{n,n} : U^*U = I\big\}. \tag{17.26}$$

Similar to the real case, see Remark 17.5, we also define the following complex manifold

$$\mathcal{C}^{m,n} \doteq \big\{V \in \mathbb{C}^{m,n} : V^*V = I;\ \mathrm{Re}\big([V]_{1,i}\big) > 0, \mathrm{Im}\big([V]_{1,i}\big) = 0,\ i = 1, \ldots, n\big\}. \tag{17.27}$$

Remark 17.12 (Complex Haar invariant distributions) The uniform distribution over the unitary group $\mathcal{G}_{\mathcal{U}}^n$ is the (complex) Haar invariant distribution, here denoted as $\mathcal{U}_{\mathcal{G}_{\mathcal{U}}^n}$. Similarly, the uniform distribution over the complex manifold $\mathcal{C}^{n,n}$ of normalized unitary matrices is denoted as $\mathcal{U}_{\mathcal{C}^{n,n}}$.

The next result relates the pdf of a σ radial complex matrix to the pdfs of its SVD factors. In order to make the SVD mapping one-to-one, the theorem requires the additional conditions $V \in \mathcal{C}^{m,n}$ and $\sigma \in \mathcal{D}_\sigma$, see Remark 17.7 for a discussion regarding the case of positive definite matrices. A similar reasoning also applies to complex matrices.

Theorem 17.5 (σ radial matrices in $\mathbb{C}^{n,m}$) *Let the complex random matrix* $\mathbf{X} \in \mathbb{C}^{n,m}$, $m \geq n$, *be factored as* $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$ *according to Definition* 17.6, *with* $\boldsymbol{\sigma} \in \mathcal{D}_\sigma$ *and* $\mathbf{V} \in \mathcal{C}^{m,n}$. *The following statements are equivalent*:

1. $\mathbf{X}$ *is* σ *radial with defining function* $g(\bar{\sigma})$;
2. $\mathbf{U}$, $\boldsymbol{\Sigma}$ *and* $\mathbf{V}$ *are independent, and their densities are given by*

$$f_{\mathbf{U}}(U) = \mathcal{U}_{\mathcal{G}_{\mathcal{U}}^n}; \tag{17.28}$$

$$f_{\boldsymbol{\Sigma}}(\Sigma) = \Upsilon_{\mathbb{C}} g(\bar{\sigma}) \prod_{i=1}^{n} \sigma_i^{2(m-n)+1} \prod_{1 \leq i < k \leq n} \left(\sigma_i^2 - \sigma_k^2\right)^2; \tag{17.29}$$

$$f_{\mathbf{V}}(V) = \mathcal{U}_{\mathcal{C}^{m,n}} \tag{17.30}$$

where the normalization constant $\Upsilon_{\mathbb{C}}$ *is*

$$\Upsilon_{\mathbb{C}} = 2^n \pi^{nm} \prod_{i=1}^{n} \frac{1}{\Gamma(n-i+1)\Gamma(m-i+1)}. \tag{17.31}$$

Proof This proof is similar to that of Theorem 17.4. Indeed, the derivation up to the expression of Z as

$$Z = [S_u \Sigma - \Sigma S_v + \mathrm{d}\Sigma \quad \Sigma Q]$$

is identical to the real case, considering that all the quantities involved are now complex (and, therefore, matrix transpose should be treated as conjugate transpose), and S_v, S_u are skew-Hermitian. In particular, we have that

$$J(X \to U, \Sigma, V) = J(Z \to S_u, \mathrm{d}\Sigma, \bar{S}_v). \tag{17.32}$$

We now examine the number of free variables that describe the quantities of interest. The matrix X is described by means of $2nm$ real variables, the unitary matrix U by $n_u \doteq n^2$ real variables, and Σ by means of its n diagonal entries; therefore, V is described by the remaining $n_v \doteq 2nm - n^2 - n$ real variables. Since an $m \times n$ complex matrix with orthonormal columns is described by $2nm - n^2$ variables, see e.g. [210], we notice that the normalization imposed by Definition 17.6 on the columns of V fixes n of the free variables. The differentials $\mathrm{d}U$, $\mathrm{d}V$, $\mathrm{d}\Sigma$ are described by the same number of free variables as U, V, and Σ. Therefore, S_u and

$\bar{S}_v^* = [S_v^* \; Q]$ are described by n_u and n_v variables respectively. Since $\mathrm{d}\Sigma$ is real diagonal, we choose its n free variables as the diagonal entries $\eta_i = \mathrm{d}\sigma_i$, $1 \le i \le n$. Since S_u is skew-Hermitian, we choose the n_u free variables $\mu_{ik}^{\mathbb{R}}$, $\mu_{ik}^{\mathbb{I}}$ as the coefficients of the standard orthonormal basis of the space of $n \times n$ skew-Hermitian matrices. In particular, we write

$$S_u = \sum_{1 \le i < k \le n} \left(\mu_{ik}^{\mathbb{R}} B_{ik}^{\mathbb{R}} + \mu_{ik}^{\mathbb{I}} B_{ik}^{\mathbb{I}}\right) + \sum_{i=1}^{n} \mu_{ii}^{\mathbb{I}} D_i$$

where $B_{ik}^{\mathbb{R}}$, $B_{ik}^{\mathbb{I}}$, and D_i are the elements of the basis. Denoting by E_{ik} an $n \times n$ matrix having one in position (i,k) and zero otherwise, the elements of the basis are defined as

$$\begin{aligned}
B_{ik}^{\mathbb{R}} &\doteq \frac{1}{\sqrt{2}}(E_{ik} - E_{ki}), \quad 1 \le i < k \le n; \\
B_{ik}^{\mathbb{I}} &\doteq \frac{j}{\sqrt{2}}(E_{ik} + E_{ki}), \quad 1 \le i < k \le n; \\
D_i &\doteq j E_{ii}, \;\; 1 \le i \le n.
\end{aligned}$$

Similarly, considering that the matrix $\bar{S}_v = [S_v^* \; Q]$ is the first (block) column of the $m \times m$ skew-Hermitian matrix $\bar{V}^* \mathrm{d}\bar{V}$, we choose $n^2 - n$ free variables $\nu_{ik}^{\mathbb{R}}$, $\nu_{ik}^{\mathbb{I}}$ such that

$$S_v = \sum_{1 \le i < k \le n} \left(\nu_{ik}^{\mathbb{R}} B_{ik}^{\mathbb{R}} + \nu_{ik}^{\mathbb{I}} B_{ik}^{\mathbb{I}}\right) + \sum_{i=1}^{n} h_i\left(\nu^{\mathbb{R}}, \nu^{\mathbb{I}}\right) D_i$$

where $h_i(\nu^{\mathbb{R}}, \nu^{\mathbb{I}})$ is a function of the variables $\nu^{\mathbb{R}}$, $\nu^{\mathbb{I}}$. The remaining $n_v - (n^2 - n) = 2n(m-n)$ free variables $q_{ik}^{\mathbb{R}}$, $q_{ik}^{\mathbb{I}}$ are needed to describe Q

$$[Q]_{i,k} = \frac{1}{\sqrt{2}}\left(q_{ik}^{\mathbb{R}} + j q_{ik}^{\mathbb{I}}\right), \quad 1 \le i \le n; \; 1 \le k \le m - n.$$

The (i,k) entry of Z may now be expressed as

$$[Z]_{i,k} = \begin{cases}
\frac{1}{\sqrt{2}}(\mu_{ik}^{\mathbb{R}} + j\mu_{ik}^{\mathbb{I}})\sigma_k - \frac{1}{\sqrt{2}}(\nu_{ik}^{\mathbb{R}} + j\nu_{ik}^{\mathbb{I}})\sigma_i; & 1 \le i < k \le n; \\
-\frac{1}{\sqrt{2}}(\mu_{ki}^{\mathbb{R}} + j\mu_{ki}^{\mathbb{I}})\sigma_k + \frac{1}{\sqrt{2}}(\nu_{ki}^{\mathbb{R}} - j\nu_{ki}^{\mathbb{I}})\sigma_i; & 1 \le k < i \le n; \\
\eta_i + j\mu_{ii}^{\mathbb{I}}\sigma_i - j h_i(\nu^{\mathbb{R}}, \nu^{\mathbb{I}})\sigma_i; & i = k, \; 1 \le i \le n; \\
\frac{1}{\sqrt{2}}(q_{ir}^{\mathbb{R}} + j q_{ir}^{\mathbb{I}})\sigma_i; & r = k - n; \\
 & 1 \le i \le n; \; n < k \le m.
\end{cases}$$

To compute the Jacobian $J(Z \to S_u, \mathrm{d}\Sigma, \bar{S}_v)$, we construct the following scheme of partial derivatives

	$\mathrm{Re}([Z]_{i,i})$ $1 \le i \le n$	$\mathrm{Im}([Z]_{i,i})$ $1 \le i \le n$	$\mathrm{Re}([Z]_{i,k})$ $1 \le i < k \le n$	$\mathrm{Re}([Z]_{i,k})$ $1 \le k < i \le n$	$\mathrm{Im}([Z]_{i,k})$ $1 \le i < k \le n$	$\mathrm{Im}([Z]_{i,k})$ $1 \le k < i \le n$	$\mathrm{Re}([Z]_{i,k})$ $1 \le i \le n;$ $1 \le n < k \le m$	$\mathrm{Im}([Z]_{i,k})$ $1 \le i \le n;$ $1 \le n < k \le m$
η_i	I	0	0	0	0	0	0	0
$\mu_{ii}^{\mathbb{I}}$ $1 \le i \le n$	0	Σ	0	0	0	0	0	0
$\mu_{ik}^{\mathbb{R}}$	0	0	C	$-D$	0	0	0	0
$v_{ik}^{\mathbb{R}}$ $1 \le i < k \le n$	0	$H^{\mathbb{R}}$	$-D$	C	0	0	0	0
$\mu_{ik}^{\mathbb{I}}$	0	0	0	0	C	D	0	0
$v_{ik}^{\mathbb{I}}$ $1 \le i < k \le n$	0	$H^{\mathbb{I}}$	0	0	$-D$	$-C$	0	0
$q_{ik}^{\mathbb{R}}$	0	0	0	0	0	0	F	0
$q_{ik}^{\mathbb{I}}$ $1 \le i \le n$ $1 \le k \le m-n$	0	0	0	0	0	0	0	F

where

$$C = \frac{1}{\sqrt{2}} \operatorname{bdiag}\big(\operatorname{diag}\big([\sigma_2 \cdots \sigma_n]\big), \operatorname{diag}\big([\sigma_3 \cdots \sigma_n]\big), \ldots, \sigma_n\big);$$

$$D = \frac{1}{\sqrt{2}} \operatorname{bdiag}(\sigma_1 I_{n-1}, \sigma_2 I_{n-2}, \ldots, \sigma_{n-2} I_2, \sigma_{n-1});$$

$$F = \frac{1}{\sqrt{2}} \operatorname{bdiag}(\sigma_1 I_{m-n}, \sigma_2 I_{m-n}, \ldots, \sigma_{n-1} I_{m-n}, \sigma_n I_{m-n}).$$

Since the matrix of partial derivatives is block triangular, the matrices $H^{\mathbb{R}}$, $H^{\mathbb{I}}$ do not affect the value of its determinant. Therefore, we obtain

$$J(Z \to S_u, \mathrm{d}\Sigma, \bar{S}_v) = |\Sigma||F|^2 \begin{vmatrix} C & -D \\ -D & C \end{vmatrix} \begin{vmatrix} C & D \\ -D & -C \end{vmatrix}.$$

Using the Schur complement, we have that

$$\begin{aligned} J(Z \to S_u, \mathrm{d}\Sigma, \bar{S}_v) &= |\Sigma||F|^2 \big|C^2 - D^2\big|^2 \\ &= 2^{n(1-m)} \prod_{i=1}^{n} \sigma_i^{2(m-n)+1} \prod_{1 \le i < k \le n} \big(\sigma_i^2 - \sigma_k^2\big)^2. \end{aligned} \tag{17.33}$$

Now, by means of Theorem A.1 (see Appendix A.1), we write the joint pdf of the random matrices $\mathbf{U}$, $\boldsymbol{\Sigma}$, $\mathbf{V}$ as

$$f_{\mathbf{U},\boldsymbol{\Sigma},\mathbf{V}}(U, \Sigma, V) = g(\bar{\sigma}) J(X \to U, \Sigma, V).$$

Using Eqs. (17.32) and (17.33) we immediately obtain

$$f_{\mathbf{U},\boldsymbol{\Sigma},\mathbf{V}}(U,\Sigma,V) = g(\bar{\sigma})2^{n(1-m)}\prod_{i=1}^{n}\sigma_i^{2(m-n)+1}\prod_{1\le i<k\le n}\left(\sigma_i^2-\sigma_k^2\right)^2. \tag{17.34}$$

From this equation we conclude that $\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V}$ are statistically independent. It also follows that $f_{\mathbf{U}}(U)$ and $f_{\mathbf{V}}(V)$ are constant over their respective domains, which proves (17.28) and (17.30). Finally, integrating (17.34) with respect to U and V, we get the marginal density (17.29) as

$$\begin{aligned} f_{\boldsymbol{\Sigma}}(\Sigma) &= \int\cdots\int f_{\mathbf{U},\boldsymbol{\Sigma},\mathbf{V}}(U,\Sigma,V)\mathrm{d}U\,\mathrm{d}V \\ &= \Upsilon_{\mathbb{C}}g(\bar{\sigma})\prod_{i=1}^{n}\sigma_i^{2(m-n)+1}\prod_{1\le i<k\le n}\left(\sigma_i^2-\sigma_k^2\right)^2 \end{aligned}$$

where the constant $\Upsilon_{\mathbb{C}}$ is given by

$$\Upsilon_{\mathbb{C}} = 2^{n(1-m)}\int_{\mathcal{G}_{\mathcal{U}}^n}\mathrm{d}U\int_{\mathcal{C}^{m,n}}\mathrm{d}V$$

and the measure of the unitary group $\mathcal{G}_{\mathcal{U}}^n$ and of the complex manifold $\mathcal{C}^{m,n}$ are given by (see for instance [210])

$$\begin{aligned} \int_{\mathcal{G}_{\mathcal{U}}^n}\mathrm{d}U &= (2\pi)^{n(n+1)/2}\prod_{i=1}^{n}\frac{1}{(n-i)!}; \\ \int_{\mathcal{C}^{m,n}}\mathrm{d}V &= \frac{(2\pi)^{mn-n(n-1)/2}}{(2\pi)^n}\prod_{i=1}^{n}\frac{1}{(m-i)!}. \end{aligned}$$

□

The following corollary, based on the results of Theorem 17.5, provides a characterization of complex matrices uniformly distributed in $\mathcal{B}_\sigma$.

Corollary 17.5 (Uniform complex matrices in $\mathcal{B}_\sigma(\mathbb{C}^{n,m})$) *Let the complex random matrix* $\mathbf{X}\in\mathbb{C}^{n,m}$, $m\ge n$, *be factored as* $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$ *according to Definition* 17.6, *with* $\boldsymbol{\sigma}\in\mathcal{D}_\sigma$ *and* $\mathbf{V}\in\mathcal{C}^{m,n}$. *The following statements are equivalent*:

1. $\mathbf{X}$ *is uniformly distributed in* $\mathcal{B}_\sigma(\mathbb{C}^{n,m})$;
2. $\mathbf{U}$, $\boldsymbol{\Sigma}$ *and* $\mathbf{V}$ *are independent, and their densities are given by*

$$\begin{aligned} f_{\mathbf{U}}(U) &= \mathcal{U}_{\mathcal{G}_{\mathcal{U}}^n}; \\ f_{\boldsymbol{\Sigma}}(\Sigma) &= K_{\mathbb{C}}\prod_{i=1}^{n}\sigma_i^{2(m-n)+1}\prod_{1\le i<k\le n}\left(\sigma_i^2-\sigma_k^2\right)^2; \\ f_{\mathbf{V}}(V) &= \mathcal{U}_{\mathcal{C}^{m,n}} \end{aligned} \tag{17.35}$$

where the normalization constant $K_{\mathbb{C}}$ *is*

$$K_{\mathbb{C}} = 2^n\prod_{i=1}^{n}\frac{\Gamma(m+i)}{\Gamma^2(i)\Gamma(m-n+i)}. \tag{17.36}$$

Proof The proof is similar to that of Corollaries 17.3 and 17.4. To obtain (17.35) we substitute the defining function (17.1) of the uniform pdf in Eq. (17.29). The normalization constant $K_{\mathbb{C}}$ can be computed by imposing

$$\int f_{\mathbf{\Sigma}}(\Sigma)\,\mathrm{d}\Sigma = 1.$$

With some algebraic manipulations, we notice that this is a Selberg integral with parameters $\gamma = 1, \alpha = m - n + 1, \beta = 1$, see Appendix A.3. □

Remark 17.13 (Volume of $\mathcal{B}_\sigma(r, \mathbb{C}^{n,m})$) From the proof of the above corollary, comparing Eqs. (17.29) and (17.35), we derive a closed-form expression for the volume of the σ norm complex ball of radius r

$$\mathrm{Vol}\big(\mathcal{B}_\sigma\big(r, \mathbb{C}^{n,m}\big)\big) = \pi^{nm} \prod_{i=1}^{n} \frac{\Gamma(i)\Gamma(m-n+i)}{\Gamma(m+i)\Gamma(m-i+1)} r^{2nm}. \tag{17.37}$$

17.4 Statistical Properties of Unitarily Invariant Matrices

The results presented in the previous two sections can be immediately extended to the more general class of unitarily invariant random matrices defined below.

Definition 17.7 (Unitarily invariant matrices) A random matrix $\mathbf{X} \in \mathbb{F}^{n,m}$ is unitarily invariant if its density function can be written as

$$f_{\mathbf{X}}(X) = g(\Sigma) \tag{17.38}$$

where $g(\Sigma)$ is the defining function of $\mathbf{X}$ and Σ is the (diagonal) singular values matrix of X.

The name unitarily invariant follows from the fact that if $\mathbf{X} \in \mathbb{C}^{n,m}$ has a unitarily invariant density, then $Q\mathbf{X} \sim \mathbf{X}$ and $\mathbf{X}W \sim \mathbf{X}$, for any given unitary matrices Q and W. Clearly, any σ radial matrix is also unitarily invariant. However, there are some important examples of unitarily invariant matrices that are not σ radial. One of these examples is the so-called Wishart density, which is discussed next.

Example 17.3 (Wishart density) Let $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_m]$, where $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$, $m \geq n$, are iid random vectors normally distributed, i.e. $\mathbf{x}_i \sim \mathcal{N}_{0,W}$, $i = 1, \ldots, m$. Construct the random matrix $\mathbf{Y} = \mathbf{X}\mathbf{X}^T$, $\mathbf{Y} \in \mathbb{S}^m$. It can be shown, see e.g. [22], that $\mathbf{Y}$ is positive definite (with probability one) and has the Wishart density, defined as

$$\mathcal{W}_W \doteq K_{\mathcal{W}} |Y|^{(m-n-1)/2} \mathrm{e}^{-\frac{1}{2}\mathrm{Tr}\, W^{-1} Y}, \quad Y \succ 0$$

where the normalization constant $K_{\mathcal{W}}$ is given by

$$K_{\mathcal{W}} = \left(2^{nm/2} |W|^{m/2} \pi^{n(n-1)/4} \prod_{i=1}^{n} \Gamma\big((m-i+1)/2\big) \right)^{-1}.$$

When $W = I$ the Wishart density $\mathcal{W}_I$ is unitarily invariant. In fact, $|Y| = |\Sigma|$ and $\operatorname{Tr} Y = \operatorname{Tr} \Sigma$, where Σ is the singular values matrix of Y. Therefore $\mathcal{W}_I = g(\Sigma)$ where the defining function $g(\Sigma)$ is given by

$$g(\Sigma) = K_{\mathcal{W}} |\Sigma|^{(m-n-1)/2} \mathrm{e}^{-\frac{1}{2}\operatorname{Tr} \Sigma}.$$

When $W \neq I$, the distribution of $\mathbf{Y}$ is no longer unitarily invariant, but it belongs to the more general class of elliptically contoured matrix distributions, which are studied for instance in [186].

Example 17.4 (Radial densities in the Frobenius norm) For $p = 2$, the Hilbert–Schmidt ℓ_p matrix norm is also known as the Frobenius norm. For a matrix $X \in \mathbb{F}^{n,m}$, the Frobenius norm is given by

$$\|X\|_2 = \operatorname{Tr} X X^T = \|\Sigma\|_2$$

where Σ is the singular values matrix of X. Therefore, the Hilbert–Schmidt ℓ_2 radial densities studied in Sect. 17.1.1 are also unitarily invariant.

The three theorems stated next, without proof, provide the extensions of Theorems 17.3, 17.4 and 17.5 to the case of random matrices with unitarily invariant distribution.

Theorem 17.6 (Unitarily invariant positive definite matrices) *Let the positive definite random matrix $\mathbf{X} \in \mathbb{S}^n$, $\mathbf{X} \succ 0$, be factored as $\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$ according to Definition 17.4, with $\boldsymbol{\sigma} \in \mathcal{D}_\sigma$ and $U \in \mathcal{R}^{n,n}$. The following statements are equivalent*:

1. $\mathbf{X}$ *is unitarily invariant with defining function* $g(\Sigma)$;
2. $\mathbf{U}$ *and* $\boldsymbol{\Sigma}$ *are independent, and their densities are given by*

$$f_{\mathbf{U}}(U) = \mathcal{U}_{\mathcal{R}^{n,n}};$$
$$f_{\boldsymbol{\Sigma}}(\Sigma) = \Upsilon_{\mathbb{S}} g(\Sigma) \prod_{1 \leq i < k \leq n} (\sigma_i - \sigma_k)$$

where the normalization constant $\Upsilon_{\mathbb{S}}$ is given in (17.8).

Theorem 17.7 (Unitarily invariant real matrices) *Let the real random matrix $\mathbf{X} \in \mathbb{R}^{n,m}$, $m \geq n$, be factored as $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ according to Definition 17.5, with $\boldsymbol{\sigma} \in \mathcal{D}_\sigma$ and $\mathbf{V} \in \mathcal{R}^{m,n}$. The following statements are equivalent*:

1. $\mathbf{X}$ *is unitarily invariant with defining function* $g(\Sigma)$;
2. $\mathbf{U}$, $\boldsymbol{\Sigma}$ *and* $\mathbf{V}$ *are independent, and their densities are given by*

$$f_{\mathbf{U}}(U) = \mathcal{U}_{\mathcal{G}^n_{\mathcal{O}}};$$
$$f_{\boldsymbol{\Sigma}}(\Sigma) = \Upsilon_{\mathbb{R}} g(\Sigma) \prod_{i=1}^{n} \sigma_i^{m-n} \prod_{1 \leq i < k \leq n} \left(\sigma_i^2 - \sigma_k^2\right);$$
$$f_{\mathbf{V}}(V) = \mathcal{U}_{\mathcal{R}^{m,n}}$$

where the normalization constant $\Upsilon_{\mathbb{R}}$ is given in (17.17).

Theorem 17.8 (Unitarily invariant complex matrices) *Let the complex random matrix* $\mathbf{X} \in \mathbb{C}^{n,m}$, $m \geq n$, *be factored as* $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$ *according to Definition* 17.6, *with* $\boldsymbol{\sigma} \in \mathcal{D}_\sigma$ *and* $\mathbf{V} \in \mathcal{C}^{m,n}$. *The following statements are equivalent*:

1. $\mathbf{X}$ *is unitarily invariant with defining function* $g(\Sigma)$;
2. $\mathbf{U}$, $\boldsymbol{\Sigma}$ *and* $\mathbf{V}$ *are independent, and their densities are given by*

$$f_{\mathbf{U}}(U) = \mathcal{U}_{\mathcal{G}_{\mathcal{U}}^n};$$
$$f_{\boldsymbol{\Sigma}}(\Sigma) = \Upsilon_{\mathbb{C}} g(\Sigma) \prod_{i=1}^{n} \sigma_i^{2(m-n)+1} \prod_{1 \leq i < k \leq n} \left(\sigma_i^2 - \sigma_k^2\right)^2;$$
$$f_{\mathbf{V}}(V) = \mathcal{U}_{\mathcal{C}^{m,n}}$$

where the normalization constant $\Upsilon_{\mathbb{C}}$ *is given in* (17.31).

Chapter 18
Matrix Randomization Methods

In this chapter we present algorithms for uniform matrix sample generation in norm bounded sets. First, we discuss the simple case of matrix sampling in sets defined by ℓ_p Hilbert–Schmidt norm, which reduces to the vector ℓ_p norm randomization problem. Subsequently we present an efficient solution to the more challenging problem of uniform generation in sets defined by the spectral norm. The algorithms presented here are available in [81, 83]. The reader interested in random generation of matrices with a Toeplitz structure may refer to [423].

18.1 Uniform Sampling in Hilbert–Schmidt Norm Balls

As discussed in Sect. 17.1.1, the vectorization operation $x = \mathrm{vec}(X)$ defined in (3.8) reduces the ℓ_p Hilbert–Schmidt matrix sample generation problem into an equivalent problem concerning vector samples. Therefore, the vector randomization algorithms presented in Chap. 16 can be used directly for the ℓ_p Hilbert–Schmidt matrix sampling problem. A simple illustration of this idea is reported next for matrix generation in the Frobenius norm ball.

Example 18.1 (Uniform matrices in the Frobenius norm ball) Suppose we are interested in generating samples of a real matrix $\mathbf{X} \in \mathbb{R}^{2,3}$ uniformly distributed in the unit Frobenius norm ball, i.e. the ℓ_2 Hilbert–Schmidt norm ball

$$\mathcal{B}_{\|\cdot\|_2}\left(\mathbb{R}^{2,3}\right) = \left\{X \in \mathbb{R}^{2,3} : \|X\|_2 \leq 1\right\}.$$

Using the vectorization operator $x = \mathrm{vec}(X)$, we observe that this problem is equivalent to the generation of uniform random vectors in the ℓ_2 norm ball $\mathcal{B}_{\|\cdot\|_2}(\mathbb{R}^6)$, which can be easily performed by means of Algorithm 16.1.

Next, we discuss uniform generation of matrix samples in the ℓ_p induced norm balls, for $p = 1$ and $p = \infty$.

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_18,

18.2 Uniform Sampling in ℓ_1 and ℓ_∞ Induced Norm Balls

The algorithm for matrix sample generation in the ℓ_1 induced norm ball follows directly from Corollary 17.1. That is, to generate a matrix $\mathbf{X} \in \mathbb{R}^{n,m}$ with uniform distribution in $\mathcal{B}_{|||\cdot|||_1}(\mathbb{R}^{n,m})$, it suffices to generate its columns independently and uniformly in the ℓ_1 vector norm ball $\mathcal{B}_{\|\cdot\|_1}(\mathbb{R}^m)$ using Algorithm 16.1. The complex case follows from direct application of Corollary 17.2, and hence Algorithm 16.4 can be used for random generation. Similarly, to generate a matrix sample $\mathbf{X}$ (real or complex) uniformly distributed in the ℓ_∞ induced norm ball $\mathcal{B}_{\|\cdot\|_\infty}(\mathbb{F}^{n,m})$, it suffices to generate $\mathbf{X}^T$ (or $\mathbf{X}^*$, in the complex case) uniformly in the ℓ_1 induced norm ball $\mathcal{B}_{\|\cdot\|_1}(\mathbb{F}^{m,n})$.

We now discuss the problem of uniform generation in the spectral norm ball, which turns out to be technically more difficult than the cases previously considered. This difficulty is mainly due to the special structure of the spectral norm, which depends on the entries of the matrix only through an implicit relation given by the singular value decomposition. In Sect. 18.3 we present several sampling schemes based on rejection, and show that these methods become rapidly inefficient as the dimension of the matrix increases. This motivates the development of the algorithms subsequently studied in Sections 18.4 and 18.5.

18.3 Rejection Methods for Uniform Matrix Generation

A simple algorithm for generating uniform matrix samples in the ℓ_2 induced (spectral) norm unit ball $\mathcal{B}_\sigma$ is given by the rejection method from a bounding set discussed in Sect. 14.3.1. We notice that for a matrix $X \in \mathbb{F}^{n,m}$, $m \geq n$, the well-known norm inequalities hold

$$\|X\|_2 \leq \sqrt{n}|||X|||_2;$$
$$\|X\|_\infty \leq |||X|||_2$$

where $|||X|||_2 = \bar{\sigma}(X)$. These inequalities in turn imply the set inclusions

$$\mathcal{B}_\sigma\left(1, \mathbb{F}^{n,m}\right) \subseteq \mathcal{B}_{\|\cdot\|_2}\left(\sqrt{n}, \mathbb{F}^{n,m}\right); \tag{18.1}$$
$$\mathcal{B}_\sigma\left(1, \mathbb{F}^{n,m}\right) \subseteq \mathcal{B}_{\|\cdot\|_\infty}\left(1, \mathbb{F}^{n,m}\right). \tag{18.2}$$

A tighter set inclusion can also be obtained by considering the inequality

$$|||X|||_2 = \max_{\|y\|_2=1} \|Xy\|_2 = \max_{\|y\|_2=1} \left\| \sum_{i=1}^{m} \xi_i y_i \right\|_2 \geq \max_{i=1,\ldots,m} \|\xi_i\|_2$$

where ξ_i is the ith column of X. In this case, it follows that

$$\mathcal{B}_\sigma\left(1, \mathbb{F}^{n,m}\right) \subseteq \mathcal{B}_{\text{col}}\left(1, \mathbb{F}^{n,m}\right) \doteq \left\{ X \in \mathbb{F}^{n,m} : \max_{i=1,\ldots,m} \|\xi_i\|_2 \leq 1 \right\}.$$

Table 18.1 Rejection rates for generating samples uniformly in $\mathcal{B}_\sigma(1, \mathbb{F}^{n,n})$ with overbounding sets given by $\mathcal{B}_{\|\cdot\|_\infty}(1, \mathbb{F}^{n,n})$, $\mathcal{B}_{\|\cdot\|_2}(\sqrt{n}, \mathbb{F}^{n,n})$ and $\mathcal{B}_{\text{col}}(1, \mathbb{F}^{n,n})$

n	$\mathcal{B}_\sigma(\mathbb{R}^{n,n})$			$\mathcal{B}_\sigma(\mathbb{C}^{n,n})$		
	$\mathcal{B}_{\|\cdot\|_\infty}$	$\mathcal{B}_{\|\cdot\|_2}$	$\mathcal{B}_{\text{col}}$	$\mathcal{B}_{\|\cdot\|_\infty}$	$\mathcal{B}_{\|\cdot\|_2}$	$\mathcal{B}_{\text{col}}$
2	2.432	3	1.5	12	8	3
3	29.57	26.72	4.244	8640	468.6	40
4	2720	640	24.61	8.71×10^{08}	1.79×10^{05}	2625
5	2.53×10^{06}	3.95×10^{04}	305	2.21×10^{16}	4.25×10^{08}	8.89×10^{05}
6	2.99×10^{10}	6.14×10^{06}	8290	2.23×10^{26}	6.17×10^{12}	1.60×10^{09}
7	5.38×10^{15}	2.38×10^{09}	5.03×10^{05}	1.28×10^{39}	5.41×10^{17}	1.55×10^{13}
8	1.72×10^{22}	2.28×10^{12}	6.88×10^{07}	5.75×10^{54}	2.84×10^{23}	8.23×10^{17}
9	1.12×10^{30}	5.38×10^{15}	2.14×10^{10}	2.63×10^{73}	8.92×10^{29}	2.41×10^{23}
10	1.67×10^{39}	3.12×10^{19}	1.53×10^{13}	1.56×10^{95}	1.67×10^{37}	3.93×10^{29}

Uniform generation in the bounding sets $\mathcal{B}_{\|\cdot\|_2}(\sqrt{n}, \mathbb{F}^{n,m})$ and $\mathcal{B}_{\|\cdot\|_\infty}(1, \mathbb{F}^{n,m})$ can be easily performed by means of the methods described in Sect. 18.1. Uniform sample generation in the set $\mathcal{B}_{\text{col}}(1, \mathbb{F}^{n,m})$ can also be easily obtained by generating independent columns $\boldsymbol{\xi}_i$, $i = 1, \ldots, m$, uniformly distributed in the ℓ_2 norm ball $\mathcal{B}_{\|\cdot\|_2}(1, \mathbb{F}^n)$.

The efficiency of the rejection method is dictated by the rejection rate, defined in Sect. 14.3.1 as the expected number of samples that should be generated in the outer set in order to have one sample in the set of interest $\mathcal{B}_\sigma$. In the case of uniform densities, the rejection rate is given by the ratio of the volumes of the outer bounding set and the set of interest.

The volume of the spectral norm ball has been derived in (17.25) and (17.37) for real and complex matrices respectively. The volumes of the ℓ_2 and ℓ_∞ Hilbert–Schmidt norm balls may be computed using Eqs. (15.8) and (15.13) for real and complex matrices respectively. As for the set $\mathcal{B}_{\text{col}}$, its volume may be derived as the product of the volumes of the m unit balls in the ℓ_2 norm in $\mathbb{F}^n$, obtaining

$$\text{Vol}\big(\mathcal{B}_{\text{col}}\big(1, \mathbb{R}^{n,m}\big)\big) = \frac{(\pi)^{nm/2}}{\Gamma^m(\frac{n}{2}+1)};$$

$$\text{Vol}\big(\mathcal{B}_{\text{col}}\big(1, \mathbb{C}^{n,m}\big)\big) = \frac{(\pi)^{nm}}{\Gamma^m(n+1)}.$$

Therefore, we can compute in closed form the rejection rates when different bounding sets are used. These rejection rates are reported in Table 18.1.

This table shows that, using $\mathcal{B}_{\text{col}}$ as the bounding set, we can construct "good" rejection schemes up to $n = 4$ for the real case and $n = 3$ for the complex case. For larger values of n, the rejection method becomes highly inefficient.

Next, we report two simple algorithms for generating uniform (real or complex) samples in the spectral norm ball by rejection from $\mathcal{B}_{\text{col}}(1, \mathbb{F}^{n,m})$.

