Randomized Algorithms for Control of Uncertain Systems with Application to Hard Disk Drives

Mohammadreza Chamanbaz

NATIONAL UNIVERSITY OF SINGAPORE

2014

Randomized Algorithms for Control of Uncertain Systems with Application to Hard Disk Drives

Mohammadreza Chamanbaz

B.Sc., Shiraz University of Technology (SUTECH)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2014

Declaration

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Mohammadreza Chamanbaz

Student's Signature

14 May 2014

Date

Acknowledgments

First and for most, I thank God for giving me the opportunity to exist and for His continuous support throughout my entire life.

The four years PhD study was a journey and I was very lucky not to be alone in this journey. Undoubtedly, this journey was impossible without the support and encouragement of my family, friends and colleagues. I thank my advisors Dr. Thomas Liew, Dr. Venkatakrishnan Venkataramanan and Prof. Qing Guo Wang for giving me the opportunity to pursue my PhD study under their supervision. I am also very grateful to Prof. Roberto Tempo and Dr. Fabrizio Dabbene who generously hosted me in IEIIT, Torino, Italy during my six months visit which formed the framework of my thesis.

Apart from technical supports, I am very blessed to have lots of good friends without whom I couldn't survive. They were my second family who made Singapore as home for me.

I also wish to thank my beloved wife Faezeh for her warm supports in the last stages of my PhD.

Lastly but most importantly, a special thanks goes to my mother who was my main supporter throughout my study from primary school till now. She had such a perseverance in inspiring me not to give up my study. I am so grateful for her unconditional support, encouragement, trust and sympathy in my life. Words are not adequate to express my gratitude towards her!

Contents

Summary

The presence of "uncertainty" in dynamical systems is inevitable. Different imperfections such as manufacturing tolerances, different raw materials and slight change in the environmental condition of the production line contribute to slight difference in the dynamics over a batch of products. In robust control, this difference is modeled as parametric and non-parametric (dynamic) uncertainties. Dynamic uncertainty can be handled efficiently using $\mu-$theory however, coming to parametric uncertainty, most deterministic approaches suffer from conservatism and computational complexity. Motivated by this, in the present thesis we propose two classes of randomized algorithms: i) Sequential randomized algorithms for solving uncertain convex optimization problems and ii) Randomized algorithms for solving uncertain linear and bilinear matrix inequalities using statistical learning theory.

Motivated by the complexity of solving convex scenario problems in one-shot, in Chapter 2 we provide a direct connection between this approach and sequential randomized methods. A rigorous analysis of the theoretical properties of two new algorithms, for full constraint satisfaction and partial constraint satisfaction, is pro-

vided. These algorithms allow enlarging the applicability domain of scenario-based methods to problems involving a large number of design variables. In this approach, we solve a set of scenario optimization problems with increasing complexity. In parallel, at each step we validate the candidate solution using Monte-Carlo simulation. Simulation results prove the effectiveness of the proposed algorithms.

In the second class of randomized algorithms, in Chapter 3 we consider the problem of minimizing a linear functional subject to uncertain linear and bilinear matrix inequalities, which depend in a possibly nonlinear way on a vector of uncertain parameters. Motivated by recent results in statistical learning theory, we show that probabilistic guaranteed solutions can be obtained by means of randomized algorithms. In particular, we show that Vapnik-Chervonenkis dimension (VC-dimension) of the two problems is finite, and we compute upper bounds on it. In turn, these bounds allow us to derive explicitly the sample complexity of these problems. Using these bounds, we derive a sequential scheme based on a sequence of optimization and validation steps. The effectiveness of this approach is shown using a linear model of a robot manipulator subject to uncertain parameters.

In the second part of thesis, we consider the problem of parametric uncertainty in hard disk drive servo systems and using the proposed algorithms of Chapter 2, we design robust $\mathcal{H}_2$ dynamic output feedback controllers to handle multiple parametric uncertainties entering in plant description in a nonlinear fashion. We also design the same controller using sequential approximation methods based on cutting plane

iterations. Extensive simulations compare the worst case track following performance and stability margins.

List of Tables

List of Figures

Chapter 1

Introduction

Recently, there have been significant efforts devoted to solving uncertain control problems. Introducing uncertainty in the problem data makes the resulting problem very difficult to solve. On the other hand, almost all industrial problems involve a number of uncertain parameters resulted from factors such as manufacturing tolerances or slightly different raw materials and environmental conditions. Ignoring uncertainty in the system can tend to erroneous result which may cause significant damages or loss. In general, there are two paradigms to tackle uncertain problems. The first approach is based on deterministic min-max or worst-case methodology. The solution obtained using this approach is feasible for the entire uncertainty set. The second approach is based on chance constraint programming in which the uncertainty vector is considered as random variable and by introducing a risk term, the solution is enforced to be feasible with the desired high probability. Chance constraint programming is very difficult to solve exactly and even if the original problem is convex, the chance constraint problem becomes non-convex in general. In contrast, in min-max approach the convexity is preserved; but, infinite number of constraints are involved which makes the problem difficult to solve. For this reason some relaxation techniques are usually employed in order to recast the infinite number of constraints into a finite number. Unfortunately, relaxation techniques are just applicable to cases where uncertain parameters appear in a "simple" form such as affine, multi-affine or rational. However, in most real world problems the uncertainty structure is very complicated. Hence, very recently, researchers proposed using *randomized algorithms* in which by generating random choices we can settle the difficulty associated with chance constraint programming.

Randomized strategies in solving complex problems have gained more attention than the recent past. Randomized strategies are useful in two classes of problems: analysis and design problems. Analysis problems arise when we want to validate a given solution and design problems appear when we want to find a solution. In the subsequent sections, we review the major contributions in both deterministic worst-case approach and probabilistic methods based on randomized algorithms.

1.1 Classical Robust Techniques

In this section we review major contributions in classical robust literature. We highlight that the discussion of the section is not a comprehensive review of all robust techniques. Interested readers are referred to [21, 23, 18, 45, 55, 108, 136, 103, 63, 49, 56, 113] for extensive discussions.

1.1.1 Historical Notes

Linear Quadratic Gaussian (LQG) and Kalman filter can be considered as the earliest efforts addressing uncertainty. In this form, uncertainty is observed as exogenous disturbance having stochastical representation, while the dynamical plant is assumed to be known exactly. The approach is known as *classical stochastic* method. There have been some efforts since early 1980s to introduce uncertainty directly into the dynamical plant. In most cases the goal is to design a controller that remains robust against all possible uncertainty scenarios. The paradigm is known as *worst-case* approach. The most important breakthrough in the worst case methodology was the formulation of Zames for $\mathcal{H}_\infty$ problem [135] in 1981. In early 1990s, robust control was well-known in industry with applications in aerospace, chemical, electrical and mechanical engineering. At the same time, some of the theoretical limitations of classical robust techniques such as conservatism and computational complexity were realized in the robust control community. A few years latter, some tools from robust optimization discipline such as semidefinite programming (SDP) [120] were introduced in robust control. Most robust control problems such as $\mathcal{H}_2$, $\mathcal{H}_\infty$, and $\mu-$synthesis were formulated into the form of linear matrix inequalities (LMIs) which is a convex optimization problem encompassing linear, quadratic and conic programs. Introduc-

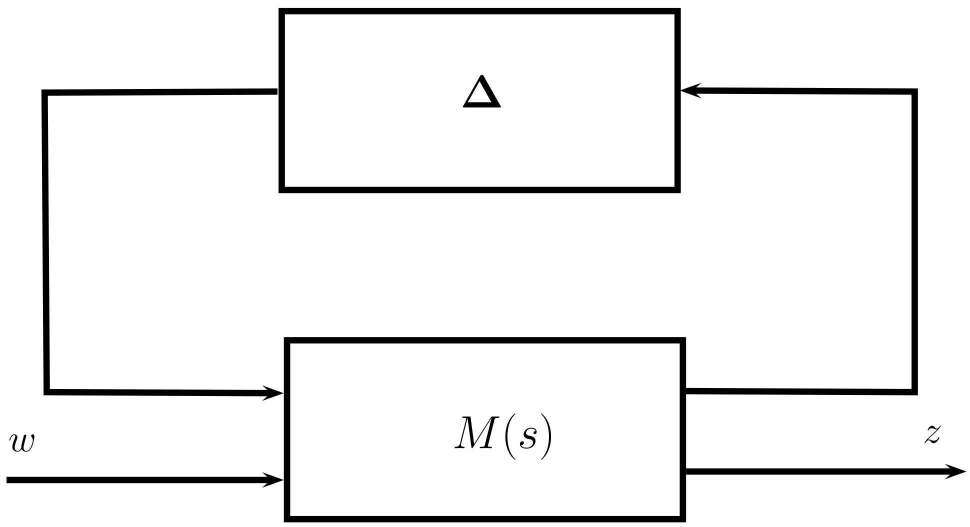

Figure 1.1: $M - \Delta$ configuration with disturbance w and output z.

ing LMI in robust control can be considered as the second breakthrough after Zame's formulation. A number of numerically efficient softwares and algorithms such as interior point method in particular [94] were developed for solving LMIs. See [21] for a comprehensive discussion on LMIs in systems and control theory.

1.1.2 Robustness Analysis

All sources of uncertainty can be categorized into two main groups:

- Parametric uncertainty
- Dynamic uncertainty

The former refers to the case where some parameters in the plant are uncertain such as uncertain resonance frequency or damping ratio. The later refers to the case where nothing is known about the source of uncertainty except that it is bounded such as high frequency un-modeled dynamics. In order to handle dynamic uncertainty, the uncertain system needs to be formulated in the standard description of $M - \Delta$ configuration shown in Figure 1.1. $M(s)$ represents the combination of the nominal plant and controller transfer matrices while $\boldsymbol{\Delta}$ contains parametric as well as non-parametric uncertainties in its diagonal element:

$$\boldsymbol{\Delta} = \{\text{Blockdiag}[\boldsymbol{\Delta_1}, \boldsymbol{\Delta_2}, \ldots, \boldsymbol{\Delta_{n_d}}, \mathbf{q_1 I_1}, \mathbf{q_2 I_2}, \ldots, \mathbf{q_{n_p} I_{n_p}}]\}$$

where q_i, $i \in \{1, \ldots, n_p\}$ are parametric uncertainties, I_i is the identity matrix of dimension i and Δ_i, $i \in \{1, \ldots, n_d\}$ are dynamic uncertainties extracted from the uncertain control system. The earliest approach for evaluating robustness of the uncertain control system depicted in Figure 1.1 was based on small gain theorem (see e.g. [136]) in which the internal stability of the interconnected system is examined by evaluating $\mathcal{H}_\infty$ norm of $M(s)$ and $\boldsymbol{\Delta}$. However, small gain theorem is conservative in the sense that it does not take into consideration the structure of $\boldsymbol{\Delta}$. Structured singular value also known as $\mu-$theory [97] was introduced to overcome this limitation. Nevertheless, computing structured singular value μ is an NP-hard problem [22] for which there is no polynomial time algorithm.

In cases where the uncertain system contains a number of parametric uncertainties, the optimization problem used for computing μ , known as $D-K$ iteration, fails to converge. Therefore, $\mu-$analysis is not an efficient tool for evaluating robustness when the uncertain plant contains a number of parametric uncertainties. The earliest attempt directly dealing with analysis of polynomials affected by parametric uncertainty was the Kharitonov theorem [79]. In this approach, four specially designed polynomials known as "Kharitonov polynomials" are formulated; the stability of the uncertain polynomial is evaluated by checking the stability of Kharitonov polynomials. This approach has been improved in [59, 71, 130]. Kharitonov approach is very powerful in the sense that it only requires checking four "extreme" polynomials. Nevertheless, it is only applicable to cases where polynomial coefficients are independent and bounded in an interval. This limitations was partially addressed using edge theorem [12]. In order to apply edge theorem to a polynomial, the dependence of polynomial coefficients on uncertain parameter needs to be "affine". The value set analysis [23] is another important tool for evaluating the stability of a given uncertain polynomial in frequency domain. This approach can handle cases where coefficients of polynomial are "multi-affine" function of uncertainty vector.

The polynomial techniques which are presented very recently are deterministic methods based on tools from algebraic geometry leading to generalization of the linear matrix inequality and semi-definite programming. Recent activities in this line of research are mainly due to sum of squares relaxations [36, 100] and moment problems formulation in dual spaces [83]. This approach reformulate the control and optimization problems subject to multivariate polynomial inequalities. The question regarding when a non-negative polynomial can be expressed as sum of squares was studied in classical texts, see [19] for historical notes on polynomial non-negativity.

The link between sum of squares and convexity is discussed in [107] and the specific relation with semi-definite programming is discussed in many papers, see e.g. [36, 100, 83, 81]. These relaxation techniques build a hierarchy of convex relaxations of the uncertain optimization problems. The relaxations provide a conservative solution to the original uncertain optimization problem. Under mild assumptions they provide asymptotic convergence of the solution of the convex relaxations to the solution of the uncertain optimization problem. The main difficulty in using such relaxations is their complexity making such approaches difficult to use in practice.

1.1.3 Robust Synthesis

The formulation of Zames for $\mathcal{H}_\infty$ [135] was the first attempt to introduce uncertainty directly into the plant description. Later, some classical optimal methods were developed such as the idea of structured singular value also known as $\mu-$theory [97] which led to the μ-synthesis controller, the optimization methods based on semi definite programming which in engineering is known as Linear Matrix Inequality (LMI) [21] and l_1 optimal control theory [42]. Later on the state feedback design based on multi objective optimization was introduced [15, 78, 47]; however, these methods were suffering from two drawbacks: Firstly, the design procedure was based on state feedback. Secondly, they required selected input or output channels to be the same for all the objectives. In 1997, the design of multi objective dynamic output feedback was proposed by Scherer [105]. The proposed design procedure by Scherer didn't suffer from two previously mentioned limitations. The design objective could be combination of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ performance, passivity, asymptotic disturbance rejection, time domain constraints and constraints on the closed loop pole location. The whole idea was to express the closed loop objectives in terms of LMI; usually expressing the closed loop state space matrices in terms of plant model and controller matrices (or design parameters) causes the problem to be non-linear (or rather non-affine) with respect to design parameters. Hence, by introducing some non-linear transformations and change of variables the problem is changed back to LMI format. In the design approach base on [105], all Lyapunov matrices were required to be the same for all objectives which is rather conservative. The idea of using multiple Lyapunov functions was proposed by De Oliveira in [43] and the controller design based on this approach was presented in [44] by the same authors. In this framework control variables were independent from Lyapunov matrices that are used to test stability of the closed loop system; this feature allows using parameter dependent Lyapunov function which considerably reduces conservatism. In all approaches which are mentioned so far, no uncertainty is considered in the plant model. In case where controller parametrization does not explicitly depends on the state space matrices of the controlled system, extension to polytypic uncertainty is trivial. For instance, state feedback controller design for $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control [98] can be mentioned. It is well known that design of a globally optimal full order output feedback controller for polytypic uncertain system is non-convex NP-hard optimization problem which can be represented in the form of Bilinear Matrix Inequality (BMI) optimization problem [119]. In [74] a computationally efficient locally optimal controller was presented. The design procedure is guaranteed to converge to a local optimum. There are a couple of approaches for solving BMI optimization problems. The simplest one is based on coordinate decent method which fixes one variable (change BMI to LMI) and solves the LMI optimization problem next, fixes the other design variable and does the same [69]. This approach is not guaranteed to converge to a local optimum. The interior point method [85], path following [57], rank minimization [68] are some other alternatives. Nevertheless, non of them is guaranteed to converge to a local optimum. The method of center [53] has guaranteed local convergence, nevertheless, it is computationally very expensive. Considering above mentioned points the approach proposed in [74] is the best for dealing with parametric uncertainty; however, the computational complexity grows exponentially with respect to the number of uncertain parameters. Hence, it can only manage a limited number of uncertain parameters.

1.2 Limitation of Deterministic Worst-Case Approach

Although classical robust methods have been improved since 1980's, there are still some limitations and bottlenecks for applying this approach to practical problems. AS an example, computing the structured singular value μ is proved to be $\mathcal{NP}-$hard.

In general, the limitations of deterministic paradigm can be categorized into two different classes discussed in the next two subsections.

1.2.1 Computational Complexity

Running any arithmetic operation takes an specific amount of time in processing unit. Hence, running time is the sum of all time intervals which are required to solve the problem under consideration. When an algorithm runs in "polynomial time", it means that there exists an integer k such that:

$$T(n) = O(n^k)$$

where $T(n)$ is the running time which is a function of the size of problem at hand n. Generally speaking, problems which have polynomial time algorithm are solvable. Then the term $\mathcal{NP}-$hard stands for non-deterministic polynomial time-hard problems for which there is no polynomial time algorithm. In other words, when a problem is $\mathcal{NP}-$hard, it implies there is no upper limit in terms of time that we can make sure that the problem will be solved within this time interval. There are a lot of problems in robust control which belong to the category of $\mathcal{NP}-$hard problems. On the other hand, even when a problems has a polynomial-time algorithm, it does not mean that it can be solved efficiently. There are some problems which have polynomial-time algorithms and can't be solved due to the huge computational burden associated with them.

1.2.2 Conservatism

In addition to the complexity problem, conservatism is also a challenge for the deterministic robust approach. It is well known that in cases where real parametric uncertainty enters affinely into plant transfer function, it is possible to compute the robustness margin exactly. However, in real world problems, we usually deal with non-linear non-convex uncertainty. In order to handle this problem in classical robust paradigm, the non-linear uncertainty is embedded into affine structure by replacing the original set by a larger one. In other words, multipliers and scaling variables are introduced to relax the problem [14] which are associated with an evident conservatism. On the other hand, it is well known that robustness and performance are two contradicting requirements, which means increase in robustness tends to degradation in performance. In critical applications where performance is of vital importance, unnecessary conservatism is not desired and should be avoided.

1.3 Probabilistic Methods in Robust Control

In this section, we discuss the probabilistic and randomized methods used in robust control for analysis and synthesis of uncertain systems. Interested readers are referred to [30, 113] for a comprehensive treatment.

1.3.1 Historical Notes

The concept of probabilistic robust control is quite recent although its root goes back to 1980 in the field of flight control [109]. Some papers have been published during 1980's and early 1990's mostly dealing with analysis problem based on Monte Carlo simulation. The concept of probability of instability was introduced in this period. The new era of this field was started by papers [77, 112] in 1996 which derived an explicit sample bound based on which, we can estimate probability of satisfaction or violation of a given cost function. Subsequently, the results based on statistical learning theory [125, 124] by Vidyasagar was proposed which plays an important role in solving non-convex problems. Randomized algorithms for solving uncertain linear quadratic regulator (LQR) [101] and uncertain linear matrix inequalities (LMIs) [24] were a stepping stone in the field of randomized algorithms. Nevertheless, this approach can only solve feasibility problems. The non-sequential method for solving uncertain convex optimization problems, the so-called scenario approach, was introduced in 2004 [26] which was the only approach capable of directly solving optimization problems. The direct application of statistical learning theory for solving non-convex problem were also introduced in [5]. The class of sequential probabilistic validation algorithms were recently presented in [4] proposing a unified scheme which can be efficiently used in sequential synthesis methods such as gradient iteration.

1.3.2 Randomized Algorithms for Analysis

The main ingredient in analysis techniques based on randomized algorithms is to extract N independent and identically distributed (iid) samples from the uncertainty set and examine a performance function for all the random samples. In general, there are two problems to be tackled in probabilistic analysis:

1. Reliability estimation.

2. Performance level estimation.

In reliability estimation, we aim at "estimating a probability" which can be the probability of satisfaction (or violation) of a given performance index such as $\mathcal{H}_\infty$ norm. This problem historically goes back to Markov [88] and Chebychev [35] inequalities. Hoeffding [61] and Bernstein [17] derived the required sample bounds for estimating an unknown probability. This line of research has a very rich background, interested readers are referred to [111, 102, 99, 89].

In Performance level estimation, the goal is to estimate the "worst case" performance of a given performance index over the uncertainty set. Worst case $\mathcal{H}_\infty$ norm estimation is an example for which this methodology can be effectively used. The famous log-over-log bound [112] propose a sample complexity bound for solving such problems. We highlight that introducing log-over-log bound was a stepping stone in probabilistic robustness analysis. Sequential probabilistic validation algorithms (SPV) highly rely on log-over-log bound. This class of algorithms was *formally* introduced in [4]. Nevertheless, they have been widely used in probabilistic robust literature such as [29, 95, 52, 41] as a part of sequential randomized algorithm for controller synthesis.

1.3.3 Randomized Algorithms for Control Synthesis

Figure 1.2 shows an overview of all techniques which are used in probabilistic robust design. Starting from the top, problems can be divided into two classes: convex

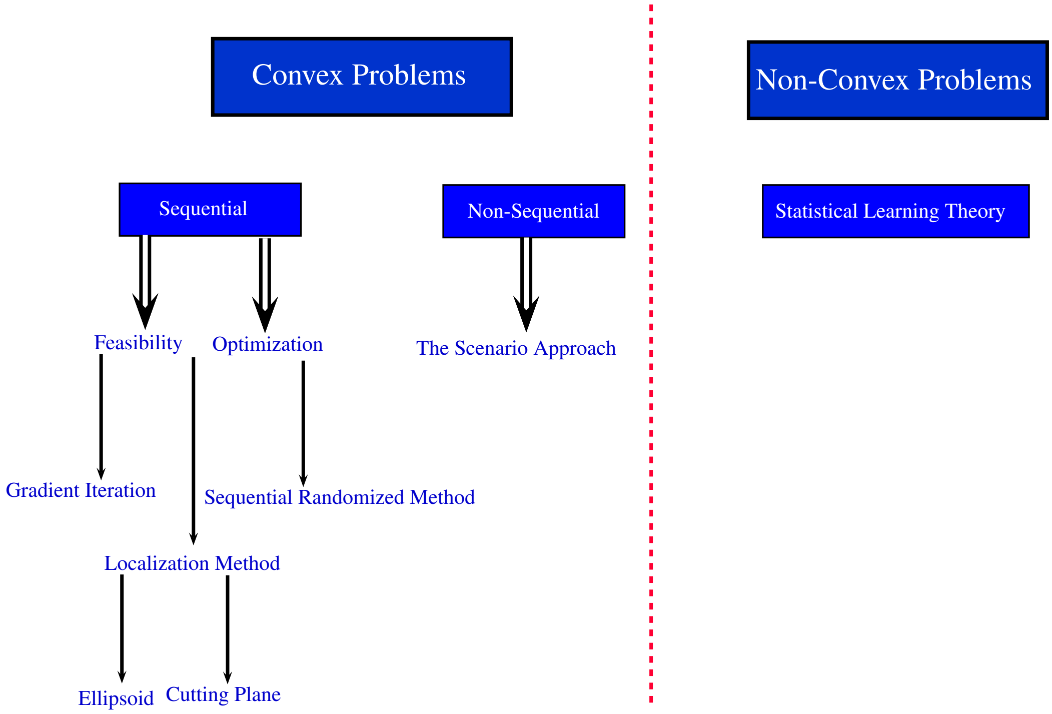

Figure 1.2: Probabilistic design methods

and non-convex problems. For convex problems, there are two classes of sequential and non-sequential randomized methods. Sequential methods are based on a sequence of design and validation steps which are performed iteratively to find a probabilistic solution. The design part is purely deterministic which roots in stochastic optimization techniques for feasibility problems. The simplest one is the gradient method while more sophisticated methods such as ellipsoid [73, 95] and cutting-plane [29, 41] are localization methods trying to shrink a localization set at each iteration. At each iteration of such algorithms, a candidate solution is constructed in the design step and is then validated using a Monte Carlo simulation. In the case that Monte Carlo simulation declares the solution as a probabilistic robust feasible solution, the algorithm is terminated otherwise the algorithm goes back to the design step and constructs another candidate solution. The convergence of such algorithms is proved under some mild assumptions. Sequential randomized methodology for optimization was first introduced in [34], see Chapter 2 for a detailed discussion of this approach. The non-sequential algorithm (the scenario approach) [26] is based on extracting random samples from the uncertainty set and solving an optimization problem subject to finite number of constraints. The approach was extended in [31, 27, 32, 25] dealing with the so called scenario with discarded constraints in which we purposely discard a number of constraint in favor of improving the objective value. Statistical learning theory [125, 5] is the only approach for which the convexity does not play any role and hence is suitable for non-convex problems. Nevertheless, it is not very easy to apply this concept on control problems since it requires the computation of a combinatorial parameter called Vapnik-Chervonenkis dimension (VC-dimension). In probabilistic methods for controller design, the stability of the closed loop system is treated in a probabilistic sense, then the closed loop plant may tend to instability for some very unfortunate scenarios. This limitation was addressed in [28] which divides the performance specifications into two categories, hard and soft. Hard ones are those which must satisfy in deterministic sense (such as stability) and soft ones are those which their violation does not result in a big failure. This approach, tries to find the stabilizing controller set based on Youla parametrization and next, the stabilizing set is searched for a controller which gives the best performance.

1.4 Outline of the Thesis

This section aims at providing an outline of the thesis. In particular, we provide an overview of the relationship between different contributions presented in different chapters. The reader can use this section to find his/her way through the thesis.

1.4.1 Sequential Randomized Algorithms for Samples Convex Optimization

The approach presented in the Chapter 2 aims at alleviating the conservatism associated with the scenario approach also known as sampled convex program. As briefly mentioned earlier, the scenario approach is a non-sequential method in which we solve an optimization problem subject to finitely many random constraints extracted from the uncertainty set. In other words, this approach reduces infinite number of constraints to a finite number. The drawback of the scenario approach is its computational complexity. The number of random constraints needs to be extracted from the uncertainty set is sometimes very large leading to a very complex optimization problem which is beyond the capability of the current computational tools. Motivated by this limitation, we mixed the scenario approach with sequential randomized methods used in sequential approximation methods based on gradient [24, 101], ellipsoid [73] and cutting plane [29] iterations and introduced sequential randomized algorithms for solving uncertain convex optimization problems. The main philosophy is to form a temporary solution by solving a "reduced size" scenario problem and then to check the candidate solution in a validation (analysis) step to see if the solution satisfies the desired probabilistic behaviour. In the case that validation step fails declaring the candidate solution a "bad" solution, an optimization problem subject to larger number of random constraints is solved to obtain a "more robust" solution. In the algorithm the two steps are iteratively performed to obtain a probabilistic robust solution. The convergence of algorithms are rigorously proved in Chapter 2. We highlight that the proposed sequential randomized algorithms are the first sequential methods capable of directly solving uncertain optimization problems. They also extend the applicability domain of the scenario approach: there are problems for which the scenario approach cannot solve the optimization problem due to its complexity but, using the sequential randomized algorithms developed in this thesis we can efficiently solve the problem.

1.4.2 Vapnik-Chervonenkis Dimension of Uncertain LMI and BMI

Statistical learning theory is a very powerful tool in solving uncertain complex problems. In this line of research statistical learning theory is used to extend the convergence property of the empirical mean, which can be obtained using a Monte Carlo simulation, from finite families to infinite family of functions. Nevertheless, applying results from statistical learning theory to control problems requires the computation of a parameter called Vapnik-Chervonenkis dimension (VC-dimension) which is very difficult in general. There are some attempts in the literature to compute the VC-dimension for some control problems such as static output feedback control presented in [127]. Motivated by this results, we computed VC-dimension for uncertain linear and bilinear matrix inequalities frequently encountered in robust and optimal control. In particular, we reformulated the problem of checking positive (negative) definiteness (semi-definiteness) as binary functions and computed the corresponding VC-dimension. Given the computed VC-dimension, we developed randomized algorithms capable of finding probabilistic guaranteed solutions. The only drawback of the approaches based on statistical learning theory is that they usually come up with huge sample bounds making the optimization problem very difficult to solve. To circumvent this, we used the developed strategy discussed in subsection 1.4.1 to develop a sequential randomized algorithm which can efficiently solve the problem with manageable computational effort.

1.4.3 Robust Track Following Control of Hard Disk Drives

Using the developed randomized algorithms, we solved a challenging industrial problem regarding the track following control of hard disk drives. In particular, we designed an $\mathcal{H}_2$ dynamic output feedback controller addressing several uncertain parameters which appear in the dynamical equations in a non-affine manner. To make a comparison we also designed a controller using sequential approximation method base on cutting plane iteration presented in [29] and compared the designed controller with two controllers designed by the sequential randomized algorithms discussed in subsection 1.4.1. We highlight that the performance of all controllers are fairly similar but, the controllers designed using the proposed methodology take considerably less time to design compared to the one designed by the sequential approximation method based on cutting plane algorithm.

Chapter 2

Sequential Randomized Algorithms for Uncertain Convex Optimization

2.1 Introduction

The approach that has emerged recently regarding non-sequential randomized methods is the so-called scenario approach, which has been introduced in [26, 27]. Taking random samples of the uncertainty $q \in \mathbb{Q}$, the main idea of this particular line of research is to reformulate a semi-infinite convex optimization problem as a sampled optimization problem subject to a finite number of random constraints. Then, a key problem is to determine the sample complexity i.e., the number of random constraints that should be generated, so that the so-called probability of violation is smaller than a given accuracy $\epsilon \in (0, 1)$, and this event holds with a suitably large confidence

$1 - \delta \in (0, 1)$. A very nice feature of the scenario approach is that the sample complexity is determined a priori, that is before the sampled optimization problem is solved, and it depends only on the number of design parameters, accuracy and confidence. On the other hand, if accuracy and confidence are very small, and the number of design parameters is large, then the sample complexity may be huge, and the sampled convex optimization problem cannot be easily solved in practice.

