

# Efficient Computational Methods for Robustness Analysis

by

Chung-Yao Kao

B.S., Mechanical Engineering, National Taiwan University (1994)  
M.S., Mechanical Engineering, National Taiwan University (1996)

Submitted to the Department of Mechanical Engineering  
in partial fulfillment of the requirements for the degree of

Doctor of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2002

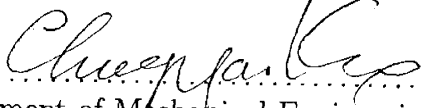
ARCHIVES

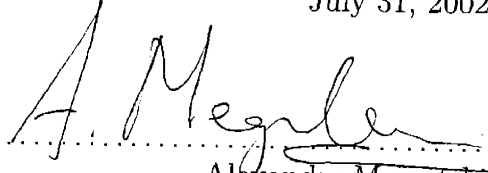
MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

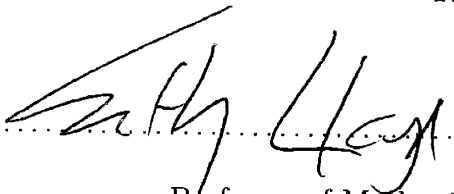
OCT 25 2002


LIBRARIES

© Massachusetts Institute of Technology 2002. All rights reserved.

Author .....   
Department of Mechanical Engineering  
July 31, 2002

Certified by .....   
Alexandre Megretski  
Associate Professor of Electrical Engineering  
Thesis Supervisor

Read by .....   
Seth Lloyd  
Professor of Mechanical Engineering  
Thesis Reader

Accepted by .....   
Ain Sonin  
Chairman, Departmental Committee on Graduate Students

# Efficient Computational Methods for Robustness Analysis

by

Chung-Yao Kao

Submitted to the Department of Mechanical Engineering  
on July 31, 2002, in partial fulfillment of the  
requirements for the degree of  
Doctor of Science in Mechanical Engineering

## Abstract

Issues of robust stability and performance have dominated the field of systems and control theory because of their practical importance. The recently developed Integral Quadratic Constraint (IQC) based analysis method provides a framework for systematically checking robustness properties of large complex dynamical systems. In IQC analysis, the system to be analyzed is represented as a nominal, Linear Time-Invariant (LTI) subsystem interconnected with a perturbation term. The perturbation is characterized in terms of IQCs. The robustness condition is expressed as a feasibility problem which can be solved using interior point algorithms.

Although the systems to be analyzed have nominal LTI subsystems in many applications, this is not always the case. A typical example is the problem of robustness analysis of the oscillatory behavior of nonlinear systems, where the nominal subsystem is generally Linear Periodically Time-Varying (LPTV). The objective of the first part of this thesis is to develop new techniques for robustness analysis of LPTV systems. Two different approaches are proposed. In the first approach, the harmonic terms of the LPTV nominal model are extracted, and the system is transformed into the standard setup for robustness analysis. Robustness analysis is then performed on the transformed system based on the IQC analysis method. In the second approach, we allow the nominal system to remain periodic, and we extend the IQC analysis method to include the case where the nominal system is periodically time-varying. The robustness condition of this new approach is posed as semi-infinite convex feasibility problems which requires a new method to solve. A computational algorithm is developed for checking the robustness condition.

In the second part of the thesis, we consider the optimization problems arising from IQC analysis. The conventional way of solving these problems is to transform them into semi-definite programs which are then solved using interior-point algorithms. The disadvantage of this approach is that the transformation introduces additional decision variables. In many situations, these auxiliary decision variables become the main computational burden, and the conventional method then becomes very inefficient and time consuming. In the second part of the thesis, a number of specialized algorithms are developed to solve these problems in a more efficient fashion. The crucial advantage in this development is that it avoids the equivalent transformation. The results of numerical experiments confirm that these algorithms can solve a problem arising from IQC analysis much faster than the conventional approach does.

