The Pennsylvania State University

The Graduate School

College of Engineering

PROBABILISTIC ROBUST CONTROL SYSTEM DESIGN BY STOCHASTIC OPTIMIZATION

A Thesis in

Electrical Engineering

by

Xiang Li

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2004

We approve the thesis of Xiang Li.

Date of Signature

Constantino M. Lagoa
Associate Professor of Electrical Engineering
Thesis Adviser
Chair of Committee

Mario Sznaier
Professor of Electrical Engineering

David J. Miller
Associate Professor of Electrical Engineering

Tom M. Cavalier
Professor of Industrial Engineering

W. Kenneth Jenkins
Professor of Electrical Engineering
Head, Department of Electrical Engineering

Abstract

This dissertation concentrates on recent results on probabilistic robust controller design. In contrast to approaches taken in classical robustness theory, probabilistic robust controller design allows for a small risk of performance violation. This results, in many cases, in a significant reduction of the computational complexity of the controller design cycle and/or a significant reduction of the controller complexity even for a low level of risk of performance violation. In contrast to several of the probabilistic approaches in the control literature, we explore the problems' structure, i.e., convexity, to design more efficient algorithms. For a class of design problems which are convex in controller parameters, we introduce stochastic optimization methods to solve them. For a large class of non-convex problems, we provide a new approach which is shown to converge to the desired robust controller. This is accomplished by choosing an appropriate set of intermediate optimization variables at each iteration. Most of the results provided address the problem of designing robust output feedback controllers, where one directly determines the transfer function of the controller. Preliminary results are also presented on the design of robust static linear state feedback controllers.

Table of Contents

List of Tables x

List of Figures xi

Acknowledgments xii

Chapter 1. Introduction 1

1.1 Classical Versus Probabilistic Robust Controller Design 1

1.2 Why Consider Probabilistic Robust Design 2

1.2.1 Conservatism 2

1.2.2 Computational Complexity 2

1.3 Current State of the Art 4

1.4 Organization of the Thesis 5

Chapter 2. Probabilistically Constrained Linear Program 7

2.1 Probabilistically Constrained Linear Program 7

2.1.1 PCLP 8

2.1.2 Convexity of the Feasible Set 8

2.1.3 Example 8

2.2 Preliminaries: Log-concavity 9

2.3 Convexity of the PCLP 9

2.4 Remarks 10

2.4.1 Asymmetric Log-concave Probability Distribution Case 10

2.4.2 Joint Inequalities Case 11

2.5 Deterministic Equivalent of the PCLP 11

2.5.1 Floating Body 12

2.5.2 Additional Notation 12

2.6 Application to Control Systems Design 14

2.7 Concluding Remarks 18

Chapter 3. Probabilistic Quadratic Stabilization 19

3.1 Introduction 19

3.2 Preliminaries 20

3.3 Probabilistic Quadratic Stability 20

3.3.1 Robust Quadratic Stability 20

3.3.2 Probabilistic Quadratic Stability 21

3.4 Numerical Algorithms for Probabilistic Quadratic Stability 22

3.5 Stochastic Approximation 23

3.6 Convergence of the Stochastic Approximation Algorithm 25

3.7 Simulation Results 28

3.7.1 Example 1 29

3.7.2 Example 2 30

3.7.3 Remark 31

3.8 Concluding Remarks 31

Chapter 4. Stochastic Optimization Algorithms for Receding Horizon Control of LPV systems . . . 33
4.1 Introduction . . . 33
4.2 Preliminaries . . . 34
4.2.1 The Quadratic Regulator Problem for Constrained LPV Systems . . 34
4.2.2 Risk Adjusted Receding Horizon . . . 35
4.3 Stochastic Approximation I . . . 38
4.4 Stochastic Approximation II . . . 40
4.4.1 Remarks . . . 44
4.5 Illustrative Example . . . 45
4.6 Conclusions . . . 47

Chapter 5. Probabilistic Robust Suboptimal Output Feedback . . . 48
5.1 Introduction . . . 48
5.2 Notation and Preliminaries . . . 50
5.2.1 Notation . . . 50
5.2.2 Convex Functions and Subgradients . . . 51
5.2.3 Closed Loop Transfer Function Parametrization . . . 51
5.3 Controller Design Algorithm . . . 53
5.3.1 Problem Statement . . . 53
5.3.2 Controller Design Algorithm . . . 53
5.3.3 Remark . . . 54
5.3.4 Stopping Criterion . . . 55

5.4 Main Result 55

5.4.1 Remark 55

5.4.2 Robust Controller Gap 56

5.4.3 Proof of Theorem 5.4.1 56

5.4.4 Remark 59

5.5 Multi-Disk Design 60

5.5.1 Problem Statement 60

5.5.2 Remark 60

5.5.3 Multi-Disk Controller Design Algorithm 61

5.6 Example: Robust H_2 Design 62

5.6.1 Computing the Subgradient of Weighted H_2 Norm 62

5.6.2 Numerical Example: Single Objective 63

5.6.3 Numerical Example: Multidisk design 65

5.7 Concluding Remarks 67

Chapter 6. Probabilistic Robust Optimal Output Feedback 69

6.1 Preliminaries 69

6.2 Controller Design Algorithm I – Projection Algorithm 70

6.2.1 Problem Statement 70

6.2.2 Design Algorithm I 71

6.2.3 Convergence Theorem 72

6.3 Controller Design Algorithm 2 – Decomposition Algorithm 77

6.3.1 Problem Statement 77

6.3.2 Design Algorithm II 78

6.3.3 Convergence Theorem 78

6.4 Numerical Examples 80

6.5 Concluding Remarks 82

Chapter 7. Probabilistic Robust Stabilization 83

7.1 Introduction 83

7.2 Probabilistic Robust Stability 84

7.2.1 Robust Stability 84

7.2.2 Probabilistic Robust Stability 85

7.3 Numerical Algorithms for Probabilistic Robust Stability 85

7.4 Convergence of the Stochastic Approximation Algorithm 87

7.5 Numerical Examples 90

7.5.1 Example 1 90

7.5.2 Example 2 92

7.6 Concluding Remarks 93

Chapter 8. Conclusion and Directions for Further Research 96

8.1 Concluding Remarks 96

8.2 Directions for Further Research 97

8.2.1 Numerical Tools for PCLP 97

8.2.2 Probabilistic Design for general LMIs 98

8.2.3 More Applicable Stochastic Subgradient Algorithms 98

Appendix A . Proof of Lemma 2.3.1 99

Appendix B . Appendix: Computing Gradient of $g(\lambda_{max}(L_K(q)))$ 102

Appendix C . Proof of Lemma 3.6.1 103

Appendix D . Supermartingale Convergence Theorem 106

References 107

List of Tables

3.1 Example 1: Feasible Case 29

3.2 Example 1: Infeasible Case 30

3.3 Example 2: Feasible Case 31

3.4 Example 2: Infeasible Case 32

List of Figures

1.1 ”Icicle” Phenomenon 3

2.1 Pole Placement Assignment 18

3.1 Approximation for indicator function 24

4.1 Trajectory of States 46

4.2 Trajectory of the Performance Index. 47

5.1 Closed Loop System 49

5.2 DC Servomotor Control System 64

5.3 DC Motor: Risk History 65

5.4 MC Simulations 1: Multi-disk Design 67

5.5 MC Simulations 2: Multi-disk Design 68

6.1 MC Simulations: DC Motor Optimal Design 81

6.2 Maximal magnitude of closed loop poles 82

7.1 MC Simulations: Our Algorithm 92

7.2 MC Simulations: Calafiore-Polyak’s Stepsize 93

7.3 MC Simulations: Our Algorithm 94

7.4 MC Simulations: Calafiore-Polyak’s Step size 94

Acknowledgments

I am most indebted to my thesis adviser, Professor Constantino M. Lagoa, for the guidance, patience, encouragement and support he has shown me during my time here at Penn State. His insight and enthusiasm in research subjects have a profound impact on how I work. This will definitely carry with me in to the future. It has been a great privilege to have him as an adviser.

I am also grateful to Professor Mario Sznaier, for inspiration and enlightening discussions on a wide variety of topics. My great thankfulness also goes to Professor David J. Miller for his kind help and insightful suggestions. I sincerely thank Professor Tom M. Cavalier for serving on my dissertation committee and for his useful commentary on my work.

I want to express my most gratitude to my beloved wife Zhe, to my parents and to my sisters for all the continuous encouragement, incentive and support.

This work was supported by the National Science Foundation under Grant ECS-9984260.

Chapter 1

Introduction

The design of robust control systems has long been considered one of the more important problems in the control systems area. Well known approaches to address this problem include, among others, H_∞ theory and the structured singular value; e.g., see [47]. However, results to date are only applicable to specific uncertainty structures and/or can be conservative. To overcome these difficulties, recently a new approach, referred to as probabilistic robust design, has been developed to address the problem of robustness analysis and robust controller design within a probabilistic framework. In this dissertation, some preliminary results obtained by us in this area are presented.

1.1 Classical Versus Probabilistic Robust Controller Design

Consider a plant $P(\triangle)$ which is subject to uncertainty $\triangle$, where $\triangle \in \Delta$ and Δ is the admissible uncertainty set. Let $P_{CL}(\triangle)$ be the corresponding closed loop system and $\mathcal{P}$ denote a performance specification regarding $P_{CL}(\triangle)$; e.g., stability, bounds on overshoot, bounds on rise time. It can also represent a combination of several specifications. The objective of classical robustness theory is to design a controller such that the closed loop system $P_{CL}(\triangle)$ satisfies property $\mathcal{P}$ for all uncertainty $\triangle \in \Delta$.

In this thesis, as in probabilistic robustness theory, we take a different point of view. One starts by assuming that the uncertain parameters $\triangle \in \Delta$ are random variables. Then, the objective of probabilistic robust design is to design a controller such that

$$\text{Prob}\,\{P_{CL}(\triangle) \text{ satisfies Property } \mathcal{P}\} \geq\ 1-\varepsilon$$

where $\varepsilon > 0$ is the risk level. The rationale behind this design approach is *Borel's Law* [9]: Phenomena with very small probabilities do not occur. Put it into our scenario: if ε is small enough, the end user will not be able to differentiate between the performance of a robust controller (if there exists one) and the performance of a probabilistic robust controller.

Note that the distribution of the uncertain parameters might not have a physical meaning. If one has information about the distribution, one should use it. However, in many algorithms in this thesis, *a priori* knowledge on the distribution is not necessary. The main requirement to carry out probabilistic robust design is that uncertainty samples with a given distribution can be generated. Actually, this is a rather mild assumption. Efficient algorithms have been developed for generating both random samples of static uncertain parameters (e.g., see [14, 17, 23]) as well as dynamic uncertain parameters; see [16, 40, 37].

1.2 Why Consider Probabilistic Robust Design

The relaxation of the robust design problem described above might seem arbitrary, but there are very compelling reasons why one should consider it.

1.2.1 Conservatism

By definition, classical robustness theory concentrates on the worst case scenario. This inevitably results in *conservatism*. Take the case of *classical robustness margin* as an example. Assume that the admissible set for the uncertain real parameters $\triangle = (\triangle_1, \triangle_2, \ldots, \triangle_l)$ is a hyper-rectangle Δ_r with a fixed shape and radius $r > 0$, i.e., given a box $\Delta \subseteq R^l$

$$\Delta_r \doteq r\Delta \doteq \{r\triangle \in R^l \ : \triangle \in \Delta\}.$$

Define the *classical robustness margin*

$$r_{max} \doteq \sup\{r : \text{ Property } \mathcal{P} \text{ is satisfied for all } \triangle \in \Delta_r\}.$$

In many cases, one can go well above the margin r_{max} and only incur a very small risk of performance violation. This phenomenon has been studied by Ray and Stengel [55] [58]. Let Δ_{good} represent the set of uncertainties for which property $\mathcal{P}$ is satisfied and let Δ_{bad} represent its complement. In many cases, especially when the dimension of $\triangle$ is high, the set Δ_{bad} behaves as if it is a union of "icicles". We provide a representation of this phenomenon in Figure 1.1. In this case, for uncertainty radius r, one has such a small volume of $\triangle_{bad}$ that it may make sense to operate the system well above r_{max}.

1.2.2 Computational Complexity

Another limitation of classical robustness theory is *computational complexity*. In many situations, the number of elementary computer operations needed to perform analysis and/or

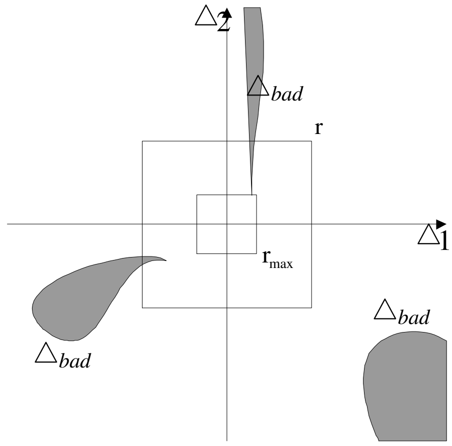


Fig. 1.1. "Icicle" Phenomenon

design a control system increases exponentially with the dimension l of the uncertainty vector. This leads to a prohibitive computational cost even for relatively low dimension of the uncertainty vector $\triangle$. The computation of the structured singular value μ illustrates this point. It has been shown that determining μ is a so-called NP-hard problem [13], which is an inherent property of the problem itself, not of any specific algorithm. This raises the necessity of properly modifying the original problem such that it is possible to develop computationally efficient algorithms. If one considers a probabilistic robust approach to system analysis, one sees that the number of elementary computations grows moderately with the problem size. More precisely, a *Monte Carlo* approach to this problem leads to a needed number of samples which is independent of the problem size [31, 62]. In this case, the computation growth is only due to the problem of checking the satisfaction of Property $\mathcal{P}$ for each sample. On the synthesis side, it has been proven that in solving a class of parameter-dependent LMI, to achieve an ε level probabilistic solution with confidence level δ, the number of iterations needed is independent of the dimension of the uncertainty vector [50]. Thus by changing our point of view to a probabilistic one, we can circumvent the "curse of dimensionality" discussed above.

The considerations above provide the motivation for the approach taken in this thesis.

1.3 Current State of the Art

The study of the application of probability to system analysis was pioneered in the work by Stengel, Ray and Marrison in [45, 55, 58]. Subsequent work can be found in [2, 18, 20, 31, 44, 62, 69, 71]. In these early works, it was shown that classical robustness margins can be very conservative.

Compared with probabilistic robust analysis, controller synthesis is still in its early stage. Two different approaches have been taken to address the problem of controller design within a probabilistic framework: The first class originates from statistical learning theory [64]. In [19, 55, 58, 70], random search algorithms are employed in robust controller design. To obtain the controller, a random sampling approach is used to search for the controller parameters. For a given risk level ε and a confidence level δ, the sample size can be determined *a priori* and shown to be independent of the dimension of uncertainty [65]. This work has also shown that classical robustness theory can be very conservative; i.e., one can greatly reduce the order of the controller and/or enlarge the admissible set of uncertainties and still have a very low risk of performance violation. Moreover, these methods are very general and applicable regardless of the convexity of the problem, the structure of the uncertainty, as long as random uncertainty samples can be generated. However, the sample size estimation can be very conservative and thus not very helpful in practice. Also the statement on quality of final controller is usually very weak due to randomization over the controller parameters.

The results in [1, 35, 36] indicate that some probabilistic robust design problems are indeed convex. Thus, it is potentially beneficial to investigate the problems' structure. More recently, a new algorithm based on the stochastic gradient method has been proposed to overcome the limitations of these early methods. The new approach has been applied to the solution of robust LMIs [15], design of state feedback controllers [49, 53] and LPV controllers [25]. Our algorithms are aligned along this line of research. As the current state of the art, some aspects of this class of algorithms have not been addressed:

1. Some control design problems are not convex in controller C but can be reformulated as a convex problem in some intermediate variables. For example, the closed loop transfer function P_{CL} is not convex in controller C. However, for a given controller C_0 and a sample of uncertainty $\triangle_0$, it is possible to find a *Youla* parameter, $Q_{C_0,\triangle_0}$, such that P_{CL} can be expressed as an affine function in $Q_{C_0,\triangle_0}$. Thus if the performance index itself is convex, the design problem still possesses a convex structure during each iterations.

2. Most algorithms are carried out in finite-dimensional space while in some cases it is more desirable to work in infinite-dimensional space such that we can obtain the controller's transfer function directly.

3. Most current probabilistic algorithms only deal with the case where the constraints are deterministic. Many problems lead to probabilistic constraints. For such cases, traditional methods do not work and new approaches have to be developed..

1.4 Organization of the Thesis

In this report, we have considered several probabilistic robust control design problems:

In Chapter 2, the Probabilistically Constrained Linear Program (PCLP), a counterpart of the classical linear program, is studied. It is shown that, for a wide class of distributions, the PCLP is a convex program. A deterministic equivalent of the PCLP is presented which provides insight on numerical implementation. The so-called *super stability* and probabilistic robust pole placement are considered within the framework of the PCLP. The examples presented suggest that one can greatly reduce the order of the controller if one is willing to accept a small well defined risk level of performance violation.

Chapter 3 concentrates on a probabilistic robust version of the well known quadratic stabilization problem for uncertain linear systems. For a wide class of probability density functions, we provide stochastic approximation algorithms which converge to its optimal performance. It is demonstrated that for small values of the risk ε, the controller gains which are required can be much smaller than their counterparts obtained via classical robust theory.

In Chapter 4, we investigate the application of stochastic optimization algorithm in receding horizon control for Linear Parameter Varying systems (LPV). To address the computational complexity problem, the receding horizon control was recast into a probabilistic framework by Sznaier [61]. We proposed two algorithms to solve this problem and proved the convergence to optimal solutions.

The focal point of Chapter 5 is the design of robust output feedback controllers for linear time-invariant uncertain systems. Given bounds on performance (defined by a convex performance evaluator), the algorithm provided converges to a controller that robustly satisfies the specifications. It is proven that the probability of performance violation tends to zero asymptotically faster than $1/k$, where k is the iteration step. Moreover, this algorithm can be applied to any uncertain plant, independently of the uncertainty structure. This algorithm is further modified to address the so-called multi-disk design problem; i.e., design a robustly stabilizing controller with guaranteed performance in a subset of the admissible uncertainty volumes.

The algorithms in Chapter 5 mentioned above are essentially suboptimal algorithms. They try to make sure that the system performance is below some given performance level. In Chapter 6, we drop the requirement of a priori knowledge of an achievable performance level. More precisely, we propose algorithms to minimize the expectation of the performance and simultaneously stabilize the closed loop systems. For the first algorithm, the assumption on achievable performance level is dropped. The second algorithm drops the assumption on the existence of a robustly stabilizing controller and optimizes performance while minimizing the set of uncertainties for which the closed loop system is unstable.

Chapter 7 deals with the robust stabilization problem within the probabilistic framework. Comparing with probabilistic robust quadratic stabilization in Chapter 3, we do not assume that the same Lyapunov function "works" for all values of the uncertainty. However, once one drops the assumption on a common Lyapunov function, the problem of static state feedback is no longer a convex problem. This problem is solved by choosing the "proper" intermediate optimization variables and solved using stochastic approximation algorithms.

Basically, we address several types of probabilistic robust controller design problems: probabilistic objective function with deterministic constraints, e.g., Chapter 3, 5, 6 and 7; deterministic objective function with probabilistic constraints, e.g., Chapter 2 and probabilistic objective function with probabilistic constraints, e.g., Chapter 4. Results obtained show that, if a small risk of performance violation is acceptable, one can solve problems for which there is no solution using classical robustness methods. Moreover, examples have been provided that illustrate the fact that even if the classical robust problem is solvable, the allowance of a small risk of performance violation can greatly reduce the complexity of the control system.

Chapter 2

Probabilistically Constrained Linear Program

In this chapter, we provide some results aimed at designing probabilistic robust controllers by using the so-called Probabilistically Constrained Linear Program (PCLP). We extend the class of probabilistic robust design problems which are known to be convex. More precisely, we extend the results in [36] and show how they can be used in a control system design context. Most of the work in this chapter has also been reported in [38].

This chapter is organized as follows: We provide the definition of Probabilistically Constrained Linear Program in Section 2.1. Section 2.2 is dedicated to the definition of the class of admissible distributions for the uncertain parameters: log-concave symmetric distributions. The main result is presented in Section 2.3 which states that the PCLP is a convex program. In Section 2.5, we provide some insights on a numerical implementation of the PCLP. Section 2.6 is dedicated to the application of the results in the context of control system design. Finally, in Section 3.8, some concluding remarks are presented.

2.1 Probabilistically Constrained Linear Program

The main result of this chapter addresses the convexity of PCLP, i.e., for a large class of probability distributions, the probabilistic version of the classical linear program is convex. The class of distributions being considered is the class of *log-concave symmetric* distributions. This class includes many of the “typical” distributions used to date in the area of probabilistic robustness, e.g., uniform distribution over convex, symmetric sets and normal distributions are members of this class. Furthermore, we show how the PCLP can be applied in a controller design context.

Indeed, consider the “classical” linear program described by

$$\min c^T x$$

subject to

$$x^T a^i \leq b_i; \quad i = 1, 2, \ldots, k$$

where $c, x, a^i \in \mathbf{R}^\ell$ and $b_i \in \mathbf{R}$, $i = 1, 2, \ldots, k$. In the PCLP framework, the constraint vectors a^i and b above are treated as random and the deterministic constraints are replaced by probabilistic constraints. There are a number of versions of the PCLP problem and the one that is used in this chapter is the same that is used in [36].

2.1.1 PCLP

Given acceptable risk levels $0 \leq \varepsilon_i \leq 1$, $i = 1, 2, \ldots, k$, find

$$\min c^T x$$

subject to

$$\text{Prob}\{x^T a^i \leq b_i\} \geq 1 - \varepsilon_i; \;\; i = 1, 2, \ldots, k$$

where $c, x \in \mathbf{R}^\ell$ and a^i, b are random vectors of appropriate dimensions.

2.1.2 Convexity of the Feasible Set

A fundamental question about the PCLP is the following: Is the PCLP a convex program? In other words, is the feasible set

$$\mathcal{X}_\varepsilon \doteq \{x \in \mathbf{R}^\ell : \text{Prob}\{x^T a^i \leq b_i\} \geq 1 - \varepsilon_i, i = 1, \ldots, k\}$$

convex? It turns out that without additional conditions on the distribution of the pair (a^i, b_i), one can easily generate examples where the answer is "no."

2.1.3 Example

Suppose $a,\ b \in \{-1, 0, 1\}$ with equal probability at each point and $x\ \in \mathbf{R}$. Take $\varepsilon = 1/3$, then

$$\begin{aligned} \mathcal{X}_\varepsilon &= \{x \in \mathbf{R} :\ \text{Prob}\{ax \leq b\} \geq 2/3\} \\ &= \{-1\} \cup \{0\} \cup \{1\}. \end{aligned}$$

In this case, $\mathcal{X}_\varepsilon$ is not a convex set.

In Appendix A, we prove that the PCLP is a convex program when the distribution of the random parameters is log-concave and symmetric; see Section 2.2 for a precise definition of this class of distributions. Convexity results are available for other kinds of distributions: In [22], it is proven that $\mathcal{X}_\varepsilon$ is convex when $0 \leq \varepsilon_i \leq 1/2$ and b_i and the components of the a^i are independent and normally distributed. This result was later extended for the case when a^i and b_i have stable distributions; e.g., see [63]. Finally, the work in [36] shows that, for $0 \leq \varepsilon_i \leq 1/2$, the PCLP is convex if the uncertain parameters are uniformly distributed over a convex symmetric set. In this chapter we extend the results in [36]. We prove that for a large class of distributions (which includes uniform distributions over convex symmetric sets), the PCLP is a convex program. Also, we show how to apply it in a controller design context.

2.2 Preliminaries: Log-concavity

Before presenting the main result in this chapter, we need to elaborate on what probability density functions are admissible for the uncertain parameters. To this end, we require the definition of log-concave functions; see [54].

DEFINITION 2.2.1. *A function $f : \mathbf{R}^\ell \to [0, \infty)$ is said to be log-concave if the following condition holds: Given any $x^0, x^1 \in \mathbf{R}^\ell$ and $\lambda \in [0, 1]$,*

$$f((1-\lambda)x^0 + \lambda x^1) \geq [f(x^0)]^{1-\lambda}[f(x^1)]^\lambda.$$

In the sequel, let $\mathcal{F}$ denote the class of log-concave symmetric probability density functions. Without loss of generality, one can assume that the center of symmetry is the origin; i.e., if $f \in \mathcal{F}$ then for any $x \in \mathbf{R}^\ell$, we have $f(x) = f(-x)$. In this chapter, we assume that the probability density function f of the vector of uncertain parameters is log-concave and symmetric; i.e., $f \in \mathcal{F}$. It is important to note that the class $\mathcal{F}$ is quite rich. Most "common" probability density functions (such as uniform or normal) are readily shown to be log-concave and symmetric. Hence, the main result to follow applies to typical density functions used in the probabilistic robustness literature to date.

2.3 Convexity of the PCLP

In this section, we study some of the properties of the PCLP. More precisely, Theorem 2.3.1 to follow indicates that, if the distribution of the uncertain parameters is log-concave and symmetric and for risk levels satisfying $0 \leq \varepsilon_i \leq 1/2$, the PCLP is a convex program. Although the result below only involves the convexity of a PCLP with one constraint, the extension to the case with an arbitrary number of constraints is immediate. This extension is a consequence of the fact that an intersection of convex sets is still convex. Throughout this chapter, we write $a^i = a_0^i + \triangle a^i; i = 1, 2, \ldots, k$ and $b = b_0 + \triangle b$ and assume that the pair $(\triangle a^i, \triangle b_i)$ has a log-concave symmetric distribution function. For simplicity, it is assumed that the vector b is deterministic; i.e., $b = b_0$. However, it is noted that the formulation and the results presented can be easily generalized for the case when b is random.

THEOREM 2.3.1. *Let $a_0 \in \mathbf{R}^\ell$, $b \in \mathbf{R}$ and the risk level $0 \leq \varepsilon \leq 1/2$ be given. Also, let the random vector $\triangle a$ have a log-concave symmetric distribution. Then, the set*

$$\mathcal{X}_\varepsilon \doteq \{x \in \mathbf{R}^\ell : Prob\{x^T(a_0 + \triangle a) \leq b\} \geq 1 - \varepsilon\}$$

is convex.

Proof: See Appendix A.

2.4 Remarks

In this section, we will show that some conditions in Theorem 2.3.1 are necessary, i.e., the distribution of $\triangle a$ must be symmetric and the theorem only holds for the case of a single inequality. In general, if these conditions are not met, one can obtain optimization problems which are not convex. Two examples are provided below.

2.4.1 Asymmetric Log-concave Probability Distribution Case

For $0 \leq \varepsilon \leq 1/2$, our focus now is on

$$\chi_\varepsilon \ \doteq \ \{x \in R^l : \text{Prob}\{x^T(a_0 + \triangle a) \leq b\} \geq 1 - \varepsilon\}.$$

Below, we provide an example that shows if the distribution of $\triangle a$ is not symmetric, χ_ε is not necessarily a convex set.

In the example, we take $x \in R^2$, $a_0 = (0\ 0)'$ and $(\triangle a_1\ \triangle a_2)'$ uniformly distributed over the set

$$\{(\triangle a_1\ \triangle a_2) \mid \triangle a_1^2 + \triangle a_2^2 \leq 1,\ \triangle a_1 \geq 0,\ \triangle a_2 \geq 0\}.$$

Take $b = 0.45$, $\varepsilon = 0.45$, $x_1 = (1\ 0)'$, $x_2 = (0\ 1)'$ and $x_3 = (0.5\ 0.5)'$. One can prove that

$$x_1 \in \chi_{0.45},\ x_2 \in \chi_{0.45}$$

while

$$\text{Prob}\{x_3^T(a_0 + \triangle a) \leq b\} = 0.52.$$

Thus x_3 which is a convex combination of x_1 and x_2 does not belong to $\chi_{0.45}$. Hence, $\chi_{0.45}$ is not a convex set.

2.4.2 Joint Inequalities Case

For $0 \leq \varepsilon \leq 1/2$, our focus now is on

$$\chi_\varepsilon \doteq \{x \in R^l : \text{Prob}\{x^T(a_0 + \triangle a) \leq b \text{ and } x^T(c_0 + \triangle c) \leq d\} \geq 1 - \varepsilon\}.$$

We now present an example that shows, in this case, χ_ε is not necessarily a convex set.

In the example, we take $x \in R^2$, $a_0 = (0\ 0)'$, $c_0 = (0\ 0)'$, $(\triangle c_1\ \triangle c_2)' = (0\ \triangle a_2)'$ and $(\triangle a_1\ \triangle a_2)'$ uniformly distributed over the set

$$\{(\triangle a_1 \triangle a_2) \mid \triangle a_1| \leq 1,\ |\triangle a_2| \leq 1\}.$$

We now take $b = d = 0.15$, $\varepsilon = 0.45$, $x_1 = (1\ 0)'$, $x_2 = (0\ 1)'$ and $x_3 = (0.5\ 0.5)'$. It can be shown that

$$x_1 \in \chi_{0.45},\ x_2 \in \chi_{0.45}$$

while

$$\text{Prob}\{x_3^T(a_0 + \triangle a) \leq b \text{ and } x_3^T(c_0 + \triangle c) \leq d\} = 0.53.$$

Hence, x_3 which is a convex combination of x_1 and x_2 does not belong to $\chi_{0.45}$. Therefore, $\chi_{0.45}$ is not a convex set.

2.5 Deterministic Equivalent of the PCLP

The result in the previous section indicates that the PCLP is a convex program. However, it does not provide any indication on how to solve the resulting optimization problem. In this section, we present the concept of *floating body* which provides some insights on how one can solve the PCLP.