Motivated by this discussion, in this chapter we develop a novel sequential method specifically tailored to the solution of the scenario-based optimization problem. The proposed approach iteratively solves *reduced-size* scenario problems of increasing size, and it is particularly appealing for large-size problems. This line of research follows and improves upon the schemes previously developed for various control problems, which include linear quadratic regulators, linear matrix inequalities and switched systems discussed in [30, 113]. The main idea of these sequential methods is to introduce the concept of validation samples. That is, at step k of the sequential algorithm, a "temporary solution" is constructed and, using a suitably generated validation sample set, it is verified whether or not the probability of violation corresponding to the temporary solution is smaller than a given accuracy ε, and this event holds with confidence $1-\delta$. Due to their sequential nature, these algorithms may have wider applications than the scenario approach, in particular in real-world problems where fast computations are needed because of very stringent time requirements due to on-line implementations.

Compared to the sequential approaches discussed above, the methods proposed in this chapter have the following distinct main advantages: 1. no feasibility assumption of the original uncertain problem is required; 2. the termination of the algorithm does not require the knowledge of some user-determined parameters such as the center of a feasibility ball; 3. the methods can be immediately implemented using existing off-the-shelf convex optimization tools, and no ad-hoc implementation of specific update rules (such as stochastic gradient, ellipsoid or cutting plane) is needed. We also remark that the methods presented here directly apply to optimization problems, whereas all the sequential methods discussed in [30, 113] are limited to feasibility.

In this chapter, we study two new sequential algorithms for optimization with full constraint satisfaction and partial constraint satisfaction, respectively, and we provide a rigorous analysis of their theoretical properties regarding the probability of violation. These algorithms fall into the class of Sequential Probabilistic Validation (SPV) algorithms, but exploit specific convexity and finite convergence properties of scenario methods, thus showing computational improvements upon those presented in [4], see Section 2.3.1. In particular, the sample complexity of both algorithms is derived and it enters directly into the validation step. The sample complexity increases very mildly with probabilistic accuracy, confidence and number of design parameters, and depends on a termination parameter which is chosen by the user. In the worst case, an optimization problem having the same size of the scenario approach should be solved.

In the second part of the chapter, using a non-trivial example regarding the control of a multivariable model for the lateral motion of an aircraft, we provide extensive numerical simulations which compare upfront the sample complexity of the scenario approach with the number of iterations required in the two sequential algorithms previously introduced. We remark again that the sample complexity of the scenario approach is computed a priori, while for sequential algorithms, the numerical results regarding the size of the validation sample set are random. For this reason, mean values, standard deviation and other related parameters are experimentally computed for both proposed algorithms by means of extensive Monte Carlo simulations. Please see Chapter 4 for more sophisticated numerical example regarding the track-following control of hard disk drive.

2.2 Problem Formulation and Preliminaries

An uncertain convex problem has the form

$$\min_{\theta \in \Theta} \quad c^T \theta \tag{2.1}$$

$$\text{subject to } f(\theta, q) \leq 0 \text{ for all } q \in \mathbb{Q}$$

where $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ is the vector of optimization variables and $q \in \mathbb{Q}$ denotes random uncertainty acting on the system, $f(\theta, q) : \Theta \times \mathbb{Q} \to \mathbb{R}$ is convex in θ for any fixed value of $q \in \mathbb{Q}$ and Θ is a convex and closed set. We note that most uncertain convex problems can be reformulated as (2.1). In particular, multiple scalar-valued constraints $f_i(\theta, q) \leq 0,\ i = 1, \ldots, m$ can always be recast into the form (2.1) by defining $f(\theta, q) = \max_{i=1,\ldots,m} f_i(\theta, q)$.

In this chapter, we study a probabilistic framework in which the uncertainty vector q is assumed to be a random variable and the constraint in (2.1) is allowed to be violated for some $q \in \mathbb{Q}$, provided that the rate of violation is sufficiently small. This concept is formally expressed using the notion of "probability of violation".

Definition 2.1 (Probability of Violation)**.** *The probability of violation of θ for the function $f : \Theta \times \mathbb{Q} \to \mathbb{R}$ is defined as*

$$V(\theta) \doteq \Pr\{q \in \mathbb{Q} \ : \ f(\theta, q) > 0\}\,. \tag{2.2}$$

The exact computation of $V(\theta)$ is in general very difficult since it requires the computation of multiple integrals associated to the probability in (3.3). However, this probability can be estimated using randomization. To this end, assuming that a probability measure is given over the set $\mathbb{Q}$, we generate N independent identically distributed (i.i.d.) samples within the set $\mathbb{Q}$

$$\mathbf{q} = \{q^{(1)}, \ldots, q^{(N)}\} \in \mathbb{Q}^N,$$

where $\mathbb{Q}^N \doteq \mathbb{Q} \times \mathbb{Q} \times \cdots \times \mathbb{Q}$ (N times). Next, a Monte Carlo approach is employed to obtain the so called "empirical violation" which is introduced in the following definition.

Definition 2.2 (Empirical Violation)**.** *For given $\theta \in \Theta$ the empirical violation of*

$f(\theta, q)$ with respect to the multisample $\mathbf{q} = \{q^{(1)}, \ldots, q^{(N)}\}$ is defined as

$$\widehat{V}(\theta, \mathbf{q}) \doteq \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_f(\theta, q^{(i)}) \tag{2.3}$$

where $\mathbb{I}_f(\theta, q^{(i)})$ is an indicator function defined as

$$\mathbb{I}_f(\theta, q^{(i)}) \doteq \begin{cases} 0 & \textit{if}\, f(\theta, q) \leq 0 \\ \\ 1 & \textit{otherwise.} \end{cases}$$

It is clear that, based on the definition of $\mathbb{I}_f(\theta, q^{(i)})$, the empirical violation is a random variable bounded in the closed interval $[0, 1]$.

2.2.1 The Scenario Approach

In this subsection, we briefly recall the so-called scenario approach, also known as random convex programs, which was first introduced in [26, 27], see also [31] for additional results. In this approach, a set of independent identically distributed random samples of cardinality N is extracted from the uncertainty set and the following scenario problem is formed

$$\begin{aligned} &\min_{\theta \in \Theta} \quad c^T \theta \\ &\text{subject to } \; f(\theta, q^{(i)}) \leq 0, \; i = 1, \ldots, N. \end{aligned} \tag{2.4}$$

The function $f(\theta, q)$ is convex for fixed $q \in \mathbb{Q}$ and a further assumption is that the problem (2.4) attains a unique solution $\widehat{\theta}_N$. These assumptions are now formally stated.

Assumption 2.1 (Convexity). *$\Theta \subset \mathbb{R}^{n_\theta}$ is a convex and closed set and $f(\theta, q)$ is convex in θ for any fixed value of $q \in \mathbb{Q}$.*

Assumption 2.2 (Uniqueness). *If the optimization problem (2.4) is feasible, it admits a unique solution.*

We remark that the uniqueness assumption can be relaxed in most cases by introducing a tie-breaking rule (see Section 4.1 of [26]).

The probabilistic property of the optimal solution obtained from (2.4) are stated in the next lemma taken from [25]. The result was first established in [31] under the additional assumption that the scenario problem is feasible with probability one (in this case n_θ in (2.5) can be replaced by $n_\theta - 1$).

Lemma 2.1. *Let Assumptions 2.1 and 2.2 hold and let δ, $\varepsilon \in (0, 1)$ and N satisfy the following inequality*

$$\sum_{i=0}^{n_\theta} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \leq \delta. \tag{2.5}$$

Then, with probability at least $1-\delta$ either the optimization problem (2.4) is unfeasible or its optimal solution $\widehat{\theta}_N$ satisfies the inequality $V(\widehat{\theta}_N) \leq \varepsilon$.

2.2.2 Scenario with Discarded Constraints

The idea of scenario with discarded constraints [25, 32] is to generate N i.i.d. samples and then purposely discard $r < N - n_\theta$ of them. In other words, we solve the following optimization problem

$$\min_{\theta\in\Theta} \quad c^T\theta \tag{2.6}$$

$$\text{subject to } \ f(\theta, q^{(i)}) \leq 0,\ i = 1, \ldots, N - r.$$

The r discarded samples are chosen so that the largest improvement in the optimal objective value is achieved. We remark that the optimal strategy to select r discarded samples is a mixed-integer optimization problem, which may be hard to solve numerically. The following lemma [25] defines the probabilistic properties of the optimal solution obtained from (2.6).

Lemma 2.2. *Let Assumptions 2.1 and 2.2 hold and let* δ, $\varepsilon \in (0, 1)$, N *and* $r < N - n_\theta$ *satisfy the following inequality*

$$\binom{r+n_\theta}{r} \sum_{i=0}^{r+n_\theta} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \leq \delta. \tag{2.7}$$

Then, with probability at least $1 - \delta$ *either the optimization problem (2.6) is unfeasible or its optimal solution* $\widehat{\theta}_N$ *satisfies the inequality* $V(\widehat{\theta}_N) \leq \varepsilon$.

Note that there exist different results in the literature that derive explicit sample complexity bounds on the N such that (2.5) or (2.7) are satisfied for given values of $\varepsilon, \delta \in (0, 1)$, see e.g. [6] and [25]. These bounds depend linearly on $1/\varepsilon$ and n_θ and logarithmically on $1/\delta$. However, in practice, the required number of samples can be very large even for problems with moderate number of decision variables. Therefore, the computational complexity of the random convex problems (2.4) and

(2.6) might be beyond the capability of the available computational tools. Motivated by this limitation, in the next section we propose two novel sequential randomized algorithms for optimization.

2.3 The Sequential Randomized Algorithms

The main philosophy behind the proposed sequential randomized algorithms lies on the fact that it is easy from the computational point of view to evaluate a given "candidate solution" for a large number of random samples extracted from $\mathbb{Q}$. On the other hand, it is clearly more expensive to solve the optimization problems (2.4) or (2.6) when the sample bound N is large. The sequential randomized algorithms, which are presented next generate a sequence of "design" sample sets $\{q_d^{(1)}, \ldots, q_d^{(N_k)}\}$ with increasing cardinality N_k which are used in (2.4) and (2.6) for solving the optimization problem. In parallel, "validation" sample sets $\{q_v^{(1)}, \ldots, q_v^{(M_k)}\}$ of cardinality M_k are also generated by both algorithms in order to check whether the given candidate solution, obtained from solving (2.4) or (2.6), satisfies the desired violation probability.

The first algorithm is in line with those presented in [29] and [95], in the sense that it uses a similar strategy to validate the candidate solution. However, while these algorithms have been designed for feasibility problems, the proposed algorithms deal with optimization problems. More generally, the two presented algorithms fall into the class of general SPV algorithms studied in [4].

2.3.1 Full Constraint Satisfaction

The first sequential randomized algorithm is presented in Algorithm 2.1, and its theoretical properties are stated in the following theorem.

Theorem 2.1. *Suppose that Assumptions 2.1 and 2.2 hold. If at iteration k Algorithm 2.1 exits with a probabilistic solution θ_{sol}, then it holds that $V(\theta_{\text{sol}}) \leq \varepsilon$ with probability no smaller than $1-\delta$, that is*

$$\Pr\{V(\theta_{\text{sol}}) \leq \varepsilon\} \geq 1-\delta.$$

Proof. See Appendix 2.6.1. □

We note that in steps 3 and 4, to preserve the i.i.d. assumptions, the design and validation samples need to be redrawn at each iteration, and sample-reuse techniques are not applicable.

Remark 2.1 (Optimal Value of α)**.** *The sample bound (2.10) has some similarities with the one derived in [30, Theorem 2], originally proven in [41], and also used in [4]. However, since we are using a finite sum*[1]*, thanks to the finite scenario bound obtained solving (2.8), we can use the finite hyperharmonic series $\mathcal{S}_{k_{t-1}}(\alpha) = \sum_{j=1}^{k_{t-1}} j^{-\alpha}$ (also known as p-series) instead of the Riemann Zeta function $\sum_{j=1}^{\infty} j^{-\alpha}$. Indeed, the Riemann Zeta function does not converge when α is smaller than one, while in the presented bound (2.10) α may be smaller than one, which improves the overall sample complexity in particular for large values of k_t. The optimal value of α which*

[1]See in particular the summation (2.18) in the proof of Theorem 2.1.

Algorithm 2.1 SEQUENTIAL RANDOMIZED ALGORITHM: FULL CONSTRAINT SATISFACTION

1. INITIALIZATION

   Set iteration counter to zero ($k = 0$). Choose probabilistic levels ε, δ and number of iterations $k_t > 1$.

2. UPDATE

   Set $k = k + 1$ and $N_k \geq N\frac{k}{k_t}$ where N is the smallest integer satisfying

$$\sum_{i=0}^{n_\theta} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \leq \delta/2. \tag{2.8}$$

3. DESIGN

   - Draw N_k i.i.d. samples $\mathbf{q}_d = \{q_d^{(1)}, \ldots, q_d^{(N_k)}\} \in \mathbb{Q}$ based on the underlying distribution.
   - Solve the following *reduced-size scenario problem*

$$\begin{aligned} \widehat{\theta}_{N_k} = \arg\min_{\theta \in \Theta} \quad & c^T\theta \\ \text{subject to} \quad & f(\theta, q_d^{(i)}) \leq 0, \quad i = 1, \ldots, N_k. \end{aligned} \tag{2.9}$$

   - **If** the optimization problem (2.9) is not feasible, the original problem (2.1) is not feasible as well.
   - **Else if** the last iteration is reached ($k = k_t$), set $\theta_{\text{sol}} = \widehat{\theta}_{N_k}$ and **Exit**.
   - **Else**, continue to the next step.

4. VALIDATION

- Draw

$$M_k \geq \frac{\alpha \ln k + \ln\left(\mathcal{S}_{k_t-1}(\alpha)\right) + \ln \frac{2}{\delta}}{\ln\left(\frac{1}{1-\varepsilon}\right)} \tag{2.10}$$

  i.i.d. samples $\mathbf{q}_v = \{q_v^{(1)}, \ldots, q_v^{(M_k)}\} \in \mathbb{Q}$ based on the underlying distribution, and $\mathcal{S}_{k_t-1}(\alpha) = \sum_{j=1}^{k_t-1} j^{-\alpha}$, where $\alpha > 0$ is a tuning parameter.

- **If** $\mathbb{I}_f(\widehat{\theta}_{N_k}, q_v^{(i)}) = 0$ for $i = 1, \ldots, M_k$; set $\theta_{\text{sol}} = \widehat{\theta}_{N_k}$ and **Exit**.

- **Else**, goto step (2).

minimizes the sample bound (2.10) has been computed using numerical simulations for different values of the termination parameter k_t. The "almost" optimal value of α minimizing (2.10) for a wide range of k_t is $\alpha = 0.1$. The bound (2.10) (for $\alpha = 0.1$) improves upon the bound (17) in [30], by 5% to 15% depending on the termination parameter k_t. It also improves upon the bound in [95], which uses finite sum but in a less effective way. Finally, we note that the dependence of M_k upon the parameters ε and δ is logarithmic in $1/\delta$ and substantially linear in $1/\varepsilon$.

2.3.2 Partial Constraint Satisfaction

In the "design" and "validation" steps of Algorithm 2.1, *all* elements of the design and validation sample sets are required to satisfy the constraint in (2.1). However, it is sometimes impossible to find a solution satisfying the constraint in (2.1) for the entire set of uncertainty. For this reason, in Algorithm 2.2, we consider the scenario design with discarded constraints where we allow a limited number of design and validation samples to violate the constraint in (2.1).

We now provide a theorem stating the theoretical properties of Algorithm 2.2.

Theorem 2.2. *Suppose that Assumptions 2.1 and 2.2 hold. If at iteration k Algorithm 2.2 exits with a probabilistic solution θ_{sol}, then it holds that $V(\theta_{\text{sol}}) \leq \varepsilon$ with probability no smaller than $1-\delta$*

$$\Pr\left\{V(\theta_{\text{sol}}) \leq \varepsilon\right\} \geq 1-\delta.$$

Proof. See Appendix 2.6.2. □

Algorithm 2.2 is different from the algorithm presented in [5], which was derived for non-convex problems, in a number of aspects. That is, the cardinality of the sequence of sample sets used for design and validation increases linearly with iteration counter k, while it increases exponentially in [5]. Furthermore, the cardinality of the validation sample set at the last iteration M_{k_t} in [5] is chosen to be equal to the cardinality of the sample set used for design at the last iteration N_{k_t} while, in the presented algorithm M_{k_t} and hence β_w are chosen based on the additive Chernoff bound which is less conservative.

We also note that both Algorithms 2.1 and 2.2 fall within the class of SPV algorithms in which the "design" and "validation" steps are independent, see [4]. As a result, in principle we could use the same strategy as Algorithm 2.1 to tackle discarded

Algorithm 2.2 SEQUENTIAL RANDOMIZED ALGORITHM: PARTIAL CONSTRAINT SATISFACTION

1. INITIALIZATION

   Set the iteration counter to zero ($k = 0$). Choose probabilistic levels ε, δ, number of iterations $k_t > 1$, number of discarded constraints r and define the following parameters:

$$\beta_v \doteq \max\left\{1, \beta_w \left(k_t \ln \frac{2k_t}{\delta}\right)^{-1}\right\}, \quad \beta_w \doteq \frac{1}{4\varepsilon} \ln \frac{1}{\delta}. \tag{2.11}$$

2. UPDATE

   Set $k = k + 1$, $N_k \geq N\frac{k}{k_t}$ and $N_{k,r} \geq \frac{(N-r)k}{k_t}$ where N is the smallest integer satisfying

$$\binom{r + n_\theta}{r} \sum_{i=0}^{r+n_\theta} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \leq \delta/2. \tag{2.12}$$

3. DESIGN

   - Draw N_k i.i.d. samples $\mathbf{q}_d = \{q_d^{(1)}, \ldots, q_d^{(N_k)}\} \in \mathbb{Q}$ based on the underlying distribution.
   - Solve the following *reduced-size scenario problem*

$$\begin{aligned} \widehat{\theta}_{N_k,r} = \arg \min_{\theta \in \Theta} & \quad c^T \theta \\ \text{subject to} & \quad f(\theta, q_d^{(i)}) \leq 0, \quad i = 1, \ldots, N_{k,r}. \end{aligned} \tag{2.13}$$

   - **If** the optimization problem (2.13) is not feasible, the original problem (2.1) is not feasible as well.

- **Else if** the last iteration is reached ($k = k_t$), set $\theta_{\text{sol}} = \widehat{\theta}_{N_k,r}$ and **Exit**.

- **Else**, continue to the next step.

1. Validation

- Draw

$$M_k \geq 2k\beta_v \frac{1}{\varepsilon} \ln \frac{2k_t}{\delta} \tag{2.14}$$

i.i.d. samples $\mathbf{q}_v = \{q_v^{(1)}, \ldots, q_v^{(M_k)}\} \in \mathbb{Q}$ based on the underlying distribution.

- **If**

$$\frac{1}{M_k} \sum_{i=1}^{M_k} \mathbb{I}_f(\widehat{\theta}_{N_k,r}, q_v^{(i)}) \leq \left(1 - (k\beta_v)^{-1/2}\right) \varepsilon \tag{2.15}$$

set $\theta_{\text{sol}} = \widehat{\theta}_{N_k,r}$ and **Exit**.

- **Else**, goto step (2).

---

constraints problems. Nevertheless, Algorithm 2.2 appears to be more suitable for discarded constraints problems, since (2.13) forces the solution to violate some constraints.

2.3.3 Algorithms Termination and Overall Sample Complexity

Note that the maximum number of iterations of both Algorithms 1 and 2 is chosen by the user by selecting the termination parameter k_t. This choice affects directly the cardinality of the sample sets used for design N_k and validation M_k at each iteration, although they converge to fixed values (independent of k_t) at the last iteration. In problems for which the original scenario sample complexity is large, we suggest to use larger k_t. In this way, the sequence of sample bounds N_k starts from a smaller number and does not increase significantly with the iteration counter k.

We also remark that, in Algorithm 2.2, the right hand side of the inequality (2.15) cannot be negative, which in turn requires β_v to be greater than one. This condition is taken into account in defining β_v in (2.11). However, we can avoid generating $\beta_v < 1$ by the appropriate k_t. To this end, we solve the inequality $\beta_v \geq 1$ for k_t as follows:

$$\beta_w \left(k_t \ln \frac{2k_t}{\delta} \right)^{-1} \geq 1 \Rightarrow k_t \ln \frac{2k_t}{\delta} \leq \beta_w \Rightarrow \frac{2k_t}{\delta} \ln \frac{2k_t}{\delta} \leq \frac{2\beta_w}{\delta}.$$

For implementation purposes, it is useful to use the function “LambertW” also known as “Omega function” or “product logarithm”[2]

$$k_t \leq \frac{\beta_w}{\mathrm{LambertW}\left(\frac{2\beta_w}{\delta}\right)}.$$

Furthermore, note that the overall sample complexity of Algorithm 2.1 and 2.2 is a random variable, because the number of iterations is random. Indeed, the number of

[2]This function is the inverse function of $f(W) = We^W$. In other words, $W = \mathrm{LambertW} f(W)$; see e.g. [40] for more details.

iterations in which the algorithm terminates (N_k and M_k) is only known *a posteriori*, while in the scenario approach we can establish *a priori* sample bounds. We remark that the computational cost of solving convex optimization problems does not increase linearly with the number of constraints. Hence, we conclude that, if the algorithms terminate with smaller number of design samples than the original sample complexity of the scenario problem, which is the case most of the times, the reduction in the number of design samples can significantly improve the overall computational cost. In the particular case when the constraints are linear matrix inequalities (LMIs), then the reduced-size scenario problem (2.9) can be reformulated as a semidefinite program by combining N_k LMIs into a single LMI with block-diagonal structure. It is known, see [13], that the computational cost of this problem with respect to the number of diagonal blocks N_k is of the order of $N_k^{3/2}$. Similar discussions hold for Algorithm 2.2. We conclude that a decrease in N_k, can significantly reduce the computational complexity.

Finally, note that the computational cost of validation steps in both presented algorithms is not significant since they just require *analysis* of a candidate solution for a number of i.i.d. samples extracted from the uncertainty set.

2.4 Numerical Simulation

In this section, we employ the developed sequential randomized algorithms of Section 2.3 to solve a non-trivial control problem. The plant under consideration is a

Table 2.1: Uncertainty vector q and its nominal value $\overline{q}$

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

multivariable model for the lateral motion of an aircraft. The example is studied in [7] and [113], and it is adopted originally from [117]. The state space description of the model is given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & L_p & L_\beta & L_r \\ g/V & 0 & Y_\beta & -1 \\ N_{\dot\beta}(g/V) & N_p & N_\beta + N_{\dot\beta} Y_\beta & N_r - N_{\dot\beta} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & L_{\delta_a} \\ Y_{\delta_r} & 0 \\ N_{\delta_r} + N_{\dot\beta} Y_{\delta_r} & N_{\delta_a} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.16}$$

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. The objective is to design a state feedback controller of the form $u = Kx$ which, firstly, stabilizes the plant (2.16); secondly, enforces all the eigenvalues of the closed loop system to have real part smaller than $-\alpha$ ($\alpha > 0$) and, finally, we would like the trace of Lyapunov matrix, used for testing the stability of closed loop system, to be minimized. All parameters appearing in the model (2.16) are considered to be uncertain. Representing the plant

Table 2.2: Simulation results obtained using Algorithm 2.1

ε	δ	k_t	Design Samples			Validation Samples			Iteration Number			Computational Time (S)		
			Mean	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*
0.1	10^{-4}	10	211.8	53.4	291	109.9	0.31	110	5.1	1.3	7	10.9	11.2	58
0.05	10^{-5}	12	308.5	134.4	451	272.7	1.25	274	4.1	1.8	6	24.7	25.5	135.1
0.02	10^{-6}	20	1035.2	499.1	1827	832.5	2.5	836	8.5	3.7	15	65.1	56.1	239.1
0.01	10^{-8}	25	2149.8	1045.7	4030	2152.5	4.7	2160	9.6	4.7	18	144.6	124	643.1
0.005	10^{-9}	30	4689	434.9	5166	4813.6	2.2	4816	11.8	1.1	13	325	338.2	1774.7

(2.16) as

$$\dot{x} = A(q)x + B(q)u$$

the uncertainty vector q is of dimension 13 and the uncertainty set $\mathbb{Q}$ is assumed to be a 13 dimensional hyperrectangle centered at the nominal value $\overline{q}$ with radius R i.e.,

$$\mathbb{Q} = \{q_i \in \mathbb{R} : \ q_i \in [(1-R)\overline{q}, (1+R)\overline{q}],\ i = 1, \ldots, 13\}\,.$$

Using the approach of [21], we solve the following optimization problem

$$\begin{aligned} &\underset{P,Y}{\text{minimize}}\ \text{Tr}\ (P) \\ &\text{subject to } A(q)P + PA^T(q) + B(q)Y + Y^T B^T(q) + 2\alpha P \preceq 0 \end{aligned} \tag{2.17}$$

where $P \in \mathbb{R}^{4\times 4}$ is a symmetric positive definite matrix and $Y \in \mathbb{R}^{2\times 4}$ is an auxiliary variable. The feedback gain K can be constructed as $K = YP^{-1}$. We note that the constraint (2.17) is in the form of a linear matrix inequality (LMI) and by introducing the convex function $\lambda_{\max}(.)$, which represents the maximum eigenvalue of the matrix, the problem can be recast in the form (2.1).

Table 2.3: Simulation results obtained using Algorithm 2.2

ε	δ	k_t	*Design Samples*			*Validation Samples*			*Iteration Number*			*Computational Time (S)*		
			Mean	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*
0.1	10^{-4}	10	108.2	23	125	599	125.9	691	2.6	0.54	3	12.8	4.4	25.6
0.05	10^{-5}	12	266	53	301	1680	395.9	2240	3	0.7	4	31	12.5	58.7
0.02	10^{-6}	20	389.8	133.1	487	5380.2	1841.4	6725	3.2	1.1	4	115.7	51.7	301.1
0.01	10^{-8}	25	761.6	122.7	896	14715	2370.5	17312	3.4	0.54	4	324.1	127.6	557.7
0.005	10^{-9}	30	1431.2	466.4	2385	49737	16217	82894	3.6	1.17	6	854	355.5	2170.7

Table 2.4: The scenario bound and the required computational time for the same probabilistic levels as Tables. 2.2 and 2.3

ε	δ	*The Scenario Bound*	*Computational Time (S)*
0.1	1×10^{-4}	414	13
0.05	1×10^{-5}	901	24.6
0.02	1×10^{-6}	2434	108.4
0.01	1×10^{-8}	5596	479.3
0.005	1×10^{-9}	11920	2295

Sequential algorithms of Section 2.3 are implemented in Matlab using the the toolbox Randomized Algorithm Control Toolbox (RACT) [116]. In the simulation, we assumed that the relative uncertainty is 15% ($R = 0.15$) and the probability density function of all uncertain parameters is uniform. The choice of uniform distribution is chosen due to its worst case nature [10]. The vector of nominal values $\overline{q}$ is given in Table 2.1. The optimization problem (2.17) is solved for different values of ε and δ. Furthermore, we run the simulation 100 times for each pair of ε and δ. The mean, standard deviation and worst case values of the number of design samples, optimization samples, the iteration number in which the algorithm exits and the total computational time are tabulated in Tables 2.2 and 2.3[3]. Table 2.4 shows the scenario bound along with the computational time required for solving the random convex problem for the same probabilistic levels as Tables 2.2 and 2.3. It can be seen that using the proposed algorithm, we can achieve the same probabilistic levels with much smaller number of design samples. The computational time of Tables 2.2 and 2.3 is much smaller than Table 2.4 which further proves the effectiveness of the proposed sequential randomized algorithms. We note that the number of validation samples in Table 2.3 is bigger than the one in Table 2.2 which means Algorithm 2.2 needs more validation samples to verify the feasibility of the candidate solution. The increase in the number of validation samples in Algorithm 2.2 has two reasons: i) since we allow a number of validation samples to violate the constraint in (2.1), the algorithm requires more samples to be checked; and ii) the bound M_k in Algorithm 2.2 is purely based on Chernoff inequalities which is more conservative than log-over-log bounds.

2.5 Conclusions

We proposed two new sequential methods for solving in a computational efficient way uncertain convex optimization problems. The main philosophy behind the proposed sequential randomized algorithms stems from the consideration that it is easy,

[3] All the simulations are carried on a work station with 2.83 Giga Hertz Core 2 Quad CPU and 8 Giga Byte RAM.

from a computational viewpoint, to validate a given “candidate solution” for a large number of random samples. The algorithms have been tested on a numerical example, and extensive numerical simulations show how the total computational effort is “diluted” by applying the proposed sequential methodology.