Algorithm 18.1 (Uniform generation in $\mathcal{B}_\sigma(r, \mathbb{R}^{n,m})$ by rejection) *Given n, m and r this algorithm returns a random matrix $\mathbf{X} \in \mathbb{R}^{n,m}$ uniformly distributed in the (real) spectral norm ball of radius r.*

1. Generate m independent random columns $\boldsymbol{\xi}_i \sim \mathcal{U}_{\mathcal{B}_{\|\cdot\|_2}(\mathbb{R}^n)}$, $i = 1, \ldots, m$, using Algorithm 16.1;
2. Construct matrix $\mathbf{X} = [\boldsymbol{\xi}_1 \ \cdots \ \boldsymbol{\xi}_m]$;
3. If $\bar{\sigma}(\mathbf{X}) \leq 1$ return $r\mathbf{X}$ else goto 1.

Algorithm 18.2 (Uniform generation in $\mathcal{B}_\sigma(r, \mathbb{C}^{n,m})$ by rejection) *Given n, m and r this algorithm returns a random matrix $\mathbf{X} \in \mathbb{C}^{n,m}$ uniformly distributed in the (complex) spectral norm ball of radius r.*

1. Generate m independent random columns $\boldsymbol{\xi}_i \sim \mathcal{U}_{\mathcal{B}_{\|\cdot\|_2}(\mathbb{C}^n)}$, $i = 1, \ldots, m$, using Algorithm 16.4;
2. Construct matrix $\mathbf{X} = [\boldsymbol{\xi}_1 \ \cdots \ \boldsymbol{\xi}_m]$;
3. If $\bar{\sigma}(\mathbf{X}) \leq 1$ return $r\mathbf{X}$ else goto 1.

The inefficiency of the rejection method for large dimension motivates the need for more sophisticated techniques for direct generation of uniform samples, which are discussed in the next section. We first concentrate on the complex case, which turns out to be easier than the real one.

18.4 Uniform Generation of Complex Matrices

In this section we show how to generate uniform matrix samples $X \in \mathbb{C}^{n,m}$, by first generating the samples of the SVD factors U, Σ, V according to their respective densities, and then constructing $X = U\Sigma V^*$. We analyze the generation of Σ in the next section, and subsequently discuss a technique for generating U and V in Sect. 18.4.2.

18.4.1 Sample Generation of Singular Values

We recall that if the random matrix $\mathbf{X} \in \mathbb{C}^{n,m}$, $m \geq n$, is uniformly distributed over the set $\mathcal{B}_\sigma(\mathbb{C}^{n,m})$, then from Corollary 17.5 the pdf of $\boldsymbol{\Sigma}$ is

$$f_{\boldsymbol{\Sigma}}(\Sigma) = K_{\mathbb{C}} \prod_{i=1}^{n} \sigma_i^{2(m-n)+1} \prod_{1 \leq i < k \leq n} \left(\sigma_i^2 - \sigma_k^2\right)^2, \quad \sigma \in \mathcal{D}_\sigma \tag{18.3}$$

where the (ordered) domain $\mathcal{D}_\sigma$ is defined in (17.5) and the constant $K_{\mathbb{C}}$ is given in (17.36). For subsequent developments, it is useful to remove the ordering condition $\sigma \in \mathcal{D}_\sigma$ on the singular values, obtaining the (unordered) density function

$$f_{\boldsymbol{\Sigma}}(\Sigma) = \frac{K_{\mathbb{C}}}{n!} \prod_{i=1}^{n} \sigma_i^{2(m-n)+1} \prod_{1 \le i < k \le n} \left(\sigma_i^2 - \sigma_k^2\right)^2 \tag{18.4}$$

defined on the domain $\{\sigma \in \mathbb{R}^n : \sigma_i \in (0,1),\ i = 1, \ldots, n\}$. We remark that the factorial term $(n!)$ in this equation is obtained by observing that the ordered case is one of the $n!$ possible permutations of the n unordered singular values. For convenience, we introduce the change of variables

$$\varsigma_i = \sigma_i^2, \quad i = 1, \ldots, n.$$

The Jacobian of the transformation from the random variable $\boldsymbol{\varsigma}$ to $\boldsymbol{\sigma}$ is

$$J(\varsigma \to \sigma) = \frac{1}{2^n} \prod_{i=1}^{n} \varsigma_i^{-1/2}.$$

Then, applying Theorem 14.2 on the transformation between random variables, we obtain the density function of $\boldsymbol{\varsigma}$

$$f_{\boldsymbol{\varsigma}}(\varsigma) = \frac{K_{\mathbb{C}}}{n!2^n} \prod_{i=1}^{n} \varsigma_i^{m-n} \prod_{1 \le i < k \le n} (\varsigma_i - \varsigma_k)^2 \tag{18.5}$$

with domain $\{\varsigma \in \mathbb{R}^n : \varsigma_i \in (0,1), i = 1, \ldots, n\}$. This density function can be written in terms of the determinant of a Vandermonde matrix, as detailed in the following remark.

Remark 18.1 (Vandermonde determinant) Given $\varsigma = [\varsigma_1 \ \cdots \ \varsigma_n]^T$, define the vector

$$\mathcal{V}(\varsigma_i) \doteq \begin{bmatrix} 1 & \varsigma_i & \varsigma_i^2 & \cdots & \varsigma_i^{n-1} \end{bmatrix}^T, \quad i = 1, \ldots, n. \tag{18.6}$$

Then, the Vandermonde matrix $\mathcal{V}(\varsigma_1, \ldots, \varsigma_n)$ associated with vector ς is defined as

$$\mathcal{V}(\varsigma_1, \ldots, \varsigma_n) = \begin{bmatrix} \mathcal{V}(\varsigma_1) & \cdots & \mathcal{V}(\varsigma_n) \end{bmatrix}. \tag{18.7}$$

Similarly, for $i = 1, \ldots, n$ we define the *truncated* Vandermonde matrix as

$$\mathcal{V}(\varsigma_1, \varsigma_2, \ldots, \varsigma_i) = \begin{bmatrix} \mathcal{V}(\varsigma_1) & \cdots & \mathcal{V}(\varsigma_i) \end{bmatrix}. \tag{18.8}$$

For notational convenience, we write $\mathcal{V}_i$ to indicate $\mathcal{V}(\varsigma_1, \ldots, \varsigma_i)$. It is well known that the determinant of a Vandermonde matrix is given by

$$\det \mathcal{V}(\varsigma_1, \ldots, \varsigma_n) = \prod_{1 \le i < k \le n} (\varsigma_i - \varsigma_k).$$

Using this fact, it follows immediately that the density function (18.5) can be written as

$$f_{\boldsymbol{\varsigma}}(\varsigma) = \frac{K_{\mathbb{C}}}{n!2^n} \left|\mathcal{V}(\varsigma_1, \ldots, \varsigma_n)\right|^2 \prod_{i=1}^{n} \varsigma_i^{m-n}. \tag{18.9}$$

We now focus on the generation of random samples distributed according to (18.5). To this end, we apply the conditional density method introduced in Sect. 14.3.2. That is, we write the density (18.5) as

$$f_{\boldsymbol{\varsigma}_1, \ldots, \boldsymbol{\varsigma}_n}(\varsigma_1, \ldots, \varsigma_n) = f_{\boldsymbol{\varsigma}_1}(\varsigma_1) f_{\boldsymbol{\varsigma}_2|\boldsymbol{\varsigma}_1}(\varsigma_2|\varsigma_1) \cdots f_{\boldsymbol{\varsigma}_n|\boldsymbol{\varsigma}_1, \ldots, \boldsymbol{\varsigma}_{n-1}}(\varsigma_n|\varsigma_1, \ldots, \varsigma_{n-1})$$

where the conditional densities $f_{\boldsymbol{\varsigma}_i|\boldsymbol{\varsigma}_1,\ldots,\boldsymbol{\varsigma}_{i-1}}(\varsigma_i|\varsigma_1,\ldots,\varsigma_{i-1})$ are defined as the ratio of marginal densities

$$f_{\boldsymbol{\varsigma}_i|\boldsymbol{\varsigma}_1,\ldots,\boldsymbol{\varsigma}_{i-1}}(\varsigma_i|\varsigma_1,\ldots,\varsigma_{i-1}) = \frac{f_{\boldsymbol{\varsigma}_1,\ldots,\boldsymbol{\varsigma}_i}(\varsigma_1,\ldots,\varsigma_i)}{f_{\boldsymbol{\varsigma}_1,\ldots,\boldsymbol{\varsigma}_{i-1}}(\varsigma_1,\ldots,\varsigma_{i-1})}. \tag{18.10}$$

In turn, the marginal densities $f_{\boldsymbol{\varsigma}_1,\ldots,\boldsymbol{\varsigma}_i}(\varsigma_1,\ldots,\varsigma_i)$ are given by

$$f_{\boldsymbol{\varsigma}_1,\ldots,\boldsymbol{\varsigma}_i}(\varsigma_1,\ldots,\varsigma_i) = \int_0^1 \cdots \int_0^1 f_{\boldsymbol{\varsigma}}(\varsigma_1,\ldots,\boldsymbol{\varsigma}_n)\,\mathrm{d}\varsigma_{i+1}\cdots\mathrm{d}\boldsymbol{\varsigma}_n. \tag{18.11}$$

Therefore, a random vector $\boldsymbol{\varsigma}$ with density (18.5) can be obtained by generating sequentially the random variables $\boldsymbol{\varsigma}_i$, $i = 1,\ldots,n$, where $\boldsymbol{\varsigma}_i$ is distributed according to the univariate conditional density $f_{\boldsymbol{\varsigma}_i|\boldsymbol{\varsigma}_1,\ldots,\boldsymbol{\varsigma}_{i-1}}(\varsigma_i|\varsigma_1,\ldots,\varsigma_{i-1})$. The following theorem provides a closed-form expression for the marginal density (18.11), without requiring symbolic computation of the integral.

Theorem 18.1 *The marginal density* (18.11) *is equal to*

$$\begin{aligned} &f_{\boldsymbol{\varsigma}_1,\ldots,\boldsymbol{\varsigma}_i}(\varsigma_1,\ldots,\varsigma_i) \\ &\quad = K_{\mathbb{C}}\frac{(n-i)!}{n!2^n|H|^2}\left|\mathcal{V}(\varsigma_1,\ldots,\varsigma_i)^T H^T H\mathcal{V}(\varsigma_1,\ldots,\varsigma_i)\right|\prod_{k=1}^{i}\varsigma_k^{m-n} \end{aligned} \tag{18.12}$$

with $\varsigma_k \in (0,1)$, $k = 1,\ldots,i$, *and where* $\mathcal{V}(\varsigma_1,\ldots,\varsigma_i)$ *is defined in* (18.8), *and* $H \doteq R^{-T}$, *being* R *the upper-triangular factor of the Cholesky decomposition* $M = R^T R$ *of the symmetric matrix with entries*

$$[M]_{r,\ell} \doteq \frac{1}{r+\ell+m-n-1}, \quad r,\ell = 1,\ldots,n.$$

Proof Following the discussion in Remark 18.1, we rewrite (18.5) in the form (18.9)

$$f_{\boldsymbol{\varsigma}}(\varsigma) = \frac{K_{\mathbb{C}}}{n!2^n}|\mathcal{V}_n|^2\prod_{k=1}^{n}\varsigma_k^{m-n}$$

where $\mathcal{V}_n = \mathcal{V}(\varsigma_1,\ldots,\varsigma_n)$ is the Vandermonde matrix. Then, for any given nonsingular matrix $H \in \mathbb{R}^{n,n}$, we have that

$$f_{\boldsymbol{\varsigma}}(\varsigma) = \frac{K_{\mathbb{C}}}{n!2^n|H|^2}\left|\mathcal{V}_n^T H^T H\mathcal{V}_n\right|\prod_{k=1}^{n}\varsigma_k^{m-n}. \tag{18.13}$$

Notice further that

$$H\mathcal{V}(\varsigma) = \begin{bmatrix} L_0(\varsigma) & L_1(\varsigma) & L_2(\varsigma) & \cdots & L_{n-1}(\varsigma)\end{bmatrix}^T$$

where $L_k(\varsigma)$, $k = 0,\ldots,n-1$, are polynomials of degree $n-1$ in the variable ς

$$L_k(\varsigma) = h_{k,0} + h_{k,1}\varsigma + h_{k,2}\varsigma^2 + \cdots + h_{k,n-1}\varsigma^{n-1} \tag{18.14}$$

and $h_{r-1,\ell-1} = [H]_{r,\ell}$ denotes the (r,ℓ) entry of H.

We observe in particular that the matrix H can be chosen such that the polynomials $L_k(\varsigma)$, $k = 0,1,\ldots,n-1$, form an orthogonal polynomial basis on the interval $\varsigma \in [0,1]$, with respect to the weight function ς^{m-n}. That is, for $k,\ell = 0,1,\ldots,n-1$, we impose that

$$\int_0^1 L_k(\varsigma)L_\ell(\varsigma)\varsigma^{m-n}\,\mathrm{d}\varsigma = \begin{cases} 1 & \text{if } k=\ell; \\ 0 & \text{otherwise.} \end{cases} \tag{18.15}$$

This condition can be written in matrix form as

$$HMH^T = I_n \tag{18.16}$$

where

$$M = \left(\int_0^1 \mathcal{V}(\varsigma)\mathcal{V}^T(\varsigma)\varsigma^{m-n}\,\mathrm{d}\varsigma\right).$$

The integral term M is easily evaluated as

$$[M]_{r,\ell} = \frac{1}{r+\ell+m-n-1}$$

for $r, \ell = 1, \ldots, n$. Let $M = R^T R$ be the Cholesky decomposition of M, where R is upper triangular. Then, the orthogonality condition (18.16) is satisfied for the choice $H = R^{-T}$, where the resulting matrix H is lower triangular. Define now, for $i = 1, \ldots, n$, the symmetric matrix

$$Z_i = Z_i(\varsigma_1, \ldots, \varsigma_i) \doteq \mathcal{V}_i^T H^T H \mathcal{V}_i. \tag{18.17}$$

It is straightforward to show that the matrix Z_i satisfies the conditions of the Dyson–Mehta theorem for the integral of certain determinants; see Appendix A.4. In particular, we have that the (r, ℓ) entry of Z_i is function of $\varsigma_r, \varsigma_\ell$, i.e. $[Z_i]_{r,\ell} = \psi(\varsigma_r, \varsigma_\ell)$, with

$$\psi(\varsigma_r, \varsigma_\ell) = \sum_{k=0}^{n-1} L_k(\varsigma_r)L_k(\varsigma_\ell).$$

The conditions of the Dyson–Mehta theorem are met for the function $\psi(\varsigma_r, \varsigma_\ell)$ with $\mathrm{d}\mu(\varsigma) = \varsigma^{m-n}\mathrm{d}\varsigma$. In particular, we have

$$\int_0^1 \psi(\varsigma_i, \varsigma_i)\varsigma_i^{m-n}\,\mathrm{d}\varsigma_i = n.$$

Therefore, from Theorem A.3 in Appendix A.4 we obtain

$$\int_0^1 \det\big(Z_i(\varsigma_1, \ldots, \varsigma_i)\big)\varsigma_i^{m-n}\,\mathrm{d}\varsigma_i = (n-i+1)\det\big(Z_{i-1}(\varsigma_1, \ldots, \varsigma_{i-1})\big).$$

Applying this equation recursively, going backwards from n to $i+1$, and noticing that $|Z_i| = \det(Z_i)$ since $\det(Z_i)$ is always positive, we have

$$\begin{aligned} &\int_0^1 \cdots \int_0^1 \big|Z_n(\varsigma_1, \ldots, \varsigma_n)\big|(\varsigma_{i+1}\cdots\varsigma_n)^{m-n}\,\mathrm{d}\varsigma_{i+1}\cdots\mathrm{d}\varsigma_n \\ &\quad = (n-i)!\big|Z_i(\varsigma_1, \ldots, \varsigma_i)\big|. \end{aligned}$$

Then, by means of (18.13) we obtain the marginal density

$$f_{\varsigma_1,\ldots,\varsigma_i}(\varsigma_1, \ldots, \varsigma_i) = K_{\mathbb{C}}\frac{(n-i)!}{n!2^n|H|^2}\big|Z_i(\varsigma_1, \ldots, \varsigma_i)\big|\prod_{k=1}^{i}\varsigma_k^{m-n}. \tag{18.18}$$

The proof is then completed by substituting (18.17) in this expression. □

Remark 18.2 (Relationship with orthogonal polynomials) In the proof of Theorem 18.1, the polynomials (18.14) of degree $n-1$ in the variable ς

$$L_k(\varsigma) = h_{k,0} + h_{k,1}\varsigma + h_{k,2}\varsigma^2 + \cdots + h_{k,n-1}\varsigma^{n-1}$$

should satisfy the orthogonality condition (18.15). Indeed, such polynomial basis may be recognized as the family of Jacobi polynomials $G_{n-1}(p, q, \varsigma)$ with $p = q = (m-n+1)$. Therefore, the entries of the matrix H may be derived in closed form using the formulas in [4] for the coefficients of $G_{n-1}(p, q, \varsigma)$, thus obtaining

$$[H]_{r,\ell} = (-1)^{r-\ell}\binom{r}{\ell}\frac{\ell\sqrt{m-n+2r-1}}{r!}\frac{\Gamma(m-n+r+\ell-1)}{\Gamma(m-n+\ell)} \tag{18.19}$$

for $r = 1, \ldots, n$, $\ell = 1, \ldots, r$, and $[H]_{r,\ell} = 0$ for $r > \ell$. From this expression, we immediately obtain

$$|H| = \prod_{\ell=1}^{n} \frac{\sqrt{m-n+2\ell-1}}{(\ell-1)!}\frac{\Gamma(m-n+2\ell-1)}{\Gamma(m-n+\ell)}.$$

Theorem 18.1 provides a closed-form expression for the multiple integral (18.11). To apply the conditional density method, we need to compute recursively the conditional density (18.10) of the random variable $\boldsymbol{\varsigma}_i$ when the values $\varsigma_1, \ldots, \varsigma_{i-1}$ are given. This is shown in the following corollary.

Corollary 18.1 *Let $i = 2, \ldots, n$. Then, the conditional density of $\boldsymbol{\varsigma}_i$ given $\boldsymbol{\varsigma}_1 = \varsigma_1$, $\boldsymbol{\varsigma}_2 = \varsigma_2, \ldots, \boldsymbol{\varsigma}_{i-1} = \varsigma_{i-1}$, is a polynomial of order $2(n-1)$ expressed as*

$$f_{\boldsymbol{\varsigma}_i|\varsigma_1,\ldots,\varsigma_{i-1}}(\varsigma_i|\varsigma_1, \ldots, \varsigma_{i-1}) = (n-i+1)^{-1}\varsigma_i^{m-n}\sum_{k=0}^{2(n-1)} b_{i,k}\varsigma_i^k. \tag{18.20}$$

The coefficients $b_{i,k} = b_{i,k}(\varsigma_i|\varsigma_1, \ldots, \varsigma_{i-1})$, $k = 0, 1, \ldots, 2(n-1)$, are given by

$$b_{i,k} \doteq \sum_{\{r+\ell=k+2\}} [W_{i-1}]_{r,\ell} \tag{18.21}$$

where

$$W_{i-1} \doteq H^T\left(I - H\mathcal{V}_{i-1}Z_{i-1}^{-1}\mathcal{V}_{i-1}^T H^T\right)H; \tag{18.22}$$

$$Z_{i-1} \doteq \mathcal{V}_{i-1}^T H^T H\mathcal{V}_{i-1} \tag{18.23}$$

and H is given in (18.19). Moreover, the marginal density of $\boldsymbol{\varsigma}_1$ is given by the polynomial

$$f_{\boldsymbol{\varsigma}_1}(\varsigma_1) = \frac{K_{\mathbb{C}}}{n2^n|H|^2}\varsigma_1^{m-n}\sum_{k=0}^{2(n-1)} b_{1,k}\varsigma_1^k \tag{18.24}$$

where $b_{1,k} \doteq \sum_{\{r+\ell=k+2\}}[W_0]_{r,\ell}$ and $W_0 = H^T H$.

Proof As in (18.17) in the proof of Theorem 18.1, we define

$$Z_i = Z_i(\varsigma_1, \ldots, \varsigma_i) \doteq \mathcal{V}_i^T H^T H\mathcal{V}_i.$$

Next, recalling that $\mathcal{V}_i = [\mathcal{V}_{i-1}\mathcal{V}(\varsigma_i)]$, we obtain

$$|Z_i| = \begin{vmatrix} Z_{i-1} & \mathcal{V}_{i-1}^T H^T H \mathcal{V}(\varsigma_i) \\ \mathcal{V}^T(\varsigma_i) H^T H \mathcal{V}_{i-1} & \mathcal{V}^T(\varsigma_i) H^T H \mathcal{V}(\varsigma_i) \end{vmatrix}.$$

Using the Schur rule for this determinant, for $i = 2, \ldots, n$, we get

$$\big|Z_i(\varsigma_1, \ldots, \varsigma_i)\big| = \big|Z_{i-1}(\varsigma_1, \ldots, \varsigma_{i-1})\big| \mathcal{V}^T(\varsigma_i) W_{i-1} \mathcal{V}(\varsigma_i) \tag{18.25}$$

where

$$W_{i-1} = H^T \big(I - H\mathcal{V}_{i-1} Z_{i-1}^{-1} \mathcal{V}_{i-1}^T H^T\big) H.$$

The term $\mathcal{V}^T(\varsigma_i) W_{i-1} \mathcal{V}(\varsigma_i)$ can be written as a polynomial in the variable ς_i, with coefficients depending on $\varsigma_1, \ldots, \varsigma_{i-1}$. It is straightforward to verify that these coefficients are given by the sum of the elements of the anti-diagonals of W_{i-1}. That is, we have

$$\mathcal{V}^T(\varsigma_i) W_{i-1} \mathcal{V}(\varsigma_i) = \sum_{k=0}^{2(n-1)} b_{i,k} \varsigma_i^k \tag{18.26}$$

where, for $k = 0, 1, \ldots, 2(n-1)$,

$$b_{i,k} = \sum_{\{r+\ell=k+2\}} [W_{i-1}]_{r,\ell}.$$

Moreover, $|Z_1(\varsigma_1)| = \mathcal{V}_1^T W_0 \mathcal{V}_1$, with $W_0 = H^T H$.

Combining the expressions (18.18), (18.25) and (18.26) we obtain that the marginal density (18.11) is given by

$$f_{\varsigma_1,\ldots,\varsigma_i}(\varsigma_1, \ldots, \varsigma_i) = K_{\mathbb{C}} \frac{(n-i)!}{n! 2^n |H|^2} \varsigma_i^{m-n} \sum_{k=0}^{2(n-1)} b_{i,k} \varsigma_i^k.$$

Taking $i = 1$ in the latter expression, we prove (18.24). Equation (18.20) is then immediately obtained using the definition of conditional density (18.10). □

Remark 18.3 (Application of the conditional density method) Corollary 18.1 gives an expression of the conditional densities in the form of the polynomial (18.20), where the variables $\varsigma_1, \ldots, \varsigma_{i-1}$ are separated from the variable ς_i. In fact, at the ith step of the conditional density method the variables up to $i-1$ are given, and only the dependence on ς_i is required. In this case, Eq. (18.20) represents a polynomial in the ς_i variable. The coefficients $b_{i,k} = b_{i,k}(\varsigma_1, \ldots, \varsigma_{i-1})$ can be easily computed according to (18.21), once the values of $\varsigma_1, \varsigma_2, \ldots, \varsigma_{i-1}$ are known.

Remark 18.4 (Computational improvements) We observe that, for $i \geq 2$, the factor Z_{i-1}^{-1} appearing in (18.22) can be computed recursively, so that no matrix inversion or determinant computation is required. In particular, we have

$$Z_i = \mathcal{V}_i^T H^T H \mathcal{V}_i = \begin{bmatrix} Z_{i-1} & \mathcal{V}_{i-1}^T H^T H \mathcal{V}(\varsigma_i) \\ \mathcal{V}^T(\varsigma_i) H^T H \mathcal{V}_{i-1} & \mathcal{V}^T(\varsigma_i) H^T H \mathcal{V}(\varsigma_i) \end{bmatrix}.$$

Using the block matrix inversion formula, we obtain

$$Z_i^{-1} = \begin{bmatrix} Z_{i-1}^{-1} + \Omega_i \Omega_i^T/\delta_i & -\Omega_i/\delta_i \\ -\Omega_i^T/\delta_i & 1/\delta_i \end{bmatrix} \tag{18.27}$$

where $\Omega_i = Z_{i-1}^{-1} \mathcal{V}_{i-1}^T H^T H \mathcal{V}(\varsigma_i)$ and $\delta_i = \mathcal{V}^T(\varsigma_i) W_{i-1} \mathcal{V}(\varsigma_i) > 0$.

The generation of samples of the singular values distributed according to the pdf (18.4) is reported in Algorithm 18.3.

In this algorithm, each $\boldsymbol{\varsigma}_i$ is generated according to a univariate polynomial density. Standard and efficient algorithms for the generation of samples distributed according to a given polynomial density are available in the literature. Among these techniques we recall a classical one based on the inversion method presented in Algorithm 14.1.

Algorithm 18.3 (Singular values generation) *Given n, m, $m \geq n$, this algorithm returns a random vector $\boldsymbol{\sigma} = [\sigma_1 \ \cdots \ \sigma_n]^T$ distributed according to the pdf* (18.4).

1. Initialization.
   - ▹ Set $i = 1$, $W_0 = H^T H$, where H is given in (18.19);
   - ▹ Let $b_{1,k} \doteq \sum_{\{r+\ell=k+2\}} [W_0]_{r,\ell}$;
   - ▹ Generate $\boldsymbol{\varsigma}_1$ according to the polynomial marginal density

$$f_{\boldsymbol{\varsigma}_1}(\varsigma_1) = \frac{K_{\mathbb{C}}}{n 2^n |H|^2} \varsigma_1^{m-n} \sum_{k=0}^{2(n-1)} b_{1,k} \varsigma_1^k;$$

   - ▹ Let $Z_1 = \mathcal{V}^T(\boldsymbol{\varsigma}_1) W_0 \mathcal{V}(\boldsymbol{\varsigma}_1)$;
2. Update.
   - ▹ Set $\delta_i = \mathcal{V}^T(\boldsymbol{\varsigma}_i) W_{i-1} \mathcal{V}(\boldsymbol{\varsigma}_i)$ and $\Omega_i = Z_{i-1}^{-1} \mathcal{V}_{i-1}^T H^T H \mathcal{V}(\boldsymbol{\varsigma}_i)$;
   - ▹ Compute

$$Z_i^{-1} = \begin{bmatrix} Z_{i-1}^{-1} + \Omega_i \Omega_i^T/\delta_i & -\Omega_i/\delta_i \\ -\Omega_i^T/\delta_i & 1/\delta_i \end{bmatrix};$$
$$W_i = H^T \left(I - H \mathcal{V}_i Z_i^{-1} V_i^T H^T \right) H;$$

3. Generation.
   - ▹ Set $b_{i,k} = \sum_{\{r+\ell=k+2\}} [W_{i-1}]_{r,\ell}$, $k = 1, \ldots, 2(n-1)$;
   - ▹ Generate $\boldsymbol{\varsigma}_i \in (0, 1)$ according to the polynomial density

$$f_{\boldsymbol{\varsigma}_i | \boldsymbol{\varsigma}_1, \ldots, \boldsymbol{\varsigma}_{i-1}}(\varsigma_i | \boldsymbol{\varsigma}_1, \ldots, \boldsymbol{\varsigma}_{i-1}) = (n-i+1)^{-1} \varsigma_i^{m-n} \sum_{k=0}^{2(n-1)} b_{i,k} \boldsymbol{\varsigma}_i^k;$$

4. Loop.
   - ▹ If $i < n$, set $i = i + 1$ and goto 2;
5. Return $\boldsymbol{\sigma} = [\sqrt{\boldsymbol{\varsigma}_1} \ \cdots \ \sqrt{\boldsymbol{\varsigma}_n}]^T$.

18.4.2 Uniform Generation of Unitary Matrices

In this section we concentrate on the generation of samples of U and V according to the Haar invariant distribution (17.28) and to the conditional Haar invariant distribution (17.30) respectively.

We first consider the problem of generating a random matrix $\mathbf{U}$ uniformly distributed in the unitary group $\mathcal{G}_{\mathcal{U}}^n$. From the properties of the Haar invariant distribution, we have that the distribution of $\mathbf{U}$ should be the same as the distribution of $W\mathbf{U}$, for any given unitary matrix W. Consider a random matrix $\mathbf{X} \in \mathbb{C}^{n,n} = \mathrm{Re}(\mathbf{X}) + j\mathrm{Im}(\mathbf{X})$, such that the entries of $\mathrm{Re}(\mathbf{X})$ and $\mathrm{Im}(\mathbf{X})$ are independent and normally distributed with zero mean and variance equal to one. The invariance property of the normal distribution under unitary transformations implies that, for any unitary matrix W, the distribution of $W\mathbf{X}$ is the same of the distribution of $\mathbf{X}$. Now, let $\mathbf{X} = \mathbf{QR}$ be the QR factorization of $\mathbf{X}$, where the diagonal entries of $\mathbf{R}$ are forced to be real and positive in order to make the representation unique. Then, since $W\mathbf{X} \sim \mathbf{X}$, it follows that $W\mathbf{Q} \sim \mathbf{Q}$. That is, $\mathbf{Q}$ is distributed according to the Haar invariant distribution.

This discussion suggests the following simple algorithm for the generation of samples according to the Haar invariant distribution.

Algorithm 18.4 (Generation of Haar unitary matrices) *Given n, the algorithm returns a sample of the random unitary matrix $\mathbf{U} \in \mathbb{C}^{n,n}$ distributed according to the Haar invariant distribution* (17.28).

1. Generation of Gaussian $\mathbf{Y}$.
   - ▷ Generate $\mathbf{Y}^{\mathbb{R}}, \mathbf{Y}^{\mathbb{I}} \in \mathbb{R}^{n,n}$, where each entry of $\mathbf{Y}^{\mathbb{R}}$ and $\mathbf{Y}^{\mathbb{I}}$ is distributed according to $\mathcal{N}_{0,1}$;
   - ▷ Construct $\mathbf{Y} = \mathbf{Y}^{\mathbb{R}} + j\mathbf{Y}^{\mathbb{I}}$;
2. QR factorization.
   - ▷ Factorize $\mathbf{Y}$ as $[\mathbf{Q}, \mathbf{R}] = \mathrm{QR}(\mathbf{Y})$;
   - ▷ Set $\mathbf{U} = \mathbf{Q}\,\mathrm{diag}([\mathrm{e}^{-j\boldsymbol{\phi}_1} \ \cdots \ \mathrm{e}^{-j\boldsymbol{\phi}_n}])$, where $\boldsymbol{\phi}_i$ is the phase of the (i,i) entry of $\mathbf{R}$;
3. Return $\mathbf{U}$.

This algorithm is one of the simplest methods for generation of uniform unitary matrices. Other known methods are based, for example, on products of elementary Euler transformations; see e.g. [424]. We remark that the Haar invariant distribution may also be introduced for rectangular random matrices $\mathbf{V} \in \mathbb{C}^{m,n}$ having orthonormal columns. In this case, it can be observed that uniform samples of V can be obtained from uniform samples of a square $m \times m$ unitary matrix, using Algorithm 18.4, and simply neglecting the last $m - n$ columns. This fact follows from properties of the unitary group, see for instance [210].

Finally, we consider the generation of random matrices $\mathbf{V}$ with conditional Haar invariant distribution, i.e. uniform in the manifold $\mathcal{C}^{m,n}$ defined in (17.27). We notice that the normalization condition on the columns of $\mathbf{V}$ may be written as $\mathbf{V} = \widetilde{\mathbf{V}}\boldsymbol{\Theta}$, where $\widetilde{\mathbf{V}} \in \mathbb{C}^{m,n}$ is a (non-normalized) unitary matrix and $\boldsymbol{\Theta}$ is a diagonal unitary matrix $\boldsymbol{\Theta} = \text{diag}([\text{e}^{-j\boldsymbol{\theta}_1} \ \cdots \ \text{e}^{-j\boldsymbol{\theta}_n}])$, where $\boldsymbol{\theta}_i$ is the phase of $[\widetilde{\mathbf{V}}]_{1,i}$. It can be shown that if $\widetilde{\mathbf{V}}$ is distributed according to the Haar invariant distribution, then $\mathbf{V}$ is distributed according to the conditional Haar invariant distribution. Samples of matrices drawn from the latter distribution may, therefore, be obtained by normalizing the samples drawn from the Haar invariant distribution.

We conclude this section by reporting the algorithm for direct generation of complex random matrices uniformly distributed in the spectral norm ball.

Algorithm 18.5 (Uniform generation in $\mathcal{B}_\sigma(r, \mathbb{C}^{n,m})$) *Given n, m, $m \geq n$, and r this algorithm returns a sample of the random matrix $\mathbf{X} \in \mathbb{C}^{n,m}$ with uniform distribution in the (complex) spectral norm ball of radius r.*

1. Generation of $\boldsymbol{\Sigma}$.
   - ▷ Generate $\boldsymbol{\sigma} = [\sigma_1 \ \cdots \ \sigma_n]^T$ using Algorithm 18.3;
   - ▷ Construct $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma})$;
2. Generation of $\mathbf{U}$ and $\mathbf{V}$.
   - ▷ Generate $\mathbf{U} \in \mathbb{C}^{n,n}$ and $\mathbf{V} \in \mathbb{C}^{m,n}$ using Algorithm 18.4;
3. Return $\mathbf{X} = r\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$.

18.5 Uniform Generation of Real Matrices

Similar to the developments of the complex case, we now study the generation of uniform matrix samples in the ball $\mathcal{B}_\sigma(\mathbb{R}^{n,m})$. This approach is based on the generation of the samples of the SVD factors $\mathbf{U}$, $\boldsymbol{\Sigma}$, $\mathbf{V}$ according to their respective densities derived in Corollary 17.4, and then on the construction of $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$. In the next section, we analyze the generation of $\boldsymbol{\Sigma}$, and subsequently we discuss a technique for generating $\mathbf{U}$ and $\mathbf{V}$ in Sect. 18.5.2.