Thesis Supervisor: Alexandre Megretski

Title: Associate Professor of Electrical Engineering

For My Parents



## Acknowledgments

First and foremost, I wish to express my deepest gratitude and appreciation to my research supervisor and mentor Professor Alexandre Megretski, for his guidance, support, and encouragement, which have helped tremendously my doctoral study. There is no doubt in my mind that Alex is one of the best researchers in my field of study and the best research advisor one can possibly hope to have. I feel very fortunate to be one of his students.

I want to thank Professor Seth Lloyd and Professor Eric Feron for their input to this work while serving as members of my thesis committee. Their valuable suggestions have helped to make this a better thesis. I would also like to express my gratitude to Professor George Verghese, Dr. Anuradha Annaswamy, Dr. Shen Liu, and Dr. Marija Ilić for attending my thesis presentation and for all the comments and questions regarding this work.

My deepest appreciation also goes to my long-time colleague Dr. Ulf Jönsson of KTH. I first met Ulf when he came to Laboratory for Information and Decision Systems (LIDS) as a postdoctoral associate in 1997. We have worked and published papers together since then. I appreciate the countless discussions with him, face to face or through e-mails, and his valuable input to my research work over the years.

For creating such an enjoyable and intellectually stimulating environment to work in, I thank my colleagues at LIDS in general and some of the formal and current members of Control System Group in particular: Prof. Dahleh, Prof. Mitter, Soosan Beheshti, Nicola Elia, Emilio Frazzoli, Jorge Gonçalves, John Harper, Neal Jameson, Fadi Karamneh, Georgios Kotsalis, Nuno Martins, Reza Olfati-Saber, Muralidhar Ravuri, Keith Santarelli, Sridevi Sarma, Danielle Tarraf, Saligrama Venkatesh, and Sean Warnick. I enjoyed very much our interaction and the way we work together over the years. I am indebted to Danielle, Keith, and Neal for their help on editing this thesis. Without their help, the readability of this thesis would be far more worse.

I would also like to acknowledge the assistance from the administrative staff at LIDS and the Graduate Office of Department of Mechanical Engineering, especially Doris Inslee, Fifa Monserrate, and Leslie Regan. Joan Kravit and Lynne Dell have also been very helpful. Thank you very much.

On a more personal note, I want to thank all my Taiwanese friends in the greater Boston area, especially those from the Republic of China Student Association (ROCSA) at MIT.

My life at MIT would have been hardly sustainable without them around. In the past five years, they added a colorful and pleasant social dimension of my MIT experience. There are too many of them for me to list all their names here. Among them, I especially want to acknowledge Wen-Hua Kuo, Wen-Tang Kuo and his wife Yu-Ru Liu, Ching-Yu Lin and his wife Chun-Ling Chou for their wonderful companionship through my years at MIT.

Last but not the least, I wish to express my most sincere appreciation to my family and relatives in Taiwan and around the world. I could not have accomplished any of this without their unconditional and endless support. And to *you* : *you* will always be the most important person in my life, and I will forever be indebted to *you*. This thesis is dedicated to those who would never cease caring about me.

The research in this thesis was supported in part by the National Science Foundation (NSF) under Grants ECS-9796033 and ECS-9796099, in part by the Air Force Office of Scientific Research (AFOSR) under Grant F49620-00-1-0096, and in part by the Defense Advanced Research Projects Agency (DARPA).