2.6 Appendix

2.6.1 Proof of the Theorem 2.1

Following the same reasoning as in [95], we introduce the events

$$\begin{aligned}
&\mathrm{Iter}_k \doteq \{\text{the } k\text{th outer iteration is reached}\},\\
&\mathrm{Feas}_k \doteq \{\widehat{\theta}_{N_k} \text{ is declared as feasible in the ``validation'' step}\},\\
&\mathrm{Bad}_k \doteq \{V(\widehat{\theta}_{N_k}) > \varepsilon\},\\
&\mathrm{ExitBad}_k \doteq \{\text{Algorithm 2.1 exits at iteration } k \cap \mathrm{Bad}_k\},\\
&\mathrm{ExitBad} \doteq \{\text{Algorithm 2.1 exits at some unspecified}\\
&\qquad\qquad\qquad \text{iteration } k \cap \mathrm{Bad}_k\}.
\end{aligned}$$

The goal is to bound the probability of the event "ExitBad". Since $\mathrm{ExitBad}_i \cap \mathrm{ExitBad}_j = \emptyset$ for $i \neq j$, the probability of the event "ExitBad" can be reformulated in terms of the event "$\mathrm{ExitBad}_k$" as

$$\begin{aligned}
\Pr\{\mathrm{ExitBad}\} &= \Pr\{\mathrm{ExitBad}_1 \cup \mathrm{ExitBad}_2 \cup \cdots \cup \mathrm{ExitBad}_{k_t}\}\\
&= \Pr\{\mathrm{ExitBad}_1\} + \Pr\{\mathrm{ExitBad}_2\} + \cdots +\\
&\quad \Pr\{\mathrm{ExitBad}_{k_t-1}\} + \Pr\{\mathrm{ExitBad}_{k_t}\}. \qquad (2.18)
\end{aligned}$$

From the definition of the event "$\mathrm{ExitBad}_k$" and by considering that to exit at iteration $k \leq k_{t-1}$, the algorithm needs i) to reach kth iteration and ii) to declare $\widehat{\theta}_{N_k}$

feasible in the validation step, for $k = 1, \dots, k_{t-1}$, we arrive at

$$\begin{aligned}
\Pr\{\text{ExitBad}_k\} &= \Pr\{\text{Feas}_k \cap \text{Bad}_k \cap \text{Iter}_k\} \\
&= \Pr\{\text{Feas}_k \cap \text{Bad}_k \mid \text{Iter}_k\} \Pr\{\text{Iter}_k\} \\
&\le \Pr\{\text{Feas}_k \cap \text{Bad}_k \mid \text{Iter}_k\} \\
&= \Pr\{\text{Feas}_k \mid \text{Bad}_k \cap \text{Iter}_k\} \Pr\{\text{Bad}_k \mid \text{Iter}_k\} \\
&\le \Pr\{\text{Feas}_k \mid \text{Bad}_k \cap \text{Iter}_k\}. \qquad (2.19)
\end{aligned}$$

Using the result of Theorem 1 in [29], we can bound the right hand side of (2.19)

$$\Pr\{\text{Feas}_k \mid \text{Bad}_k \cap \text{Iter}_k\} < (1-\varepsilon)^{M_k}. \qquad (2.20)$$

Combining (2.18) and (2.20) results in

$$\begin{aligned}
\Pr\{\text{ExitBad}\} &< (1-\varepsilon)^{M_1} + (1-\varepsilon)^{M_2} + \cdots + (1-\varepsilon)^{M_{k_t-1}} \\
&+ \Pr\{\text{ExitBad}_{k_t}\} = \sum_{k=1}^{k_t-1} (1-\varepsilon)^{M_k} + \Pr\{\text{ExitBad}_{k_t}\}. \qquad (2.21)
\end{aligned}$$

The summation in (2.21) can be made arbitrary small by an appropriate choice of M_k. In particular, by choosing

$$(1-\varepsilon)^{M_k} = \frac{1}{k^\alpha} \frac{1}{\mathcal{S}_{k_t-1}(\alpha)} \frac{\delta}{2}, \qquad (2.22)$$

we have

$$\sum_{k=1}^{k_t-1} (1-\varepsilon)^{M_k} = \sum_{k=1}^{k_t-1} \frac{1}{k^\alpha} \frac{1}{\mathcal{S}_{k_t-1}(\alpha)} \frac{\delta}{2} = \frac{\delta}{2}. \qquad (2.23)$$

Note that the choice of the number of design samples in the last iteration guarantees that $\Pr\{\text{ExitBad}_{k_t}\} \le \delta/2$. The statement follows, combining (2.21) with (2.23) and noting that the bound (2.10) is obtained solving (2.22) for M_k.

2.6.2 Proof of the Theorem 2.2

To prove the statement, define the events Iter_k, Feas_k, Bad_k, $\mathrm{ExitBad}_k$ and ExitBad as in the proof of Theorem 2.1. Then, note that the event Feas_k can be written as follows

$$\mathrm{Feas}_k = \left\{ \widehat{V}(\widehat{\theta}_{N_{k,r}}, \mathbf{q}_v) \leq \left(1 - (k\beta_v)^{-1/2}\right) \varepsilon \right\},$$

that is, $\widehat{\theta}_{N_k}$ is declared feasible whenever the feasibility test (2.15) is passed. Again, the goal is to bound the probability of the event "ExitBad", which can be written as the summation of the events "$\mathrm{ExitBad}_k$" as in (2.18). In turn, for $k \leq k_{t-1}$, we can write

$$\Pr\{\mathrm{ExitBad}_k\} \leq \Pr\{\mathrm{Feas}_k \cap \mathrm{Bad}_k \mid \mathrm{Iter}_k\} \doteq \Pr\{\mathrm{MisClass}_k\},$$

where we denoted $\mathrm{MisClass}_k$ the event of misclassification at iteration k.

$$\mathrm{MisClass}_k = \left\{ \widehat{V}(\widehat{\theta}_{N_{k,r}}, \mathbf{q}_v) \leq \left(1 - (k\beta_v)^{-1/2}\right) \varepsilon \right\} \cap \left\{ V(\widehat{\theta}_{N_{k,r}}) > \varepsilon \right\}, \quad k = 1, \ldots, k_{t-1}.$$

By defining $\rho_k \doteq \left(1 - (k\beta_v)^{-1/2}\right) \varepsilon$ and $\varepsilon_k \doteq (k\beta_v)^{-1/2}\varepsilon$, this event can be rewritten as

$$\mathrm{MisClass}_k = \left\{ \widehat{V}(\widehat{\theta}_{N_{k,r}}, \mathbf{q}_v) \leq \rho_k \right\} \cap \left\{ V(\widehat{\theta}_{N_{k,r}}) - \widehat{V}(\widehat{\theta}_{N_{k,r}}, \mathbf{q}_v) > \varepsilon_k \right\}, \quad k = 1, \ldots, k_{t-1}.$$

Applying the results of [5, Theorem 1], we can bound this event as follows

$$\Pr\left\{\mathrm{MisClass}_k\right\} \leq \Pr\left\{ \frac{V(\widehat{\theta}_{N_{k,r}}) - \widehat{V}(\widehat{\theta}_{N_{k,r}}, \mathbf{q}_v)}{\sqrt{V(\widehat{\theta}_{N_{k,r}})}} > \frac{\varepsilon_k}{\sqrt{\varepsilon_k + \rho_k}} \right\}. \tag{2.24}$$

For any $\eta \in (0,1)$, the one-sided multiplicative Chernoff inequality [113] guarantees that

$$\Pr\{V(\widehat{\theta}_{N_{k,r}}) - \widehat{V}(\widehat{\theta}_{N_{k,r}}, \mathbf{q}_v) \geq \eta V(\widehat{\theta}_{N_{k,r}})\} \leq e^{\frac{-V(\widehat{\theta}_{N_{k,r}})M_k\eta^2}{2}}. \tag{2.25}$$

Setting $\eta = \frac{\varepsilon_k}{\sqrt{\varepsilon_k+\rho_k}}\frac{1}{\sqrt{V(\widehat{\theta}_{N_{k,r}})}}$ in (2.25), combining with inequality (2.24), we obtain, for $k = 1, \ldots, k_{t-1}$

$$\Pr\left\{\mathrm{MisClass}_k\right\} \leq e^{\frac{-\varepsilon_k^2 M_k}{2(\varepsilon_k+\rho_k)}} \leq \frac{\delta}{2k_t}$$

where the last inequality follows from the choice of M_k in (2.14). Notice also that the choice of the number of design samples at the last iteration N_{k_t} guarantees that the probability of misclassification at the last iteration ($k = k_t$) is at most $\delta/2$. Therefore, we can write

$$\begin{aligned}
\Pr\{\mathrm{ExitBad}\} &\leq \sum_{k=1}^{k_t} \Pr\{\mathrm{MisClass}_k\} \\
&\leq \sum_{k=1}^{k_t-1} \frac{\delta}{2k_t} + \Pr\{\mathrm{MisClass}_{k_t}\} = \frac{\delta(k_t-1)}{2k_t} + \frac{\delta}{2} \leq \delta
\end{aligned}$$

which proves the statement.

Chapter 3

A Statistical Learning Theory Approach to Uncertain LMI and BMI

3.1 Introduction

Statistical learning theory is a very effective tool in dealing with various applications, which include neural networks and control systems, see for instance [126]. The main objective of this theory is to extend convergence properties of the empirical mean, which can be computed with a Monte Carlo simulation, from finite families to infinite families of functions. For finite families, these properties can be easily established by means of a repeated application of the so-called Hoeffding inequality, and are related to the well-known law of large numbers, see for instance [113]. On the other hand, for infinite families much deeper technical tools are needed and have been developed in the seminal work of Vapnik and Chervonenkis [123]. In this case, the main issue is to establish *uniform* convergence of empirical means. In particular, this requires to establish whether or not a combinatorial parameter called the Vapnik-Chervonenkis dimension (VC-dimension) is finite, see [122].

Subsequent contributions on statistical learning theory by Vidyasagar [125] followed two main research directions: First, to demonstrate that this theory is indeed an effective tool for control of systems affected by uncertainty. Second, to "invert" the bounds provided by Vapnik and Chervonenkis, introducing the concept of *sample complexity*. Roughly speaking, when dealing with control of uncertain systems, the sample complexity provides the number of random samples of the uncertainty that should be drawn to derive a stabilizing controller (or a controller which attains a given $\mathcal{H}_\infty$-norm bound on the closed-loop sensitivity function), with sufficiently high probabilistic accuracy and confidence. Since the sample complexity is a function of the accuracy, confidence and the VC-dimension, specific bounds on this combinatorial parameter should be derived. In turn, this involves a problem reformulation in terms of Boolean functions, and the evaluation of the number of required polynomial inequalities, their order and the number of design variables.

For various stabilization problems, which include stability of interval matrices and simultaneous stabilization with static output feedback, bounds on the VC-dimension have been derived in [127]. In this chapter, we continue this specific line of research, and we compute the VC-dimension for control problems formulated in terms of *uncertain* linear matrix inequalities (LMIs) and bilinear matrix inequalities (BMIs). It is well-known that many robust and optimal control problems can be indeed formulated in these forms, see for instance [21, 54, 74, 119]. However, due to the presence of uncertainty it is often unclear how uncertain LMIs and BMIs can be effectively solved, for example when the uncertainty enters nonlinearly into the control system. In these cases, relaxation techniques are usually introduced, leading to conservative results.

We provide new bounds for the VC-dimension for uncertain LMIs and BMIs. These bounds are then combined with recent results in [5] to establish the sample complexity of uncertain LMIs and BMIs. We remark that the sample complexity is independent from the number of uncertain parameters entering into the LMIs and BMIs, and on their functional relationship. Hence, the related randomized algorithms run in polynomial-time. However, for relatively small values of the probabilistic accuracy and confidence, the sample complexity turns out to be very large, as usual in the context of statistical learning theory. For this reason, randomized algorithms based on a direct application of these bounds may be of limited use in practice. To alleviate this difficulty, in the second part of the chapter we propose a sequential algorithm specifically tailored to the problem at hand. This algorithm has some similarities with other sequential algorithms previously developed for other problems in the area of randomized algorithms for control of uncertain systems, see [4, 5, 82] and in particular sequential randomized methods of Chapter 2 (Algorithms 2.1 and 2.2).

Finally, the effectiveness of this approach is shown by a numerical example related to the static output feedback stabilization of an uncertain robot manipulator joint. In particular, the objective is to design a static output feedback controller which minimizes the worst-case $\mathcal{H}_\infty$ norm. The numerical performance of the proposed sequential algorithm is evaluated and compared with the theoretical sample-complexity previously derived.

3.2 Problem Formulation

We now formally state the uncertain LMI and BMI problems discussed in the Introduction.

Problem 3.1 (Uncertain strict LMI optimization)**.** *Find the optimal value of x, if it exists, which solves the optimization problem*

$$\begin{aligned} &\underset{x}{\textit{minimize}}\ \ c_x^T x \\ &\textit{subject to}\ \ F_{LMI}(x,q) \doteq F_0(q) + \sum_{i=1}^{m_x} x_i F_i(q) \succ 0,\ \forall q \in \mathbb{Q} \end{aligned} \tag{3.1}$$

where $x \in \mathbb{R}^{m_x}$ is the vector of optimization variables, $q \in \mathbb{Q} \subset \mathbb{R}^\ell$ is the vector of uncertain parameters bounded in the set $\mathbb{Q}$ and $F_i = F_i^T \in \mathbb{R}^{n\times n}$, $i = 0, \ldots, m_x$. The inequality $F_{LMI}(x,q) \succ 0$ means that $F_{LMI}(x,q)$ is positive definite.

Problem 3.2 (Uncertain strict BMI optimization)**.** *Find the optimal values of x and y, if they exist, which solve the optimization problem*

$$
\begin{aligned}
&\underset{x,y}{\text{minimize}} \quad c_x^T x + c_y^T y \\
&\text{subject to} \quad F_{BMI}(x,y,q) \doteq F_0(q) + \sum_{i=1}^{m_x} x_i F_i(q) \\
&+ \sum_{j=1}^{m_y} y_j G_j(q) + \sum_{i=1}^{m_x} \sum_{j=1}^{m_y} x_i y_j H_{ij}(q) \succ 0, \ \forall q \in \mathbb{Q}
\end{aligned} \tag{3.2}
$$

where $x \in \mathbb{R}^{m_x}$ and $y \in \mathbb{R}^{m_y}$ are the vectors of optimization variables, $q \in \mathbb{Q} \subset \mathbb{R}^\ell$ is the vector of uncertain parameters, $F_0 = F_0^T \in \mathbb{R}^{n \times n}$, and $F_i = F_i^T \in \mathbb{R}^{n \times n}$, $G_j = G_j^T \in \mathbb{R}^{n \times n}$, $H_{ij} = H_{ij}^T \in \mathbb{R}^{n \times n}$, $i = 1, \ldots, m_x$, $j = 1, \ldots, m_y$.

In order to allow a unified treatment of Problems 1 and 2, we now formally define the design parameters for LMIs and BMIs.

Definition 3.1 (Design parameters for LMIs/BMIs)**.** *For Problem 1, we define*

$$\theta \doteq x, \quad m_\theta = m_x, \quad c_\theta = c_x;$$

and, for Problem 2, we define

$$\theta \doteq \begin{bmatrix} x \\ y \end{bmatrix}, \quad m_\theta = m_x + m_y, \quad c_\theta = \begin{bmatrix} c_x \\ c_y \end{bmatrix}.$$

Next, we assume that q is a random variable and a probability measure $\Pr_q$ over the Borel σ-algebra of $\mathbb{Q} \subset \mathbb{R}^\ell$ is given. Then, the constraints in (3.1) and (3.2) become chance-constraints, see e.g. [118], which may be violated for some $q \in \mathbb{Q}$. This concept is formally expressed using the notion of "probability of violation".

Definition 3.2 (Probability of violation)**.** *The probability of violation of θ for the binary-valued function $g : \mathbb{R}^{m_\theta} \times \mathbb{Q} \to \{0, 1\}$ is defined as*

$$V_g(\theta) \doteq \Pr_q \{q \in \mathbb{Q} \ : \ g(\theta, q) = 1\} \tag{3.3}$$

where, for Problem 3.1,

$$g(\theta, q) \doteq \begin{cases} 0 & \mathit{if}\, F_{LMI}(\theta, q) \succ 0 \\ 1 & \mathit{otherwise} \end{cases} \tag{3.4}$$

and, for Problem 3.2,

$$g(\theta, q) \doteq \begin{cases} 0 & \mathit{if}\, F_{BMI}(\theta, q) \succ 0 \\ 1 & \mathit{otherwise} \end{cases} . \tag{3.5}$$

We remark that the probability of violation is in general very hard to evaluate, due to the difficulty of computing the multiple integrals associated with (3.3). Nevertheless, we can "estimate" this probability using randomization. To this end, we extract N independent identically distributed (i.i.d) samples from the set $\mathbb{Q}$

$$\mathbf{q} = \{q^{(1)}, \ldots, q^{(N)}\} \in \mathbb{Q}^N,$$

according to the measure $\Pr_q$, where $\mathbb{Q}^N \doteq \mathbb{Q} \times \mathbb{Q} \times \cdots \times \mathbb{Q}$ (N times). Next, a Monte Carlo approach is employed to obtain the so called "empirical violation"; see e.g. [126].

Definition 3.3 (Empirical violation). *For given $\theta \in \mathbb{R}^{m_\theta}$ the empirical violation of $g(\theta, q)$ with respect to the multisample $\mathbf{q} = \{q^{(1)}, \ldots, q^{(N)}\}$ is defined as*

$$\widehat{V}_g(\theta, \mathbf{q}) \doteq \frac{1}{N} \sum_{i=1}^{N} g(\theta, q^{(i)}). \tag{3.6}$$

3.2.1 Randomized Strategy to Optimization Problems

There are several randomized methodologies in the literature which are based on randomization in the uncertainty space, design parameter space or both. For example, in [125] randomization in both uncertainty and design parameter spaces is employed for minimizing the empirical mean. Similarly, a bootstrap learning method and a min-max approach are presented in [82] and [51], respectively, but these papers deal with finite families. In [5] the authors proposed a randomized algorithm for infinite families which is applicable to convex and non-convex problems. Finally, a non-sequential randomized methodology for uncertain convex problems is introduced in [26, 27, 31]. In Algorithm 3.1 we present a non-sequential randomized strategy for solving Problems 3.1 and 3.2.

We remark that introducing the level parameter $\rho > 0$ enables us to handle probabilistic (soft) constraints, in the same spirit of [5]. The main objective of the present chapter is to derive the explicit sample complexity bound on N based on statistical learning theory results. Finally, we remark that, in the case of LMI constraints, problem (3.7) is a semidefinite optimization problem (SDP) that can be solved efficiently, see [121] and [115] for a discussion on the numerical aspects of solving SDP problems.

Algorithm 3.1 A RANDOMIZED STRATEGY FOR UNCERTAIN LMIS/BMIS

- Given the underlying probability density function (pdf) over the uncertainty set $\mathbb{Q}$ and the level parameter $\rho \in [0, 1)$, extract N independent identically distributed samples from $\mathbb{Q}$ based on the underlying pdf

$$\mathbf{q} = \{q^{(1)}, \ldots, q^{(N)}\}.$$

- Find the optimal value, if it exists, of the following optimization problem

$$\begin{aligned} &\underset{\theta}{\text{minimize}} \quad c_\theta^T \theta \\ &\text{subject to} \quad \widehat{V}_g(\theta, \mathbf{q}) \leq \rho \end{aligned} \tag{3.7}$$

In the case of BMI constraints, efficient solvers such as PENBMI [80] are available, but global solutions to the optimization problem in general cannot be obtained.

3.3 Vapnik-Chervonenkis Theory

In this section, we give a very brief overview of the Vapnik-Chervonenkis theory. The material presented is classical, but a summary is instrumental to our next developments. In particular, we review some bounding inequalities which are used in the subsequent sections to derive the explicit sample bounds for solving Problems 3.1 and 3.2.

Definition 3.4 (Probability of two-sided failure). *Given* N, $\varepsilon \in (0, 1)$ *and* $g : \mathbb{R}^{m_\theta} \times$

$\mathbb{Q} \to \{0,1\}$*, the probability of two-sided failure denoted by* $q_g(N,\varepsilon)$ *is defined as*

$$q_g(N,\varepsilon) \doteq \Pr_q\left\{\mathbf{q} \in \mathbb{Q}^N : \sup_{\theta\in\mathbb{R}^{m_\theta}} |V_g(\theta) - \widehat{V}_g(\theta,\mathbf{q})| > \varepsilon\right\}.$$

The probability of two-sided failure determines how close the empirical violation is to the true probability of violation. In other words, if we extract a multisample $\mathbf{q}$ with cardinality N from the uncertainty set $\mathbb{Q}$, we guarantee that the empirical violation (3.6) is within ε of the true probability of violation (3.3) for all $q \in \mathbb{Q}$ except for a subset having probability measure at most $q_g(N,\varepsilon)$. The parameter $\varepsilon \in (0,1)$ is called accuracy.

Let $\mathcal{G}$ denote the family of functions $\{g(\theta,q) : \theta \in \mathbb{R}^{m_\theta}\}$ where $g : \mathbb{R}^{m_\theta} \times \mathbb{Q} \to \{0,1\}$ is defined in (3.4) or in (3.5). The family $\mathcal{G}$ is said to satisfy the property of *uniform convergence of empirical mean* (UCEM) if $q_g(N,\varepsilon) \to 0$ as $N \to \infty$ for any $\varepsilon \in (0,1)$. We remark that if $\mathcal{G}$ includes finite family of functions, it indeed has the UCEM property. However, infinite families do not necessarily enjoy the UCEM property, which means even if we extract infinite number of samples from the uncertainty set and form the empirical violation, the probability of two-sided failure doesnot go to zero, see [126] for several examples of this type. Problems 3.1 and 3.2 belong to the class of infinite family of functions.

We define the family $\mathcal{S}_g$ containing all possible sets $S_g \doteq \{q \in \mathbb{Q} : g(\theta,q) = 1\}$, for g varying in $\mathcal{G}$. Now consider a multisample $\mathbf{q} = \{q^{(1)},\ldots,q^{(N)}\}$ of cardinality N. For the family of functions $\mathcal{G}$, let

$$\mathbb{N}_{\mathcal{G}}(\mathbf{q}) \doteq \mathrm{Card}\left(\mathbf{q} \cap S_g, S_g \in \mathcal{S}_g\right).$$

In words, we say that $\mathcal{S}_g$ "shatters" $\mathbf{q}$ when $\mathbb{N}_{\mathcal{G}}$ is equal to 2^N. The notion of "shatter coefficient", also known as "growth function", is now defined formally.

Definition 3.5 (Shatter Coefficient)**.** *The shatter coefficient of the family $\mathcal{G}$, denoted by $\mathbb{S}_{\mathcal{G}}(N)$, is defined as*

$$\mathbb{S}_{\mathcal{G}}(N) \doteq \max_{\mathbf{q}\in\mathbb{Q}^N} \mathbb{N}_{\mathcal{G}}(\mathbf{q}).$$

A bound on the shatter coefficient can be obtained by Sauer lemma [104], which in turn requires the computation of the VC-dimension, defined next.

Definition 3.6 (VC-dimension)**.** *The* VC*-dimension of the family of functions $\mathcal{G}$ is defined as the largest integer d for which $\mathbb{S}_{\mathcal{G}}(N) = 2^d$.*

The following result establishes a bound on the probability of two-sided failure in terms of VC-dimension.

Theorem 3.1 (Vapnik and Chervonenkis)**.** *Let d denote the* VC*-dimension of the family of functions $\mathcal{G}$. Then, for any $\varepsilon \in (0, 1)$*

$$q_g(N, \varepsilon) \leq 4e^{2\varepsilon} \left(\frac{2eN}{d}\right)^d e^{-N\varepsilon^2} \tag{3.8}$$

where e is the Euler number.

This result has been proven in [123] and it is stated in [122, Theorem 4.4].

3.4 Main Results

In view of Theorem 3.1, we conclude that families with finite VC-dimension $d < \infty$ enjoy the UCEM property. Hence, it is important i) to show that the collection $\mathcal{G}$ of functions has finite VC-dimension and, ii) to derive upper bounds on the VC-dimension.

3.4.1 Computation of Vapnik-Chervonenkis Dimension

In the next theorem, which is one of the main contributions of this chapter, we derive an upper bound on the VC-dimension of the uncertain LMI and BMI in Problems 3.1 and 3.2.

Theorem 3.2 (VC bounds for strict LMIs/BMIs)**.** *Consider the notation introduced in Definition 3.1. Then, the* VC*-dimension of uncertain LMIs and BMIs (Problems 3.1 and 3.2) is upper bounded by*

$$d \leq 2m_\theta \lg(4en^2) \tag{3.9}$$

where lg(.) *denotes the logarithm to the base* 2*.*

Proof See Appendix 3.9.1.

It is interesting to observe that the VC-dimension of uncertain LMIs and BMIs is linear in the number of design variables m_θ. In the next subsection, we derive explicit sample bounds to be used in Algorithm 3.1 for solving Problems 3.1 and 3.2.

3.4.2 Sample Complexity Bounds

In this section, we study a number of sample bounds guaranteeing that the probability of failures is bounded by a confidence parameter $\delta \in (0,1)$. We remark that there are several results in the literature to derive sample complexity bounds. To the best of our knowledge, the least conservative is stated in Corollary 3 in [5]. For given $\varepsilon, \delta \in (0,1)$, the probability of two-sided failure (see Definition 3.4) is bounded by δ provided that at least

$$N \geq \frac{1.2}{\varepsilon^2}\left(\ln \frac{4e^{2\varepsilon}}{\delta} + d \ln \frac{12}{\varepsilon^2}\right) \tag{3.10}$$

samples are drawn, where $d < \infty$ denotes the VC-dimension of the family of functions $\mathcal{G}$, and ln is the natural logarithm. This result is exploited in the next corollary, that provides the explicit sample complexity bound for the probability of two-sided failure.

Corollary 3.1. *Consider the notation introduced in Definition 3.1, and suppose that* $\varepsilon, \delta \in (0,1)$ *are given. Then, the probability of two-sided failure is bounded by* δ *if at least*

$$N \geq \frac{1.2}{\varepsilon^2}\left(\ln \frac{4e^{2\varepsilon}}{\delta} + 2m_\theta \lg(4en^2) \ln \frac{12}{\varepsilon^2}\right) \tag{3.11}$$

samples are drawn for the Problems 3.1 and 3.2.

Proof The statement of Corollary 3.1 follows immediately by combining (3.10) and the results of Theorem 3.2.

A weaker notion than the probability of two-sided failure is the "probability of one-sided constrained failure" introduced in the following definition.

Definition 3.7 (Probability of one-sided constrained failure)**.** *Given N, $\varepsilon \in (0,1)$, $\rho \in [0,1)$ and $g : \mathbb{R}^{m_\theta} \times \mathbb{Q} \to \{0,1\}$, the probability of one-sided constrained failure, denoted by $p_g(N, \varepsilon, \rho)$, is defined as*

$$p_g(N,\varepsilon,\rho) \doteq \mathrm{Pr}_q \Bigg\{ \mathbf{q} \in \mathbb{Q}^N : \text{ there exists } \theta \in \mathbb{R}^{m_\theta} \\ \text{such that } \widehat{V}_g(\theta, \mathbf{q}) \leq \rho \text{ and } V_g(\theta) - \widehat{V}_g(\theta, \mathbf{q}) > \varepsilon \Bigg\}.$$

Following the same lines of Corollary 3.1, sample complexity bounds for the probability of one-sided constrained failure are derived.

Corollary 3.2. *Consider the notation introduced in Definition 3.1, and suppose that $\varepsilon \in (0,1)$, $\delta \in (0,1)$ and $\rho \in [0,1)$ are given. Then, the probability of one-sided constrained failure is bounded by δ if at least*

$$N \geq \frac{5(\rho+\varepsilon)}{\varepsilon^2} \left(\ln \frac{4}{\delta} + 2m_\theta \lg(4en^2) \ln \frac{40(\rho+\varepsilon)}{\varepsilon^2} \right) \tag{3.12}$$

samples are drawn for the Problems 3.1 and 3.2.

Proof This result is an immediate consequence of Theorem 7 in [5], which states that, for given $\varepsilon, \delta \in (0,1)$ and $\rho \in [0,1)$, the probability of one-sided constrained failure is bounded by δ provided that at least

$$N \geq \frac{5(\rho+\varepsilon)}{\varepsilon^2} \left(\ln \frac{4}{\delta} + d \ln \frac{40(\rho+\varepsilon)}{\varepsilon^2} \right) \tag{3.13}$$

samples are drawn, where $d < \infty$ denotes the VC-dimension of the family of functions $\mathcal{G}$. The statement in Corollary 3.2 is derived by substituting the results of Theorem 3.2 into (3.13).

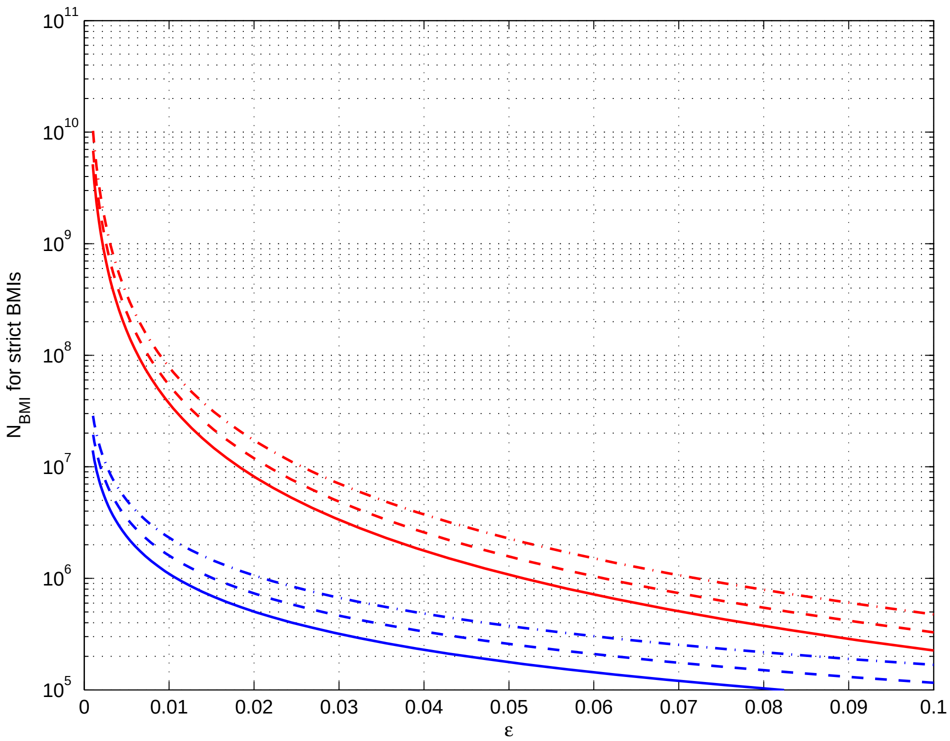

Figure 3.1: Sample complexity bounds for strict BMIs, for $\delta = 1\times10^{-8}$, $m_x+m_y = 13$, and for different BMI dimensions: $n = 10$ (continuous line) $n = 50$ (dashed line) and $n = 100$ (dash-dotted line). The red plots show the two-sided bound (3.11), while the blue plots show the one-sided constrained failure bound (3.12) for $\rho = 0$.