18.5.1 Sample Generation of Singular Values

If we assume that the random matrix $\mathbf{X} \in \mathbb{R}^{n,m}$, $m \geq n$, is uniformly distributed over the set $\mathcal{B}_\sigma(\mathbb{R}^{n,m})$, then it follows from Corollary 17.4 that the pdf of $\boldsymbol{\Sigma}$ is given by

$$f_{\boldsymbol{\Sigma}}(\Sigma) = K_{\mathbb{R}} \prod_{i=1}^{n} \sigma_i^{m-n} \prod_{1 \leq i < k \leq n} \left(\sigma_i^2 - \sigma_k^2\right), \quad \sigma \in \mathcal{D}_\sigma \tag{18.28}$$

where the (ordered) domain $\mathcal{D}_\sigma$ is defined in (17.5) and the constant $K_{\mathbb{R}}$ is given in (17.24). As in Sect. 18.4, we introduce the change of variables

$$\varsigma_i = \sigma_i^2, \quad i = 1, \dots, n$$

so that the density may be written in terms of a Vandermonde determinant, see Remark 18.1. Therefore, we obtain

$$f_{\boldsymbol{\varsigma}}(\varsigma_1, \dots, \varsigma_n) = \frac{K_{\mathbb{R}}}{2^n} \left| \mathcal{V}(\varsigma_1, \dots, \varsigma_n) \right| \prod_{i=1}^{n} \varsigma_i^{\nu}, \quad \nu = (m - n - 1)/2 \tag{18.29}$$

defined over the ordered domain

$$\mathcal{D}_\varsigma \doteq \left\{ \varsigma \in \mathbb{R}^n : 1 > \varsigma_1 > \cdots > \varsigma_n > 0 \right\}. \tag{18.30}$$

Remark 18.5 (Ordering condition on the singular values) We recall that in the complex case the ordering condition $\boldsymbol{\sigma} \in \mathcal{D}_\sigma$ has been removed to facilitate the computation of the marginal density. In the real case, the same approach is not helpful. This is due to the different form of the pdf, and in particular to the fact that in (18.29) the Vandermonde factor $|\mathcal{V}(\varsigma_1, \dots, \varsigma_n)|$ appears without the square power, which is present in (18.9).

To generate a vector $\boldsymbol{\varsigma}$ distributed according to the density (18.29), we apply the conditional density method described in Sect. 14.3.2. To this end, we need to compute the marginal density

$$f_{\boldsymbol{\varsigma}_1, \dots, \boldsymbol{\varsigma}_i}(\varsigma_1, \dots, \varsigma_i) = \frac{K_{\mathbb{R}}}{2^n} \mathcal{I}(\varsigma_1, \dots, \varsigma_i) \prod_{k=1}^{i} \varsigma_k^{\nu} \tag{18.31}$$

where

$$\mathcal{I}(\varsigma_1, \dots, \varsigma_i) = \int \cdots \int \left| \mathcal{V}(\varsigma_1, \dots, \varsigma_n) \right| \prod_{k=i+1}^{n} \varsigma_k^{\nu} \, \mathrm{d}\varsigma_k \tag{18.32}$$

with domain of integration $\{\varsigma_i > \cdots > \varsigma_n > 0\}$, and $\nu = (m - n - 1)/2$.

The following theorem gives an explicit closed-form solution for the multiple integral (18.32).

Theorem 18.2 *The multiple integral* (18.32) *is given by*

$$\mathcal{I}(\varsigma_1, \dots, \varsigma_i) = \varsigma_i^{\alpha_i} \det{}^{1/2} \left[\begin{array}{c|c} Z(\varsigma_i) & \begin{matrix} \mathcal{V}(\varsigma_1, \dots, \varsigma_{i-1}) \\ 0 \end{matrix} \\ \hline \begin{matrix} -\mathcal{V}^T(\varsigma_1, \dots, \varsigma_{i-1}) & 0 \end{matrix} & 0 \end{array} \right] \tag{18.33}$$

for $i = 2, \dots, n$, *and*

$$\mathcal{I}(\varsigma_1) = \varsigma_1^{\alpha_1} \det{}^{1/2} Z(\varsigma_1)$$

where $\alpha_i \doteq (\nu + 1)(n - i)$, *and*

$$Z(\varsigma_i) \doteq \begin{cases} \begin{bmatrix} H(\varsigma_i) & \mathcal{V}(\varsigma_i) \\ -\mathcal{V}^T(\varsigma_i) & 0 \end{bmatrix} & \text{if } n-i \text{ even}; \\ \begin{bmatrix} H(\varsigma_i) & \mathcal{V}(\varsigma_i) & h(\varsigma_i) \\ -\mathcal{V}^T(\varsigma_i) & 0 & 0 \\ -h^T(\varsigma_i) & 0 & 0 \end{bmatrix} & \text{if } n-i \text{ odd}; \end{cases} \tag{18.34}$$

$$\left[H(\varsigma_i)\right]_{r,\ell} \doteq \frac{r-\ell}{(r+\nu)(\ell+\nu)(r+\ell+2\nu)} \varsigma_i^{r+\ell-2}, \quad r,\ell = 1,\ldots,n; \tag{18.35}$$

$$h_\ell(\varsigma_i) \doteq \frac{\varsigma_i^{\ell-1}}{\ell+\nu}, \quad \ell = 1,\ldots,n. \tag{18.36}$$

Remark 18.6 (Theory of skew-symmetric matrices) The proof of this result, given in [81], is quite involved and it is based on the theory of skew-symmetric matrices and, in particular, on the so-called de Bruijn integral, see [126]. Notice that the matrix appearing in (18.33) is a skew-symmetric polynomial matrix of even order and, therefore, its determinant is always a perfect square in the entries of the matrix, see e.g. [403]. We remark that the square root of the determinant of a skew-symmetric matrix of even order is related to the so-called Pfaffian, see e.g. [281] for definitions and properties.

Remark 18.7 (Algorithm for generation of the singular values) Theorem 18.2 provides a closed-form expression for the marginal density (18.31), so that the conditional density method can be applied. This result has been exploited in [81] to develop an efficient recursive algorithm for the generation of the singular values. This algorithm is similar to Algorithm 18.3, but more complicated. In fact, four different cases, corresponding to the combinations of n and i in (18.31) being even or odd, need to be considered. The main feature of the algorithm given in [81] is to reduce the sample generation of a vector $\boldsymbol{\sigma}$ distributed according to (18.28) to n univariate generations according to a polynomial density. The only operations required are polynomial matrix additions and multiplications, and no polynomial matrix inversion or computation of determinants are needed. As previously observed, the generation of samples according to the resulting univariate polynomial density may be performed very efficiently. For illustrative purposes, some plots of conditional densities for square real matrices of order $n = 4, 5, 6$ are shown in Fig. 18.1.

18.5.2 Uniform Generation of Orthogonal Matrices

In this section we present an algorithm, similar to Algorithm 18.4, for the generation of orthogonal Haar matrices. This algorithm is based on the QR decomposition and is also reported in [369].

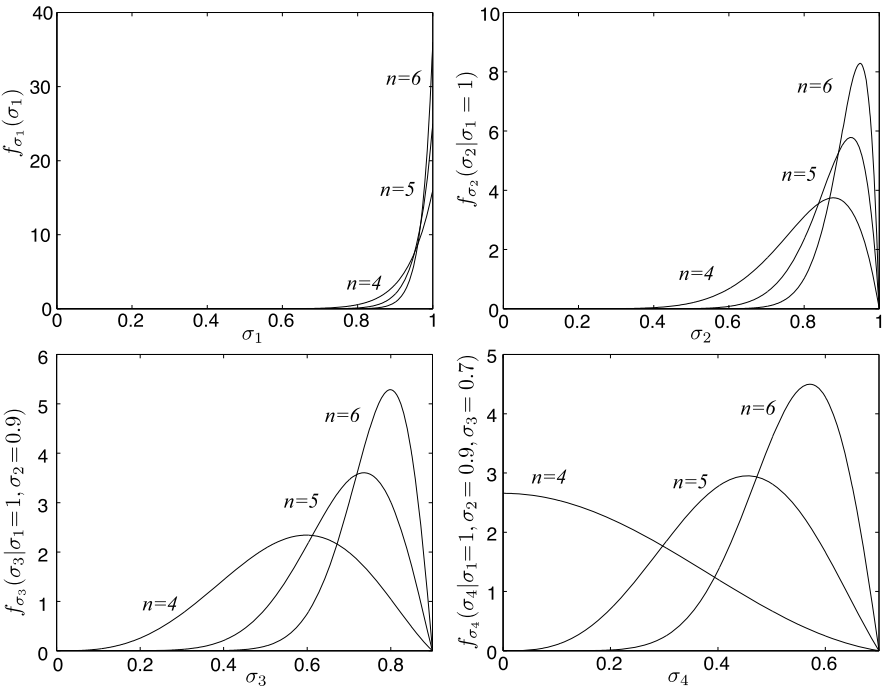

Fig. 18.1 Conditional probability densities of the first four singular values for uniform real square matrices of order $n = 4, 5, 6$

Algorithm 18.6 (Generation of Haar orthogonal matrices) *Given* n, *this algorithm returns a sample of the random orthogonal matrix* $\mathbf{U} \in \mathbb{R}^{n,n}$ *distributed according to the Haar invariant distribution* (17.14).

1. Generation of Gaussian $\mathbf{Y}$.
   - ▷ Generate $\mathbf{Y} \in \mathbb{R}^{n,n}$, where each entry of $\mathbf{Y}$ is $\mathcal{N}_{0,1}$;
2. QR factorization.
   - ▷ Factorize $\mathbf{Y}$ *as* $[\mathbf{Q}, \mathbf{R}] = \mathrm{QR}(\mathbf{Y})$;
   - ▷ Set $\mathbf{U} = \mathbf{Q}\,\mathrm{diag}([\mathbf{s}_1 \ \cdots \ \mathbf{s}_n])$, where $\mathbf{s}_i$ is the sign of the (i, i) entry of $\mathbf{R}$;
3. Return $\mathbf{U}$.

Regarding the generation of samples distributed according to the conditional Haar invariant distribution, i.e. uniform in the manifold $\mathcal{R}^{m,n}$ defined in (17.4), comments similar to those made for the complex case in Sect. 18.4.2 also apply here.

We conclude this chapter by reporting the algorithm for direct generation of real random matrices uniformly distributed in the spectral norm ball.

Algorithm 18.7 (Uniform generation in $\mathcal{B}_\sigma(r, \mathbb{R}^{n,m})$) *Given n, m, $m \geq n$, and r this algorithm returns a random matrix $\mathbf{X} \in \mathbb{R}^{n,m}$ with uniform distribution in the* (*real*) *spectral norm ball of radius r.*

1. Generation of $\boldsymbol{\Sigma}$.
   - ▷ Generate $\boldsymbol{\sigma} = [\sigma_1 \ \cdots \ \sigma_n]^T$ using the algorithm in [81];
   - ▷ Construct $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma})$;
2. Generation of $\mathbf{U}$ and $\mathbf{V}$.
   - ▷ Generate $\mathbf{U} \in \mathbb{R}^{n,n}$ and $\mathbf{V} \in \mathbb{R}^{m,n}$ using Algorithm 18.6;
3. Return $\mathbf{X} = r\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$.

Chapter 19
Applications of Randomized Algorithms

Probabilistic design methods and randomized algorithms have been developed for several applications related to systems and control. In this chapter, we first present an overview, which also provides pointers to the relevant literature, of some of the main areas where these methods have been successfully used. However, we do not discuss computer science, computational geometry and optimization, since applications in these areas are extensively covered, for instance, in [288, 290, 293]. Subsequently, we provide a more detailed description of some selected applications, which include the computation of PageRank in the Google search engine, control design of UAVs (unmanned aerial vehicles), congestion control of high-speed communication networks, robustness of flexible structures and quadratic stability of sampled-data quantized systems.

19.1 Overview of Systems and Control Applications

Multi-agent Systems and PageRank Computation The current approaches addressing distributed control over networks concentrate on traffic management of multiple vehicle systems. Problems in this domain include, for example, traffic control of autonomous vehicles, and may have different and possibly conflicting objectives (e.g., free cruise assistance for boats near the harbor, collision avoidance between cars at street intersections or platoon formation of underwater vehicles). Multi-agent systems may offer obvious significant advantages with respect to single-agent systems, such as speedup in task execution and robustness with respect to failure of one or more agents. On the other hand, a multi-agent approach is conceptually and algorithmically much more involved, in particular when dealing with distributed information, choice of communication protocols, verification and validation of decentralized control laws and coordination between agents. In [316] a decentralized approach for studying collision avoidance of a large number of autonomous vehicles moving in an uncertain environment is considered. The proposed method is based on a classical Monte Carlo analysis to provide a probabilistic assessment on

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0_19,

the satisfaction of certain design specifications. In [307] a Monte Carlo analysis for distributed abstract optimization via constraint consensus is presented, while partitioning and coverage control for gossiping robots is studied in [151]. Quantized consensus and averaging is studied in [77]. Distributed filtering with $\mathcal{H}_\infty$ consensus of estimates subject to LMI constraints is studied in [394] using a gradient-based approach.

An apparently unrelated problem deals with ranking web pages for facilitating the search of specific engines such as Google or Yahoo!. The core of the problem, as originally stated in [72], requires the computation of the eigenvector (called the PageRank) corresponding to the largest eigenvalue of a positive stochastic matrix representing the link structure of the web. It is reported that PageRank is computed at Google once a month, and this operation requires about a week due to the size of the link matrix which currently consists of at least 10 billion pages. The computation is performed centrally at Google, where the data on the whole web structure is collected by crawlers continuously browsing the web, see [252]. A decentralized randomized algorithm having the objective of facilitating the computation of PageRank is proposed in [215]. In the same paper, the close connections between PageRank and multi-agent consensus problems are also outlined. In [216] a technique for aggregating the web pages into groups by exploiting the sparsity of the web is presented, and in [164] a general class of PageRank optimization problems which consists in finding an optimal outlink strategy for a website subject to design constraints is discussed. Other relevant contributions include [128] and [294].

Systems Biology Systems biology is receiving increasing attention within the control community, see the special issues [15, 232]. In this area, a key problem is to construct hypothesis that can be subsequently tested by means of biological experiments. Probabilistic models are particularly appealing in this context because of the large number of uncertainty factors that are present in the experiments. In [242] the objective is to study the so-called genomic machine, which is a set of genes that cooperate to achieve a common objective. The random walk algorithm NetWalk is introduced to score the relevance of each interaction using a given set of data and also information regarding the local connectivity of the biological network. In [407] this genomic machine problem is formulated in the identification context using a Markov chain approach. This requires the computation of the stationary distribution of the chain and a flow parameter called the "doublet frequency." For a number of genes (or genes products) of the order 20,000 to 30,000, the PageRank algorithm [72, 215] is analyzed for performing this ranking and for a computation with reduced complexity. In particular, in [407] the goal is to develop randomized algorithms for *approximately* computing the stationary distribution and the doublet frequency. The approximation should be sufficiently accurate so that the sign of the figure of merit (which represents the flow of the connection between genes) is correctly computed. The PageRank algorithm has been used for protein ranking in [419].

Aerospace Control Aerospace control is a particularly appealing area of application of randomized methods. In this area of research, a probabilistic setup is

appropriate to describe changes in the flight conditions, uncertainties in the aerodynamic model, and inaccuracies in geometric and inertial data. In addition, since the mathematical plant models are generally obtained via linearization of the full-order nonlinear system representing aircraft dynamics, analytic relationships between the state space matrices of the linearized system and the uncertain parameters are not available. In these cases, simulation-based methods and lookup tables are often used.

The first paper studying the impact of parameter variations into the lateral-directional stability of aircraft by means of a randomized approach is probably [366]. In this and other subsequent papers, see, e.g., [275, 332, 411], various techniques, mainly based on Monte Carlo simulations, have been successfully utilized for the computation of the so-called probability of instability, and related performance criteria. More sophisticated design techniques, based on statistical learning theory, have been developed in [404] for the longitudinal stabilization of an unstable fighter aircraft. For unmanned aerial vehicles for surveillance and fire detection missions, other randomized methods are proposed in [266]. A different line of research, based on linear parameter-varying (LPV) methods, is discussed in [269] for the control of an F-16 aircraft. Applications of Monte Carlo methods for conflict resolution in air traffic control have been studied in [257]. In particular, optimization of the expected value of a given resolution criterion is carried out through an iterative procedure based on Markov chain Monte Carlo. In [174], the scenario approach is developed for H_∞ controller design of a multivariable LV100 gas turbine engine represented by a continuous-time state space model with five states, two inputs and two outputs.

Control of Hard Disk Drives Presently, it is estimated that more than 52 % of the entire data storage is dedicated to hard disk drives [201] and the gap between the available capacity and the amount of digital data that is generated worldwide is dramatically increasing. In order to reduce this gap, it is necessary to augment the so-called aerial density by designing more accurate control positioning algorithms which provide higher track per inch. Unfortunately, classical control techniques based on PID or Lead-Lag, which are currently used in disk drive servo systems, have already reached their limits of performance, and the development of more advanced techniques and algorithms is needed. A very interesting line of research regarding randomized sequential algorithms for track following servo control design of hard disk drives has been recently initiated in [97–99]. We refer to these papers for further details and experimental results.

Congestion Control of High-Speed Networks High-speed communication networks have received increasing attention in the control literature. One of the critical issues at the heart of efficient operations is congestion control, in particular for Available Bit Rate (ABR) in Asynchronous Transmission Mode (ATM) networks. The objective is to develop feedback controllers for improving the so-called Quality of Service (QoS). This involves the problem of regulating the source rates in a decentralized and distributed fashion, so that the available bandwidths on different links are used more efficiently while minimizing, or totally eliminating, loss of

packets due to queues at buffers exceeding their capacities. This goal needs to be accomplished under variations in network conditions (which can be probabilistically described) such as packet delays due to propagation as well as to queuing and bottleneck nodes.

Stability and QoS performance of communication networks have been addressed in [17] using simulation techniques. Specific randomized algorithms, based on Monte Carlo as well as on Quasi-Monte Carlo methods, have been developed for various network topologies. The results obtained provide a complete analysis of congestion control algorithms for Internet style networks with a single bottleneck node as well as for networks with more general random topologies. In [2] congestion control in the ABR class of ATM networks is studied by means of probabilistic methods and statistical learning theory. In this paper, a probabilistic robust controller with fixed structure for a high-speed communication network with multiple uncertain propagation delays is designed.

Stability of Quantized and Switched Systems When dealing with feedback loops that include digital channels, the need for quantization of control signals inevitably arises. Hence, it is of interest to reduce the data rate necessary for the transmission of these signals. A fundamental issue in this case is to determine the minimum information required to achieve the control objectives. Clearly, if a quantized discrete-time signal takes only a finite number of fixed values, then the trajectories may go close to an equilibrium but not converge. The result is that asymptotic stability may not be achieved and various technical problems arise. For example, it is of interest to clarify how close the trajectories get to the equilibrium point and, if the sampling period is large, how close do the trajectories stay at the equilibrium between sampling instants.

An approach based on Lyapunov functions (and quadratically attractive sets in particular) for quantized sampled-data systems has been developed in [211]. This study leads to the construction of specific sequential randomized algorithms. The proposed method provides a tool to determine less conservative estimates of the performance of the designed system, at the expense of obtaining a probabilistic solution instead of a guaranteed one.

For switched systems, classical stability problems require the construction of common quadratic Lyapunov functions. In [263] sequential randomized algorithms are developed for both finite and infinite families so that probabilistic and deterministic convergence results may be obtained. Extensions to multimodal systems are considered in [212]. In this case, due to the nonconvexity of the problem, probability one results are derived using a suitable combination of randomization techniques and branch-and-bound methods. Finally, other recent developments include the construction of switching rules to select the most stable system in a certain class. This rule is based on the ordering of the systems using a common Lyapunov function approach. In particular, randomized algorithms (of Las Vegas type) for ordering the systems as well as for finding a subset of systems for which a common Lyapunov function exists are provided in [214].

Reachability Analysis Probabilistic reachability over a finite horizon is investigated in [1] for stochastic hybrid systems with control inputs. In this framework, the reachability problem amounts to determining a set of initial conditions which guarantee with a given probability that the system will evolve within a desired safe region of the state space. Randomized algorithms for controllability analysis for discrete-time piecewise affine (PWA) systems are discussed in [32].

In the works [193, 194], randomized algorithms based on sequential design methods are developed for the construction of a probabilistic output admissible (POA) set for uncertain systems with output constraints, where the POA is defined as a set of initial states which probabilistically guarantee the constraint satisfaction.

MPC, Fault Detection and ILC Model predictive control (MPC) has emerged as a very effective strategy in process industry, and numerous successful applications have been reported over the years, see for instance [92]. Nevertheless, in some cases, the use of MPC is still difficult, for example when general (nonlinear) uncertainty terms enter into the plant. An approach which consists of a combination of Kalman filtering and finite-horizon MPC for robust output feedback has been developed in [225]. More precisely, the MPC problem is reformulated as a min-max (worst-case) optimization problem. The class of uncertainties considered in this setting is quite general, so that nonlinear relationships between system matrices and uncertain parameters can be handled. The resulting optimization problem may be solved at each time instant in a probabilistic framework by means of an iterative randomized ellipsoid algorithm. Randomized methods for receding horizon navigation have been presented in [378]. Recently, an approach based on scenario optimization has been proposed in [91].

Fault detection and isolation (FDI) problems have been the subject of intense research, leading to a variety of solutions. Amongst them, model-based approaches are specially appealing, since they do not require additional hardware implementation. Early FDI methods were based on the availability of an accurate model of the monitored system. However, in practice such an assumption is hardly satisfied because of modeling errors. The resulting mismatch between the actual plant and the model used in the FDI algorithm may give raise to false alarms. For this reason, more recently, robust FDI methods have been introduced. A potential disadvantage of these methods is the difficulty to isolate the exact location of the fault and to detect simultaneous faults. The problem further complicates when the faults enter nonlinearly in the system matrices, thus becoming computationally intractable. Furthermore, deterministic methods based on relaxations unavoidably lead to conservative results. An alternative approach based on probabilistic assumptions of the faults has been developed in [271]. In this case, the resulting randomized algorithm has polynomial-time complexity and the solution obtained guarantees an a-priori determined (small) risk of failure.

Regarding fault diagnosis of analog electric circuits, various issues have been carefully analyzed. In particular, the complex nature of the fault mechanism (i.e. the physical/chemical process leading to a failure) and the unknown values of the actual component parameters are taken into account. Parameter deviations, however, depend on the intrinsic nature of the production process of the component and

on-the-field deviations, such as those related to aging or thermal effects. Such situations, which do not affect the circuit topology, are commonly defined as "soft faults" and may lead to unpredictable incorrect operations depending on their influence on the circuit performance. In [14] a methodology for selecting the appropriate test input stimuli and nodes is presented for analog circuits. The method is based on a sensitivity analysis carried out by means of a probabilistic approach based on Monte Carlo simulations.

Recently, randomized algorithms have been developed in the area of iterative learning control (ILC), see [188] and references therein. In particular, in this paper, bounds on the model uncertainty of the ILC system representation in the trial domain are obtained. Weighting matrices of the so-called *norm-optimal* ILC are derived, so that robust monotonic convergence conditions are guaranteed.

Electric Circuits and Embedded Systems The design of embedded systems requires the development of sophisticated computer-aided tools to meet the high-level specifications which are nowadays needed. In particular, various physical sources of uncertainties (including finite precision representation, fluctuations of physical parameters and battery power variations) entering into the embedded system should be analyzed by means of an accurate probabilistic description. This information is subsequently exploited by suitably choosing hardware and software requirements before an architectural design implementation takes place. In [12, 13] a randomization-based methodology is proposed for estimating the performance degradation of embedded systems in the presence of uncertainty. The final objective of this research is to address various issues related to analog versus digital (or fixed versus floating point) in order to validate a specific architectural choice.

Furthermore, the performance analysis of complex electrical circuits cannot disregard the unavoidable parametric uncertainty entering each network component. This uncertainty usually arises from physical tolerances introduced in the manufacturing process and/or, in the case of thin-film circuits, imprecisions in the deposition processes. In the classical approach used in circuit analysis, a stochastic description of the parametric uncertainty is considered with the goal of estimating the average behavior of the circuit. In other words, the objective is to estimate the probability that a given system property holds. Contrary to standard approaches aiming at computing "soft bounds" using randomized algorithms, in [250] a different method which provides "hard bounds" on the probability of performance is proposed. The resulting randomized algorithm has no longer polynomial-time complexity.

In [230], a different type of simulation-based approach for circuits is considered. Specific results are obtained for the class of resistive networks. More precisely, given suitable upper and lower bounds on the value of each resistor (but no probability distribution in this interval is given a priori), the problem of determining a suitable probabilistic measure of performance is studied. In other words, since no a priori probability distribution for the uncertain resistors is assumed, a certain type of "distributional robustness" is analyzed. Some of the performance levels obtained via this approach may differ considerably from those computed using Monte Carlo techniques.

The problem of synthesizing real-time embedded controllers taking into account constraints deriving from the implementation platform, and hence exploring the relation between the processors time, is studied in [317], where a validation approach based on randomized algorithms is presented.

Structural Analysis and Design In the context of the analysis and design of mechanical structures, uncertainty arises from imprecise knowledge of material characteristics and/or loading configurations. In [358], a Monte Carlo based algorithm for the simulation of the dynamic response of tall buildings under turbulent winds is presented. In [89] a reliable structural design is obtained using techniques based on the interplay of convex optimization and randomization.

Linear Parameter Varying (LPV) Systems In various applications, linear parameter varying (LPV) systems provide a good starting point for analysis and design of more general gain scheduling problems. Since the original motivation for introducing gain scheduling is to cope with plant nonlinearities, the resulting LPV model generally depends on the scheduling parameters in a nonlinear fashion. From the computational point of view, this observation implies that, in principle, the solution of some parameter-dependent linear matrix inequalities (LMIs) is required. To address this critical issue, a specific functional dependence on the scheduling parameters needs to be enforced. Two different approaches are generally followed in the related literature. The first approach, denoted as approximation, amounts to restricting the attention to a specific class of functions of the scheduling parameters. For example, one can assume that the matrices of the LPV model are multiaffine functions or linear fractional transformations of the underlying parameters. The original problem is then reduced to more tractable formulae which involve a finite number of LMIs but, unfortunately, some conservatism is introduced in the approximation. The second approach, often denoted as gridding, is to grid the bounding set of parameters. In this case, the original problem is reformulated by means of the solution of a finite number of LMIs, but this number depends on the grid points and generally increases exponentially with the number of scheduling parameters. Moreover, the fulfillment of the LMIs at the grid points does not give any guarantee that they are also satisfied for the whole parameter set. A third alternative approach has been developed in [168]. In this case, the scheduling parameters are treated as random variables and a sequential randomized algorithm (gradient-based) is developed. Other randomized algorithms for LPV problems in the context of aerospace applications are proposed in [269].

Automotive and Driver Assistance Systems The increasing demand of safer passenger vehicles receives a growing attention in the automotive industry. This task requires the development of advanced driver assistance systems (ADAS) which are in fact adaptive cruise control systems used for precrash sensing. Hence, these systems improve the driving comfort and can also assist the driver in reacting to dangerous situations to avoid collisions between vehicles. However, the complexity of these intelligent vehicle control systems contradicts the increasing demand

for reliability. To improve fault management, therefore, redundant components and fault tolerant controllers are often implemented in ADAS. In various practical situations, unfortunately, it may be difficult to validate their effectiveness. Therefore, simulations and prototype test drives are used to check the effectiveness of ADAS, but these tests are costly. It is therefore very important to develop efficient methodologies to conduct hardware-in-the-loop experiments in a laboratory environment with full-scale intelligent vehicles. To this end, a methodology to cope with various failure modes, as well as complex operating conditions, is developed in [179]. This approach relies on randomized algorithms that form the basis for off-line Monte Carlo simulations with the ADAS control system. The strength of this approach is that the control system analysis does not depend on the level of complexity of the underlying system (in contrast to a deterministic approach which is often adopted).

Quantum Systems and Control Randomized methods for quantum systems and control have been addressed in various papers, including [341] which identifies control scenarios where a randomized design may substantially improve the performance of dynamical decoupling methods. In [289] a probabilistic characterization of quantum dynamics and quantum process tomography in the presence of noisy measurements is discussed. The advantages of randomization at long evolution times in closed quantum systems is studied in [342]. This paper also analyzes the consequences of faulty control in deterministic versus randomized schemes and discusses mixed strategies and protocols. In [75] methods for efficiently generating random quantum states and unitary operators are analyzed for arbitrary statistical moments of quantum circuits. We refer to these papers for pointers to the related literature.

19.2 PageRank Computation and Multi-agent Systems

In the search engine of Google, the PageRank algorithm quantifies the importance of each webpage based on the link structure of the web. We first provide an overview of the original problem and then we analyze a distributed randomized scheme based on a local update of the webpages. Finally, we discuss the relations between the PageRank computation and consensus of multi-agent systems.

19.2.1 Search Engines and PageRank

In the last decade, search engines have become indispensable tools for searching the web. In these engines, the search results need to suitably rank the pages so that the web users may have a quick access to the most relevant information. The PageRank algorithm at Google is one of the successful algorithms which quantify the importance of each webpage. This algorithm was initially proposed in [72] and an overview can be found in, e.g., [252].

One of the main features of the PageRank algorithm is that it is based entirely on the link structure of the web. The idea behind this algorithm is that links from important pages make a page more important. In other words, each page is considered to be voting the pages to which it is linked to. Then, in the ranking of a page, the total number of votes as well as the importance of the voters are reflected. This problem is mathematically formulated as determining the eigenvector corresponding to the largest eigenvalue of a certain (column) stochastic matrix[1] associated with the web structure.

For the PageRank computation, a critical aspect is the size of the web, which currently consists of at least 10 billion pages. In practice, the class of algorithms that can be applied is limited: the well-known power method is employed, but it is reported that this computation takes about a week [252]. In this regard, some alternative approaches have been proposed. For example, in [223] an adaptive computation method is developed. This method classifies webpages into groups based on the speed of convergence to the PageRank values and allocates computational resources accordingly. Another line of research is based on Monte Carlo methods for distributed computation performed on multiple servers communicating to each other, see [31].

In this section, we study the randomized distributed approach introduced in [215]. This approach enjoys the following three main features:

1. Each page can compute its own PageRank value locally by communicating with the pages that are connected by direct links. That is, each page exchanges its value with the pages that it is linked to and those linked to it;
2. The pages make the decision to initiate this communication at random times which are independent from page to page. This means that, in its implementation, there is neither a fixed order among the pages nor a centralized agent in the web that determines the pages that need to update their values;
3. The computation required for each page is very mild.

The main result of [215] shows that the randomized algorithm converges to the true PageRank value in a mean square sense. This is achieved by computing the time average of the local rankings at each page. From a technical viewpoint, a crucial feature of the proposed approach is that the stochasticity of the matrix in the original problem is preserved and exploited.

19.2.2 PageRank Problem

We now provide a brief introductory description of the PageRank problem; this material can be found in, e.g. [72, 252]. We study a network of n webpages which is represented by the *directed* graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of

[1] A nonnegative matrix $A \in \mathbb{R}^{n,n}, [A]_{i,\ell} \geq 0, i = 1, \ldots, n; \ell = 1, \ldots, n$, is said to be column stochastic if $\sum_{i=1}^{n} [A]_{i,\ell} = 1$ for $\ell = 1, \ldots, n$.

vertices corresponding to the webpage indices and $\mathcal{E}$ is the set of edges representing the links among the pages. The vertex i is connected to the vertex ℓ by an edge, i.e. $(i, \ell) \in \mathcal{E}$, if page i has an outgoing link to page ℓ, or, in other words, page ℓ has an incoming link from page i. To avoid trivial situations, we assume $n \geq 2$.

The objective of the PageRank algorithm is to provide some measure of importance of each webpage. The PageRank value, or simply the value, of page $i \in \mathcal{V}$ is a real number in $[0, 1]$ denoted by x_i^*. The values are ordered such that $x_i^* > x_\ell^*$ implies that page i is "more important" than page ℓ. The basic idea in ranking the pages in terms of the values is that a page having links from important pages is also important. This is realized by determining the value of one page as a sum of the contributions from all pages that have links to it. In particular, the value x_i^* of page i is defined as

$$x_i^* = \sum_{\ell \in \mathcal{L}_i} \frac{x_\ell^*}{n_\ell}$$

where $\mathcal{L}_i = \{\ell : (\ell, i) \in \mathcal{E}\}$ is the index set of pages linked to page i and n_ℓ is the number of outgoing links from page ℓ. The values are suitably normalized as $\sum_{i=1}^n x_i^* = 1$.

Letting the values be in the vector form $x^* \in [0, 1]^n$, the PageRank is then defined as

$$x^* = Ax^*, \quad x^* \in [0, 1]^n, \quad \sum_{i=1}^n x_i^* = 1$$

where the matrix $A \in \mathbb{R}^{n,n}$, called the link matrix, is given by

$$[A]_{i,\ell} = \begin{cases} \frac{1}{n_\ell} & \text{if } \ell \in \mathcal{L}_i; \\ 0 & \text{otherwise.} \end{cases}$$

Note that the value vector x^* is a nonnegative unit eigenvector corresponding to the eigenvalue 1 of the nonnegative matrix A. In general, for existence and uniqueness of this eigenvector, it is sufficient that the web as a directed graph is strongly connected,[2] see e.g. [207]. However, the web is not strongly connected.

To address existence and uniqueness issues, the PageRank problem needs to be suitably redefined. First, we observe that in the web the so-called dangling nodes, which are pages having no outgoing links to other pages, are abundant. An example of a dangling node is a pdf file having no outgoing link. These pages introduce zero columns into the link matrix making it substochastic instead of stochastic. To resolve this issue, an artificial link may be added to represent the back button of the browser. The result is that the graph is redefined so that the link matrix A becomes a column stochastic matrix. This implies that there exists at least one eigenvalue equal to 1.