2.5.1 Floating Body

Central to the results presented in this chapter is the concept of *floating body* of a probability measure. Given $0 < \varepsilon < 1/2$, the floating body K_ε of a probability distribution is a convex symmetric set for which each supporting hyper-plane "cuts-off" a set of probability ε. More precisely, given $0 < \varepsilon < 1/2$ and $u \in \mathbf{R}^\ell$, $\|u\|_2 = 1$, let $H(u,\varepsilon)$ be the supporting hyper-plane of K_ε normal to u. Also, let $\mathcal{H}^+(u,\varepsilon)$ be the half-space defined by $H(u,\varepsilon)$ which does not contain the origin. Then, K_ε is a floating body of the given probability measure if

$$\mathrm{Prob}(\mathcal{H}^+(u,\varepsilon)) = \varepsilon.$$

for all $\|u\|_2 = 1$. Not every probability measure has a floating body. However, the results in [48] indicate that every probability distribution in the class $f \in \mathcal{F}$ does have a floating body K_ε for any $0 < \varepsilon < 1/2$.

2.5.2 Additional Notation

Let $\|\cdot\|$ be a norm in $\mathbf{R}^\ell$. We define the dual norm as

$$\|x\|_* \doteq \max\{x^T y : \|y\| \leq 1\}.$$

Now, recalling that the probability distribution of Δa is log-concave and symmetric, define the norm associated with its floating body K_ε as

$$\|\triangle a\|_\varepsilon \doteq \inf\left\{\rho \in \mathbf{R}^+ : \triangle a \in \rho K_\varepsilon\right\}$$

and let $\|\cdot\|_{\varepsilon,*}$ denote its dual norm as defined above.

2.5.3 Deterministic Equivalent of the PCLP: Since

$$\{\triangle a \in \mathbf{R}^\ell : x^T(a_0 + \triangle a) \leq b\}$$

is an half-space, the definition of the floating body presented in Section 2.5.1 indicates that requiring

$$\text{Prob}\{x^T(a_0 + \triangle a) \leq b\} \geq 1 - \varepsilon$$

for $0 < \varepsilon < 1/2$ is equivalent to requiring

$$x^T(a_0 + \triangle a) \leq b$$

for all $\triangle a \in K_\varepsilon$, where K_ε is the floating body of the probability distribution of $\triangle a$ as defined in Section 2.5.1. Now, given the definition of dual norm above, this is equivalent to

$$\|x\|_{\varepsilon,*} \leq b - x^T a_0.$$

Therefore, the probabilistic constraints of the PCLP can be replaced by deterministic ones of the form above. Hence, if the quantity $\|x\|_{\varepsilon,*}$ can be easily determined, this leads to an immediate numerical implementation for solving the PCLP.

2.5.4 Elliptical Log-concave Distributions: It turns out that there are cases where $\|x\|_{\varepsilon,*}$ is easily determined. An example is the case where the probability distribution of the uncertain parameters is an *elliptical log-concave* distribution. An elliptical log-concave distribution is a distribution whose probability density function is of the form

$$f(y) = g(y^T W y)$$

where $g : \mathbf{R}_0^+ \to \mathbf{R}_0^+$ is a log-concave non-increasing function and W is a positive definite matrix. Examples of such distributions are multivariable normal distributions and uniform distributions over balls. For such probability distributions it is easy to prove that the convex floating body is an ellipsoid with the same aspect ratio as the ellipsoid

$$\mathcal{E} \doteq \{\triangle a \in \mathbf{R}^\ell : \triangle a^T W \triangle a \leq 1\}.$$

The actual "radius" of the ellipsoid K_ε can be determined analytically for some probability distributions. If one cannot determine this radius analytically, an easy one line search optimization problem can be setup to numerically obtain this value. Therefore, for such probability distributions, the PCLP reduces to a convex quadratic optimization problem. More precisely, consider an elliptical log-concave probability density function of the form above. Then, for any $0 < \varepsilon < 1/2$, the floating body K_ε is of the form

$$K_\varepsilon = \{\triangle a \in \mathbf{R}^\ell : \triangle a^T W \triangle a \leq r^2(\varepsilon)\}$$

for some $r(\varepsilon) > 0$. It can be easily shown that requiring

$$x^T(a_0 + \triangle a) \leq b$$

for all $\triangle a \in K_\varepsilon$ is equivalent to requiring

$$\|r(\varepsilon)W^{-1/2}x\|_2 \leq b - x^T a_0$$

which is a convex quadratic constraint on x.

2.6 Application to Control Systems Design

We now show how the PCLP can be used in the context of controller design. First, we apply the PCLP to the design of super stable systems. A second example shows how the theory in this chapter can be applied to robust pole assignment.

2.6.1 Super Stability: In contrast to the concept of stability, where only asymptotic behavior is considered, super stability allows for computing the worst-case value of the ℓ^∞ norm of the output due to ℓ^∞ bounded disturbances and initial conditions. It also provides an upper bound on the ℓ^∞ induced norm of the system (which is exact for FIR systems). We now briefly review some of the properties of super stable systems; see [8] and [52] for proofs and additional properties. Consider a discrete-time linear time invariant system

$$y(z) = G(z)w(z),\; G(z) = b(z)/(1 + a(z))$$

where w are exogenous disturbances, y is the output, z is the delay operator: $zx[k] = x[k-1]$ and where the polynomial $a(z)$ does not have a constant term, i.e.

$$a(z) = a_1 z + a_2 z^2 + \cdots + a_n z^n;$$

Defining $\|a\|_1 = \sum_{i=1}^{n} |a_i|$, a system is said to be super stable if $\|a\|_1 < 1$. Moreover, in [8], it is shown that in this case the ℓ^∞ induced norm of the system is bounded by

$$\|G(q)\|_{\ell^\infty \to \ell^\infty} \leq \frac{\|b\|_1}{1 - \|a\|_1}$$

This property was exploited in [8] to synthesize low order ℓ^1 controllers. Synthesizing a controller such that the ℓ^1 norm of the closed–loop system is less than or equal to a given μ reduces to finding the parameters of the controller transfer function such that

$$\mu \|d_{cl}\|_1 + \|n_{cl}\|_1 \leq \mu.$$

where d_{cl} and n_{cl} are the coefficients of the denominator and numerator of the closed loop transfer function. This problem can be easily recast in an LP format. Moreover, as noted in [8], this approach can also address the problem of fragility exhibited by some optimal control design methods [30]. Assume that the plant is subject to parametric uncertainty of the form

$$G(z) = \frac{b(z)}{1 + a(z)} = \frac{\sum_{i=0}^{m} (b_{0,i} + \triangle b_i) z^i}{1 + \sum_{j=1}^{n} (a_{0,j} + \triangle a_j) z^j}$$

where $b_{0,i}$ and $a_{0,i}$ are the nominal values of the coefficients and $\triangle b_i$ and $\triangle a_i$ represent the uncertainty. Also, consider a controller of the form[1]

$$G_c(z) = \frac{b_c(z)}{1 + a_c(z)}.$$

In this case robust performance is achieved if

$$\mu \|d_{cl}\|_1 + \|n_{cl}\|_1 \leq \mu$$

holds for all admissible values of the uncertainty, a problem that can be easily recast as finding a feasible point of a set of linear inequalities on the coefficients of the controller. However, there is a major difference between the nominal and robust performance case: while it can be

[1]For notational simplicity, here we assume that the controller is not subject to uncertainty, but the proposed procedure can be easily modified to take controller uncertainty into account.

shown that the former always admits a solution if the controller order is chosen to be at least equal to the order of the plant, the later may not have a solution even for high order controllers. On the other hand, as we illustrate next with a simple example, it might be possible to find low order risk–adjusted controllers, even for very small values of ε, the probability of violating the constraints. These controllers can be found by solving the risk-adjusted counterpart of the LP problem described in [8], which is easily seen to be a PCLP.

2.6.2 Numerical Example: We now consider the example in [8]. The discrete time system presented has nominal transfer function

$$P(z) = \frac{n(z)}{d(z)} = \frac{z - 2.5z^2 + 1.501z^3}{1 - 2.7z + 23.5z^2 - 4.6z^3}$$

and we assume that all coefficients are subject to uncertainty. Moreover, we assume that the uncertainty vector is uniformly distributed on a hyper-sphere with radius 0.05. We assume that the controller has the form

$$G_c(z) = \frac{b_c(z)}{1 + a_c(z)} = \frac{b_{c,0} + b_{c,1}z + \ldots + b_{c,m_c}z^{m_c}}{1 + a_{c,1}z + \ldots + a_{c,n_c}z^{n_c}}.$$

We first tried to design a controller that will result in a robustly super stable closed loop system. We tried controllers up until order $m_c = n_c = 6$ and were not able to find one. Then, we allowed for a risk of $\varepsilon = 1.25 \times 10^{-4}$. We were then able to find the following risk-adjusted controller

$$C(z) = \frac{4.5819 - 17.7802z - 1.0245z^2 + 0.8795z^3}{1 - 1.8819z + 0.6538z^2 + 0.287z^3}$$

which has order 3. Having these results, a Monte Carlo simulation was performed to compute the risk of violating super stability (recall that ε is the risk of violating each inequality). The number of samples used was 10^7 and the estimated probability of violating super stability obtained is 0.78%, showing that one can obtain a low order controller even for small risk levels.

2.6.3 Robust Pole Assignment: We now describe how one can apply the results of PCLP to the problem of robust pole assignment. We start with a continuous uncertain open loop plant described by the following transfer function

$$G(s) = \frac{(b_{0,0} + \triangle b_0) + (b_{0,1} + \triangle b_1)s + \cdots + (b_{0,m} + \triangle b_m)s^m}{(a_{0,0} + \triangle a_0) + (a_{0,1} + \triangle a_1)s + \cdots + (a_{0,n} + \triangle a_n)s^n}$$

where $b_{0,i}$ and $a_{0,i}$ are the nominal values of the coefficients of the numerator and denominator respectively and $\triangle b_i$ and $\triangle a_i$ represent the uncertainty. Now, since uncertainty is present, one cannot determine a controller that will assign the closed loop poles to a specific location. As in [29], one instead aims at designing a controller such that the closed loop poles lead to the satisfaction of the desired specifications. In other words, each of the coefficients of the closed loop characteristic polynomial should belong to a given interval. More precisely, given a controller of the form

$$G_c(s) = \frac{b_{c,0} + b_{c,1}s + \cdots + b_{c,m_c}s^{m_c}}{a_{c,0} + a_{c,1}s + \cdots + a_{c,n_c}s^{n_c}}$$

one aims at finding the coefficients of the controller such that the closed loop characteristic polynomial belongs to the family of polynomials

$$s^{n_{cl}} + [\delta^-_{n_{cl}-1}, \delta^+_{n_{cl}-1}]s^{n_{cl}-1} + \cdots + [\delta^-_1, \delta^+_1]s + [\delta^-_0, \delta^+_0]$$

for all admissible uncertainty values, where $n_{cl} = n_c + n$ is the degree of the closed loop characteristic polynomial. Therefore, the search for the coefficients of the controller reduces to finding a feasible solution to a set of linear inequalities to be satisfied for all admissible values of $\triangle a_0, \ldots, \triangle a_n$ and $\triangle b_0, \ldots, \triangle b_m$. For most common types of uncertainties, the problem above is easily proven to be convex. However, the designing of a robust controller can result in controllers which are complex. Therefore, we take a risk-adjusted approach; i.e., instead of requiring that each inequality is satisfied for all admissible values of the uncertain parameters, we require that the risk of violating each of the inequalities is less than or equal to a prescribed risk level ε. In other words, we solve a PCLP version of the problem above.

2.6.4 Numerical Example: The example presented here is a modification of one of the examples in [29]. Consider an uncertain plant with transfer function

$$G(s) = \frac{(0.75 + \triangle b_1)s + 1.25 + \triangle b_0}{s^2 + (0.75 + 4\triangle a_1)s + \triangle a_0}$$

where the uncertain parameter vector is uniformly distributed over the hyper-sphere of radius 0.25. We now aim at designing a controller such that the closed loop polynomial belongs to the family

$$\Delta_T(s) = s^2 + [1,3]s + [1,3].$$

Therefore, the controller transfer function is constant $G_c(s) = b_0$. We tried to find a robust controller for the system above. In this case, this was not possible. Then, a risk of $\varepsilon = 0.02$ was allowed in the PCLP version of the problem above. In this case a risk-adjusted constant controller exists and has the form $G_s(s) = 1.555$. The pole cluster distributions of the desired system and the actual closed loop system are shown in Figure 2.1. A Monte Carlo simulation was

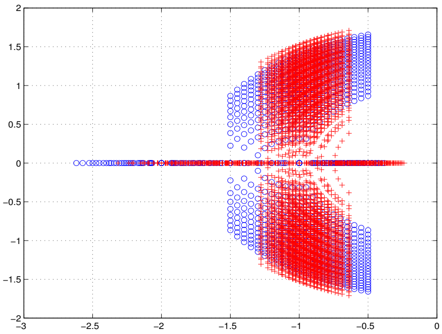


Fig. 2.1. Desired pole location "o" and actual one "+".

performed to estimate the actual risk of violating the specifications. The estimated value of the risk is approximately 3.6%, showing that, even for low risk values, one can obtain risk-adjusted controllers in cases where a robust controller does not exist. Furthermore, in this case, we obtain robust stability as an added benefit; see Figure 2.1.

2.7 Concluding Remarks

In this chapter, we extended the results in [36] and showed that the probabilistically constrained linear program is a convex optimization problem for any log-concave symmetric distribution. Also, a deterministic equivalent was provided which can be easily implemented in the case of elliptical distributions, such as normal or uniform over balls. Finally, this result was applied in the systems design context, showing that, even for very low levels of risk, one can obtain controllers that are substantially less complex than their robust counterparts.

Chapter 3

Probabilistic Quadratic Stabilization

In Chapter 2, with the aid of the concept of *floating body*, a probabilistic constrained optimization problem is converted into a deterministic one which can, in principle, be solved by well known algorithms. The key is to determine the *floating body*. At this time, this is only computationally doable for a limited class of distributions with special geometry structures, i.e., uniform or normal distribution over balls. From this chapter on, we develop algorithms based on stochastic optimization methods which are applicable to any probabilistic distributions, as long as one can generate samples with the given distribution.

The focal point of this chapter is the controller design for quadratic stability within a probabilistic framework. That is, we concentrate on a probabilistic version of the classical design problem involving the selection of a state feedback controller and associated quadratic Lyapunov function to guarantee stability of an uncertain system; e.g., see [41] for the pioneering research involving matching conditions or [3], [21] and [27] for variations, extensions and generalizations on this theme. Most of the work in this chapter has been presented in [42].

3.1 Introduction

Consistent with early literature, we consider a system with uncertain parameters

$$\triangle = (\triangle_1, \triangle_2, \ldots, \triangle_\ell),$$

$\triangle \in \Delta$, where Δ is the admissible uncertainty set. We assume that the system is described by a state space model described by the pair

$$(A_0 + \triangle A_0, B) \in \mathbf{R}^{n\times n} \times \mathbf{R}^{m\times n}$$

having nominal (A_0, B) which is assumed to be controllable and uncertainty structure of the form

$$\triangle A_0 = \triangle_1 A_1 + \triangle_2 A_2 + \cdots + \triangle_\ell A_\ell$$

with the A_i above being known $n \times n$ matrices.

As in Chapter 2, the uncertain parameters $\triangle_i$ are treated as random variables. The objective here is to design a controller leading to the maximal probability of quadratic stability, given constraints on the gain matrix. In contrast to the results in Chapter 2, where probabilistic constraints were utilized, here we aim at minimizing the risk of performance violation. Stochastic approximation algorithms are provided which maximize the probability of quadratic stability for a wide class of uncertainty probability distributions. Convergence to the optimal performance is shown.

3.2 Preliminaries

Before proceeding to the main result, we need to look at what probability density functions are admissible for the vector of uncertain parameters $\triangle$. As in Chapter 2, we focus on the log-concave functions; see Section 2.2. In this chapter, we assume that the probability density function f for the vector of uncertain parameters $\triangle$ is log-concave. Furthermore, we assume that its support is finite; i.e., there exits a $\gamma > 0$ such that $f(\triangle) = 0$ for all $\|\triangle\| > \gamma$.

We now present a classical result by Prekopa on log-concave functions which plays a crucial role on approaches developed in this chapter.

LEMMA 3.2.1. *(see [54] for proof): Assume that $f(x,y)$ is a log-concave function of $(x,y) \in \mathbf{R}^{n_1+n_2}$. Then, it follows that the function*

$$g(y) = \int_{\mathbf{R}^{n_1}} f(x,y)dx$$

is a log-concave function of $y \in \mathbf{R}^{n_2}$.

3.3 Probabilistic Quadratic Stability

In this section, we provide a precise definition of the problem addressed in this chapter. To this end, we first briefly review the concept of robust quadratic stability.

3.3.1 Robust Quadratic Stability

Consider the system

$$\dot{x} = (A_0 + \triangle A_0)x + Bu$$

with n-dimensional state x and m-dimensional control u. Assume that the nominal matrix A_0 is stable and that, as mentioned before, the uncertainty has the form

$$\triangle A_0 \; = \; \triangle_1 A_1 + \triangle_2 A_2 + \ldots + \triangle_l A_l. \tag{3.1}$$

Now, let P be a nominally determined Lyapunov matrix; i.e., provided $Q = Q^T > 0$, let $P = P^T > 0$ be the solution of Lyapunov equation

$$A_0^T P + P A_0 = -Q.$$

Given this, we say that the system above is robustly quadratically stabilizable if there exists a state feedback matrix $K \in \mathrm{R}^{m \times n}$ such that

$$L_K(\triangle) \; \doteq \; -Q + P\triangle A_0 + \triangle A_0^T P + K^T B^T P + PBK < 0 \tag{3.2}$$

for all admissible $\triangle$. In other words, the feedback control law

$$u = Kx$$

results in a quadratically stable system for all admissible values of the uncertainty. Hence, we can define the feasible set for the problem above

$$K_{QS} \doteq \{K \in \mathrm{R}^{m \times n} : L_K(\triangle) < 0 \text{ for all admissible } \triangle\}.$$

3.3.2 Probabilistic Quadratic Stability

Finding a quadratically stabilizing controller is, in general, a computationally complex problem. For example, if the admissible set Δ for the uncertain vector $\triangle$ is a polytope, finding the gain matrix K requires a sweep of the vertices of Δ. Moreover, in many cases, the resulting matrix K has very large entries, which is often undesirable. Therefore in this chapter (as in [1]) we take a different approach. Instead of trying to find a robust controller, we search for a controller gain matrix which maximizes the probability of quadratic stability. More precisely, given a compact convex set of admissible gain matrices Ω, we aim at finding

$$K^* \doteq \arg\max_{K \in \Omega} P(K) = \arg\max_{K \in \Omega} \mathrm{Prob}\{L_K(\triangle) < 0\}.$$

3.4 Numerical Algorithms for Probabilistic Quadratic Stability

The numerical algorithms proposed here are based on the well known theory of stochastic approximation; e.g. see [32] and [56]. These algorithms require the computation of the so-called stochastic gradient at each step. However, if one looks closely at the objective function $P(K)$, it is immediately seen to have the form

$$P(K) = \text{Prob}\{L_K(\triangle) < 0\} = \int_\Delta I_{\{L_K(\triangle)<0\}}(\triangle) f(\triangle) d\triangle, \tag{3.3}$$

where $I_A(\triangle)$ is the indicator function of the set A and $f(\triangle)$ is the probability density function of the uncertainty vector $\triangle$. It can be proven that $P(K)$ is a log-concave function of K [12]. However, as it can be easily seen, the gradient with respect to K of $I_{\{L_K(\triangle)<0\}}(\triangle)$ is zero for almost all $\triangle$. Therefore, traditional stochastic approximation methods cannot be directly applied to the problem at hand.

Instead of maximizing the true probability, we optimize the following approximate probability of quadratic stability

$$G(K) \doteq \int_\Delta g(\lambda_{\max}(L_K(\triangle)) f(\triangle) d\triangle$$

where $\lambda_{\max}(L)$ is the maximum eigenvalue of a symmetric matrix L and $g(\cdot)$ is a log-concave non-increasing function satisfying $g(x) = 1$ if $x \leq 0$ and $g(x) > 0$ for all x. For clarity of presentation, in the remainder of this chapter, we assume

$$g(z) = \begin{cases} 1 & \text{if } z < 0 \\ e^{-\beta z} & \text{if } z \geq 0, \end{cases} \tag{3.4}$$

where

$$z(K, \triangle) \doteq \lambda_{max}(L_K(\triangle)), \tag{3.5}$$

and β is a free design parameter. However, any function satisfying the conditions above will be a suitable "pseudo-indicator" function. Also note that for the particular $g(.)$ above we have

$$\lim_{\beta \to \infty} G(K) = P(K)$$

and therefore it can provide a good approximation of the true probability of quadratic stability. Note that the log-concavity of $g(.)$ ensures that we still have a convex optimization problem; i.e. as stated in [3], the objective function is still log-concave. We now formally present this result.

LEMMA 3.4.1. *Suppose $f(\triangle)$ is a log-concave probability density function, the objective function*

$$G(K) = \int_{\Delta} g(\lambda_{max}(L_K(\triangle)))f(\triangle)d\triangle \quad ,$$

is log-concave in $K \in R^n$, where $g(.)$ is a non-increasing log-concave function.

Proof:
By Lemma 3.2.1, we only need to consider the integrated

$$\psi(K,\triangle) \doteq f(\triangle)g(\lambda_{max}(L_K(\triangle))).$$

It is well known that $\lambda_{max}(L_K(\triangle))$ is a convex function of K and $\triangle$. Since $\log g(z)$ is a non-increasing concave function in z, composition rules indicate that $\log g(z) = \log g(\lambda_{max}(L_K(\triangle)))$ is a concave function of K and $\triangle$. Thus $g(\lambda_{max}(L_K(\triangle)))$ is log-concave in K and $\triangle$. Since log-concavity is closed under multiplication,

$$\psi(K,\triangle) = f(\triangle)g(\lambda_{max}(L_K(\triangle)))$$

is *log-concave* in K and $\triangle$. Applying Lemma 3.2.1, we conclude that

$$\int_{\Delta} \psi(K,\triangle)d\triangle = \int_{\Delta} g(\lambda_{max}(L_K(\triangle)))f(\triangle)d\triangle$$

is a log-concave function in K. This completes the proof.

3.5 Stochastic Approximation

We are now ready to present the stochastic approximation algorithms used to solve the probabilistic quadratic stability problem. Two different algorithms are proposed.

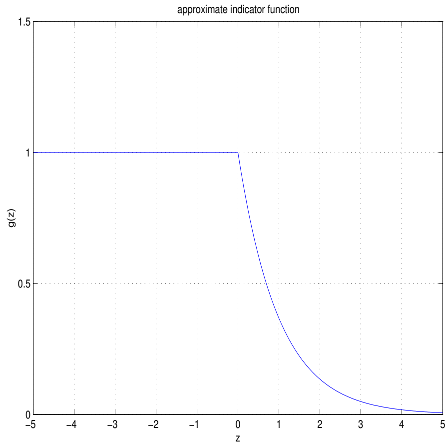


Fig. 3.1. Approximation for indicator function

The first one is related to the Robbins-Monro algorithm [56], which requires the computation of the stochastic gradient of the objective function; i.e., we consider

$$K(n+1) = \pi_\Omega \left[K(n) + a(n)\xi(n, K(n), \triangle)\right], \quad n = 1, 2, \dots \tag{3.6}$$

where

$$\xi(n) = \frac{\partial g(\lambda_{max}(L_K(\triangle)))}{\partial K}. \tag{3.7}$$

The second one, based on the Kiefer-Wolforwitz method [32], does not require the computation of the above gradient. More precisely, the quantity $\xi(n)$ is given by

$$\xi(n) = \frac{g(K(n) + r(n) + c(n), \triangle) - g(K(n) + r(n), \triangle')}{c(n)} \tag{3.8}$$

Comparing with the original presentation of the Kiefer-Wolforwitz approach, we introduce the perturbation $r(n)$, which is uniformly distributed on $[-h(n)/2, h(n)/2]$ and $\lim_{n\to\infty} h(n) = 0$. This can provide better convergence results [24]. In both of the algorithms above, π_Ω represents the projection into the admissible compact convex set Ω for the design parameter K; i.e.,

$$\pi_\Omega(y) \doteq \operatorname{argmin}\{\|y - x\|_2^2 : x \in \Omega\}$$

3.6 Convergence of the Stochastic Approximation Algorithm

We are now ready to present the stochastic approximation algorithms used to solve the probabilistic quadratic stability problem. These algorithms are straight-forward generalizations of two different algorithms; i.e., for approximate probability function of quadratic stability:

$$G(K) \doteq \int_{\Delta} g(\lambda_{\max}(L_K(\triangle))f(\triangle)d\triangle,$$

we consider

$$K(n+1) = \pi_{\Omega}\left[K(n) + a(n)\xi(n, K(n), \triangle)\right], \quad n = 1, 2, \ldots \tag{3.9}$$

where

$$\xi(n) = \frac{\partial g(\lambda_{max}(L_K(\triangle)))}{\partial K} \tag{3.10}$$

for the Robbins-Monro algorithm and

$$\xi(n) = \frac{g(K(n) + r(n) + c(n), \triangle) - g(K(n) + r(n), \triangle')}{c(n)} \tag{3.11}$$

for the Kiefer-Wolforwitz algorithm and

$$\pi_{\Omega}(y) \doteq \text{argmin}\{\|y - x\|_2^2 : x \in \Omega\}$$

Let

$$E\{\xi(n)|K(0), K(1), \ldots, K(n)\} \doteq \ G_K(K(n)) + b(n). \tag{3.12}$$

and

$$\gamma(n) \doteq \ -\langle b(n), K^* - K(n)\rangle. \tag{3.13}$$

The following conditions regarding step sizes are assumed:

1. $a(n) \geq 0$, $\sum_{n=0}^{\infty} a(n) = \infty$, and $\sum_{n=0}^{\infty} a^2(n) < \infty$;

2. $r(n)$ is uniformly distributed on $[-h(n)/2, h(n)/2]$ and $\lim_{n\to\infty} h(n) = 0$;

3. $c(n) \geq 0$;

4. $\sum_{n=0}^{\infty} a(n)\frac{c(n)}{h(n)} < \infty$.

According to [24], if $G(K)$ is locally Lipschitz, we have

$$\|b(n)\| \leq C\frac{c(n)}{h(n)}$$

and $C > 0$ is a constant. This will be used in the proof of convergence.

To prove the convergence of the algorithm, we extend a lemma from [24] to the case of log-concave objective functions. The proof is similar to that in [24] and is presented in Appendix C.

LEMMA 3.6.1. *Assume that:*

1. *G(K) is a log-concave function;*
2. *During iterations,* $G(K(n)) = \int_\Delta g(\lambda_{max}(L_{K(n)}(\triangle)))f(\triangle)\, d\triangle \ \geq p > 0$, *where p is a constant;*
3. $$a(n) \geq 0, \quad \sum_{n=0}^{\infty} a(n) \ = \infty, \quad \sum_{n=0}^{\infty} E\{a(n)|\gamma(n)| + a^2(n)\|\xi(n)\|^2\} < \infty.$$

Then, given a sequence $K(n)$ obtained using algorithm (3.9), we have

$$\lim_{n\to\infty} G(K(n)) \ = \ G(K^*)$$

with probability 1, where $K^* \in \mathcal{K}^*$.

We now present the main result:

THEOREM 3.6.1. *Take the processes (3.9) (3.10) and (3.9) (3.11) and let* $a(n) = \frac{1}{n}$, $c(n) = \frac{1}{n^{1/3}}$, $h(n) = \frac{1}{n^{1/6}}$. *Assume* $f(.)$, $g(.)$ *meet the conditions in Lemma 3.4.1, then*

$$\lim_{n\to\infty} G(K(n)) \ = \ G(K^*)$$

with probability 1, where $K^* \in \mathcal{K}^*$.

Proof: By Lemma 3.4.1, we know the objective function $G(K)$ is log-concave in K. So condition 1 of Lemma 3.6.1 is met. Furthermore, the form of "pseudo-indicator" function $g(.)$ and finiteness of $K(n)$ lead to immediate satisfaction of condition 2 of Lemma 3.6.1. As for condition 3, since $a(n) = \frac{1}{n}$, we have

$$a(n) \geq 0, \quad \sum_{n=0}^{\infty} a(n) = \infty$$

We first concentrate on algorithm (3.9) (3.10). In this case,

$$\xi(n) = \frac{\partial g(\lambda_{max}(L_K(\triangle)))}{\partial K}$$

and, hence,

$$\begin{aligned} E\{\xi(n)|K(0), K(1), \ldots, K(n)\} &= E\{\xi(n)|K(n)\} \\ &= \int_{\Delta} \frac{\partial g(K,\triangle)}{\partial K} f(\triangle) d\triangle \\ &= G_K(K(n)) \end{aligned}$$

Moreover, by (3.12), we have $b(n) = 0$. Thus, $\gamma(n) = -\langle b(n), K^* - K(n)\rangle = 0$. On the other hand, we have

$$\begin{aligned} \xi(n) &= \frac{\partial g(\lambda_{max}(L_K(\triangle)))}{\partial K} \\ &= \begin{cases} 0 & \text{if } z \leq \delta \\ -\beta e^{-\beta z} \frac{\partial z(K,\triangle)}{\partial K} & \text{if } z > \delta, \end{cases} \end{aligned}$$

where $z(K,\triangle) = \lambda_{max}(L_K(\triangle))$ and by the results in Appendix B,

$$\frac{\partial z(K,\triangle)}{\partial K_i} = y^{*T} L_{K_i}(\triangle) y^*,$$

where y^* is an eigenvector of Euclidean norm 1 associated with maximum eigenvalue of $L_K(\triangle)$. Since each entry in $L_K(\triangle)$ is finite, $\frac{\partial z(K,\triangle)}{\partial K}$ is finite and so is $\xi(n)$. Let $\xi(n) \leq M$ for some $0 < M < \infty$. Now given our choice of $a(n)$, we have

$$\begin{aligned} \sum_{n=0}^{\infty} E\{a(n)|\gamma(n)| + a^2(n)\|\xi(n)\|^2\} &= \sum_{n=0}^{\infty} E\{a^2(n)\|\xi(n)\|^2\} \\ &\leq \sum_{n=0}^{\infty} \frac{M}{n^2} \\ &< \infty \end{aligned}$$

and condition (3) is satisfied. Since all conditions of Lemma 3.6.1 are met, we conclude that for algorithm (3.9) (3.10), with probability 1

$$G(K(n)) \;\rightarrow\; G(K^*).$$

For algorithm (3.9)(3.11), we only need to be concerned with the last condition as well (the reasoning above still applies for conditions 1 and 2). Since $G(K)$ is locally Lipschitz, by [24], we have

$$\|b(n)\| \leq M_1 \frac{c(n)}{h(n)}$$

where M_1 is a constant, $c(n) = \frac{1}{n^{1/3}}$ and $h(n) = \frac{1}{n^{1/6}}$. Since the projection is carried out in the optimization process, $\|K^* - K(n)\|$ is bounded. Therefore,

$$\begin{aligned} |\gamma(n)| &= |\langle b(n), K^* - K(n)\rangle| \\ &\leq \|b(n)\|\|K^* - K(n)\| \\ &\leq \frac{M_1'}{n^{1/6}} \end{aligned}$$

Again, since $G(K)$ is locally Lipschitz, there exists a $M_2 > 0$ such that $\|\xi(n)\| \leq M_2$. Hence, we have

$$\begin{aligned} \sum_{n=0}^{\infty} E\{a(n)|\gamma(n)| + a^2(n)\|\xi(n)\|^2\} &\leq \sum_{n=0}^{\infty} E\{a(n)|\gamma(n)|\} + \sum_{n=0}^{\infty} E\{a^2(n)\|\xi(n)\|^2\} \\ &\leq \sum_{n=0}^{\infty} \frac{M_1'}{n^{7/6}} + \sum_{n=0}^{\infty} \frac{M_2^2}{n^2} \\ &< \infty, \end{aligned}$$

Hence, all conditions of Lemma 3.6.1 are met and we have the same convergence result as for Robbins-Monro algorithm.