Note that the sample complexity of Corollary 3.2 improves upon that of Corollary 3.1, as shown in Figure 3.1. In particular, it is clear that the bound (3.11) grows as $\mathcal{O}(\frac{1}{\varepsilon^2}\ln\frac{1}{\varepsilon^2})$, which implies that if the accuracy level ε is chosen to be very small, the sample bounds can be very large, while (3.12) grows as $\mathcal{O}(\frac{1}{\varepsilon}\ln\frac{1}{\varepsilon})$.

3.5 Semidefinite Constraints

In this section, we compute upper bounds on the VC-dimension of the semidefinite versions of Problems 3.1 and 3.2 where strict inequalities ($\succ 0$) are replaced with nonstrict inequalities ($\succcurlyeq 0$)[1]. Semidefinite constraints appear in some control problems such as dissipativity; furthermore, some modeling languages such as YALMIP [86] treat strict inequalities using nonstrict ones by adding a slight perturbation. Hence, it is important to derive sample complexities for uncertain semidefinite LMI and BMI problems.

In the following theorem, we establish upper bounds on the VC-dimension of uncertain semidefinite LMI and BMI problems. The proof of this result is reported in Appendix 3.9.2.

Theorem 3.3 (VC bounds for nonstrict LMIs/BMIs)**.** *Consider the notation introduced in Definition 3.1. Then, the* VC*-dimension of uncertain semidefinite LMI and BMI problems is upper bounded by*

$$d \leq 2m_\theta \lg(4en2^n).$$

Remark 3.1 (Strict and nonstrict LMIs/BMIs)**.** *Comparing the bounds of Theorems 3.2 and 3.3, it can be seen that the bounds on the VC-dimension of strict and nonstrict LMIs/BMIs differ only in the terms n^2 and $n2^n$ appearing in the arguments of the logarithm. That is, the quadratic dependence on n of strict LMIs/BMIs becomes*

[1] Throughout the chapter, nonstrict (semidefinite) versions of Problems 3.1 and 3.2 are called "uncertain semidefinite LMI problem" and "uncertain semidefinite BMI problem" respectively.

exponential for nonstrict ones. Note however that this effect is largely mitigated by the logarithm. This difference is not surprising, and it follows from the fact that checking positive semi-definiteness requires non-negativity of all principle minors, as discussed in Appendix 3.9.2. To show this fact, consider the matrix

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

introduced in [16]. This matrix has leading principal minors equal to 1, 0 *and* 0, *which are nonnegative, but it is not positive semidefinite, because its eigenvalues are* 2.732, 0, *and* -0.732. *Note that the same issue arises in [127, Theorem 4], regarding positive definiteness and semi-definiteness of interval matrices.*

Remark 3.2 (Explicit sample complexity for nonstrict LMIs/BMIs)**.** *Using the results of Theorem 3.3, we can establish bounds on sample complexity which guarantee the probability of two-sided failure and the probability of one-sided constrained failure of uncertain semidefinite LMI and BMI problems to be bounded by the confidence parameter δ. It should be noted that for semidefinite problems of this section, Definition 3.2 is revised accordingly such that strict inequalities in (3.4) and (3.5) are replaced with nonstrict ones. This also affects empirical violation, probability of two-sided failure and probability of one-sided constrained failure.*

Then, the results of Corollaries 3.1 and 3.2 for the uncertain semidefinite LMI and BMI problems immediately hold provided that the VC-dimension bound $2m_\theta \lg(4en^2)$

is replaced by $2m_\theta \lg(4en2^n)$. *The sample complexity bounds for semidefinite BMIs are illustrated in Figure 3.2.*

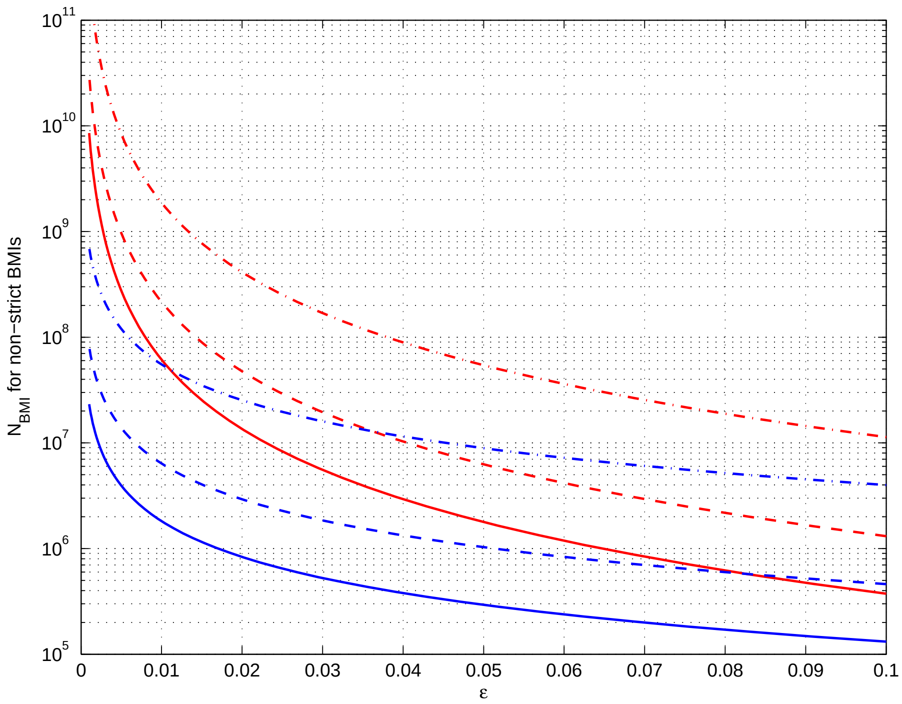

Figure 3.2: Sample complexity bounds for nonstrict BMIs, for $\delta = 1\times10^{-8}$, $m_x+m_y = 13$, and for different BMI dimensions: $n = 10$ (continuous line) $n = 50$ (dashed line) and $n = 100$ (dash-dotted line). The red plots show the two-sided bound, while the blue plots show the one-sided constrained failure bound for $\rho = 0$.

It should be also noted that the sample complexity bounds derived in this chapter for the uncertain strict and semidefinite LMI and BMI problems can be quite large. This is a usual situation in the context of statistical learning, that may lead to computationally expensive optimization problems if Algorithm 3.1 is applied in one-shot. This motivates the developments of the next section, where a *sequential* randomized algorithm for bounding the probability of one-sided constrained failure is presented.

The sequential algorithm can alleviate the computational burden of directly solving (3.7).

3.6 Sequential Randomized Algorithm

Sequential methods in probabilistic design usually follow an iterative scheme which includes optimization steps to update the design parameters, followed by randomization steps to check the feasibility of the candidate solution [113]. The first step is deterministic, while the second one is probabilistic. Examples of such scheme are probabilistic design methods based on gradient [101, 24], ellipsoid [73, 95] and cutting plane [29] update rules, see [113] for more details.

Recently [4] introduced a general framework for nonconvex problems, defining the class of sequential probabilistic validation (SPV) algorithms. In this section we propose a sequential randomized algorithm specifically tailored for the problem at hand, which mitigates the conservatism of the bound (3.12) or its corresponding sample bound for the uncertain semidefinite LMI and BMI problems. This is accomplished by generating a sequence of "design" sample sets $\{q_d^{(1)}, \ldots, q_d^{(N_k)}\}$ with increasing cardinality N_k which are used in (3.7) for solving the optimization problem. In parallel, "validation" sample sets $\{q_v^{(1)}, \ldots, q_v^{(M_k)}\}$ of cardinality M_k are also generated by the algorithm in order to check whether the given candidate solution, obtained from solving (3.7), satisfies the desired probability of violation. The proposed scheme is reported in Algorithm 3.2.

Algorithm 3.2 A SEQUENTIAL RANDOMIZED ALGORITHM

1. INITIALIZATION

   Set the iteration counter to zero ($k = 0$). Choose the desired accuracy $\varepsilon \in (0, 1)$, confidence $\delta \in (0, 1)$ and level $\rho \in [0, 1)$ parameters and the desired number of iterations $k_t > 1$.

2. UPDATE

   Set $k = k + 1$ and $N_k \geq N\frac{k}{k_t}$ where N satisfies (3.12).

3. DESIGN

   - Draw N_k i.i.d samples $\mathbf{q}_d = \{q_d^{(1)} \dots q_d^{(N_k)}\}$ from the uncertainty set $\mathbb{Q}$ based on the underlying distribution.
   - Solve the following optimization problem

$$\begin{aligned} &\underset{\theta}{\text{minimize}} \quad c_\theta^T \theta \\ &\text{subject to} \quad \widehat{V}_g(\theta, \mathbf{q}_d) \leq \rho. \end{aligned} \tag{3.14}$$

   - **If** the optimization problem (3.14) is not feasible, the original problem is not feasible as well.
   - **Else if**, the last iteration is reached ($k = k_t$), $\widehat{\theta}_{N_k}$ is a probabilistic robust solution and **Exit**.
   - **Else**, continue to the next step.

---

1. VALIDATION

- Draw
$$M_k \geq \frac{\alpha \ln k + \ln\left(\mathcal{S}_{k_t}(\alpha)\right) + \ln \frac{1}{\delta}}{\ln\left(\frac{1}{(\rho+\varepsilon)a^{\rho-1}+a^{\rho}(1-(\rho+\varepsilon))}\right)} \tag{3.15}$$
i.i.d. samples $\mathbf{q}_v = \{q_v^{(1)} \ldots q_v^{(M_k)}\}$ from the uncertainty set $\mathbb{Q}$ based on the underlying distribution.

- **If**
$$\widehat{V}_g(\widehat{\theta}_{N_k}, \mathbf{q}_v) \leq \rho$$
then, $\widehat{\theta}_{N_k}$ is a probabilistic solution and **Exit**.

- **Else**, goto step (2).

Note that step (3) of this Algorithm is for the case of strict LMIs/BMIs. In the nonstrict case the bound (3.12) needs to be replaced by the bound discussed in Remark 3.2. Note also that, the validation bound (3.15) in step (4), the parameters $a \geq 1$ and $\alpha > 0$ are real and $\mathcal{S}_{k_t}(\alpha)$ is a finite hyperharmonic series also known as p-series, i.e.
$$\mathcal{S}_{k_t}(\alpha) = \sum_{k=1}^{k_t} \frac{1}{k^{\alpha}}.$$

The theoretical properties of Algorithm 3.2 are summarized in the next theorem, see Theorem 5 in [33] for proof.

Lemma 3.1. *Suppose $\epsilon, \delta \in (0, 1)$ are given. Then, if at iteration k Algorithm 3.2 exits with a probabilistic solution $\widehat{\theta}_{N_k}$, then it holds that $V_g(\widehat{\theta}_{N_k}) \leq \rho+\varepsilon$ with probability* *no smaller than* $1-\delta$*, i.e.*

$$\Pr_q\left\{V_g(\widehat{\theta}_{N_k}) \le \rho + \varepsilon\right\} \ge 1-\delta.$$

Remark 3.3 (Comments on Algorithm 3.2 and related results)**.** *Algorithm 3.2 follows the general scheme of other sequential algorithms previously developed in the area of randomized algorithms for control of uncertain systems, see [30], and in particular [4, 5, 34, 82]. However, we remark that the sample bound* M_k *in Algorithm 3.2 is strictly less conservative than the bound derived in [4] because the infinite sum (Riemann Zeta function) is replaced with a finite sum, following ideas similar to those recently introduced in [34]. This enables us to choose* $\alpha < 1$ *in (3.15) resulting in up to 30% improvement in the sample complexity.*

Another important difference is on how the cardinality of the design sample set N_k *appears in the sequential algorithm. In [34, Algorithm 1], the constraints are required to be satisfied for all the samples extracted from the set* $\mathbb{Q}$ *while, in Algorithm 3.2, we allow a limited number of samples to violate the constraints in (3.1) and (3.2), or their semidefinite versions, in both "design" and "validation" steps. Finally, we note that the sequential randomized algorithm in [34, Algorithm 2] is purely based on additive and multiplicative Chernoff inequalities and hence may provide larger sample complexity than (3.15).*

It should also be remarked that the optimal values of the constants a *and* α *depend on other parameters of the algorithm, such as the termination parameter* k_t*, the accuracy level* ε*, and the level parameter* ρ*. Suboptimal values of* a *and* α *for which the* *sample bound (3.15) is minimized for a wide range of probabilistic levels are* $a = 3.05$ *and* $\alpha = 0.9$. *Note also that for* $\rho = 0$ *the optimal value of these parameters is* $a = \infty$ *and* $\alpha = 0.1$, *and the bound (3.15) reduces to bound (12) in [34] . The parameter* ρ *plays a key role in the algorithm. Note that, as pointed out in [4], the choice of* $\rho = 0$ *may lead to an unfeasible optimization problem in (3.14) whenever the original robust LMI/BMI is unfeasible. On the other hand, if* $\rho > 0$, *problem (3.14) becomes immediately a mixed-integer program, which is numerically hard to solve even in the LMI case.*

Finally, we point out that the termination parameter k_t *defines the maximum number of iterations of the algorithm which can be chosen by the user. For problems in which the bound* N_{MI} *in Algorithm 3.2 is large, larger values of* k_t *may be used. In this way, the sequence of sample bounds* N_k *would start from a reasonably small number and would not increase dramatically with the iteration counter* k.

3.7 Numerical Simulations

We illustrate the effectiveness of the previous results for a linear model of a robot manipulator joint taken from [75]. The state-space model of the plant is given by

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ z(t) = C_1 x(t) + D_{11} w(t) \\ y(t) = Cx(t) + D_{21} w(t) \end{cases}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{c}{M^2 I_m} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{\beta}{I_{son}} & -\frac{c}{M^2 I_m} - \frac{c}{I_{son}} & -\frac{\beta}{I_{son}} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ \frac{L_t}{M I_m} \\ 0 \\ -\frac{L_t}{M I_m} \end{bmatrix}, \quad C = \begin{bmatrix} 0 & M & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \quad D_{21} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \text{ and } D_{11} = 1.$$

The nominal values of the parameters are as follows: gearbox ratio $M = -260.6$, motor torque constant $L_t = 0.6$, damping coefficient $\beta = 0.4$, input axis inertia $I_m = 0.001$, output system inertia $I_{son} = 400$, spring constant $c = 130$. We considered all plant parameters to be uncertain by 15%. The objective is to design a static output feedback controller which minimizes the worst case $\mathcal{H}_\infty$ norm of the transfer function from the disturbance channel w to the controlled output z. This problem can be formulated in terms of a bilinear matrix inequality [84] of the form

$$\begin{aligned}
&\underset{F,X,\gamma}{\text{minimize}}\ \gamma \qquad (3.16)\\
&\text{subject to } X \succ 0,\\
&\begin{bmatrix} A_F^T X + X A_F & X(B_1 + BFD_{21}) & (C_1 + D_{12}FC)^T \\ \star & -\gamma & (D_{11} + D_{12}FD_{21})^T \\ \star & \star & -\gamma \end{bmatrix} \prec 0
\end{aligned}$$

where $A_F \doteq A + BFC$, $X = X^T \in \mathbb{R}^{4\times 4}$, $F \in \mathbb{R}^{1\times 2}$ and $\star$ denotes entries that follow from symmetry.

Algorithms 3.1 and 3.2 were implemented using the *Randomized Algorithm Control Toolbox* (RACT) [116], and we used PENBMI [80] for solving BMI optimization problems. The probability density functions of all 6 uncertain parameters was assumed to be uniform due to its worst-case nature. The level parameter ρ in all simulations was chosen to be zero ($\rho = 0$). A bound on the VC-dimension of the BMI problem (3.16) can then be computed using Theorem 3.2, taking into account that $m_x + m_y = 13$ (for the design variables F, X and γ), and that $n = 6 + 4 + 1 = 11$. Applying Corollary 3.2, the corresponding bounds necessary for applying Algorithm 1 can be computed, and are reported in Table 3.1 (third column) for different values of δ and ϵ.

Clearly, these sample bounds are quite large. For this reason, we used Algorithm 3.2 to efficiently solve the problem. Since the sample complexities M_k and N_k in which Algorithm 3.2 terminates are random variables, we run the simulations 100

Table 3.1: Sample complexity bounds and simulation results obtained using Algorithm 3.2. The third column is the original sample complexity bound (3.11) for strict BMIs, and the fifth column is the sample complexity achieved using Algorithm 3.2.

ε	δ	*Bound (3.11)*	k_t	*Design samples*			*Validation samples*			*Iteration*		
				Mean	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*
0.2	10^{-2}	3.58×10^4	5×10^3	60.6	24.04	149	56.74	0.44	57	4.8	1.9	12
0.1	10^{-4}	8.12×10^4	5×10^3	149.5	58.7	336	163.2	0.49	164	5.34	2	12
0.05	10^{-6}	1.82×10^5	10^4	268.7	117.8	594	437.5	0.98	439	8.6	3.7	19
0.01	10^{-8}	1.13×10^6	10^4	1276.5	484.8	2522	2686.5	3.9	2694	6.6	2.5	13
0.005	10^{-10}	2.45×10^6	10^4	2881.9	1093.3	6310	6305.9	7.9	6323	6.8	2.6	15

times for each pair of probabilistic accuracy and confidence parameters. The mean, standard deviation and worst case values of the number of design samples, validation samples and the iteration number in which the algorithm exits are tabulated in Table 3.1. We conclude that with Algorithm 3.2 we can achieve the same probabilistic levels with a much smaller number of design samples.

3.8 Conclusions

In this chapter, we computed explicit bounds on the Vapnik-Chervonenkis dimension (VC-dimension) of two problems frequently arising in robust control, namely the solution of uncertain LMIs and BMIs. In both cases, we have shown that the VC-dimension is linear in the number of design variables. These bounds are then used in a sequential randomized algorithm that can be efficiently applied to obtain probabilistic optimal solutions to uncertain LMI/BMI. Since the sample complexity is independent of the number of uncertain parameters, the proposed algorithm runs in polynomial time.

3.9 Appendix

3.9.1 Proof of Theorem 3.2

First, we introduce the following definition.

Definition 3.8 ((γ, η)-Boolean Function)**.** *The function $g : \mathbb{R}^{m_\theta} \times \mathbb{Q} \to \{0, 1\}$ is a (γ, η)-Boolean function if for fixed q it can be written as expressions consisting of Boolean operators involving η polynomials*

$$\beta_1(\theta), \ldots, \beta_\eta(\theta)$$

in the components θ_i, $i = 1, \ldots, m_\theta$ and the maximum degree of these polynomials with respect to θ_i, $i = 1, \ldots, m_\theta$ is no larger than γ.

The following lemma [126] which is an improvement on the original result of [76], states an upper bound on the VC-dimension of $(\gamma, \eta)-$Boolean functions.

Lemma 3.2. *Suppose that $g : \mathbb{R}^{m_\theta} \times \mathbb{Q} \to \{0, 1\}$ is an (γ, η)-Boolean function, then*

$$\mathrm{VC}_g \leq 2m_\theta \lg(4e\gamma\eta). \tag{3.17}$$

In view of this lemma, in order to find the VC-dimension of the uncertain LMI and BMI problems, it suffices to represent the constraints in (3.1) and (3.2) as polynomial inequalities. It is well known that an $n \times n$ real symmetric matrix is positive definite if and only if all 2^n principal minors are positive. However, this condition is equivalent to checking positivity of all n leading principal minors.

Since LMIs are a special case of BMIs, we first prove Theorem 3.2 for the more general case of BMIs. Let $F_{\mathrm{BMI},ij}(x,y,q)$ be the ij-th element of the BMI in (3.2). The leading principal minors of $F_{\mathrm{BMI}}(x,y,q)$ are

$$
\begin{aligned}
&F_{\mathrm{BMI},11}(x,y,q),\\
&\det\left(\begin{bmatrix} F_{\mathrm{BMI},11}(x,y,q) & F_{\mathrm{BMI},12}(x,y,q) \\ F_{\mathrm{BMI},21}(x,y,q) & F_{\mathrm{BMI},22}(x,y,q) \end{bmatrix}\right),\ldots,\\
&\det\left(\begin{bmatrix} F_{\mathrm{BMI},11}(x,y,q) & \cdots & F_{\mathrm{BMI},1k}(x,y,q) \\ \vdots & & \vdots \\ F_{\mathrm{BMI},k1}(x,y,q) & \cdots & F_{\mathrm{BMI},kk}(x,y,q) \end{bmatrix}\right),\ldots,\\
&\det\left(F_{\mathrm{BMI}}(x,y,q)\right).
\end{aligned}
$$

Since the number of leading principal minors is n, we need to check n polynomial inequalities. Next, we need to find the maximum degree of each polynomial inequality with respect to design variables x_i, $i = 1,\ldots,m_x$ and y_j, $j = i,\ldots,m_y$. Based on the definition of determinant, k-th leading principal minor of the BMI in (3.2) for $k = 3,\ldots,n$ can be written as

$$D_k = \sum_{\ell=1}^{k} (-1)^{\ell+1} F_{\mathrm{BMI},\ell 1}(x,y,q) M_{\ell 1} \tag{3.18}$$

where D_k is the k-th principal minor and $M_{\ell 1}$ is the $(\ell,1)$ minor of a matrix formed by the first k rows and columns of the BMI in (3.2). Then, we have that the k-th leading principal minor has maximum degree k with respect to design variables $x_i, i = 1,\ldots,m_x$ and $y_j, j = 1,\ldots,m_y$. From the definition of the BMI in (3.2), it is clear that every element of the BMI, including the first leading principal minor, has maximum degree 1 with respect to the design variables $x_i, i = 1, \ldots, m_x$ and $y_j, j = 1, \ldots, m_y$. The second leading principal minor of the BMI in (3.2)

$$\begin{aligned} D_2 = & F_{\mathrm{BMI},11}(x,y,q)F_{\mathrm{BMI},22}(x,y,q) - \\ & F_{\mathrm{BMI},21}(x,y,q)F_{\mathrm{BMI},12}(x,y,q) \end{aligned}$$

is a polynomial of maximum degree 2. For $k > 2$, the maximum degree of D_k in (3.18) is defined by the multiplication of $F_{\mathrm{BMI},\ell 1}(x,y,q)$ and $M_{\ell 1}$. The former has the maximum degree 1 and the latter has maximum degree equal to D_{k-1} because they are of the same order. Hence, the maximum degree of the k-th leading principal minor with respect to the design variables for $k = 1, \ldots, n$ is k.

Therefore, checking positive definiteness of the BMI in (3.2) is equivalent to checking n polynomial inequalities of degree ranging from 1 to n which can be represented as an $(\gamma, \eta)-$Boolean function with $\gamma = \eta = n$. The result of Theorem 3.2 follows by substituting the obtained values of γ and η into (3.17). We notice that the same reasoning holds for the case of LMI and we can represent the LMI in (3.1) as an $(\gamma, \eta)-$Boolean function with $\gamma = \eta = n$.

3.9.2 Proof of Theorem 3.3

The result follows observing that an $n \times n$ symmetric matrix is positive semidefinite if and only if *all* 2^n principal minors are nonnegative. Then, following similar reasoning as in the proof of Theorem 3.2, it follows that checking positive semidefiniteness of (3.1) and (3.2) is equivalent to evaluating 2^n polynomial inequalities of degree ranging from 1 to n. This can be represented as $(\gamma, \eta)-$Boolean function with $\gamma = n$ and $\eta = 2^n$. The results of Theorem 3.3 follow by substituting the obtained values of γ and η in (3.17).

3.9.3 Proof of Theorem 3.1

The following lemma is instrumental to prove the result of Theorem 3.1.

Lemma 3.3. *At the iteration k of Algorithm 3.2, if a candidate solution $\widehat{\theta}_{N_k}$ is declared as feasible by the "validation" step then, with probability no larger than $\Big((\rho+\varepsilon)a^{\rho-1} + a^{\rho}\Big(1-(\rho+\varepsilon)\Big)\Big)^{M_k}$ it holds that $V_g(\widehat{\theta}_{N_k}) > \rho+\varepsilon$.*

Proof. The objective is to bound the probability of obtaining $\widehat{\theta}_{N_k}$ which satisfies $\widehat{V}_g(\widehat{\theta}_{N_k}, \mathbf{q}_v) \le \rho$ and $V_g(\widehat{\theta}_{N_k}) > \rho+\varepsilon$. In the following chain of inequalities, we bound this probability using a binomial distribution

$$
\begin{aligned}
&\Pr\left\{\widehat{V}_g(\widehat{\theta}_{N_k}, \mathbf{q}_v) \le \rho \text{ and } V_g(\widehat{\theta}_{N_k}) > \rho+\varepsilon\right\} = \\
&\Pr\left\{\sum_{i=1}^{M_k} g(\widehat{\theta}_{N_k}, q) \le \rho M_k \text{ and } V_g(\widehat{\theta}_{N_k}) > \rho+\varepsilon\right\} \le \\
&\Pr\left\{\sum_{i=1}^{M_k} g(\widehat{\theta}_{N_k}, q) \le \rho M_k \text{ and } V_g(\widehat{\theta}_{N_k}) = \rho+\varepsilon\right\} = \\
&\sum_{i=1}^{\rho M_k} \binom{M_k}{i} (\rho+\varepsilon)^i \Big(1-(\rho+\varepsilon)\Big)^{M_k-i} \le \\
&a^{\rho M_k}\left(\frac{\rho+\varepsilon}{a} + 1 - (\rho+\varepsilon)\right)^{M_k}.
\end{aligned}
$$

We note that the last inequality results from Lemma 1 in [6]. □

We begin the proof of Theorem 3.1 by introducing the following events

$$\begin{aligned}
&\mathrm{Iter}_k \doteq \{\text{the } k\text{th outer iteration is reached}\},\\
&\mathrm{Feas}_k \doteq \{\widehat{\theta}_{N_k} \text{ is declared as feasible in the "validation"}\\
&\ \text{step}\},\\
&\mathrm{Bad}_k \doteq \{V_g(\widehat{\theta}_{N_k}) > \rho + \varepsilon\},\\
&\mathrm{ExitBad}_k \doteq \{\text{Algorithm 3.2 exits at iteration } k \cap \mathrm{Bad}_k\},\\
&\mathrm{ExitBad} \doteq \{\text{Algorithm 3.2 exits at some unspecified iteration}\\
&\qquad\qquad k \cap \mathrm{Bad}_k\}.
\end{aligned}$$

The goal is to bound the probability of the event "ExitBad". Since $\mathrm{ExitBad}_i \cap \mathrm{ExitBad}_j = \emptyset$ for $i \neq j$, the probability of the event "ExitBad" can be reformulated in terms of the event 'ExitBad$_k$" as

$$\begin{aligned}
\Pr\{\mathrm{ExitBad}\} &= \Pr\{\mathrm{ExitBad}_1 \cup \mathrm{ExitBad}_2 \cup \ldots \cup \mathrm{ExitBad}_{k_t}\}\\
&= \Pr\{\mathrm{ExitBad}_1\} + \Pr\{\mathrm{ExitBad}_2\} + \ldots +\\
&\quad \Pr\{\mathrm{ExitBad}_{k_t}\}. \qquad (3.19)
\end{aligned}$$

From the definition of the event "ExitBad$_k$" and by considering the point that to exit at iteration k, Algorithm 3.2 needs to reach kth iteration and declares $\widehat{\theta}_{N_k}$ as feasible, we arrive at

$$\begin{aligned}
\Pr\{\text{ExitBad}_k\} &= \Pr\{\text{Feas}_k \cap \text{Bad}_k \cap \text{Iter}_k\} \\
&= \Pr\{\text{Feas}_k \cap \text{Bad}_k \mid \text{Iter}_k\} \Pr\{\text{Iter}_k\} \\
&\le \Pr\{\text{Feas}_k \cap \text{Bad}_k \mid \text{Iter}_k\} \\
&= \Pr\{\text{Feas}_k \mid \text{Bad}_k \cap \text{Iter}_k\} \Pr\{\text{Bad}_k \mid \text{Iter}_k\} \\
&\le \Pr\{\text{Feas}_k \mid \text{Bad}_k \cap \text{Iter}_k\}. \qquad (3.20)
\end{aligned}$$

Using the result of Lemma 3.3, we bound the right hand side of (3.20)

$$\begin{aligned}
&\Pr\{\text{Feas}_k \mid \text{Bad}_k \cap \text{Iter}_k\} < \\
&\Big((\rho+\varepsilon)a^{\rho-1} + a^{\rho}\Big(1-(\rho+\varepsilon)\Big)\Big)^{M_k}. \qquad (3.21)
\end{aligned}$$

Combining (3.19) and (3.21) results in

$$\Pr\{\text{ExitBad}\} < \sum_{k=1}^{k_t} \Big((\rho+\varepsilon)a^{\rho-1} + a^{\rho}\Big(1-(\rho+\varepsilon)\Big)\Big)^{M_k}. \qquad (3.22)$$

The summation in (3.22) can be made arbitrary small by an appropriate choice of M_k. By choosing

$$\Big((\rho+\varepsilon)a^{\rho-1} + a^{\rho}\Big(1-(\rho+\varepsilon)\Big)\Big)^{M_k} = \frac{1}{k^{\alpha}} \frac{1}{\mathcal{S}_{k_t}(\alpha)} \delta \qquad (3.23)$$

where $\delta \in (0,1)$ is a (small) desired probability level, we have

$$\begin{aligned}
\Pr\{\text{ExitBad}\} &< \sum_{k=1}^{k_t} \frac{1}{k^{\alpha}} \frac{1}{\mathcal{S}_{k_t}(\alpha)} \delta \\
&= \frac{1}{\mathcal{S}_{k_t}(\alpha)} \delta \sum_{k=1}^{k_t} \frac{1}{k^{\alpha}} \\
&= \frac{1}{\mathcal{S}_{k_t}(\alpha)} \delta \mathcal{S}_{k_t}(\alpha) = \delta.
\end{aligned}$$

Therefore, the appropriate choice of M_k which guarantees $\Pr\{\text{ExitBad}\} < \delta$ can be computed by solving (3.23) for M_k which results in the bound (3.15).