[2] A directed graph is said to be strongly connected if, for any two vertices $i, \ell \in \mathcal{V}$, there exists a sequence of edges which connects i to ℓ. In terms of the link matrix A, strong connectivity of the graph is equivalent to A being irreducible.

Next, to guarantee uniqueness, a modified version of the PageRank problem has been introduced in [72] as follows: let m be a parameter such that $m \in (0, 1)$, and let the modified link matrix $M \in \mathbb{R}^{n,n}$ be defined as a convex combination of the matrix link A and a rank-one matrix $S \in \mathbb{R}^{n,n}$ having all entries equal to 1. The entries of the matrix S represent the equal probability that a web surfer is "teleported" to a certain page during the random walk between webpages. To this end, the matrix M is often called the *teleportation matrix*. Following the choice made in [72], the parameter m is generally fixed to 0.15 and this value is used in this section, see further discussions in [252]. Formally, we write

$$M = (1 - m)A + \frac{m}{n}S.$$

Notice that M is a positive column stochastic matrix because it is a convex combination of the nonnegative matrices A and S and $m \in (0, 1)$. By the Perron theorem, see e.g. [207], it follows that this matrix is primitive.[3] In particular, the eigenvalue 1 is of multiplicity 1 and it is the unique eigenvalue of maximum modulus (i.e. with the maximum absolute value). Furthermore, the corresponding eigenvector is positive. We now formally redefine the PageRank value x^* by using the matrix M instead of A.

Problem 19.1 (PageRank computation) *Compute the PageRank value $x^* \in \mathbb{R}^n$ defined as*

$$x^* = Mx^*, \quad x^* \in [0, 1]^n, \quad \sum_{i=1}^{n} x_i^* = 1$$

where

$$M = (1 - m)A + \frac{m}{n}S$$

and $A \in \mathbb{R}^{n,n}$ is a column stochastic matrix, $S \in \mathbb{R}^{n,n}$ is a rank-one matrix having all entries equal to 1 and $m = 0.15$.

As previously discussed, due to the large dimension of the link matrix, the computation of PageRank is very difficult. The solution that is used in practice is based on the power method, see e.g. [207]. That is, the value vector x^* is computed through the recursion

$$x(k + 1) = Mx(k) = (1 - m)Ax(k) + \frac{m}{n}b \tag{19.1}$$

where $b = [1 \ \cdots \ 1]^T$, $x(k) \in \mathbb{R}^n$ and the initial condition $x(0) \in \mathbb{R}^n$ is a probability vector.[4] The second equality above immediately follows from the fact $Sx(k) = b = [1 \ \cdots \ 1]^T$ for all $k > 0$. This equation is useful because it is certainly preferable to

[3] A nonnegative matrix $A \in \mathbb{R}^{n,n}$ is said to be primitive if it is irreducible and has only one eigenvalue of maximum modulus.

[4] A probability vector is a nonnegative vector $x \in \mathbb{R}^n$, $x_i \geq 0$, $i = 1, \ldots, n$, such that $\sum_{i=1}^{n} x_i = 1$.

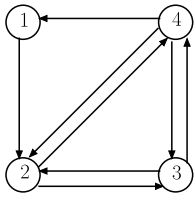

Fig. 19.1 A web with four pages

compute x^* using the sparse matrix A instead of the dense matrix M. Convergence of the power method is stated in the next result, see, e.g. [207].

Lemma 19.1 (Convergence of the power method) *For any initial condition* $x(0) \in \mathbb{R}^n$ *which is a probability vector, we have*

$$x(k) \to x^* \quad \text{for } k \to \infty$$

where $x(k)$ *is given by* (19.1).

We now make a few comments about the convergence rate of this scheme. Denote by $\lambda_1(M)$ and $\lambda_2(M)$ the largest and the second largest eigenvalues of M in magnitude, respectively. The asymptotic rate of convergence of the power method is exponential and depends on the ratio $|\lambda_2(M)/\lambda_1(M)|$. Since M is a positive stochastic matrix, we have $\lambda_1(M) = 1$ and it can be easily shown that $|\lambda_2(M)| \leq 1 - m = 0.85$. Next, we provide a simple example for illustration.

Example 19.1 (PageRank computation) Consider the web with four pages shown in Fig. 19.1. As a graph, this web is strongly connected, and there are no dangling nodes. The link matrix A and the modified link matrix M can be easily determined obtaining

$$A = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{3} \\ 1 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}, \qquad M = \begin{bmatrix} 0.0375 & 0.0375 & 0.0375 & 0.3208 \\ 0.8875 & 0.0375 & 0.4625 & 0.3208 \\ 0.0375 & 0.4625 & 0.0375 & 0.3208 \\ 0.0375 & 0.4625 & 0.4625 & 0.0375 \end{bmatrix}.$$

Then, the value vector x^* is computed

$$x^* = [0.119 \quad 0.331 \quad 0.260 \quad 0.289]^T.$$

We notice that page 2 has the largest value since it is linked from three pages while page 1, which has only one link to it, has the smallest value. On the other hand, pages 3 and 4 have the same number of incoming links, but page 4 has a larger value. This is because page 4 has more outgoing links, and thus it receives more contribution from page 3 than what it gives back.

19.2.3 Distributed Randomized Approach

We now study a distributed randomized approach to compute the PageRank value x^*. The basic protocol of the scheme is as follows: at time k, page i initiates its PageRank value update

1. sending the value of page i to the page having outgoing links from page i;
2. requesting the values from the pages having incoming links to page i.

We notice that the first step of this protocol is feasible in a real web, while the second one is not realistic because it may be very difficult to collect the values from the pages having incoming links to page i. However, this step present in [215] has been subsequently removed in [216].

To implement the scheme in a distributed manner, we assume that the pages taking the update action are determined in a random fashion. This is specified by the random process $\boldsymbol{\xi}(k) \in \mathcal{V}$, $k > 0$. If at time k, $\xi(k) = i$, then page i initiates an update of the value by communicating and exchanging the values with the pages connected by incoming and outgoing links according to the protocol outlined above. Specifically, $\boldsymbol{\xi}(k)$ is assumed to be an iid random process with uniform probability distribution given by

$$\Pr\{\boldsymbol{\xi}(k) = i\} = \frac{1}{n} \tag{19.2}$$

for all $k > 0$. This means that each page takes the update action with probability equal to $1/n$. This scheme may be implemented asynchronously without requiring a centralized clock, a common decision maker or any fixed order among the pages.

In particular, consider the distributed update scheme of the form

$$x(k+1) = (1 - \hat{m})A_{\boldsymbol{\xi}(k)}x(k) + \frac{\hat{m}}{n}b \tag{19.3}$$

where $\boldsymbol{\xi}(k) \in \{1, \ldots, n\}$, $x(k) \in \mathbb{R}^n$ is the random state whose initial condition $x(0)$ is a probability vector, $\hat{m} \in (0, 1)$ is a design parameter, $b = [1 \ \cdots \ 1]^T$ and A_i, $i = 1, \ldots, n$, are called the *distributed link matrices* which are specially constructed in the next section. The objective is to design this distributed update scheme by finding the appropriate link matrices A_i and the parameter $\hat{m}$ so that the PageRank values are computed through the time average of the state $x(k)$. Let $y(k)$ be the average of the sample path $x(0), \ldots, x(k)$

$$y(k) = \frac{1}{k+1}\sum_{\ell=0}^{k} x(\ell).$$

We say that, for the distributed update scheme, the PageRank value x^* is obtained through the time average $y(k)$ if, for each initial state $x(0)$ that is a probability vector, $y(k)$ converges to x^* in the mean square sense as follows

$$\mathrm{E}_{\boldsymbol{\xi}(0),\ldots,\boldsymbol{\xi}(k)}\left(\left\|y(k) - x^*\right\|^2\right) \to 0 \quad \text{for } k \to \infty.$$

We remark that expectation $\mathrm{E}(\cdot)$ is taken with respect to the random process $\boldsymbol{\xi}(0), \ldots, \boldsymbol{\xi}(k)$. This type of convergence is called ergodicity for stochastic processes, see [319].

19.2.4 Distributed Link Matrices and Their Average

In agreement with the protocol outlined in Sect. 19.2.3, we further develop the distributed update scheme of (19.3) by means of two steps: first we define the distributed link matrices A_i, then we select the parameter $\hat{m}$. For each i, the matrix $A_i \in \mathbb{R}^{n,n}$ is obtained as follows:

1. The ith row and column of A_i coincide with those of A;
2. The remaining diagonal entries are equal to $1 - [A]_{i,\ell}$, $\ell = 1, \ldots, n$, $\ell \neq i$;
3. All the remaining entries are equal to zero.

More formally, we have

$$[A_i]_{j,\ell} = \begin{cases} [A]_{j,\ell} & \text{if } j = i \text{ or } \ell = i; \\ 1 - [A]_{i,\ell} & \text{if } j = \ell \neq i; \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, n$. We observe that these matrices are column stochastic by construction because the original link matrix A possesses this property, which is critical for convergence of the scheme.

Example 19.2 (Distributed link matrices) We continue with the Example 19.1. The link matrices A_i are given by

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{3} \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{2}{3} \end{bmatrix}; \qquad A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{2}{3} \end{bmatrix};$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{2}{3} \end{bmatrix}; \qquad A_4 = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}.$$

To clarify the properties of the link matrices A_i, we consider the simple update scheme

$$x(k+1) = A_{\boldsymbol{\xi}(k)} x(k)$$

where $x(k)$ is the state, $x(0)$ is a probability vector and the random process $\boldsymbol{\xi}(k)$ is specified in (19.2). In particular, in [215] various properties regarding the average dynamics are shown. Since $\boldsymbol{\xi}(k)$ is an iid random process, in this paper it is shown that the expected value $\mathrm{E}(x(k))$ of the state $x(k)$ evolves according to the recursion

$$\mathrm{E}\big(x(k+1)\big) = \left[\frac{1}{n}\sum_{i=1}^{n} A_i\right] \mathrm{E}\big(x(k)\big).$$

19.2.5 Convergence of Distributed Update Scheme

To guarantee uniqueness of the eigenvector, we introduced the teleportation matrix M. Therefore, we now study a modified version of the distributed link matrices A_i. Since the link matrices A_i are column stochastic, we write the distributed update scheme in (19.3) as

$$x(k+1) = M_{\xi(k)} x(k), \tag{19.4}$$

where the *modified distributed link matrices* $M_{\xi(k)}$ are given by

$$M_i = (1-\hat{m})A_i + \frac{\hat{m}}{n} S$$

for $i = 1, \ldots, n$, and $\hat{m} \in (0,1)$ is a design parameter. Notice that M_i are positive stochastic matrices. Similarly to the argument for the link matrices A_i defined in the previous section, we observe that the expected value of $x(k+1)$ evolves according to the recursion

$$\mathrm{E}\big(x(k+1)\big) = \left[\frac{1}{n}\sum_{i=1}^{n} M_i\right] \mathrm{E}\big(x(k)\big).$$

The next result shows that the time average indeed converges to the PageRank value in a mean square sense, provided that the parameter $\hat{m}$ is appropriately chosen.

Theorem 19.1 *Let $\hat{m}$ be given by*

$$\hat{m} = \frac{2m}{n - m(n-2)} \in (0,1).$$

Then, for any initial state that is a probability vector, the time average $y(k)$ of the distributed update scheme (19.4) converges to the PageRank value x^* in the mean square sense as follows

$$\mathrm{E}_{\xi(0),\ldots,\xi(k)}\big(\|y(k) - x^*\|^2\big) \to 0 \quad \text{for } k \to \infty.$$

This theorem highlights an ergodic property in the proposed update scheme. Its proofs can be shown by general Markov process results as in, e.g. [109]. A more specific proof is given in [215]. Regarding the convergence of this algorithm, it can be shown that it is of order $1/k$ and moreover depends on the size of n linearly through the parameter $\hat{m}$. This convergence result has been extended in various directions, which include simultaneous update of multiple webpages, link failure and aggregation of web pages, see [216].

19.2.6 Relations to Consensus Problems

The randomized distributed techniques previously discussed have been also motivated by the research on distributed consensus, agreement and flocking problems,

see e.g. [3, 26]. We now outline the relations between consensus and PageRank. First, we describe a stochastic version of consensus, which has been studied in, e.g. [69, 195, 376, 414]; see also [384] for an introductory overview from the viewpoint of randomized algorithms. We consider a network of agents represented by the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges. If agent i has a communication link to agent ℓ, then the vertex i is connected to the vertex ℓ by an edge $(i, \ell) \in \mathcal{E}$. We assume that the graph is strongly connected. The objective is that all agents reach a common value by communicating to each other, where the pattern in the communication protocol is randomly determined at each time.

In particular, let $x_i(k)$ be the value of agent i at time k, and let $x(k) = [x_1(k) \ \cdots \ x_n(k)]^T \in \mathbb{R}^n$. The values are updated via the recursion

$$x(k+1) = A_{\boldsymbol{\xi}(k)} x(k) \tag{19.5}$$

where $\boldsymbol{\xi}(k) \in \{1, \ldots, d\}$ is the mode specifying the random communication pattern among the agents and d is the number of such patterns. The communication pattern we consider is an iid random process with uniform probability distribution given by

$$\Pr\{\boldsymbol{\xi}(k) = i\} = \frac{1}{d}$$

for all $k > 0$. The communication pattern is constructed such that each $i \in \{1, \ldots, d\}$ corresponds to the subset $\mathcal{E}_i \subset \mathcal{E}$ of the edge set. Then, the entries $[A_i]_{j,\ell}$ are nonnegative if and only if $(\ell, j) \in \mathcal{E}_i$. We now present an example illustrating the construction of the matrices A_i.

Example 19.3 (Communication patterns for consensus) Consider the graph in Example 19.1 with four agents. We introduce four communication patterns arising from the protocol in the distributed PageRank algorithm: the edge subset $\mathcal{E}_i$ contains all (i, j) and (j, i) in the original edge set $\mathcal{E}$ including (i, i) that corresponds to a self-loop for i, $j = 1, 2, 3, 4$. The matrices A_i can be constructed as

$$A_1 = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \qquad A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix};$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}; \qquad A_4 = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}.$$

The main difference with the link matrices A_i in Example 19.2 is that these matrices are *row stochastic* and moreover the diagonal entries are all positive, which indicates the presence of self-loops which are absent in the web.

Table 19.1 Consensus and distributed PageRank

Consensus	PageRank
all agent values become equal	page values converge to a constant
strongly connected graph of agents	web is not strongly connected
row stochastic matrices A_i	column stochastic matrices A_i, M_i
presence of self-loops in the graph	no self-loops are present in the web
$\Pr\{\lim_{k\to\infty} \lvert x_i(k) - x_\ell(k)\rvert = 0\} = 1$	$\mathrm{E}(\lVert y(k) - x^*\rVert^2) \to 0$ for $k \to \infty$
any initial condition $x(0) \in \mathbb{R}^n$	$x(0) \in \mathbb{R}^n$ stochastic vector
time averaging not necessary	time averaging $y(k)$ required

We say that consensus is achieved if the relation

$$\Pr\left\{\lim_{k\to\infty} \left|x_i(k) - x_\ell(k)\right| = 0\right\} = 1 \tag{19.6}$$

holds for all $i, \ell \in \mathcal{V}$ and for any initial condition $x(0) \in \mathbb{R}^n$. A well-known approach to achieve consensus is to update the value of each agent by taking the average of the values received at time k. In this case, the matrix A_i is constructed as

$$[A_i]_{j,\ell} = \begin{cases} \frac{1}{n_{ij}} & \text{if } (\ell, j) \in \mathcal{E}_i; \\ 0 & \text{otherwise} \end{cases}$$

where n_{ij} is the number of agents ℓ with $(\ell, j) \in \mathcal{E}_i$, i.e. those transmitting their values to agent j.

We now present a convergence result for consensus, see e.g. [376] for proof.

Lemma 19.2 *Assume that the graph is strongly connected and the properties*

1. $(\ell, \ell) \in \mathcal{E}_i$ *for all* ℓ;
2. $\bigcup_{i=1}^{d} \mathcal{E}_i = \mathcal{E}$;
3. *The matrix* A_i *is a row stochastic matrix*

are satisfied. Then, the scheme (19.5) *achieves consensus in the sense of* (19.6).

To conclude this section, in Table 19.1 we summarize some of the key differences and similarities between consensus and the distributed PageRank problems.

19.3 Control Design of Mini-UAVs

Unmanned aerial vehicles (UAVs) are flying objects, often of reduced dimensions, that behave like aerial robots whose mobility can deploy a useful micropayload to a remote or hazardous location, where they may perform a variety of missions, see [125]. Many successful UAV designs have been built for either research, commercial or military purposes by several universities, companies and government-funded agencies. Recently, UAVs have been the subject of considerable interest and development within the systems and control community, see e.g. [398].

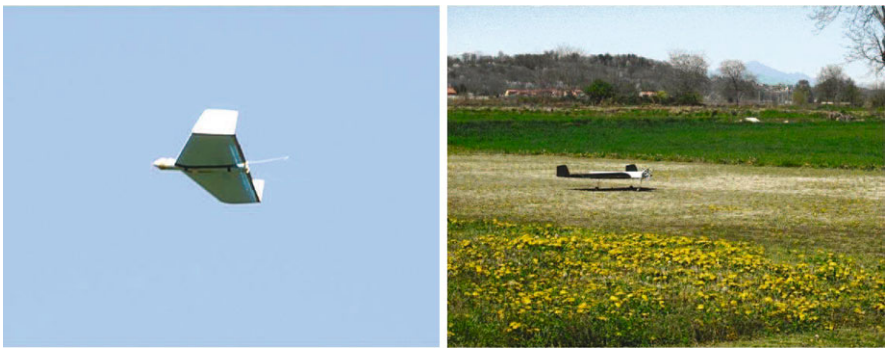

Fig. 19.2 MH1000 platform (*left*) and MH2000 platform (*right*). These platforms have been designed for archaeological sites reconnaissance, natural disaster and fire detection monitoring, in the frame of several projects, including ITHACA. MH1000 and MH2000 are based on the MicroHawk configuration, developed at the Mechanical and Aerospace Engineering Departments, Politecnico di Torino, Italy

Fig. 19.3 The archaeological site of Bene Vagienna located in Italy. The picture has been taken during a flight test of MH2000 using a visual camera. On the *left*, the *dark shadow* of the UAV may be recognized when flying over the trees

The aerial platform discussed in this section is named MH1000, see [185], see Figs. 19.2 and 19.3. This is a small autonomous aerial vehicle characterized by 3.28 ft wingspan and a total take-off weight of approximately 3.3 lb. This platform is based on a conventional layout, characterized by a fixed wing, tailless integrated wing-body, tractor propeller driven. MH1000 flies at speed ranging from 33 ft/s to 66 ft/s and maximum operating altitude of about 32 ft. Experimental on-site tests demonstrated that a 40 minutes flight can be achieved at an average speed of about 40 ft/s. The platform includes an embedded real-time control system and is equipped with various onboard sensors and two cameras (color and infrared) to detect environmental conditions and to gather images, which are transmitted on-line to a ground

station, of the target area. Depending on the user's need, MH1000 has been employed for autonomous flight as well as for a remotely piloted flight. Trajectory optimization, dynamic response analysis, controller architecture design and control synthesis have been carried out by modeling the aircraft dynamics and the surrounding environment conditions. The performance and the compliance to project requirements have been tested by means of simulations regarding mission profiles, see [266].

19.3.1 Modeling the MH1000 Platform

The nonlinear model of the aerial platform is characterized by a set of 12 coupled nonlinear differential equations. The equations of motion are given with reference to the wind-axes frame, i.e. a system having origin at the vehicle center of gravity and axes aligned to the flight trajectory. Four blocks of three state variables each have been identified. The first three state variables are flight speed, angle of attack and sideslip angle, respectively, the next three variables are the stability-axes components of the angular velocity vector, the third block consists of Euler angles describing aircraft attitudes. The last block contains the aircraft position coordinates with respect to the local navigation reference system, see details about the nonlinear model in [368].

The sensitivity to changes in flight conditions, the assumptions related to the aerodynamic model, the inaccuracies in geometric and inertial data represent uncertainties in plant and environment modeling. The design of a flight control system which guarantees a suitable level of tolerance to environmental changes and platform manufacturing/modeling inaccuracies plays a key role whenever stability and performance requirements should be satisfied. Furthermore, the mathematical model under attention is obtained by numerical linearization of the full-order nonlinear system representing the aircraft dynamics, so that explicit relationships between the state space matrices and the uncertain parameters are not available. In fact, the linearization step should be repeated whenever the values of the uncertain parameters change, and this leads to a new linearized model. For these reasons, a simulation-based approach which utilizes randomized algorithms is particularly attractive, while standard robustness tools or gain scheduling techniques do not seem very effective in this case. The simulation-based approach makes use of uncertainty randomization for both controller synthesis as well as probabilistic analysis.

The numerical simulations that have been performed are based on the assumption of decoupled dynamics and deal with the longitudinal plane dynamics stabilization. In particular, attention is focused on a full state feedback longitudinal control of the form $u = -Kx = \eta_e$, which is controlled by the symmetrical elevon deflection η_e. The state vector is defined as $x = [V \ \alpha \ \beta \ \theta]^T$, where V is the flight speed, α is the angle of attack, β is the pitch rate and θ is the pitch angle. The state and input

Table 19.2 Plant and flight condition uncertainties

Index	Parameter	pdf	$\bar{q}_i$	%	q_i^-	q_i^+
1	flight speed [ft/s]	$\mathcal{U}$	42.65	±15	36.25	49.05
2	altitude [ft]	$\mathcal{U}$	164.04	±100	0.00	328.08
3	mass [lb]	$\mathcal{U}$	3.31	±10	2.98	3.64
4	wingspan [ft]	$\mathcal{U}$	3.28	±5	3.12	3.44
5	mean aero chord [ft]	$\mathcal{U}$	1.75	±5	1.67	1.85
6	wing surface [ft^2]	$\mathcal{U}$	5.61	±10	5.06	6.18
7	moment of inertia [lb ft^2]	$\mathcal{U}$	1.34	±10	1.21	1.48

matrices that have been obtained by numerical linearization at the nominal reference condition $V = 43$ ft/s and $h = 164$ ft are given by

$$A = \begin{bmatrix} -0.293 & -0.486 & -0.0002 & -9.812 \\ -0.113 & -6.181 & 0.914 & -0.0003 \\ 0.000 & -64.83 & -8.074 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} -0.7914 \\ -3.9250 \\ -443.487 \\ 0.000 \end{bmatrix}.$$

19.3.2 Uncertainty Description

Structured parametric uncertainties include flight conditions (dynamic pressure), aerodynamic data (stability and control derivatives), geometric and inertial data, which may take into account manufacturing inaccuracies. Uncertainties related to the flight conditions can be ascribed to the real flight in a non-ideally-calm air and to the need to cover a portion of the flight envelope as large as possible. Uncertainties concerning the aerodynamic data may be related to experimental measurement errors or computational approximations due to round-off errors.

Uniform and Gaussian probability density functions have been used to describe random parametric uncertainties. In particular, 7 geometric, inertial and operational uncertainties are characterized by a uniform probability density function $\mathcal{U}$ with given lower and upper bounds q_i^-, q_i^+, $i = 1, 2, \ldots, 7$. The uncertainties related to the aerodynamic database are characterized by a Gaussian probability density function $\mathcal{N}$ with mean value $\bar{q}_i$, $i = 8, 9, \ldots, 16$ and standard deviation σ_i, $i = 8, 9, \ldots, 16$. Gaussian pdf has been chosen in this case due to the different nature of the parameters: the value of the aerodynamic derivative, for example, should have higher probability close to the nominal value experimentally or numerically obtained. Although the uncertainties are partially coupled through the aerodynamic database, they are treated as independent, so that the random samples may be generated by iid Monte Carlo simulations.

Table 19.2 shows the nominal values $\bar{q}_i$ of 7 uncertain parameters, the prescribed relative values of uncertainty and the resulting lower and upper bounds q_i^-, q_i^+

Table 19.3 Aerodynamic database uncertainties

Index	Parameter	pdf	$\bar{q}_i$	σ_i
8	C_X coefficient [–]	$\mathcal{N}$	−0.01215	0.00040
9	C_Z coefficient [–]	$\mathcal{N}$	−0.30651	0.00500
10	C_m coefficient [–]	$\mathcal{N}$	−0.02401	0.00040
11	C_{X_q} coefficient [rad^{-1}]	$\mathcal{N}$	−0.20435	0.00650
12	C_{Z_q} coefficient [rad^{-1}]	$\mathcal{N}$	−1.49462	0.05000
13	C_{m_q} coefficient [rad^{-1}]	$\mathcal{N}$	−0.76882	0.01000
14	C_{X_q} coefficient [rad^{-1}]	$\mathcal{N}$	−0.17072	0.00540
15	C_{Z_q} coefficient [rad^{-1}]	$\mathcal{N}$	−1.41136	0.02200
16	C_{m_q} coefficient [rad^{-1}]	$\mathcal{N}$	−0.94853	0.01500

(for simplicity only approximate values of these bounds are given in the table). Table 19.3 shows the numerical values of 9 aerodynamic coefficients $\bar{q}_i$ which enter in the nonlinear model and their standard deviation σ_i, see [266] for a precise definition of these coefficients. They are considered as random parameters with a Gaussian density with the mean value and standard deviation shown in the table.

19.3.3 Randomized Control Algorithms

Three randomized algorithms, which should be used sequentially, have been developed in [266]; the block diagrams are shown in Figs. 19.4 and 19.5. The main tasks performed by these algorithms are now summarized.

1. *Algorithm RGS* (*Random Gain Synthesis*) is based upon the selection of two "critical uncertain parameters" which are the flight speed and the mass. The selection of a reduced number of critical uncertain parameters reduces the computational workload in the design phase. All the remaining 14 parameters are set to their nominal values. Taking a uniform pdf for the controller gains, and imposing suitable bounds, Algorithm RGS provides a set of candidate controller gains which satisfy a given specification property. A termination criterion regarding the number of randomly generated samples, which guarantees a given probabilistic accuracy and confidence, is imposed.
2. *Algorithm RSRA* (*Random Stability Robustness Analysis*) uses the set of candidate gains previously obtained and a given specification property regarding a root confinement region for the closed-loop poles of the linearized system. This algorithm is based on randomization of all uncertain parameters (and not only the critical ones), according to the specified probability distributions. Empirical probabilities that the closed-loop specification property is satisfied are computed using the Chernoff bound.
3. *Algorithm RPRA* (*Random Performance Robustness Analysis*) has a structure similar to Algorithm RSRA, but uses a different specification property, based on

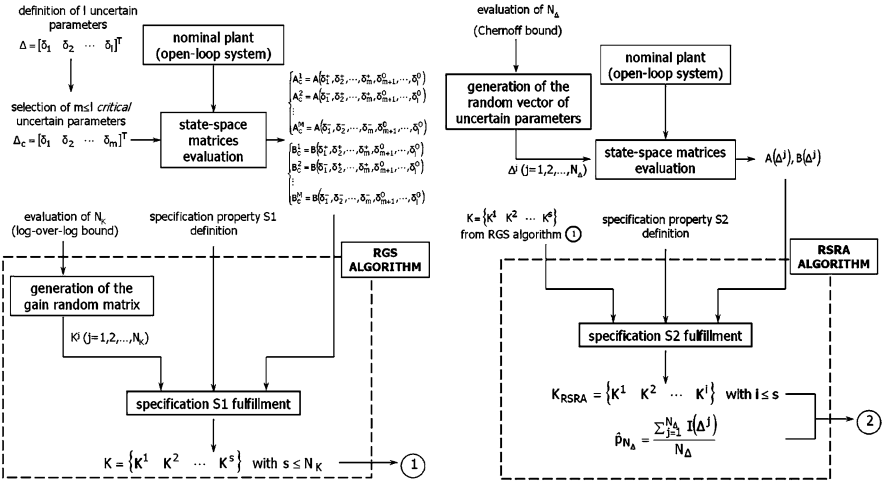

Fig. 19.4 Block diagrams of the algorithms RGS (*left*) and RSRA (*right*)

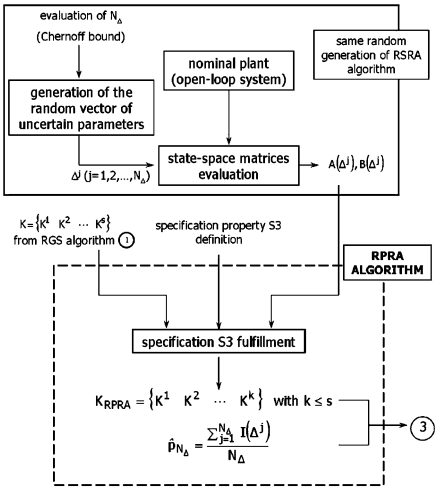

Fig. 19.5 Block diagram of the algorithm RPRA

performance metrics such as flying and handling qualities, see [24]. Empirical probabilities of achieving these performance metrics are also computed.

The set of controller gains which have been obtained in [266] is given by

$$K = [0.000109 \quad 0.091832 \quad 0.015300 \quad -0.004044]^T.$$

These gains satisfy a good compromise between stability and performance requirements, in terms of probability degradation features. For validation purposes, the time domain responses of the complete nonlinear system have been analyzed in the closed-loop configuration, by implementing these controller gains, see the numerical values reported in [266].

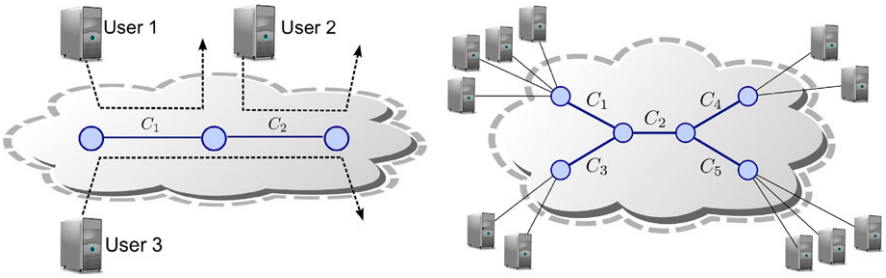

Fig. 19.6 Examples of two network topologies

19.4 Performance of High-Speed Networks

Wired and wireless communication networks have received increasing attention in the control literature, as evidenced by the appearance of several special issues, see [19, 25, 26, 76, 183]. Various approaches and solutions have been developed in this context, including modeling of TCP/IP traffic, congestion control for available bit rate (ABR) service in asynchronous transmission mode (ATM) networks, packet marking schemes and protocols for the Internet, application of low-order controllers for active queue management (AQM) and related problems. One of the critical issues at the heart of efficient operations of high-speed networks is *congestion control*. This involves the problem of regulating the source rates in a decentralized and distributed fashion, so that the available bandwidths on different links are used most efficiently while minimizing or totally eliminating loss of packets due to queues at buffers exceeding their capacities. This issue needs to be accomplished under variations in network conditions such as packet delays due to propagation as well as to queuing and bottleneck nodes.

In this section, which is based on [17], randomized algorithms are used to perform a stability analysis of a model introduced in [16], which makes use of noncooperative game theory [34].

19.4.1 Network Model

Fluid models, which replace discrete packets with continuous flows, are widely used in addressing a variety of network control problems, such as congestion control, routing and pricing. The topology of the network studied here is characterized by a set of nodes $\mathcal{N} = \{1, \ldots, N\}$ and a set of links $\mathcal{L} = \{1, \ldots, L\}$, with each link $\ell \in \mathcal{L}$ having a fixed capacity $C_\ell > 0$ and an associated buffer size $b_\ell \geq 0$. The set of users is denoted by $\mathcal{M} = \{1, \ldots, M\}$. Each user is associated with a unique connection between a source and a destination node. The connection is a path that connects various nodes, see Fig. 19.6 for an illustration.

For the ith user, we consider a path $\overline{\mathcal{L}}_i$ which is a subset of $\mathcal{L}$. The nonnegative flow x_i sent by the ith user over the path $\overline{\mathcal{L}}_i$ satisfies the bounds

$$0 \le x_i \le x_{i,\max}$$

where the upper bound $x_{i,\max}$ on the ith user's flow rate may be a user-specific physical limitation. This upper bound cannot exceed the minimum capacity of the links on the route

$$x_{i,\max} \le \min_{\ell \in \overline{\mathcal{L}}_i} C_\ell.$$

As in [229], the model studied here makes use of a binary routing matrix R that describes the relation between the set of routes associated with the users and links. That is, for each user $i \in \mathcal{M}$ and link $\ell \in \mathcal{L}$ the entries of the routing matrix R are given by

$$[R]_{\ell,i} = \begin{cases} 1 & \text{if source } i \text{ uses link } \ell; \\ 0 & \text{otherwise.} \end{cases}$$

Using this matrix representation, we have the inequality

$$Rx \le C$$

where $x = [x_1 \ \cdots \ x_M]^T$ is the users flow rate vector and $C = [C_1 \ \cdots \ C_L]^T$ is the link capacity vector. If the aggregate sending rate of users whose flows pass through link ℓ exceeds the capacity C_ℓ of the link, then the arriving packets are queued in the buffer b_ℓ of the link. The total flow on link ℓ is denoted as $\overline{x}_\ell(t)$ and is given by

$$\overline{x}_\ell(t) = \sum_{\{i:\ell \in \overline{\mathcal{L}}_i\}} x_i(t).$$

Ignoring boundary effects, the buffer level b_ℓ at link ℓ evolves in agreement with the differential equation

$$\dot{b}_\ell(t) = \overline{x}_\ell - C_\ell.$$

19.4.2 Cost Function

For Internet-style networks, an important indication of congestion is the variation in queuing delay, defined as the difference between the actual delay experienced by a packet and the propagation delay of the connection. If the incoming flow rate to a router exceeds the capacity of the outgoing link, then packets are queued, generally on a first-come first-served basis, in the corresponding buffer of the router, thus leading to an increase in the round-trip time (RTT) of packets. Hence, the RTT on a congested path is longer than the base RTT, which is defined as the sum of propagation and processing delays on the path of a packet. The queuing delay τ_ℓ at a link can be modeled as

$$\dot{\tau}_\ell(x,t) = \frac{1}{C_\ell}\dot{b}_\ell(t) = \frac{1}{C_\ell}\big(\overline{x}_\ell(t) - C_\ell\big).$$

Thus, the queuing delay that a user experiences is the sum of queuing delays on its path

$$\overline{\tau}_i(x,t) = \sum_{\ell \in \overline{\mathcal{L}}_i} \tau_\ell(x,t).$$

The goal is to make use of variations in RTT to devise a congestion control and pricing scheme. Then, the cost function for the ith user at time t is the difference between a linear pricing function proportional (through a parameter q_i) to the queuing delay, and a strictly increasing logarithmic utility function multiplied by a user preference parameter u_i

$$J_i(x,t) = \sum_{\ell \in \overline{\mathcal{L}}_i} \left(q_i \tau_\ell(x,t) x_i\right) - u_i \log(x_i + 1).$$

Since the users pick their flow rates in a way that would minimize their cost functions, we adopt a dynamic update model whereby each user changes the flow rate proportional to the gradient of the cost function with respect to the flow rate. Thus, the algorithm for the ith user is defined as

$$\dot{x}_i(t) = -\dot{J}_i(x,t) = \frac{u_i}{x_i + 1} - q_i \overline{\tau}_i(t)$$

where we have ignored the effect of the ith user's flow on the delay $\overline{\tau}_i$.