3.7 Simulation Results

We now present two examples which illustrate the concepts put forward in this chapter.

3.7.1 Example 1

Consider a system

$$\mathbf{A_0} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1.25 & -1.5 & -2.15 \end{pmatrix}$$

and

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}'$$

where we have uncertainty in each entry of the last row of A. The uncertainty is assumed to be uniformly distributed over $[-1,1]$. First, we take $\Omega = \{K : \ |K(i)| \leq 10^5\}$, which makes the original optimization problem an approximately unconstrained one. The results obtained are presented in Tab. 3.1. We used the LMI toolbox to solve the robust quadratic stability problem. For the stochastic approximation simulation, we use $\beta = 30$ and performed 10000 iterations. We take $a(n) = 1/n$, $c(n) = 1/n^{\frac{1}{2}}$ and $h(n) = 10^{-6}/n^{\frac{1}{3}}$.

approach	K	risk
LMI	[-1.2713 -5.7110 -3.2954]	0
RM	[-0.5237 -2.0187 -1.1497]	0.37%
KW	[-0.5195 -2.0228 -1.0743]	0.44%

Table 3.1. Example 1: Feasible Case

Tab. 3.1 indicates that we can greatly reduce the gain of the controller with a very small risk of instability. In Tab. 3.2 we show the case where “harsh” constraints are put on the gains of the controller. More precisely, we assumed that $|K(i)| \leq 2$. The robust quadratical stabilization

approach	K	risk
LMI	infeasible	/
RM	[-0.3751 -2.0000 -1.3321]	0.64%
KW	[-0.5535 -1.9861 -1.0815]	0.69%

Table 3.2. Example 1: Infeasible Case

problem is infeasible in this case. However, we were able to find gains that keep the risk of instability less than 0.7%.

3.7.2 Example 2

The system is

$$\mathbf{A_0} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -5.0625 & -20.25 & -54.125 & -121.5 & -140.0625 & -101.25 & -44.25 & -10.5 \end{pmatrix}$$

and

$$\mathbf{B} = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)'$$

Uncertainty enters into the system in the same manner as previous example. We take $a(n) = 1/n$, $c(n) = 1/n^{\frac{1}{2}}$ and $h(n) = 10^{-9}/n^{\frac{1}{3}}$. First we determined the state feedback gain using "very loose" constraints; i.e. $|K(i)| \leq 10^6$. The results obtained are shown in Tab. 3.3: Again, the example shows that we can obtain a controller with very small risk of instability and with much smaller gains than the robust controller. Next, we consider the case $|K(i)| \leq 21$. The results are

approach	K	risk
LMI	[-0.4299 -10.2780 -36.0871 -46.8103 -36.6329 -17.5467 -4.9345 -0.6772]	0
RM	[-0.1970 -5.2541 -18.2222 -23.6793 -18.4798 -8.8363 -2.4995 -0.4327]	1.29%
KW	[-0.2991 -5.0299 -17.4843 -22.7072 -17.7386 -8.5683 -2.4667 -0.3618]	2.08%

Table 3.3. Example 2: Feasible Case

presented in Tab. 3.4: The LMI toolbox could not find a solution if the gains were constrained to $|K(i)| \leq 42$. Again we were able to find controllers with very small risk of instability. Both algorithms are able to find gains leading to a risk of instability less than 6%.

3.7.3 Remark

As the dimension of the gain matrix K increases, we observed that speed of convergence can be very slow. This can be circumvented by first taking small β to speed up the iteration and later increasing its value to better approximating the indicator function.

3.8 Concluding Remarks

In this chapter, the probabilistic quadratic stability design was introduced. Instead of designing a “traditional robust” controller, we provided algorithms which maximize the probability of quadratic stability. This enables us to put “harsh” restrictions on the gains of the controller and still be able to design a low risk controller even if a robust controller does not exist. Examples provided show that we can greatly reduce the gains of the controller and still have a very small risk of instability.

approach	K	risk
LMI	infeasible	/
RM	[-0.1196 -4.6077 -16.2038 -20.9997 -16.4535 -7.8822 -2.1859 -0.3027]	3.95%
KW	[-0.2219 -4.7625 -16.2999 -20.9998 -16.6144 -7.9602 -2.2924 -0.4787]	5.43%

Table 3.4. Example 2: Infeasible Case

Chapter 4

Stochastic Optimization Algorithms for Receding Horizon Control of LPV systems

In this chapter, we aim to apply stochastic optimization algorithms to receding horizon control for Linear Parameter Varying (LPV) systems. It has been shown that, by searching over a set of strategies, this problem can be reduced to finding a solution to a finite set of Linear Matrix Inequalities (LMIs) [61]. Finding the exact solution to this problem has computational complexity that grows exponentially with the horizon length. To circumvent this difficulty, a risk–adjusted receding horizon algorithm is also presented in [61] which is essentially a stochastic optimization problem. Based on these results, we propose two stochastic gradient algorithms to solve this problem in this chapter. Both algorithms are guaranteed to converge to the solution with probability one. Moreover, computational complexity grows only polynomially with system size.

4.1 Introduction

Comparing with the time invariant counterparts, there are relatively few controller design methods available for LPV systems. A popular approach is the receding horizon control [28, 46, 59, 60]. As originally shown in [59], the control strategy can be obtained by solving an optimization problem constrained by a sequence of LMIs over a moving horizon. The resulting controller is guaranteed to internally stabilize the plant and to outperform a controller designed solely on the basis of the Control Lyapunov Function (CLF), or equivalently, the solution to the set of functional Affine Matrix Inequalities (AMIs). However, the online computation burden associated with solving LMIs is a great impediment for practical application. Thus, this approach does not address the computational complexity of the problem.

In [5], a randomized algorithm is proposed in the context of Receding Horizon for the first time. In this chapter, in order to relieve the computation cost, we also apply the stochastic optimization algorithms to solving the Receding Horizon control for LPV systems. This is a direct application of stochastic gradient algorithm within the framework of the risk–adjusted receding horizon control proposed in [61]. Compared with [5], the algorithm in [61] consider the case of LPV dynamics and obtain a controller that minimizes the worst case performance, rather than its expected value, over all trajectories compatible with the current parameter value. Two gradient algorithms are introduced to solve this constrained stochastic optimization algorithms: One is based on the estimation of gradient and function value [26]; the other is a Min–Max algorithm [34].

4.2 Preliminaries

4.2.1 The Quadratic Regulator Problem for Constrained LPV Systems

Consider the following class of LPV systems:

$$\begin{aligned} x(t+1) &= A[\triangle(t)]x(t) + B_2[\triangle(t)]u(t) \\ z(t) &= C_1[\triangle(t)]x(t) + D_{12}[\triangle(t)]u(t) \end{aligned} \tag{4.1}$$

where $x \in R^{n_x}, u \in S_u \subseteq R^{n_u}$, and $z \in R^{m_z}$ represent the state, control, and regulated variables respectively, S_u is a convex set containing the origin in its interior, $\triangle$ denotes a vector of time–varying random variables that can be measured in real time, and where all matrices involved are continuous functions of $\triangle$. Further, assume that the set of admissible parameter trajectories is of the form:

$$\mathcal{F}_\Theta = \{\triangle\colon \triangle(t+1) \in \Theta[\triangle(t)],\, t = 0, 1, \ldots\} \tag{4.2}$$

where $\mathcal{P} \subset R^{n_\triangle}$ is a compact set and $\Theta\colon \mathcal{P} \to \mathcal{P}$ is a given set valued map[1]. The goal is, given an initial condition x_o, and an initial value of the parameter $\triangle_o$, to find an admissible parameter dependent state–feedback control law $u[x(t), \triangle(t)] \in S_u$ that minimizes the following performance index:

$$J(x_o, \triangle_o, u) = \sup_{\triangle \in \mathcal{F}_\Theta, \triangle(0) = \triangle_o} \sum_{k=0}^{\infty} z^T(k) z(k),\; x(0) = x_o \tag{4.3}$$

In the sequel, for simplicity, we make the following standard assumptions:

$$D_{12}^T D_{12} = I,\; C_1^T D_{12} = 0 \tag{4.4}$$

In addition, the explicit dependence of matrices on $\triangle$ will be omitted, when it is clear from the context.

[1] This is a generalization of the usual rate bounds $\underline{v}_i \leq \dot{\triangle}_i \leq \overline{v}_i$ that allows for considering for instance discrete parameter values and parameter variations with memory [57].

DEFINITION 4.2.1. *A function* $\Psi\colon R^{n_x} \times \mathcal{P} \to R_+$ *that satisfies the following condition:*

$$\max_{\triangle\in\mathcal{P}} \left\{ \min_{u\in S_u} \left\{ \max_{\theta\in\Theta(\triangle)} \Psi[A(\triangle)x + B_2(\triangle)u, \theta] - \Psi(x,\triangle) \right\} \right\} < 0 \tag{4.5}$$

is said to be a parameter dependent Control Lyapunov Function (CLF) for system (4.1).

The following result characterizes a family of CLFs in terms of the solution to a (functional) Affine Matrix Inequality [61].

LEMMA 4.2.1. *Assume that the pair* $\{A(.), C(.)\}$ *is observable for all parameter trajectories. If there exist a continuous matrix function* $Y(\triangle) > 0$ *such that for all* $\triangle \in \mathcal{P}$*, we have:*

$$\sup_{\theta\in\Theta(\triangle)} \begin{bmatrix} -Y(\triangle) + B_2(\triangle)B_2^T(\triangle) & Y(\triangle)A^T(\triangle) - B_2(\triangle)B_2^T(\triangle) & Y(\triangle)C_1^T(\triangle) \\ A(\triangle)Y(\triangle) - B_2(\triangle)B_2^T(\triangle) & -Y(\theta) & 0 \\ C_1(\triangle)Y(\triangle) & 0 & -I \end{bmatrix} \leq 0, \tag{4.6}$$

then $V(x,\triangle) = x^T Y^{-1}(\triangle)x$ *is a parameter dependent CLF for system (4.1), with associated control action given by*

$$u(x,\triangle) = -B_2^T(\triangle)Y^{-1}(\triangle)x \tag{4.7}$$

Moreover, the corresponding trajectory satisfies the following bound:

$$\sup_{\triangle\in\mathcal{F}_\Theta, \triangle(0)=\triangle_o} \sum_{k=0}^{\infty} z^T(k)z(k) \leq x_o^T Y^{-1}(\triangle_o)x_o \tag{4.8}$$

4.2.2 Risk Adjusted Receding Horizon

In this section, we provide a brief review of results in [61] which constitutes the cornerstones for algorithms in this chapter. Let $\Psi\colon R^{n_x} \times \mathcal{P} \to R_+$ be a CLF for system (4.1) such that it satisfies the additional condition:

$$\max_{\triangle\in\mathcal{P}} \left\{ \min_{u\in S_u} \left\{ \max_{\theta\in\Theta(\triangle)} \Psi[A(\triangle)x + B_2(\triangle)u, \theta] + z^T z - \Psi(x,\triangle) \right\} \right\} \leq 0 \tag{4.9}$$

and, given a horizon N, define (recursively) the following function $J(x,\triangle,n,N)$:

$$\begin{aligned} J(x,\triangle,N,N) &= \Psi(x,\triangle) \\ J(x,\triangle,i,N) &= \min_{u\in S_u}\left\{ z^T(i)z(i) + \max_{\theta\in\Theta(\triangle)} J[A(\triangle)x+B(\triangle)u,\theta,i+1,N]\right\},\ i<N \end{aligned} \tag{4.10}$$

Let $x(t),\triangle(t)$ denote the present state and parameter values, and consider the following Receding Horizon control law:

$$u_{RH}[x(t),\triangle(t)] = \operatorname*{argmin}_{u\in S_u} J[x(t),\triangle(t),t,t+N] \tag{4.11}$$

Define

$$M_n(i) \doteq$$

$$\begin{bmatrix} -X(n+i)+B_2[\triangle(n+i)]B_2^T[\triangle(n+i)] & X(n+i)A^T[\triangle(n+i)]-B_2[\triangle(n+i)]B_2^T[\triangle(n+i)] & X(n+i)C_1^T[\triangle(n+i)] \\ A[\triangle(n+i)]X(n+i)-B_2[\triangle(n+i)]B_2^T[\triangle(n+i)] & -X(n+i+1) & 0 \\ C_1[\triangle(n+i)]X(n+i) & 0 & -I \end{bmatrix} \tag{4.12}$$

where $i=0,1,\ldots,N-1$, the following receding horizon type control algorithm is proposed in [61]:

ALGORITHM 4.2.1. *0.- Data: A horizon N, a CLF $Y(\triangle)$ that satisfies (4.6).*

1.- Let $\triangle(n)$ denote the measured value of the parameter at time n and solve the following LMI optimization problem in $\mathcal{X}\in\mathbf{X}_N$:

$$\min_{\mathcal{X}\in\mathbf{X}_N} \gamma \tag{4.13}$$

subject to:

$$\begin{bmatrix} -\gamma & x^T(n) \\ x(n) & -X(n) \end{bmatrix} \leq 0$$

$$\max_{\triangle(n+i+1)\in\Theta[\triangle(n+i)]} M_n(i) \leq 0$$

i=0,1,...,N-1 with boundary condition $X(n+N) = Y[\triangle(n+N)]$*, and where, with a slight notational abuse, we denote* $X[\triangle(t),t]$ *simply as* $X(t)$*.*

2.- *At time n use as control action*

$$u[\triangle(n),x(n),N] = -B_2^T[\triangle(n)]X^{-1}[\triangle(n),n]x(n) \tag{4.14}$$

3.- *Set* $n = n+1$ *and go to step 1.*

It has been proven that the receding horizon control law (4.14) outperforms the AMI–based control (4.7). However, its computational complexity is comparable (or worse), since it requires finding both a feasible solution to (4.6) and to the set of LMIs (4.13). To reduce the computational complexity of solving the set of LMIs(4.13, a risk–adjusted point of view is taken in [61]. Assume a probability distribution for the vector $\hat{\triangle} = [\triangle(n+1)\triangle(n+2)\cdots\triangle(n+N)]$ whose probability density is nonzero for all admissible values of the scheduling variable. The results that follow in the sequel, hold for any such distribution. Then, the LMI optimization problem presented above is equivalent to

$$\begin{aligned}
&\min \mathbf{E}[\gamma] \\
&\text{subject to:} \\
&\mathbf{E}\left[\tilde{g}\left(\lambda_{\max}\left(\begin{bmatrix} -\gamma & x^T(n) \\ x(n) & -X(n)\end{bmatrix}\right)\right)\right] \leq 0 \\
&\mathbf{E}\left[\tilde{g}\left(\lambda_{\max}\left(M_n(i)\right)\right)\right] \leq 0 \\
&i = 0,1,\ldots,N-1
\end{aligned} \tag{4.15}$$

with boundary condition $X[\triangle(n+N)] = Y[\triangle(n+N)]$, where $\mathbf{E}[\,]$ denotes expectation and given $\zeta > 0$,

$$\tilde{g}(x) = \frac{e^{\zeta x} - 1}{\zeta}. \tag{4.16}$$

(4.15) is a constrained stochastic optimization problem. We provide two algorithms to solve this problem in next two sections.

4.3 Stochastic Approximation I

The optimization problem (4.13) is infinite dimensional, since the optimization is carried out over all continuous matrix functions that causally map N–length parameter trajectories to sequences of positive definite matrices. In principle, the problem can be (approximately) converted to a finite dimensional optimization by using a finite expansion $X(\triangle,t) = \sum_{i=1}^{m} X_i(t) f_i(\triangle)$, where $f_i(.)$ are known continuous functions. Let $\mathbf{x}_n$ be a vector containing all of the optimization variables; i.e., $\mathbf{x}_n$ contains γ and the entries of $X_i(t)$, $i = 0,1,\ldots,m$; $t = n,\ldots,n+N$. Now define the following functions

$$f_0(\mathbf{x}_n, \triangle(n), \hat{\triangle}) \doteq \gamma; \tag{4.17}$$

$$f_1(\mathbf{x}_n, \triangle(n), \hat{\triangle}) \doteq \tilde{g}\left(\lambda_{\max}\left(\begin{bmatrix} -\gamma & x^T(n) \\ x(n) & -X(n) \end{bmatrix}\right)\right)$$

and

$$f_{i+2}(\mathbf{x}_n, \triangle(n), \hat{\triangle}) \doteq \tilde{g}\left(\lambda_{\max}\left(M_n(i)\right)\right); \quad i = 0,1,\ldots,N-1$$

We can now define the approximate optimization problem

$$\begin{aligned} &\min \mathbf{E}\left[f_0(\mathbf{x}_n, \triangle(n), \hat{\triangle})\right] \\ &\text{subject to:} \\ &\mathbf{E}\left[f_l(\mathbf{x}_n, \triangle(n), \hat{\triangle})\right] \leq 0; \quad l = 1,2,\ldots,N+1 \end{aligned} \tag{4.18}$$

One can easily see that the solution of problem (4.18) tends to the solution of problem (4.15) as $\zeta \to \infty$. We are now ready to provide an algorithm for solving the approximate problem (4.18). Note that the function $\tilde{g}(\lambda_{max}(P))$ is convex in matrix P, so the optimization problem above is convex. For technical reasons, in the sequel we will assume that the solution to this problem is known to belong to a given compact convex set $\mathcal{X}$ (where the matrices $X(n+i)$ have bounded entries and are positive definite). Let $\pi(\cdot)$ denote the projection onto $\mathcal{X}$; i.e.,

$$\pi(\mathbf{x}) = \arg \min_{\tilde{\mathbf{x}} \in \mathcal{X}} \|\mathbf{x} - \tilde{\mathbf{x}}\|_2.$$

and consider the following algorithm:

ALGORITHM 4.3.1.

1.- *Generate a feasible solution* $Y(\triangle)$ *to (4.6), using for instance the procedure proposed in [25].*

2.- *Initialization: Determine* $\mathbf{x}_n^0$, y_i^0; $i = 0,1,2,\ldots,N+1$ *and* z_j^0; $j = 1,2,\ldots,N+1$. *Let* $k = 0$.

3.- *Generate a sample* $\hat{\triangle}^k = [\triangle^k(n+1),\ldots,\triangle^k(n+N-1)]$

4.- *Let* j^* *be such that* $z_{j^*}^k = \max_j z_j^k$. *If* $z_{j^*}^k \leq -\tau_0/k^\tau$ *then*

$$\mathbf{x}_n^{k+1} = \pi\left[\mathbf{x}_n^k - b_k y_0^k\right].$$

Otherwise,

$$\mathbf{x}_n^{k+1} = \pi\left[\mathbf{x}_n^k - b_k y_{j^*}^k\right].$$

5.- *Let*[2]

$$y_i^{k+1} = y_i^k + a_k\left(\left.\frac{\partial f_i(\mathbf{x}_n,\triangle(n),\hat{\triangle}^k)}{\partial \mathbf{x}_n}\right|_{\mathbf{x}_n=\mathbf{x}_n^k} - y_i^k\right); \quad i = 0,1,2,\ldots,N+1$$

and

$$z_j^{k+1} = z_j^k + a_k\left(f_j(\mathbf{x}_n,\triangle(n),\hat{\triangle}^k) - z_j^k\right); \quad j = 1,2,\ldots,N+1.$$

6.- *If* $\max_j z_j^l < 0$ *for* $l = k - N_{good}+1,\ldots,k+1$ *and* $|\tau_{k-1}-\tau_k| < \varepsilon$ *stop. Otherwise, let* $k = k+1$ *and go to step 3.*

THEOREM 4.3.1. *Let*

$$a_k = \frac{\alpha_0}{k^\alpha}; \quad b_k = \frac{\beta_0}{k^\beta}$$

where α_0, α, β_0 *and* β *are positive constants. Furthermore, assume that* τ_0 *and* τ *are also positive. Then, if*

$$\sum_{k=0}^{\infty} a_k = \sum_{k=0}^{\infty} b_k = \infty; \quad \sum_{k=0}^{\infty} a_k^2 < \infty; \lim_{k\to\infty}\frac{b_k}{a_k} = 0$$

[2] *For the computation of the subgradient of* $\lambda_{\max}(M)$ *see Appendix B*

and

$$2\beta - \alpha - 2\tau > 1$$

then, the sequence $\mathbf{x}_n^k$ *converges with probability one to the solution of the problem (4.18).*

Proof Direct application of Theorem 1 in [26]

4.4 Stochastic Approximation II

In this section, we will address another kind of stochastic approximation which was developed by Kushner and Sanvicente [34]. It is essentially a min–max algorithm for constrained stochastic optimization problems. For the approximate problem described in (4.18), we have the following Lagrangian function for the optimization problem at time instant n:

$$L(x_n, \lambda_n) = \mathbf{E}\left[f_0(\mathbf{x}_n, \triangle(n), \hat{\triangle})\right] + \sum_{i=1}^{N+1} \lambda_{ni} \mathbf{E}\left[f_i(\mathbf{x}_n, \triangle(n), \hat{\triangle})\right] \tag{4.19}$$

Also for the convenience of notation, we define

$$l(x_n, \lambda_n) = f_0(\mathbf{x}_n, \triangle(n), \hat{\triangle}) + \sum_{i=1}^{N+1} \lambda_{ni} f_i(\mathbf{x}_n, \triangle(n), \hat{\triangle}), \tag{4.20}$$

which is a noisy estimate of $L(x_n, \lambda_n)$. Assume there exists a pair $(\bar{x}, \overline{\lambda})$, such that

$$L(\bar{x}, \lambda) \ \leq L(\bar{x}, \overline{\lambda}) \ \leq L(x, \overline{\lambda})$$

and there is some known A, $0 < A < \infty$, so that

$$|\bar{x}_i| < A, \ \ 0 \leq \overline{\lambda}_i < A.$$

We will construct an iteration process which simultaneously maximizes $L(x_n, \lambda_n)$ with respect to λ_n and minimizes it with respect to x_n.

ALGORITHM 4.4.1.

1.- Generate a feasible solution $Y(\triangle)$ to (4.6), using for instance the procedure proposed in [25].

2.- Initialization: Determine $\mathbf{x}_n^0$, λ_{ni}^0 *and maximal number of iterations* N_n; $i = 1, 2, \ldots, N+1$.

3.- Generate a sample $\hat{\triangle}^k = [\triangle^k(n+1), \ldots, \triangle^k(n+N-1)]$

4.- Carry out the following iterations

$$\tilde{x}_n^{k+1} = x_n^k - a_k\,(\frac{\partial f_0}{\partial x_n}|_{x_n^k} + \sum_{i=1}^{N+1} \frac{\partial f_i}{\partial x_n}|_{x_n^k}).$$

and

$$\tilde{\lambda}_n^{k+1} = \max[\,0,\ \lambda_n^k + a_k \sum_{i=1}^{N+1} f_i(x_n^k, \triangle(n), \hat{\triangle}^k)].$$

5.- Projection:

$$x_n^{k+1} = \begin{cases} \tilde{x}_n^{k+1} & \text{if } |\tilde{x}_n^{k+1}| \le A \\ A & \text{if } \tilde{x}_n^{k+1} > A \\ -A & \text{if } \tilde{x}_n^{k+1} < -A, \end{cases}$$

$$\lambda_n^{k+1} = \begin{cases} \tilde{\lambda}_n^{k+1} & \text{if } \tilde{\lambda}_n^{k+1} \le A \\ A & \text{if } \tilde{\lambda}_n^{k+1} > A, \end{cases}$$

6.- If N_n *is achieved, stop. Otherwise, let* $k = k+1$ *and go to step 3.*

Algorithm 4.4.1 is very similar to the one in [34]. But since the gradients of the integrands in formulation (4.19) can be computed efficiently, the Robbins-Monro algorithm can be applied directly.

THEOREM 4.4.1. *Let*

$$\sum_{k=1}^{\infty} a_k = \infty,\ \sum_{k=1}^{\infty} a_k^2 < \infty,\ a_k > 0$$

and assume

1. $f_0(.)$ *is strictly convex and* $f_i(.)$, $i = 1, 2, \ldots, N+1$ *are convex.*

2. $f_i(.)$, $i = 1, 2, \ldots, N+1$ *are continuously differentiable*

3. *The set* $C \doteq \{x : \mathbf{E}[f_i(.)] \leq 0,\ i = 1, 2, \ldots, N+1\}$ *contains a nonempty interior.*

4. *There is a saddle point* $(\bar{x}_n, \overline{\lambda}_n)$ *of* $L(x_n, \lambda_n)$ *such that*

$$|\bar{x}_n(i)| < A,\ \ 0 \leq \overline{\lambda}_n(i) < A.$$

Then, Algorithm 4.5.1 produces a sequence x_n^k *which converges to the optimal solution* $\bar{x}_n$ *as* $k \to \infty$ *with probability one.*

Proof: Let $(\bar{x}_n, \overline{\lambda}_n)$ be the saddle point for $L(x_n, \lambda_n)$ and $\bar{x}_n < A$, $0 \leq \overline{\lambda}_n < A$. By Algorithm 4.4.1, we have

$$|\tilde{\lambda}_n^{k+1} - \overline{\lambda}_n|^2 \ \leq\ |\lambda_n^k - \overline{\lambda}_n|^2 + 2a_k(\lambda_n^k - \overline{\lambda}_n)' l_{\lambda_n}(x_n^k, \lambda_n^k) + a_k^2 |l_{\lambda_n}(x_n^k, \lambda_n^k)|^2 \tag{4.21}$$

and

$$|\tilde{x}_n^{k+1} - \bar{x}_n|^2 \ \leq\ |x_n^k - \bar{x}_n|^2 - 2a_k(x_n^k - \bar{x}_n)' l_{x_n}(x_n^k, \lambda_n^k) + a_k^2 |l_{x_n}(x_n^k, \lambda_n^k)|^2. \tag{4.22}$$

Let $\mathcal{B}_{nk}$ denote the smallest σ-algebra generated by $\tilde{x}_n^1, \ldots, \tilde{x}_n^k$, $\tilde{\lambda}_n^1, \ldots, \tilde{\lambda}_n^k$. According to (4.19) and (4.20), let

$$l_{\lambda_n}(x_n^k, \lambda_n^k) = L_{\lambda_n}(x_n^k, \lambda_n^k) + \nu_{\lambda_n}(x_n^k, \lambda_n^k)$$

and

$$l_{x_n}(x_n^k, \lambda_n^k) = L_{x_n}(x_n^k, \lambda_n^k) + \nu_{x_n}(x_n^k, \lambda_n^k),$$

where $\mathbf{E}[\nu_{\lambda_n}(x_n^k, \lambda_n^k)|B_{nk}] = \mathbf{E}[\nu_{x_n}(x_n^k, \lambda_n^k)|B_{nk}] = 0$. Take expectations conditioned on $\mathcal{B}_{nk}$ on both sides of (4.21) and (4.22), we get

$$\mathbf{E}[|\tilde{\lambda}_n^{k+1} - \overline{\lambda}_n|^2 |\mathcal{B}_{nk}] \quad \leq \quad |\lambda_n^k - \overline{\lambda}_n|^2 + 2a_k(\lambda_n^k - \overline{\lambda}_n)' L_{\lambda_n}(x_n^k, \lambda_n^k) + a_k^2 \mathbf{E}[|l_{\lambda_n}(x_n^k, \lambda_n^k)|^2 |\mathcal{B}_{nk}]$$

and

$$\mathbf{E}[|\tilde{x}_n^{k+1} - \bar{x}_n|^2 |\mathcal{B}_{nk}] \quad \leq \quad |x_n^k - \bar{x}_n|^2 - 2a_k(x_n^k - \bar{x}_n)' L_{x_n}(x_n^k, \lambda_n^k) + a_k^2 \mathbf{E}[|l_{x_n}(x_n^k, \lambda_n^k)|^2 |\mathcal{B}_{nk}].$$

Define

$$|Z_n^k - \overline{Z}_n|^2 \ \doteq\ |x_n^k - \bar{x}_n|^2 + \ |\lambda_n^k - \overline{\lambda}_n|^2.$$

and

$$Q(x_n^k, \lambda_n^k) \ \doteq\ (x_n^k - \bar{x}_n)' L_{x_n}(x_n^k, \lambda_n^k) - (\lambda_n^k - \overline{\lambda}_n)' L_{\lambda_n}(x_n^k, \lambda_n^k)$$

where

$$\begin{aligned} Q(x_n^k, \lambda_n^k) &\geq [L(x_n^k, \lambda_n^k) - L(\bar{x}_n, \lambda_n^k)] - [L(x_n^k, \lambda_n^k) - L(x_n^k, \overline{\lambda}_n)] \\ &\geq 0. \end{aligned}$$