# Contents

<b>1</b>	<b>Introduction</b>	<b>17</b>
1.1	Analysis of Uncertain Periodic Systems . . . . .	19
1.2	Specialized Fast Solvers for Standard IQC Problems . . . . .	20
1.3	Contributions of This Thesis . . . . .	23
1.4	Thesis Organization . . . . .	25
<b>2</b>	<b>Mathematical Preliminaries</b>	<b>27</b>
2.1	Notations and Standard Concepts . . . . .	27
2.1.1	Normed Vector Spaces . . . . .	28
2.1.2	Inner Product Spaces . . . . .	29
2.1.3	Operators and Induced Norms . . . . .	30
2.1.4	Adjoint Operators and Quadratic Forms . . . . .	32
2.2	Linear Matrix Inequalities and Semi-Definite Programs . . . . .	32
2.3	S-procedure . . . . .	33
<b>3</b>	<b>Robustness Analysis of Periodic Systems</b>	<b>35</b>
3.1	Introduction . . . . .	35
3.2	Systems Under Consideration and Problem Formulation . . . . .	37
3.3	Fourier Series Expansion Method . . . . .	39
3.4	Periodic IQC Approach . . . . .	43
3.4.1	The State-space Realization of the Multiplier $\Pi$ . . . . .	46
3.4.2	Parameterizing and Optimizing IQCs . . . . .	46
3.5	Parameterizations of Some IQCs . . . . .	50
3.5.1	Periodically Varying Parameter . . . . .	50
3.5.2	Time-varying uncertainty . . . . .	50

3.5.3	Uncertain parameter . . . . .	51
3.5.4	Dynamic Uncertainty . . . . .	54
3.6	Summary . . . . .	55
<b>4</b>	<b>Analysis of Periodically Forced Uncertain Feedback Systems</b>	<b>57</b>
4.1	Introduction . . . . .	57
4.2	Systems Under Consideration . . . . .	59
4.3	Existence and Uniqueness of Stationary Solutions . . . . .	60
4.4	Stability of the Stationary Solution . . . . .	61
4.5	Harmonic Performance Analysis . . . . .	63
4.5.1	Supremum Norm of Periodic Output . . . . .	65
4.6	Parameterizing and Optimizing IQCs . . . . .	67
4.7	Summary . . . . .	68
<b>5</b>	<b>Cutting Plane Algorithms for Analysis of Periodic Systems</b>	<b>69</b>
5.1	Introduction . . . . .	69
5.2	A Kelley Type Cutting Plane Algorithm . . . . .	71
5.2.1	Kelley's Cutting Plane Algorithm . . . . .	71
5.2.2	The Equivalent Eigenvalue Maximization Problem . . . . .	73
5.2.3	A Kelley Type Cutting Plane Algorithm . . . . .	76
5.3	The Oracle for Problems Arising from Chapter 3 . . . . .	79
5.3.1	Verification of Linear Matrix Inequalities . . . . .	80
5.3.2	Verification of the Integral Inequality . . . . .	80
5.3.3	Some Issues Regarding Numerical Computation . . . . .	85
5.4	The Oracle for Problems Arising from Chapter 4 . . . . .	87
5.4.1	A Frequency Theorem . . . . .	90
5.4.2	Generate Separating Hyperplanes . . . . .	91
5.5	Examples . . . . .	93
5.6	Proof of Theorem 5.3 . . . . .	108
5.7	Summary . . . . .	114
<b>6</b>	<b>Cutting Plane Algorithms for Standard IQC Problems</b>	<b>117</b>
6.1	Introduction . . . . .	117