Next, we observe that the users update their flow rates only at discrete time instances corresponding to multiples of RTT. Hence, we discretize this equation, obtaining

$$x_i(k+1) = x_i(k) + \kappa_i \left[\frac{u_i}{x_i(k) + 1} - q_i \sum_{\ell \in \overline{\mathcal{L}}_i} \tau_\ell(k) \right] \tag{19.7}$$

where $x_i(0) = 0$ and κ_i is a user-specific stepsize constant, which can be set to one without loss of generality. The queue model is discretized in a similar manner, obtaining

$$\tau_\ell(k+1) = \tau_\ell(k) + \frac{1}{C_\ell} \sum_{\{i : \ell \in \overline{\mathcal{L}}_i\}} x_i(k) - 1 \tag{19.8}$$

where $\tau_\ell(0) = 0$.

19.4.3 Robustness for Symmetric Single Bottleneck

In the case of a single bottleneck node we essentially have a single link of interest, for which we denote the associated delay with $\tau = \tau_\ell$ and the capacity with $C = C_\ell$. Then, the unique equilibrium state of the system described by (19.7) and (19.8) is given by

$$
\begin{aligned}
x_i^* &= \frac{u_i}{q_i \tau^*} - 1; \\
\tau^* &= \frac{1}{C + M} \sum_{i=1}^{M} \frac{u_i}{q_i}.
\end{aligned}
\tag{19.9}
$$

Letting $\widetilde{x}_i(k) \doteq x_i(k) - x_i^*$ and $\widetilde{\tau} \doteq \tau(k) - \tau^*$, the system (19.7) and (19.8) with a single bottleneck link and $\kappa_i = 1$ can be rewritten around the equilibrium state as

$$
\begin{aligned}
\widetilde{x}_i(k+1) &= \widetilde{x}_i(k) + \frac{u_i}{\widetilde{x}_i(k) + x_i^* + 1} - q_i\big(\widetilde{\tau}(k) + \tau^*\big); \\
\widetilde{\tau}(k+1) &= \widetilde{\tau}(k) + \frac{1}{C} \sum_{i=1}^{M} \widetilde{x}_i(k).
\end{aligned}
\tag{19.10}
$$

Linearizing this equation around $\widetilde{x}^* = 0$ and $\widetilde{\tau}^* = 0$, we easily obtain

$$
\begin{aligned}
\widetilde{x}_i(k+1) &= \left[1 - \frac{u_i}{(x_i^* + 1)^2}\right] \widetilde{x}_i(k) - q_i \widetilde{\tau}(k); \\
\widetilde{\tau}(k+1) &= \widetilde{\tau}(k) + \frac{1}{C} \sum_{i=1}^{M} \widetilde{x}_i(k).
\end{aligned}
\tag{19.11}
$$

Letting

$$
\begin{aligned}
q &= [q_1 \quad \cdots \quad q_M]^T; \\
v &= \left[\frac{u_1}{(x_1^* + 1)^2} \quad \cdots \quad \frac{u_M}{(x_M^* + 1)^2}\right]^T
\end{aligned}
$$

we rewrite (19.11) in matrix form

$$
\begin{bmatrix} \widetilde{x}(k+1) \\ \widetilde{\tau}(k+1) \end{bmatrix} = A(q, v, C) \begin{bmatrix} \widetilde{x}(k) \\ \widetilde{\tau}(k) \end{bmatrix}
\tag{19.12}
$$

where the matrix $A(q, v, C)$ is given by

$$
A(q, v, C) = \begin{bmatrix}
1 - v_1 & 0 & 0 & \dots & -q_1 \\
0 & 1 - v_2 & 0 & \dots & -q_2 \\
0 & 0 & 1 - v_3 & \dots & -q_3 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{1}{C} & \frac{1}{C} & \frac{1}{C} & \dots & 1
\end{bmatrix}.
$$

This system is (locally) stable if and only if $A(q, v, C)$ is a Schur matrix, i.e. all its eigenvalues $\lambda(q, v, C)$ lie in the open unit circle. Thus, the objective is to determine conditions on the parameters q, v and C such that $|\lambda(q, v, C)| < 1$. This task proves to be prohibitively complex in general, and hence we consider first the special situation when the parameter v is symmetric across all M users. That is,

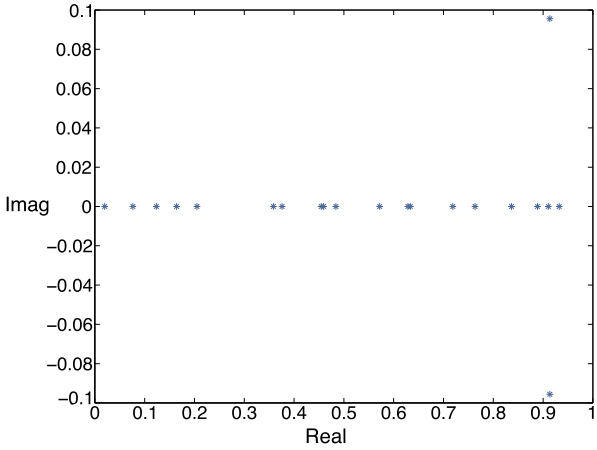

Fig. 19.7 Eigenvalues of the matrix $A(q, v, C)$ in the complex plane for the symmetric bottleneck case with 20 users

we take $v_i = v$ for all $i = 1, \ldots, M$. In this case, the characteristic equation of the matrix $A(q, v, C)$ is given by

$$\det\big(\lambda I - A(q, v, C)\big) = (\lambda - 1 + v)^{M-1}\left[\lambda^2 - (2 - v)\lambda + 1 - v + \sum_{i=1}^{M} \frac{q_i}{C}\right]$$

and the matrix $A(q, v, C)$ has $M - 1$ repeated real eigenvalues at $1 - v$ and two (possibly complex) eigenvalues at

$$1 - \frac{v}{2} \pm \sqrt{\frac{v^2}{4} - \sum_{i=1}^{M} \frac{q_i}{C}}\,.$$

First, in order to gain further insight into stability properties, the eigenvalues of the matrix $A(q, v, C)$ for $q_i = 1{,}000$, $i = 1, \ldots, 20$, $v = 1$ and $C = 20{,}000$ are computed and shown in Fig. 19.7. We note that this matrix is ill-conditioned with a condition number in the order of 10^5.

Finally, we conclude that the single bottleneck link system given by (19.10) is (locally) stable around its equilibrium state (19.9) if and only if the parameters $q_1, \ldots, q_M$, v and C lie in the region defined by the inequalities

$$\frac{1}{C}\sum_{i=1}^{M} q_i < v < 2.$$

The general nonsymmetric case is studied next using randomized algorithms.

19.4.4 Randomized Algorithms for Nonsymmetric Case

We saw in the previous section that (local) stability and robustness can be studied analytically when the user utility preference parameters are the same for all users.

If this is not the case, however, the eigenvalues of $A(q, v, C)$ cannot be expressed in closed form, and robustness of the system (19.12) is studied using randomized algorithms.

In particular, we investigate stability and robustness under parameter variations of the linearized single bottleneck link system (19.12) through randomization. That is, we investigate the effect of pricing and user parameters on the (local) stability of the system when $\mathbf{q}$ and $\mathbf{v}$ are random vectors with given pdfs $f_{\mathbf{q}}(q)$ and $f_{\mathbf{v}}(vu)$ and support $\mathcal{B}_q$ and $\mathcal{B}_v$, respectively. More precisely, for various values of C, the objective is to compute the probability

$$\text{Pr}\{\text{network stability}\} = \int_{\mathcal{B}_G} f_{\mathbf{q}}(q)\, f_{\mathbf{v}}(v)\, \mathrm{d}q\, \mathrm{d}v$$

where the good set is given by

$$\mathcal{B}_G = \big\{q \in \mathcal{B}_q,\ v \in \mathcal{B}_v : A(q, v, C) \text{ Schur}\big\}.$$

In particular, we performed simulations for the case of $M = 4$ users with parameter ranges

$$\begin{aligned} \mathcal{B}_q &= \big\{q \in \mathbb{R}^4 : q_i \in \big[0, 1\times 10^3\big],\ i = 1, \ldots, 4\big\}; \\ \mathcal{B}_v &= \big\{v \in \mathbb{R}^4 : v_i \in [0, 1],\ i = 1, \ldots, 4\big\} \end{aligned} \tag{19.13}$$

and 22 fixed values of capacity in the interval $C \in [0, 3\times 10^4]$. The values of C are chosen by performing a set of experiments regarding the stability of the network for different values of link capacity, obtaining

$$C = \{0.1, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18, 20, 22, 25, 30\}\times 10^3. \tag{19.14}$$

19.4.5 Monte Carlo Simulation

We now present the results of the simulations, which are obtained by means of Algorithm 10.1. This randomized algorithm is based upon the Monte Carlo method discussed in Chap. 7. In particular, we consider random vectors with uniform probability density functions $f_{\mathbf{q}}(q)$ and $f_{\mathbf{v}}(v)$ and support sets given in (19.13). Since these sets are rectangles, linear congruential generators described in Sect. 14.1.1, or more sophisticated methods studied in the same chapter, can be immediately used for generation of pseudo-random samples of $\mathbf{q}$ and $\mathbf{v}$. First, we choose a level of confidence $\delta = 0.001$ and accuracy $\epsilon = 0.003$, and we determine the sample size N necessary to guarantee the required probabilistic levels. To this end, we use the Chernoff bound given in (8.14), obtaining $N \geq 4.23\times 10^5$. Then, we choose $N = 450{,}000$ and construct the multisample

$$\mathbf{q}^{(1\ldots N)} = \big\{\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(N)}\big\}.$$

Similarly, we generate

$$\mathbf{v}^{(1\ldots N)} = \{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(N)}\}.$$

Subsequently, for fixed values of C given in (19.14), we compute

$$A(\mathbf{q}^{(i)}, \mathbf{v}^{(i)}, C)$$

for $i = 1, \ldots, N$. Then, the empirical probability that the system (19.12) is stable is given by

$$\widehat{\mathbf{p}}_N = \frac{\mathbf{N}_G}{N}$$

where $\mathbf{N}_G$ is the number of "good" samples for which the system is stable. More precisely, we construct the indicator function of the good set

$$\mathbb{I}_{\mathcal{B}_G}(q, v, C) = \begin{cases} 1 & \text{if } q, v \in \mathcal{B}_G; \\ 0 & \text{otherwise.} \end{cases} \tag{19.15}$$

Then, the empirical probability is given by

$$\widehat{\mathbf{p}}_N = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{\mathcal{B}_G}(\mathbf{q}^{(i)}, \mathbf{v}^{(i)}, C).$$

Hence, we conclude that the inequality

$$\left| \text{Pr}\{\text{network stability}\} - \widehat{\mathbf{p}}_N \right| \leq 0.003$$

holds with probability at least 0.999.

19.4.6 Quasi-Monte Carlo Simulation

To validate the results, different sampling schemes for the parameters q and v are studied. That is, for comparison, we also compute an estimate of the volume of $\mathcal{B}_G$ using the quasi-Monte Carlo method (see Chap. 7 for a presentation of this method), which is a deterministic mechanism for generating samples which are "evenly distributed" within the sets of interest. In this case, no probability density function is specified for the parameter vectors q and v and the samples are constructed within the unit box. The samples obtained are then subsequently rescaled in the set $\mathcal{B}_q$ defined in (19.13). In particular, the Halton sequence, see Definition 7.5, is used for generation of the point sets

$$q^{(1\ldots N)} = \{q^{(1)}, \ldots, q^{(N)}\};$$
$$v^{(1\ldots N)} = \{v^{(1)}, \ldots, v^{(N)}\}.$$

These point sets have the property to minimize an upper bound on the star discrepancies $D_N^*(q^{(1\ldots N)})$ and $D_N^*(v^{(1\ldots N)})$, see Theorem 7.8. In particular, considering a binary base (i.e. $b_i = 2$ for $i = 1, 2, 3, 4$), a sample size $N = 300{,}000$ guarantees that $D_N^*(q^{(1\ldots N)}) < 0.0421$, see the bound (7.21). Obviously, the same bound is

obtained for the point set $v^{(1\ldots N)}$. Notice that, in principle, these bounds on the discrepancy cannot be used to estimate the integration error (i.e. the error made when estimating $\mathrm{Vol}(\mathcal{B}_G)$) using the Koksma–Hlawka inequality, see Theorem 7.5, since the function to be integrated is the indicator function of the set $\mathcal{B}_G$ and does not have bounded variation $V^{(n)}(g)$. Nevertheless, for comparison purposes, we still employ the QMC technique here, without relying on the theoretical bound of Theorem 7.5. Hence, for fixed values of C given in (19.14), we evaluate

$$A\big(q^{(i)}, v^{(i)}, C\big)$$

for $i = 1, \ldots, N$, and we compute

$$\frac{N_G}{N} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_{\mathcal{B}_G}\big(q^{(i)}, v^{(i)}, C\big)$$

where the sample size $N = 300{,}000$ is used.

The same approach for computing N_G/N is then followed using the Sobol' and Niederreiter sequences instead of the Halton sequence. Additional experiments regarding the stability of the network are performed using the quasi-Monte Carlo method for optimization, see Sect. 7.3. In this case, we construct an "optimal" grid minimizing the dispersions $d_N(q^{(1\ldots N)})$ and $d_N(v^{(1\ldots N)})$ according to the Sukharev sampling criterion, see Theorem 7.12. In particular, the point set $q^{(1\ldots N)}$ is constructed as follows

$$q_i^{(k)} = 25(2k-1)$$

for $i = 1, 2, 3, 4$ and $k = 1, \ldots, 20$. The sample size is therefore $N = 20^4 = 160{,}000$. It can be immediately verified, see Theorem 7.12, that this sample size guarantees $d_N(q^{(1\ldots N)}) \geq 0.025$. Similarly, for $v^{(1\ldots N)}$, we take

$$v_i^{(k)} = 0.025(2k-1)$$

for $i = 1, 2, 3, 4$ and $k = 1, \ldots, 20$. Then, we compute

$$\frac{N_G}{N} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_{\mathcal{B}_G}\big(q^{(i)}, v^{(i)}, C\big)$$

where $N = 160{,}000$.

19.4.7 Numerical Results

The results of the numerical experiments involving Monte Carlo and quasi-Monte Carlo methods are given in Figs. 19.8 and 19.9, which show the network stability degradation versus capacity. In particular, for the Monte Carlo method the empirical probability $\widehat{\mathbf{p}}_N$ is plotted for the 22 values of capacity given in (19.14). For the quasi-Monte Carlo method, deterministic estimates N_G/N of $\mathrm{Vol}(\mathcal{B}_G)$ for Halton,

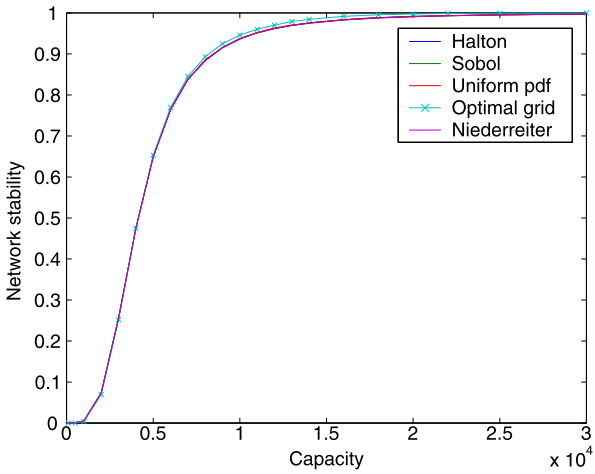

Fig. 19.8 Network stability versus capacity for $M = 4$ users using Monte Carlo and quasi-Monte Carlo methods

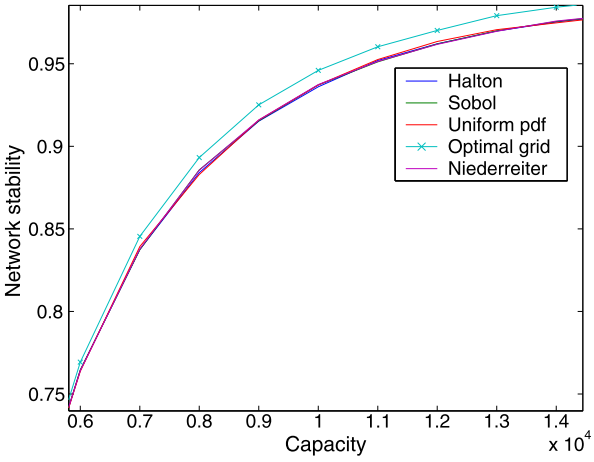

Fig. 19.9 A closer look at Fig. 19.8 for larger values of network stability

Sobol', Niederreiter sequences as well as for the optimal grid are also shown for the same values of capacity.

As a general comment, as expected, we observe that the stability of the system improves as capacity C increases. We also notice that the difference between the various sampling schemes is relatively small. However, the gridding method produces results which are slightly more "optimistic." Additional simulations show that these results are quite accurate even with a smaller sample size. One explanation for this phenomenon may be that the linearized system has relatively simple stability boundaries in the parameter space, see [17] for details. In the same paper, further results for a larger number of users are given, and stability properties of general network topologies with multiple bottleneck links are analyzed.

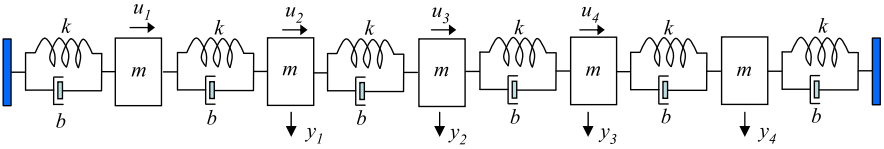

Fig. 19.10 Flexible structure with four noncolocated sensors and actuators

19.5 Probabilistic Robustness of Flexible Structures

We consider an example concerning a five-mass spring–damper model with four force actuators and four position sensors, as shown in Fig. 19.10. This plant has a standard second-order representation of the form

$$M\ddot{\xi} + L\dot{\xi} + K\xi = E_a u;$$
$$y = E_s \xi$$

where $\xi \in \mathbb{R}^5$ is the mass displacement vector, $u \in \mathbb{R}^4$ is the input force vector, $y \in \mathbb{R}^4$ is the output displacement vector, and

$$E_a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T, \qquad E_s = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

are the input and output influence matrices. The mass, damping and stiffness matrices, M, L, K, are given by

$$M = \operatorname{diag}\big([m \quad m \quad m \quad m \quad m]\big);$$
$$L = \begin{bmatrix} 2b & -b & 0 & 0 & 0 \\ -b & 2b & -b & 0 & 0 \\ 0 & -b & 2b & -b & 0 \\ 0 & 0 & -b & 2b & -b \\ 0 & 0 & 0 & -b & 2b \end{bmatrix};$$
$$K = \begin{bmatrix} 2k & -k & 0 & 0 & 0 \\ -k & 2k & -k & 0 & 0 \\ 0 & -k & 2k & -k & 0 \\ 0 & 0 & -k & 2k & -k \\ 0 & 0 & 0 & -k & 2k \end{bmatrix}.$$

The nominal values of the parameters are (in normalized units) $m = 1$, $k = 100$, and $b = 1$. A regulator $R(s)$ has been synthesized based on the nominal plant model, in order to improve the dynamic response of the plant. The state space representation of this regulator is given by

$$A_R = \begin{bmatrix} 0 & -0.1654 & -0.0761 & -0.0301 & -0.0095 \\ 0 & -0.3897 & -0.1951 & -0.0858 & -0.0301 \\ 0 & -0.1951 & -0.3983 & -0.1948 & -0.0764 \\ 0 & -0.0858 & -0.1948 & -0.3882 & -0.1644 \\ 0 & -0.0301 & -0.0764 & -0.1644 & -0.3114 \\ -200.0136 & 100.0057 & -0.0012 & -0.0139 & -0.0084 \\ 99.9777 & -199.6176 & 99.9514 & -0.0224 & -0.0139 \\ -0.0157 & 99.8529 & -199.6369 & 99.9510 & -0.0003 \\ -0.0019 & -0.0900 & 99.8533 & -199.6191 & 100.0042 \\ 0 & -0.0272 & -0.0578 & 99.9230 & -199.5654 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -2.3073 & 0.8348 & -0.0769 & -0.0304 & -0.0096 \\ 0.8348 & -2.3845 & 0.8042 & -0.0864 & -0.0303 \\ -0.0769 & 0.8042 & -2.3947 & 0.8039 & -0.0767 \\ -0.0304 & -0.0864 & 0.8039 & -2.3860 & 0.8339 \\ 0 & 0 & 0 & 1 & -2.0000 \end{bmatrix};$$

$$B_R = \begin{bmatrix} 0.1654 & 0.0761 & 0.0301 & 0.0095 \\ 0.3897 & 0.1951 & 0.0858 & 0.0301 \\ 0.1951 & 0.3983 & 0.1948 & 0.0764 \\ 0.0858 & 0.1948 & 0.3882 & 0.1644 \\ 0.0301 & 0.0764 & 0.1644 & 0.3114 \\ -0.0073 & 0.0085 & 0.0167 & 0.0098 \\ -0.4009 & 0.0561 & 0.0378 & 0.0229 \\ 0.1167 & -0.3798 & 0.0715 & 0.0343 \\ 0.0719 & 0.1109 & -0.3885 & 0.0555 \\ 0.0272 & 0.0578 & 0.0770 & -0.4346 \end{bmatrix};$$

$$C_R = \begin{bmatrix} -0.0136 & -0.0016 & 0.0073 & 0.0029 & 0.0014 \\ -0.0223 & -0.0185 & 0.0074 & 0.0154 & 0.0090 \\ -0.0157 & -0.0305 & -0.0166 & 0.0225 & 0.0340 \\ -0.0019 & -0.0181 & -0.0357 & -0.0076 & 0.0596 \end{bmatrix}$$

$$\begin{bmatrix} -0.3073 & -0.1652 & -0.0769 & -0.0304 & -0.0096 \\ -0.1652 & -0.3845 & -0.1958 & -0.0864 & -0.0303 \\ -0.0769 & -0.1958 & -0.3947 & -0.1961 & -0.0767 \\ -0.0304 & -0.0864 & -0.1961 & -0.3860 & -0.1661 \end{bmatrix}$$

and $D_R = 0_{4,4}$. Notice that, although the uncontrolled system is structurally stable, the introduction of feedback may cause instabilities for some parameter values. Hence, the objective is to analyze the robustness properties of the feedback connection of the plant and the above regulator, with respect to variations of the plant

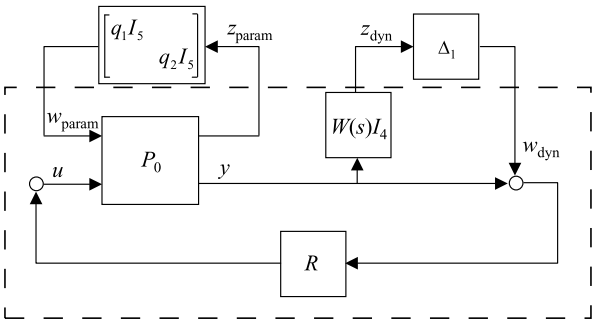

Fig. 19.11 Controlled flexible structure with uncertainty

parameters and unmodeled dynamics. Specifically, we consider parametric uncertainty on the damping and stiffness parameters

$$b = b_0 + 0.5q_1, \qquad k = k_0 + 0.5q_2$$

with $|q_1| \leq 1$ and $|q_2| \leq 1$, and dynamic uncertainty with a frequency shape described by the weight function

$$W(s) = \frac{0.1035s + 1}{0.02071s + 1}.$$

With a standard procedure, we express the uncertain plant as a feedback connection of an augmented nominal plant (say P_0) and the uncertainty. The augmented plant P_0 has ten additional inputs and outputs, corresponding to the uncertain parameters b and k, each of which is repeated five times in the state space representation of the plant. The flexible structure with controller and uncertainty is then modeled in the classical M–Δ form, as shown in Fig. 19.11, where the "M" part of the interconnection is enclosed in the dashed box.

The matrix Δ representing the uncertainty in the system is structured and consists of two repeated real parameters q_1, q_2 and one full dynamic block $\Delta_1 \in \mathbb{C}^{4,4}$. That is, Δ is assumed to belong to the structured set

$$\mathbb{D} = \left\{ \Delta : \Delta = \text{bdiag}(q_1 I_5, q_2 I_5, \Delta_1) \right\}.$$

For this M–Δ system, lower and upper bounds $1/\mu_+$ and $1/\mu_-$ of the robustness margin $1/\mu$ have been computed with the Matlab μ Analysis and Synthesis Toolbox [38], obtaining

$$1/\mu_+ = 1.172, \quad 1/\mu_- = 1.185.$$

This deterministic analysis shows that the system interconnection is stable for all structured perturbations Δ having norm smaller than $1/\mu_+$, see Sect. 3.7.

Next, we proceed to a probabilistic analysis of the robust stability properties of the system for perturbations whose radius goes beyond the deterministic margin $1/\mu_+$. That is, we study how the probability of stability degrades with increasing radius of the uncertainty. Formally, for fixed $\rho > 0$, we consider the structured set

$$\mathcal{B}_{\mathbb{D}}(\rho) = \left\{ \Delta \in \mathbb{D} : \bar{\sigma}(\Delta) \leq \rho \right\}$$

and assume that the uncertainty $\boldsymbol{\Delta}$ is a random matrix with uniform probability distribution in this set. Hence, letting (A, B, C, D) be a state space representation of system M, we define

$$\text{PR}\{\text{stability}\} = \text{PR}\big\{\boldsymbol{\Delta} \in \mathcal{B}_{\mathbb{D}}(\rho) : A + B\boldsymbol{\Delta}(I - D\boldsymbol{\Delta})^{-1}C \text{ is stable}\big\}$$

and, for given $\epsilon \in (0, 1)$, we define the probabilistic stability radius

$$\bar{r}(\epsilon) \doteq \sup\big\{\rho : \text{PR}\{\text{stability}\} \geq 1 - \epsilon\big\}.$$

Given a probability level ϵ, the probabilistic stability radius $\bar{r}(\epsilon)$ gives the maximum size of the perturbation $\boldsymbol{\Delta}$, measured according to the spectral norm, so that the probability PR{stability} is at least $1 - \epsilon$. Once PR{stability} is estimated by means of a randomized algorithm (see e.g. Sect. 10.3), we construct the performance degradation function, i.e. the plot of the probability of stability as a function of the radius ρ. This plot may be compared with the classical worst-case stability margin $1/\mu_+$, obtaining

$$\bar{r}(\epsilon) \geq 1/\mu_+$$

for any $\epsilon \in (0, 1)$. This fact, in turn, implies that the margin computed with probabilistic methods is always larger than the classical worst-case margin, at the expense of a risk expressed in probability.

Taking $\epsilon = \delta = 0.02$, by means of the Chernoff bound (see Sect. 8.3)

$$N \geq \frac{\log \frac{2}{\delta}}{2\epsilon^2}$$

we obtained $N \geq 23,026$. Then, we estimated the performance degradation function for 40 equispaced values of ρ in the range $[0.15, 2.1]$. For each grid point $\rho_k \in [0.15, 2.1]$ the probability of stability is estimated as

$$\widehat{\mathbf{p}}_N(\rho_k) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(\boldsymbol{\Delta}^{(i)}\big)$$

where

$$\mathbb{I}\big(\boldsymbol{\Delta}^{(i)}\big) = \begin{cases} 1 & \text{if } A + B\boldsymbol{\Delta}^{(i)}(I - D\boldsymbol{\Delta}^{(i)})^{-1}C \text{ is stable;} \\ 0 & \text{otherwise.} \end{cases}$$

where $\boldsymbol{\Delta}^{(i)}$ is extracted uniformly at random in the set $\mathcal{B}_{\mathbb{D}}(\rho_k)$. The accuracy of this estimation is such that

$$\text{PR}\big\{\big|\widehat{\mathbf{p}}_N(\rho_k) - \text{PR}\{\text{stability}\}\big| \leq 0.02\big\} \geq 0.98.$$

The plot of the obtained realizations of $\widehat{\mathbf{p}}_N(\rho)$ as a function of ρ is shown in Fig. 19.12 together with the deterministic robustness margin $1/\mu_+$. From this plot we observe, for instance, that if a 1 % loss of probabilistic performance may be tolerated, then the stability margin may be increased by approximately 64 % with respect to its deterministic counterpart. In fact, the risk-adjusted stability margin for a probability level $\epsilon = 0.01$ is $\bar{r}(0.01) \approx 1.93$. In addition, we notice that the estimated probability is equal to one up to $\rho \approx 1.4$. We conclude that, in this example,

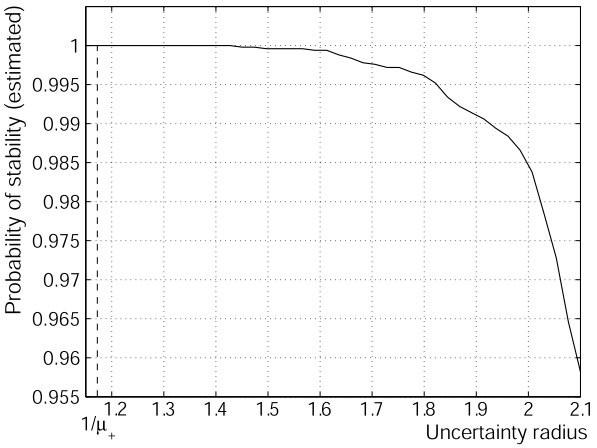

Fig. 19.12 Degradation of the probability of stability for the controlled structure

even if the upper and lower bounds of μ approximately coincide, so that $1/\mu$ is a nonconservative deterministic measure of robustness, this measure turns out to be quite conservative in a probabilistic sense.

19.6 Stability of Quantized Sampled-Data Systems

In this section we study the application of RAs for quadratic stability of sampled-data systems with memoryless quantizers, see [211] for a more detailed analysis. Quantization involved in control systems has recently become an active research topic, see e.g. [73, 159, 213]. The need for quantization inevitably arises when digital networks are part of the feedback loop and it is of interest to reduce the data rate necessary for the transmission of control signals. Then, a fundamental issue is to determine the minimum information to achieve the control objectives. Clearly, if a quantized discrete-time signal takes only a finite number of fixed values, then the trajectories may go close to an equilibrium but not converge, so that asymptotic stability is not achieved. Then, various problems may be posed. For example, it is of interest to clarify how close the trajectories get to the equilibrium point and, if the sampling period is large, how close do the trajectories stay at the equilibrium between sampling instants. These questions are addressed in the references previously listed, and bounds on the trajectories are determined analytically, albeit at the expense of crude conservatism.

19.6.1 Problem Setting

We now present the setup of quantized control systems and formulate the related quadratic stability problem. Consider the continuous-time system depicted in Fig. 19.13.

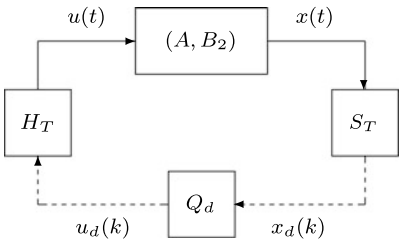

Fig. 19.13 Quantized sampled-data system where S_T is the sampler, H_T is the zero-th order hold and Q_d is the quantizer

The pair (A, B_2), which is assumed stabilizable, represents a linear time-invariant plant with the state equation

$$\dot{x} = Ax + B_2 u \tag{19.16}$$

with initial state $x(0) = x_0 \in \mathbb{R}^{n_s}$ given but arbitrary. We study the case when A is not stable, because otherwise the problem becomes trivial. The output of the sampler S_T is a discrete-time signal given by

$$x_d(k) = x(kT)$$

where k is a positive integer and $T > 0$ is the sampling period. The output of zero-th order hold H_T is a continuous-time signal defined by

$$u(t) = u_d(k)$$

where $t \in [kT, (k+1)T)$.

We now introduce the definitions of cell and (memoryless) quantizer.

Definition 19.1 (Cell of the quantizer) Given a countable index set $\mathcal{I}$, a partition $\{\mathcal{Q}_i\}_{i\in\mathcal{I}} \subset \mathbb{R}^{n_s}$ consists of bounded sets (called the *cell* of the quantizer) such that:

1. $\mathcal{Q}_i \cap \mathcal{Q}_k = \emptyset$ for $i \neq k$ and $\bigcup_{i\in\mathcal{I}} \mathcal{Q}_i = \mathbb{R}^{n_s}$;
2. $0 \in \mathcal{Q}_0$;
3. $0 \notin \partial\mathcal{Q}_i$, $i \in \mathcal{I}$.