By carrying out the projection as stated in Algorithm 4.4.1, we have

$$|x_n| \leq A \quad \text{and} \quad 0 \leq \lambda_n \leq A.$$

The entries in the matrix as defined in (4.12) and (4.15) are all finite, and so are the eigenvalues which are the arguments of $f_i(.)$ as described in (4.17), $i = 1, 2, \ldots, N+1$. Since $f_i(.)$ are all continuous by Assumption 1 in Theorem 4.4.1, there exists $A_2 > 0$ such that

$$|f_i| \; < \; A_2, \;\; i = 1, 2, \ldots, N+1.$$

Similarly, by Assumption 2 in Theorem 4.4.1, it is easy to see

$$\left| \frac{\partial f_i}{\partial x_n} \right| \; < \; A_1, \;\; i = 0, 1, \ldots, N+1,$$

where A_1 is a positive constant. By assumption 4, the point $(x_n^{k+1}, \lambda_n^{k+1})$ is no farther from $(\bar{x}_n, \overline{\lambda}_n)$ than is the point $(\tilde{x}_n^{k+1}, \tilde{\lambda}_n^{k+1})$ and therefore

$$\begin{aligned} \mathbf{E}[|Z_n^{k+1} - \overline{Z}_n|^2 | \mathcal{B}_{nk}] - |Z_n^k - \overline{Z}_n|^2 &\leq \mathbf{E}[|\tilde{Z}_n^{k+1} - \overline{Z}_n|^2 | \mathcal{B}_{nk}] - |Z_n^k - \overline{Z}_n|^2 \\ &\leq -2a_k Q(x_n^k, \lambda_n^k) + a_k^2 (\mathbf{E}[|l_{\lambda_n}(x_n^k, \lambda_n^k)|^2 | \mathcal{B}_{nk}] \\ &\quad + \mathbf{E}[|l_{x_n}(x_n^k, \lambda_n^k)|^2 | \mathcal{B}_{nk}]) \\ &\leq a_k^2 (\mathbf{E}[|l_{\lambda_n}(x_n^k, \lambda_n^k)|^2 | \mathcal{B}_{nk}] + \mathbf{E}[|l_{x_n}(x_n^k, \lambda_n^k)|^2 | \mathcal{B}_{nk}]) \\ &\leq a_k^2 [NA^2 A_2^2 + (1 + NA^2) A_1^2]. \end{aligned}$$

According to the definition in Appendix C, the series $\{Z_n^k\}$ is a stochastic quasi-feyer sequence. By Lemma C.0.1 in Appendix C, the sequence $|Z_n^k - \overline{Z}_n|^2$ converges with probability 1 for any

Z_n^k, such that $\mathbf{E}[|Z_n^k - \overline{Z}_n|^2] < C < \infty$. Define the sets

$$\begin{aligned} N_\varepsilon &= \{x: |x-\overline{x}_n| \le \varepsilon\} \\ C_{3\varepsilon} &= \{x: 2\varepsilon \le |x-\overline{x}_n| \le 3\varepsilon\} \\ N_{3\varepsilon} &= \{x: |x-\overline{x}_n| \le 3\varepsilon\} \end{aligned}$$

Since $f_0(.)$ is strictly convex and $f_i(.),\ i=1,2,\ldots,N_1$ are convex,

$$Q(x_n^k, \lambda_n^k) > 0 \qquad \forall x_n^k \ne \overline{x}_n.$$

Thus, for each $\varepsilon > 0$, there is a $\delta_\varepsilon > 0$ such that

$$Q(x_n^k, \lambda_n^k) \ge \delta_\varepsilon \quad \text{if} \quad |x_n^k - \overline{x}_n| \ge \varepsilon.$$

Take expectations on both sides of the above inequality, we get

$$\mathbf{E}[|Z_n^k - \overline{Z}_n|^2|] - \mathbf{E}[|Z_n^0 - \overline{Z}_n|^2] \le -2\mathbf{E}\sum_{i=0}^{k-1} a_i Q(x_n^i, \lambda_n^i) + \mathbf{E}\sum_{i=0}^{k-1} a_i^2 (|l_{\lambda_n}(x_n^i, \lambda_n^i)|^2 + |l_{x_n}(x_n^i, \lambda_n^i)|^2)$$

By conditions on a_i as stated in the theorem, for any given $\varepsilon > 0$, we at least have $x_n^i \in N_\varepsilon$ infinitely often with probability 1 or $\sum_{i=0}^{k-1} a_i Q(x_n^i, \lambda_n^i) \to \infty$ on some non-null set. Next, following exactly the same logic in [34], we can prove

1. Asymptotically, x_n^k cannot leave $N_{3\varepsilon}$ from N_ε without going through $C_{3\varepsilon}$.

2. $C_{3\varepsilon}$ can only be entered finite number of times w.p.1..

Since ε is arbitrary, we can prove the convergence. Q.E.D.

4.4.1 Remarks

1. Comparing to (4.17), to make the objective function to be strictly convex, one can use $f_0 = \gamma^2$ instead of $f_0 = \gamma$. Since $\mathbf{E}^2[\gamma] \le \mathbf{E}[\gamma^2]$, we actually minimize the upper-bound of the original objective function .

2. The choice of $\tilde{g}$ as expressed in (4.16) results in an immediate satisfaction of Assumption 2 of Theorem 4.4.1.

4.5 Illustrative Example

Consider a discrete time LPV system with the following state space realization

$$A(\triangle(t)) = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 - 0.1\triangle(k) \end{bmatrix}$$

$$B_2(\triangle(t)) = \begin{bmatrix} 0 & 0.0787 \end{bmatrix}'$$

$$C_1(\triangle(t)) = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}$$

$$D_{12}(\triangle(t)) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix}'$$

with admissible parameter set

$$\mathcal{F}_\Theta = \{\triangle(t) \colon \triangle(t+1) \in [-1,1],\, t = 0,1,\ldots\}. \tag{4.23}$$

As a first step, we computed a CLF for the system above. It can be verified that the following matrix function $Y(\triangle)$ satisfies the required AMIs (4.6)

$$Y(\triangle_k) = Y_0 + Y_1\triangle_k + Y_2{\triangle^2}_k$$

$$Y_0 = \begin{bmatrix} 0.0296 & -0.0196 \\ -0.0196 & 0.0290 \end{bmatrix}$$

$$Y_1 = \begin{bmatrix} 0.0003 & -0.0016 \\ -0.0016 & 0.0044 \end{bmatrix}$$

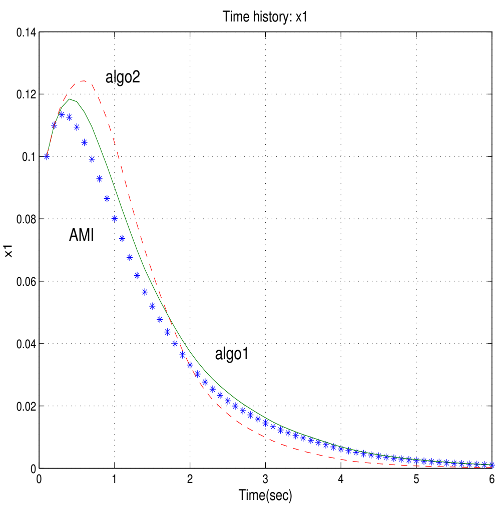
$$Y_2 = 10^{-3} \begin{bmatrix} -0.0691 & 0.0302 \\ 0.0302 & 0.2542 \end{bmatrix}$$

and, hence, $V(x,\triangle) = x^T Y^{-1}(\triangle)x$ is a parameter dependent CLF for the plant.

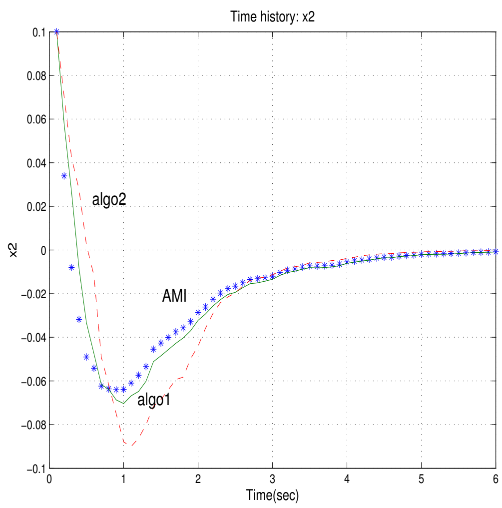
Algorithms 4.3.1 and 4.4.1 were then applied to compute the control action. More precisely, the initial condition chosen was $x_0 = [0.1\ 0.1]'$ and a control action of the form (4.14) was used where, at each time instant, $X(n)$ was computed using Algorithm 4.3.1 and Algorithm 4.4.1 respectively. The values chosen for the parameters were horizon $N = 10$, $\zeta = 20$. For Algorithm 4.3.1: $\alpha = 0.6$, $\alpha_0 = 1$, $\beta = 1$, $\beta_0 = 10^{-3}$, $\tau = 0.15$ and $\tau_0 = 10^{-6}$. For Algorithm 4.4.1: $a_k = 1/k$. Furthermore, the parameterization used was

$$X(\triangle_k, k) = X_0(k) + X_1(k)\triangle_k + X_2(k)\triangle_k{}^2.$$

Finally, we used a uniform distribution for $\triangle$. At each time instant, we ran 1000 iterations of the stochastic optimization algorithm. The results obtained are depicted in Figures 4.1(a) and 4.1(b) where we compare the state trajectories of the proposed probabilistic robust controllers with the state trajectory obtained using the AMI-based controller (4.7). In Figure 4.2, we compare the time history of the performance index. The probabilistic robust receding horizon controllers yield costs $J_1 = 1.0980$ for Algorithm 4.3.1 and $J_2 = 1.1369$ for Algorithm 4.4.1 versus a cost of $J = 1.3191$ for the AMI-based controller; i.e., the risk adjusted controllers yield roughly 18% and 17% improvement on the performance.



(a) First State Variable



(b) Second State Variable

Fig. 4.1. Trajectory of States

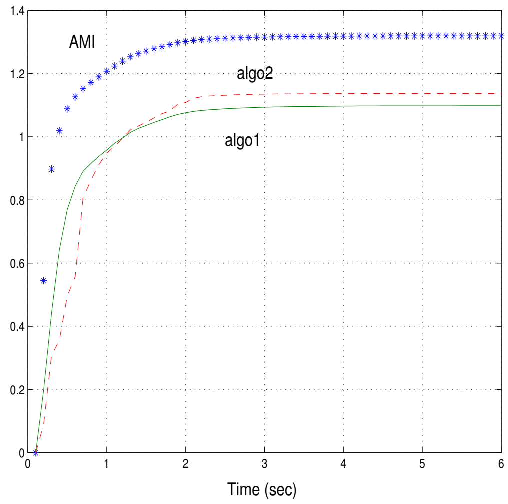


Fig. 4.2. Trajectory of the Performance Index.

4.6 Conclusions

The computational complexity in control of LPV systems is an issue which has not been fully addressed. Many existed algorithms involve on–line solution of a set of functional LMIs which scales exponentially with system size [31]. To relieve the computation cost, the concept of risk-adjusted receding horizon control was proposed in [61] where it showed the receding horizon control synthesis can be formulated as a stochastic optimization problem. In this chapter, we applied two stochastic gradient algorithms to solve this problem and proved the convergence to optimal solutions.

These results were illustrated with a simple example where probabilistic robust receding horizon controllers were used to control a second order LPV plant. As shown, the proposed probabilistic robust receding horizon controllers improve performance vis-a-vis a conventional LPV controller, while substantially reducing the computational effort required by a comparable Receding Horizon controller.

Chapter 5

Probabilistic Robust Suboptimal Output Feedback

Convexity plays a very important role in the field of optimization. In the previous three chapters, all the design problems considered can be formulated as an optimization problem which is convex in the design parameters, e.g., the coefficients of controller's transfer functions, the state feedback gain, etc. The same statement cannot be made about large classes of controller design problems. In the remainder of this thesis, we focus on some of these non-convex design problems. By carefully choosing the intermediate optimization variables, we develop algorithms that are proven to solve the non-convex design problems considered in this thesis. In this chapter, we address the problem of robust output feedback controller design for linear systems with arbitrary dependence on the uncertain parameters.

5.1 Introduction

Consider an uncertain plant $G(z,\triangle)$, where $\triangle \in \Delta$ represents uncertainty and Δ is the uncertainty support set. The uncertainty can be either static or dynamic and no assumption is made on the way $G(z,\triangle)$ depends on $\triangle$. The only assumption is that one can generate random samples $\triangle \in \Delta$. Throughout this chapter, we focus on the closed loop system in Figure 5.1, whose closed loop transfer function is denoted by $T_{CL}(z,\triangle,C)$. Given a convex objective function $g(.)$ whose subgradient can be computed and a performance level γ, the objective is to design a controller $C^*(z)$ such that

$$g\left[T_{CL}(z,\triangle,C^*)\right] \leq \gamma$$

for all $\triangle \in \Delta$. It should be noted that g can represent both a single specification or a set of specifications. If one wants $g_i\left[T_{CL}(z,\triangle,C)\right] \leq \gamma_i$ for $i = 1,2,\ldots,n$, just take

$$g\left[T_{CL}(z,\triangle,C)\right] = \max_i \frac{g_i\left[T_{CL}(z,\triangle,C)\right]}{\gamma_i}$$

and $\gamma = 1$. Given the probability measure underlying the random samples generation, the algorithm provided in this chapter produces a sequence of controllers $C_k(z)$ having the property that

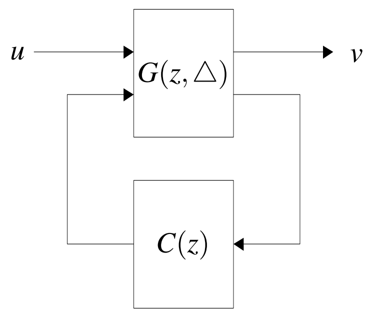


Fig. 5.1. Closed Loop System

the risk of violating the performance specifications

$$P_k \doteq \text{Prob}\{g\left[T_{CL}(z,\triangle,C_k)\right] > \gamma\}$$

tends to zero as $k \to \infty$. Moreover, a bound on the decay rate is given. More precisely, it is proven that

$$\sum_{k=0}^{\infty} P_k < \infty.$$

Hence, P_k tends to zero asymptotically faster than $1/k$.

The general nature of the algorithm provided, enables one to address many problems in robust controller design. In particular, these procedure can be used to solve the open problem of robust H_2 controller design in the presence of causal uncertainty. An example illustrating this particular instance of our algorithm is also provided. Most results in this chapter are also presented in [37] and [39].

This chapter is organized as follows: In Section 5.2, the notation used is introduced and some ancillary results are provided. The problem formulation and the algorithm itself are presented in Section 5.3. The main result concerning the convergence of the algorithm is provided and proven in Section 5.4. As an extension, a multidisk design algorithm is developed in Section 5.5. These algorithms are applied to the design of robust H_2 controllers in Section 5.6 and concluding remarks are presented in Section 5.7.

5.2 Notation and Preliminaries

We now state the notation used throughout this chapter as well as some standard results needed for the presentation of our results.

5.2.1 Notation

Let $H_2^{n\times m}$ denote the Hilbert space of functions $H : \mathbf{C} \to \mathbf{C}^{n\times m}$ analytic in the set $\{z \in \mathbf{C} : |z| \geq 1\}$, equipped with the inner product

$$\langle H, T\rangle = \frac{1}{2\pi}\int_0^{2\pi} \mathrm{Re}\{\mathrm{Trace}[H(e^{j\theta})^* T(e^{j\theta})]\} d\theta$$

where Re denotes the real part, Trace(A) is the trace of the matrix A and A^* denotes the conjugate transpose of A. Hence, the H_2 space has norm

$$\|T\|_2 = \left(\frac{1}{2\pi}\int_0^{2\pi} \mathrm{Re}\{\mathrm{Trace}[T(e^{j\theta})^* T(e^{j\theta})]\} d\theta\right)^{\frac{1}{2}}$$

Also, let RH_2 denote the subspace of all rational functions in H_2 analytic in $\{z \in \mathbf{C} : |z| \geq 1\}$.

Moreover, define the space $\mathcal{G}$ as the space of rational functions $G : \mathbf{C} \to \mathbf{C}^{n\times m}$ that can be represented as

$$G(z) = G_s(z) + G_u(z).$$

where $G_s(z)$ is analytic in the set $\{z \in \mathbf{C} : |z| \geq \alpha\}$ and $G_u(z)$ is strictly proper and analytic in the the set $\{z \in \mathbf{C} : |z| < \alpha\}$ and $0 < \beta < \alpha < 1$. Now, given two functions $G, H \in \mathcal{G}$, define the distance function d as

$$d(G,H) \doteq \left(\|G_s(z) - H_s(z)\|_2^2 + \|G_u(\beta/z) - H_u(\beta/z)\|_2^2)\right)^{\frac{1}{2}}.$$

The results later in this chapter that make use of this distance function are similar for any value of α and β. However, α is usually taken to be very close to 1. Finally, define the projection $\pi_s : \mathcal{G} \to H_2$

$$\pi_s(G) \doteq G_s.$$

5.2.2 Convex Functions and Subgradients

Consider a convex function $g : H_2 \rightarrow \mathbf{R}$. Then, given any $G_0 \in H_2$, there exists a $\partial_G g(G_0) \in H_2$ such that

$$g(G) - g(G_0) \geq \langle \partial g_G(G_0), G - G_0 \rangle. \tag{5.1}$$

for all $G \in H_2$. The quantity $\partial_G g(G_0)$ is said to be a subgradient of g at the point G_0. For example if $g(G) = \|G\|_2$ and $m = n = 1$; i.e.,

$$g(G) = \left(\frac{1}{2\pi} \int_0^{2\pi} |G(e^{j\theta})|^2 d\theta \right)^{1/2}$$

then the results in [10] indicate that

$$\partial_G g(G) = \frac{1}{2\pi\|G\|_2} \, G. \tag{5.2}$$

5.2.3 Closed Loop Transfer Function Parametrization

Central to the results presented here is the parametrization of all closed loop transfer functions. Consider the closed loop plant in Figure 5.1 with uncertain parameters $\triangle \in \mathbf{\Delta}$. The uncertainty $\triangle$ can include static uncertainty, uncertain transfer function matrices or a combination of both. The Youla parametrization (e.g., see [47]) indicates that, given $\triangle \in \mathbf{\Delta}$ and a stabilizing controller $C \in \mathcal{G}$, the closed loop transfer function can be represented as

$$T_{CL}(z, \triangle, C) = T^1_\triangle(z) + T^2_\triangle(z) Q_{\triangle,C}(z) T^3_\triangle(z), \tag{5.3}$$

where $T^1_\triangle, T^2_\triangle, T^3_\triangle \in RH_2$ are determined by the plant $G(z, \triangle)$ (and, hence, they also depend on the uncertainty $\triangle$) and $Q_{\triangle,C} \in RH_2$ depends on both the open loop plant $G(z, \triangle)$ and the controller $C(z)$. Also, given any $Q_{\triangle,C}(s) \in RH_2$, there exists a controller $C \in \mathcal{G}$ such that the equality above is satisfied.

This parametrization also holds for all closed loop transfer functions, stable and unstable. Using a frequency scaling reasoning one can prove the following result: Given $\triangle \in \mathbf{\Delta}$ and a controller $C \in \mathcal{G}$, the closed loop transfer function can be represented as

$$T_{CL}(z, \triangle, C) = T^1_\triangle(z) + T^2_\triangle(z) Q_{\triangle,C}(z) T^3_\triangle(z), \tag{5.4}$$

where $T^1_\triangle, T^2_\triangle, T^3_\triangle \in RH_2$ are the same as above and $Q_{\triangle,C}(s) \in \mathcal{G}$. Furthermore, given any $Q_{\triangle,C}(s) \in \mathcal{G}$ there exists a controller $C \in \mathcal{G}$ such that the equality above is satisfied.

To see this, let $G(z,\triangle) = N_r(z)D_r(z)^{-1} = D_l(z)^{-1}N_l(z)$ be right and left coprime factorization of $G(z,\Delta)$ with $N_r(z), D_r(z), D_l(z)$ and $N_l(z)$ being transfer function matrices in RH_∞. Note, that the transfer functions $T^1_\triangle, T^2_\triangle$ and $T^3_\triangle$ can be determined using these coprime factorization; e.g., see [47]. Now, consider the problem of finding all realizable rational closed loop transfer function matrices analytic in the set

$$C_\rho \doteq \{z \in \mathbf{C} : |z| \geq \rho\}$$

with $\rho > 1$. Note this is equivalent to finding all achievable stable rational closed loop transfer functions when the open loop plant is $G(z/\rho,\triangle)$. Given the fact that $N_r(z/\rho), D_r(z/\rho), D_l(z/\rho)$ and $N_l(z/\rho)$ are coprime factorizations of $G(z/\rho,\triangle)$, then all achievable stable rational closed loop transfer functions for the modified problem are given by

$$T^1_\triangle(z/\rho) + T^2_\triangle(z/\rho)Q(z)T^3_\triangle(z/\rho)$$

where $Q(z)$ is any stable transfer function matrix of appropriate dimension. Now, this means that

$$T^1_\triangle(z) + T^2_\triangle(z)Q(\rho z)T^3_\triangle(z)$$

parameterizes all achievable rational closed loop transfer function matrices which are analytic in C_ρ. Since $Q(z)$ was taken to be any stable transfer function matrix, $Q(\rho z)$ is any transfer function matrix analytic in C_ρ. Hence, the "traditional" Youla parametrization also parameterizes all achievable rational closed loop transfer functions that are analytic in C_ρ if instead of using stable transfer function matrices Q, we use transfer function matrices that are analytic in C_ρ. Since this reasoning holds for any $\rho > 1$, then all achievable rational closed loop transfer functions are of the form

$$T^1_\triangle(z) + T^2_\triangle(z)Q(z)T^3_\triangle(z)$$

with $Q \in \mathcal{G}$.

5.3 Controller Design Algorithm

Before providing the controller design algorithm, we first provide a precise definition of the problem to be solved and the assumptions that are made.

5.3.1 Problem Statement

Consider the closed-loop system in Figure 5.1 and a convex objective function $g : H_2 \rightarrow \mathbf{R}$. Given a performance level γ, we aim at designing a controller $C^*(z)$ such that the closed loop system $T_{CL}(z, \Delta, C^*)$ is stable for all admissible values of the uncertainty and satisfies

$$g\left[T_{CL}(z, \triangle, C^*)\right] \leq \gamma$$

for all $\triangle \in \Delta$. Throughout this chapter, we will assume that the problem above is feasible. More precisely, the following assumption is made:

ASSUMPTION 5.3.1. *There exists a controller C^* and an $\varepsilon > 0$ such that*

$$d(Q_{\triangle, C^*}, Q) < \varepsilon \Rightarrow g\left[T^1_{\triangle}(z) + T^2_{\triangle}(z)Q(z)T^3_{\triangle}(z)\right] \leq \gamma$$

for all $\triangle \in \Delta$.

5.3.2 Controller Design Algorithm

We now state the proposed robust controller design algorithm. This algorithm has a free parameter η that has to be specified. This parameter can be arbitrarily chosen from the interval $(0, 2)$.

Controller Design Algorithm

Step 0. Let $k = 0$. Pick a controller $C_0(z)$.

Step 1. Draw sample $\triangle^k$. Given $G(z, \triangle^K)$, compute $T^1_{\triangle^k}(z)$, $T^2_{\triangle^k}(z)$, $T^3_{\triangle^k}(z)$ as described in [47].

Step 2. Let $Q_k(z)$ be such that the closed loop transfer function using controller $C_k(z)$ is

$$T_{CL}(z, \triangle^k, C_k) = T^1_{\triangle^k}(z) + T^2_{\triangle^k}(z)Q_k(z)T^3_{\triangle^k}(z)$$

Step 3. Do the stabilizing projection[1]

$$Q_{k,s}(z) \;=\; \pi_s(Q_k(z)).$$

Step 4. Perform update

$$Q_{k\to k+1}(z) = Q_{k,s}(z) - \alpha_k(Q_{k,z},\triangle^k)(z)\partial_Q g(T_{CL}(z,\triangle^k,Q))|_{Q_{k,s}} \tag{5.5}$$

where

$$\alpha_k(Q_k,\triangle) = \begin{cases} \eta\,\frac{g(T_{CL}(z,\triangle,Q_k))-\gamma+\varepsilon\,\|\partial_Q g(T_{CL}(z,\triangle,Q))|_{Q_k}\|_2}{\|\partial_Q g(T_{CL}(z,\triangle,Q))|_{Q_k}\|_2^2} & \text{if } \; g(T_{CL}(z,\triangle,Q_k)) > \gamma \\ 0 & \text{otherwise,} \end{cases} . \tag{5.6}$$

Step 5. Determine the controller $C_{k+1}(z)$ so that

$$Q_{\triangle^k,C_{k+1}} = Q_{k\to k+1}.$$

Step 6. Let $k = k+1$. Go to Step 1.

5.3.3 Remark

In the algorithm above we assume knowledge of the quantity ε. If the value of ε is not available, one can instead use a decreasing sequence $\varepsilon_k > 0$ whose limit is zero and

$$\sum_{k=1}^{\infty} \varepsilon_k^2 = \infty.$$

The results presented in this chapter can be easily altered to allow for this modification. However, if the value of ε is available, one should use it since the introduction of the sequence ε_k reduces the speed of convergence.

[1] Note that, since C_k is not guaranteed to be a robustly stabilizing controller, Q_k might not be stable.

5.3.4 Stopping Criterion

In a practical implementation of the algorithm above, a possible stopping criterion is the following: Periodically perform a Monte Carlo simulation to estimate the risk of performance violation and stop if the risk is below a given threshold.

5.4 Main Result

We now present the main result of this chapter; i.e., the algorithm described in the previous section converges to a controller that robustly satisfies the performance specifications. The exact statement is given below.

THEOREM 5.4.1. *Let $g : H_2 \to \mathbf{R}$ be a convex function with subgradient $\partial g \in RH_2$ and let $\gamma > 0$ be given. Define the risk of performance violation as*

$$P_k \doteq Prob\{g(T_{CL}(z, \triangle, C_k)) > \gamma\}.$$

Then, if Assumption 5.3.1 holds, the algorithm described in section 5.3.2 generates a sequence of controllers C_k for which the risk of performance violation satisfies

$$\sum_{k=1}^{\infty} P_k < \infty$$

and

$$\lim_{k\to\infty} P_k = 0.$$

Hence, risk tends to zero as $k \to \infty$.

5.4.1 Remark

For simplicity, in the sequel, we only consider the case where the subgradient of the objective function is rational. However, the result above can be modified to the case of non-rational $\partial g \in H_2$ that can be arbitrarily approximated by a rational function; i.e., the result above can be extended to the case where the subgradient belongs to the closure of RH_2.

Before the proof of Theorem 5.4.1 is presented, we introduce the concept of robust controller gap.

5.4.2 Robust Controller Gap

In order to prove the result above, one needs a measure of how far a controller is from the optimal. Hence, the concept of *robust controller gap* is introduced. This "distance" measure uses the difference between closed loop transfer functions as an indication of how far are the controllers.

Let f be the probability density function used to generate the samples in the controller design algorithm. Then, given two controllers C_1 and C_2, the robust gap is

$$r_{gap}(C_1,C_2) = \int_{\Delta} d^2(Q_{\triangle,C_1}, Q_{\triangle,C_2}) f(\triangle) d\triangle.$$

Hence, given three controllers C_1, C_2 and C^*, we have

$$d^2(Q_{\triangle,C_1}, Q_{\triangle,C^*}) - d^2(Q_{\triangle,C_2}, Q_{\triangle,C^*}) = r_{gap}(C_1,C_*) - r_{gap}(C_2,C_*) + V$$

with

$$\mathbf{E}[V|C_1,C_2,C^*] = 0.$$

where $\mathbf{E}[X|Y]$ denotes the conditional expectation of X given Y.

We are now ready to present the proof.