6.2	A Motivating Example . . . . .	122
6.3	The Oracle for the Standard IQC Optimization Problem . . . . .	126
6.3.1	Generate Separating Hyperplanes . . . . .	128
6.4	The Results of Solving Problem (6.9) Using the Kelley Type Cutting Plane Algorithm . . . . .	128
6.5	The Ellipsoid Algorithm . . . . .	131
6.5.1	The Basic Algorithm . . . . .	131
6.5.2	The Ellipsoid Algorithm for Solving IQC Optimization Problems . . . . .	136
6.6	Analytical Center Cutting Plane Method . . . . .	138
6.6.1	The Algorithm . . . . .	139
6.6.2	The ACCPM for Solving IQC Optimization Problems . . . . .	143
6.7	Examples . . . . .	147
6.8	Comparison with the Conventional Method . . . . .	158
6.9	Summary . . . . .	160
<b>7</b>	<b>Interior Path-following Algorithms for Standard IQC Problems</b>	<b>163</b>
7.1	Introduction . . . . .	163
7.2	Notations and Problem Formulation . . . . .	165
7.3	The Interior Path-Following Algorithm . . . . .	167
7.3.1	The Barrier Function and the Central Path . . . . .	167
7.3.2	The Path-following Algorithm . . . . .	168
7.3.3	Feasibility and the Phase-I Method . . . . .	170
7.3.4	Convergence and Complexity Analysis . . . . .	171
7.4	New Barrier Functions for IQC Optimization Problems . . . . .	176
7.4.1	The First Barrier Function . . . . .	177
7.4.2	The Second Barrier Function . . . . .	183
7.5	Path-following Algorithms for IQC Optimization Problems . . . . .	187
7.6	Comparison with the Conventional Method . . . . .	189
7.7	Examples . . . . .	190
7.8	Proof of Theorem 7.5 . . . . .	196
7.9	Summary . . . . .	202

<b>8</b>	<b>Conclusions and Future Research</b>	<b>205</b>
8.1	Suggested Future Research . . . . .	206
	<b>Bibliography</b>	<b>209</b>

# List of Figures

1-1	Standard block diagram for robustness analysis. . . . .	18
3-1	Systems for robust stability and performance analysis. . . . .	37
3-2	Illustration of the Fourier Series Expansion Method. Figure on the left-hand-side: the original system has a LPTV nominal subsystem $G$ and a perturbation term $\Delta$ . Figure on the right-hand-side: the transformed system has a LTI nominal subsystem $\tilde{G}$ . The time-varying part of system $G$ is extracted and lumped into the feedback loop. Let $x$ denotes the states of $\tilde{G}$ . Then signal $v_1$ is equal to $[x' \ x' \ x']'$ , while signal $v_2$ is equal to $[w' \ w' \ w']'$ . . . . .	41
4-1	Systems for robust stability and performance analysis. . . . .	59
5-1	The idea of the computational algorithm is to generate a sequence of piece-wise linear functions that approximate $q(\lambda)$ around its maxima. Figure on the left-hand-side: the test point satisfies $S_\lambda > \gamma I$ , and an improved lower bound of $y_{opt}$ is obtained. Figure on the right-hand-side: the test point does not satisfy $S_\lambda > \gamma I$ , and a hyperplane is returned by the oracle to improve the piece-wise linear approximation of $q(\lambda)$ . Eventually, the maximum values of these piece-wise linear functions converge to $y_{opt}$ . . . . .	77
5-2	Setup for applying Fourier Series Expansion Method to estimate $L_2$ -gain of the system in Example 5.1. . . . .	94
5-3	Setup for robust stability analysis of the uncertain coupled Mathieu equations (5.40) using Fourier Series Expansion Method. Here $c(t) = \cos(\omega_F t)$ . . . . .	99
5-4	Stability region for the uncertain coupled Mathieu equations. . . . .	101
5-5	An inverted pendulum on a vertically vibrating platform. . . . .	102
5-6	Stability region for the inverted pendulum system. . . . .	105