Definition 19.2 (Memoryless quantizer) Given a set of cells $\{\mathcal{Q}_i\}_{i\in\mathcal{I}}$ and a set of inputs $\{u_i\}_{i\in\mathcal{I}}$, a quantizer Q_d is a mapping from $\mathbb{R}^{n_s}$ to $\{u_i\}_{i\in\mathcal{I}}$ defined by

$$Q_d(x) = u_i \quad \text{if } x \in \mathcal{Q}_i,\ i \in \mathcal{I}. \tag{19.17}$$

In words, a quantizer Q_d maps a state $x \in \mathbb{R}^{n_s}$ in a cell $\mathcal{Q}_i$ to the corresponding input u_i. Various types of quantizer are studied, including the uniform and the logarithmic quantizer, see Fig. 19.14 for an illustration of the latter.

Next, following the quadratic stabilizability approach described in Sect. 4.2.2, we define a quadratic Lyapunov function $V(x) = x^T Px$, for positive definite $P \succ 0$. The time derivative of this function along trajectories of the system (19.16) is given by

$$\dot{V}(x,u) = \frac{\mathrm{d}}{\mathrm{d}t} V\big(x(t)\big) = (Ax + B_2u)^T Px + x^T P(Ax + B_2u).$$

Furthermore, for positive definite $R \succ 0$, the set

$$\left\{x \in \mathbb{R}^{n_s} : \dot{V}(x,u) \leq -x^T R x\right\}$$

is the set of states at which the Lyapunov function decreases when the control u is applied. In fact, taking the state feedback $u = Kx$, we compute

$$\dot{V}(x,u) = x^T (A + B_2 K)^T P x + x^T P (A + B_2 K) x.$$

Hence, we can choose

$$R = (A + B_2 K)^T P + P(A + B_2 K).$$

Based on this *control Lyapunov function* approach, we study quadratically attractive sets for quantized sampled-data systems. This study leads to the construction of a specific randomized algorithm. We now state the definition of a quadratically attractive set.

Definition 19.3 (Quadratically attractive set) Given $\overline{r} > 0$, positive definite matrices $P, R \succ 0$ and a fixed quantizer Q_d, for the closed-loop system in Fig. 19.13, the ball $\mathcal{B}_{\|\cdot\|_2}(\underline{r}) = \{x \in \mathbb{R}^{n_s} : \|x\|_2 \leq \underline{r}\}$ of radius $\underline{r} > 0$, is *quadratically attractive* from $\mathcal{B}_{\|\cdot\|_2}(\overline{r})$ with respect to P, R if every trajectory of the system (19.16) with $x(0) \in \mathcal{B}_{\|\cdot\|_2}(\overline{r})$ satisfies either

$$\dot{V}\big(x(t), u(t)\big) \leq -x(t)^T R x(t)$$

or

$$x(t) \in \mathcal{E}\big(0, \underline{r}^2 \lambda_{\min} P^{-1}\big) = \left\{x \in \mathbb{R}^{n_s} : x^T P x \leq \underline{r}^2 \lambda_{\min}\right\}. \tag{19.18}$$

for all $t \geq 0$, where $\lambda_{\min}$ denotes the smallest eigenvalue of P.

We note that quadratic attractiveness coincides with (asymptotic) stability if $\underline{r} = 0$ and $\overline{r} = \infty$. We also observe that this definition of attractiveness for sampled-data system is in the continuous-time domain. Hence, the Lyapunov function $V(x(t))$ must decrease at a certain rate even between sampling instants and the ellipsoid $\mathcal{E}(0, \underline{r}^2 \lambda_{\min} P^{-1}) = \{x \in \mathbb{R}^{n_s} : V(x) \leq \underline{r}^2 \lambda_{\min}\}$ is an invariant set. In particular, $\mathcal{E}(0, \underline{r}^2 \lambda_{\min} P^{-1})$ is the largest level set of $V(x) = x^T P x$ contained in $\mathcal{B}_{\|\cdot\|_2}(\underline{r})$. We now formally define the quadratic stability problem of quantized sampled-data systems.

Problem 19.2 (Quadratic stability of quantized systems) *Given $\overline{r} > 0$, a fixed quantizer Q_d, a sampling period $T > 0$ and $R \succ 0$, find $P \succ 0$ and $\underline{r} > 0$ such that, for the closed-loop system in Fig. 19.13, the ball $\mathcal{B}_{\|\cdot\|_2}(\underline{r})$ is quadratically attractive from the ball $\mathcal{B}_{\|\cdot\|_2}(\overline{r})$ with respect to P, R.*

Remark 19.1 (Quadratic attractiveness of quantized systems) The setup of this problem is similar to that in [213], where stabilization of linear sampled-data systems with memoryless quantizers is studied. More precisely, the quantizer design problem considered in this reference can be roughly summarized as follows: given $\overline{r} > 0$,

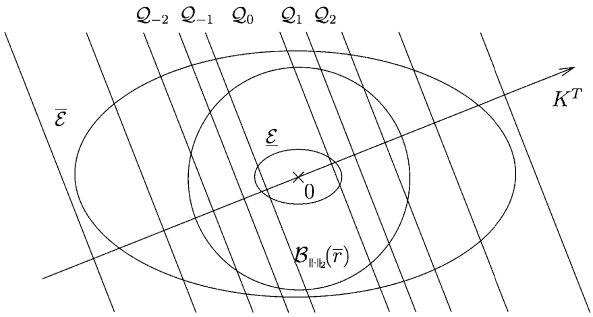

Fig. 19.14 Invariant sets and partition of a logarithmic quantizer

state feedback $u = Kx$ such that $A + B_2K$ is stable, and matrices $P, R \succ 0$, design a quantizer Q_d with sampling period T and $\underline{r}$ for quadratic attractiveness with respect to P, R. As discussed previously, we can easily verify that, taking the state feedback $u = Kx$, the closed-loop continuous-time system $\dot{x} = (A + B_2K)x$ is quadratically stable with respect to P, R, where $R = (A + B_2K)^T P + P(A + B_2K)$.

The class of quantizers considered in [213] is somewhat restricted compared with that in Definition 19.2, but it allows the finding of an analytic solution. However, the drawback of the approach proposed in [213] lies in the conservatism in the design, especially in the derivation of T and $\underline{r}$. The analysis method based on randomized algorithms provides a way to obtain less conservative estimates of the performance of the designed system, at the expense of obtaining a probabilistic solution instead of a guaranteed one.

We denote by $\phi(x_0, u, t)$ the state of the system (19.16) at time t corresponding to the initial conditions $x_0 = x(0) \in \mathbb{R}^{n_s}$ and the constant control input u. That is, we have

$$\phi(x_0, u, t) = \mathrm{e}^{At}x_0 + \left[\int_0^t \mathrm{e}^{A\tau} B_2 \,\mathrm{d}\tau\right]u.$$

Similar to (19.18), we also consider the ellipsoid

$$\mathcal{E}\left(0, \overline{r}^2\lambda_{\max}P^{-1}\right) = \left\{x \in \mathbb{R}^{n_s} : V(x) \le \overline{r}^2\lambda_{\max}\right\}$$

where $\lambda_{\max}$ is the largest eigenvalue of P. We note that $\mathcal{E}(0, \overline{r}^2\lambda_{\max}P^{-1})$ is the smallest ellipsoid containing $\mathcal{B}_{\|\cdot\|_2}(\overline{r})$. In addition, we consider two ellipsoids $\underline{\mathcal{E}} \subseteq \overline{\mathcal{E}} \subset \mathbb{R}^{n_s}$ which provide estimates of the invariant sets $\mathcal{E}(0, \underline{r}^2\lambda_{\min}P^{-1})$ and $\mathcal{E}(0, \overline{r}^2\lambda_{\max}P^{-1})$. A sketch of these sets and a partition of a logarithmic quantizer is given in Fig. 19.14. We notice that the partition cells are strips orthogonal to the subspace spanned by K^T and become wider as the distance from the origin grows.

We now state without proof a simple sufficient condition for quadratic attractiveness, see [211] for details.

Lemma 19.3 (Quadratic attractiveness) *Suppose that* $P \succ 0$ *and* $\underline{r} > 0$ *satisfy the following conditions*:

1. $\underline{\mathcal{E}} \subset \mathcal{E}(0, \underline{r}^2\lambda_{\min}P^{-1}) \subset \mathcal{E}(0, \overline{r}^2\lambda_{\max}P^{-1}) \subset \overline{\mathcal{E}}$;

2. *For every* $x_0 \in \overline{\mathcal{E}}$ *and* $t \in [0, T]$, *we have*

$$\phi\big(x_0, Q_d(x_0), t\big) \in \big\{x \in \mathbb{R}^{n_s} : \dot{V}\big(x, Q_d(x_0)\big) \leq -x^T R x\big\} \cup \underline{\mathcal{E}}. \tag{19.19}$$

Then, the ball $\mathcal{B}_{\|\cdot\|_2}(\underline{r})$ *is quadratically attractive from* $\mathcal{B}_{\|\cdot\|_2}(\overline{r})$ *with respect to* P, R *for the closed-loop system shown in Fig.* 19.13.

There are several consequences of this lemma. First, since the ellipsoid $\mathcal{E}(0, \overline{r}^2 \lambda_{\min} P^{-1})$ is an invariant set contained in $\overline{\mathcal{E}}$, all trajectories starting in the ball $\mathcal{B}_{\|\cdot\|_2}(\overline{r})$ remain in this set. A second consequence is that the region where the Lyapunov function $V(x(t))$ increases is contained in $\underline{\mathcal{E}}$. Therefore, all trajectories of the system (19.16) enter $\mathcal{E}(0, \underline{r}^2 \lambda_{\min} P^{-1})$, because this is also an invariant set.

19.6.2 Randomized Algorithm

In this section, we describe the iterative algorithm used for constructing a quadratic Lyapunov function and for finding $P \succ 0$ such that the second condition in Lemma 19.3 holds. This algorithm falls in the general category of sequential randomized algorithms described in Chap. 11. However, we require probability density functions having supporting sets depending on the ellipsoid $\overline{\mathcal{E}}$ and the quantization cells $\mathcal{Q}_i$ and we perform random generation in both state and time. In particular, we consider a probability density function $f_{\mathbf{x},\mathbf{i}}(x, i)$ associated with the state x and the cell index i of the quantizer and a pdf $f_{\mathbf{t}}(t) > 0$ for $t \in [0, T]$. At the kth iteration, the algorithm randomly generates a pair of state and index $(\mathbf{x}^{(k)}, \mathbf{i}^{(k)})$ according to $f_{\mathbf{x},\mathbf{i}}(x, i)$ and also randomly generates a set of ℓ time instants $\{\mathbf{t}^{(i)}\}_{i=0}^{\ell-1} \subset [0, T]$ according to $f_{\mathbf{t}}(t)$. In particular, for $P, R \succ 0$, x, i and $t \in [0, T]$, we consider the function

$$v(P, x, i, t) = \dot{V}\big(\phi(x, u_i, t), u_i\big) + \phi(x, u_i, t)^T R \phi(x, u_i, t). \tag{19.20}$$

Clearly, this function has the property that

$$v(P, x, i, t) \leq 0 \tag{19.21}$$

if and only if

$$\phi(x, u_i, t) \in \big\{x \in \mathbb{R}^{n_s} : \dot{V}(x, u_i) \leq -x^T R x\big\}.$$

Therefore, checking whether the state of the system at time t enters the set of states $\{x \in \mathbb{R}^{n_s} : \dot{V}(x, u_i) \leq -x^T R x\}$ at which the Lyapunov function decreases can be executed by a verification of the sign of the function (19.20). Then, at each step of the algorithm, for randomly generated $(\mathbf{x}, \mathbf{i})$ and $\mathbf{t}$, a matrix P is sought for achieving condition (19.21). Since v is linear in P, its gradient can be easily computed in closed form. Taking an arbitrary initial matrix P, an update in P may be computed using a gradient-based or ellipsoid algorithm, see Chap. 11 for details.

Remark 19.2 (Randomized algorithm for quantized systems) In [211], the various steps of the algorithm are described precisely. In particular, it is shown that convergence to a feasible solution $P \succ 0$ in a finite number of iterations is achieved with probability one, provided that a solution exists and minor technical assumptions are satisfied. This feasible solution is a matrix $P \succ 0$ which meets the two conditions given by Lemma 19.3.

The algorithm provides a systematic way to analyze the quantized system with less conservatism than other approaches, but with the drawback that probabilistic results, instead of guaranteed solutions, are found. On the other hand, the algorithm is based on the simple sufficient condition of Lemma 19.3 and has certain redundancy. For example, the lemma requires that trajectories for all $x_0 \in \overline{\mathcal{E}}$ and all $t \in [0, T]$ satisfy (19.19). Therefore, the number of trajectories is generally very high and the randomized algorithm may become practically intractable. In order to improve its efficiency, in [211] it is shown that the number of random trajectories which are generated can be greatly reduced. In particular, it is shown that it is not necessary to define density functions and perform randomization of the set $\overline{\mathcal{E}}$, but it suffices to consider the boundaries of $\underline{\mathcal{E}}$, $\overline{\mathcal{E}}$ and of the quantization cells $\mathcal{Q}_i$.

More generally, in [211] the specific structure of the quantized sampled-data systems is exploited in order to reduce the computational complexity of the randomized algorithm. For nonlinear systems, deterministic computational methods are sought for obtaining less conservative invariant sets, see e.g. [59, 60]. Hence, this method can be viewed as an alternative to finding probabilistically invariant sets for quantized systems.

19.6.3 Numerical Experiments

The randomized algorithm has been utilized for the magnetic ball levitation system shown in Fig. 19.15. Details regarding this apparatus are given in [213], where the quadratic attractiveness for the sampled-data system with a logarithmic quantizer designed by an analytic method is analyzed. This method, however, is known to be subject to conservatism. In this section we present numerical results regarding probabilistic bounds obtained by means of a randomized algorithm, and clarify the extent of conservatism of the analytic deterministic condition.

In Fig. 19.15, a steel ball of mass M is levitated by the electromagnet. The position y of the ball is kept at an equilibrium through controlling the voltage v. The current in the coil is i, and the resistance and inductance of the magnet are R and L respectively. The system is linearized around an equilibrium $[y_0 \ \dot{y}_0 \ i_0]^T$ for the nominal voltage $v_0 = 10$ V. The resulting state is given by

$$x = [y - y_0 \quad \dot{y} - \dot{y}_0 \quad i - i_0]^T$$

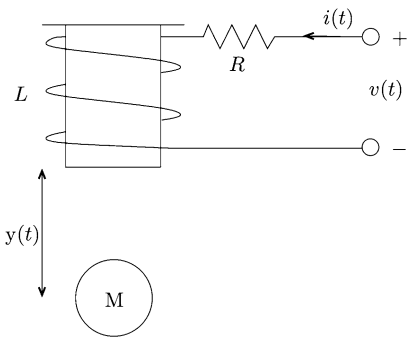

Fig. 19.15 Magnetic ball levitation system

and the system matrices A and B_2 are

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 2\frac{Rg}{v_0}\sqrt{\frac{Mg}{\kappa}} & 0 & -2\frac{Rg}{v_0} \\ 0 & 0 & -\frac{R}{L} \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}$$

where $M = 0.068$ kg, $R = 10\Omega$, $L = 0.41$ H, $\kappa = 3.3\times10^{-5}$ Nm^2/A^2 and $g = 9.8$ m/s^2.

The sampled-data controller designed in [213] for this system is now described. First, the optimal state feedback K is determined solving a Riccati equation corresponding to the linear quadratic regulator

$$A^T P_0 + P_0 A - P_0 B_2 Q_{uu}^{-1} B_2^T P_0 + Q_{xx} = 0$$

where $Q_{uu} = 0.878$ and

$$Q_{xx} = \begin{bmatrix} 1.15\times10^7 & 2.17\times10^5 & -5.52\times10^4 \\ 2.17\times10^5 & 4.11\times10^3 & -1.04\times10^3 \\ -5.52\times10^4 & -1.04\times10^3 & 265 \end{bmatrix}.$$

Further details regarding the selection of these values are given in [213]. The solution P_0 of the Riccati equation is given by

$$P_0 = \begin{bmatrix} 7.80\times10^5 & 1.48\times10^4 & -3.75\times10^3 \\ 1.48\times10^4 & 279 & -70.9 \\ -3.75\times10^3 & -70.9 & 18.0 \end{bmatrix}$$

whose eigenvalues are

$$\lambda(P_0) = \begin{bmatrix} 7.81\times10^5 & 2.09\times10^{-4} & 9.40\times10^{-5} \end{bmatrix}^T.$$

Then, a logarithmic quantizer Q_d and a sampling period T are designed so that the ball $\mathcal{B}_{\|\cdot\|_2}(\underline{r})$ is quadratically attractive from $\mathcal{B}_{\|\cdot\|_2}(\overline{r})$ with respect to P_0, γR_0, where $\underline{r} = 32$, $\overline{r} = 10$, $\gamma = 0.10$ and

$$R_0 = (A + B_2 K)^T P_0 + P_0 (A + B_2 K).$$

We notice that the presence of sampling and quantization requires sacrificing the decay rate by introducing γ.

We now describe the quantizer considered here. That is, the index set is $\mathcal{I} = \{0, \pm 1, \pm 2, \ldots\}$ and the partition cells $\mathcal{Q}_i, i \in \mathcal{I}$, are given by

$$\mathcal{Q}_i = \begin{cases} \{x : Kx \in (-\alpha, \alpha)\} & \text{if } i = 0; \\ \{x : Kx \in [\text{sgn}(i)\alpha\delta^{|i|-1}, \text{sgn}(i)\alpha\delta^{|i|})\} & \text{otherwise} \end{cases}$$

where $\alpha = 0.451$ and $\delta = 1.78$. The control input values are

$$u_i = \text{sgn}(i)\beta\delta^{|i|-1}$$

where $\beta = 0.652$ and $i \in \mathcal{I}$. The designed sampling period is $T_s = 3.07 \times 10^{-3}$. We observe that the data considered here are different than the original in [213] for a change in the coordinate system.

We now discuss the conservatism in the deterministic design considered. First, although $\underline{r} = 32$ is a relatively large value, we observed in simulation that the trajectories of the resulting system generally entered a ball of radius 0.02. We notice that the different orders of magnitude are partially due to the largest eigenvalue of P_0, which is equal to 7.81×10^5 and makes its level sets very "narrow." Moreover, even when the size of T was doubled, the trajectories still entered a ball of a similar radius. To obtain less conservative results, we had to run the algorithm several times consecutively, starting each run with the matrix P resulting from the previous run. For each run, we used 10,000 samples in state and, for each sampled state, four samples in time. When there was no update for two runs in a row, we modified $\underline{r}$ and/or T.

We now describe the runs with $T = 1.5T_s$. In the first run, we set $P^{(1)} = P_0$ and we also took $\underline{\mathcal{E}}$ and $\overline{\mathcal{E}}$ to be level sets of $P^{(1)}$ so that $\mathcal{B}_{\|\cdot\|_2}(\overline{r}) \subset \overline{\mathcal{E}}$ and $\underline{\mathcal{E}} \subset \mathcal{B}_{\|\cdot\|_2}(\underline{r})$ with $\underline{r} = 6.0$. Two updates in the cells of the quantizer were observed and $P^{(2)}$ was obtained. In the next run, we started with $P^{(2)}$ and the corresponding sets $\overline{\mathcal{E}}$ and $\underline{\mathcal{E}}$. In this case there was no update in this run nor in the next. Therefore, we set $\underline{r}$ to 0.50 and continued in this manner. The results of the runs are summarized in Table 19.4. After 19 runs with 28 updates, we obtained $\underline{r} = 0.053$. The resulting matrix is given by

$$P^{(19)} = \begin{bmatrix} 7.80 \times 10^5 & 1.48 \times 10^4 & -3.75 \times 10^3 \\ 1.48 \times 10^4 & 290 & -70.3 \\ -3.75 \times 10^3 & -70.3 & 21.2 \end{bmatrix}$$

with eigenvalues

$$\lambda\left(P^{(19)}\right) = \begin{bmatrix} 7.81 \times 10^5 & 11.3 & 3.20 \end{bmatrix}^T.$$

We remark that the largest eigenvalues of $P^{(1)}$ and $P^{(19)}$ almost coincide, but the others are much larger in $P^{(19)}$. This implies that the level sets of $P^{(19)}$ are "rounder," and hence a smaller attractive ball could be obtained.

Subsequently, a time response was calculated for the initial state

$$x(0) = \begin{bmatrix} 0.70 \times 10^{-3} & 0 & 0 \end{bmatrix}^T.$$

Table 19.4 The results of the 19 runs for different values of $\underline{r}$ and $T = 1.5T_s$

Run	$\underline{r}$	Number of updates	Cell indices at updates
1	6	2	0, −10
2, 3	6	0	none
4	0.5	15	0, 1, −14
5	0.5	9	−6, −11, 13
6	0.5	1	12
7	0.5	1	13
8, 9	0.5	0	none
10–19	0.25–0.053	0	none

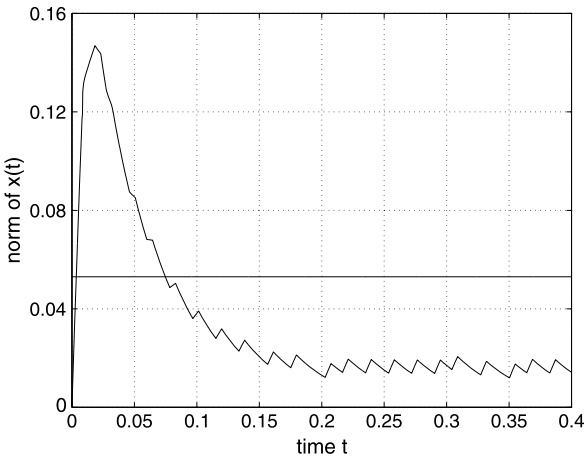

Fig. 19.16 Plot of Euclidean norm of $x(t)$ as a function of t

The plot of the Euclidean norm of $x(t)$ is depicted in Fig. 19.16 as a function of t. The horizontal line is shown for $\underline{r} = 0.053$. The trajectory of the system goes below this line; for this trajectory we plotted $V(x(t))$ and the related function $v(P, x, i, t)$ in Fig. 19.17 where the solid lines are for $P = P^{(19)}$ and the dashed lines for $P = P^{(1)}$.

In the top plot the horizontal line corresponds to the size of the level set $\mathcal{E}(0, \underline{r}^2 \lambda_{\min} P^{-1})$ corresponding to $P^{(19)}$ entered by all trajectories. A similar line for the level set corresponding to $P^{(1)}$ is plotted as well, but this is not visible because it is too close to zero. For quadratic attractiveness, the function in the bottom plot should be smaller than zero whenever $V(x(t))$ is above the horizontal line in the top plot for every t. The solid lines satisfy this, but not the dashed lines. In a similar way, we obtained $\underline{r} = 0.040$ for $T = T_s$ and $\underline{r} = 0.062$ for $T = 1.7T_s$.

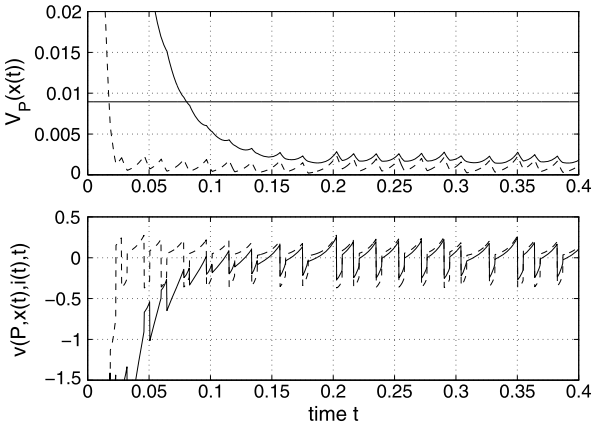

Fig. 19.17 *Top*: plot of $V(x(t))$ for $P^{(19)}$ (*solid*) and $V(x(t))$ for $P^{(1)}$ (*dashed*). *Bottom*: plot of $v(P, x, i, t)$ for $P^{(19)}$ (*solid*) and $v(P, x, i, t)$ for $P^{(1)}$ (*dashed*)

19.7 Randomized Algorithms Control Toolbox

A MATLAB toolbox has been developed in order to facilitate and diffuse the use of randomized techniques within the systems and control community. The RANDOMIZED ALGORITHMS CONTROL TOOLBOX (RACT), see [390], provides convenient uncertain object manipulation and implementation of randomized methods using state-of-the-art theoretical and algorithmic results. Two main features of the package are a functional approach with `m`-file templates and a definition of design problems in generic LMI format using the widely used YALMIP syntax. This first release of the toolbox provides an easy-to-use interface of current randomized algorithms for control and is intended to be used by researchers, engineers and students interested in uncertain systems, robust control, optimization and related applications. The package can be freely downloaded from http://ract.sourceforge.net.

RACT features currently include

1. Definition of a variety of uncertain objects: scalar, vector and matrix uncertainties, with different density functions;
2. Easy and fast sampling of uncertain objects of almost any type;
3. Randomized algorithms for probabilistic performance verification and probabilistic worst-case performance;
4. Randomized algorithms for feasibility of uncertain LMIs using stochastic gradient, ellipsoid or cutting plane methods;
5. Optimal design methods using scenario approach.

Appendix

A.1 Transformations Between Random Matrices

We next give a generalization of Theorem 14.2 to the case of functions of random matrices, see for instance [312].

Theorem A.1 (Functions of random matrices) *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be two random matrices with the same number of free elements* $(x_1, \dots, x_p)$ *and* $(y_1, \dots, y_p)$ *respectively. Let the pdf of* $\mathbf{X}$ *be* $f_{\mathbf{X}}(X)$, *and let* $\mathbf{X}, \mathbf{Y}$ *be related by a one-to-one transformation* $\mathbf{Y} = g(\mathbf{X})$. *Let* $h(\cdot) \doteq g^{-1}(\cdot)$, *then the pdf* $f_{\mathbf{Y}}(Y)$ *is*

$$f_{\mathbf{Y}}(Y) = f_{\mathbf{X}}\big(h(Y)\big) J(X \to Y)$$

where the Jacobian $J(X \to Y)$ *is defined as*

$$J(X \to Y) \doteq \begin{vmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_2}{\partial y_1} & \dots & \frac{\partial x_p}{\partial y_1} \\ \frac{\partial x_1}{\partial y_2} & \frac{\partial x_2}{\partial y_2} & \dots & \frac{\partial x_p}{\partial y_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial x_1}{\partial y_p} & \frac{\partial x_2}{\partial y_p} & \dots & \frac{\partial x_p}{\partial y_p} \end{vmatrix}.$$

For the purpose of calculation, it is sometimes desirable to express the Jacobian in terms of the free elements of X and Y, for example $J(X \to Y)$ might be written as $J(x_1, \dots, x_p \to y_1, \dots, y_p)$. More generally, if the matrices $X_1, \dots, X_k$ and $Y_1, \dots, Y_m$ satisfy the equations $\mathbf{Y}_i = g_i(x_1, \dots, x_k)$, $i = 1, \dots, m$, and $(X_1, \dots, X_k)$, $(Y_1, \dots, Y_m)$ have the free elements $(x_1, \dots, x_p)$ and $(y_1, \dots, y_p)$ respectively, then the Jacobian of the transformation from $(X_1, \dots, X_k)$ to $(Y_1, \dots, Y_m)$ will be denoted as $J(X_1, \dots, X_k \to Y_1, \dots, Y_m)$.

Remark A.1 (Many-to-few mappings) To handle the case when a transformation maps the random variables $\mathbf{x} \in \mathbb{R}^n$ to $\mathbf{y} \in \mathbb{R}^m$, with $m < n$ (i.e. the transformation is not one-to-one, but maps many to fewer variables) we may proceed as follows. Let the original transformation be

$$y_i = g_i(x_1, \dots, x_n), \quad i = 1, \dots, m; \quad m < n.$$

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0,

If additional slack functions $y_i = g_i(x_1, \dots, x_n)$, $i = m+1, \dots, n$, can be determined such that the transformation between $x_1, \dots, x_n$ and the augmented set of variables $\widetilde{y} \doteq [y_1 \cdots y_m y_{m+1} \cdots y_n]^T$ satisfy the hypotheses of Theorem 14.2, then the pdf of $\mathbf{y}$ can be obtained by computing the marginal density

$$f_{\mathbf{y}}(y_1, \dots, y_m) = \int \cdots \int f_{\mathbf{x}}\big(g^{-1}(y_1, \dots, y_n)\big) J(x \to \widetilde{y})\, \mathrm{d}y_{m+1} \cdots \mathrm{d}y_n.$$

A.2 Jacobians of Transformations

We report here several rules for the computation of Jacobians of matrix transformations. More comprehensive results related to Jacobians are given for instance in [129, 186, 312].

Rule A.1 (Chain rule for Jacobians)

$$J(Y \to X) = J(Y \to Z) J(Z \to X).$$

Rule A.2 (Jacobian of the derivatives) *Given a matrix transformation (linear or not) $Y = F(X)$, then the transformation of the differentials, $\mathrm{d}Y = \mathrm{d}F(X)$ is linear, and*

$$J(Y \to X) = J(\mathrm{d}Y \to \mathrm{d}X).$$

Rule A.3 (Jacobian of $Y = AX$) *The Jacobian of the linear matrix transformation*

$$Y = AX$$

where $X \in \mathbb{R}^{n,m}$, $A \in \mathbb{R}^{n,n}$, is given by

$$J(Y \to X) = |A|^m.$$

Similarly, the Jacobian of the matrix transformation $Y = XB$, with $B \in \mathbb{R}^{m,m}$, is given by

$$J(Y \to X) = |B|^n.$$

Rule A.4 (Jacobian of real $Y = AXB$) *The Jacobian of the matrix transformation*

$$Y = AXB$$

where $X \in \mathbb{R}^{n,m}$, $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{m,m}$, is given by

$$J(Y \to X) = |A|^m |B|^n.$$

If A and B are orthogonal, then $J(Y \to X) = 1$.

Rule A.5 (Jacobian of complex $Y = AXB$) *The Jacobian of the matrix transformation*

$$Y = AXB$$

where $X \in \mathbb{C}^{n,m}$, $A \in \mathbb{C}^{n,n}$, $B \in \mathbb{C}^{m,m}$, *is given by*

$$J(Y \to X) = |\widetilde{A}|^{2m} |\widetilde{B}|^{2n}$$

where

$$\widetilde{A} = \begin{bmatrix} \mathrm{Re}(A) & -\mathrm{Im}(A) \\ \mathrm{Im}(A) & \mathrm{Re}(A) \end{bmatrix}, \qquad \widetilde{B} = \begin{bmatrix} \mathrm{Re}(B) & -\mathrm{Im}(B) \\ \mathrm{Im}(B) & \mathrm{Re}(B) \end{bmatrix}.$$

Proof To prove this, notice that, by Rule A.1, $J(Y \to X) = J(Y \to Z)J(Z \to X)$, where $Z = AX$. Write then the linear equation $Z = AX$ in terms of the real and imaginary parts

$$\begin{bmatrix} \mathrm{Re}(Z) \\ \mathrm{Im}(Z) \end{bmatrix} = \widetilde{A} \begin{bmatrix} \mathrm{Re}(X) \\ \mathrm{Im}(X) \end{bmatrix}$$

then, by Rule A.3, $J(Z \to X) = |\widetilde{A}|^{2m}$. Similarly, $J(Y \to Z) = |\widetilde{B}|^{2n}$. □

Notice that if A is unitary, then it can be easily seen that $\widetilde{A}$ is orthogonal. Therefore, for A, B unitary, $J(Y \to X) = 1$.

A.3 Selberg Integral

We present the solution of the so-called Selberg integral, derived in [353].

Theorem A.2 (Selberg integral) *For any positive integer* n, *let*

$$\varphi(x) = \varphi(x_1, \ldots, x_n) = \prod_{1 \le i < k \le n} (x_i - x_k)$$

if $n > 1$ *and* $\varphi(x) = 1$ *for* $n = 1$, *and*

$$\Phi(x) = \left|\varphi(x)\right|^{2\gamma} \prod_{i=1}^{n} x_i^{\alpha-1} (1 - x_i)^{\beta-1}.$$

Then

$$\int_0^1 \cdots \int_0^1 \Phi(x)\, \mathrm{d}x_1 \cdots \mathrm{d}x_n = \prod_{i=0}^{n-1} \frac{\Gamma(1 + \gamma + i\gamma)\Gamma(\alpha + i\gamma)\Gamma(\beta + i\gamma)}{\Gamma(1 + \gamma)\Gamma(\alpha + \beta + (n + i - 1)\gamma)},$$

for any $\alpha, \beta, \gamma \in \mathbb{C}$ *such that*

$$\mathrm{Re}(\alpha) > 0, \quad \mathrm{Re}(\beta) > 0, \quad \mathrm{Re}(\gamma) > -\min\left\{\frac{1}{n}, \frac{\mathrm{Re}(\alpha)}{n-1}, \frac{\mathrm{Re}(\beta)}{n-1}\right\}.$$

A.4 Dyson–Mehta Integral

The next theorem reports a result on the computation of the integral of certain determinants. The proof of this theorem can be found in [281].

Theorem A.3 (Dyson–Mehta) *Let $Z_n \in \mathbb{R}^{n,n}$ be a $n \times n$ symmetric matrix such that*:

1. $[Z_n]_{i,j} = \psi(x_i, x_j)$, *i.e.* $[Z_n]_{i,j}$ *depends only on* x_i *and* x_j;
2. $\int \psi(x, x)\, \mathrm{d}\mu(x) = c$;
3. $\int \psi(x, y)\psi(y, z)\, \mathrm{d}\mu(y) = \psi(x, z)$

where $\mathrm{d}\mu(x)$ *is a suitable measure and c is a constant. Then*

$$\int \det(Z_n)\, \mathrm{d}\mu(x_n) = (c - n + 1) \det(Z_{n-1}) \tag{A.1}$$

where Z_{n-1} *is the* $(n-1) \times (n-1)$ *matrix obtained from* Z_n *by removing the row and the column containing* x_n.