Chapter 4

Application to Hard Disk Drive Servo Systems

4.1 Hard Disk Drive Servo Design

Hard disk drive (HDD) is one of the most important means of data storage. The first random access memory in the form of hard disk drive was RAMAC which was presented in 1956 (Figure 4.1). It could store up to 5 MB of data on 50 disks of two feet diameter. The average seek time for RAMAC was 600 milliseconds (ms). One of the most important factors which shows the ability of HDD to store user data is aerial density. Aerial density is represented in bit per square inch (bit/in^2) and indicates how many bits can be stored in a square inch of disk surface. Aerial density has been roughly following Moore's law since 1956. There have been a number of milestones contributing to rapid growth in aerial density such as giant magnetoresistance (GMR) or secondary actuators. The amount of data storage worldwide is estimated to be 600 Exabyte which is equivalent to 600 million disk drives each with a capacity of 1 terabyte (TB). More than 52% of this data storage requirement is met by using HDDs [60]. Continuing trend in the growth of internet, cloud computing and other similar technologies means a growing demand for higher data storage capacity and hence number of HDDs produced. To meet such increasing demand, it is important not only to increase the production volume but also to increase storage density in each HDD. Commercially produced drives at present have achieved storage density less than 500 gigabits per square inch (Gb/in^2). The HDD industry projects to achieve storage density of 10 Tb/in^2 in the near future.

Data is stored in concentric data tracks on circular disks of magnetic media. Higher storage density implies smaller dimensions for each bit, which requires reduction in distance between adjacent data tracks as well as reduction in length of each bit on a track. Assuming a bit aspect ratio (ratio between bit length and bit width) of 2:1, storage density of 10 Tb/in^2 demands track density of 2,200,000 tracks per inch (TPI). Achievable track density depends on the performance of the head positioning servomechanism of HDD, i.e., how well the read/write head is made to follow the center of a data track. The most important performance measure in HDD servo is track misregistration (TMR) which is the variance of the deviation of read/write head from the center of a data track [87]. Writing and readback of data with bit

Figure 4.1: First HDD presented by IBM [2]

error rate less than required tolerance level, demands for TMR to be less than 10% of track pitch (distance between the center of two adjacent tracks). This translates into a TMR less than 1.16 nanometer to meet the requirements for 10 Tb/in^2 storage density.

4.1.1 Hard Disk Drive Components

A commercially available hard disk drive, with its cover removed, is shown in Figure 4.2. User data is written on some concentric data tracks on disk surface. Disks are spinning with constant speed by means of spindle motor. Spinning speed varies from 5400 revolutions per minute (rpm) to more than 22000 rpm in some

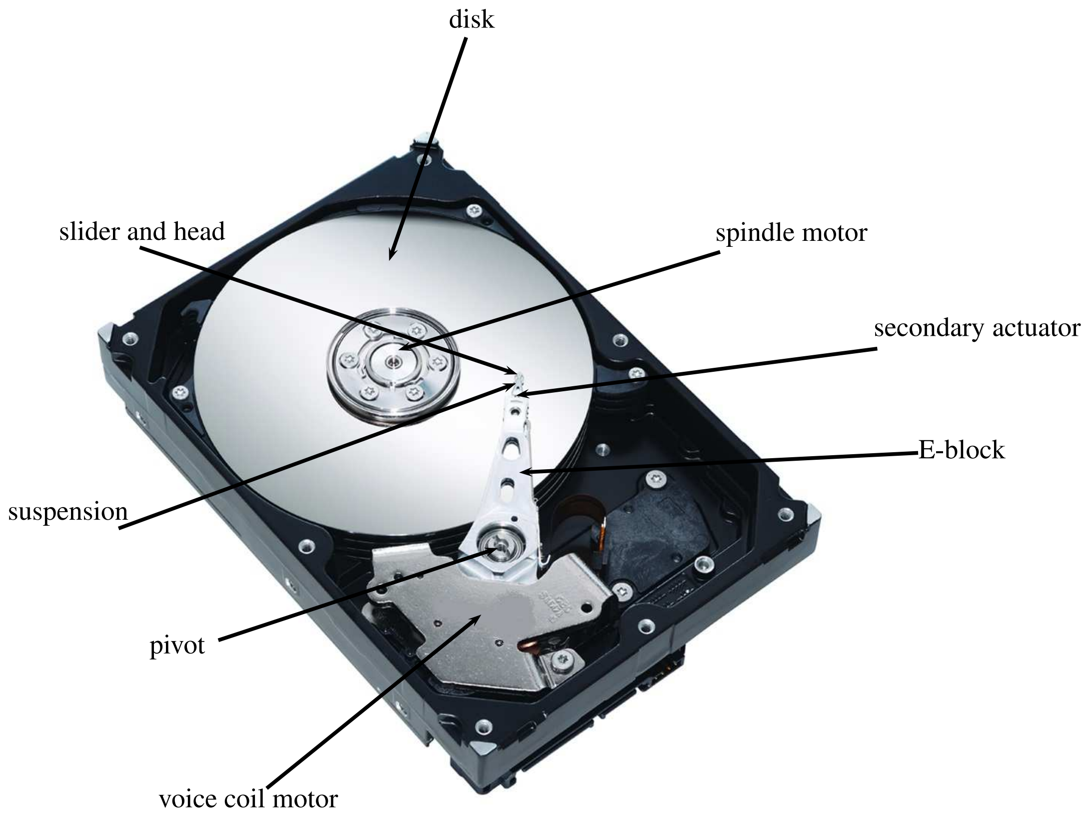

Figure 4.2: Components of hard disk drive [1]

enterprize hard disk drives. Voice coil motor (VCM) is located between two strong permanent magnets. The applied current in the coil generates a magnetic field and the interaction between generated magnetic field and the field of permanent magnets creates the required torque to move the E-block. In some disk drives additional secondary actuator is used to enhance the positioning accuracy. There are three types of secondary actuators used in hard disk drive which are shown in Figure 4.3. One of the differences between three types of actuators depicted in Figure 4.3 is the location of actuator with respect to actuator arm and read/write head. Actuated suspension

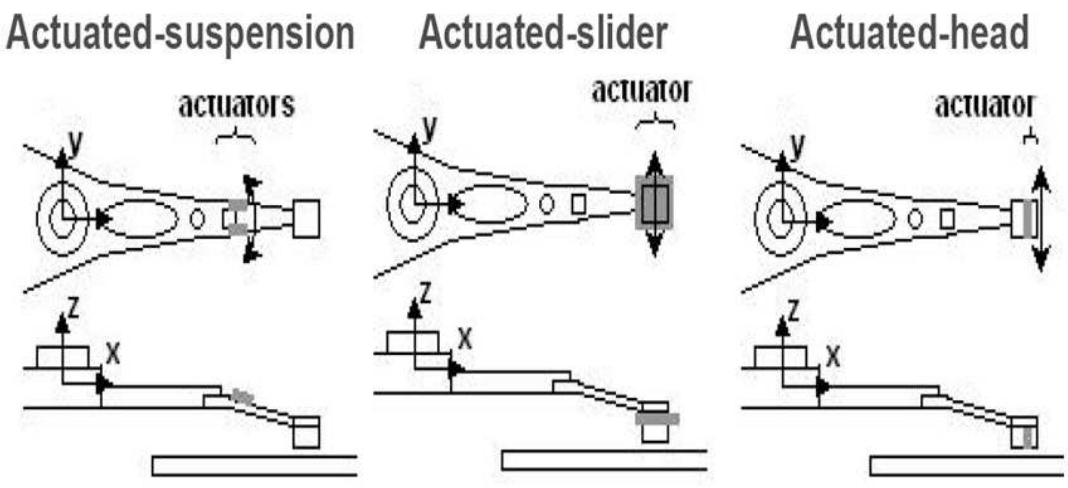

Figure 4.3: Different secondary actuators

is the most common used type in which the push-pull actuator is placed between arm and suspension. It generates the largest displacement of the head compared to the other two types; the maximum displacement is around 500 nm. In actuated slider, the actuator moves the slider and has smaller displacement than actuated suspension (around 40 nm). In actuated head, the actuator is placed inside the slider and it moves the read/write head and hence it has the smallest displacement. We highlight that the actuated head has larger bandwidth than the other two types and its first resonance mode occurs at very high frequency.

4.1.2 Servo Algorithm in Hard Disk Drive

The goal in servo design is to access the target track in minimum time (track seeking) and to keep read-write head on desired track in the presence of different disturbances (track following). Time optimal controllers such as proximate time optimal servo-mechanism (PTOS) [131, 64] is usually used for track seeking. Optimization based technique such as reference signal shaping [20] was also recently proposed to minimize the seeking time while generating minimum vibration. Track following controller tries to reject all the disturbances and keep read/write head on the center of data track. There are two types of disturbances in disk drive [3]: i) repeatable runout (RRO) and ii) non-repeatable runout (NRRO). As it is inferred from their names, the former has repeatable and hence predictive nature while the later does not enjoy such a property. Disk eccentricity, spindle motor induced motion and servo writing written in error are the main contributors of RRO. Due to predictive nature of RRO, it can be compensated by adaptive feedforward [132, 133] or repetitive control [72]. Actuator windage, external shock and disk flutter are main sources of NRRO. Because of their unpredictable nature, it is difficult to compensate them. External sensor [129], instrumented suspension [65] and self sensing actuator [46, 8, 134] are three common techniques for compensating such disturbances. We remark that the mentioned compensation techniques are independent from the controller doing the main task of positioning such as PID, state feedback, etc.; we call such a controller the central controller. Recently in some robust techniques the compensator is designed along with the central controller and there is no clear distinction between central controller and disturbance compensator, see e.g. [37]. Track density in hard disk drive is determined by the performance of its servomechanism. As mentioned earlier, aerial density of $10\,Tb/in^2$ requires TMR to be less than $1.16\,nm$. Such high performance has to be achieved in a robust manner, that is, for all drives produced in a mass production line. A number of sources of uncertainty in HDD, for example, manufacturing tolerance, change in environmental condition, different raw materials, etc., can contribute to deterioration in performance of HDD servo. We cannot expect exactly the same characteristics over a batch of HDDs and changes in system dynamics are unavoidable. These changes can be modeled as uncertainties and a robust controller is required to mitigate those uncertainties. Such approach was not essential in the past when TMR tolerance was not very stringent. The trend in increasing data density now makes use of robust controller essential for HDD.

There are two types of uncertainties in hard disk drive: i) parametric uncertainty and ii) non-parametric (dynamic) uncertainty. Therefore, there are two categories of robust controllers used in servo design for handling parametric and dynamic uncertainties. Coming to dynamic uncertainty, μ-synthesis is the most commonly used technique. In a historical point of view, [58] can be considered as the first paper using $\mu-$synthesis for hard disk drive. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ [106] minimizes $\mathcal{H}_2$ norm while constraining $\mathcal{H}_\infty$ norm of the uncertainty channel to be smaller than a prescribed value to keep the closed loop stable against dynamic uncertainty. The sampling rate of positioning signal in HDD is limited by rotational speed and the number of servo sectors hence, the controller can only compensate disturbances up to Nyquist frequency. In multi-rate and multi-sensing control design, the suspension is instrumented with a pizo-electric sensor where the sensor information is used for compensating disturbances in the feedback loop. Since the sampling frequency of the sensor is not limited, the controller can be designed to use higher sampling rate for the instrumented sensor and lower sampling rate for the positioning signal which is read from the disk surface. Multi-rate multi-sensing mixed objective control for disk drive servo design was presented in [67], [66], and [93]. In this approach, the $\mathcal{H}_2$ norm of the sensitivity transfer function is minimized while the $\mathcal{H}_\infty$ norm bound of closed-loop complimentary sensitivity transfer function for both VCM and micro actuator loops are guaranteed to be smaller than a prescribed value. Coming to parametric uncertainty, there are not a lot of results due to its complexity. To the best of our knowledge, the first paper dealing with parametric uncertainty is [92] where the idea of local BMI optimization [74] is used for designing a robust $\mathcal{H}_2$ controller. Nevertheless, this approach has a number of disadvantages: i) it is computationally expensive and ii) the design procedure is conservative due to replacement of the original nonlinear uncertainty set with a larger affine one and iii) this approach cannot take into account dynamic uncertainty. To partially address conservativeness, the design based on parameter dependent Lyapunov function was presented in [37]; however, the computational complexity associated with this approach is significantly larger than [92] up to the point that the number of parametric uncertainties is reduced to two in order to solve the problem. The other approach capable of handling parametric and dynamic uncertainty is based on $\mathcal{H}_2$ guaranteed cost technique in [39] and [38]. However, it turns out that the worst case performance of the closed loop system obtained using Monte Carlo is significantly different from the guaranteed cost defined in the control algorithm.

In this chapter, we use the results presented in Chapter 2 and design a dynamic output feedback controller minimizing the worst case -over the uncertainty set- $\mathcal{H}_2$ norm of the transfer function from disturbance to output. We also compare the obtained results with sequential approximation method based on cutting plane iteration.

4.2 Problem Formulation

In this section, we first present the experimental setup and then we discuss the identification technique used for obtaining the voice coil motor (VCM) and pizo-electric actuator (PZT) models. Next, the problem of designing robust track following $\mathcal{H}_2$ dynamic output feedback controller is formulated in the form of uncertain linear matrix inequality (LMI) to be solved using randomized methodologies in the subsequent sections.

4.2.1 System Identification

A widely accepted technique for obtaining the mathematical model of hard disk drive servo system is frequency domain system identification technique in which the voice coil motor (VCM) and pizo-electric actuator (PZT) are excited using a sweep sine signal and the displacement is measured from written in servo pattern or by using laser doppler vibrometer (LDV). The experimental set-up consists of LDV[1], VCM amplifier, PZT amplifier, dynamic signal analyzer[2] (DSA) and the real-time processor which is a DSP based system[3]. LDV is used for measuring the displacement of the read/write head; the DSA is used for generating the required sweep sine signal and to measure the frequency response of the actuators based on the generated sine signal and the displacement signal obtained from LDV. VCM amplifier is a current amplifier for driving the VCM while PZT amplifier amplifies its input voltage by a large factor (20 in our case) and finally, the designed control algorithm is executed in a real-time processor which is equipped with high accuracy analog input and outputs. A commercial disk drive is chosen as the platform to examine the designed controller. In order to let the head tip to be accessible to the LDV, the aluminium foil which is covering a small hole on the casing is replaced with a transparent glass. The glass isolates the internal environment from dust and keeps the internal airflow the same as normal operating condition. As shown in Figure 4.4, a touchlight is used as illumination module in order to make read/write head and slider visible. The rotational speed is fixed at 7200 *rpm* using spindle motor driver[4] and all experiments are carried on a vibration free table to minimize the effect of external vibrations.

In order to identify VCM and PZT actuator models, starting from low frequency

[1]Polytec OFV 5000, Polytec, Waldbronn, Germany.
[2]HP 35670A, Hewlett Packard Company, Washington.
[3]DSpace DS1103, product of dSPACE GmbH, Paderborn, Germany.
[4]Spin-Box SPB-NL-1205, DSI, Singapore

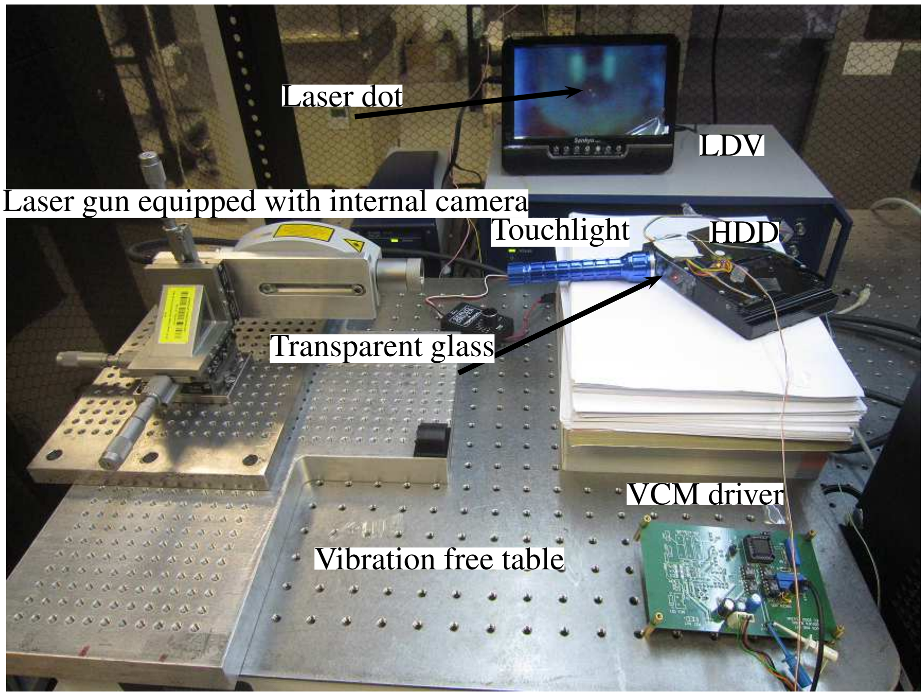

Figure 4.4: Experimental set-up

we apply a sine signal to the actuator and at the same time the displacement of VCM or PZT is measured using LDV. The magnitude and phase of bode plot representing the frequency response of actuator can be obtainn by comparing input and output signals. By increasing the frequency of sine signal we can obtain the frequency response for the entire frequency range which is usually 10 Hz to 20 KHz for VCM and 100 Hz to 30 KHz for PZT. The measured frequency response obtained from these experiments as well as the identified one are shown in Figure 4.5 and 4.6 for VCM and PZT, respectively. The transfer function of VCM and micro-actuator are

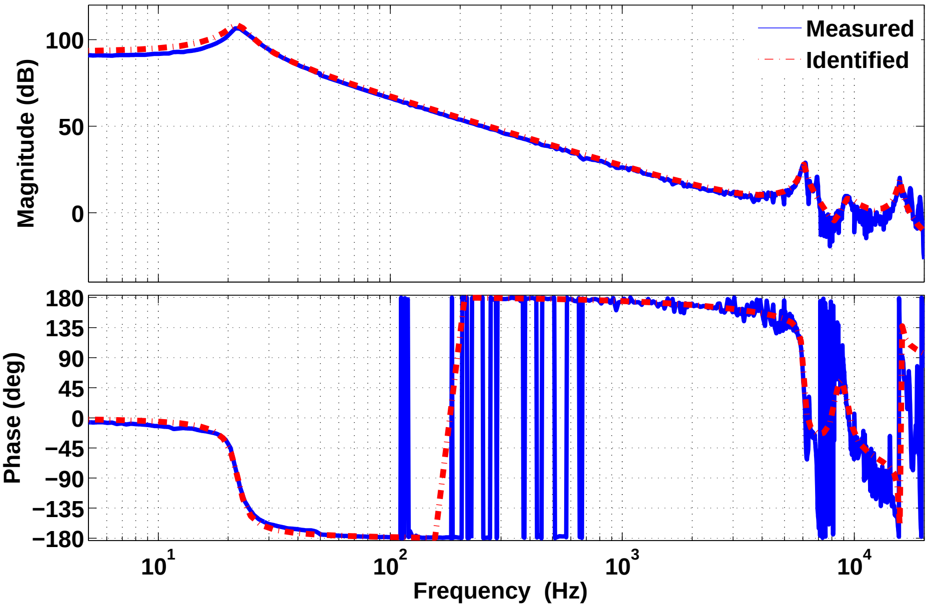

Figure 4.5: Measured as well as identified frequency response of VCM actuator

Table 4.1: Nominal VCM parameters

Parameter	$\overline{\omega}_1$	$\overline{\zeta}_1$	A_1	$\overline{\omega}_2$	$\overline{\zeta}_2$	A_2	$\overline{\omega}_3$	$\overline{\zeta}_3$	A_3	$\overline{\omega}_4$	$\overline{\zeta}_4$	A_4
Nominal Value	*207.55*	*0.5*	*5.96* $\times 10^8$	*3.64* $\times 10^4$	*0.04*	*-8.70* $\times 10^8$	*5.79* $\times 10^4$	*0.05*	*-7.19* $\times 10^8$	*9.8* $\times 10^4$	*0.03*	*1.49* $\times 10^9$

in the form

$$G_{VCM} = \sum_{i=1}^{4} \frac{A_i}{s^2 + 2\zeta_i \omega_i s + \omega_i^2}$$

$$G_{PZT} = \sum_{i=1}^{2} \frac{A_{P_i}}{s^2 + 2\zeta_{P_i} \omega_{P_i} s + \omega_{P_i}^2}$$

where parameters $\omega_i, \zeta_i, A_i, \omega_{P_i}, \zeta_{P_i}$ and A_{P_i} are plant parameters. The nominal value of all plant parameters are reported in Tables. 4.1 and 4.2 for VCM and PZT, respectively. All damping ratios ζ_i (ζ_{P_i}) and natural frequencies ω_i (ω_{P_i}) are assumed

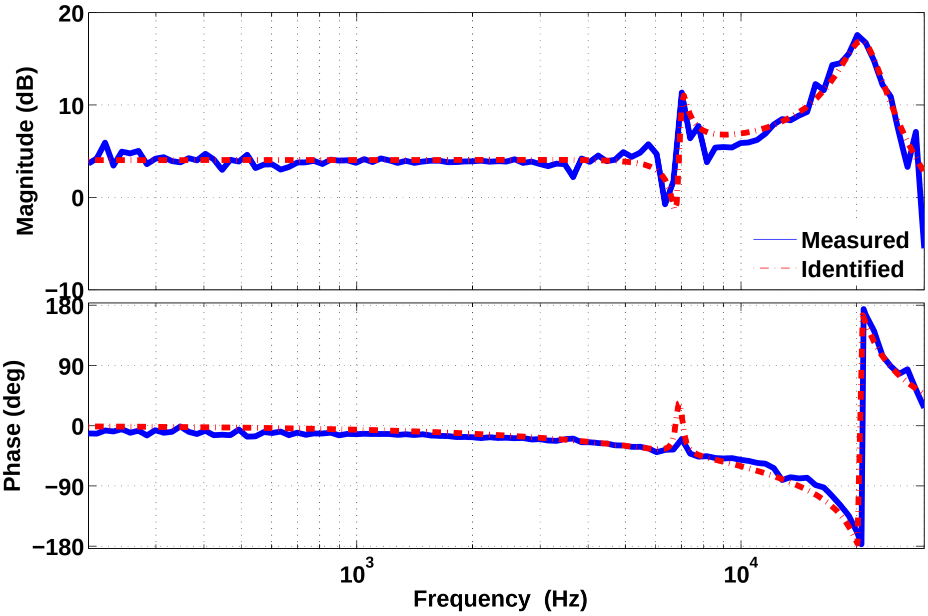

Figure 4.6: Measured as well as identified frequency response of PZT actuator

Table 4.2: Nominal PZT parameters

Parameter	$\overline{\omega}_{P_1}$	$\overline{\zeta}_{P_1}$	A_{P_1}	$\overline{\omega}_{P_2}$	$\overline{\zeta}_{P_2}$	A_{P_2}
Nominal Value	$2.31_{\times 10^4}$	0.014	$-5.8_{\times 10^7}$	$4.4_{\times 10^4}$	0.02	$-2.1_{\times 10^8}$

to be uncertain by 10% and 5% respectively. The parametric uncertainty is expressed in multiplicative form as

$$\zeta_i = \overline{\zeta}_i(1 + 0.1\eta_{zi}), \qquad \omega_i = \overline{\omega}_i(1 + 0.05\eta_{wi}), \qquad i = 1, \ldots, 4.$$

$$\zeta_{P_i} = \overline{\zeta}_{P_i}(1 + 0.1\eta_{zpi}), \qquad \omega_{P_i} = \overline{\omega}_{P_i}(1 + 0.05\eta_{wpi}), \qquad i = 1, 2.$$

where η's are unknown but bounded within the interval $[-1, 1]$.

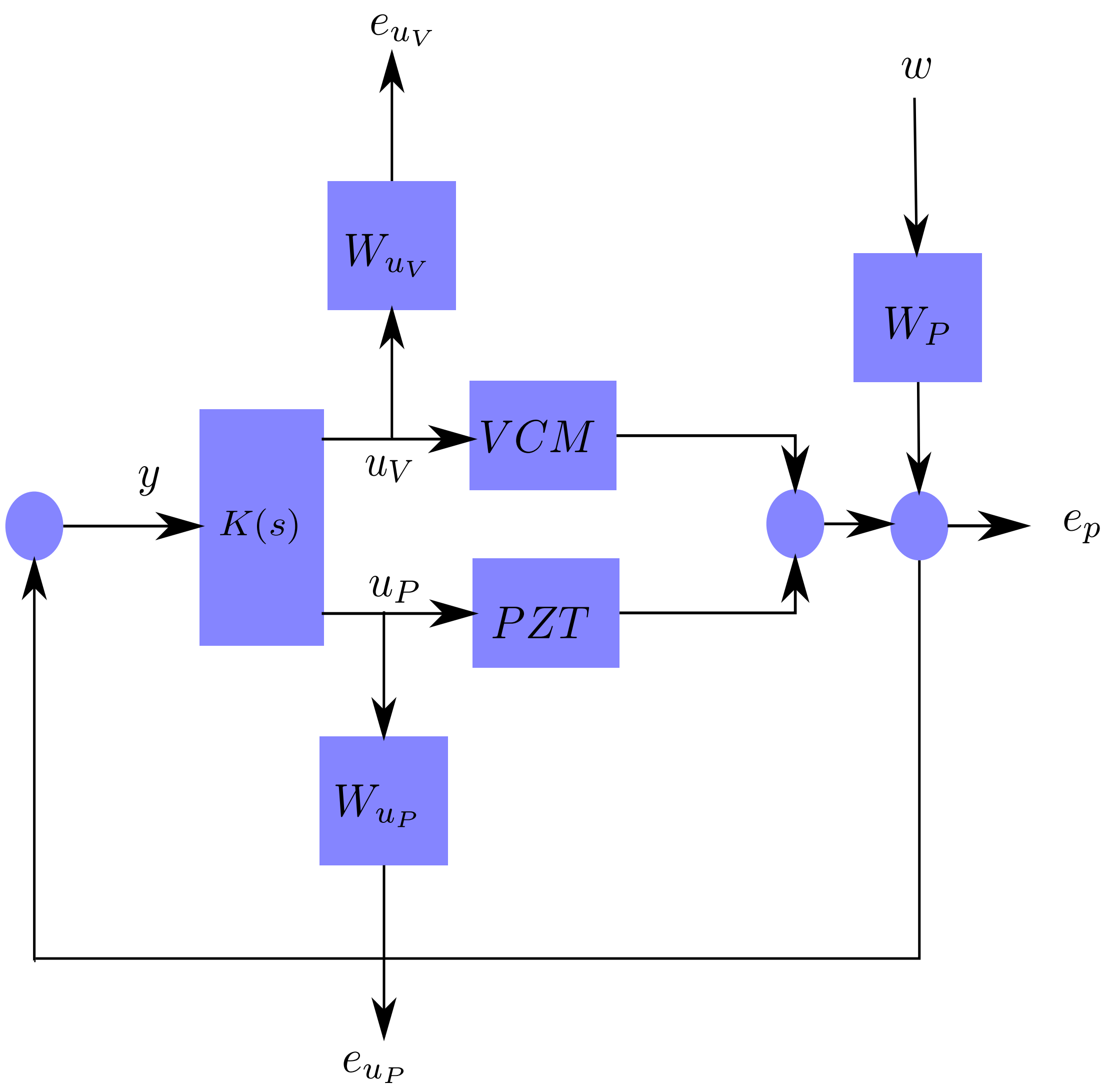

Figure 4.7: Augmented open loop

4.2.2 $\mathcal{H}_2$ Controller Formulation

In disk drive track following servo design, the goal is to minimize root mean square (RMS) of the positioning error signal (PES) while disturbance applied to the system is assumed to be Gaussian white noise. Using Parseval's Theorem, we can represent PES minimization as a standard $\mathcal{H}_2$ control synthesis. The plant needs to be augmented with necessary weighting functions to tune the performance of the designed controller in terms of closed loop bandwidth and the control input applied to the plant. The block diagram of the augmented open loop system is shown in Figure 4.7. W_{u_V}, W_{u_P} and W_p are the weighting functions and $K(s)$ is the dynamic output feedback controller to be designed. Since the closed loop sensitivity transfer function roughly follows the inverse of the performance weighting function W_p, we should design W_p in a way that it is large for low frequencies and tends to smaller values for higher frequencies. The crossover frequency can roughly be determined by the frequency in which W_p crosses the 0 dB line. The control weightings W_{u_V} and W_{u_P} are chosen to be a high-pass filters with different cutoff frequencies. Therefore, high frequency signals are prevented from being applied to the plant. The performance weighting function W_p is chosen to be in the form

$$W_p = \left(\frac{s/\overline{S} + w_c}{s + w_c \underline{S}} \right)^k$$

for which w_c is the desired crossover frequency , $\overline{S}$ determines the desired bound on the sensitivity peak, $\underline{S}$ determines the minimum level of the closed loop sensitivity transfer function in low frequencies and finally, k determines the slope of the closed loop sensitivity transfer function. The crossover frequency should be chosen very carefully as it directly affect the value of track misregistration (TMR). In the presented $\mathcal{H}_2$ methodology, we can easily achieve the crossover frequency of larger than 3 KHz by the appropriate choice of weightings. However, due to limitations imposed by Bode's sensitivity integral, pushing the crossover frequency to extremes would increase the peak of the closed loop sensitivity transfer function and hence, increases the value of TMR.