5-7	Setup for robustness analysis of periodically forced uncertain coupled Mathieu equations. . . . .	106
6-1	Standard block diagram for robustness analysis. . . . .	123
6-2	The figure shows the amount of time that the MATLAB LMI Control Toolbox took to solve SDP (6.10). For each $n_s$ , five testing problems were randomly generated. As we can see, the amount of time that MATLAB LMI Control Toolbox took to solve a problem increases almost exponentially as $n_s$ grows. It took approximately two hours for the LMI Control Toolbox to solve a problem with 70 states. . . . .	125
6-3	The figure shows the amount of time that the Kelley type cutting plane algorithm took to solve problem (6.9). For each $n_s$ , five testing problems were randomly generated. Compared with Figure 6-2, we see that the amount of time that the Kelley type cutting plane algorithm took to solve a problem is less than the conventional method. Furthermore, as the number of states in a problem increases, the difference in speed becomes more significant. For $n_s = 70$ , the cutting plane algorithm took only a few minutes to solve a problem which the conventional method took approximately two hours to solve. . . . .	129
6-4	Illustration of central cuts and deep cuts. Let $\Omega$ be the feasible set and $\lambda_t$ be a trial point. The shaded regions are the parts of ellipsoids removed by the cutting planes. Figure on the left-hand-side: a central cut is a cutting hyperplane that passes through the test point $\lambda_t$ . Figure on the right-hand-side: a deep cut passes between the test point and the feasible set and cuts off a larger piece of the outbound set. . . . .	134
6-5	Illustration of the idea of using multiple cutting planes. Here, $\Omega$ denotes the feasible set. (1) Cutting planes $C_1$ and $C_2$ are placed. (2) Cutting plane $C_1$ is used to remove a part of the original ellipsoid and generate a new ellipsoid of smaller volume. (3) Cutting plane $C_2$ is then used to remove a part of the newly generated ellipsoid. . . . .	136

6-6	The figure shows the amount of time that the modified ellipsoid algorithm described in Section 6.5 took to solve problem (6.9). For each $n_s$ , five test problems were randomly generated. The algorithm starts with an initial ball of radius equal to $10^6$ . As we can see from the figure, all problems were solved in less than 10 minutes. Compared with Figure 6-2, we see that the ellipsoid algorithm is much faster than the conventional method when the problem to be solved has a large state space. . . . .	147
6-7	The figure shows the amount of time that the analytical center cutting plane method described in Section 6.6 took to solve problem (6.9). For each $n_s$ , five test problems were randomly generated. The algorithm started with an initial box constraint $\lambda_i \in [-10^6, 10^6]$ . The figure indicates that all problems were solved in less than 2 minutes. Compared with the Figure 6-2, we see that the ACCPM is significantly faster than the conventional method when the problem to be solved has a large state space. Furthermore, the ACCPM is no worse than the conventional method even when the problem has only a few states. . . . .	149
6-8	Seismic isolation control of a building. The building is modelled as a series connection of masses, springs, and dampers. . . . .	151
6-9	MATLAB SIMULINK diagram for design of seismic isolation controllers. . . . .	152
6-10	Setup for robustness analysis of the seismic control systems. . . . .	153
6-11	Schematic diagram of the lateral flight control system of the Space Shuttle. . . . .	155
6-12	Setup for robustness analysis of the Space Shuttle lateral axis flight control system. . . . .	157
7-1	This figure shows the amount of time that the interior path-following algorithm based on the first barrier function took to solve problem (6.9). For each $n_s$ , five test problems were randomly generated. As we can see from the figure, all problems were solved in less than 15 minutes. Compared with the Figure 7-3, we see that the proposed algorithm is significantly faster than the conventional method when the problem to be solved has a large state space. Furthermore, the proposed algorithm is no worse than the conventional method even when the problem has only a few states. . . . .	191

7-2	This figure shows the amount of time that the interior path-following algorithm based on the second barrier function took to solve problem (7-1). We see the performance of the interior path-following algorithm based on the second barrier function is very similar to the performance of the algorithm based on the first barrier. The amount of time they spent to solve a problem is at the same level. . . . .	192
7-3	This figure (same as Figure 6-2) shows the amount of time that the MATLAB LMI Control Toolbox took to solve problem (6.10). . . . .	193