5.4.3 Proof of Theorem 5.4.1

The first part of the proof is similar to the one in [15]. Let C^* be a controller which achieves the robust performance specification. Given a sample $\triangle \in \Delta$, let $Q^*_{\triangle} \in H_2$ be the corresponding *Youla* parameter. Define

$$\overline{Q}_{\triangle^k} \doteq Q^*_{\triangle^k} + \frac{\varepsilon}{\|\partial_Q g(T_{\triangle^k,Q}(z))|_{Q_{k,s}}\|_2} \partial_Q g(T_{\triangle^k,Q}(z))|_{Q_{k,s}}. \tag{5.7}$$

By Assumption 5.3.1, $\overline{Q}_{\triangle^k}$ satisfies the performance specifications; i.e.,

$$g(T_{\Delta,\overline{Q}_{\triangle^k}}(z)) \leq \gamma, \; \forall \Delta \in \Delta$$

Equation (5.5) indicates that

$$d(Q_{k\to k+1}, Q^*_{\triangle^k})^2 = d(Q_{k,z} - \alpha_k(Q_{k,z},\triangle^k)\partial_Q g(T_{CL}(\triangle^k,Q)|_{Q_{k,z}}, Q^*_{\triangle^k})^2.$$

Now, given the fact that g is a convex functional on H_2, $\partial g \in H_2$ exists and (5.1) is satisfied. Thus,

$$Q_{k\to k+1}(z) = Q_{k,s}(z) - \alpha_k(Q_{k,s}, \triangle^k)\partial_Q g(T_{CL}(z,\triangle^k,Q))|_{Q_{k,s}}$$

belongs to H_2. For simplicity, denote $T_{CL}(z,\triangle^k,C)$ as $T_{\triangle^k,C}$. Hence, we have

$$\begin{aligned} d(Q_{k\to k+1}, Q^*_{\triangle^k})^2 &= \|Q_{k,s} - \alpha_k(Q_{k,s},\triangle^k)\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}} - Q^*_{\triangle^k}\|_2^2 \\ &= \|Q_{k,s} - Q^*_{\triangle^k}\|_2^2 + \alpha_k^2\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2 - 2\alpha_k\langle Q_{k,s} - \overline{Q}_{\triangle^k}, \partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\rangle \\ &\quad -2\alpha_k\langle \overline{Q}_{\triangle^k} - Q^*_{\triangle^k}, \partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\rangle \end{aligned}$$

Since $g(T_{\Delta,Q}(z)) = \|T_{\Delta,Q}(z)\|_2$ is convex in Q, the following inequality holds

$$\langle Q_{k,s} - \overline{Q}_{\triangle^k}, \partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\rangle \geq g(T_{\triangle^k,Q_{k,s}}) - g(T_{\triangle^k,\overline{Q}_{\triangle^k}}) \geq g(T_{\triangle^k,Q_{k,s}}) - \gamma.$$

On the other hand, equation (5.7) can be used to obtain

$$\langle \overline{Q}_{\triangle^k} - Q^*_{\triangle^k}, \partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\rangle = \varepsilon\|\partial_Q g(T_{\triangle^k,Q}(s))|_{Q_{k,s}}\|_2$$

Hence,

$$\begin{aligned} d(Q_{k\to k+1}, Q^*_{\triangle^k})^2 &\leq \|Q_{k,s} - Q^*_{\triangle^k}\|_2^2 + \alpha_k^2\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2 \\ &\quad -2\alpha_k(g(T_{\triangle^k,Q_{k,s}}) - \gamma + \varepsilon\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2). \end{aligned}$$

If $g(T_{\triangle^k,Q_k}) > \gamma$, we get

$$\begin{aligned} d(Q_{k\to k+1}, Q^*_{\triangle^k})^2 &\leq -\frac{\eta(2-\eta)}{\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2}(g(T_{\Delta,Q_{k,s}}) - \gamma + \varepsilon\,\|\partial_Q g(T_{\Delta,Q}(s))|_{Q_{k,s}}\|_2)^2 \\ &\quad +\|Q_{k,s} - Q^*_{\triangle^k}\|_2^2 \\ &\leq \|Q_{k,s} - Q^*_{\triangle^k}\|_2^2 - \varepsilon^2\eta(2-\eta) \end{aligned}$$

Now, define the indicator function

$$\mathbf{I}_{\{g(T_{\triangle^k,Q_{k,s}})>\gamma\}} \doteq \begin{cases} 1 & \text{if } g(T_{\triangle^k,Q_{k,s}}) > \gamma \\ 0 & \text{otherwise,} \end{cases}$$

and obtain the following inequality

$$d(Q_{k\to k+1}, Q^*_{\triangle^k})^2 \;\le\; \|Q_{k,s} - Q^*_{\triangle^k}\|_2^2 - \varepsilon^2\eta(2-\eta)\mathbf{I}_{\{g(T_{\triangle^k, Q_{k,s}})>\gamma\}}.$$

Since

$$\|Q_{k,s} - Q^*_{\triangle^k}\|_2^2 \;\le d(Q_k, Q^*_{\triangle^k})^2,$$

we have

$$d(Q_{k\to k+1}, Q^*_{\triangle^k})^2 \;\le\; d(Q_k, Q^*_{\triangle^k})^2 - \varepsilon^2\eta(2-\eta)\mathbf{I}_{\{g(T_{\triangle^k, Q_{k,s}})>\gamma\}}.$$

Given the definition of robust gap, provided in the previous section, the equation above can be rewritten in the following form

$$r_{gap}(C_{k+1}, C^*) \le\; r_{gap}(C_k, C^*) - \varepsilon^2\eta(2-\eta)\mathbf{I}_{\{g(T_{\triangle^k, Q_{k,s}})>\gamma\}} + V_k$$

where

$$\mathbf{E}[V_k | C_{k+1}, C_k, C^*] = 0.$$

Now let $\mathcal{F}_k = \sigma(r_{gap}(C_1, C^*), \ldots, r_{gap}(C_k, C^*))$ be the σ-algebra generated by $r_{gap}(C_1, C^*)$, $r_{gap}(C_2, C^*)$, $\ldots$, $r_{gap}(C_k, C^*)$. Now, take the expectation conditioned on $\mathcal{F}_k$. By the *tower property* of conditional expectation; i.e., if Z is a function of Y, then $\mathbf{E}[\mathbf{E}(X|Y)|Z] = \mathbf{E}(X|Z)$, we get

$$\begin{aligned}
\mathbf{E}[V_k|\mathcal{F}_k] &= \mathbf{E}[\mathbf{E}(V_k|C_1, \ldots, C_k, C^*)|\mathcal{F}_k] \\
&= \mathbf{E}[\mathbf{E}(\mathbf{E}(V_k|C_1, \ldots, C_k, C_{k+1}, C^*)|C_1, \ldots, C_k, C^*)|\mathcal{F}_k] \\
&= 0.
\end{aligned}$$

Then,

$$\mathbf{E}[r_{gap}(C_{k+1}, C^*)|\mathcal{F}_k] \le r_{gap}(C_k, C^*) - \varepsilon^2\eta(2-\eta)P_k, \tag{5.8}$$

where $0 < \eta < 2$. Now, note that $r_{gap}(C_k, C^*) \ge 0$. Furthermore, one can easily prove that $\mathbf{E}[r_{gap}(C_0, C^*)] < \infty$. Hence, the process $\{r_{gap}(C_k, C^*), k \ge 1\}$ is a *supermartingale* and it is bounded in $\mathcal{L}^1$. Therefore, $r_{gap}(C_k, C^*)$ converges to a finite value with probability one; e.g., see [68]. Hence, its expectation also converges to a finite value. Now, compute the expected value of both sides of (5.8) and get

$$\mathbf{E}[r_{gap}(C_{k+1}, C^*)] \le \mathbf{E}[r_{gap}(C_k, C^*)] - \varepsilon^2\eta(2-\eta)P_k$$

Hence,

$$\mathbf{E}[r_{gap}(C_{k+1},C^*)] \leq r_{gap}(C_0,C^*) - \varepsilon^2\eta(2-\eta)\sum_{i=0}^{k} P_i$$

Given the fact that $\mathbf{E}[r_{gap}(C_{k+1},C^*)]$ converges to a finite value, we have to have

$$\sum_{k=0}^{\infty} P_k \;<\; \infty.$$

Thus

$$\lim_{k\to\infty} P_k \;=\; 0$$

Q.E.D.

If we define A_k as the event when $g(T_{CL}(z,\Delta,C_k)) > \gamma$ happens, we have the following:

COROLLARY 5.4.1. *Suppose all conditions hold as in Theorem 5.4.1, only finitely many of the events A_k can occur (w.p.1).*

Proof: By Theorem 5.4.1, we have

$$\sum_{k=1}^{\infty} \text{Prob}\{A_k\} \;<\; \infty \; w.p.1$$

By the Borel-Cantelli Lemma [43], we have

$$\text{Prob}\{A_k i.o.\} \;=\; 0.$$

Thus A_k can only happen a finite number of times w.p.1. Q.E.D.

5.4.4 Remark

By Corollary 5.4.1, for any given uncertainty sample $\triangle \in \Delta$, we are almost sure that it will only satisfy a finite number of the $A_k s$. This actually provides another perspective on robust control problems. Let admissible uncertainty set Δ and controller solution set $\mathcal{C}^*$ be two sides of a game and the nominal plant be the "battlefield". The design goal is to find $C^* \in \mathcal{C}^*$ such that C^* can "defeat" any $\triangle \in \Delta$ for traditional robust design or most of the $\triangle$s for probabilistic robust design. From the point of view of uncertainty, if the designer can find a controller solution set such that any $\triangle \in \Delta$ will be "defeated" by all or almost all elements in the solution set, the design objective can also be deemed to be achieved.

5.5 Multi-Disk Design

In this section, we extend the work presented in previous sections to the problem of multi-disk design. The goal is to design a robustly stabilizing controller that results in guaranteed performance in a subset of the uncertainty support set. Before presenting the controller design algorithm, we first provide a precise formulation of the problem and the assumptions that are made.

5.5.1 Problem Statement

Consider the closed-loop system in Figure 5.1 and a convex objective function $g_1\colon H_2 \rightarrow \mathbf{R}$. Given a performance value γ_1 and uncertainty radii $r_2 > r_1 > 0$, we aim at designing a controller $C^*(z)$ such that the closed loop system $T_{CL}(z, \triangle, C^*)$ is stable for all $\|\triangle\|_\infty \leq r_2$ and satisfies

$$g_1[T_{CL}(z, \triangle, C^*)] \ \leq \ \gamma_1$$

for all $\|\triangle\|_\infty \leq r_1$. Similarly to the results presented in the previous sections, we assume that the following holds:

ASSUMPTION 5.5.1. *There exists a controller C^* and an $\varepsilon > 0$ such that*

$$d(Q_{\triangle,C^*}, Q) < \varepsilon \Rightarrow g_1\left[T^1_\triangle(z) + T^2_\triangle(z)Q(z)T^3_\triangle(z)\right] \leq \gamma_1$$

for all $\|\triangle\|_\infty \leq \ r_1$ and there exists a γ_2 (sufficiently large) such that

$$d(Q_{\triangle,C^*}, Q) < \varepsilon \Rightarrow g_2\left[T^1_\triangle(z) + T^2_\triangle(z)Q(z)T^3_\triangle(z)\right] \doteq \ \|T^1_\triangle(z) + T^2_\triangle(z)Q(z)T^3_\triangle(z)\| \leq \gamma_2$$

for all $\|\triangle\|_\infty \leq \ r_2$.

5.5.2 Remark

Even though the above is a slightly stronger requirement than robust stability, the existence of a large constant γ_2 satisfying the second condition above can be considered to be equivalent to robust stability from a practical point of view.

5.5.3 Multi-Disk Controller Design Algorithm

We now state the proposed robust controller design algorithm. This algorithm also has a free parameter η that has to be specified and can be arbitrarily chosen from the interval (0,2).

Multi-Disk Controller Design Algorithm

Step 0. Let $k = 0$. Pick a controller $C_0(z)$.

Step 1. Generate sample i^k with equal probability of being 1 or 2.

Step 2. Draw sample $\triangle^k$ over $\Delta(r_{i^k})$. Given $G(z,\triangle^k)$, compute $T^1_{\triangle^k}(z)$, $T^2_{\triangle^k}(z)$, $T^3_{\triangle^k}(z)$ as described in [47].

Step 3. Let $Q_k(z)$ be such that the closed loop transfer function using controller $C_k(s)$ is

$$T_{CL}(z,\triangle^k,\ C_k) = T^1_{\triangle^k}(z) + T^2_{\triangle^k}(z)Q_k(z)T^3_{\triangle^k}(z)$$

Step 4. Do the stabilizing projection[2]

$$Q_{k,s}(z) \ = \ \pi_s(Q_k(z)).$$

Step 5. Perform update

$$Q_{k\to k+1}(z) = Q_{k,s}(z) - \alpha_k(Q_{k,z},\triangle^k)(z)\partial_Q g_{i^k}(T_{CL}(z,\triangle^k,Q))|_{Q_{k,s}} \tag{5.9}$$

where

$$\alpha_k(Q_k,\triangle) = \begin{cases} \eta\,\frac{g_{i^k}(T_{CL}(z,\triangle,Q_k)) - \gamma_{i^k} + \varepsilon\,\|\partial_Q g_{i^k}(T_{CL}(z,\triangle,Q))|_{Q_k}\|_2}{\|\partial_Q g(T_{CL}(z,\triangle,Q))|_{Q_k}\|_2^2} & \text{if } \ g_{i^k}(T_{CL}(z,\triangle,Q_k)) > \gamma_{i^k} \\ 0 & \text{otherwise,} \end{cases} . \tag{5.10}$$

Step 6. Determine the controller $C_{k+1}(z)$ so that

$$Q_{\triangle^k,C_{k+1}} = Q_{k\to k+1}.$$

[2]Note that, since C_k is not guaranteed to be a robustly stabilizing controller, Q_k might not be stable.

Step 7. Let $k = k + 1$. Go to Step 1.

It can be proven that the algorithm described above indeed converges to a controller that robustly satisfies the performance specifications. The exact statement is given below. The proof follows the same line of reasoning of the proof of Theorem 5.4.1.

THEOREM 5.5.2. *Let $g_1 : H_2 \rightarrow \mathbf{R}$ be a convex function with subgradient $\partial g_1 \in RH_2$ and let $\gamma_1 > 0$ be given. Also let $g_2(H) = \|H\|_2$. Define*

$$P_{k,1} \doteq Prob\{g_1(T_{CL}(z, \triangle, C_k)) > \gamma_1\}$$

with $\triangle$ having the distribution over $\Delta(r_1)$ used in the algorithm. Similarly take

$$P_{k,2} \doteq Prob\{g_2(T_{CL}(z, \triangle, C_k)) > \gamma_2\}$$

with $\triangle$ having the distribution over $\Delta(r_2)$ used in the algorithm. Given this, define

$$P_k \doteq \frac{1}{2}P_{k,1} + \frac{1}{2}P_{k,2}.$$

Then, if Assumption 5.5.1 holds, the algorithm described above generates a sequence of controllers C_k for which the risk of performance violation satisfies

$$\lim_{k \to \infty} P_k = 0.$$

Hence, risk tends to zero as $k \rightarrow \infty$.

5.6 Example: Robust H_2 Design

We now turn our attention to the case of robust weighted H_2 controller design. We start by indicating how a subgradient of the objective functional is computed. This section is then followed by numerical examples.

5.6.1 Computing the Subgradient of Weighted H_2 Norm

For simplicity of exposition, we are going to consider the single input/single output case. A straightforward extension can be done to the case of multiple inputs and/or outputs. Given

$W \in RH_2$, consider the weighted H_2 norm defined as

$$g(G) = \left(\frac{1}{2\pi} \int_0^{2\pi} |W(e^{j\theta})G(e^{j\theta})|^2 d\theta \right)^{1/2}.$$

Now, since we are considering the case of a single input/ single output system, given a controller C and an uncertainty value $\triangle \in \Delta$, the closed loop transfer function can be represented in the form

$$T_{CL}(z,\triangle,C) = T^1_{\triangle}(z) + T^2_{\triangle}(z)Q_{\triangle,C}(z).$$

Now, the results in [10] indicate that, in this case, the subgradient with respect to Q of the objective function is given by

$$\partial_Q g(T_{CL}(z,\triangle^k,C))(Q) = \frac{1}{2\pi \|T_{CL}(z,\triangle,Q)\|_2} \, T_{CL}(z,\triangle,C) \, T^2_{\triangle}(z)W(z).$$

5.6.2 Numerical Example: Single Objective

A simplified model of a DC armature-controlled servomotor is

$$P(s,\triangle) = \frac{\omega^2}{s(s+2\delta\omega)}$$

where both ω and δ can be estimated through experiments. We assume that these parameters are uncertain; i.e.,

$$\omega = \omega_0 + \triangle\omega, \;\; \delta = \delta_0 + \triangle\delta.$$

In simulations, we take $\omega_0 = 6, \; \delta_0 = 0.3$ and $\triangle\omega, \triangle\delta$ are uniformly distributed on $[-r,r]$ and $[-\varepsilon r, \varepsilon r]$ respectively. In our example, we take $r = 1$, $\varepsilon = 0.1$ and set $\gamma = 0.15$. To facilitate the control of the motor by a digital microprocessor, we need to consider the H_2 control problem in discrete time. Given $\gamma > 0$ and weighting function,

$$W(z) = \frac{0.03333z + 0.04536}{z - 0.6065}.$$

we aim at finding $C(z)$ such that

$$\|W(z)(1 + C(z)P(z,\triangle))^{-1}\|_2 \leq \; \gamma$$

for all admissible $\triangle$. The sample period is $0.5s$. The configuration of the system is shown in Figure 5.2:

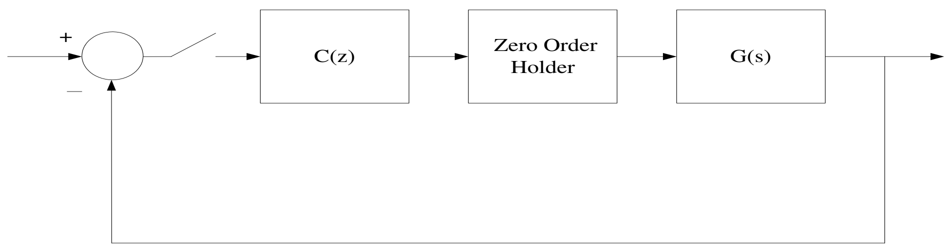


Fig. 5.2. DC Servomotor Control System

As the first step, an optimal nominal controller is designed using the *Matlab* function *dh2lqg*. We get

$$C_{nom}(z) = \frac{0.3034z^2 - 0.3536z + 0.05016}{z^3 + 1.291z^2 - 0.4047z - 0.4527}$$

leading to a nominal optimal performance of $\|T_C(z)\|_2 = 0.0759$.

Next, we apply Algorithm in Section 5.3.2. After 30000 iterations, the following controller was obtained

$$C_1(z) = \frac{0.3043z^3 - 0.4643z^2 + 0.2114z - 0.0102}{z^4 + 0.6533z^3 - 0.9549z^2 + 0.002239z + 0.413}$$

To assess the performance of this controller, a Monte Carlo simulation with $100,000$ samples was performed and the estimated risk of performance violation was found to be 0. When $C_1(z)$ is replaced by $C_{nom}(z)$, the risk rises to around 10.6%. We also estimated the risk of performance violation as a function of the iteration number. These estimates were obtained through Monte Carlo simulation with $3,000$ samples for each controller and the results obtained are shown in Fig. 5.3. As it can be seen, the risk decreases rapidly to very low level. At the 7000th iteration, the risk is approximately zero.

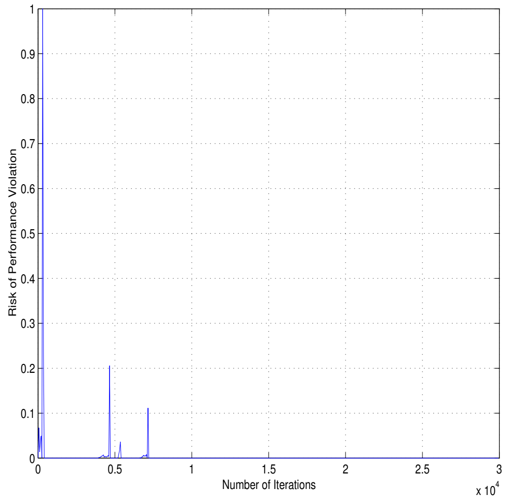


Fig. 5.3. DC Motor: Risk History

5.6.3 Numerical Example: Multidisk design

Consider the uncertain system

$$P(z,\triangle) \;=\; P_0(z,\triangle)+\triangle(z),$$

with nominal plant

$$P_0(z,\triangle) \;=\; \frac{0.006135z^2+0.01227z+0.006135}{z^2-1.497z+0.5706}$$

and stable causal dynamic uncertainty $\triangle$. The objective is to find a controller $C(z)$ such that, for all $\|\triangle\|_\infty \leq r_1 = 1$,

$$\|W(z)(1+C(z)P(z,\triangle))^{-1}\|_2 \leq \;\gamma_1 \;=\; 0.089$$

where

$$W(z) = \frac{0.0582z^2+0.06349z+0.005291}{z^2+0.2381z-0.6032}$$

and the closed loop system is stable for $\|\triangle\|_\infty \leq r_2 = 2$. The random samples of causal, linear time-invariant uncertain transfer functions were generated according to [37]. Considering the nominal plant only, we determined the H_2 optimal controller using the *Matlab* function *dh2lqg* whose transfer function is

$$C_{nom}(z) = \frac{138.2z^3 - 93.78z^2 - 90.4z + 64.5}{z^4 + 2.238z^3 + 0.8729z^2 - 0.9682z - 0.6031}$$

and $\|T_{cl}(z)\|_2 = 0.0583$. However, it does not robustly stabilizes the closed loop system for all $\|\Delta\| \leq r_2$. We next set $\gamma_1 = 0.089$ and $\gamma_2 = 10^9$ and apply Algorithm 5.5.3. After 1500 iterations, the following controller was obtained

$$\begin{aligned} C_1(z) \quad = \quad & \frac{-0.003808z^{14} - 0.01977z^{13}}{z^{14} - 0.1778z^{13} + 0.6376z^{12} + 0.09269z^{11} + 0.2469z^{10}} \\ & \frac{-0.002939z^{12}}{+0.06291z^9 + 0.08426z^8 + 0.0433z^7 + 0.07403z^6 + 0.0004446z^5} \\ & \frac{+0.04627z^{11}}{-0.1107z^4 - 0.07454z^3 - 0.08156z^2 - 0.05994z + 0.01213} \end{aligned}$$

As in Section 5.5, define

$$P_{k,1} \doteq \text{Prob}\{\|W(z) - W(z)P(z,\triangle)Q_k(z)\|_2 > \ \gamma_1\}$$

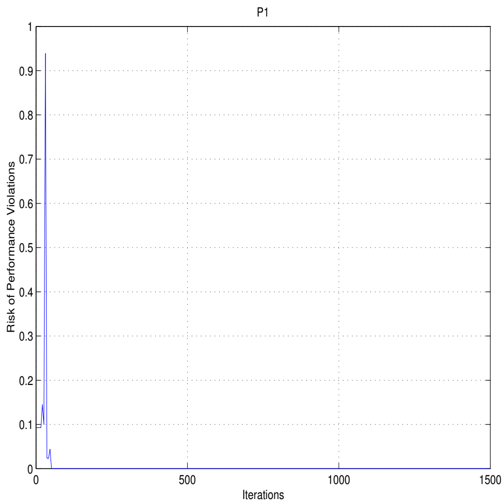
with $\|\triangle P(z)\|_\infty \leq r_1$ and

$$P_{k,2} \doteq \text{Prob}\{\|W(z) - W(z)P(z,\triangle)Q_k(z)\|_2 > \ \gamma_2\}$$

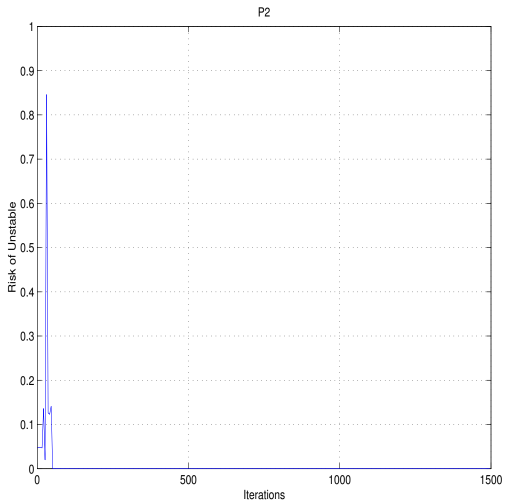
with $\|\triangle P(z)\|_\infty \leq r_2$.

Next, we investigate the asymptotic characteristics of $P_{k,1}$ and $P_{k,2}$ as the optimization process goes on. We run Monte Carlo simulations to estimate $P_{k,1}$ and $P_{k,2}$ for each controller $C_k(z)$ at each iteration of the algorithm. The results obtained are shown in Figure 5.4(a) Figure 5.4(b) and Figure 5.5(a) Figure 5.5(b) show the asymptotic behavior of maximal magnitude of closed loop poles.

It can be easily seen from the figures that $P_{k,1}$ and $P_{k,2}$ tend to zero as expected. Figure 5.5(b) also shows that the estimated risk of instability is approximately zero after just 200 iterations of the algorithm..



(a) Risk of Performance Violation

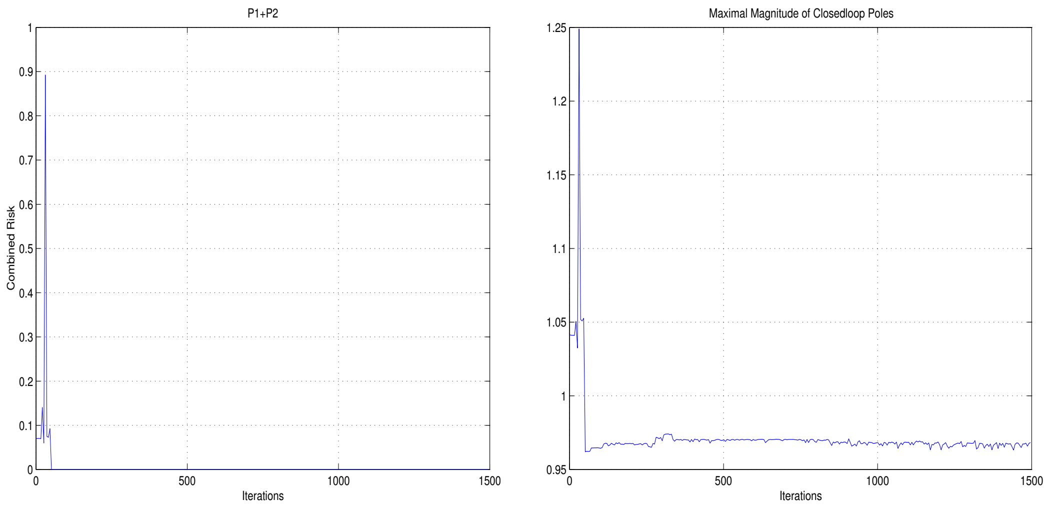


(b) Risk of Instability

Fig. 5.4. MC Simulations 1: Multi-disk Design

5.7 Concluding Remarks

In this chapter, we addressed the problem of robust controller design for linear time invariant systems with arbitrary uncertainty structure. Given bounds on a convex performance function, the proposed algorithm converges to an output feedback controller that robustly satisfies the specifications. Moreover, it is proven that this stochastic gradient like procedure produces a sequence of controllers with a risk of performance violation that decreases to zero asymptotically faster than $1/k$, where k is the number of iterations. Moreover, a multidisk design approach was proposed to obtain a robustly stabilizing controller with guaranteed performance on a subset of the uncertainty support set. As an example, the problem of robust H_2 performance was considered and numerical examples were provided.



(a) Combined Risk (b) Maximal Magnitude of Poles

Fig. 5.5. MC Simulations 2: Multi-disk Design

Chapter 6

Probabilistic Robust Optimal Output Feedback

In this chapter, we further study the problem of robust output feedback design for linear time invariant systems with arbitrary uncertainty structures, which was first addressed in the previous chapter. The algorithms in Chapter 5 are suboptimal in the following sense: For a given threshold value γ, the algorithms try to push the performance level below γ. They do not further optimize the performance. If the given performance level is robustly achievable, the algorithms in the previous chapter provide a sequence of controllers whose probability of performance violation converges to zero under the assumption that there exists a "ball" around the optimal solution within which the performance requirement is met. However this assumption might not be easy to verify in practice. In this chapter, we propose two algorithms to overcome these limitations of the preliminary results. More precisely, we provide two algorithms which do not require *a priori* knowledge of an achievable performance level.

6.1 Preliminaries

Similar to Section 5.2, define the space $\mathcal{G}$ as the space of rational functions $G : \mathbf{C} \to \mathbf{C}^{n\times m}$ that can be represented as

$$G(z) = G_s(z) + G_{us}(z). \tag{6.1}$$

where G_s is analytic in the set $\{z \in \mathbf{C} : |z| \geq \alpha\}$ and G_{us} is strictly proper and analytic in the set $\{z \in \mathbf{C} : |z| < \alpha\}$ and $0 < \alpha < 1$. Given two transfer functions $G,\ H\ \in\ \mathcal{G}$ and $0 < \beta < \alpha$, we have the "distance" between G and H as

$$d^2(G,H) \doteq \|G_s(z) - H_s(z)\|_2^2 + \|G_{us}(\beta/z) - H_{us}(\beta/z)\|_2^2. \tag{6.2}$$

Defining

$$\|G_{us}(z)\| \doteq \|G_{us}(\beta/z)\|_2, \tag{6.3}$$

one has

$$\|G(z)\|^2 \doteq \|G_s(z)\|_2^2 + \|G_{us}(z)\|^2. \tag{6.4}$$

Given $\theta > 0$ and $\vartheta > 0$, define the projection $\pi_s : \mathcal{G} \rightarrow H_2$:

$$\pi_s(G) \doteq G_s,$$

and the projection $\psi_\vartheta : \mathcal{G} \rightarrow \mathcal{G}$:

$$\psi_\vartheta(G) \doteq G_\vartheta = G_{\vartheta,s} + G_{\vartheta,us}$$

where

$$G_{\vartheta,s} \doteq \begin{cases} G_s & \|G_s\|_2 \leq \vartheta, \\ G_s \frac{\vartheta}{\|G_s\|_2} & \text{otherwise} \end{cases}$$

and

$$G_{\vartheta,us} \doteq \begin{cases} G_{us} & \|G_{us}\| \leq \vartheta, \\ G_{us} \frac{\vartheta}{\|G_{us}\|} & \text{otherwise} \end{cases}$$

The definition of $\|G_{us}\|$ is given in (6.3). Furthermore, define

$$\pi_{s,\theta}(G) \doteq G_{s,\theta} \doteq \begin{cases} G_s & \|G_s\|_2 \leq \theta, \\ G_s \frac{\theta}{\|G_s\|_2} & \text{otherwise} \end{cases}$$

6.2 Controller Design Algorithm I – Projection Algorithm

In this section, we provide an algorithm for controller design for the case where it is known *a priori* that the system is robustly stabilizable.