List of Symbols[1]

Vector Spaces and Cones

$\mathbb{R}^n$	space of real n-dimensional vectors
$\mathbb{C}^n$	space of complex n-dimensional vectors
$\mathbb{F}^n$	space of n-dimensional vectors with entries from $\mathbb{R}$ or $\mathbb{C}$
$\mathbb{R}^{n,m}$	space of real n-by-m matrices
$\mathbb{C}^{n,m}$	space of complex n-by-m matrices
$\mathbb{F}^{n,m}$	space of n-by-m matrices with entries from $\mathbb{R}$ or $\mathbb{C}$
$\mathbb{R}^n_+$	nonnegative orthant
$\mathbb{S}^n$	space of n-by-n real symmetric matrices
$\mathbb{S}^n_+$	cone of n-by-n positive semidefinite real symmetric matrices
$\mathbb{SK}^n$	space of n-by-n real skew-symmetric matrices
$\mathbb{H}^n$	space of n-by-n complex Hermitian matrices
$\mathbb{HK}^n$	space of n-by-n complex skew-Hermitian matrices
$\mathcal{G}^n_{\mathcal{O}}$	group of orthogonal matrices in $\mathbb{R}^{n,n}$; (17.3)
$\mathcal{G}^n_{\mathcal{U}}$	group of unitary matrices in $\mathbb{C}^{n,n}$; (17.26)
$\mathcal{H}_2^{n,m}$	$\mathcal{H}_2$ space of n-by-m transfer functions; Definition 3.3
$\mathcal{RH}_2^{n,m}$	$\mathcal{RH}_2$ space of n-by-m transfer functions; Definition 3.3
$\mathcal{H}_\infty^{n,m}$	$\mathcal{H}_\infty$ space of n-by-m transfer functions; Definition 3.2
$\mathcal{RH}_\infty^{n,m}$	$\mathcal{RH}_\infty$ space of n-by-m transfer functions; Definition 3.1
$[0, 1]^n$	n-dimensional unit cube

Basic Operations

$\lceil x \rceil$	minimum integer greater or equal to $x \in \mathbb{R}$
$\lfloor x \rfloor$	largest integer smaller or equal to $x \in \mathbb{R}$
$\log(x)$	natural logarithm (to the base e) of $x \in \mathbb{R}$
$\log_2(x)$	binary logarithm (to the base 2) of $x \in \mathbb{R}$
I_n	n-by-n identity matrix

[1] We denote vector or scalar variables with lower case letters and matrix variables with upper case. Boldface indicates random variables and matrices.

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0,

$0_{n,m}$	n-by-m zero matrix
X^T	transpose of X
X^*	Hermitian of X
X^{-1}	inverse of (nonsingular) X
$\operatorname{rank} X$	rank of matrix X
$X^\perp$	orthogonal complement of X, i.e. a matrix of maximum rank such that $X^T X^\perp = 0$, $X^{\perp T} X^\perp = I$
$\det X$	determinant of X
$\lvert X \rvert$	absolute value of the determinant of X
$[X]_{i,k}$	(i,k) entry of X
$\operatorname{Re}(x), \operatorname{Im}(x)$	real and imaginary parts of $x \in \mathbb{C}$
$\langle x, y \rangle$	inner product of vectors x and y
$\operatorname{Tr} X$	trace of X
$\operatorname{vec}(X)$	column vectorization of matrix X; (3.8)
$\operatorname{diag}(x)$	diagonal matrix formed with the entries of vector x
$\operatorname{bdiag}(X_1, \ldots, X_n)$	block diagonal matrix formed with $X_1, \ldots, X_n$
$\rho_\lambda(X)$	spectral radius of X
$[X]_+$	projection of $X \in \mathbb{S}^n$ onto the cone $\mathbb{S}^n_+$; (10.7)
$X \succ 0$	positive definite symmetric matrix
$X \succeq 0$	positive semidefinite symmetric matrix
$X \prec 0$	negative definite symmetric matrix
$X \preceq 0$	negative semidefinite symmetric matrix
$\mathbb{I}_S(\cdot)$	indicator function of the set S
$\operatorname{Vol}(S)$	volume of the set S; (3.14)
$\operatorname{Surf}(S)$	surface of the set S
$\operatorname{Card}(S)$	cardinality of the finite set S
$\Gamma(\cdot)$	Gamma function

Vector Norms and Balls

$\lVert x \rVert_p$	ℓ_p norm of the vector x; (3.1)
$\mathcal{B}_{\lVert\cdot\rVert_p}(\rho, \mathbb{F}^n)$	ball of radius ρ in the ℓ_p norm in $\mathbb{F}^n$; (3.2)
$\partial\mathcal{B}_{\lVert\cdot\rVert_p}(\rho, \mathbb{F}^n)$	boundary of $\mathcal{B}_{\lVert\cdot\rVert_p}(\rho, \mathbb{F}^n)$; (3.3)
$\lVert x \rVert_2^W$	weighted ℓ_2 norm of the vector x; (3.4)
$\mathcal{B}_{\lVert\cdot\rVert_2^W}(\rho, \mathbb{R}^n)$	ball of radius ρ in the ℓ_2^W norm in $\mathbb{F}^n$; (3.5)
$\mathcal{E}(m, W)$	ellipsoid of center m and shape matrix $W \succ 0$; (3.6)

Matrix Norms and Balls

$\lVert X \rVert_p$	ℓ_p Hilbert–Schmidt norm of the matrix X; (3.7)
$\lvert\lvert\lvert X \rvert\rvert\rvert_p$	ℓ_p-induced norm of the matrix X; (3.9)
$\mathcal{B}_{\lvert\lvert\lvert\cdot\rvert\rvert\rvert_p}(\rho, \mathbb{F}^{n,m})$	ball of radius ρ in the ℓ_p-induced norm in $\mathbb{F}^{n,m}$; (3.12)
$\mathcal{B}_\sigma(\rho, \mathbb{F}^{n,m})$	ball of radius ρ in the spectral norm in $\mathbb{F}^{n,m}$; (3.13)

Probability

$(\Omega, \mathcal{S}, \Pr\{S\})$	probability space
$f_{\mathbf{X}}(X)$	probability density function of $\mathbf{X}$
$F_{\mathbf{X}}(X)$	cumulative distribution function of $\mathbf{X}$

$\mathrm{E}_{\mathbf{X}}(J(X))$	expected value of $J(X)$ taken with respect to $\mathbf{X}$
$\mathrm{Var}(\mathbf{x})$	variance of the random variable $\mathbf{x}$
$\mathrm{Cov}(\mathbf{x})$	covariance matrix of the random vector $\mathbf{x}$
$f_{\mathbf{x}_1,\ldots,\mathbf{x}_i}$	marginal density function; (2.1)
$f_{\mathbf{x}_i \mid x_1 \ldots x_{i-1}}$	conditional density function; (2.2)
$J(x \to y)$	Jacobian of the function $x = h(y)$; (14.7)

Density Functions

$\mathrm{b}_{n,p}$	binomial density with parameters n, p; (2.3)
$\mathrm{B}_{n,p}$	binomial distribution with parameters n, p; (2.4)
$\mathcal{N}_{\bar{x},\sigma^2}$	normal density with mean $\bar{x}$ and variance σ^2; (2.5)
$\mathcal{N}_{\bar{x},W}$	multivariate normal density with mean $\bar{x}$ and covariance W; (2.6)
$\mathcal{U}_{[a,b]}$	uniform density in the interval $[a, b]$; (2.7)
$\mathcal{U}_S$	uniform density over the set S; (2.8)
$G_{a,b}$	Gamma density with parameters a, b; (2.12)
$G_{a,c}$	generalized Gamma density with parameters a, c; (2.13)

Robust Control

$\widetilde{\mathbb{D}}$	structured operator uncertainty set; (3.25)
$\mathbb{D}$	structured matrix uncertainty set; (3.27)
$\mathcal{B}_{\widetilde{\mathbb{D}}} = \mathcal{B}_{\widetilde{\mathbb{D}}}(\rho)$	ball of radius ρ in $\widetilde{\mathbb{D}}$; (3.26)
$\mathcal{B}_{\mathbb{D}} = \mathcal{B}_{\mathbb{D}}(\rho)$	ball of radius ρ in $\mathbb{D}$; (3.28)
$\mathcal{B}_q = \mathcal{B}_q(\rho)$	ball of radius ρ of parametric uncertainty; (3.45)
$\mathcal{F}_u(\cdot)$	upper linear fractional transformation; (3.29)
$\mathcal{F}_l(\cdot)$	lower linear fractional transformation; (4.3)
$\mu_{\mathbb{D}}(M)$	structured singular value of the matrix M; (3.38)
$r_{\mathbb{R}}, r_{\mathbb{C}}$	real and complex stability radii; (3.36)
$r_{\mathbb{D}}$	stability radius under structured perturbations; (3.40)
$J(\Delta)$	performance function for analysis
$J(\Delta, \theta)$	performance function for design
$\mathcal{B}_G, \mathcal{B}_B$	good and bad sets; (6.2)

Randomization and Learning

$p(\gamma)$	probability of performance; (6.6)
$\boldsymbol{\Delta}^{(1\ldots N)}$	multisample $\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(N)}$ of $\boldsymbol{\Delta}$; (7.1)
$\widehat{\mathbf{p}}_N(\gamma)$	empirical probability of performance; (7.2)
$\widehat{\mathbf{E}}_N(J(\boldsymbol{\Delta}))$	empirical mean of $J(\boldsymbol{\Delta})$; (7.5)
$\mathrm{degrad}(\rho)$	performance degradation function; (6.14)
$x^{(1\ldots N)}$	deterministic point set; (7.14)
$D_N(\mathcal{S}, x^{(1\ldots N)})$	discrepancy of $x^{(1\ldots N)}$ with respect to $\mathcal{S}$; (7.15)
$d_N(x^{(1\ldots N)})$	dispersion of $x^{(1\ldots N)}$; (7.23)
$\mathbb{S}_{\mathcal{J}}(N)$	shatter coefficient of the family $\mathcal{J}$; (9.3)
$\mathrm{VC}(\mathcal{J})$	VC dimension of the family $\mathcal{J}$; Definition 9.3
$\text{P-DIM}(\mathcal{J})$	P dimension of the family $\mathcal{J}$; Definition 9.4
$V(\theta)$	probability of violation for the design θ; (10.4)
$R(\theta)$	reliability of the design θ; (10.5)

References

1. Abate A, Prandini M, Lygeros J, Sastry S (2008) Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. Automatica 44:2724–2734
2. Abdallah C, Ariola M, Dorato P, Panchenko D (2001) Statistical-learning control of multiple-delay systems with applications to ATM networks. Kybernetica 120:355–365
3. Abdallah CT, Tanner HG (2007) Complex networked systems: introduction to the special section. IEEE Control Syst Mag 27:30–32
4. Abramowitz M, Stegun IA (eds) (1970) Handbook of mathematical functions. Dover, New York
5. Ackermann JE, Hu HZ, Kaesbauer D (1990) Robustness analysis: a case study. IEEE Trans Autom Control 35:352–356
6. Agmon S (1954) The relaxation method for linear inequalities. Can J Math 6:382–392
7. Aho AV, Hopcroft JE, Ullman JD (1974) The design and analysis of computer algorithms. Addison-Wesley, Reading
8. Ahrens JH, Dieter U (1974) Computer methods for sampling from gamma, beta, Poisson and binomial distributions. Computing 12:223–246
9. Alamo T, Tempo R, Camacho EF (2009) A randomized strategy for probabilistic solutions of uncertain feasibility and optimization problems. IEEE Trans Autom Control 54:2545–2559
10. Alamo T, Tempo R, Luque A (2010) On the sample complexity of probabilistic analysis and design methods. In: Hara S, Ohta Y, Willems JC (eds) Perspectives in mathematical system theory, control, and signal processing. Springer, Berlin, pp 39–50
11. Alamo T, Tempo R, Ramirez DR, Camacho EF (2008) A new vertex result for robustness problems with interval matrix uncertainty. Syst Control Lett 57:474–481
12. Alippi C (2002) A probably approximately correct framework to estimate performance degradation in embedded systems. IEEE Trans Comput-Aided Des Integr Circuits Syst 21:749–762
13. Alippi C (2002) Randomized algorithms: a system-level, poly-time analysis of robust computation. IEEE Trans Comput 51:740–749
14. Alippi C, Catelani M, Fort A, Mugnaini M (2002) SBT soft fault diagnosis in analog electronic circuits: a sensitivity-based approach by randomized algorithms. IEEE Trans Instrum Meas 51:1116–1125
15. Allgöwer F, Doyle F (2011) Introduction to the special issue on systems biology. Automatica 47:1095–1096
16. Alpcan T, Başar T (2005) A utility-based congestion control scheme for internet-style networks with delay. IEEE Trans Netw 13(6):1261–1274
17. Alpcan T, Başar T, Tempo R (2005) Randomized algorithms for stability and robustness analysis of high speed communication networks. IEEE Trans Neural Netw 16:1229–1241

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0,

18. Altman NS (1988) Bitwise behavior of random number generators. SIAM J Sci Stat Comput 9:941–949
19. Anantharam V, Walrand J (1990) Special issue on control methods for communication networks—editorial. Automatica 35:1891
20. Anderson BDO, Bose NK, Jury EI (1975) Output feedback stabilization and related problems—solution via decision methods. IEEE Trans Autom Control 20:53–66
21. Anderson BDO, Moore JB (1990) Optimal control: linear quadratic methods. Prentice-Hall, Englewood Cliffs
22. Anderson TW (1958) An introduction to multivariate statistical analysis. Wiley, New York
23. Anderson TW, Darling DA (1952) Asymptotic theory of certain "goodness-of-fit" criteria based on stochastic processes. Ann Math Stat 23:193–212
24. Anonymous (1997) Flying qualities of piloted aircraft. Technical Report MIL-HDBK-1797, Department of Defense, USA
25. Antsaklis PJ, Baillieul J (2004) Guest editorial—special issue on networked control systems. IEEE Trans Autom Control 49(9):1421–1422
26. Antsaklis PJ, Baillieul J (2007) Special issue on the technology of networked control systems. Proc IEEE 95:5–8
27. Applegate D, Kannan R (1991) Sampling and integration of near log-concave functions. In: Proceedings of the ACM symposium on theory of computing
28. Ataei A, Wang Q (2012) An ellipsoid algorithm for linear optimization with uncertain LMI constraints. In: Proceedings of the American control conference
29. Atkinson DS, Vaidya PM (1995) A cutting plane algorithm for convex programming that uses analytic centers. Math Program, Ser B 69:1–43
30. Au SK, Beck JL (2001) Estimation of small failure probability in high dimensions simulation. Probab Eng Mech 16:263–277
31. Avrachenkov K, Litvak N, Nemirovsky D, Osipova N (2007) Monte Carlo methods in PageRank computation: when one iteration is sufficient. SIAM J Numer Anal 45:890–904
32. Azuma S-I, Imura J-I (2007) Polynomial-time probabilistic controllability analysis of discrete-time piecewise affine systems. IEEE Trans Autom Control 52:470–482
33. Babai L (1979) Monte Carlo algorithms in graph isomorphism testing. Technical report, Départment de Mathématique et de Statistique, Université de Montréal
34. Başar T, Olsder GJ (1999) Dynamic noncooperative game theory. SIAM, Philadelphia
35. Bai E-W, Nagpal KM, Tempo R (1996) Bounded-error parameter estimation: noise models and recursive algorithms. Automatica 32:985–999
36. Bai E-W, Tempo R, Fu M (1998) Worst-case properties of the uniform distribution and randomized algorithms for robustness analysis. Math Control Signals Syst 11:183–196
37. Balakrishnan V, Boyd S, Balemi S (1992) Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. Int J Robust Nonlinear Control 1:295–317
38. Balas GJ, Doyle JC, Glover K, Packard A, Smith R (1993) μ-analysis and synthesis toolbox. MUSYN Inc and The MathWorks Inc, Natick
39. Ball K (1997) An elementary introduction to modern convex geometry. In: Levy S (ed) Flavors of geometry. Cambridge University Press, Cambridge, pp 1–58
40. Bárány I, Füredi Z (1987) Computing the volume is difficult. Discrete Comput Geom 2:319–326
41. Barmish BR (1994) New tools for robustness of linear systems. MacMillan, New York
42. Barmish BR, Kang HI (1993) A survey of extreme point results for robustness of control systems. Automatica 29:13–35
43. Barmish BR, Khargonekar PP, Shi Z, Tempo R (1990) Robustness margin need not be a continuous function of the problem data. Syst Control Lett 15:91–98
44. Barmish BR, Lagoa CM (1997) The uniform distribution: a rigorous justification for its use in robustness analysis. Math Control Signals Syst 10:203–222
45. Barmish BR, Shcherbakov PS (2002) On avoiding vertexization of robustness problems: the approximate feasibility concept. IEEE Trans Autom Control 47:819–824

46. Bartlett AC, Hollot CV, Huang L (1988) Root locations of an entire polytope of polynomials: it suffices to check the edges. Math Control Signals Syst 1:61–71
47. Beadle ER, Djurić PM (1997) Uniform random parameter generation of stable minimum-phase real ARMA (p, q) processes. IEEE Signal Process Lett 4(9):259–261
48. Bellare M, Goldwasser S, Micciancio D (1997) "Pseudo-random" number generation within cryptographic algorithms: the DSS case. Springer, New York
49. Bellman R (1957) Dynamic programming. Princeton University Press, Princeton
50. Ben-Tal A, Nemirovski A (1998) Robust convex optimization. Math Oper Res 23:769–805
51. Ben-Tal A, Nemirovski A (2009) On safe tractable approximations of chance constrained linear matrix inequalities. Math Oper Res 34(1):1–25
52. Bennett G (1962) Probability inequalities for the sum of independent random variables. J Am Stat Assoc 57:33–45
53. Bentley JL (1993) Engineering a sort function. Softw Pract Exp 23:1249–1265
54. Bernoulli J (1713) Ars conjectandi, Paris
55. Bernstein SN (1946) The theory of probabilities. Gostehizdat Publishing House, Moscow (in Russian)
56. Bertsimas D, Sethuraman J (2000) Moment problems and semidefinite optimization. In: Wolkowicz H, Saigal R, Vandenberghe L (eds) Handbook of semidefinite programming. Kluwer Academic Publishers, Boston, pp 469–509
57. Bertsimas D, Vempala S (2004) Solving convex programs by random walks. J ACM 51:540–556
58. Bhattacharyya SP, Chapellat H, Keel LH (1995) Robust control: the parametric approach. Prentice-Hall, Upper Saddle River
59. Blanchini F (1999) Set invariance in control. Automatica 35:1747–1767
60. Blanchini F, Miani S (2008) Set-theoretic methods in control. Birkhäuser, Boston
61. Blondel VD, Tsitsiklis JN (2000) A survey of computational complexity results in systems and control. Automatica 36:1249–1274
62. Blum L, Blum M, Shub M (1986) A simple unpredictable pseudo-random number generator. SIAM J Comput 15:364–383
63. Blum L, Cucker F, Shub M, Smale S (1997) Complexity and real computation. Springer, New York
64. Bollobás B (1997) Volume estimates and rapid mixing. In: Levy S (ed) Flavors of geometry. Cambridge University Press, Cambridge, pp 151–194
65. Bondarko VA, Yakubovich VA (1992) The method of recursive aim inequalities in adaptive control theory. Int J Adapt Control Signal Process 6:141–160
66. Boucheron S, Lugosi G, Massart P (2003) Concentration inequalities using the entropy method. Ann Probab 31:1583–1614
67. Boyd S, Barratt CH (1991) Linear controller design—limits of performance. Prentice-Hall, Englewood Cliffs
68. Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. SIAM, Philadelphia
69. Boyd S, Ghosh A, Prabhakar B, Shah D (2006) Randomized gossip algorithms. IEEE Trans Inf Theory 52:2508–2530
70. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge
71. Braatz RP, Young PM, Doyle JC, Morari M (1994) Computational complexity of μ calculation. IEEE Trans Autom Control 39:1000–1002
72. Brin S, Page L (1998) The anatomy of a large-scale hypertextual Web search engine. Comput Netw ISDN Syst 30:107–117
73. Brockett RW, Liberzon D (2000) Quantized feedback stabilization of linear systems. IEEE Trans Autom Control 45:1279–1289
74. Brooks SH (1958) A discussion of random methods for seeking maxima. Oper Res 6:244–251
75. Brown WG, Viola L (2010) Convergence rates for arbitrary statistical moments of random quantum circuits. Phys Rev Lett 104:250501/1–4

76. Bushnell LG (2001) Special issue on networks and control—editorial. IEEE Control Syst Mag 21:22–23
77. Cai K, Ishii H (2011) Quantized consensus and averaging on gossip digraphs. IEEE Trans Autom Control 56:2087–2100
78. Calafiore G (2010) Random convex programs. SIAM J Optim 20(6):3427–3464
79. Calafiore G, Campi MC (2006) The scenario approach to robust control design. IEEE Trans Autom Control 51(5):742–753
80. Calafiore G, Dabbene F (2001) Loop gain under random feedback. In: Proceedings of the IEEE conference on decision and control
81. Calafiore G, Dabbene F (2002) A probabilistic framework for problems with real structured uncertainty in systems and control. Automatica 38:1265–1276
82. Calafiore G, Dabbene F, Tempo R (1999) Radial and uniform distributions in vector and matrix spaces for probabilistic robustness. In: Miller DE, Qiu L (eds) Topics in control and its applications. Springer, New York, pp 17–31
83. Calafiore G, Dabbene F, Tempo R (2000) Randomized algorithms for probabilistic robustness with real and complex structured uncertainty. IEEE Trans Autom Control 45:2218–2235
84. Calafiore G, Dabbene F, Tempo R (2007) A survey of randomized algorithms for control synthesis and performance verification. J Complex 23(3):301–316
85. Calafiore G, Dabbene F, Tempo R (2011) Research on probabilistic methods for control system design. Automatica 47:1279–1293
86. Calafiore G, Polyak BT (2001) Stochastic algorithms for exact and approximate feasibility of robust LMIs. IEEE Trans Autom Control 46:1755–1759
87. Calafiore G, Campi MC (2005) Uncertain convex programs: randomized solutions and confidence levels. Math Program 102(1):25–46
88. Calafiore G, Dabbene F (2007) A probabilistic analytic center cutting plane method for feasibility of uncertain LMIs. Automatica 43:2022–2033
89. Calafiore G, Dabbene F (2008) Optimization under uncertainty with applications to design of truss structures. Struct Multidiscip Optim 35(3):189–200
90. Calafiore G, Dabbene F (2008) A reduced vertex set result for interval semidefinite optimization problems. J Optim Theory Appl 139:17–33
91. Calafiore G, Fagiano L (2012) Robust model predictive control via scenario optimization. IEEE Trans Autom Control, in press. doi:10.1109/TAC.2012.2203054
92. Camacho EF, Bordons C (2003) Model predictive control. Springer, London
93. Cambanis S, Huang S, Simons G (1981) On the theory of elliptically contoured distributions. J Multivar Anal 11:368–385
94. Campi MC, Calafiore G, Garatti S (1990) Interval predictor models: identification and reliability. Automatica 45(2):382–392
95. Campi MC, Garatti S (2008) The exact feasibility of randomized solutions of robust convex programs. SIAM J Optim 19:1211–1230
96. Campi MC, Garatti S (2011) A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. J Optim Theory Appl 148(2):257–280
97. Chamanbaz M, Keikha E, Venkataramanan V, Al Mamun A, Wang Q-G, Liew T (2011) $\mathcal{H}_\infty$ probabilistic robust control of hard disk drive. In: Proceedings 37th annual conference of IEEE industrial electronics society, pp 3394–3399.
98. Chamanbaz M, Keikha E, Venkataramanan V, Wang Q-G, Al Mamun A (2012) Probabilistic robust approach for discrete multi-objective control of track-following servo systems in hard disk drives. In: Proceedings 7th IFAC symposium on robust control design
99. Chamanbaz M, Venkataramanan V, Wang Q-G (2012) Robust $\mathcal{H}_2$ track following controller based on probabilistic analytic center cutting plane method. In: Proceedings joint international conference on micromechatronics for information and precision equipment
100. Chebychev P (1874) Sur les valeurs limites des intégrales. J Math Pures Appl 19:157–160
101. Chen C-T (1999) Linear systems theory and design. Oxford University Press, New York

102. Chen H-F (2002) Stochastic approximation and its application, vol 64. Kluwer Academic Publishers, Dordrecht, pp 175–182
103. Chen X, Zhou K (1998) Order statistics and probabilistic robust control. Syst Control Lett 35
104. Chernoff H (1952) A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. Ann Math Stat 23:493–507
105. Chesi G, Garulli A, Tesi A, Vicino A (2003) Solving quadratic distance problems: an LMI-based approach. IEEE Trans Autom Control 48:200–212
106. Chiang RY, Safonov MG (1996) The robust control toolbox. The MathWorks Inc, Natick
107. Ching J, Au SK, Beck JL (2005) Reliability estimation for dynamical systems subject to stochastic excitation using subset simulation with splitting. Comput Methods Appl Mech Eng 194:1557–1579
108. Chung KL (2001) A course on probability theory. Academic Press, London
109. Cogburn R (1986) On products of random stochastic matrices. Contemp Math 50:199–213
110. Colaneri P, Locatelli A, Geromel JC (1997) Control theory and design. A $\mathcal{RH}_2$-$\mathcal{RH}_\infty$ viewpoint. Academic Press, San Diego
111. Conover WJ (1980) Practical nonparametric statistics. Wiley, New York
112. Cook SA (1971) The complexity of theorem proving procedures. In: Proceedings of the ACM symposium on theory of computing, pp 117–128
113. Couture R, L'Ecuyer P (1998) Special issue on uniform random number generation—editorial. ACM Trans Model Comput Simul 8:1–2
114. Cover TM (1965) Geometrical and statistical properties of system of linear inequalities with applications in pattern recognition. IEEE Trans Electron Comput 14:326–334
115. Coxson GE, De Marco CL (1994) The computational complexity of approximating the minimal perturbation scaling to achieve instability in an interval matrix. Math Control Signals Syst 7:279–291
116. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines. Cambridge University Press, Cambridge
117. Dabbene F, Gay P, Polyak BT (2003) Recursive algorithms for inner ellipsoidal approximation of convex polytopes. Automatica 39(10):1773–1781
118. Dabbene F, Polyak BT, Tempo R (2007) On the complete instability of interval polynomials. Syst Control Lett 56(6):431–4381
119. Dabbene F, Shcherbakov PS, Polyak BT (2010) A randomized cutting plane method with probabilistic geometric convergence. SIAM J Optim 20:3185–3207
120. Dabbene F, Tempo R (2010) Probabilistic and randomized tools for control design. In: Levine WS (ed) The control handbook, 2nd edn. Control system advanced methods. CRC Press, Boca Raton, pp 65.1–65.23
121. Dahleh MA, Diaz-Bobillo IJ (1995) Control of linear systems: a linear programming approach. Prentice-Hall, Englewood Cliffs
122. Dantzig GB, Infanger G (1993) Multi-stage stochastic linear programs for portfolio optimization. Ann Oper Res 45:59–76
123. Davis M (1973) Hilbert's tenth problem is unsolvable. Math Mon 80:233–269
124. Davis PJ, Rabinowitz P (1984) Methods of numerical integration. Academic Press, New York
125. Davis WR (1996) Micro UAV. In: Proceedings 23rd annual AUVSI symposium
126. de Bruijn NG (1955) On some multiple integrals involving determinants. J Indian Math Soc 19:133–151
127. De Gaston R, Safonov MG (1988) Exact calculation of the multiloop stability margin. IEEE Trans Autom Control 33:156–171
128. de Kerchove C, Ninove L, Van Dooren P (2008) Influence of the outlinks of a page on its PageRank. Linear Algebra Appl 429(5–6):1254–1276
129. Deemer WL, Olkin I (1951) The Jacobians of certain matrix transformations useful in multivariate analysis. Biometrika 38:345–367

130. Dembo A, Zeitouni O (1993) Large deviations techniques and applications. Jones and Bartlett, Boston
131. Demmel JW (1992) The component-wise distance to the nearest singular matrix. SIAM J Matrix Anal Appl 13:10–19
132. Devroye L, Györfi L, Lugosi G (1996) A probabilistic theory of pattern recognition. Springer, New York
133. Devroye LP (1986) Non-uniform random variate generation. Springer, New York
134. Devroye LP (1997) Random variate generation for multivariate unimodal densities. ACM Trans Model Comput Simul 7:447–477
135. Diaconis P, Hanlon P (1992) Eigen-analysis for some examples of the Metropolis algorithm. Contemp Math 138:99–117
136. Djaferis TE (1995) Robust control design: a polynomial approach. Kluwer Academic Publishers, Boston
137. Djavdan P, Tulleken HJAF, Voetter MH, Verbruggen HB, Olsder GJ (1989) Probabilistic robust controller design. In: Proceedings of the IEEE conference on decision and control
138. Doob JL (1990) Stochastic processes. Wiley, New York
139. Dorato P, Famularo D, Abdallah CT, Yang W (1999) Robust nonlinear feedback design via quantifier elimination theory. Int J Robust Nonlinear Control 9:817–822
140. Dorato P, Kun L, Kosmatopoulos EB, Ioannou PA, Ryaciotaki-Boussalis H (2000) Quantified multivariate polynomial inequalities. The mathematics of practical control design problems. IEEE Control Syst Mag 20:48–58
141. Dorato P, Tempo R, Muscato G (1993) Bibliograpy on robust control. Automatica 29:201–213
142. Doyle J (1978) Guaranteed margins for LQG regulators. IEEE Trans Autom Control AC-23:756–757
143. Doyle J (1982) Analysis of feedback systems with structured uncertainties. IEE Proc 129(D):242–250
144. Doyle JC, Glover K, Khargonekar PP, Francis BA (1989) State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. IEEE Trans Autom Control 34:831–847
145. Dudley RM (1978) Central limit theorems for empirical measures. Ann Probab 6:899–929
146. Dudley RM (1979) Balls in $\mathbb{R}^k$ do not cut all subsets of $k+2$ points. Adv Math 31:306–308
147. Dudley RM (1984) A course on empirical processes. Springer, New York
148. Dudley RM (1999) Uniform central limit theorems. Cambridge University Press, Cambridge
149. Dullerud GE, Paganini F (2000) A course in robust control theory: a convex approach. Springer, New York
150. Durbin J (1960) The fitting of time series models. Revue Inst Int Stat 28:233–243
151. Durham JW, Carli R, Frasca P, Bullo F (2012) Discrete partitioning and coverage control for gossiping robots. IEEE Trans Robot 28(2):364–378
152. Dyer ME, Frieze AM, Kannan R (1991) A random polynomial-time algorithm for approximating the volume of convex bodies. J ACM 38:1–17
153. Edelman A (1989) Eigenvalues and condition numbers of random matrices. PhD dissertation, Massachusetts Institute of Technology, Cambridge
154. Edelman A, Kostlan E, Shub M (1994) How many eigenvalues of a random matrix are real? J Am Math Soc 7:247–267
155. Efron B, Stein C (1981) The jackknife estimate of variance. Ann Stat 9:586–596
156. El Ghaoui L, Nisulescu S-I (eds) (2000) Advances in linear matrix inequality methods in control. SIAM, New York
157. El Ghaoui L, Oustry F, AitRami M (1997) A cone complementary linearization algorithm for static output-feedback and related problems. IEEE Trans Autom Control 42:1171–1176
158. El Ghaoui L, Oustry F, Lebret H (1998) Robust solutions to uncertain semidefinite programs. SIAM J Optim 9:33–52
159. Elia N, Mitter SK (2001) Stabilization of linear systems with limited information. IEEE Trans Autom Control 46:1384–1400

160. Fam A, Meditch J (1978) A canonical parameter space for linear systems design. IEEE Trans Autom Control 23(3):454–458
161. Fam AT (1989) The volume of the coefficient space stability domain of monic polynomials. In: Proceedings of the international symposium on circuits and systems, pp 1780–1783
162. Fang KT, Kotz S, Ng KW (1990) Symmetric multivariate and related distributions. Chapman & Hall, New York
163. Faure H (1982) Discrépance de suites associées à un système de numération (en dimension s). Acta Arith 41:337–351
164. Fercoq O, Akian M, Bouhtou M, Gaubert S (2012) Ergodic control and polyhedral approaches to PageRank optimization. IEEE Trans Autom Control 57, provisionally accepted
165. Fomin VN (1976) Mathematical theory of learning recognizing systems. LGU, Leningrad (in Russian)
166. Frieze A, Hastad J, Kannan R, Lagarias JC, Shamir A (1988) Reconstructing truncated linear variables satisfying linear congruences. SIAM J Comput 17:262–280
167. Fujimoto RM (2000) Parallel and distributed simulation systems. Wiley Interscience, New York
168. Fujisaki Y, Dabbene F, Tempo R (2003) Probabilistic robust design of LPV control systems. Automatica 39:1323–1337
169. Fujisaki Y, Kozawa Y (2006) Probabilistic robust controller design: probable near minmax value and randomized algorithms. In: Calafiore G, Dabbene F (eds) Probabilistic and randomized methods for design under uncertainty. Springer, London, pp 317–329
170. Fujisaki Y, Oishi Y (2007) Guaranteed cost regulator design: a probabilistic solution and a randomized algorithm. Automatica 43:317–324
171. Fujisaki Y, Oishi Y, Tempo R (2008) Mixed deterministic/randomized methods for fixed order controller design. IEEE Trans Autom Control 53(9):2033–2047
172. Gahinet P (1996) Explicit controller formulas for LMI-based $\mathcal{H}_\infty$ synthesis. Automatica 32:1007–1014
173. Gahinet P, Apkarian P (1994) A linear matrix inequality approach to $\mathcal{H}_\infty$ control. Int J Robust Nonlinear Control 4:421–448
174. Galdos G, Karimi A, Longchamp R (2010) $\mathcal{H}_\infty$ controller design for spectral MIMO models by convex optimization. J Process Control 20:1175–1182
175. Gantmacher FR (1959) The theory of matrices. American Mathematical Society, Providence
176. Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, New York
177. Gentle JE (1998) Random number generation and Monte Carlo methods. Springer, New York
178. Gevers M, Bombois X, Codrons B, Scorletti G, Anderson BDO (2003) Model validation for control and controller validation in a prediction error identification framework—Part I: theory. Automatica 39(3):403–415
179. Gietelink OJ, De Schutter B, Verhaegen M (2005) Probabilistic approach for validation of advanced driver assistance systems. Transp Res Rec 1910:20–28
180. Girko VL (1990) Theory of random determinants. Kluwer Academic Publishers, Dordrecht
181. Goffin J-L, Vial J-P (2002) Convex non-differentiable optimization: a survey focused on the analytic center cutting plane method. Optim Methods Softw 17:805–867
182. Goh K-C, Safonov MG, Ly JH (1996) Robust synthesis via bilinear matrix inequalities. Int J Robust Nonlinear Control 6:1079–1095
183. Gong W, Başar T (2002) Special issue on systems and control methods for communication networks—editorial. IEEE Trans Autom Control 47:877–879
184. Green M, Limebeer DJN (1995) Linear robust control. Prentice-Hall, Englewood Cliffs
185. Guglieri G, Pralio B, Quagliotti F (2006) Flight control system design for a micro aerial vehicle. Aircr Eng Aerosp Technol 78:87–97
186. Gupta AK, Nagar DK (1999) Matrix variate distributions. CRC Press, Boca Raton
187. Gupta AK, Song D (1997) Characterization of p-generalized normality. J Multivar Anal 60:61–71