# List of Tables

5.1	List of $\mu$ and estimated $L_2$ -gain. . . . .	97
6.1	Results of solving (6.9) using MATLAB LMI Control Toolbox, the ellipsoid algorithm, and the ACCPM. “LMI Tool“ denotes MATLAB LMI Control Toolbox, “ACCPM“ denotes the Analytical Center Cutting Plane Method, and “Ellipsoid“ denotes the ellipsoid algorithm. In this set of testing problems, the ACCPM starts with a initial box constraint $\{x :  x_i  \leq 10^9\}$ , and the ellipsoid algorithm starts with a ball constraints $\{x : \ x\  \leq 10^9\}$ . The objective value is accurate up to the $4^{th}$ digit. When $n_s = 200$ (the number of states in the problem is 200), the ACCPM and the ellipsoid algorithm can still solve the problem in a reasonable period of time. . . . .	150
6.2	Numbers of iterations that the ellipsoid algorithm (denoted by “Ellipsoid“) and the analytical center cutting plane method (denoted by “ACCPM“) took to solve (6.9). . . . .	150
6.3	Results of solving the $L_2$ -gain estimation problem in Example 6.2 using MATLAB LMI Control Toolbox, the ellipsoid algorithm, and the ACCPM. “LMI Tool“ denotes MATLAB LMI Control Toolbox. “ACCPM“ denotes the Analytical Center Cutting Plane Method. “Ellipsoid“ denotes the ellipsoid algorithm. The numbers in the column “var“ indicate the number of decision variables in a problem. In this set of testing problems, the ACCPM starts with an initial box constraint $\{x :  x_i  \leq 10^6\}$ , and the ellipsoid algorithm starts with a ball constraints $\{x : \ x\  \leq 10^6\}$ . . . . .	154

6.4	Results of solving the IQC optimization problem arising from the robustness analysis of Space Shuttle lateral flight control system using MATLAB LMI Control Toolbox, the ellipsoid algorithm, and the ACCPM. “LMI Tool“ denotes MATLAB LMI Control Toolbox. “ACCPM“ denotes the Analytical Center Cutting Plane Method. “Ellipsoid“ denotes the ellipsoid algorithm. In this set of testing problems, the ACCPM starts with a initial box constraint $\{x :  x_i  \leq 10^6\}$ , and the ellipsoid algorithm starts with a ball constraints $\{x : \ x\  \leq 10^6\}$ . . . . .	158
7.1	Numbers of iterations that the MATLAB LMI Control Toolbox (denoted by “LMI Tool“), the first interior path-following algorithm (denoted by “IPA-1“), and the second interior path-following algorithm (denoted by “IPA-2“) took to solve problem (6.9). . . . .	194
7.2	Results of solving the $L_2$ -gain estimation problem in Example 6.2 using MATLAB LMI Control Toolbox, the path-following algorithms. “LMI Tool“ denotes MATLAB LMI Control Toolbox. “IPA-1“ denotes the interior path-following algorithm based on the first barrier function. “IPA-2“ denotes the interior path-following algorithm based on the second barrier function. The numbers in the column “var“ indicate the number of decision variables in a problem. . . . .	196



# Chapter 1

## Introduction

Engineers typically base their work on mathematical models of actual processes. The mathematical models rarely describe the actual processes accurately, and they might even behave quite differently from the real process. The source of the mismatch between the model and the actual process could be: (1) statistical processes: consider for example, coefficients of material properties (such as Young's modulus, Poisson's ratio, conductivity, etc.). The values of these coefficients are usually set equal to the average obtained from empirical data. (2) Simplification of the model: Higher order dynamics of complex dynamical systems are usually ignored for the sake of simplicity. As a result, the model's behavior can be quite different from the true response of the real system when the system is subject to a high frequency input. (3) Linearization of complicated nonlinear dynamics: for example, the dynamics of robots and aircrafts are highly nonlinear, and controllers for these systems are usually designed based on one or more linearized models. The mismatch between the model and the true system can seriously deteriorate the system's stability and performance. For sensitive applications, such as control of aircrafts and high speed elevators, there is a strong need for rigorous analysis to ensure that these control systems behave in a tolerable range even in the presence of certain unaccounted factors.