6.2.1 Problem Statement

Given $\theta > 0$, let $A_\theta \doteq \{C \in \mathcal{G} : \ \|Q_{C,s}\|_2 \leq \theta, \ \forall \triangle \in \Delta\}$. Consider the discrete-time closed loop linear time invariant system depicted in Figure 5.1 where $\triangle \in \Delta$ represents the uncertainty. As before, given an uncertainty value $\triangle \in \Delta$ and a controller $C(z)$, let $T_{CL}(z, \triangle, C)$ be the corresponding closed loop transfer function matrix. Given a convex function $g : \ H_2 \ \rightarrow \ R$ and a probability distribution for the uncertainty $\triangle$, we aim at finding a robustly stabilizing controller $C^*(z) \in A_\theta$ that minimizes the expect performance of the closed loop system; i.e., we aim at finding

$$C^* \doteq \mathrm{argmin}_{C \in A_\theta} \mathbf{E}_\triangle \{ g\, [T_{CL}(z, \triangle, Q_{C,s})] \}.$$

To simplify the notation, we will omit z as the transfer function's argument wherever it will not cause confusion. For the algorithm in this section, the following assumption is made:

ASSUMPTION 6.2.1. *There exist $\tau > \theta > 0$, $\delta > 0$ and $\zeta > 0$ such that:*

1. *There exists a robustly stabilizing optimal controller $C^* \in A_\theta$ such that for any robustly stabilizing controller C, we have*

$$\mathbf{E}_\Delta \{ g\, [T_{CL}(\triangle, Q_{C^*})] \} \leq \mathbf{E}_\Delta \{ g\, [T_{CL}(\triangle, Q_{C,s})] \}.$$

Furthermore, for any controller C having $0 < \mathbf{E}_\Delta \{ \|Q_{C,us}\| \} \leq \delta$, one has

$$\mathbf{E}_\Delta \{ g\, [T_{CL}(\triangle, Q_{C^*})] \} + \zeta \leq \mathbf{E}_\Delta \{ g\, [T_{CL}(\triangle, Q_{C,s})] \} \tag{6.5}$$

2. *For any controller C, if there is a $\triangle_0 \in \Delta$ such that $\|Q_{C,\triangle_0,s}\|_2 \leq \theta$, we have*

$$\|Q_{C,\triangle,s}\|_2 \leq \tau \text{ for all } \triangle \in \Delta.$$

3. *There exists a constant $N > 0$ such that if $\|Q\|_2 \leq \tau$,*

$$\|\partial_Q g(T_{CL}(\triangle, Q))\|_2 < N \quad \text{for all } \triangle \in \Delta.$$

Remark: Note that assumption 1 above is a rather mild one since for many common performance measures, the value of g will increase as one gets "closer" to instability. Assumption 3 has its root in the finite-dimensional space case where it has been proven that subgradients of a convex function are bounded on any bounded set [51].

6.2.2 Design Algorithm I

We now present an algorithm that converges to a solution of the problem described above.

ALGORITHM 6.2.1.

Step 0. Let $k = 0$*. Pick a controller* $C_0(z)$ *and* $\theta > 0$.

Step 1. Draw sample $\triangle^k$*. Given* $G(z, \triangle^K)$*, compute* $T^1_{\triangle^k}(z)$, $T^2_{\triangle^k}(z)$, $T^3_{\triangle^k}(z)$ *as described in [47].*

Step 2. Let $Q_k(z)$ *be such that the closed loop transfer function using controller* $C_k(z)$ *is*

$$T_{CL}(\triangle^k,\, C_k) = T^1_{\triangle^k} + T^2_{\triangle^k} Q_k T^3_{\triangle^k}$$

Step 3. Do the stabilizing projection

$$Q_{k,s}(z) \ = \ \pi_s(Q_k(z)).$$

Step 4. Perform update

$$Q_{k+1} = \psi_\theta(\ Q_{k,s} - \varepsilon_k \partial_Q g(T_{CL}(\triangle^k, Q))|_{Q_{k,s}}). \tag{6.6}$$

Step 5. Determine the controller $C_{k+1}(z)$ *so that*

$$Q_{\triangle^k, C_{k+1}} = Q_{k+1}.$$

Step 6. Let $k = k+1$*. Go to Step 1.*

6.2.3 Convergence Theorem

We now prove convergence of the algorithm above; i.e., we prove that the algorithm described in the previous section converges to a robustly stabilizing controller that minimizes the expected performance specifications. The exact statement is given below.

THEOREM 6.2.1. *Let* $g : H_2 \to \mathbf{R}^+_0$ *be a convex function with subgradient* $\partial g \in RH_2$ *and let* C^* *be the robustly stabilizing optimal controller. Then, if Assumption 6.2.1 holds and*

$$\varepsilon_k > 0,\ \sum_{k=1}^{\infty} \varepsilon_k = \ \infty,\ \sum_{k=1}^{\infty} \varepsilon_k^2 = \ \beta < \ \infty$$

algorithm 6.2.1 generates a sequence of controllers C_k for which with probability 1

$$\mathbf{E}_\Delta\{g\,[T_{CL}(\triangle, Q_{C_k,s})]\} \rightarrow \ \mathbf{E}_\Delta\{g\,[T_{CL}(\triangle, Q_{C^*})]\}$$

and there exists a step k' such that $\forall k \geq k'$,

$$\mathbf{E}_\Delta\{\|Q_{C_k,us}\|\} \ = \ 0.$$

Proof: We start by introducing an auxiliary objective function $F_{aux}(C_k)$ as

$$F_{aux}(C_k) \ \doteq \ \mathbf{E}_\Delta\{g\,[T_{CL}(\triangle, Q_{C_k,s})]\} + M_k \mathbf{E}_\Delta\{\|Q_{C_k,us}\|^2\}$$

where

$$M_k \doteq \begin{cases} \frac{\mathbf{E}_\Delta\{g[T_{CL}(Q_{C^*})]\}+\eta}{\mathbf{E}_\Delta\{\|Q_{C_k,us}\|^2\}} & \mathbf{E}\{\|Q_{C_k,us}\|\} > \delta \\ 0 & \text{otherwise} \end{cases}$$

and η is a given positive constant. By *Jensen's Inequality* [7], for the first case of M_k,

$$\mathbf{E}_\Delta\{\|Q_{C_k,us}\|^2\} \ \geq \mathbf{E}^2\{\|Q_{C_k,us}\|\} \ > \delta^2.$$

By definition,

$$F_{aux}(C^*) \ = \ \mathbf{E}_\Delta\{g\,[T_{CL}(\Delta, Q_{C^*})]\}.$$

Now consider the following modified version of the design algorithm:

Step 0. Let $k = 0$. Pick a controller $C_0(z)$ and a large value θ.

Step 1. Draw sample $\triangle^k$. Given $G(z, \triangle^k)$, compute $T^1_{\triangle^k}(z)$, $T^2_{\triangle^k}(z)$, $T^3_{\triangle^k}(z)$ as described in [47].

Step 2. Let $Q_k(z)$ be such that the closed loop transfer function using controller $C_k(z)$ is

$$T_{CL}(\triangle^k,\, C_k) = T^1_{\triangle^k} + T^2_{\triangle^k} Q_k T^3_{\triangle^k}.$$

Step 3. Perform update

$$Q_{k+1} = \pi_{s,\theta}[Q_k - \varepsilon_k M_k \partial_Q \|Q\|_2^2|_{Q_{k,us}} - \varepsilon_k \partial_Q g(T_{CL}(\triangle^k, Q))|_{Q_{k,s}}]. \qquad (6.7)$$

Step 4. Determine the controller $C_{k+1}(z)$ so that

$$Q_{\triangle^k,C_{k+1}} = Q_{k+1}.$$

Step 5. Let $k = k+1$. Go to Step 1.

Since

$$\partial_Q \|Q\|_2^2|_{Q_{k,us}} = \frac{Q_{k,us}}{\pi}$$

is unstable and is eliminated after the stabilization projection in Equation (6.7), the above algorithm is equivalent to the one proposed in Section 6.2.2. Define

$$\overline{Q}_{k+1} \doteq Q_k - \varepsilon_k \partial_Q g(T_{CL}(\triangle^k, Q))|_{Q_{k,s}} - \varepsilon_k M_k \partial_Q \|Q\|_2^2|_{Q_{k,us}}.$$

With the structure of (6.1) in mind, let

$$\begin{aligned}
\overline{Q}_{k+1,s} &\doteq Q_{k,s} - \varepsilon_k \partial_Q g(T_{CL}(\triangle^k, Q))|_{Q_{k,s}} \\
\overline{Q}_{k+1,us} &\doteq Q_{k,us} - \varepsilon_k M_k \partial_Q \|Q\|_2^2|_{Q_{k,us}}.
\end{aligned}$$

For simplicity, $T_{CL}(\triangle, Q)$ will be denoted as $T_{\triangle,Q}$ in the following deduction. Now we have

$$\begin{aligned}
& d^2(\overline{Q}_{k+1}, Q^*_{\triangle^k}) \\
&= \|\overline{Q}_{k+1,s} - Q^*_{\triangle^k}\|_2^2 + \|\overline{Q}_{k+1,us}\|^2 \\
&= \|Q_{k,s} - Q^*_{\triangle^k}\|_2^2 + \varepsilon_k^2 \|\partial_Q g[T_{\triangle^k,Q}]|_{Q_{k,s}}\|_2^2 - 2\varepsilon_k \langle Q_{k,s} - Q^*_{\triangle^k}, \partial_Q g[T_{\triangle^k,Q}]|_{Q_{k,s}}\rangle + \|Q_{k,us}\|^2 \\
&+ \varepsilon_k^2 M_k^2 \|\partial_Q \|Q\|_2^2|_{Q_{k,us}}\|^2 - 2\varepsilon_k M_k \langle Q_{k,us}, \partial_Q \|Q\|_2^2|_{Q_{k,us}}\rangle \\
&\le d^2(Q_k, Q^*_{\triangle^k}) + \varepsilon_k^2 (\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2 + M_k^2 \|\partial_Q \|Q\|_2^2|_{Q_{k,us}}\|^2) - 2\varepsilon_k (g(T_{\triangle^k,Q_{k,s}}) \\
&+ M_k \|Q_{k,us}\|^2 - g(T_{\triangle^k,Q^*})).
\end{aligned}$$

Since $d^2(Q_{k+1}, Q^*_{\triangle^k}) \le d^2(\overline{Q}_{k+1}, Q^*_{\triangle^k})$, we get

$$\begin{aligned}
d^2(Q_{k+1}, Q^*_{\triangle^k}) \le\ & d^2(Q_k, Q^*_{\triangle^k}) - 2\varepsilon_k (g(T_{\triangle^k,Q_{k,s}}) + M_k \|Q_{k,us}\|^2 - g(T_{\triangle^k,Q^*})) \\
& + \varepsilon_k^2 (\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2 + M_k^2 \|\partial_Q \|Q\|_2^2|_{Q_{k,us}}\|^2).
\end{aligned}$$

Consider the definition of *robust controller gap* provided in Section 5.4.2. Then, the equation above implies that

$$\begin{aligned} r_{gap}(C_{k+1},C^*) \quad &\leq \quad r_{gap}(C_k,C^*) - 2\varepsilon_k(g(T_{\triangle^k,Q_{k,s}}) + M_k\|Q_{k,us}\|^2 - g(T_{\triangle^k,Q^*})) \\ &\quad + \varepsilon_k^2(\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2 + M_k^2\|\partial_Q\|Q\|_2^2|_{Q_{k,us}}\|^2) + V_k. \end{aligned}$$

where

$$\mathbf{E}[V_k|C_{k+1},C_k,C^*] = 0.$$

Now let $\mathcal{F}_k = \sigma(r_{gap}(C_1,C^*),\ldots,r_{gap}(C_k,C^*))$ be the σ-algebra generated by $r_{gap}(C_1,C^*)$, $r_{gap}(C_2,C^*)$, ..., $r_{gap}(C_k,C^*)$. Take the expectation conditioned on $\mathcal{F}_k$, then,

$$\begin{aligned} &\mathbf{E}\{r_{gap}(C_{k+1},C^*)|\mathcal{F}_k\} \\ &\leq r_{gap}(C_k,C^*) + \varepsilon_k^2(\mathbf{E}\{\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2\} + M_k^2\mathbf{E}\{\|\partial_Q\|Q\|_2^2|_{Q_{k,us}}\|^2\}) \\ &+2\varepsilon_k(\mathbf{E}\{g(T_{\triangle^k,Q^*})\} - \mathbf{E}\{g(T_{\triangle^k,Q_{k,s}})\} - M_k\mathbf{E}\{\|Q_{k,us}\|^2\}) \\ &\leq r_{gap}(C_k,C^*) - 2\varepsilon_k(F_{aux}(C_k) - F_{aux}(C^*)) \\ &+\varepsilon_k^2(\mathbf{E}\{\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2\} + M_k^2\mathbf{E}\{\|\partial_Q\|Q\|_2^2|_{Q_{k,us}}\|^2\}). \end{aligned} \tag{6.8}$$

When $\mathbf{E}\{\|Q_{C_k,us}\|\} > \delta$, one gets

$$\begin{aligned} F_{aux}(C_k) - F_{aux}(C^*) \quad &= \quad \mathbf{E}\{g(T_{\triangle^k,Q_{k,s}})\} + M_k\mathbf{E}\{\|Q_{k,us}\|^2\} - \mathbf{E}\{g(T_{\triangle^k,Q^*})\} \\ &= \quad \mathbf{E}\{g(T_{\triangle^k,Q_{k,s}})\} + \mathbf{E}_\Delta\{g(T_{\triangle^k,Q^*})\} + \eta - \mathbf{E}\{g(T_{\triangle^k,Q^*})\} \\ &> \quad 0 \end{aligned}$$

On the other hand, we have $\|Q_{k,s}\|_2 \leq \tau$, for all $k \geq 2$. Thus, by (3) of Assumption 6.2.1,

$$\begin{aligned} &\mathbf{E}\{\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2\} + M_k^2\mathbf{E}\{\|\partial_Q\|Q\|^2|_{Q_{k,us}}\|^2\} \\ &\leq N^2 + \frac{(\mathbf{E}_\Delta\{g\left[T_{\triangle,Q_{C^*}}\right]\} + \eta)^2}{\pi^2\mathbf{E}_\Delta\{\|Q_{C_k,us,\theta}\|^2\}} \\ &< N^2 + \frac{(\mathbf{E}_\Delta\{g\left[T_{\triangle,Q_{C^*}}\right]\} + \eta)^2}{\pi^2\delta^2} \end{aligned}$$

which is finite. For the case when $\mathbf{E}\{\|Q_{C_k,us}\|\} \le \delta$, (1) of assumption 6.2.1 implies that

$$\begin{aligned} F_{aux}(C_k) - F_{aux}(C^*) &= \mathbf{E}\{g(T_{\triangle^k,Q_{k,s}})\} - \mathbf{E}\{g(T_{\triangle^k,Q^*})\} \\ &\geq 0. \end{aligned}$$

and

$$\begin{aligned} &\mathbf{E}\{\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2\} + M_k^2 \mathbf{E}\{\|\partial_Q \|Q\|_2^2|_{Q_{k,us}}\|^2\} \\ &= \mathbf{E}\{\|\partial_Q g(T_{\triangle^k,Q})|_{Q_{k,s}}\|_2^2\} \\ &\leq N^2. \end{aligned}$$

Hence, $r_{gap}(C_k, C^*)$ satisfies the conditions in Lemma D.0.1 and we get

$$\sum_{k=1}^{\infty} \varepsilon_k (F_{aux}(C_k) - F_{aux}(C^*)) < \infty \; w.p.1.$$

Since $\sum_{k=1}^{\infty} \varepsilon_k = \infty$, we must have

$$F_{aux}(C_k) - F_{aux}(C^*) \to 0 \quad w.p.1.$$

Thus, with probability 1,

$$\mathbf{E}_\Delta\{g\left[T_{\triangle,Q_{C_k,s}}\right]\} \to \mathbf{E}_\Delta\{g\left[T_{\triangle,Q_{C^*}}\right]\} \tag{6.9}$$

and

$$M_k \mathbf{E}_\Delta\{\|Q_{C_k,us}\|^2\} \to 0.$$

Now by (6.5), there exists $k' \geq 0$ such that

$$\mathbf{E}\{Q_{C_k,us}\} = 0, \; \forall k \geq k'$$

or it will contradict (6.9). Q.E.D.

6.3 Controller Design Algorithm 2 – Decomposition Algorithm

In this section, we present another design approach where there is no assumption on the existence of a robustly stabilizing controller. Therefore, a controller is designed by penalizing the "size" of the unstable region.

6.3.1 Problem Statement

Given $\vartheta > 0$, let

$$B_\vartheta \doteq \{C \in \mathcal{G} : \ \|Q_{C,s}\|_2 \leq \vartheta \ \text{ and } \|Q_{C,us}\| \leq \vartheta, \ \forall \triangle \in \Delta\}.$$

Consider the discrete-time closed loop linear time invariant system depicted in Figure 5.1. Given an uncertainty value $\triangle \in \Delta$ and a controller $C(z)$, let $T_{\triangle,C}(z)$ be the corresponding closed loop transfer function matrix. We aim at designing a controller $C^*(z) \in B_\vartheta$ minimizing the following objective function:

$$U(C) \doteq \mathbf{E}_\Delta[g(T_{\triangle,Q_{C,s}})] + \ \mu\mathbf{E}_\Delta[\|Q_{C,us}\|], \tag{6.10}$$

where $g : H_2 \to \mathbf{R}$ is a convex function and $\mu > 0$ denotes the penalty on the norm of an unstable transfer function as defined in (6.3).

We make the following assumption on the problem being considered in this section:

ASSUMPTION 6.3.1. *There exist $\eta > \vartheta > 0$ such that*

1. *For any controller C, if there is a $\triangle_0 \in \Delta$ such that max$\{\|Q_{C,\triangle_0,s}\|_2, \|Q_{C,\triangle_0,us}\|\} \leq \vartheta$, we have*

   $$max\{\|Q_{C,\triangle,s}\|_2, \|Q_{C,\triangle,us}\|\} \ \leq \ \eta, \ \forall \triangle \in \Delta.$$

   If $C \in B_\eta$, we have

   $$U(C^*) \leq U(C). \tag{6.11}$$

   where

   $$C^* \ \doteq \ argmin_{C \in B_\vartheta} U(C).$$

2. *Also, we assume that there is a constant $N \ > 0$ such that if $\|Q\|_2 \leq \eta$*

   $$\|\partial_Q g(T_{\triangle,Q})\|_2 < \ N, \ \forall \triangle \in \Delta.$$

6.3.2 Design Algorithm II

We now present an algorithm that converges to a solution of the problem described above.

ALGORITHM 6.3.1.

Step 0. *Let* $k = 0$*. Pick a controller* $C_0(z)$ *and a large value* ϑ*.*

Step 1. *Draw sample* $\triangle^k$*. Given* $G(z, \triangle^k)$*, compute* $T^1_{\triangle^k}(z)$*,* $T^2_{\triangle^k}(z)$*,* $T^3_{\triangle^k}(z)$ *as described in [47].*

Step 2. *Let* $Q_k(z)$ *be such that the closed loop transfer function using controller* $C_k(z)$ *is*

$$T_{CL}(\triangle^k,\, C_k) = T^1_{\triangle^k} + T^2_{\triangle^k} Q_k T^3_{\triangle^k}$$

Step 3. *Do the decomposition*

$$Q_k(z) \;=\; Q_{k,s}(z) + \; Q_{k,us}(z),$$

Step 4. *Perform update*

$$\bar{Q}_{k+1} = Q_k - \varepsilon_k \left[\frac{\partial g(T_{\triangle^k,Q})}{\partial Q}|_{Q_{k,s}} + \mu \frac{\partial \|Q\|_2}{\partial Q}|_{Q_{k,us}} \right].$$

Step 5. *Determine the controller* $C_{k+1}(z)$ *so that*

$$Q_{\triangle^k, C_{k+1}} = \psi_\vartheta(\bar{Q}_{k+1}).$$

Step 6. *Let* $k = k+1$*. Go to Step 1.*

6.3.3 Convergence Theorem

We now establish the convergence of Algorithm 6.3.1.

THEOREM 6.3.1. *Let* $g : H_2 \to \mathbf{R}^+_0$ *be a convex function with subgradient* $\partial g \in RH_2$ *and let* $C^* \in B_\vartheta$ *be the optimal controller of (6.10). Then, if Assumption 6.3.1 holds and*

$$\varepsilon_k > 0, \;\; \sum_{k=1}^{\infty} \varepsilon_k = \; \infty, \;\; \sum_{k=1}^{\infty} \varepsilon_k^2 = \; \beta < \; \infty$$

Algorithm 6.3.1 generates a sequence of controllers C_k for which with probability 1

$$U(C_k) \;\rightarrow\; U(C^*).$$

Proof: Given Algorithm 6.3.1, we have

$$\begin{aligned}
& d^2(Q_{k+1}, Q_{C^*,\triangle^k}) \\
& \leq d^2(\bar{Q}_{k+1}, Q_{C^*,\triangle^k}) \\
& = \|Q_k - Q_{C^*,\triangle^k} - \varepsilon_k[\frac{\partial g(T_{\triangle^k,Q})}{\partial Q}|_{Q_{k,s}} + \mu\frac{\partial \|Q\|_2}{\partial Q}|_{Q_{k,us}}]\|^2 \\
& \leq d^2(Q_k, Q_{C^*,\triangle^k}) + \mu^2\varepsilon_k^2\|\frac{\partial \|Q\|_2}{\partial Q}|_{Q_{k,us}}\|^2 + \varepsilon_k^2\|\frac{\partial g(T_{\triangle^k,Q})}{\partial Q}|_{Q_{k,s}}\|_2^2 \\
& -2\varepsilon_k(Q_k - Q_{C^*,\triangle^k})'[\frac{\partial g(T_{\triangle^k,Q})}{\partial Q}|_{Q_{k,s}} + \mu\frac{\partial \|Q\|_2}{\partial Q}|_{Q_{k,us}}] \\
& \leq \varepsilon_k^2(N^2 + \frac{\mu^2}{4\pi^2}) - 2\varepsilon_k(Q_{k,s} - Q_{C^*,\triangle^k,s})'\frac{\partial g(T_{\triangle^k,Q})}{\partial Q}|_{Q_{k,s}} \\
& -2\mu\varepsilon_k(Q_{k,us} - Q_{C^*,\triangle^k,us})'\frac{\partial \|Q\|_2}{\partial Q}|_{Q_{k,us}} + d^2(Q_k, Q_{C^*,\triangle^k}) \\
& < d^2(Q_k, Q_{C^*,\triangle^k}) - 2\varepsilon_k[g(T_{\triangle^k,Q_{k,s}}) - g(T_{\triangle^k,Q_{C^*,\triangle^k,s}}) + \mu(\|Q_{k,us}\| - \|Q_{C^*,\triangle^k,us}\|)] \\
& +\varepsilon_k^2(N^2 + \frac{\mu^2}{4\pi^2})
\end{aligned}$$

With the aid of *robust controller gap* as defined in Section 5.4.2, we have

$$\begin{aligned}
& r_{gap}(C_{k+1}, C^*) \\
& < r_{gap}(C_k, C^*) - 2\varepsilon_k[g(T_{\triangle^k,Q_{k,s}}) - g(T(\triangle^k, Q_{C^*,\triangle^k,s})) \\
& +\mu\|Q_{k,us}\| - \mu\|Q_{C^*,\triangle^k,us}\| + \frac{\varepsilon_k}{2}(N^2 + \frac{\mu^2}{4\pi^2})] + V_k.
\end{aligned}$$

Let $\mathcal{F}_k$ be the smallest σ-algebra generated by $r_{gap}(C_1, C^*), \ldots, r_{gap}(C_k, C^*)$. Take expectations conditioned on $\mathcal{F}_k$ and obtain

$$\begin{aligned}
& \mathbf{E}[r_{gap}(C_{k+1}, C^*)|\mathcal{F}_k] \\
& \leq r_{gap}(C_k, C^*) + \varepsilon_k^2(N^2 + \frac{\mu^2}{4\pi^2}) - 2\varepsilon_k[U(C_k) - U(C^*)].
\end{aligned}$$

From 1 in Assumption 6.3.1, we have

$$U(C_k) - U(C^*) \geq 0.$$

Hence, by Lemma D.0.1, we have

$$\sum_{k=2}^{\infty} \varepsilon_k [U(C_k) - U(C^*)] \ < \ \infty \ \ w.p.1.$$

Thus,

$$U(C_k) \ \rightarrow \ U(C^*) \ \ w.p.1. \tag{6.12}$$

Q.E.D.

6.4 Numerical Examples

We again consider a DC armature-controlled servomotor whose transfer function is

$$P(s, \triangle) = \frac{\omega^2}{s(s + 2\delta\omega)}.$$

To facilitate the control of the motor by a digital microprocessor, we consider the H_2 control problem in discrete time. Given the weighting function

$$W(z) = \frac{0.06667z + 0.01333}{z - 0.6},$$

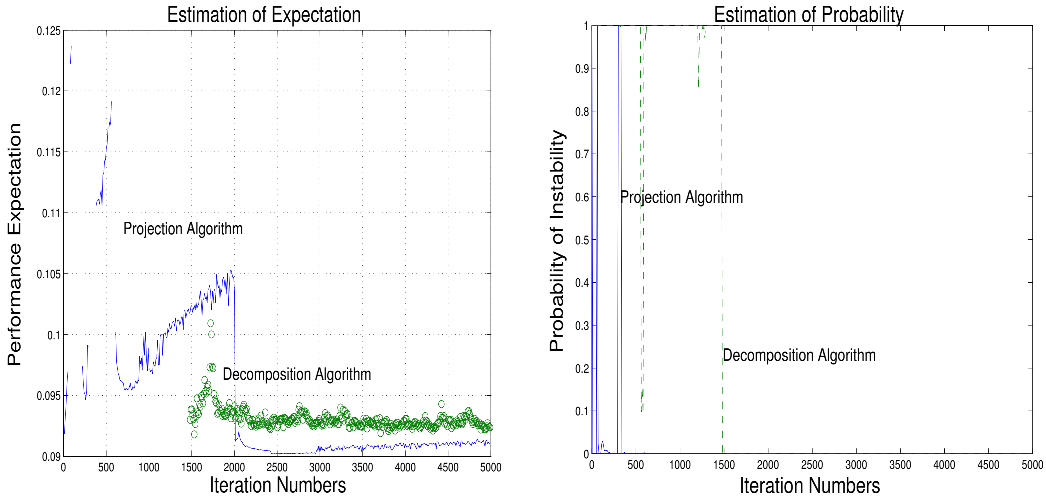
we aim at finding a robustly stabilizing controller $C(z)$ that minimizes

$$\mathbf{E}\|W(z)(1 + C(z)P(z, \triangle))^{-1}\|_2$$

The sample time T is 0.5s.

We assume both ω and δ are uncertain parameters; i.e.,

$$\omega = \omega_0 + \triangle\omega, \ \delta = \delta_0 + \triangle\delta.$$



(a) Performance Expectation (b) Risk of Instability

Fig. 6.1. MC Simulations: DC Motor Optimal Design

In simulations, we take $\omega_0 = 6$, $\delta_0 = 0.3$ and $\triangle\omega, \triangle\delta$ are uniformly distributed on $[-r, r]$ and $[-\varepsilon r, \varepsilon r]$ respectively. In our example, we take $r = 1$, $\varepsilon = 0.1$. Using *Matlab* function *dh2lqg*, we obtain the following nominal controller

$$C_{nom}(z) = \frac{0.6754z^2 - 0.7109z + 0.03555}{z^3 + 1.4z^2 - 0.2001z - 0.5999}$$

which results in a nominal weighted H_2 performance to be 0.0667. After 5000 iterations, the following controllers were obtained by Algorithm 6.2.1 and Algorithm 6.3.1 respectively

$$C_1(z) = \frac{0.2326z^3 - 0.1748z^2 + 0.0225z - 0.0008}{z^4 + 1.049z^3 + 0.0552z^2 - 0.0958z - 0.0023}$$

$$C_2(z) = \frac{0.0195z^3 + 0.0441z^2 - 0.0075z + 0.0004}{z^3 + 0.6706z^2 + 0.1536z + 0.003}$$

Monte Carlo simulations were performed to estimate the risk of instability. The number of samples used was 5000. The estimated risk of instability for controllers $C_1(z)$ and $C_2(z)$ is zero. As a comparison, the estimated risk of instability when using controller C_{nom} is about 37.36%.

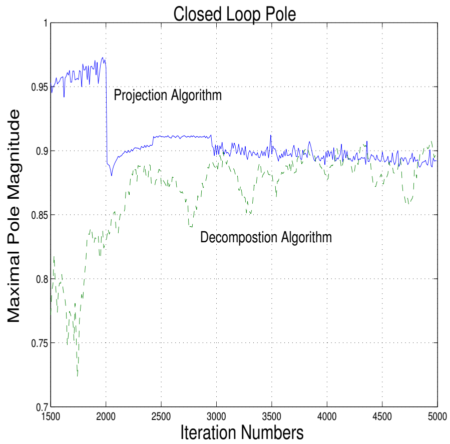


Fig. 6.2. Maximal magnitude of closed loop poles

Next, we investigate the asymptotic characteristics of the risk of performance violation. We run Monte Carlo simulation for each controller obtained during the optimization process. The asymptotic behavior of estimated performance expectation, estimated instability probability and estimated maximal magnitude of closed loop poles are shown in Figure 6.1(a), Figure 6.1(b) and Figure 6.2, respectively. Based on the simulation results, we can conclude that both Algorithm 6.2.1 and Algorithm 6.3.1 can minimize their respective objective functions and, a controller with an estimated risk of instability of zero is obtained by both algorithms.

6.5 Concluding Remarks

In this chapter, we addressed the problem of optimal robust controller design for arbitrary uncertainty structures. Two algorithms were presented. In contrast to the suboptimal algorithms in Chapter 5, the two algorithms presented do not need *a priori* knowledge of an achievable performance level. Almost sure convergence of these algorithms was shown.

Chapter 7

Probabilistic Robust Stabilization

In Chapter 3, probabilistic quadratic stabilization was discussed. However, it has been shown that quadratic stability can be very conservative since it requires one Lyapunov function to work for the whole plant family [33]. Moreover, it is well known that there are systems which are stable but not quadratic stabilizable [11]. This provides the motivation for the problem addressed in this chapter: robust stabilization via static linear state feedback. The main obstacle to solving this problem is the fact that the problem is, again, a non-convex design problem. In this chapter, as in the rest of the thesis, we address the robust stabilizing controller design problem within a probabilistic framework.