The state space realization of the augmented open loop plant depicted in Figure 4.7 is in the form

$$P(s): \quad \begin{pmatrix} z \\ y \end{pmatrix} = \left(\begin{array}{c|cc} A(q) & B_w & B_u(q) \\ \hline C_z(q) & 0 & D_{zu}(q) \\ C_y(q) & D_{yw} & 0 \end{array}\right) \begin{pmatrix} w \\ u \end{pmatrix} \tag{4.1}$$

where $q \in \mathbb{Q}$ is the vector of uncertain parameters and $z = [e_p, e_{u_V}, e_{u_V}]^T$ is the vector of controlled outputs, y is the vector of measurement signals and $u = [u_V, u_P]^T$ is the vector of control inputs. The goal is to design a dynamic output feedback controller of the form

$$y = K(s)u = \left(\begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array}\right) u \tag{4.2}$$

which minimizes the worst case (over the uncertainty set) $\mathcal{H}_2$ norm of the transfer function matrix from disturbance to output channel in the closed loop system. In other words, we aim at solving the following problem

$$\underset{K(s)}{\text{minimize}} \ \max_{q \in \mathbb{Q}} \ \|T_{z \leftarrow w}\|_2 \tag{4.3}$$

where $T_{z \leftarrow w}$ represents the closed-loop transfer function matrix from w to z. The optimization problem (4.3) can be reformulated in linear matrix inequality form [105]. Since $D_{zw} = D_{yu} = 0$, the simplified form of the LMI equivalent to (4.3) is presented in the next problem.

Problem 4.1 (Optimal $\mathcal{H}_2$ Track Following Controller Design)**.** *Solve the following uncertain LMI optimization problem*

$$\underset{\mathbf{X},\mathbf{Y},\mathbf{W},\widehat{\mathbf{A}},\widehat{\mathbf{B}},\widehat{\mathbf{C}},\gamma}{\textit{minimize}} \ \gamma$$

$$\textit{trace}(\mathbf{W}) < \gamma,$$

$$\begin{bmatrix} \mathbf{W} & C_z(q)\mathbf{X} + D_u(q)\widehat{\mathbf{C}} & C_z(q) \\ \star & \mathbf{X} & I \\ \star & \star & \mathbf{Y} \end{bmatrix} \succ 0,$$

$$\begin{bmatrix} A(q)\mathbf{X} + B_u(q)\widehat{\mathbf{C}} + \mathbf{X}A(q)^T + \widehat{\mathbf{C}}^{\mathbf{T}}B_u(q) & A(q) + \widehat{\mathbf{A}} & B_w \\ \star & \mathbf{Y}A(q) + A(q)^T\mathbf{Y} + \widehat{\mathbf{B}}C_y(q) + C_y(q)^T\widehat{\mathbf{B}}^{\mathbf{T}} & \mathbf{Y}B_w + \widehat{\mathbf{B}}D_yw \\ \star & \star & -I \end{bmatrix} \prec 0 \tag{4.4}$$

where $\mathbf{X}, \mathbf{Y}, \mathbf{X}, \mathbf{Y}, \mathbf{W}, \widehat{\mathbf{A}}, \widehat{\mathbf{B}}, \widehat{\mathbf{C}}$ *and* γ *are optimization variables and* I *is the identity matrix of appropriate dimension.*

For controller reconstruction, first we need to compute M and N having the same block structure as $\mathbf{X}$ and $\mathbf{Y}$ and satisfying

$$NM^T = I - \mathbf{X}\mathbf{Y}.$$

The controller parameters can be computed by solving the following equations for A_k, B_k and C_k

$$\begin{cases} \widehat{\mathbf{A}} = NA_kM^T + NB_kC_y\mathbf{X} + \mathbf{Y}B_u(q)C_kM^T + \mathbf{Y}A\mathbf{X} \\ \widehat{\mathbf{B}} = NB_k \\ \widehat{\mathbf{C}} = C_kM^T \end{cases}.$$

We note that the presence of uncertainty vector q in the LMI (4.4), makes the optimization problem very difficult to solve. In the next section we study a number of randomized algorithms for solving the LMI (4.4) with arbitrary high accuracy.

4.3 Randomized Algorithms for $\mathcal{H}_2$ Track-Following Design

In this section, we solve the uncertain LMI optimization Problem 4.1 using randomized algorithms. Mainly, we examine two approaches for solving (4.4):

1. Sequential methods for optimization.
2. Sequential methods for feasibility.

The former is the sequential randomized algorithms proposed in Chapter 2 which is capable of "directly" solving uncertain convex optimization problems. We highlight that the scenario approach [25, 26, 31] and Algorithms 2.1 and 2.2 are the only randomized algorithms capable of directly solving uncertain optimization problems.

Randomized algorithms for solving feasibility problems are sequential approximation methods based on gradient, ellipsoid and cutting plane iterations. These approaches cannot directly solve optimization problems although, there are modified versions of such algorithms capable of solving optimization problems please see e.g. [9, 50, 128] for further details. In [9] for instance authors used the idea of objective cuts to reduce the objective value after finding a feasible point. However, objective cut may result in loosing feasibility of the candidate solution and hence another set of feasibility cuts need to be performed to recover feasibility. Sequential feasibility methods are iterative algorithms having two main steps: i) probabilistic oracle and ii) update rule. The former is probabilistic and the later is deterministic. The role of probabilistic oracle is to examine the feasibility of the candidate solution at each iteration. The update rule is to update the previous candidate solution with the new one.

The first step in using the randomized feasibility methods is to formulate the feasibility version of optimization problem presented in Problem 4.1.

Problem 4.2 (Suboptimal $\mathcal{H}_2$ Track Following Controller Design)**.** *For fixed value* *of γ, check if the following set of uncertain LMIs are feasible*

$$
\begin{aligned}
&\mathit{trace}(\mathbf{W}) < \gamma,\\
&\begin{bmatrix} \mathbf{W} & C_z(q)\mathbf{X} + D_u(q)\widehat{\mathbf{C}} & C_z(q) \\ \star & \mathbf{X} & I \\ \star & \star & \mathbf{Y} \end{bmatrix} \succ 0,\\
&\begin{bmatrix} A(q)\mathbf{X} + B_u(q)\widehat{\mathbf{C}} + \mathbf{X}A(q)^T + \widehat{\mathbf{C}}^{\mathbf{T}}B_u(q) & A(q) + \widehat{\mathbf{A}} & B_w \\ \star & \mathbf{Y}A(q) + A(q)^T\mathbf{Y} + \widehat{\mathbf{B}}C_y(q) + C_y(q)^T\widehat{\mathbf{B}}^{\mathbf{T}} & \mathbf{Y}B_w + \widehat{\mathbf{B}}D_yw \\ \star & \star & -I \end{bmatrix} \prec 0.
\end{aligned} \tag{4.5}
$$

Algorithm 4.1 demonstrates a sequential randomized scheme for solving feasibility Problem 4.2. For simplicity of notation we denote the set of design parameters in (4.5) with $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$, the set of uncertain LMIs are also denoted by $F(\theta, q)$ and it can be decomposed as

$$F(\theta, q) = F_0(q) + \sum_{i=1}^{n_\theta} \theta_i F_i(q). \tag{4.6}$$

The equivalent version of Problem 4.2 with the new notation is presented next.

Problem 4.3. *Find the value of θ, satisfying the uncertain LMI constraint*

$$F(\theta, q) = F_0(q) + \sum_{i=1}^{n_\theta} \theta_i F_i(q) \preceq 0. \tag{4.7}$$

Algorithm 4.1 A RANDOMIZED FEASIBILITY SCHEME

- INITIALIZATION

  Set the iteration counter to zero ($k = 0$) and select an initial condition θ_0.

- PROBABILISTIC ORACLE

  - Draw $M(k)$ independent and identically distributed (iid) samples $\mathbf{q} = \{q^{(1)}, \ldots, q^{(M(k))}\}$ from the uncertainty set based on the underlying probability density function.

  - **For** $i = 1 : M(k)$

    If $F(\theta_k, q^{(i)}) \succ 0$, **then,** goto Update Rule.

    end

  - **Return** θ_k as a probabilistic solution and **exit**.

- UPDATE RULE

  Having "violation certificate" $q^{(i)}$ for which $F(\theta_k, q^{(i)}) \succ 0$, update θ_k by

$$\theta_{k+1} = \psi_{\text{upd}}(\theta_k, q^{(i)}) \tag{4.8}$$

  where $\psi_{\text{upd}}(\theta_k, q^{(i)})$ is based on gradient, ellipsoid or cutting plane iterations.

- OUTER ITERATION

  Increase the iteration counter by one ($k = k + 1$) and goto Probabilistic Oracle.

In the next two subsections we study probabilistic oracle and update rule.

4.3.1 Probabilistic Oracle

As briefly mentioned earlier, probabilistic oracle checks the probabilistic feasibility of the candidate solution by means of Monte Carlo simulation [91]. Monte Carlo has been widely used in simulation of very complex systems in physics and mathematics. The modern era of Monte Carlo algorithm is contributed from physicists N. Metropolis, S.M.Vlam, J. Von Neumann, and E. Fermi [91, 90]. The name Monte Carlo is originated from a very famous casino in Monaco and has some implications that the nature of the algorithm is similar to gambling in casinos [114]. A key problem is to determine the sample complexity, i.e., the number of random samples that should be generated, so that the so-called probability of violation is smaller than a given accuracy $\epsilon \in (0, 1)$, and this event holds with a suitably large confidence $1 - \delta \in (0, 1)$.

Definition 4.1 (Probability of Violation)**.** *The probability of violation of θ is defined as*

$$V(\theta) \doteq \Pr\{q \in \mathbb{Q} \ : \ F(\theta, q) \succ 0\}. \tag{4.9}$$

From historical point of view, Markov [88] and Chebychev [35] inequalities are the first attempts to derive probability inequalities bounding the distance between true probability of violation and the empirical violation obtained using Monte-Carlo. Next theorem [77, 112] presents a sample bound on assessing the worst case performance.

Theorem 4.1. *For any small probabilistic level $\epsilon \in (0, 1)$, with probability greater* *then* $1-(1-\varepsilon)^{M(k)}$*, either the candidate solution* θ_k *is declared as infeasible by "probabilistic oracle" of Algorithm 4.1 or the probability of violation of* θ_k *is not larger than* ε $(V(\theta_k) \leq \varepsilon)$.

Corollary 4.1 (Sample Complexity)**.** *By bounding* $(1-\varepsilon)^{M(k)}$ *with a desired small confidence parameter* $\delta \in (0,1)$*, we can derive the explicit sample complexity* $M(k)$ *guaranteeing the same results as Theorem 4.1*

$$M(k) > \frac{\ln \frac{1}{\delta}}{\ln \frac{1}{1-\epsilon}}. \tag{4.10}$$

The result of Theorem 4.1 and Corollary 4.1 derives sample complexity bounds for bounding the probability of violation at "each iteration" of Algorithm 4.1. However, since Algorithm 4.1 is a sequential method, we need to bound its "probability of misclassification".

Definition 4.2 (Probability of Misclassification)**.** *It is the probability by which Algorithm 4.1 declares a candidate solution* θ_k *as feasible and* $V(\theta_k) > \varepsilon$.

Assumption 4.1. *It is assumed that the update rule is such that it can find a robust feasible solution in finite* (k_t) *number of outer iterations.*

The theoretical properties of Algorithm 4.1 is summarized in the next theorem.

Theorem 4.2. *For any small probability levels* $\varepsilon, \delta \in (0,1)$*, let Assumption 4.1 holds. If the sample complexity* $M(k)$ *is chosen such that*

$$M_k > \left\lceil \frac{\alpha \ln k + \ln\left(\mathcal{S}_{k_t}(\alpha)\right) + \ln \frac{1}{\delta}}{\ln\left(\frac{1}{1-\varepsilon}\right)} \right\rceil \tag{4.11}$$

where the parameter $\mathcal{S}_{k_t}(\alpha)$ *in (4.11) is a finite hyperharmonic series* $\mathcal{S}_{k_t}(\alpha) = \sum_{k=1}^{k_t} \frac{1}{k^\alpha}$ *then, if Algorithm 4.1 terminates at some outer iteration* $k < k_t$ *the probability of misclassification of the obtained solution* θ_k *is smaller than* δ*.*

Proof. The proof is very similar to the proof of Theorem 2.1 and hence is not presented here. ☐

Remark 4.1 (Optimal Value of α)**.** *The optimal value of* α *which minimizes the sample bound (4.11) has been computed using numerical simulations for different values of the termination parameter* k_t*. The almost optimal value of* α *minimizing (4.11) for a wide range of* k_t *is* $\alpha = 0.1$*. The bound (4.11) (for* $\alpha = 0.1$*) improves upon the bound (17) in [30], by* 5% *to* 15% *depending on the termination parameter* k_t*. It also improves upon the bound in [95], which uses finite sum but in a less effective way.*

Remark 4.2 (Comparison with Other Results)**.** *The sample complexity (4.11) improves upon the similar result in two ways:*

1. *The finite sum* $\mathcal{S}_{k_t}(\alpha)$ *enables us to choose* $\alpha < 1$ *which considerably improves the sample complexity.*

2. *The finite sum* $\mathcal{S}_{k_t}(\alpha)$ *is clearly smaller than its infinite version which corresponds to Riemann Zeta function used in similar results.*

We note that the sample complexity (4.11) is exactly the same as (2.10) derived in Chapter 2.

4.3.2 Update Rule

Update rule is a purely deterministic procedure which roots in stochastic optimization. The very early update rule was based on gradient [24, 101] iteration. Latter algorithms based on localization methods such as ellipsoid [73] and cutting plane [29, 41] iterations were introduced into the literature. In this subsection, we study the update rule based on cutting plane iteration. In order to solve Problem 4.3, a scalar function $\tau(\theta, q)$, related to the constraint (4.7), is introduced. The function $\tau(\theta, q)$ measures the level of violation of performance function and is called *performance violation function*. There are two performance violation functions introduced in the literature: i) largest eigenvalue of the LMI and ii) norm of projection of the LMI on the cone of symmetric positive semi-definite matrices which is defined as the following

$$[F(\theta, q)]_+ \doteq \arg \min_{X \in \mathcal{S}} \|F(\theta, q) - X\|$$

where $\mathcal{S}$ denotes the cone of symmetric positive semi-definite matrices and $\|\cdot\|$ is the Frobenius norm. We used maximum eigenvalue as the performance violation function hence, $\tau(\theta, q) = \lambda_{\max}(F(\theta, q))$ where $\lambda_{\max}$ represents the largest eigenvalue and $F(\theta, q)$ is defined in (4.7). The function $\tau(\theta, q)$ is non-differentiable whenever the largest eigenvalue has multiplicity of greater than one [96]. We can still compute the sub-gradient using variational characterization of the largest eigenvalue [62]. A subgradient of $\tau(\theta, q)$ at $\theta = \theta_k$ has been computed as

$$\partial_\theta \tau(\theta_k, q) = \left[\xi_{\max}^T F_1(q)\xi_{\max}, \ldots, \xi_{\max}^T F_{n_\theta}(q)\xi_{\max}\right]$$

where $\xi_{\max}$ is a unit norm eigenvector associated with the largest eigenvalue of $F(\theta_k, q)$. The subgradient $\partial_\theta \tau(\theta_k, q)$ is obtained observing that $\lambda(F(\theta, q)) = \xi^T F(\theta, q)\xi = \xi^T F_0(q)\xi + \sum_{i=1}^{n_\theta} \theta_i \xi^T F_i(q)\xi$ which is affine and hence convex function of θ. Letting $\xi_{\max}$ be the a vector such that the largest eigenvalue is attained, we obtain the subgradient $\partial_\theta \tau(\theta_k, q)$. In localization methods such as cutting plane algorithm, at each iteration we update a localization set $\mathcal{L}_k$ which is guaranteed to contain the feasible set $\mathcal{S}$. Having the violation certificate $q^{(i)}$ for which $F(\theta_k, q^{(i)}) \succ 0$, $\mathcal{L}_k$ is updates with a smaller polytope. Since $F(\theta_k, q^{(i)}) \succ 0$, hence the largest eigenvalue is greater than zero which implies $\tau(\theta_k, q^{(i)}) > 0$. From convexity of $\tau(\theta, q)$ and the definition of subgradient it holds that

$$\tau(\theta, q^{(i)}) \geq \tau(\theta_k, q^{(i)}) + \partial_\theta \tau(\theta_k, q^{(i)})(\theta - \theta_k).$$

Therefore, for all points in the half space $\overline{H}_k = \{\theta : \partial_\theta \tau(\theta_k, q^{(i)})^T \theta > \partial_\theta \tau(\theta_k, q^{(i)})^T \theta_k\}$, the violation function $\tau(\theta, q^{(i)})$ is non-negative ($\tau(\theta, q^{(i)}) \geq 0$) and we can conclude that the solution is not in the intersection of the current polytope $\mathcal{L}_k$ and the half space $\overline{H}_k$ and it can be *cut* from the solution set. The procedure is graphically depicted in the Figure 4.8. We note that there are a number of strategies for obtaining the updated candidate solution θ_{k+1}. The new query point θ_{k+1} can be the center of gravity of $\mathcal{L}_{k+1}$, the center of the maximum volume ellipsoid containing $\mathcal{L}_{k+1}$,

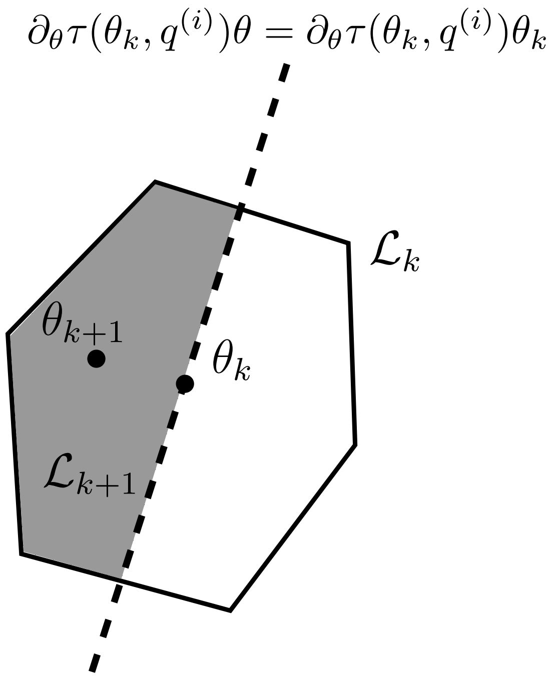

Figure 4.8: Analytic center cutting plane method

Chebyshev center of $\mathcal{L}_{k+1}$ or its analytic center. Nevertheless, analytic center of $\mathcal{L}_{k+1}$ is the most popular method used in the literature which can be efficiently computed using Newton algorithm by minimizing a logarithmic barrier function.

Remark 4.3 (Convergence)**.** *The maximum number of outer iterations k_t required for convergence in the analytic center cutting plane method is bounded by, see [29]*

$$k_t = \max\left\{50n_\theta, 13.87n_\theta^2, 8n_\theta^2\left(\frac{R}{r}\right)^{2.1}\right\} \tag{4.12}$$

where R is the radius of the initial hypercube $\mathcal{L}_0$ guaranteed to contain the feasible set $\mathcal{S}$ and r is the radius of a ball contained in the feasible set.

4.4 Simulation Study

In this section, we first study the performance of the $\mathcal{H}_2$ dynamic output feedback controller designed of Section 4.3. Next, we solve the same problem using the proposed methodologies of Chapter 2. Finally, we compare controllers in terms of robustness and track following performance.

4.4.1 Randomized Feasibility Design

Simulations are performed to prove the effectiveness of the designed controller in Section 4.3. The objective is to design a dynamic output feedback controller minimizing the worst case (over the uncertainty set) $\mathcal{H}_2$ norm of the transfer function from the disturbance channel w to outputs e_p, e_{u_V} and e_{u_P}, see Figure 4.7. In a classical robust approach, in cases where uncertainty is non-parametric (e.g. high frequency unmodeled dynamics), it can be represented in the form of linear fractional transformation (LFT) and thanks to small gain theorem (or $\mu-$synthesis), the controller objective can be reformulated into controller design in the absence of uncertainty. However, for parametric uncertainty following the same approach is over-conservative and when the number of parametric uncertainties increase, the optimization procedure fails to converge. Hence, one of the novelties in probabilistic controller design is that by accepting a very small risk, the $\mathcal{H}_2$ controller is designed which robustly stabilizes the closed-loop plant with the desired probabilistic levels. To reduce the number of outer iterations of Algorithm 4.1, we solved the $\mathcal{H}_2$ problem for the nominal plant using

YALMIP [86] and formed a hypercube centered at the solution to the nominal case. The hypercube is given to Algorithm 4.1 as the initial localization set guaranteed to contain a feasible set. After a number of iterations which depends on the user defined parameters ε and δ , Algorithm 4.1 comes up with the design parameters that make the closed-loop plant robustly stable (in a probabilistic sense). The probabilistic levels ε and δ are user defined parameters in the algorithm. As it is clear from (4.11), smaller ε and δ tend to larger sample bound requiring more computational effort in the "Probabilistic Oracle" to validate the candidate solution. Therefore, there is a tradeoff between computational complexity and smaller probabilistic risk and confidence levels. We chose 10^{-2} and 10^{-6} for ε and δ respectively and using Algorithm 4.1 which was implemented in Matlab [116], we solved the control design problem. We remark that, it is very difficult (if not impossible) to determine the probability density function (pdf) of the uncertain parameters in practice. Nevertheless, uniform pdf exhibits a worst case property [10] and is used in cases where the underlying pdf is unknown. Therefore, in this chapter we used uniform pdf while sampling the uncertainty set.

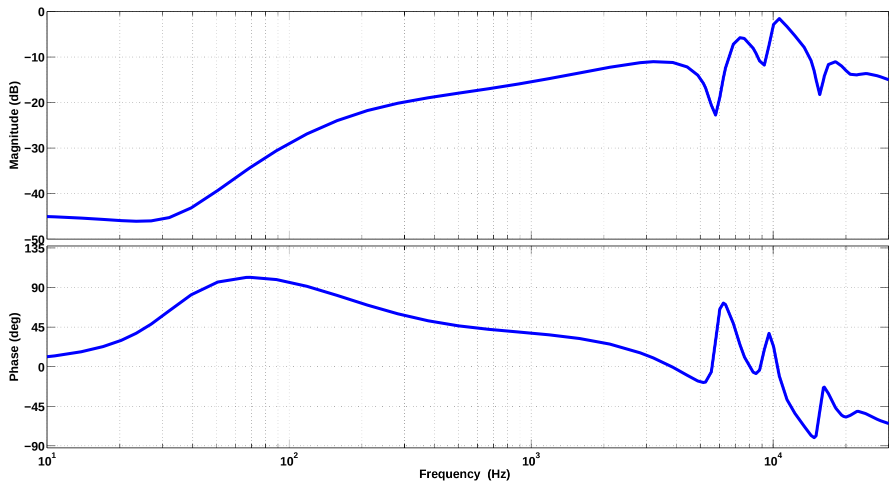

Figure 4.9: The VCM controller transfer function designed using Algorithm 4.1 while the iterative method based on cutting-plane update rule has been used.

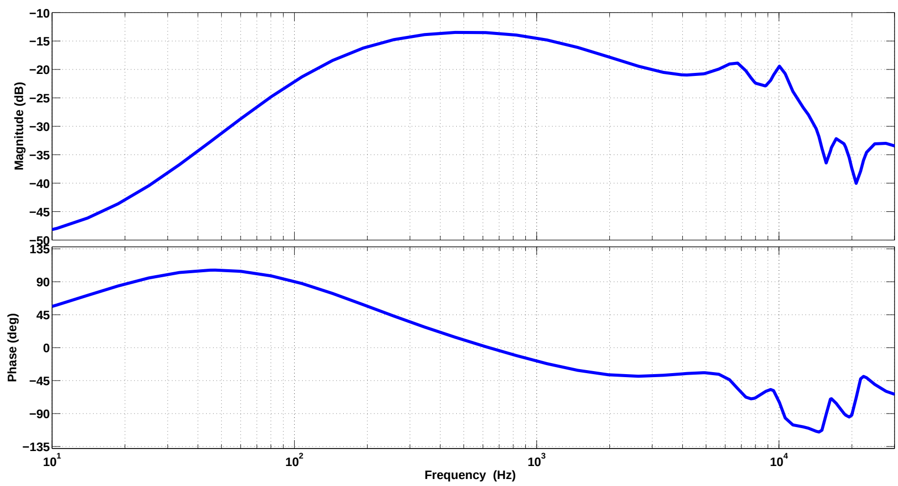

Figure 4.10: The PZT controller transfer function designed using Algorithm 4.1 while the iterative method based on cutting-plane update rule has been used.

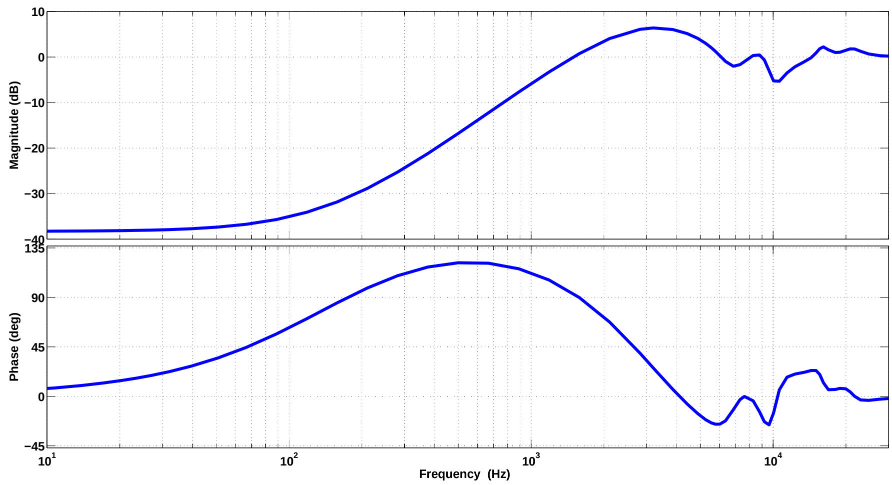

Figure 4.11: The sensitivity transfer function resulted from the controller designed using Algorithm 4.1 while the iterative method based on cutting-plane update rule has been used.

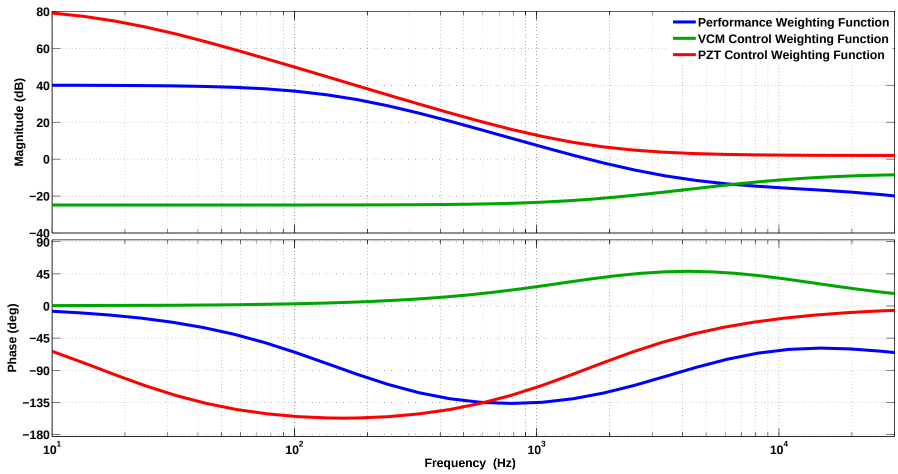

Figure 4.12: The performance weighting function along with VCM and PZT control weighting functions leading to the controller transfer function depicted in Figure 4.9 and 4.10.

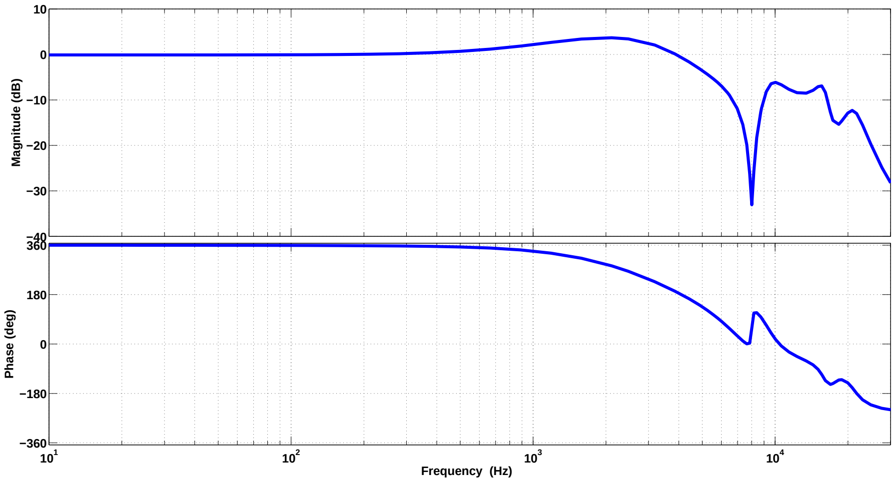

Figure 4.13: The closed-loop transfer function resulted from the controller designed using Algorithm 4.1 while the iterative method based on cutting-plane update rule has been used.