Issues of robust stability and performance have dominated the field of systems and control theory because of their practical importance. A variety of approaches that treat these issues has been developed since the 1940s: some are associated with absolute stability theorems [50, 63, 83, 42], while other are based on the small gain and passivity theorems [91, 71, 70], dissipativity theory [78], multi-loop generalization of the circle criterion based

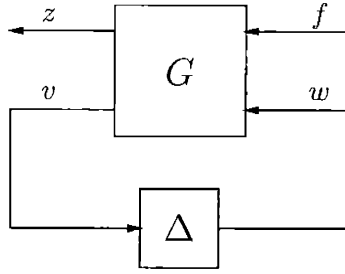


Figure 1-1: Standard block diagram for robustness analysis.

on D-scaling [69], structured singular value (or  $\mu$ -analysis) [17], etc.. All these analysis methods are, explicitly or implicitly, based on a concept called “Integral Quadratic Constraint“ (IQC). Analysis based on the concept of IQCs was first suggested in the 1960s by Yakubovich [84]. The IQC-based analysis approach was further developed in the early 1990s, when a minimal framework for the stability analysis of feedback interconnections was presented [52, 64, 65, 55, 36]. We refer to this development as *standard IQC analysis*, or *analysis in the standard IQC framework*.

In standard IQC analysis, the system to be analyzed is represented as a nominal system (denoted by  $G$ ) interconnected with a perturbation term (denoted by  $\Delta$ ). See Figure 1-1 for an illustration. The nominal subsystem  $G$  is usually assumed to be Linear Time-Invariant (LTI), while the perturbation term  $\Delta$  is not restricted to a particular class of system and represents the “trouble making“ (nonlinear, time-varying, or uncertain) components of the system. Signals  $f$  and  $z$  denote the external excitation and the signal representing the performance of the system, respectively. The perturbation  $\Delta$  is characterized by the relationship between its input signal  $v$  and output signal  $w$ . The relationship is described in terms of IQCs. The robustness condition is either expressed as a feasibility problem:

find  $\lambda_i$ , such that

$$\begin{bmatrix} M(j\omega) \\ I \end{bmatrix}^* \left( \Sigma_0 + \sum_{i=1}^n \lambda_i \Sigma_i \right) \begin{bmatrix} M(j\omega) \\ I \end{bmatrix} > 0 \quad \forall \omega \in [0, \infty], \quad (1.1)$$

or expressed an optimization problem:

$$\inf c' \lambda, \quad \text{subject to the constraint in (1.1),} \quad (1.2)$$

where  $M(s) := (sI - A)^{-1}B$  is a rational transfer matrix, and  $A, B, \Sigma_i = \Sigma_i'$  are given matrices. We refer to problems (1.1) and (1.2) as standard IQC problems.

## 1.1 Analysis of Uncertain Periodic Systems

While in many applications the system to be analyzed can be easily represented as a nominal LTI subsystem in a feedback interconnection with an uncertain operator, this is not always the case; there are problems where the natural feedback representation results in nominal subsystems which are not LTI. A typical example is the problem of robustness analysis of forced or unforced oscillations in a nonlinear system. Oscillatory behavior can be observed in many engineering systems, including delta-sigma modulators, automatic tuning of PID controllers, rotor motion of a magnetic bearing or a helicopter, or any machine which has a rotating part [19, 41]. To study the effect of perturbations on the oscillatory behavior of a nonlinear system, one can often linearize the perturbed system along the periodic trajectory of interest and perform analysis on the linearized system [39]. This approach results in an uncertain linear periodically time-varying system. Other engineering applications in which one would encounter LPTV systems include analysis of the horizontal plane motion of ships in waves [73], study of the stability of satellite orbits [74], and stability analysis of beams and plates subject to periodic excitation [18] (See also survey paper [8] for more examples, motivation, and an overview of the status of the field).