7.1 Introduction

Consider a system with uncertain parameters

$$\triangle = (\triangle_1, \triangle_2, \ldots, \triangle_\ell),$$

$\triangle \in \Delta$, where Δ is the admissible uncertainty set. In this chapter, the system is subjected to static parametric uncertainty. More precisely, we assume that the system is described by the state space model

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

where the pair $(A(\triangle), B(\triangle))$ is assumed to be controllable for all admissible values of uncertainty $\triangle \in \Delta$.

As before, the uncertain parameters $\triangle_i$ are treated as random variables and the objective is to design a controller of the form $u = Kx$ leading to robust stability. We provide an algorithm based on classical stochastic approximation which solves the robust stabilization problem. Convergence to an optimal state feedback gain is shown; i.e., gain minimizing a given nonnegative objective function which is zero if the risk of instability is zero.

A lot of effort has been put into the subject of robust stability and many important results have been obtained; e.g., see [66, 67]. In this chapter, we address the problem of probabilistic robust stabilization which, by definition, is less restrictive than quadratic stability design. We provide an iterative design algorithm which is similar to that in Chapter 5 in structure. In contrast to the results in Chapter 3, we do not require any assumptions on the uncertainty structure of $A(\triangle)$, distribution of $\triangle$ or stability of nominal system A_0. For simplicity of presentation, we assume that B is not subjected to uncertainty. However, our algorithm can also handle the case where A and B are both uncertain.

7.2 Probabilistic Robust Stability

In this section, we provide a precise definition of the problem addressed in this chapter. To this end, we first briefly review the concept of robust stability.

7.2.1 Robust Stability

Consider the system

$$\dot{x} = A(\triangle)x + Bu$$

with n-dimensional state x and m-dimensional control u. Now the system above is robustly stabilizable via linear static state feedback if and only if there exists a state feedback matrix $K \in \mathrm{R}^{m\times n}$ such that for any admissible $\triangle$ there is a matrix $P(\triangle) > 0$,

$$L_{K,P}(\triangle) \doteq (A(\triangle)+BK)P(\triangle)+P(\triangle)(A(\triangle)+BK)^T = -I, \tag{7.1}$$

where I is an identity matrix with compatible dimensions. In other words, the feedback control law

$$u = Kx.$$

Hence, we can define the feasible set for the problem above

$$K_S \doteq \{K \in \mathrm{R}^{m\times n} : \forall \text{ admissible } \triangle \in \Delta,\ \exists P(\triangle) \in \mathrm{R}^{n\times n} \text{ such that } L_{K,P}(\triangle) = -I\}.$$

7.2.2 Probabilistic Robust Stability

Finding a robust stabilizing controller is, in general, a computationally complex problem. Therefore, in this chapter, we take a different approach. Using the well known change of variables $W(\triangle) \doteq KP(\triangle)$, (7.1) can be expressed as

$$A(\triangle)P(\triangle)+P(\triangle)A^T(\triangle)+BW(\triangle)+W(\triangle)^T B^T = -I, \tag{7.2}$$

which is a linear matrix equality in $P(\triangle)$ and $W(\triangle)$. Thus our problem can be formulated as finding K^* such that

$$K^* = \arg\min \mathbf{E}H(P_K(\triangle)) \doteq \arg\min \mathbf{E}[\max\{0,\ \lambda_{\max}(-P_K(\triangle))\}], \tag{7.3}$$

subject to (7.2) and $W(\triangle) = KP(\triangle)$. By [12], the objective function $H(P_K(\triangle))$ is a convex function in $P_K(\triangle)$.

7.3 Numerical Algorithms for Probabilistic Robust Stability

The following lemma provides the condition for uniqueness of $P(\triangle)$ in (7.1) for a given K and $\triangle$ [4]:

LEMMA 7.3.1. *For $A \in R^{m\times m}$, $B \in R^{n\times n}$ and $C \in R^{m\times n}$, the matrix equation*

$$AX + XB = C$$

has a unique solution if and only if the eigenvalues $\alpha_1, \alpha_2, \ldots, \alpha_m$ of A and $\beta_1, \beta_2, \ldots, \beta_n$ of B satisfy

$$\alpha_i + \beta_j \neq 0 \ (\ i = 1,2,\ldots,m;\ j = 1,2,\ldots,n).$$

In our case, in order to test the uniqueness of $P(\triangle)$, we just need to check if $A(\triangle)+BK$ has any eigenvalue on the imaginary axis or any pair of eigenvalues symmetric to the imaginary axis.

We are now ready to present a design algorithm which solves the problem described above:

ALGORITHM 7.3.1.

Step 0. *Let* $i = 0$*. Pick a state feedback gain* K^0 *and a maximal iteration number* N_{max}*.*

Step 1. *Draw sample* $\triangle^i$*. Compute* $P^i_{\triangle^i}$ *according to*

$$(A(\triangle^i) + BK^i)P^i_{\triangle^i} + P^i_{\triangle^i}(A(\triangle^i) + BK^i)^T = -I \tag{7.4}$$

Step 2. *Do the transformation* $W^i = K^iP^i$ *where* $P^i = [P^i_1 \; P^i_2 \; \ldots \; P^i_n]$ *and* $W^i = [W^i_1 \; W^i_2 \; \ldots \; W^i_n]$*. Thus (7.4) can be expressed as*

$$A(\triangle^i)P^i + P^iA^T(\triangle^i) + BW^i + {W^i}^T B^T = -I \tag{7.5}$$

Step 3. *Let* $X^i_{\triangle^i} = [P^i_1 \; P^i_2 \; \ldots \; P^i_n \; W^i_1 \; W^i_2 \; \ldots \; W^i_n]'$ *and rewrite (7.5) in the form of*

$$G^iX^i_{\triangle^i} = g. \tag{7.6}$$

Step 4. *Let* F^i *be the orthonormal basis of the null space of* G^i*, i.e.,* $\mathcal{N}(G^i)$*. Then,*

$$Z^i_{\triangle^i} = {F^i}^T(X^i_{\triangle^i} - X^{i0}) \tag{7.7}$$

where X^{i0} *is any vector on hyperplane (7.6). For example, we can take* $X^{i0} = X^i_{\triangle^i}$*.*

Step 5. *Perform the following update*

$$Z^{i+1}_{\triangle^i} = \begin{cases} Z^i_{\triangle^i} & \text{if } \lambda_{\max}(-P^i) < 0 \\ Z^i_{\triangle^i} - \varepsilon_i \frac{\partial \lambda_{max}(-P^i)}{\partial Z^i}\Big|_{Z^i_{\triangle^i}} & \text{if } \lambda_{\max}(-P^i) \geq 0, \end{cases} \tag{7.8}$$

where $\frac{\partial \lambda_{max}(-P^i)}{\partial Z^i}$ *can be computed according to Appendix B.*

Step 6. *Compute* $X^{i+1}_{\triangle^i} = F^iZ^{i+1}_{\triangle^i} + X^{i0}$ *and get* P^{i+1}*,* W^{i+1} *thus* K^{i+1}*.*

Step 7. *If* $i + 1 \geq N_{max}$*, stop. Or let* $i = i + 1$*. Go to Step 1.*

7.4 Convergence of the Stochastic Approximation Algorithm

We now present the main result in this chapter which provides the conditions under which Algorithm 7.3.1 converges:

THEOREM 7.4.1. *Consider*

$$\dot{x} = A(\triangle)x + Bu,$$

where $A \in \mathbf{R}^{n\times n}$, $x \in \mathbf{R}^{n\times 1}$, $B \in \mathbf{R}^{n\times m}$, $u \in \mathbf{R}^{m\times 1}$ *and* $\triangle$ *is a bounded random variable with a given probabilistic distribution. Suppose there exists a set* $\mathcal{K}^*$, *such that for any* $K^* \in \mathcal{K}^*$ *and* $u = K^*x$,

$$K^* = arg\min\ \mathbf{E}\, H(P_K(\triangle)) \doteq arg\min\ \mathbf{E}[\max\{0,\ \lambda_{\max}(-P_K(\triangle))\}]$$

where

$$(A(\triangle) + BK^*)P(\triangle) + P(\triangle)(A(\triangle) + BK^*)^T = -I.$$

Also define $\mathcal{K}' \doteq \{K \notin \mathcal{K}^* : Prob[\lambda_{\max}(-P_K) \geq 0] = 0\}$. *Assume the condition leading to multiple solutions in Lemma 7.3.1 happens with probability 0. Let*

$$\varepsilon_i > 0,\ \sum_{i=1}^{\infty} \varepsilon_i = \ \infty,\ \sum_{i=1}^{\infty} \varepsilon_i^2 = \ \beta < \ \infty.$$

Then, Algorithm 7.3.1 provides a sequence of $\{K^i\}$ *that, with probability 1 satisfies*

$$\mathbf{E}[H(P_{K^i})]\ \rightarrow \mathbf{E}[H(P_{K^*})]$$

or there exists an instant $i_0 > 0$ *such that* $K^i\ \in\ \mathcal{K}'$ *for all* $i \geq i_0$.

In order to prove the result above, one needs a measure of how far a state feedback gain is from the optimal. Similar to the definition of *robust controller gap* in Section 5.4, let f be the probability density function used to generate the samples in the design algorithm. Then, given two state feedback gains K_1 and K_2, the robust gain gap is

$$r_{gap}(K_1, K_2) = \int_{\Delta} \|Z_{\triangle,K_1} - Z_{\triangle,K_2}\|^2 f(\triangle) d\triangle.$$

where $Z_{\triangle,K}$ can be computed according to Algorithm 7.3.1.

Remarks

1. Since in most of our design applications, we sample the uncertainty over a continuous interval. the assumption in Theorem 7.4.1 on almost sure uniqueness of $P(\triangle)$ is a very mild one.

2. As long as the eigenvalues of $A+BK$ satisfy the conditions stated in Lemma 7.3.1, for a given uncertainty sample $\triangle \in \Delta$, there is a unique $P(\triangle)$ corresponding to a given K. Thus $Z_{\triangle,K}$ can also be uniquely determined. Also, due to the subtraction structure in the integrand in the definition of *robust gain gap*, the choice of X^{i0} during each iteration has no effect on the "distance" between two state feedback strategies.

3. Given three gains K_1, K_2 and K^*, we have

$$\begin{aligned}&\|Z_{\triangle,K_1} - Z_{\triangle,K^*}\|^2 - \|Z_{\triangle,K_2} - Z_{\triangle,K^*}\|^2 \\ &= r_{gap}(K_1,K_*) - r_{gap}(K_2,K_*) + V\end{aligned}$$

   with

$$\mathbf{E}[V|K_1,K_2,K^*] = 0.$$

   where, $\mathbf{E}[X|P]$ denotes the conditional expectation of X given P.

Proof: First, according to the assumption in the theorem, the uniqueness condition for $P(\triangle)$ holds with probability 1. From now on, we only consider the case when given state feedback K and uncertainty sample $\triangle$, $P(\triangle)$ as a solution for (7.9) can be uniquely determined. The robust stabilization problem is formulated as finding K minimizing

$$\mathbf{E}H(P_K(\triangle)) \doteq \mathbf{E}[\max\{0,\ \lambda_{\max}(P_K(\triangle))\}]$$

subject to

$$A(\triangle)P(\triangle) + P(\triangle)A^T(\triangle) + BW(\triangle) + W(\triangle)^T B^T = -I, \forall \triangle \in \Delta,$$

where $W(\triangle) = KP(\triangle)$ and

$$(A(\triangle)+BK)P(\triangle) + P(\triangle)(A(\triangle)+BK)^T = -I, \forall \triangle \in \Delta. \tag{7.9}$$

As in the algorithm, denoting $X^i_{\triangle^i} = [P^i_1\ P^i_2\ \ldots\ P^i_n\ W^i_1\ W^i_2\ \ldots\ W^i_n]'$, (7.5) can be expressed in the form of $G^i X^i_{\triangle^i} = g$ which is a hyperplane in $\mathbf{R}^{n\times n+m\times n}$. Let F^i be the orthonormal basis of the null space of G^i, i.e., $\mathcal{N}(G^i)$, $F^i Z + X^i$ characterizes all vectors in this hyperplane [12]. If $K^{i'} \in \mathcal{K}'$ as defined in the theorem, then with probability 1, $K^{i'}$ remains in $\mathcal{K}'$. More specifically,

$$K^i = K^{i'},\quad \forall i \geq i'\ \ w.p.1.$$

From now on, we focus on the case when $K^i \notin \mathcal{K}'$. Thus, from this point on, we only consider the case $\text{Prob}\{\lambda_{\max}(-P_K(\triangle)) \geq 0 : K \neq K^*\} > 0$. As can be seen in expression (7.8), we only update the gain if it does not stabilize the system for the given uncertainty sample. Hence, in the reasoning below we only consider the case when $\lambda_{\max}(-P^i) \geq 0$. In this case, we have

$$\begin{aligned}
&\|Z^{i+1}_{\triangle^i} - Z^*_{\triangle^i}\|^2 \\
&= \|Z^i_{\triangle^i} - \varepsilon_i \frac{\partial \lambda_{max}(-P^i)}{\partial Z^i}|_{Z^i_{\triangle^i}} - Z^*_{\triangle^i}\|^2 \\
&= \|Z^i_{\triangle^i} - Z^*_{\triangle^i}\|^2 + \varepsilon_i^2 \|\frac{\partial \lambda_{max}(-P^i)}{\partial Z^i}|_{Z^i_{\triangle^i}}\|^2 - 2\varepsilon_i \langle Z^i_{\triangle^i} - Z^*_{\triangle^i}, \frac{\partial \lambda_{max}(-P^i)}{\partial Z^i}|_{Z^i_{\triangle^i}} \rangle
\end{aligned}$$

Since there is an affine relationship between entries in $P(\triangle)$ and those in Z as governed in (7.7), $\lambda_{max}(-P(\triangle))$ is a convex function not only in each entry of $P(\triangle)$ but also in those of Z. We have

$$\begin{aligned}
&\|Z^{i+1}_{\triangle^i} - Z^*_{\triangle^i}\|^2 \\
&\leq \|Z^i_{\triangle^i} - Z^*_{\triangle^i}\|^2 + \varepsilon_i^2 \|\frac{\partial \lambda_{max}(-P^i)}{\partial Z^i}|_{Z^i_{\triangle^i}}\|^2 - 2\varepsilon_i (\lambda_{max}(-P^i) - \lambda_{max}(-P^*)) \\
&\leq \|Z^i_{\triangle^i} - Z^*_{\triangle^i}\|^2 + \varepsilon_i^2 \|\frac{\partial \lambda_{max}(-P^i)}{\partial Z^i}|_{Z^i_{\triangle^i}}\|^2 - 2\varepsilon_i (\max\{0, \lambda_{max}(-P^i)\} - \max\{0, \lambda_{max}(-P^*)\})
\end{aligned}$$

The last inequality comes from the fact that $\lambda_{\max}(-P^i) \geq 0$. Given the definition of robust gain gap, the inequality above can be rewritten in the following form

$$\begin{aligned}
&r_{gap}(K^{i+1}, K^*) \\
&\leq\ r_{gap}(K^i, K^*) + \varepsilon_i^2 \|\frac{\partial \lambda_{max}(-P^i)}{\partial Z^i}|_{Z^i_{\triangle^i}}\|^2 - 2\varepsilon_i (\max\{0, \lambda_{max}(-P^i)\} - \max\{0, \lambda_{max}(-P^*)\}) + V_i.
\end{aligned}$$

where

$$\mathbf{E}[V_i|K^{i+1},K^i,K^*] = 0.$$

Now let $\mathcal{F}_i = \sigma(r_{gap}(K^1,K^*),\ldots,r_{gap}(K^i,K^*))$ be the σ-algebra generated by $r_{gap}(K^1,K^*)$, $r_{gap}(K^2,K^*)$, ..., $r_{gap}(K^i,K^*)$. Take expectation conditioned on $\mathcal{F}_i$, then,

$$\begin{aligned}&\mathbf{E}\{r_{gap}(K_{i+1},K^*)|\mathcal{F}_i\}\\ &\leq r_{gap}(K^i,K^*) + \varepsilon_i^2\mathbf{E}\|\frac{\partial\lambda_{max}(-P^i)}{\partial Z^i}|_{Z^i_{\triangle^i}}\|^2 - 2\varepsilon_i(\mathbf{E}[H(P_{K^i})] - \mathbf{E}[H(P_{K^*})]).\end{aligned} \tag{7.10}$$

There is an affine relationship between the entires of Z^i and P^i as given by (7.7). Since all columns of F^i are orthonormal, the affine relationship is norm bounded. According to Appendix B, $\|\frac{\partial\lambda_{min}(P^i)}{\partial Z^i}\|$ is also uniformly bounded, since one uses the eigenvectors of norm one to compute it. Thus $\sum_{i=0}^{\infty}\varepsilon_i^2\mathbf{E}\|\frac{\partial\lambda_{min}(P^i)}{\partial Z^i}\|^2 < \infty$. Also, by definition, $\mathbf{E}[H(P_{K^*})] \geq \mathbf{E}[H(P_{K^i})]$. By the *Supermartingale Convergence Theorem* in Appendix D, we have

$$\sum_{i=1}^{\infty}\varepsilon_i(\mathbf{E}[H(P_{K^*})] - \mathbf{E}[H(P_{K^i})]) \; < \; \infty \; w.p.1.$$

On the other hand,

$$\sum_{i=1}^{\infty}\varepsilon_i \; = \; \infty.$$

Hence, with probability 1, we get

$$\mathbf{E}[H(P_{K^i})] \; \rightarrow \mathbf{E}[H(P_{K^*})].$$

Q. E. D.

7.5 Numerical Examples

7.5.1 Example 1

We again consider the uncertain system first presented in Section 3.7.2

$$\mathbf{A_0} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -5.0625 & -20.25 & -64.125 & -121.5 & -140.0625 & -101.25 & -44.25 & -10.5 \end{pmatrix}$$

and

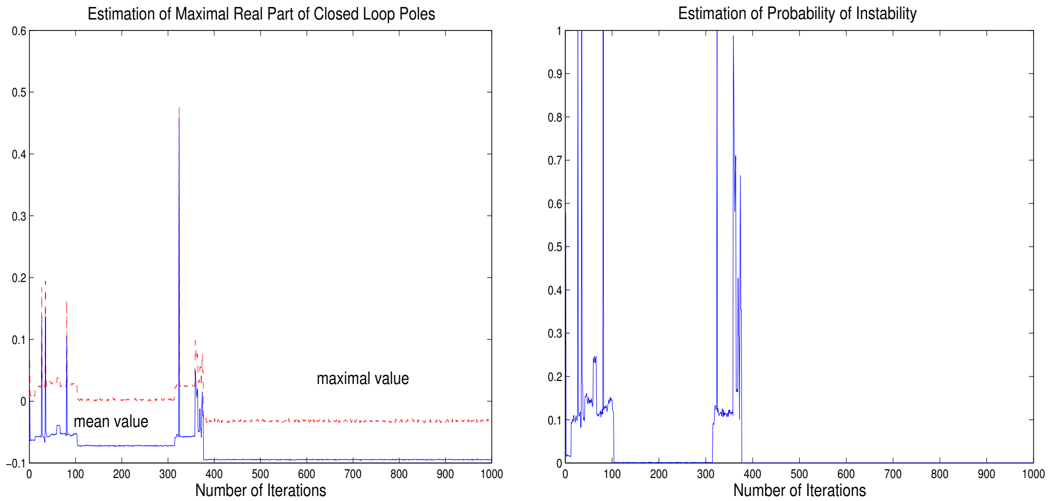
$$\mathbf{B} = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)'$$

$A(8,i) = A_0(8,i) + \triangle_i,\ i = 1,\ldots,8$, where $\triangle_i, i = 1,\ldots,8$ are uniformly distributed over $[-2,2]$. In the simulations, we take $N_{\max} = 1000$, $X^{i0} = X^i_{\triangle^i}$ and $\varepsilon_i = 10/i$. After 1000 iterations, we get

$$K_{N_{\max}} = [2.3009 - 1.5555 - 4.6995 - 3.6160 - 8.4840 - 2.2165 - 8.0852 - 0.6429].$$

After running a 10000 sample Monte Carlo simulation on the closed loop system with $K_{N_{\max}}$ as the state feedback, we get the maximal real part of closed loop poles is -0.0305, the mean value is -0.0943, the minimal real part of eigenvalues of $P(\triangle)$ is 0.0397 and the mean value is 0.0478. This shows that the closed loop system is robustly stable with high probability. Furthermore, we run a 5000 sample Monte Carlo simulation for each gain K^i obtained during the iterations. The results obtained are depicted in Figure 7.1 where the characteristics of the successive closed loop systems are presented.

We can see that the estimated risk of instability becomes 0 after about 400 steps. Compared with the example in Section 3.7.2, the state feedback by our algorithm works well even with a larger size of uncertainty set. We also simulated Algorithm 7.3.1 with the step size proposed by Calafiore and Polyak [15] as given in (5.10). It can be proven that this also results in a convergent algorithm and the simulations show that it results on faster convergence. However, it requires an assumption similar to Assumption 5.3.1 which is hard to verify in practice and is waived in our algorithm.



(a) Maximal Real Part of Poles (b) Risk of Instability

Fig. 7.1. MC Simulations: Our Algorithm

7.5.2 Example 2

This example is based on the example provided in [11] which is proven not to be quadratically stabilizable. It was modified so that the original family of plants is not robustly stable. Consider the system

$$\dot{x} = A(\triangle)x + Bu$$

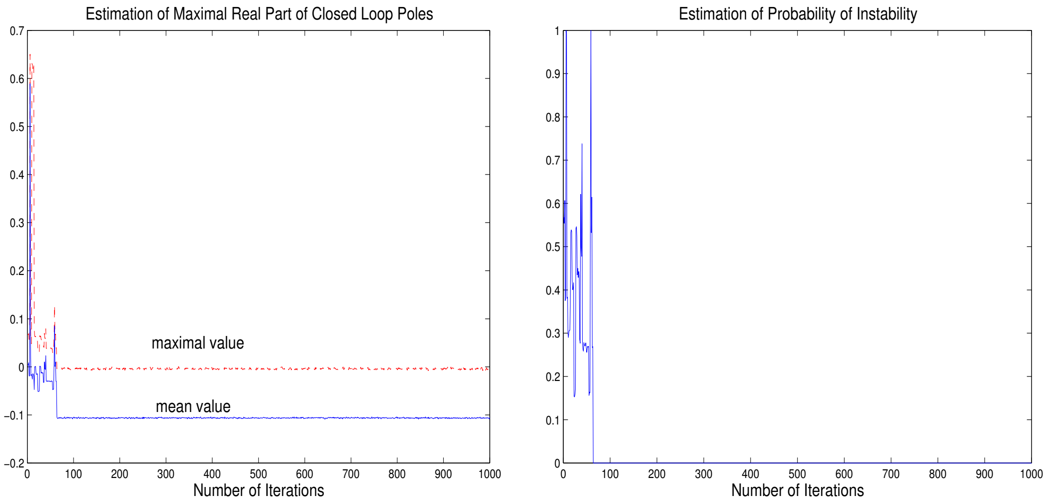
where

$$A(\triangle) = \triangle A_1 + (1-\triangle)A_2, \quad B = [0 \;\; 1]^T,$$

and

$$\mathbf{A_1} = \begin{pmatrix} -100 & 0 \\ 0 & -1 \end{pmatrix},$$

$$\mathbf{A_2} = \begin{pmatrix} 8 & -9 \\ 120 & -180 \end{pmatrix}.$$



(a) Maximal Real Part of Poles (b) Risk of Instability

Fig. 7.2. MC Simulations: Calafiore-Polyak's Stepsize

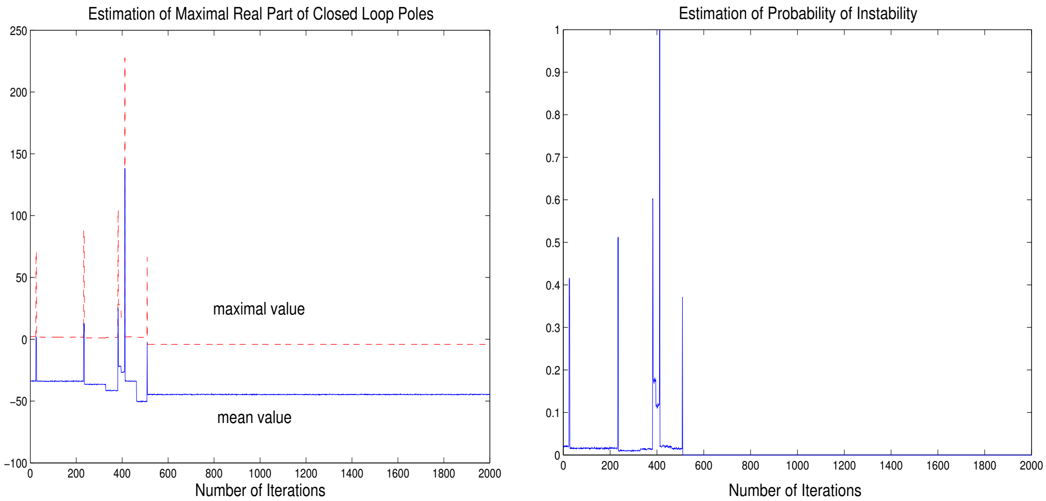
$\triangle$ is uniformly distributed over [0 1]. *MatLab* function *quadstab* shows this system is not quadratically stabilizable. Now, take $N_{\max} = 2000$, $X^{i0} = X^{i}_{\triangle^i}$ and $\varepsilon_i = 100/i$. Using the stochastic subgradient algorithm, after 2000 iterations, we get

$$K = [134.5345 \quad -12.9282].$$

After running a 10000 sample Monte Carlo simulation, we get the maximal real part of closed loop poles is -4.1356, the mean value is -44.831, the minimal real part of eigenvalues of $P(\triangle)$ is 0.0027 and the mean value is 0.0037. As in Example 1, we run a 5000 sample Monte Carlo simulation to get the Figure 7.3 describing the characteristics of closed loop poles with maximal real part and probability of instability as a function of iteration numbers. Also, simulations with Calafiore-Polyak's step size are presented in Figure 7.4 for comparison.

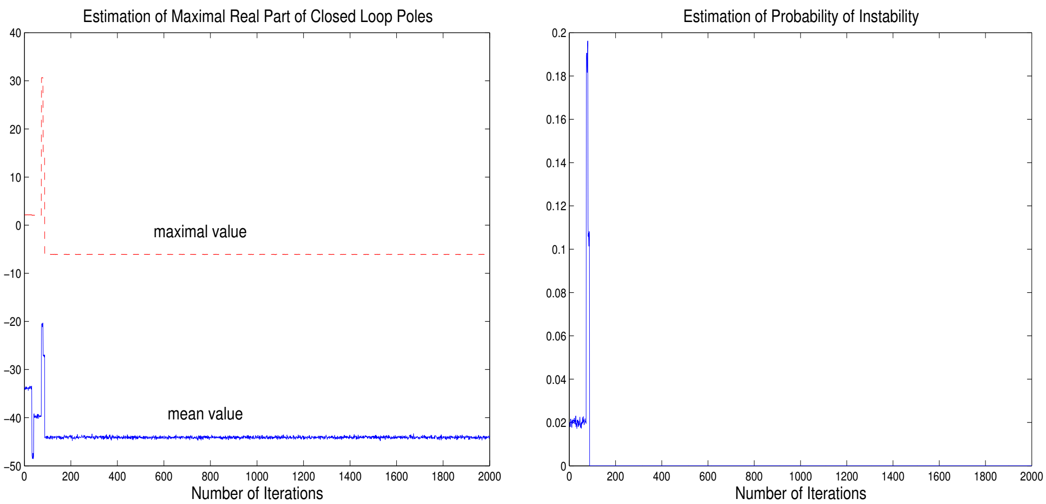
7.6 Concluding Remarks

In this chapter, we provided a new algorithm for the design of robust linear static state feedback controllers. This algorithm circumvents the conservatism of classical approaches since it does not require the existence of a single Lyapunov function that "works" for all values of the



(a) Maximal Real Part of Poles (b) Risk of Instability

Fig. 7.3. MC Simulations: Our Algorithm



(a) Maximal Real Part of Poles (b) Risk of Instability

Fig. 7.4. MC Simulations: Calafiore-Polyak's Step size

uncertainty. It also does not require a parametrization of the so-called *Lyapunov* matrix. The usefulness of the approach presented in this chapter is illustrated via two numerical examples, one of which is known not to be quadratically stabilizable and, hence, one cannot apply commonly available tools to design a static state feedback controller.

Chapter 8

Conclusion and Directions for Further Research

In this chapter, we provide some final thoughts on the results presented and, based on that, suggest some future research directions.

8.1 Concluding Remarks

Convexity plays an important role in this thesis and divides it into two parts:

In the first part, the controller synthesis problem can be formulated as an optimization problem which is convex in the design parameters. In Chapter 2, we addressed the probabilistically constrained linear program (PCLP), the counterpart of the classical linear program. We proved that for log-concave and symmetric distributions, PCLP is convex. Moreover, it is shown that PCLP is a convex quadratic optimization problem for elliptical distributions. Chapter 3 concentrates on a probabilistic version of the well known robust quadratic stabilization problem which is approximated by a log-concave function of the state feedback gain. This allows for the use of stochastic approximation algorithms to solve for the optima feedback gain; i.e., the gain that maximizes the probability of quadratic stability. Controller design for linear parameter varying (LPV) systems is studied in Chapter 4. A probabilistic version of receding horizon design was proposed which is essentially a stochastic optimization problem with probabilistic constraints.

Many control problems are not convex in the controller. In the second part, we addressed some of these non-convex design problems. The main idea is that, given a sample of the uncertainty, we can exploit an appropriate convex parameterization of all the achievable closed loop systems (valid only for that specific value of the uncertainty) to develop a stochastic approximation algorithm which converges to the desired controller. The focal point of Chapter 5 was the robust output feedback controller design for linear time-invariant uncertain systems. Given achievable bounds on performance (defined by a convex performance indicator), the proposed algorithm converges to an output feedback controller that robustly satisfies the specifications. For a given uncertainty sample $\triangle$, through *Youla* parametrization, there is a unique *Youla* parameter Q corresponding to a given controller C while the closed loop transfer function can be expressed as an affine function of Q. In Chapter 6, we directly worked on the optimal design for probabilistic output feedback design. We proved that the expectation of the convex performance function can be minimized. In this chapter, *Youla* parametrization is still employed to set up the one-to-one relationship between controller C and intermediate variable Q. In Chapter 7, a probabilistic robust stability design was discussed. A parameter dependent *Lyapunov* equation was used as the bridge between the state feedback gain K and the intermediate parameters.