The designed controller is depicted in Figure 4.9 and 4.10. The sensitivity transfer function representing the ability of the controller to reject external disturbances is also shown in Figure 4.11. The smallest frequency in which the sensitivity transfer function crosses 0 dB line is called the crossover frequency. As it is clear from Figure 4.11 for frequencies bellow the crossover frequency, the controller attenuates external disturbances applied to the system while, right after the crossover frequency the disturbance is amplified rather than attenuated. This property is referred to as "water bed effect" and results from bode integral theorem. It worth pointing that this property holds in all linear control systems. With these ingredients in mind, it is very important to i) identify the frequencies in which major disturbances occur and ii) design the weighting functions such that the crossover frequency does not coincide with these frequencies. For this reason, we designed weighting functions with great attention. In particular, there is a tradeoff between the performance of the controller which is reflected in sensitivity transfer function and the amplitude of the control signal applied to the plant. We performed extensive simulations with different weighting functions to find those with (sub)-optimal performance. The dynamical equation of the selected weighting functions are reported here and their bode plot is depicted in Figure 4.12.

$$W_P = \frac{150000s^2 + 7.069 \times 10^9 s + 8.327 \times 10^{13}}{6.25s^3 + 9.493 \times 10^5 s^2 + 1.773 \times 10^9 s + 8.327 \times 10^{11}}$$

$$W_{u_V} = \frac{s + 10000}{s + 70000}$$

$$W_{u_P} = \frac{5s^2 + 100000s + 5 \times 10^8}{4s^2 + 800s + 40000}$$

The closed loop transfer function is also shown in Figure 4.13.

4.4.2 Randomized Optimization Design

We directly applied Algorithms 2.1 and 2.2 of Chapter 2 to the formulated problem of Section 4.2.2. One of the most important superiorities of Algorithms 2.1 and 2.2 compared to the "feasibility" Algorithm 4.1 is that they can directly solve uncertain "optimization" problems while, Algorithm 4.1 is only capable of solving feasibility problems. The controller, sensitivity and closed loop transfer functions of the two designed controllers using Algorithms 2.1 and 2.2 are reported in Figure 4.14 to Figure 4.17 respectively. The number of design and validation samples in which Algorithms

2.1 and 2.2 terminate are summarized in Table 4.3. The sample complexity bound derived based on the scenario design [25] is also reported in the same table. As clear, using Algorithms 2.1 and 2.2, we can achieve the same probabilistic accuracy and confidence levels with much smaller number of design samples compared to the scenario bound. Solving this problem using the scenario bound is way beyond the capability of current computational tools.

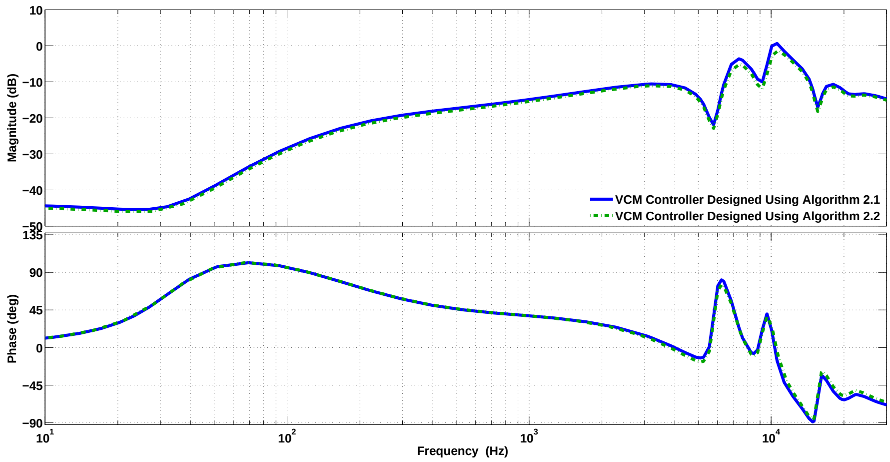

Figure 4.14: The VCM controller transfer function designed using Algorithm 2.1 (solid line) and Algorithm 2.2 (dash-dotted line).

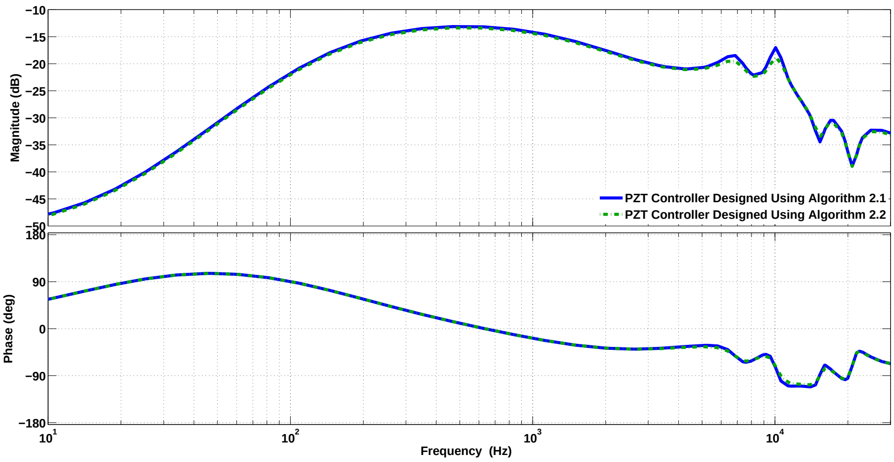

Figure 4.15: The PZT controller transfer function designed using Algorithm 2.1 (solid line) and Algorithm 2.2 (dash-dotted line).

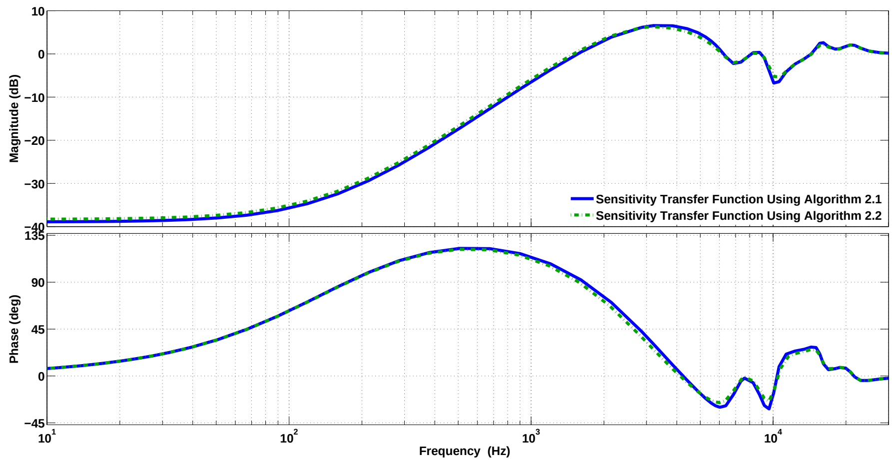

Figure 4.16: The sensitivity transfer function resulted from the controller designed using Algorithm 2.1 (solid line) and Algorithm 2.2 (dash-dotted line).

Table 4.3: The number of design and validation samples in which Algorithms 2.1 and 2.2 terminate along with the corresponding iteration number. The scenario bound for the same probabilistic accuracy and confidence level is also reported in forth column

Algorithm	ε	δ	k_t	*Scenario Bound*	*Design Samples*	*Violation Samples*	*Iteration*
Algorithm 2.1	0.01	10^{-5}	250	93618	1873	1667	5
Algorithm 2.2	0.01	10^{-5}	500	93618	1688	31910	9

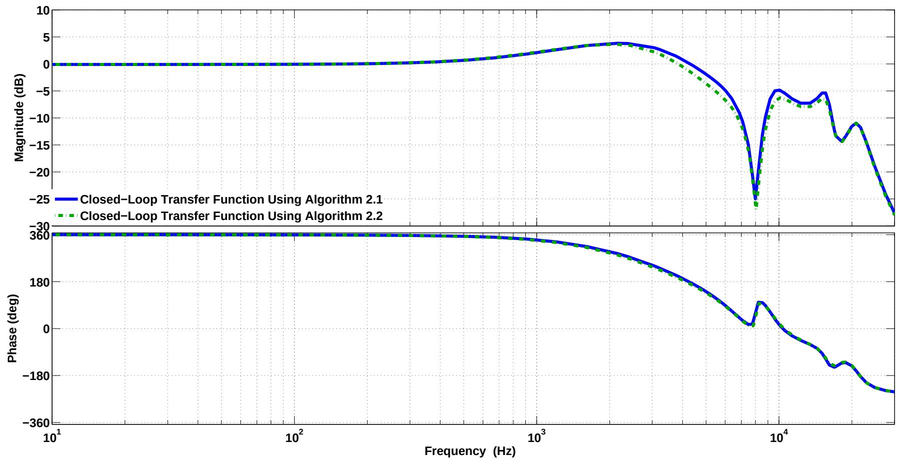

Figure 4.17: The closed loop transfer function resulted from the controller designed using Algorithm 2.1 (solid line) and Algorithm 2.2 (dash-dotted line).

4.4.3 Robustness Analysis

To further validate our design, a posteriori analysis using Monte-Carlo simulation is carried for the designed controllers. To do so, 500 random uncertain plants are chosen from the uncertainty set, then we closed the loop for each of them. A modified version of the standard disturbance [70], which includes repeatable as well as non-

Table 4.4: Comparison of the nominal and worst case (among 500 scenarios) performance specifications

Design Approach	*TMR (nm)*		*RMS(u_V) (mV)*		$\|u_V\|_\infty$*(mV)*	
	Nominal	*Worst Case*	*Nominal*	*Worst Case*	*Nominal*	*Worst Case*
Cutting-Plane	*8.39*	*8.81*	*7.5*	*8.1*	*30*	*31*
Algorithm 2.1	*8.5*	*8.85*	*7.43*	*8.2*	*30*	*31*
Algorithm 2.2	*8.4*	*8.8*	*7.4*	*8*	*30*	*31*

Table 4.5: Comparison of the nominal and worst case (among 500 scenarios) stability margins

Design Approach	*Gain Margin (dB)*		*Phase Margin (Degree)*	
	Nominal	*Worst Case*	*Nominal*	*Worst Case*
Cutting-Plane	*6.35*	*1.73*	*38.2*	*36.26*
Algorithm 2.1	*6.3*	*1.52*	*35.65*	*34.32*
Algorithm 2.2	*6.34*	*1.58*	*36.35*	*35.61*

repeatable runouts (RRO and NRRO), is used in order to evaluate the track-following performance of the designed controller. The track misregistration (TMR), root mean square (RMS) and peak values of the control input signals are tabulated in Table 4.4 for nominal as well as worst case scenarios of the 500 randomly selected uncertain plants. The nominal and worst case stability margins (phase and gain margins) are also compared for all the controllers in Table 4.5.

We note that ignoring uncertainty in the design procedure can lead to significant damage. To show this, we designed a non-robust $\mathcal{H}_2$ dynamic output feedback controller using `h2syn` command in Matlab and evaluated the closed-loop stability in the presence of uncertainty. Figure 4.18 demonstrates the closed-loop eigenvalues for 500 randomly selected uncertain plants. As it is clear, the closed-loop plant becomes unstable for some of the random scenarios extracted from the uncertainty set. Figure 4.19 to 4.21 also show the same plot for the designed probabilistic controllers. All the eigenvalues are in the open left-half plane which means the probabilistic controllers robustly stabilize the uncertain plant for all random uncertain plants. We remark that the main purpose of Figure 4.19 to 4.21 is to show that the non-robust controller cannot stabilize the uncertain plant and hence, a robust controller is a must for the plant under consideration.

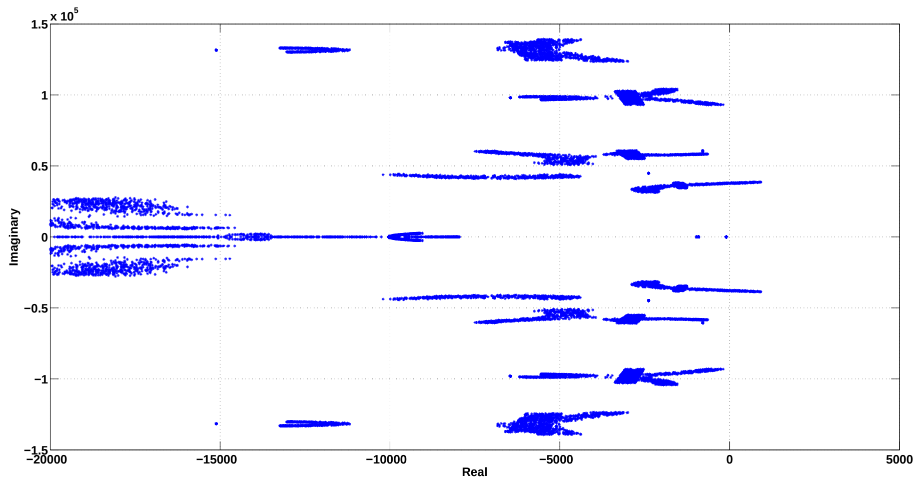

Figure 4.18: The closed loop eigenvalues for 500 randomly selected plants from the uncertainty set when a non-robust $\mathcal{H}_2$ dynamic output feedback controller is designed using `h2syn` command in Matlab.

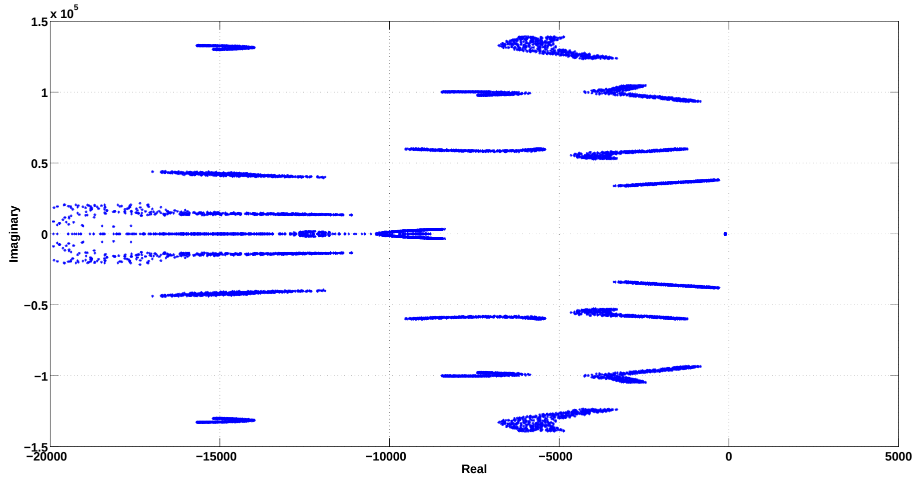

Figure 4.19: The closed loop eigenvalues for 500 randomly selected plants from the uncertainty set when a probabilistic robust $\mathcal{H}_2$ dynamic output feedback controller is designed using cutting-plane method.

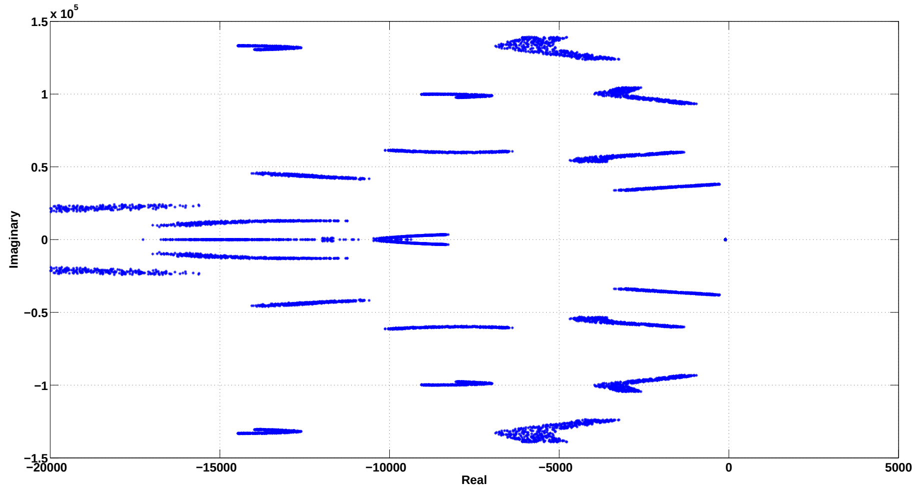

Figure 4.20: The closed loop eigenvalues for 500 randomly selected plants from the uncertainty set when a probabilistic robust $\mathcal{H}_2$ dynamic output feedback controller is designed using Algorithm 2.1.

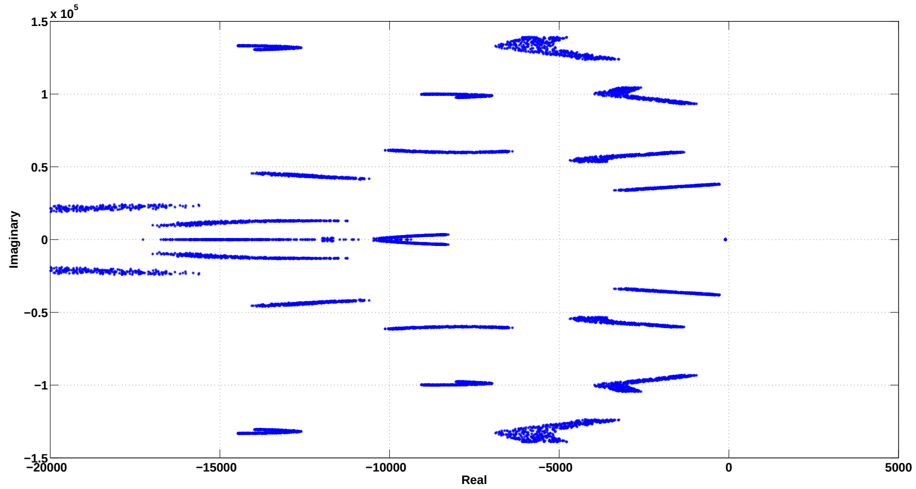

Figure 4.21: The closed loop eigenvalues for 500 randomly selected plants from the uncertainty set when a probabilistic robust $\mathcal{H}_2$ dynamic output feedback controller is designed using Algorithm 2.2.

4.5 Real Time Implementation

Simulation results in the previous section confirmed that the designed controllers achieve robust stability and performance in the presence of parametric variations in the dynamical system. The validity of the designed controllers is testified through experiment. The control algorithm designed using probabilistic sequential scheme is discretized and implemented in real time using the DSP based system with sampling frequency of 50 kHz. Since the problem is of regulation type, the output sensitivity transfer function, which shows the ability of the system in rejecting different output disturbances, is of vital importance. The transfer function from reference input to the error which represents the output sensitivity transfer function is experimentally measured using DSA for the designed probabilistic controller. The result of this experiment along with the closed-loop transfer function are shown in Figure 4.22. The controller and plant are discretized using "Tustin" and "zero-order-hold" respectively. The designed controller was also evaluated with a 50 Hz square wave signal. Figure 4.23 shows the output displacement for a step responses of 150 nm as well as the corresponding input signals to the VCM and PZT drivers. Each rise and fall in the reference signal is considered as a step trigger. The fluctuating signals on the step response is mostly due to disk rotation and the air-flow induced vibration.

There are a number of factors which affect the servo performance in our implementation setup and cause the experimental results to deviate slightly from the ones obtained in the simulation. The first, and the most important, factor is the computational delay in the DSP system. Starting from low frequency, the phase of the implemented controller drifts from the actual one; the phase drift increases with the frequency. For instance, the phase drift is 10 degrees at 2 kHz and increases to 30 degrees at 10 kHz. Secondly, in order to perform the experiment some modifications needs to be done to HDD e.g. the hole on the casing, applying transparent glass, etc. which requires disassembling and reassembling most of the components. This process can affect the original optimal structural dynamic of the VCM.

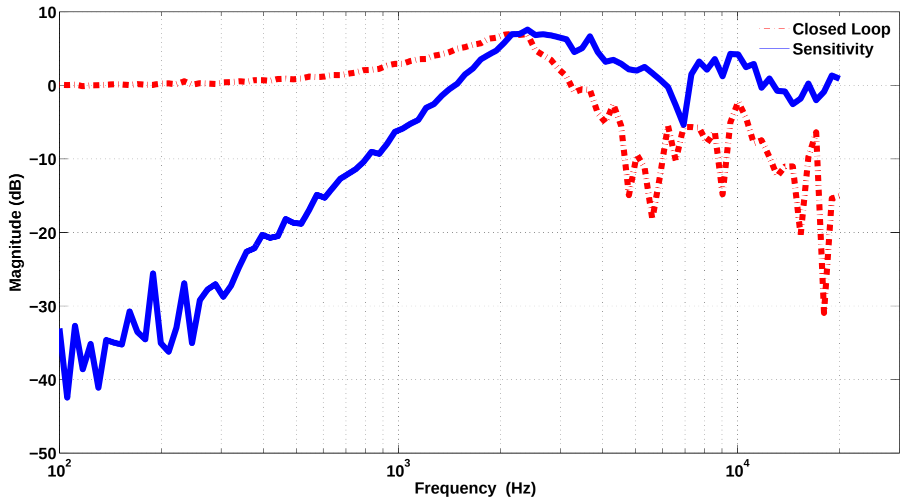

Figure 4.22: The experimental sensitivity and closed-loop transfer functions for the controller designed using sequential approximation method.

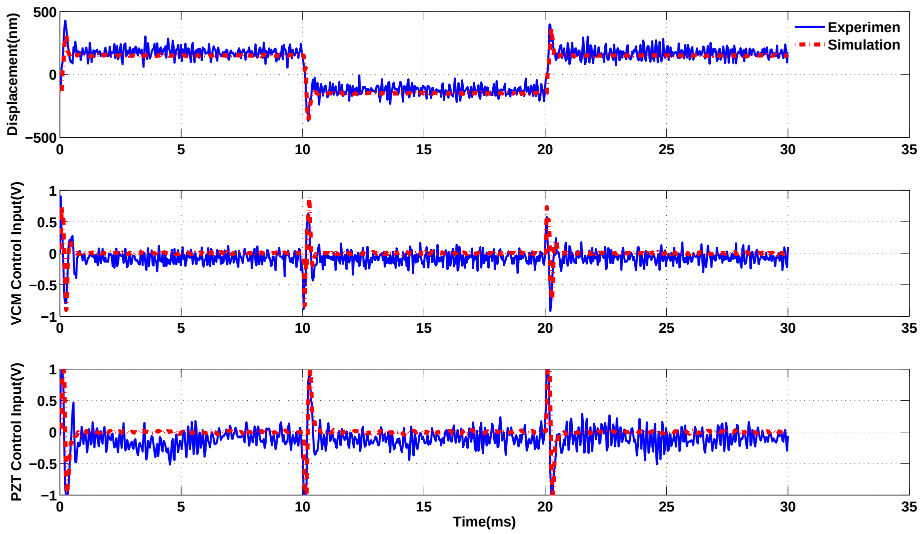

Figure 4.23: Displacement output for step trigger of 150 nm and the corresponding input signals to VCM and PZT drivers.

4.6 Conclusions

Uncertainty in the plant's dynamic is inevitable for HDDs; on the other hand, higher performance has always been in demand due to the rapid growth in world wide generated digital information. In the classical robust controller design, performance is sacrificed considerably in order to achieve robustness specially in cases where parametric uncertainty enters non-linearly into plant description. However, in the studied approach which benefits from probabilistic concepts and randomization, no conservatism has been introduced. Furthermore, classical robust design suffers from considerable computational complexity while handling parametric uncertainty. The approach based on probabilistic robust design and randomized algorithms has broken the curse of dimensionality thanks to randomization. In our case study, we designed a $\mathcal{H}_2$ dynamic output feedback controller which achieves robust stability and performance in the presence of various non-linear parametric uncertainties. Uncertainty was considered as random variable with uniform probability density. The choice of uniform probability density is chosen due to its worst case nature. We solved the problem using sequential optimization techniques of Chapter 2 and compared them with a controller designed using sequential approximation methods based on cutting-plane iteration.

Chapter 5

Summary

In this chapter, we first summarize our findings and next, we propose two promising future research directions.

5.1 Findings

This thesis addressed the problem of designing robust controllers for uncertain systems. In particular, we addressed two problems:

1. Finding the optimal value of an uncertain convex optimization problems

2. Solving uncertain linear and bilinear matrix inequalities

Many robust and optimal control problems can be formulated in either forms mentioned above. In contrary to deterministic worst-case approaches, typically associated with some degree of conservatism, we followed a probabilistic framework in which the obtained solution is guaranteed to be feasible for the entire set of uncertainty except for a very unfortunate subset having arbitrary small probability measure. We considered uncertain parameters as random variables and assumed that a probability measure is given over the Borel $\sigma-$algebra of the uncertainty set. This enabled us to extract random samples from the uncertainty set and evaluate the probabilistic behavior of the obtained solution.

Motivated by above mentioned problems, we proposed randomized algorithms to "efficiently" solve the problems at hand. Two sequential randomized algorithms for full constraint satisfaction (Algorithm 2.1) and partial constraint satisfaction (Algorithm 2.2) were presented for solving uncertain convex optimization problems. The main contribution in these algorithms is to mix the well-known scenario approach and sequential randomized algorithms previously used in the literature for feasibility problems. The main motivation behind proposing Algorithms 2.1 and 2.2 was the conservatism of the scenario bound for even relatively moderate size optimization problems. The scenario bound depends almost linearly on the number of optimization variables therefore, it turns out to be large for moderate to large size optimization problems resulting in a computationally complex random convex problem which is sometimes way beyond the capability of current computational tools. Introducing "validation step" in the randomized algorithms enabled us to find solutions guaranteeing the same probabilistic accuracy and confidence levels as the scenario approach using much smaller design samples which led to computationally less expensive optimization problem. The difference between full constraint satisfaction algorithm (Algorithm 2.1) and partial constraint satisfaction algorithm (Algorithm 2.2) is that in the later, we allow a limited number of samples to violate constraints which is done in favor of improving the objective value. This is in the same spirit of the scenario approach with discarded constraints. We highlight that the sample complexity bound in the validation step of Algorithm 2.1 improve upon similar bounds by up to 30%. The improvement was achieved thanks to replacing Riemann Zeta function, used conventionally in similar results, with a finite sum. The effectiveness of both Algorithms 2.1 and 2.2 was shown through extensive simulation regarding a multivariable model for the lateral motion of an aircraft.

We also proposed randomized algorithms for solving uncertain linear and bilinear matrix inequalities (LMIs/BMIs) frequently arising in robust control design. The algorithm was based on statistical learning theory which is an effective tool dealing with various applications. In this approach, we extract a finite number of samples, as training samples, from the uncertainty set and solve the LMI/BMI optimization problem "simultaneously" for all the samples. The goal was to derive the number of samples required to guarantee that the obtained solution remains feasible for "unseen" samples with a probability arbitrary close to one. Using statistical learning theory in robust control design requires computation of a combinatorial parameter called Vapnik-Chervonenkis dimension (VC-dimension) which is very difficult to compute in general. One of our main contributions was to firstly show that the VC-dimensions of uncertain strict and non-strict LMIs/BMIs are finite and next to derived upper bounds on them. The bounds were used in obtaining sample complexity bounds to be used in the randomized Algorithm 3.1. However, it turned out that the sample bounds are very large whenever the accuracy and confidence requirements are stringent which is typical in the context of statistical learning theory. For this reason, we proposed a sequential randomized framework in the same spirit of Algorithms 2.1 and 2.2 but tailored for the specific problem at hand. Using sequential randomized Algorithm 3.2, we *efficiently* solved a non-trivial example regarding static output feedback stabilization of an uncertain robot manipulator joint.

A common feature of all randomized algorithms presented in this thesis is that they broke the curse of dimensionality. That is, the computational complexity of the algorithms is independent form the number of uncertain parameters appearing in the problem. This feature was obtained thanks to randomization.

To see the effectiveness of the probabilistic robust theory and randomized methods on an industrial platform, we considered the problem of designing $\mathcal{H}_2$ dynamic output feedback controller for track following controller of hard disk drives (HDDs) affected by various parametric uncertainties. Due to large number of uncertain parameters entering into plant description in a nonlinear fashion, deterministic worst-case methods are unable to design the controller without imposing conservatism. That is, replacing the original nonlinear uncertainty set with a larger affine one. Using randomized algorithms we successfully designed two probabilistic controllers with manageable computational effort. In particular, we designed a suboptimal controller using probabilistic iterative methods based on cutting-plane update rule and compared it with optimal controllers designed using Algorithms 2.1 and 2.2. The robustness and track following performance of the two class of controllers were rigorously studied through extensive simulations. A probabilistic controller was also implemented in real-time on a commercial disk drive to show that the deigned controllers using this method in indeed “implementable”.

5.2 Future Research

In this section we briefly study two future research directions with good potential in improving the field of randomized algorithms and probabilistic robust theory. The first one is on using randomized algorithms for non-parametric uncertainties and the second one is regarding the problem of occasional instability of controllers designed using randomized algorithms.