188. Haber A, Fraanje R, Verhaegen M (2012) Linear computational complexity robust ILC for lifted systems. Automatica 48(6):1102–1110
189. Halmos PR (1950) Measure theory. Springer, New York
190. Halton JH (1960) On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer Math 2:84–90. Berichtigung, ibid., 2:196, 1960
191. Hansen LP, Sargent TJ (2008) Robustness. Princeton University Press, Princeton
192. Hastings WK (1970) Monte Carlo sampling methods using Markov Chains and their applications. Biometrika 57:97–109
193. Hatanaka T, Takaba K (2008) Computations of probabilistic output admissible set for uncertain constrained systems. Automatica 44(2):479–487
194. Hatanaka T, Takaba K (2008) Probabilistic output admissible set for systems with time-varying uncertainties. Syst Control Lett 57(4):315–321
195. Hatano Y, Mesbahi M (2005) Agreement over random networks. IEEE Trans Autom Control 50:1867–1872
196. Haussler D (1992) Decision theoretic generalizations of the PAC model for neural net and other learning applications. Inf Comput 100:78–150
197. Hellekalek P (1998) Good random number generators are (not so) easy to find. Math Comput Simul 46:487–507
198. Hellekalek P, Larcher G (eds) (1998) Random and quasi-random point sets. Springer, New York
199. Hernandez R, Dormido S (1996) Kharitonov's theorem extension to interval polynomials which can drop in degree: a Nyquist approach. IEEE Trans Autom Control 41:1009–1012
200. Hicks JS, Wheeling RF (1959) An efficient method for generating uniformly distributed points on the surface of an n-dimensional sphere. Commun ACM 2:17–19
201. Hilbert M, López P (2011) The world's technological capacity to store, communicate, and compute information. Science 332:60–65
202. Hinrichsen D, Pritchard AJ (1986) Stability radii of linear systems. Syst Control Lett 7:1–10
203. Hlawka E (1954) Funktionen von beschränkter variation in der Theorie der Gleichverteilung. Ann Mat Pura Appl 61:325–333
204. Hoare CAR (1962) Quicksort. Comput J 5:10–15
205. Hoeffding W (1963) Probability inequalities for sums of bounded random variables. J Am Stat Assoc 58:13–30
206. Horisberger HP, Belanger PR (1976) Regulators for linear time invariant plants with uncertain parameters. IEEE Trans Autom Control 21:705–708
207. Horn RA, Johnson CR (1991) Topics in matrix analysis. Cambridge University Press, Cambridge
208. Horowitz I (1991) Survey of quantitative feedback theory (QFT). Int J Control 53:255–291
209. Houpis CH, Rasmussen SJ (1999) Quantitative feedback theory. Marcel Dekker, New York
210. Hua LK (1979) Harmonic analysis of functions of several complex variables in the classical domains. American Mathematical Society, Providence
211. Ishii H, Başar T, Tempo R (2004) Randomized algorithms for quadratic stability of quantized sampled-data systems. Automatica 40:839–846
212. Ishii H, Basar T, Tempo R (2005) Randomized algorithms for synthesis of switching rules for multimodal systems. IEEE Trans Autom Control 50:754–767
213. Ishii H, Francis BA (2002) Limited data rate in control systems with networks. Springer, New York
214. Ishii H, Tempo R (2009) Probabilistic sorting and stabilization of switched systems. Automatica 45:776–782
215. Ishii H, Tempo R (2010) Distributed randomized algorithms for the PageRank computation. IEEE Trans Autom Control 55:1987–2002
216. Ishii H, Tempo R, Bai E-W (2012, in press) A web aggregation approach for distributed randomized PageRank algorithms. IEEE Transactions on Automatic Control 57
217. Iwasaki T, Skelton RE (1994) All controllers for the general $\mathcal{H}_\infty$ control problem: LMI existence conditions and state-space formulas. Automatica 30:1307–1317

218. Jerrum M, Sinclair A (1996) The Markov Chain Monte Carlo method: an approach to approximate counting and integration. In: Hochbaum DS (ed) Approximation algorithms for NP-hard problems. PWS Publishing, Boston, pp 482–520
219. Jönsson U, Rantzer A (2000) Optimization of integral quadratic constraints. In: Ghaoui LE, Niculescu S-I (eds) Advances in linear matrix inequality methods in control. SIAM, New York, pp 109–127
220. Kaczmarz S (1937) Angenäherte aufslösung von systemen linearer gleichunger. Bull Int Acad Pol Sci Lett A 355–357 (English translation: Approximate solution of systems of linear equations. Int J Control 57:1269–1271, 1993)
221. Kalai AT, Vempala S (2006) Simulated annealing for convex optimization. Math Oper Res 31(2):253–266
222. Kale AA, Tits AL (2000) On Kharitonov's theorem without invariant degree assumption. Automatica 36:1075–1076
223. Kamvar S, Haveliwala T, Golub G (2004) Adaptive methods for the computation of PageRank. Linear Algebra Appl 386:51–65
224. Kanev S, De Schutter B, Verhaegen M (2003) An ellipsoid algorithm for probabilistic robust controller design. Syst Control Lett 49:365–375
225. Kanev S, Verhaegen M (2006) Robustly asymptotically stable finite-horizon MPC. Automatica 42(12):2189–2194
226. Kannan R, Lovász L, Simonovits M (1997) Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. Random Struct Algorithms 11:1–50
227. Karpinski M, Macintyre A (1997) Polynomial bounds for VC dimension of sigmoidal and general Pfaffian neural networks. J Comput Syst Sci 54:169–176
228. Keel LH, Bhattacharyya SP (1997) A linear programming approach to controller design. In: Proceedings of the IEEE conference on decision and control
229. Kelly FP, Maulloo AK, Tan DKH (1998) Rate control in communication networks: shadow prices, proportional fairness and stability. J Oper Res Soc 49:237–252
230. Kettani H, Barmish BR (2008) A new Monte Carlo circuit simulation paradigm with specific results for resistive networks. IEEE Trans Circuits Syst I 53:1289–1299
231. Khachiyan LG (1989) The problem of computing the volume of polytopes is NP-hard. Usp Mat Nauk 44:179–180 (in Russian)
232. Khammash M, Tomlin CJ, Vidyasagar M (2008) Guest editorial—special issue on systems biology. IEEE Trans Autom Control and IEEE Trans Circuits Syst I: Regular Papers 4–7
233. Khargonekar P, Tikku A (1996) Randomized algorithms for robust control analysis and synthesis have polynomial complexity. In: Proceedings of the IEEE conference on decision and control
234. Khargonekar PP, Petersen IR, Zhou K (1990) Robust stabilization of uncertain linear systems: quadratic stabilizability and $\mathcal{H}_\infty$ control theory. IEEE Trans Autom Control 35:356–361
235. Kharitonov VL (1978) Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. Differ Uravn 14:2086–2088 (in Russian)
236. Kimura H (1997) Chain scattering approach to $\mathcal{H}_\infty$ control. Birkhäuser, Boston
237. Knuth DE (1998) The art of computer programming. Sorting and searching, vol 3. Addison-Wesley, Reading
238. Knuth DE (1998) The art of computer programming. Seminumerical algorithms, vol 2. Addison-Wesley, Reading
239. Koksma JF (1942–1943) Een algemeene stelling uit de theorie der gelijkmatige verdeeling modulo 1. Math B (Zutphen) 11:7–11
240. Koltchinskii V, Abdallah CT, Ariola M, Dorato P (2001) Statistical learning control of uncertain systems: theory and algorithms. Appl Comput Math 120:31–43
241. Koltchinskii V, Abdallah CT, Ariola M, Dorato P, Panchenko D (2000) Improved sample complexity estimates for statistical learning control of uncertain systems. IEEE Trans Autom Control 46:2383–2388

242. Komurov K, White MA, Ram PT (2010) Use of data-biased random walks on graphs for the retrieval of context-specific networks from genomic data. PLoS Comput Biol 6(8):1–10
243. Kothare MV, Balakrishnan V, Morari M (1996) Robust constrained model predictive control using linear matrix inequalities. Automatica 32:1361–1379
244. Kushner HJ, Yin GG (2003) Stochastic approximation and recursive algorithms and applications. Springer, New York
245. Kwakernaak H (1993) Special issue on robust control—editorial. Automatica 29:3
246. Kwakernaak H, Sivan R (1972) Linear optimal control systems. Wiley, New York
247. Kwakernaak H, Sivan R (1991) Modern signals and systems. Prentice-Hall, Englewood Cliffs
248. Lagoa CM (2003) Probabilistic enhancement of classical robustness margins: a class of nonsymmetric distributions. IEEE Trans Autom Control 48(11):1990–1994
249. Lagoa CM, Barmish BR (2002) Distributionally robust Monte Carlo simulation: a tutorial survey. In: Proceedings of the IFAC world congress, pp 1327–1338
250. Lagoa CM, Dabbene F, Tempo R (2008) Hard bounds on the probability of performance with application to circuit analysis. IEEE Trans Circuits Syst I 55:3178–3187
251. Lagoa CM, Shcherbakov PS, Barmish BR (1998) Probabilistic enhancement of classical robustness margins: the unirectangularity concept. Syst Control Lett 35:31–43
252. Langville AN, Meyer CD (2006) Google's PageRank and beyond: the science of search engine rankings. Princeton University Press, Princeton
253. Lanzon A, Anderson BDO, Bombois X (2004) Selection of a single uniquely specifiable $\mathcal{H}_\infty$ controller in the chain-scattering framework. Automatica 40:985–994
254. Lasserre JB (2001) Global optimization with polynomials and the problem of moments. SIAM J Optim 11(3):796–817
255. Laurent M (2009) Sums of squares, moment matrices and optimization over polynomials. In: Emerging applications of algebraic geometry. IMA vol math appl, vol 149. Springer, New York, pp 157–270
256. LaValle SM (2006) Planning algorithms. Cambridge University Press, Cambridge. Available at http://planning.cs.uiuc.edu/
257. Lecchini-Visintini A, Glover W, Lygeros J, Maciejowski JM (2006) Monte Carlo optimization for conflict resolution in air traffic control. IEEE Trans Intell Transp Syst 7:470–482
258. Lecchini-Visintini A, Lygeros A, Maciejowski J (2010) Stochastic optimization on continuous domains with finite-time guarantees by Markov chain Monte Carlo methods. IEEE Trans Autom Control 55:2858–2863
259. L'Ecuyer P (1994) Uniform random number generation. Ann Oper Res 53:77–120
260. L'Ecuyer P, Blouin F, Couture R (1993) A search for good multiple recursive random number generators. ACM Trans Model Comput Simul 3:87–98
261. Lehmer DH (1951) Mathematical methods in large-scale computing units. In: Proceedings of the second symposium on large-scale digital calculation machinery
262. Levinson N (1947) The Wiener RMS error criterion in filter design and prediction. J Math Phys 25:261–278
263. Liberzon D, Tempo R (2004) Common Lyapunov functions and gradient algorithms. IEEE Trans Autom Control 49:990–994
264. Liu W, Chen J (2010) Probabilistic estimates for mixed model validation problems with $\mathcal{H}_\infty$ type uncertainties. IEEE Trans Autom Control 55(6):1488–1494
265. Liu W, Chen J, El-Sherief H (2007) Probabilistic bounds for uncertainty model validation. Automatica 43(6):1064–1071
266. Lorefice L, Pralio B, Tempo R (2009) Randomization-based control design for mini-UAVs. Control Eng Pract 17:974–983
267. Lovász L (1996) Random walks on graphs: a survey. In: Sós VT, Miklós D, Szönyi T (eds) Combinatorics, Paul Erdös is eighty. János Bolyai Mathematical Society, Budapest, pp 353–398
268. Lovász L (1999) Hit-and-run mixes fast. Math Program 86:443–461

269. Lu B, Wu F (2006) Probabilistic robust linear parameter-varying control of an F-16 aircraft. J Guid Control Dyn 29(6):1454–1460
270. Lugosi G (2002) Pattern classification and learning theory. In: Györfi L (ed) Principles of nonparametric learning. Springer, New York, pp 1–56
271. Ma W, Sznaier M, Lagoa CM (2007) A risk adjusted approach to robust simultaneous fault detection and isolation. Automatica 43(3):499–504
272. Macintyre AJ, Sontag ED (1993) Finiteness results for sigmoidal "neural" networks. In: Proceedings of the ACM symposium on theory of computing, pp 325–334
273. Mansour M (2010) Discrete-time and sampled-data stability tests. In: Levine WS (ed) The control handbook, 2nd edn, Control system fundamentals. CRC Press, Boca Raton, pp 8.28–8.39
274. Markov A (1884) On certain applications of algebraic continued fractions. PhD Dissertation, St. Petersburg (in Russian)
275. Marrison CI, Stengel RF (1995) Stochastic robustness synthesis applied to a benchmark control problem. Int J Robust Nonlinear Control 5(1):13–31
276. Marshall A, Olkin I (1960) Multivariate Chebyshev inequalities. Ann Math Stat 31:1001–1014
277. Masubuchi I, Ohara A, Suda N (1998) LMI-based controller synthesis: a unified formulation and solution. Int J Robust Nonlinear Control 8:669–686
278. Matiyasevich Y (1970) Enumerable sets are diophantine. Dokl Akad Nauk SSSR 191:279–282 (in Russian)
279. Matsumoto M, Nishimura T (1998) Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans Model Comput Simul 8:3–30
280. Megretski A, Ranzer A (1997) System analysis via integral quadratic constraints. IEEE Trans Autom Control 42:819–830
281. Mehta ML (1991) Random matrices. Academic Press, Boston
282. Mengersen KL, Tweedie RL (1996) Rates of convergence of the Hastings and Metropolis algorithms. Ann Stat 24:101–121
283. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller A, Teller H (1953) Equations of state calculations by fast computing machines. J Chem Phys 21:1087–1091
284. Metropolis N, Ulam SM (1949) The Monte Carlo method. J Am Stat Assoc 44:335–341
285. Meyn SP, Tweedie RL (1996) Markov chains and stochastic stability. Springer, New York
286. Minnichelli RJ, Anagnost JJ, Desoer CA (1989) An elementary proof of Kharitonov's stability theorem with extensions. IEEE Trans Autom Control 34:995–998
287. Mitchell JE (2003) Polynomial interior point cutting plane methods. Optim Methods Softw 18:507–534
288. Mitzenmacher M, Upfal E (2005) Probability and computing: randomized algorithms and probabilistic analysis. Cambridge University Press, Cambridge
289. Mohseni M, Rezakhani AT, Lidar DA (2008) Quantum-process tomography: resource analysis of different strategies. Phys Rev A 77:032322/1–15
290. Motwani R, Raghavan P (1995) Randomized algorithms. Cambridge University Press, Cambridge
291. Motzkin TS, Schoenberg IJ (1954) The relaxation method for linear inequalities. Can J Math 6:393–404
292. Muller ME (1959) A note on a method for generating random points uniformly distributed on n-dimensional spheres. Commun ACM 2:19–20
293. Mulmuley K (1994) Computational geometry: an introduction through randomization algorithms. Prentice-Hall, Englewood Cliffs
294. Nazin A, Polyak BT (2011) Randomized algorithm to determine the eigenvector of a stochastic matrix with application to the PageRank problem. Autom Remote Control 72(2):342–352
295. Nemirovski A (1993) Several NP-hard problems arising in robust stability analysis. Math Control Signals Syst 6:99–105

296. Nemirovski A, Polyak BT (1994) Necessary conditions for the stability of polynomials and their use. Autom Remote Control 55(11):1644–1649
297. Nemirovski AS, Yudin DB (1983) Problem complexity and method efficiency in optimization. Wiley, New York
298. Nesterov Y (1995) Complexity estimates of some cutting plane methods based on the analytic barrier. Math Program, Ser B 69:149–176
299. Nesterov Y, Nemirovski AS (1994) Interior point polynomial algorithms in convex programming. SIAM, Philadelphia
300. Nesterov Y, Vial J-P (2008) Confidence level solutions for stochastic programming. Automatica 44(6):1559–1568
301. Newlin MP, Young PM (1997) Mixed μ problems and branch and bound techniques. Int J Robust Nonlinear Control 7:145–164
302. Niederreiter H (1987) Point sets and sequences with small discrepancy. Monatshefte Math 104:273–337
303. Niederreiter H (1992) Random number generation and quasi-Monte Carlo methods. SIAM, Philadelphia
304. Niederreiter H (1995) New developments in uniform pseudorandom number and vector generation. In: Niederreiter H, Shiue PJ-S (eds) Monte Carlo and quasi-Monte Carlo methods in scientific computing. Springer, New York, pp 87–120
305. Niederreiter H (2003) Some current issues in quasi-Monte Carlo methods. J Complex 23:428–433
306. Ninness BM, Goodwin GC (1995) Rapprochement between bounded-error and stochastic estimation theory. Int J Adapt Control Signal Process 9:107–132
307. Notarstefano G, Bullo F (2011) Distributed abstract optimization via constraints consensus: theory and applications. IEEE Trans Autom Control 56(10):2247–2261
308. Nurges Ü (2006) Robust pole assignment via reflection coefficients of polynomials. Automatica 42:1223–1230
309. Nurges Ü (2009) Reflection coefficients of polynomials and stable polytopes. IEEE Trans Autom Control 54(6):1314–1318
310. Oishi Y (2007) Polynomial-time algorithms for probabilistic solutions of parameter-dependent linear matrix inequalities. Automatica 43(3):538–545
311. Oishi Y, Kimura H (2004) Probabilistic model-set identification not assuming plant linearity. Int J Robust Nonlinear Control 14(11):971–981
312. Olkin I (1953) Note on the Jacobians of certain matrix transformations useful in multivariate analysis. Biometrika 40:43–46
313. Pace IS, Barnett S (1973) Numerical comparison of root-location algorithms for constant linear systems. In: Bell DJ (ed) Recent mathematical developments in control. Academic Press, London, pp 373–392
314. Packard A, Doyle J (1993) The complex structured singular value. Automatica 29:71–109
315. Paganini F, Feron E (2000) Linear matrix inequality methods for robust $\mathcal{H}_2$ analysis: a survey with comparisons. In: Ghaoui LE, Niculescu S-I (eds) Advances in linear matrix inequality methods in control. SIAM, New York, pp 129–151
316. Pallottino L, Scordio VG, Bicchi A, Frazzoli E (2007) Decentralized cooperative policy for conflict resolution in multi-vehicle systems. IEEE Trans Robot 23:1170–1183
317. Palopoli L, Pinello C, Bicchi A, Sangiovanni-Vincentelli A (2005) Maximizing the stability radius of a set of systems under real-time scheduling constraints. IEEE Trans Autom Control 50(11):1790–1795
318. Papadimitriou CH (1994) Computational complexity. Addison-Wesley, Reading
319. Papoulis A, Pillai SU (2002) Probability, random variables and stochastic processes. McGraw-Hill, New York
320. Parrilo PA (2003) Semidefinite programming relaxations for semialgebraic problems. Math Program, Ser B 96(2):293–320
321. Parrondo JM, van den Broeck C (1993) Vapnik-Chervonenkis bounds for generalization. J Phys A 26:2211–2223

322. Petersen IR, McFarlane DC (1994) Optimal guaranteed cost control and filtering for uncertain linear systems. IEEE Trans Autom Control 39:1971–1977
323. Poljak S, Rohn J (1993) Checking robust nonsingularity is NP-hard. Math Control Signals Syst 6:1–9
324. Pollard D (1984) Convergence of stochastic processes. Springer, New York
325. Pollard D (1990) Empirical processes: theory and applications. NSF-CBMS regional conference series in probability and statistics, vol 2. Institute of Mathematical Statistics
326. Polyak BT (1964) Gradient methods for solving equations and inequalities. Ž Vyčisl Mat Mat Fiz 4:995–1005 (in Russian)
327. Polyak BT, Shcherbakov PS (2000) Random spherical uncertainty in estimation and robustness. IEEE Trans Autom Control 45:2145–2150
328. Polyak BT, Tempo R (2001) Probabilistic robust design with linear quadratic regulators. Syst Control Lett 43:343–353
329. Popescu I (1999) Applications of optimization in probability, finance and revenue management. PhD dissertation, Massachusetts Institute of Technology, Cambridge
330. Prékopa A (1995) Stochastic programming. Kluwer Academic Publishers, Dordrecht
331. Qiu L, Bernhardsson B, Rantzer A, Davison EJ, Young PM, Doyle JC (1995) A formula for computation of the real stability radius. Automatica 31:879–890
332. Ray LR, Stengel RF (1993) A Monte Carlo approach to the analysis of control system robustness. Automatica 29:229–236
333. Reemtsen R, Rückmann J-J (eds) (1998) Semi-infinite programming. Kluwer Academic Publishers, Dordrecht
334. Rubinstein RY, Kroese DP (2008) Simulation and the Monte-Carlo method. Wiley, New York
335. Rugh WJ (1996) Linear system theory. Prentice-Hall, Upper Saddle River
336. Safonov MG (1982) Stability margins of diagonally perturbed multivariable feedback systems. IEE Proc 129(D):251–256
337. Safonov MG (2012) Origins of robust control: early history and future speculations. In: Proceedings 7th IFAC symposium on robust control design
338. Safonov MG, Fan MKH (1997) Special issue on multivariable stability margin—editorial. Int J Robust Nonlinear Control 7:97–103
339. Sampei M, Mita T, Nakamichi M (1990) An algebraic approach to $\mathcal{H}_\infty$ output feedback control problems. Syst Control Lett 14:13–24
340. Sánchez-Peña RS, Sznaier M (1998) Robust systems: theory and applications. John Wiley, New York
341. Santos LF, Viola L (2006) Enhanced convergence and robust performance of randomized dynamical decoupling. Phys Rev Lett 97:150501/1–4
342. Santos LF, Viola L (2008) Advantages of randomization in coherent quantum dynamical control. New J Phys 10:083009/1–36
343. Sauer N (1972) On the density of families of sets. J Comb Theory 13(A):145–147
344. Scherer C (1990) The Riccati inequality and state space $\mathcal{H}_\infty$-optimal control. PhD dissertation, University of Würzburg
345. Scherer C (1995) Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control. In: Isidori A (ed) Trends in control: a European perspective. Springer, New York, pp 173–216
346. Scherer C, Gahinet P, Chilali M (1997) Multiobjective output feedback control via LMI optimization. IEEE Trans Autom Control 42:896–911
347. Scherer CW (1992) $\mathcal{H}_\infty$ control for plants with zeros on the imaginary axis. SIAM J Control Optim 30:123–142
348. Scherer CW (1992) $\mathcal{H}_\infty$ optimization without assumptions on finite or infinite zeros. SIAM J Control Optim 30:143–166
349. Scherer CW, Hol CWJ (2006) Matrix sum-of-squares relaxations for robust semi-definite programs. Math Program, Ser B 107(1-2):189–211
350. Schrijver A (1998) Theory of linear and integer programming. Wiley, New York
351. Schweppe FC (1973) Uncertain dynamical systems. Prentice-Hall, Englewood Cliffs

352. Seidenberg A (1954) A new decision method for elementary algebra. Ann Math 60:365–374
353. Selberg A (1944) Bemerkninger om et multiplet integral. Nor Mat Tidsskr 26:71–78
354. Shcherbakov PS, Dabbene F (2011) On the generation of random stable polynomials. Eur J Control 17(2):145–161
355. Shor NZ (1977) Cut-off method with space dilation in convex programming problems. Cybernetics 13(3):94–96
356. Skelton RE, Iwasaki T, Grigoriadis KM (1998) A unified algebraic approach to linear control design. Taylor & Francis, London
357. Skogestad S, Postlethwaite I (1996) Multivariable feedback control: analysis and design. Wiley, New York
358. Smith MA, Caracoglia L (2011) A Monte Carlo based method for the dynamic "fragility analysis" of tall buildings under turbulent wind loading. Eng Struct 33(2):410–420
359. Smith RL (1984) Efficient Monte-Carlo procedures for generating points uniformly distributed over bounded regions. Oper Res 32:1296–1308
360. Snedecor GW, Cochran WG (1989) Statistical methods. Iowa State Press, Ames
361. Sobol' IM (1967) The distribution of points in a cube and the approximate evaluation of integrals. Ž Vyčisl Mat Mat Fiz 7:784–802 (in Russian)
362. Song D, Gupta AK (1997) L_p-norm uniform distribution. Proc Am Math Soc 125:595–601
363. Sontag ED (1998) VC dimension of neural networks. In: Bishop CM (ed) Neural networks and machine learning. Springer, New York
364. Spall JC (2003) Estimation via Markov Chain Monte Carlo. IEEE Control Syst Mag 23:34–45
365. Spall JC (2003) Introduction to stochastic search and optimization: estimation, simulation, and control. Wiley, New York
366. Stengel RF (1980) Some effects of parameter variations on the lateral-directional stability of aircraft. AIAA J Guid Control 3:124–131
367. Stengel RF (1986) Stochastic optimal control: theory and application. Wiley, New York
368. Stevens BL, Lewis FL (2003) Aircraft control and simulation. Wiley, New York
369. Stewart GW (1980) The efficient generation of random orthogonal matrices with an application to condition estimators. SIAM J Numer Anal 17:403–409
370. Stieltjes TJ (1894) Recherches sur les fractions continues. Ann Fac Sci Toulouse 8:1–122
371. Stieltjes TJ (1895) Recherches sur les fractions continues. Ann Fac Sci Toulouse 9:5–47
372. Sukharev AG (1971) Optimal strategies of the search for an extremum. Ž Vyčisl Mat Mat Fiz 11:910–924 (in Russian)
373. Sznaier M, Amishima T, Parrilo PA, Tierno J (2002) A convex approach to robust $\mathcal{H}_2$ performance analysis. Automatica 38:957–966
374. Sznaier M, Lagoa CM, Mazzaro MC (2005) An algorithm for sampling subsets of $\mathcal{H}_\infty$ with applications to risk-adjusted performance analysis and model (in)validation. IEEE Trans Autom Control 50(3):410–416
375. Sznaier M, Rotstein H, Juanyu B, Sideris A (2000) An exact solution to continuous-time mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control problems. IEEE Trans Autom Control 45:2095–2101
376. Tahbaz-Salehi A, Jadbabaie A (2008) A necessary and sufficient condition for consensus over random networks. IEEE Trans Autom Control AC-53:791–795
377. Talagrand M (1996) New concentration inequalities in product spaces. Invent Math 126:505–563
378. Tanner HG, Piovesan JL (2010) Randomized receding horizon navigation. IEEE Trans Autom Control 55(11):2640–2644
379. Tarski A (1951) A decision method for elementary algebra and geometry. University of California Press, Berkeley
380. Tausworthe RC (1965) Random numbers generated by linear recurrence modulo two. Math Comput 19:201–209
381. Tempo R, Bai E-W, Dabbene F (1996) Probabilistic robustness analysis: explicit bounds for the minimum number of samples. In: Proceedings of the IEEE conference on decision and control, pp 3424–3428

382. Tempo R, Bai E-W, Dabbene F (1997) Probabilistic robustness analysis: explicit bounds for the minimum number of samples. Syst Control Lett 30:237–242
383. Tempo R, Blanchini F (2010) Robustness analysis with real parametric uncertainty. In: Levine WS (ed) The control handbook, 2nd edn. Control system advanced methods. CRC Press, Boca Raton, pp 7.1–7.18
384. Tempo R, Ishii H (2007) Monte Carlo and Las Vegas randomized algorithms for systems and control: an introduction. Eur J Control 13:189–203
385. Todd MJ (2001) Semidefinite optimization. Acta Numer 10:515–560
386. Toker O (1998) On the complexity of purely complex μ computation and related problems in multidimensional systems. IEEE Trans Autom Control 43:409–414
387. Tong YL (1980) Probability inequalities in multivariate distributions. Academic Press, New York
388. Traub JF, Wasilkowski GW, Woźniakowski H (1988) Information-based complexity. Academic Press, New York
389. Traub JF, Werschulz AG (1998) Complexity and information. Cambridge University Press, Cambridge
390. Tremba A, Calafiore G, Dabbene F, Gryazina E, Polyak BT, Shcherbakov PS, Tempo R (2008) RACT: randomized algorithms control toolbox for MATLAB. In: Proceedings 17th IFAC world congress, pp 390–395
391. Truxal JG (1961) Control systems—some unusual design problems. In: Mishkin E, Braun L (eds) Adaptive control systems. McGraw-Hill, New York, pp 91–118
392. Tulino AM, Verdú S (2004) Random matrices and wireless communications. Found Trends Commun Inf Theory 1(1):1–184
393. Tyler JS, Tuteur FB (1966) The use of a quadratic performance index to design multivariable invariant plants. IEEE Trans Autom Control 11:84–92
394. Ugrinovskii V (2011) Distributed robust filtering with $\mathcal{H}_\infty$ consensus of estimates. Automatica 47(1):1–13
395. Ugrinovskii VA (2005) Randomized algorithms for robust stability and guaranteed cost control of stochastic jump parameter systems with uncertain switching policies. J Optim Theory Appl 124(1):227–245
396. Uryasev S (ed) (2000) Probabilistic constrained optimization: methodology and applications. Kluwer Academic Publishers, New York
397. Uspensky JV (1937) Introduction to mathematical probability. McGraw-Hill, New York
398. Valavanis KP (ed) (2007) Advances in unmanned aerial vehicles: state of the art and the road to autonomy. Springer, New York
399. van der Corput JG (1935) Verteilungsfunktionen i, ii. Proc K Ned Akad Wet 38(B):813–82110581066
400. Vandenberghe L, Boyd S (1996) Semidefinite programming. SIAM Rev 38:49–95
401. Vapnik VN (1998) Statistical learning theory. Wiley, New York
402. Vapnik VN, Chervonenkis AY (1971) On the uniform convergence of relative frequencies to their probabilities. Theory Probab Appl 16:264–280
403. Vein R, Dale P (1999) Determinants and their applications in mathematical physics. Springer, New York
404. Vidyasagar M (1998) Statistical learning theory and randomized algorithms for control. IEEE Control Syst Mag 18:69–85
405. Vidyasagar M (2001) Randomized algorithms for robust controller synthesis using statistical learning theory. Automatica 37:1515–1528
406. Vidyasagar M (2002) Learning and generalization: with applications to neural networks, 2nd edn. Springer, New York
407. Vidyasagar M (2011) Probabilistic methods in cancer biology. Eur J Control 17:483–511
408. Vidyasagar M, Blondel V (2001) Probabilistic solutions to some NP-hard matrix problems. Automatica 37:1397–1405
409. von Neumann J (1951) Various techniques used in connection with random digits. US Nat Bur Stand Appl Math Ser 36–38

410. Wada T, Fujisaki Y (2007) Sequential randomized algorithms for robust optimization. In: Proceedings IEEE conference on decision and control, pp 6190–6195
411. Wang Q, Stengel RF (2005) Robust nonlinear flight control of a high performance aircraft. IEEE Trans Control Syst Technol 13:15–26
412. Wenocur RS, Dudley RM (1981) Some special Vapnik-Chervonenkis classes. Discrete Math 33:313–318
413. Willems JC, Tempo R (1999) The Kharitonov theorem with degree drop. IEEE Trans Autom Control 44:2218–2220
414. Wu CW (2006) Synchronization and convergence of linear dynamics in random directed networks. IEEE Trans Autom Control AC-51:1207–1210
415. Yang KY, Hall SR, Feron E (2000) Robust $\mathcal{H}_2$ control. In: Ghaoui LE, Niculescu S-I (eds) Advances in linear matrix inequality methods in control. SIAM, New York, pp 155–174
416. Youla DC, Saito M (1967) Interpolation with positive-real functions. J Franklin Inst 284:77–108
417. Young PM (1994) The rank one mixed μ problem and "Kharitonov-type" analysis. Automatica 30:1899–1911
418. Yudin DB, Nemirovski AS (1977) Informational complexity and efficient methods for solving complex extremal problems. Matekon 13(3):25–45
419. Zaki N, Berengueres J, Efimov D (2012) Detection of protein complexes using a protein ranking algorithm. Proteins, accepted for publication
420. Zames G (1981) Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms and approximate inverses. IEEE Trans Autom Control 26:301–320
421. Zhigljavsky AA (1991) Theory of global random search. Kluwer Academic Publishers, Dordrecht
422. Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice-Hall, Upper Saddle River
423. Zhou T, Feng C (2006) Uniform sample generation from contractive block Toeplitz matrices. IEEE Trans Autom Control 51:1559–1565
424. Zyczkowski K, Kus M (1994) Random unitary matrices. J Phys 27(A):4235–4245

Index

R. Tempo et al., *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering, DOI 10.1007/978-1-4471-4610-0,