As a whole, this thesis deals with robust controller design problems within a probabilistic framework. Our goal was to relieve the conservatism and prohibitive computational cost which are inherent to classical robust control theory. Except for Chapter 2, which converts a probabilistic optimization problem into a deterministic one, we worked with stochastic optimization algorithms (more specifically, stochastic approximation) directly. Some novel features of our algorithms are:

1. In the second part, i. e., in Chapter 5, Chapter 6 and Chapter 7, we addressed some of the non-convex controller design problems by parametrization.
2. In Chapter 5 and Chapter 6, the probabilistic output feedback design was accomplished in the infinite-dimensional space.
3. In Chapter 4, the optimization problem with both probabilistic objective and constraints was solved by stochastic min-max algorithms. In Chapter 2, with the aid of the concept *floating body*, the probabilistically constrained linear program was tackled by converting it to a deterministic convex quadratic optimization problem.

8.2 Directions for Further Research

The results to date suggest several directions for further research.

8.2.1 Numerical Tools for PCLP

The numerical implementation is a very important aspect of PCLP. In our work, the construction of the floating body is limited to the case of elliptical distributions. How to characterize the floating body for non elliptical probability distributions is still an open problem. We believe that effort should be put in the development of numerical tools for solving the PCLP when the distribution is other than elliptical. Also, it seems that the "ratio" between the complexity of a robust controller and the complexity of the probabilistic robust controller increases with the dimension of the uncertainty vector. Therefore, it would be of interest to quantify how does complexity depend on the uncertainty dimension.

8.2.2 Probabilistic Design for general LMIs

In Chapters 3 and 7, instead of designing a "traditional robust" controller, we provide algorithms which maximize the probability of quadratic stability or robust stability. Examples provided show that we can greatly reduce the gains of the controller and still have a very small risk of instability. The work presented here is just a first step toward probabilistic controller design. Possible directions for future work involve the development of algorithms for general Linear Matrix Inequalities, not just for the the particular case of quadratic stability or robust stability.

8.2.3 More Applicable Stochastic Subgradient Algorithms

We address the problem of robust controller design for linear time invariant systems with arbitrary uncertainty structure in Chapters 5, 6 and 7 where algorithms are provided which are proven to converge to the optimal solution with probability one. However, these kind of convergence results are only of an asymptotic nature. It would be very desirable to design an iterative process which stops after a finite number of steps within a neighborhood of the optimal solution, where the size of the neighborhood can be predetermined. Also of interest is the problem of order of the controller. Since there are no restrictions on the order of the controllers designed using some of the algorithm presented in this report, it would be of interest to modify it so that it would take the maximum order of the controller as one of the specifications. Furthermore, the procedures presented do not assure that, at each iteration, one has a controller that robustly stabilizes the system. In many cases, this is a "hard" constraint in the sense that the final design should lead to a robustly stable system. Hence, we believe that effort should be put in the study of this problem.

Appendix A

Proof of Lemma 2.3.1

The proof is identical to the one presented in [36] and it is presented here for the sake of completeness. For a given $0 \leq \varepsilon \leq 1/2$, note that proving the convexity of the set

$$\mathcal{X}_\varepsilon \doteq \{x \in \mathbf{R}^\ell : \mathrm{Prob}\{x^T(a_0 + \Delta a) \leq b\} \geq 1 - \varepsilon\}$$

is equivalent to proving the quasi-concavity of the function

$$\varphi(x) \doteq \mathrm{Prob}\{x^T(a_0 + \Delta a) \leq b\}$$

on the set

$$\mathcal{D} \doteq \{x \in \mathbf{R}^\ell : \mathrm{Prob}\{x^T(a_0 + \Delta a) \leq b\} \geq 1/2\}.$$

Hence given $x^0, x^1 \in \mathcal{D}$, we must prove that

$$\varphi((1-\lambda)x^0 + \lambda x^1) \geq \min\{\varphi(x^0), \varphi(x^1)\}$$

for all $0 \leq \lambda \leq 1$. Notice that the definition above only makes sense if the set $\mathcal{D}$ is convex. Proceeding by contradiction and assume that the set $\mathcal{D}$ is not convex. Given the fact that $\varphi(x)$ is continuous, non convexity of $\mathcal{D}$ implies the existence of $x^0, x^1 \in \mathbf{R}^\ell$ and $0 < \lambda < 1$ such that $\varphi(x^0) = \varphi(x^1) = 1/2$ and $\varphi((1-\lambda)x^0 + \lambda x^1) < 1/2$. Now, defining

$$Q_{good}(x) = \{\Delta a \in \mathbf{R}^\ell : x^T(a_0 + \Delta a) \leq b\},$$

the symmetry of the distribution of Δa and the assumptions on x^0, x^1 and λ imply that

$$0 \in Q_{good}(x^0) \cap Q_{good}(x^1);\ 0 \notin Q_{good}((1-\lambda)x^0 + \lambda x^1).$$

However, it can be easily shown that

$$Q_{good}(x^0) \cap Q_{good}(x^1) \subseteq Q_{good}((1-\lambda)x^0 + \lambda x^1).$$

This contradicts $0 \notin Q_{good}((1-\lambda)x^0 + \lambda x^1)$. Therefore, the set $\mathcal{D}$ is convex. We now proceed with the proof of quasi-concavity of $\varphi(x)$. Proceeding by contradiction, assume there exist $x^0, x^1 \in \mathcal{D}$ and $0 < \lambda < 1$ such that

$$\varphi((1-\lambda)x^0 + \lambda x^1) < \min\{\varphi(x^0), \varphi(x^1)\}.$$

Without loss of generality, we assume that $\varphi(x^0) \leq \varphi(x^1)$ and recall that $\varphi(x)$ is a continuous function of x. Therefore, there exists a $\lambda < \lambda^* \leq 1$ such that

$$\varphi((1-\lambda^*)x^0 + \lambda^* x^1) = \varphi(x^0).$$

Note that λ^* is strictly greater than λ since we assumed that $\varphi((1-\lambda)x^0 + \lambda x^1) < \varphi(x^0)$. Letting $y^0 = x^0$, $y^1 = (1-\lambda^*)x^0 + \lambda^* x^1$ and $\zeta = \lambda/\lambda^*$, we obtain

$$(1-\lambda)x^0 + \lambda x^1 = (1-\zeta)y^0 + \zeta y^1.$$

Hence, we have

$$\varphi((1-\zeta)y^0 + \zeta y^1) < \varphi(y^0) = \varphi(y^1).$$

Now, define $y^\zeta \doteq (1-\zeta)y^0 + \zeta y^1$. Then

$$\text{Prob}(Q_{good}(y^\zeta)) < \text{Prob}(Q_{good}(y^0)) = \text{Prob}(Q_{good}(y^1)).$$

Let $\gamma = 1 - \text{Prob}(Q_{good}(y^0))$. Since $y^0 \in \mathcal{D}$, then $0 \leq \gamma \leq 1/2$. To establish quasi-concavity of $\varphi(x)$ for $x \in \mathcal{D}$ we consider several cases. In the case of $\gamma = 0$ or $\gamma = 1/2$, a contradiction is reached since the robust linear program (risk $\gamma = 0$) is a convex program and the set $\mathcal{D}$ is convex. For the intermediate case when $0 < \gamma < 1/2$, since Δa has a log-concave symmetric distribution, Proposition 2 in [48] indicate that for this range of values of γ, the floating body K_γ exists and is a convex symmetric set. Therefore,

$$K_\gamma \subseteq Q_{good}(y^0) \cap Q_{good}(y^1).$$

Now, given that $Q_{good}(y^0) \cap Q_{good}(y^1) \subseteq Q_{good}(y^\zeta)$, we have $K_\gamma \subseteq Q_{good}(y^\zeta)$. Recall that

$$\text{Prob}(Q_{good}(y^\zeta)) < 1 - \gamma \Rightarrow \text{Prob}(Q^c_{good}(y^\zeta)) > \gamma.$$

However, given the definition of K_γ, we have

$$K_\gamma \cap Q^c_{good}(y^\zeta) \neq \emptyset$$

and we reach a contradiction. Since we reached a contradiction in all of the cases above, we conclude that the function $\varphi(x)$ is quasi-concave for all $x \in \mathcal{D}$.

Appendix B

Appendix: Computing Gradient of $g(\lambda_{max}(L_K(q)))$

In this section we describe how to compute gradient of $g(\lambda_{max}(L_K(q)))$.

$$\frac{\partial g(\lambda_{max}(L_K(q)))}{\partial K}$$
$$= \begin{cases} 0 & \text{if } z \leq \delta \\ -\beta e^{-\beta z} \frac{\partial z(K,q)}{\partial K} & \text{if } z > \delta, \end{cases}$$

where small value $\delta > 0$ is a user determined small value and $z = \lambda_{max}(L_K(q))$. Note that, given $\tilde{K}$,

$$\lambda_{\max}(L_{\tilde{K}}) = \max_{\|y\|_2=1} y^T L_{\tilde{K}} y = y^{*T} L_{\tilde{K}} y^*$$

where y^* is an eigenvector of euclidean norm one associated with maximum eigenvalue of $L_K(\tilde{\mathbf{x}})$. Given that the maximum above is achieved by y^* then

$$\left.\frac{\partial \lambda_{\max}(L_K)}{\partial K}\right|_{K=\tilde{K}} = \left.\frac{\partial y^{*T} L_K y^*}{\partial K}\right|_{K=\tilde{K}}.$$

This can be easily computed since, in the case addressed in (3.2), L_K is an affine function of each entry in K.

Appendix C

Proof of Lemma 3.6.1

We first provide the definition of stochastic quasi-Feyer sequence and results regarding this sequence [24].

DEFINITION C.0.1. *A sequence of random vectors $\{z_n\}_{n=0}^{\infty}$ defined on a probability space $(q, \Re, \mu)$ is a stochastic quasi-Feyer sequence for a set $Z \subseteq R^n$, if $E\|z^0\|^2 < \infty$, and for any $z \in Z$*

$$E\{\|z - z^{n+1}\|^2 | z^0, \ldots, z^n\} \leq \|z - z^n\|^2 + d_n, \quad n = 0, 1 \ldots$$

$$d_n \geq 0, \; \sum_{n=0}^{\infty} Ed_n < \infty$$

LEMMA C.0.1. *If $\{z^n\}$ is a stochastic quasi-Feyer sequence for a set Z, then:*
1) the sequence $\|z - z^n\|^2$, $n = 0, 1, \ldots$ converges with probability 1 for any $z \in Z$, $E\|z - z^n\|^2 < C < \infty$;
2) the set of accumulation points of $z^n(q)$ is not empty for almost all q;
3) if $z'(q), z''(q)$ are two distinct accumulation points of the sequence $\{z^n(q)\}$ which do not belong to the set Z, then Z lies on the hyper-plane equidistant from the points $z'(q)$, and $z''(q)$.

We can now proceed to prove Lemma 3.6.1.
Proof of Lemma 3.6.1:
By Lemma 3.4.1, we know

$$G(K(n)) = \int g(\lambda_{max}(L_{K(n)}(q)))f(q)\, dq$$

is log-concave. So if we define auxiliary function

$$G^{aux}(K(n)) = -\log G(K(n)), \tag{C.1}$$

it is convex. Due to the monotonicity of logarithm function, maximizing $G(K)$ is equivalent to minimizing $G'(K)$. Let $K^* \in \mathcal{K}^*$ is one of the optimal points. Then we have

$$\begin{aligned}
& G^{aux}(K^*) - G^{aux}(K(n)) \\
\geq \quad & \langle G_K^{aux}(K(n)), K^* - K(n)\rangle \\
\geq \quad & -\langle \frac{1}{G(K(n))} G_K(K(n)), K^* - K(n)\rangle \\
\geq \quad & -\langle \frac{1}{G(K(n)} E\{\xi(n)|K(0),\ldots,K(n)\}, K^* - K(n)\rangle + \langle \frac{1}{G(K(n))} b(n), K^* - K(n)\rangle \\
\geq \quad & -\langle \frac{1}{G(K(n)} E\{\xi(n)|K(0),\ldots,K(n)\}, K^* - K(n)\rangle - \frac{1}{G(K(n))}\gamma(n),
\end{aligned}$$

Since $G^{aux}(K^*) - G^{aux}(K(n)) \leq 0$ and $G(K(n)) > 0$, we have

$$\gamma(n) \geq \; -\langle E\{\xi(n)|K(0),\ldots,K(n)\}, K^* - K(n)\rangle$$

By (3.9), we have

$$\begin{aligned}
& E\{\|K^* - K(n+1)\|^2 |K(0),\ldots,K(n)\} \\
\leq \quad & \|K^* - K(n)\|^2 + a(n)^2 E\{\|\xi(n)\|^2 |K(0),\ldots,K(n)\} - 2a(n)\langle E\{\xi(n)|K(0),\ldots,K(n)\}, K^* - K(n)\rangle \\
\leq \quad & \|K^* - K(n)\|^2 + a(n)^2 E\{\|\xi(n)\|^2 |K(0),\ldots,K(n)\} + 2a(n)|\gamma(n)|.
\end{aligned}$$

By Assumption (3) in Lemma 3.6.1,

$$\begin{aligned}
& \sum_{n=0}^{\infty} E[a(n)^2 E\{\|\xi(n)\|^2 |K(0),\ldots,K(n)\} + 2a(n)|\gamma(n)|] \\
= \quad & \sum_{n=0}^{\infty} E[a(n)^2 \|\xi(n)\|^2 + 2a(n)|\gamma(n)|] \\
< \quad & \infty.
\end{aligned}$$

Hence $\{K(n)\}$ is a stochastic quasi-Feyer sequence for set $\mathcal{K}^*$. Thus, the sequence $\|K^*-K(n)\|^2$ converges with probability 1 for any K^*. Furthermore,

$$\begin{aligned}
& E\{\|K^*-K(n+1)\|^2|K(0),\ldots,K(n)\} \\
\leq \;& \|K^*-K(n)\|^2+a(n)^2E\{\|\xi(n)\|^2|K(0),\ldots,K(n)\} \\
& -2a(n)\langle E\{\xi(n)|K(0),\ldots,K(n)\},K^*-K(n)\rangle
\end{aligned}$$

Take expectations on both sides, we have

$$\begin{aligned}
& E\{\|K^*-K(n+1)\|^2\} \\
\leq \;& E\{\|K^*-K(n)\|^2\}-2E\{a(n)\langle E\{\xi(n)|K(0),\ldots,K(n)\},K^*-K(n)\rangle\}+a(n)^2E\{\|\xi(n)\|^2\} \\
\leq \;& E\|K^*-K(0)\|^2-2E\sum_{i=0}^{n}a(i)\langle E\{\xi(i)|K(0),\ldots,K(i)\},K^*-K(i)\rangle+\sum_{i=0}^{n}a(i)^2E\{\|\xi(i)\|^2\} \\
\leq \;& E\|K^*-K(0)\|^2+2E\sum_{i=0}^{n}a(i)\langle G(K(i))G_K^{aux}(K(i))-b(i),K^*-K(i)\rangle+\sum_{i=0}^{n}a(i)^2E\{\|\xi(i)\|^2\} \\
\leq \;& E\|K^*-K(0)\|^2+2E\sum_{i=0}^{n}a(i)G(K(i))(G^{aux}(K^*)-G^{aux}(K(i)))+C\sum_{i=0}^{n}E\{a(i)|\gamma(n)|+a(i)^2\|\xi(i)\|^2\}
\end{aligned}$$

Thus,

$$E\sum_{i=0}^{\infty}a(i)G(K(i))(G^{aux}(K^*)-G^{aux}(K(n)))>-\infty$$

Since $\sum_{i=0}^{\infty}a(i)=\infty$ and $G(K(i))>=p$, $\forall\ i$, we have

$$\sum_{i=0}^{\infty}a(i)G(K(i))\ \geq\ p\sum_{i=0}^{\infty}a(i)\ =\ \infty.$$

Moreover,

$$G^{aux}(K^*)-G^{aux}(K(i))\leq 0.$$

Hence

$$G(K^*)-G(K(i))\rightarrow 0.$$

So we have

$$\lim_{n\rightarrow\infty}G(K(n))\ =\ G(K^*)$$

where $K^*\in\mathcal{K}^*$. Q.E.D.

Appendix D

Supermartingale Convergence Theorem

The following lemma as stated in [6] was used to prove the convergence of the algorithms provided in this thesis.

LEMMA D.0.1. *Let Y_k, Z_k and W_k, $k = 1,2,\ldots$, be three sequences of random variables and $\mathcal{F}_k$, $k = 1,2,\ldots$, be sets of random variables such that $\mathcal{F}_k \subset \mathcal{F}_{k+1}$ for all k. Suppose that:*

1. *The random variables Y_k, Z_k and W_k are nonnegative, and are functions of random variables in $\mathcal{F}_k$.*
2. *For each k, we have $\mathbf{E}\{Y_{k+1}|\mathcal{F}_k\} \leq Y_k - Z_k + W_k$.*
3. *There holds $\sum_{k=0}^{\infty} W_k < \infty$*

Then, we have $\sum_{k=0}^{\infty} Z_k < \infty$, and the sequences Y_k converges to a nonnegative random variable Y, with probability 1.

References

[1] B. R. Barmish and C. M. Lagoa. On convexity of the probabilistic design problem for quadratic stabilizability. In *Proceedings of the American Control Conference*, San Diego, CA, 1999.

[2] B. R. Barmish and B. T. Polyak. A new approach to open robustness problems based on probabilistic prediction formulae. In *Proceedings of the IFAC World Congress*, San Francisco, CA, 1996.

[3] B. R. Barmish and P. S. Shcherbakov. On avoiding vertexization of robustness problems: the approximate feasibility concept. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1031–1036, Sydney, Australia, 2000.

[4] R. H. Bartels and G. W. Stewart. Solution of the matrix equation ax+xb=c. *Communications of the ACM*, 15:820–826, 1972.

[5] I. Batina, A. A. Stoorvogel, and S. Weiland. Stochastic disturbance rejection in model predictive control by randomized algorithms. In *Proceedings of the American Control Conference*, pages 732–737, Arlington, VA, 2001.

[6] D. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scitific, Belmont, MA, 1996.

[7] P. Billingsley. *Probability and Measure*. John Wiley, New York, NY, 1979.

[8] F. Blanchini and M. Sznaier. A convex optimization approach to fixed-order controller design for disturbance rejection in siso systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1540–1545, San Diego, CA, 1997.

[9] E. Borel. *Probability and Life*. Presses Universitaire de France, 1943. Dover English translation from the original, *Les Probabilite et la Vie*.

[10] S. Boyd and C. Barratt. *Linear Controller Design - Limits of Performance*. Prentice Hall, 1991.

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

[12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[13] R. P. Braatz, P. M. Young, J. C. Doyle, and M. Morari. Computational complexity of μ calculation. *IEEE Transactions on Automatic Control*, 39:1000–1002, 1994.

[14] G. Calafiore, F. Dabbene, and R. Tempo. Uniform sample generation in l_p balls for probabilistic robustness analysis. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3335–3340, Tampa, FL, 1998.

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

[16] G. C. Calafiore and F. Dabbene. Randomization in $\mathcal{RH}_\infty$: An approach to control design with hard/soft performance specifications. In *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, 2002.

[17] G. C. Calafiore, F. Dabbene, and R. Tempo. Randomized algorithms for probabilistic robustness with real and complex structured uncertainty. *IEEE Transactions on Automatic Control*, 45:2218–2235, 2000.

[18] X. Chen and K. Zhou. A probabilistic approach to robust control. In *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, 1997.

[19] X. Chen and K. Zhou. Constrained optimal synthesis and robustness analysis by randomized algorithms. In *Proceedings of the American Control Conference*, Philadelphia, PA, 1998.

[20] X. Chen, K. Zhou, and J. L. Aravena. Fast universal algorithms for robust analysis. In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, 2003.

[21] M. Corless and G. Leitmann. Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. *IEEE Transactions on Automatic Control*, 26:1139–1144, 1981.

[22] C. Van de Panne and W. Popp. Minimum-cost cattle feed under probabilistic protein constraints. *Management Science*, pages 405–430, 1963.

[23] L. Devroye. Random variate generation for multivariate unimodal densities. *ACM Transactions on Modeling Computer Simulation*, 7:447–477, 1997.

[24] Yu. Ermoliev. *Numerical Techniques for Stochastic Optimization*, chapter Stochastic Quasigradient Methods, pages 141–185. Springer-Verlag, 1988.

[25] Y. Fujisaki, F. Dabbene, and R. Tempo. Probabilistic robust design of lpv control systems. *Automatica*, 39:1323–1337, 2001.

[26] A. M. Gupal. One stochastic programming problem with constraints of a probabilistic nature. *Kibernetika (Kiev)*, 6:94–100, 1974.

[27] S. Gutman. Uncertain dynamic systems – a lyapunov min-max approach. *IEEE Transactions on Automatic Control*, 24:437–443, 1979.

[28] A. Jadbabaie, J. Yu, and J. Hauser. Receding horizon control of the caltech ducted fan: a control lyapunov function approach. In *Proceedings of the IEEE International Conference on Control Applications*, pages 51–56, Arlington, VA, 1999.

[29] L. H. Keel and S. P. Bhattacharyya. A linear programming approach to controller design. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2139–2148, San Diego, CA, 1997.

[30] L. H. Keel and S. P. Bhattacharyya. Robust, fragile or optimal. *IEEE Transactions on Automatic Control*, 42:1098–1105, 1997.

[31] P. Khargonekar and A. Tikku. Randomized algorithms for robust control analysis and synthesis have polynomial complexity. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3470–3475, Kobe, Japan, 1996.

[32] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.*, 23:462–466, 1952.

[33] K. H. Kienitz, Q. H. Wu, and M. Mansour. Robust stabilisation of a helicopter model. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2607–2612, Hawaii, 1990.

[34] H. J. Kushner and E. Sanvicente. Stochastic approximation of constrained systems with system and constraint noise. *Automatica*, 11:375–380, 1975.

[35] C. M. Lagoa. A convex parametrization of risk-adjusted stabilizing controllers. In *Proceedings of the IEEE Conference on Decision and Control*, pages 534–539, Phoenix, AZ, 1999.

[36] C. M. Lagoa. On the convexity of probabilistically constrained linear programs. In *Proceedings of the IEEE Conference on Decision and Control*, pages 516–521, Phoenix, AZ, 1999.

[37] C. M. Lagoa, X. Li, C. M. Mazzaro, and M. Sznaier. Sampling random transfer functions. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2429–2434, Maui, HI, 2003.

[38] C. M. Lagoa, X. Li, and M. Sznaier. Application of probabilistically constrained linear program to risk-adjusted controller design. In *Proceedings of the American Control Conference*, pages 738–743, Arlington, VA, 2001. Accepted by SIAM Journal on Optimization.

[39] C. M. Lagoa, X. Li, and M. Sznaier. On the design of robust controllers for arbitrary uncertainty structures. *IEEE Transactions on Automatic Control*, 48:2061–2065, 2003.

[40] C. M. Lagoa, M. Sznaier, and B. R. Barmish. An algorithm for generating transfer functions uniformly distributed over h_∞ balls. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5038–5043, Orlando, FL, 2001.

[41] G. Leitmann. Guaranteed asymptotic stability for some linear systems with bounded uncertainties. *Journal of Dynamic Systems, Measurement and Control*, 101:212–216, 1979.

[42] Xiang Li and C. M. Lagoa. Stochastic approximation algorithms for risk adjusted quadratic stability. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2254–2259, Las Vegas, NV, 2002.

[43] M. Loeve. *Probability Theory*. Van Nostrand, Princeton, NJ, 1963.

[44] C. I. Marrison and R. F. Stengel. Stochastic robustness synthesis applied to a benchmark control problem. *International Journal of Robust and Nonlinear Control*, 5:13–31, 1995.

[45] C. I. Marrison and R. F. Stengel. Robust control system design using random search and genetic algorithms. *IEEE Transactions on Automatic Control*, 42:835–839, 1997.

[46] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[47] R. S. Sánchez-Peña and M. Sznaier. *Robust Systems: Theory and Applications*. John Wiley, New York, NY, 1991.

[48] M. Meyer and S. Reisner. *Characterizations of affinely-rotation-invariant log-concave measures by section-centroid location*, pages 145–152. Springer, 1991.

[49] Y. Oishi. Probabilistic design of robust state-feedback controller based on parameter-dependent lyapunov functions. In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, 2003.

[50] Y. Oishi and H. Kimura. Computational complexity of randomized algorithms for solving parameter-dependent linear matrix inequalities. *Automatica*, 39:2149–2156, 2003.

[51] B. Polyak. *Introduction to Optimization*. Optimization Software, Inc, New York, NY, 1987.

[52] B. T. Polyak and M. E. Halpern. The use of a new optimization criterion for discrete-time feedback controller design. In *Proceedings of the Conference on Decision and Control*, pages 1540–1545, Phoenix, AZ, 1999.

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

[54] A. Prekopa. On logarithmic concave measures and functions. *Acta Scientiarum*, 34:335–343, 1973.

[55] L. R. Ray and R. F. Stengel. A monte carlo approach to the analysis of control system robustness. *Automatica*, 29:229–236, 1993.

[56] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22:400–407, 1951.

[57] J. S. Shamma and D. Xiong. Control of rate constrained linear parameter varying systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2525–2520, New Orleans, LA, 1995.

[58] R. F. Stengel and L. R. Ray. Stochastic robustness of linear time-invariant control systems. *IEEE Transactions on Automatic Control*, 36:82–87, 1991.

[59] M. Sznaier. Receding horizon: An easy way to improve performance in lpv systems. In *Proceedings of the American Control Conference*, pages 2257–2261, San Diego, CA, 1999.

[60] M. Sznaier and R. Suarez. Control of constrained systems using receding horizon control lyapunov functions. In *Proceedings of the Workshop on Systems with Time–Domain Constraints*, Eindoven, the Netherlands, 2000.

[61] Mario Sznaier, C. M. Lagoa, A. Stoorvogel, and X. Li. Risk adjusted receding horizon control of constrained linear parameter varying systems. Technical report, the Pennsylvania State University, Electrical Engineering Department, 2003. Accepted by 2004 IFAC Symposium of Nonlinear Control Systems (NOLCOS 2004).

[62] R. Tempo, E. W. Bai, and F. Dabbene. Probabilistic robustness analysis: Explicit bounds for the minimum number of samples. *Systems and Control Letters*, 30:237–242, 1997.

[63] S. Vajda. *Probabilistic Programming*. Academic Press, New York, NY, 1972.

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

[65] M. Vidyasagar. *Learning and Generalization: With Applications to Neural Networks*. Springer-Verlag, New York NY, 2003.

[66] K. Wang and A. N. Michel. On sufficient conditions for stability of interval matrices'. *Systems and Control Letter*, 20:345–351, 1993.

[67] Q. G. Wang. Necessary and sufficient conditions for stability of a matrix polytope with normal vertex matrices'. *Automatica*, 27:887–888, 1991.

[68] D. Williams. *Probability with Martingales*. Cambridge University Press, 1991.

[69] A. Yoon and P. Khargonekar. Computational experiments using randomized algorithms for robust stability analysis. In *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, 1997.

[70] A. Yoon, P. Khargonekar, and K. Hebbale. Design of computational experiments for open-loop control and robustness analysis of clutch–to–clutch shifts in automatic transmissions. In *Proceedings of the American Control Conference*, Albuquerque, NM, 1997.

[71] X. Zhu, Y. Huang, and J. Doyle. Soft vs. hard bounds in probabilistic robustness analysis. In *Proceedings of the IEEE Conference on Decision and Control*, Kobe, Japan, 1996.

Vita

Xiang Li was born in Taizhou, China on December 21, 1973. He received the B.Eng., M.Eng both in Automatic Control from Nanjing University of Science and Technology in 1996, 1999 respectively. He then came to the Pennsylvania State University to continue his Ph.D study under the guidance of Professor Constantino Lagoa. He is a student member of IEEE.

Publications

1. "On the Design of Robust Controllers for Arbitrary Uncertainty Structures," C. M. Lagoa, X. Li and M. Sznaier, Proceedings of 2003 American Control Conference; Also Appeared on IEEE Transactions on Automatic Control, Vol. 48, No. 11;

2. "Probabilistically Constrained Linear Programs and Risk-Adjusted Controller Design," C. M. Lagoa, X. Li and M. Sznaier, Proceedings of 2001 American Control Conference; Also Accepted by SIAM Journal on Optimization.

3. "Stochastic Approximation Algorithms for Risk-Adjusted Quadratic Stability," X. Li and C. M. Lagoa, Proceedings of 2002 IEEE Conference on Decision and Control.

4. "Sampling Random Transfer Functions," C. M. Lagoa, X. Li , C. Mazzaro and M. Sznaier, Proceedings of 2003 IEEE Conference on Decision and Control. Also to appear as a book chapter of " Probabilistic and Randomized Methods for Design under Uncertainty " Ed. by R. Tempo, G. Calafiore and F. Dabbene, Springer-Verlag, London.

5. " Risk-Adjusted Robust Controller Design by Stochastic Optimization," C. M. Lagoa, X. Li and M. Sznaier, Accepted by 2004 International Symposium on Mathematical Theory of Networks and Systems.

6. "Risk-Adjusted Receding Horizon Control of Constrained Linear Parameter Varying Systems," M. Sznaier, C. M. Lagoa, A. Stoorvogel and X. Li, Accepted by 2004 IFAC Symposium on Nonlinear Control Systems (NOLCOS).

7. "On the Design of Optimal Robust Controllers for Arbitrary Uncertainty Structures," X. Li, C. M. Lagoa, To be submitted for publication.

8. "Robust Stability Design for Arbitrary Uncertainty Structures by Stochastic Approximation," C. M. Lagoa, X. Li, To be submitted for publication.