5.2.1 Randomized Algorithms for Non-parametric Uncertainty

The idea of probabilistic robust and randomized algorithms has shown great potential in handling parametric uncertainty nevertheless, their use in dealing with non-parametric uncertainty is not systematically evaluated yet. Generating random transfer function in the $\mathcal{H}_\infty$ ball with the desired probability density function is not trivial. To the best of our knowledge [110] is the only paper concerned about uniformly sampling suitably chosen subset of $\mathcal{H}_\infty$. The idea used in the mentioned paper is to use $Carath\acute{e}odory - Fej\acute{e}r$ interpolation result (see e.g. [11]). However, there is still a significant room for improvement and there are a number of problems in this line of research which are still largely open. For instance sequential randomized methods based on stochastic optimization techniques can be developed on designing robust controllers for uncertain systems affected by uncertainty. The main difficulty here seems to be generating random transfer functions in $\mathcal{H}_\infty$ with the desired probability density function.

5.2.2 Randomized Algorithms for Guaranteed Stability and Probabilistic Performance

In controller design using randomized algorithms, the stability of the closed-loop system is treated in a probabilistic fashion. That is, the closed-loop system "might" become unstable for an arbitrary small subset of the uncertainty set. In critical applications where instability can cause significant damage or loss, using probabilistic controllers is not recommended. For instance, no one is interested in boarding an aircraft which is 99.999% stable. Therefore, for such a critical application we need to design controller such that the stability is guaranteed for the entire set of the uncertainty but performance can be violated for a subset having arbitrary small probability measure. The idea is to use Youla parametrization also known as Q-parametrization

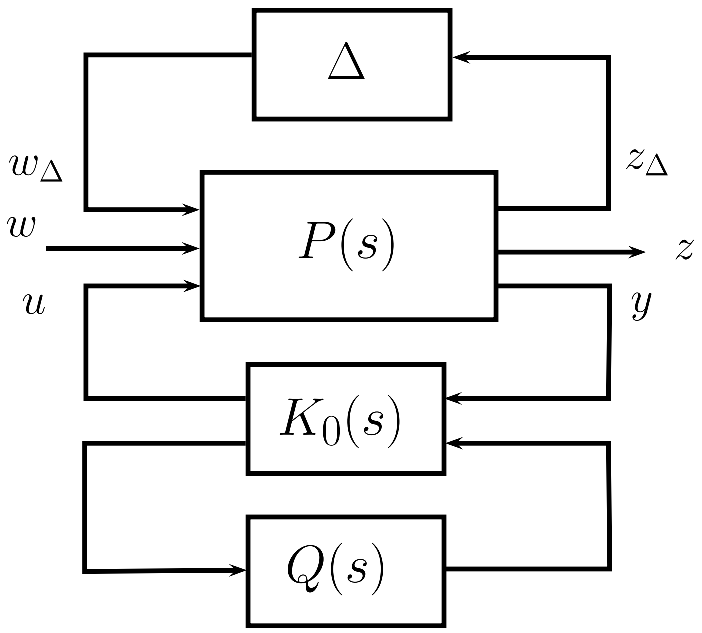

Figure 5.1: Interconnection of the plant $P(s)$ with uncertainty block Δ, central controller $K(s)$ and $Q(s)$

to characterize the set of controllers robustly stabilizing the closed-loop system and use this set as a search space to find the controller guarantying the performance index in a probabilistic sense. Consider the interconnected uncertain plant of Figure 5.1. The primary objective is to find a controller $K_0(s)$ which makes the closed-loop system to satisfy the inequality

$$\|T_{z_\Delta w_\Delta}(s, K_0)\|_\infty < \gamma \tag{5.1}$$

where $T_{z_\Delta w_\Delta}$ represent the transfer function matrix from w_Δ to z_Δ, γ is a desired level and K_0 is called the central controller. Satisfaction of (5.1) guarantees robust stability for all $\Delta \in \mathbf{\Delta}_H$ where $\mathbf{\Delta}_H$ is defined as

$$\mathbf{\Delta}_H = \{\Delta \in \mathcal{RH}_\infty : \|\Delta\|_\infty \leq \frac{1}{\gamma}\}.$$

All controllers achieving internal stability and satisfying the norm bound (5.1) can be parameterized [48] by the set $Q(\gamma) = \{Q(s) \in \mathcal{RH}_\infty : \|Q(s)\| \leq \gamma\}$. Therefore the set of all stabilizing controllers which achieve the norm bound (5.1) is defined as $K_Q(s) = \mathcal{F}_l(K_0(s), Q(\gamma))$ and $\mathcal{F}_l$ denotes the standard lower fractional transformation.

Having the stabilizing controller set $K_Q(s)$, we next search for a controller minimizing the performance objective ($\mathcal{H}_2$ norm, time domain specifications, etc.) in a probabilistic fashion. In [28] a random search in the controller space was used to find the probabilistic controller however, this method is not efficient since the optimal controller is a rare event and using a pure random search in design parameter space does not seem to be a viable solution. Optimization techniques for finding the optimal transfer function $Q(s)$ lead to a bilinear matrix inequality which in this case is numerically very difficult to solve. Therefore, finding a solution to this problem can solve one of the most important drawbacks of probabilistic robust theory and randomized methods in controller synthesis.

Bibliography

[1] Seagate ST3200822AS 200GB.

[2] History of IBM magnetic disk drives, April 2014.

[3] D. Abramovitch and G. Franklin. A brief history of disk drive control. *Control Systems Magazine, IEEE*, 22:28 –42, 2002.

[4] T. Alamo, A. Luque, D.R. Ramirez, and R. Tempo. Randomized control design through probabilistic validation. In *Proc. of the American Control Conference*, pages 839–844. see also The sample complexity of randomized methods for analysis and design of uncertain systems, `arXiv:1304.0678 [cs.SY]`, 2012.

[5] T. Alamo, R. Tempo, and E.F. Camacho. Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems. *IEEE Transactions on Automatic Control*, 54:2545–2559, 2009.

[6] T. Alamo, R. Tempo, and A. Luque. On the sample complexity of probabilistic analysis and design methods. In J. Willems, S. Hara, Y. Ohta, and H. Fujioka,

editors, *Perspectives in Mathematical System Theory, Control, and Signal Processing*, volume 398 of *Lecture Notes in Control and Information Sciences*, pages 39–50. Springer Berlin / Heidelberg, 2010.

[7] B.D.O. Anderson and J.B. Moore. *Optimal Control: Linear Quadratic Methods.* Prentice Hall, 1990.

[8] E.H. Anderson, N.W. Hagood, and J.M. Goodliffe. Self-sensing piezoelectric actuation - analysis and application to controlled structures. 1992.

[9] A. Ataei and Q. Wang. An ellipsoid algorithm for linear optimization with uncertain LMI constraints. In *Proc. of American Control Conference (ACC)*, pages 857–862, 2012.

[10] E. Bai, R. Tempo, and M. Fu. Worst-case properties of the uniform distribution and randomized algorithms for robustness analysis. *Mathematics of Control, Signals, and Systems*, 11:183–196, 1998.

[11] J. Ball, I. Gohberg, and L. Rodman. *Interpolation of rational matrix functions.* Birkhäuser Verlag, 1990.

[12] A. C. Bartlett, C. V. Hollot, and H. Lin. Root locations of an entire polytope of polynomials: It suffices to check the edges. *Mathematics of Control, Signals and Systems*, 1:61–71, 1988.

[13] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization:*

Analysis, Algorithms, and Engineering Applications. SIAM, Philadelphia,, 2001.

[14] A. Ben-Tal and A. Nemirovski. On tractable approximations of uncertain linear matrix inequalities affected by interval uncertainty. *SIAM Journal on Optimization*, 12:811–833, 2002.

[15] D. S Bernstein and W. M Haddad. LQG control with an h∞ performance bound:a riccati equation approach. *IEEE Transactions on Automatic Control*, 34:293–305, 1989.

[16] D.S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition).* Princeton University Press, 2009.

[17] S.N. Bernstein. *CThe theory of probabilities.* Gostehizdat Publishing House, Moscow, 1946.

[18] S. P. Bhattacharyya, H. Chapellat, and L. H. Keel. *Robust Control: The Parametric Approach.* Prentice Hall, Upper Saddle River, 1995.

[19] G. Blekherman. Nonnegative polynomials and sums of squares. *Journal of the American Mathematical Society*, 25:617–635, 2012.

[20] U. Boettcher, D. Fetzer, H. Li, R. A de Callafon, and F. E Talke. Reference signal shaping for closed-loop systems with application to seeking in hard disk drives. *IEEE Transactions on Control Systems Technology*, 20:335–345, 2012.

[21] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM, Philadelphia, 1994.

[22] R.P. Braatz, P.M. Young, J.C. Doyle, and M. Morari. Computational complexity of μ calculation. *IEEE Transactions on Automatic Control*, pages 1000–1002, 1994.

[23] B.R.Barmish. *New Tools For Robustness of Linear Systems.* Macmillan, NewYork, 1994.

[24] G. Calafiore and B. T Polyak. Stochastic algorithms for exact and approximate feasibility of robust LMIs. *IEEE Transactions on Automatic Control*, 46:1755–1759, 2001.

[25] G.C. Calafiore. Random convex programs. *SIAM Journal on Optimization*, 20(6), 2010.

[26] G.C. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102:25–46, 2004.

[27] G.C. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51:742–753, 2006.

[28] G.C. Calafiore and F. Dabbene. Control design with hard/soft performance specifications: a q-parameter randomization approach. *International Journal of Control*, 77:461–471, 2004.

[29] G.C. Calafiore and F. Dabbene. A probabilistic analytic center cutting plane method for feasibility of uncertain LMIs. *Automatica*, 43:2022–2033, 2007.

[30] G.C Calafiore, F. Dabbene, and R. Tempo. Research on probabilistic methods for control system design. *Automatica*, 47:1279–1293, 2011.

[31] M.C. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM J. on Optimization*, 19:1211–1230, 2008.

[32] M.C. Campi and S. Garatti. A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. 148(2):257–280, 2011 (preliminary version available on Optimization Online, 2008).

[33] M. Chamanbaz, F. Dabbene, R. Tempo, V. Venkataramanan, and Q-G. Wang. On the sample complexity of uncertain linear and bilinear matrix inequalities. In *Proc. IEEE Conference on Decision and Control*, Florence, Italy, 2013. Submitted.

[34] M. Chamanbaz, F. Dabbene, R. Tempo, V. Venkataramanan, and Q-G. Wang. Sequential randomized algorithms for sampled convex optimization. In *Proc. IEEE Multi-Conference on Systems and Control*, Hyderabad, India, 2013. to appear (extended version submitted to IEEE Transactions on Automatic Control).

[35] P.L. Chebyshev. *Sur les valeurs limites des intégrales*. Imprimerie de Gauthier-Villars, 1874.

[36] G. Chesi, A. Garulli, A. Tesi, and A. Vicino. Solving quadratic distance problems: an lmi-based approach. *IEEE Transactions on Automatic Control*, 48:200–212, 2003.

[37] R. Conway, Jongeun C., R. Nagamune, and R. Horowitz. Robust track-following controller design in hard disk drives based on parameter dependent lyapunov functions. *IEEE Transactions on Magnetics*, 46:1060–1068, 2010.

[38] R. Conway and R. Horowitz. Guaranteed cost control for linear periodically time-varying systems with structured uncertainty and a generalized h2 objective. *Mechatronics*, 20:12–19, 2010.

[39] R. Conway, J. Nie, and R. Horowitz. H_2 guaranteed cost control in Track-Following servos. In E. Eleftheriou and R. Moheimani, editors, *Control Technologies for Emerging Micro and Nanoscale Systems*, volume 413, pages 235–270. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[40] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the LambertW function. *Advances in Computational Mathematics*, 5:329–359, 1996.

[41] F. Dabbene, P. S. Shcherbakov, and B. T. Polyak. A randomized cutting plane

method with probabilistic geometric convergence. *SIAM Journal on Optimization*, 20, 2010.

[42] M.A. Dahleh, B. Diaz, and J. Ignacio. *Control of Uncertain Systems: A Linear Programming Approach*. Prentice Hall, Englewool Clifs, 1995.

[43] M. C. de Oliveira, J. Bernussou, and J. C. Geromel. A new discrete-time robust stability condition. *Systems & Control Letters*, 37:261–265, 1999.

[44] M. C. De Oliveira, J. C. Geromel, and J. Bernussou. Extended h_2 and h_∞ norm characterizations and controller parametrizations for discrete-time systems. *International Journal of Control*, 75:666–679, 2002.

[45] T.E. Djaferis. *Robust Control Design: A Polynomial Approach*. Springer, 1995.

[46] J.J. Dosch, D.J. Inman, and E. Garcia. A self-sensing piezoelectric actuator for collocated control. *Journal of Intelligent Material Systems and Structures*, 3:166–185, 1992.

[47] J. Doyle, K. Zhou, and B. Bodenheimer. Optimal control with mixed h_2 and h_∞ performance objectives. In *Proc. of American Control Conference*, pages 2065–2070. IEEE, 1989.

[48] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, 34:831–847, 1989.

[49] G.E. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach*. Springer, New York, 2000.

[50] Y. Fujisaki and Y. Kozawa. Probabilistic robust controller design: probable near minimax value and randomized algorithms. In *Proc. of 42nd IEEE Conference on Decision and Control*, pages 1938–1943, 2003.

[51] Y. Fujisaki and Y. Kozawa. Probabilistic robust controller design: Probable near minimax value and randomized algorithms. In G.C. Calafiore and F. Dabbene, editors, *Probabilistic and Randomized Methods for Design under Uncertainty*, pages 317–329. Springer, 2006.

[52] Y. Fujisaki and Y. Oishi. Guaranteed cost regulator design: A probabilistic solution and a randomized algorithm. *Automatica*, 43:317–324, 2007.

[53] K-C. Goh, M.G. Safonov, and G.P. Papavassilopoulos. Global optimization for the biaffine matrix inequality problem. *Journal of Global Optimization*, 7:365–380, 1995.

[54] K.C. Goh, M.G. Safonov, and G.P. Papavassilopoulos. Global optimization for the biaffine matrix inequality problem. *Journal of Global Optimization*, 7:365–380, 1995.

[55] M. Green and D.J.N. Limebeer. *Linear Robust Control*. Dover Publications, Incorporated, 2012.

[56] L.P. Hansen and T.J. Sargent. *Robustness*. Princeton University Press, Princeton, 2008.

[57] A. Hassibi, J. How, and S. Boyd. A path-following method for solving BMI problems in control. In *Proc. of American Control Conference*, volume 2, pages 1385–1389. IEEE, 1999.

[58] D. Hernandez, S-S. Park, R. Horowitz, and A.K. Packard. Dual-stage track-following servo design for hard disk drives. In *Proc of American Control Conference.*, volume 6, pages 4116–4121, 1999.

[59] R. Hernandez and S. Dormido. Kharitonov's theorem extension to interval polynomials which can drop in degree: a nyquist approach. *IEEE Transactions on Automatic Control*, 41:1009–1012, 1996.

[60] M. Hilbert and P. López. The world's technological capacity to store, communicate, and compute information. *Science*, 332:60 –65, 2011.

[61] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[62] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[63] C.H. Houpis. *Quantitative Feedback Theory: Fundamentals and Applications*. Taylor & Francis, New York, August 1999.

[64] B. Hredzak, G. Herrmann, and G. Guo. A proximate-time-optimal-control design and its application to a hard disk drive dual-stage actuator system. *IEEE Transactions on Magnetics*, 42:1708 –1715, 2006.

[65] F-Y. Huang, T. Semba, W. Imaino, and F. Lee. Active damping in HDD actuator. *IEEE Transactions on Magnetics*, 37:847 –849, 2001.

[66] X. Huang and R. Horowitz. Robust controller design of a Dual-Stage disk drive servo system with an instrumented suspension. *IEEE Transactions on Magnetics*, 41:2406–2413, 2005.

[67] X. Huang, R. Nagamune, R. Horowitz, and L. Yunfeng. Design and analysis of a dual-stage disk drive servo system using an instrumented suspension. In *In Proc. American Control Conference.*, volume 1, pages 535–540 vol.1, 2004.

[68] S. Ibaraki and M. Tomizuka. Rank minimization approach for solving BMI problems with random search. In *Proc. of American Control Conference*, volume 3, pages 1870–1875. IEEE, 2001.

[69] T. Iwasaki. The dual iteration for fixed-order control. *IEEE Transactions on Automatic Control*, 44:783–788, 1999.

[70] IEEE Japan. -Technical commitee for novel nanoscale servo control (NSS) systems. http://mizugaki.iis.u-tokyo.ac.jp/nss/, 2007.

[71] A.A. Kale and A.L. Tits. On kharitonov's theorem without invariant degree assumption. *Automatica*, 36:1075–1076, 2000.

[72] K. Kalyanam and T-C. Tsao. Two-period repetitive and adaptive control for repeatable and nonrepeatable runout compensation in disk drive track following. *IEEE/ASME Transactions on Mechatronics*, 17:756 –766, 2012.

[73] S. Kanev, B. De Schutter, and M. Verhaegen. An ellipsoid algorithm for probabilistic robust controller design. *Systems & Control Letters*, 49:365–375, 2003.

[74] S. Kanev, C. Scherer, M. Verhaegen, and B. De Schutter. Robust output-feedback controller design via local BMI optimization. *Automatica*, 40:1115–1127, 2004.

[75] S. Kanev and M. Verhaegen. Controller reconfiguration for non-linear systems. *Control Engineering Practice*, 8:1223–1235, 2000.

[76] M. Karpinski and A. Macintyre. Polynomial bounds for VC dimension of sigmoidal and general pfaffian neural networks. *Journal of Computer and System Sciences*, 54:169–176, 1995.

[77] P. Khargonekar and A. Tikku. Randomized algorithms for robust control analysis and synthesis have polynomial complexity. In *Decision and Control, 1996., Proceedings of the 35th IEEE*, volume 3, pages 3470–3475, 1996.

[78] P. P Khargonekar and M. A Rotea. Mixed $H_2/H\infty$ control:a convex optimization approach. *IEEE Transactions on Automatic Control*, 36:824–837, 1991.

[79] V.L. Kharitonov. Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differentsial'nye Uraveniya*, 14:61–71, 1978.

[80] M. Kočvara and M. Stingl. PENNON: a code for convex nonlinear and semidefinite programming. *Optimization Methods and Software*, 18:317–333, 2003.

[81] M. Kojima and M. Muramatsu. An extension of sums of squares relaxations to polynomial optimization problems over symmetric cones. *Mathematical Programming*, 110:315–336, 2007.

[82] V. Koltchinskii, C.T. Abdallah, M. Ariola, P. Dorato, and D. Panchenko. Improved sample complexity estimates for statistical learning control of uncertain systems. *IEEE Transactions on Automatic Control*, 45:2383–2388, 2000.

[83] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. on Optimization*, 11:796817, 2000.

[84] F. Leibfritz. COMPleib: COnstrained matrix optimization problem library. Technical report, 2004.

[85] F. Leibfritz and E. M. E Mostafa. An interior point constrained trust region

method for a special class of nonlinear semidefinite programming problems. *SIAM J. on Optimization*, 12:1048–1074, 2002.

[86] J. Lfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proc. of the CACSD Conference*, Taipei, Taiwan, 2004.

[87] A.Al. Mamun, G. Guo, and C. Bi. *Hard Disk Drive: Mechatronics and Control*. CRC Press, 1 edition, 2006.

[88] A.A. Markov. *On Certain Applications of Algebraic Continued Fraction*. PhD thesis, St Petersburg, 1884. (in Russian).

[89] A.W. Marshall. Multivariate chebyshev inequalities. *The Annals of Mathematical Statistics*, 31:1001–1014, 1960.

[90] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21:1087–1092, 1953.

[91] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44:335–341, 1949.

[92] R. Nagamune, X. Huang, and R. Horowitz. Robust H_2 synthesis for dual-stage multi-sensing track-following servo systems in HDDs. In *In Proc. American Control Conference*, pages 1284–1289, 2006.

[93] R. Nagamune, X. Huang, and R. Horowitz. Robust control synthesis techniques for multirate and multisensing Track-Following servo systems in hard disk drives. *Journal of Dynamic Systems, Measurement, and Control*, 132:1–10, 2010.

[94] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.

[95] Y. Oishi. Polynomial-time algorithms for probabilistic solutions of parameter-dependent linear matrix inequalities. *Automatica*, 43:538–545, 2007.

[96] M.L. Overton. On minimizing the maximum eigenvalue of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications*, 9:256–268, 1988.

[97] A. Packard and J. Doyle. The complex structured singular value. *Automatica*, 29:71–109, 1993.

[98] R. M Palhares, D. C.W Ramos, and P. L.D Peres. Alternative LMIs characterization of h_2 and central h_∞ discrete-time controllers. In *Proc. of the 35th IEEE Decision and Control*, volume 2, pages 1495–1496. IEEE, 1996.

[99] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. Mcgraw-Hill College, 3rd edition, 1991.

[100] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96:293–320, 2003.

[101] B. T. Polyak and R. Tempo. Probabilistic robust design with linear quadratic regulators. *Systems & Control Letters*, 43:343–353, 2001.

[102] I. Popescu. *Applications of optimization in probability, finance and revenue management*. PhD thesis, Massachusetts Institute of Technology. Operations Research Center., 1999.

[103] R.S. Sánchez-Peña and M. Sznaier. *Robust systems theory and applications*. John Wiley, New York, 1998.

[104] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13:145–147, 1972.

[105] C. Scherer, P. Gahinet, and M. Chilali. Multiobjective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, 42:896–911, 1997.

[106] D.H. Shim, H. Seong Lee, and L. Guo. Mixed-objective optimization of a track-following controller using linear matrix inequalities. *IEEE/ASME Transactions on Mechatronics*, 9:636–643, 2004.

[107] N.Z. Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23:731–734, 1987.

[108] S. Skogestad and I. Postlethwaite. *Multivariable feedback control: analysis and design*. John Wiley, New York, 1996.

[109] R.F. Stengel. Some effects of parameter variations on the lateral-directional stability of aircraft. *JOurnal of Guidence Control*, 3:124–131, 1980.

[110] M. Sznaier, C.M. Lagoa, and M.C. Mazzaro. An algorithm for sampling subsets of H_∞ with applications to risk-adjusted performance analysis and model (in)validation. *IEEE Transactions on Automatic Control*, 50:410–416, 2005.

[111] M. Talagrand. New concentration inequalities in product spaces. *Inventiones Mathematicae*, 126:505–563, 1996.

[112] R. Tempo, E. W Bai, and F. Dabbene. Probabilistic robustness analysis: explicit bounds for the minimum number of samples. In *Proc. of the 35th IEEE Decision and Control*, volume 3, pages 3424–3428, 1996.

[113] R. Tempo, G.C. Calafiore, and F. Dabbene. *Randomized Algorithms for Analysis and Control of Uncertain Systems: With Applications*. Springer, 2nd edition, 2013.

[114] R. Tempo and H. Hishii. Monte carlo and las vegas randomized algorithms for systems and control. *European Journal of Control*, 13:189–203, 2007.

[115] M. J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[116] A. Tremba, G.C. Calafiore, F. Dabbene, E. Gryazina, B. Polyak, P. Shcherbakov, and R. Tempo. RACT: randomized algorithms control toolbox

for MATLAB. In *Proc. 17th World Congress of IFAC, Seoul*, pages 390–395, 2008.

[117] J. Tyler and F. Tuteur. The use of a quadratic performance index to design multivariable control systems. *IEEE Transactions on Automatic Control*, 11(1):84–92, 1966.

[118] S. Uryasev. *Probabilistic Constrained Optimization: Methodology and Applications*. Kluwer Academic Publishers, New York, 2000.

[119] J. G. VanAntwerp and R. D. Braatz. A tutorial on linear and bilinear matrix inequalities. *Journal of Process Control*, 10(4):363–385, 2000.

[120] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[121] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[122] V.N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1 edition, 1998.

[123] V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies to their probabilities. *Theory of Probability and Its Applications*, 16:264–280, 1971.

[124] M. Vidyasagar. Statistical learning theory and randomized algorithms for control. *IEEE Control Systems*, 18:69–85, 1998.

[125] M. Vidyasagar. Randomized algorithms for robust controller synthesis using statistical learning theory. *Automatica*, 37:1515–1528, 2001.

[126] M. Vidyasagar. *Learning and Generalization: With Applications to Neural Networks*. Springer, 2nd edition, 2002.

[127] M. Vidyasagar and V. Blondel. Probabilistic solutions to some NP-hard matrix problems. *Automatica*, 37:1397–1405, 2001.

[128] T. Wada and Y. Fujisaki. Sequential randomized algorithms for robust optimization. In *Proc. 46th IEEE Conference on Decision and Control*, pages 6190–6195, 2007.

[129] M.T. White and M. Tomizuka. Increased disturbance rejection in magnetic disk drives by acceleration feedforward control and parameter adaptation. *Control Engineering Practice*, 5:741–751, 1997.

[130] J.C. Willems and R. Tempo. The kharitonov theorem with degree drop. *IEEE Transactions on Automatic Control*, 44:2218–2220, 1999.

[131] M.L. Workman. *Adaptive Proximate Time Optimal Servo-mechanism.* PhD thesis, Stanford University, United States – California, 1987. Ph.D.

[132] S-C. Wu and M. Tomizuka. Repeatable runout compensation for hard disk drives using adaptive feedforward cancellation. In *American Control Conference, 2006*, pages 382–387, 2006.

[133] S. Yabui, A. Okuyama, M. Kobayashi, and T. Atsumi. Optimization of adaptive feedforward repeatable run-out cancellation for positioning control system of hard disk drives. *Microsystem Technologies*, 18:1703–1709, 2012.

[134] H. Yamada, M. Sasaki, and Y. Nam. Control of a micro-actuator for hard disk drives using self-sensing. In *The 8th IEEE International Workshop on Advanced Motion Control, 2004.*, pages 147–152, 2004.

[135] G. Zames. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 26:301–320, 1981.

[136] K. Zhou, J. C. Doyle, and K. Glover. *Robust and optimal control*, volume 40. Prentice Hall New Jersey, 1996.

List of Publications

- Journal papers:

    1. **M. Chamanbaz**, E. Keikha, V. Venkataramanan, A. Al Mamun and W. Qing-Guo, "Design of a Probabilistic Robust Track-Following Controller for Hard Disk Drive Servo Systems", IFAC Mechatronics Journal, 2013, Accepted.
    2. **M. Chamanbaz**, F. Dabbene, R. Tempo, V. Venkataramanan, and Q.-G. Wang, "A Statistical Learning Theory Approach for Uncertain Linear and Bilinear Matrix Inequalities," Automatica, 2013, Accepted. See also arXiv:1305.4952 [math.OC].
    3. **M. Chamanbaz**, F. Dabbene, R. Tempo, V. Venkataramanan, and Q.-G. Wang, "Sequential Randomized Algorithms for Convex Optimization in the Presence of Uncertainty," IEEE Transactions on Automatic Control, 2013, Submitted. See Also arXiv:1304.2222 [cs.SY].
    4. **M. Chamanbaz**, V. Venkataramanan, and Q.-G. Wang, "Probabilistic

Analytic Center Cutting Plane Method in Robust H2 Track Following Control," Journal of Microsystems Technology, 9:1-7, 2013.

- Conference papers:

1. **M. Chamanbaz**, F. Dabbene, R. Tempo, V. Venkataramanan, and Q.-G. Wang, "Sequential Randomized Algorithms for Sampled Convex Optimization," in Proc. 2013 IEEE Multi-Conference on Systems and Control, Hyderabad, India, 2013, Accepted **(Best Student Paper Award Finalist).**

2. **M. Chamanbaz**, F. Dabbene, R. Tempo, V. Venkataramanan, and Q.-G. Wang, "On the Sample Complexity of Uncertain Linear and Bilinear Matrix Inequalities," in Proc. 2013 IEEE Conference on Decision and Control, Florence, Italy, 2013, Accepted.

3. **M. Chamanbaz**, V. Venkataramanan, and Q.-G. Wang, "Robust $\mathcal{H}_2$ Track Following Controller Based on Probabilistic Analytic Center Cutting Plane Method," in Proc. 2012 ASME-ISPS / JSME-IIP Joint International Conference on Micromechatronics for Information and Precision Equipment (MIPE2012), California, USA.

4. **M. Chamanbaz**, F. Dabbene, R. Tempo, V. Venkataramanan, and Q.-G. Wang, "A robust stability methodology for track following servo systems," in Proc. Asia Pacific Magnetic Recording Conference, 2012, pp. 1-2.

5. **M. Chamanbaz**, E. Keikha, V. Venkataramanan, Q.-G. Wang, and A. Al Mamun, "Probabilistic Robust Approach for Discrete Multi-objective Control of Track-Following Servo Systems in Hard Disk Drives," in Proc. 7$^{\text{th}}$ IFAC Symposium on Robust Control Design, Aalborg, Denmark, 2012, pp. 653-658.

6. E. Keikha, **M. Chamanbaz**, A. Al-Mamun, C.S. Bhatia "Design of track following controller of dual actuated HDD servo for 10 Tb/in2 magnetic recording," in Proc. 2nd International Conference on Control, Instrumentation and Automation (ICCIA), 2011, pp.264-269.

7. **M. Chamanbaz**, E. Keikha, V. Venkataramanan, A. Al Mamun, W. Qing-Guo, and T. Liew, "$\mathcal{H}_\infty$ probabilistic robust control of Hard Disk Drive," in Proc. IECON 2011 - 37$^{\text{th}}$ Annual Conference on IEEE Industrial Electronics Society, pp. 3394 -3399.

8. **M. Chamanbaz**, V. Venkataramanan, Q.-G. Wang, and Y. F. Liew, "Limitations to achieve high bandwidth control in hard disk drive servo systems," in Proc. Asia Pacific Magnetic Recording Conference, 2010, 2010, pp. 1-2.