Stability analysis of LPTV systems is often performed based on Floquet theory (See [12, 21, 8]). However, this technique is essentially restricted to the case where the uncertainty in the LPTV model is a perturbation of trivial size. Uncertain LPTV systems can also be analyzed to some extent by using the Nyquist Theorem developed in [33] (See also [8] for a related discussion). Another approach would be to use lifting techniques to obtain an equivalent infinite-dimensional but time-invariant system in discrete time [6, 5, 20, 8]. This approach has been used mainly to obtain  $H_\infty$  and  $H_2$  optimal controllers for discrete time LPTV systems [13, 76], and for rigorous analysis of sampled-data systems [5, 20, 57]. It can also be used for analysis of general continuous time uncertain LPTV systems, but not much work has been presented on this topic, to the author's knowledge. Absolute stability results for periodic systems have been developed by Yakubovich [81], where LPTV systems interconnected with simple nonlinearities were considered.

The objective of the first part of this thesis (Chapters 3 to 5) is to develop new techniques for robustness analysis of LPTV systems. Two different approaches will be proposed. In the first approach, the harmonic terms of the nominal LPTV model are extracted, and the time-varying coefficients are lumped into the feedback loop via a simple loop transformation; the system to be analyzed is thus transformed into the setup suited for standard robustness analysis methods. Stability analysis is then performed on the transformed system based on standard IQC analysis, and the condition for robustness is formulated as an equivalent problem of checking the feasibility of a set of Linear Matrix Inequalities (LMIs).

In the second approach, we allow the nominal system to remain periodic and we extend the standard IQC analysis to include the case where the nominal system is periodically time-varying. We also show how *periodic IQCs* can be used in the analysis. The development is completely analogous to the time-invariant case, with the exception that the nominal system and the quadratic forms that define IQCs are allowed to be periodically time-varying. Furthermore, as opposed to LMIs in the linear time-invariant case, the robustness condition of this new approach is expressed as a semi-infinite convex optimization problem of the form

$$\inf c' \lambda, \quad \text{such that } S_\lambda > 0, \quad (1.3)$$

where  $S_\lambda$  is a self adjoint operator on a particular Hilbert space and depends affinely on the parameter vector  $\lambda$ . This type of problem can be solved using the cutting plane method, a technique commonly used for solving non-differential optimization problems. A particular cutting plane algorithm is developed in Chapter 5 for solving problems of the form in (1.3).

## 1.2 Specialized Fast Solvers for Standard IQC Problems

In the second half of the thesis, we consider the optimization problems arising from standard IQC analysis.

The conventional way to treat standard IQC problems is to transform the frequency dependent matrix inequality in (1.1) into a non-frequency dependent one. The Kalman-Yakubovich-Popov (KYP) lemma states that the frequency dependent matrix inequality

holds if and only if there exists a symmetric matrix  $P$  such that

$$\begin{bmatrix} PA + A'P & PB \\ B'P & 0 \end{bmatrix} + \Sigma_0 + \sum_{i=1}^n \lambda_i \Sigma_i > 0. \quad (1.4)$$

Therefore, the inequalities in (1.1) and (1.2) can be equivalently expressed as (1.4). This transforms the standard IQC problems (1.1) and (1.2) into Semi-Definite Programs (SDPs) at the price of adding additional decision variables (i.e., the components of the matrix variable  $P$ ). Semi-definite programs can be viewed as generalizations of linear programs. Recently, many interior-point algorithms for solving linear programs have been extended to solve semi-definite programs [60, 75].

The number of additional decision variables which result from transforming the frequency dependent inequality is proportional to the square of the number of states in the system, or equivalently, the dimension of the matrix  $A$ . Therefore, when the number of states is substantially larger than the square root of the number of original decision variables, the additional decision variables play a dominant role in the equivalent SDP and become the major computational burden. Thus, most of the computational effort is spent on computing the auxiliary decision variables rather than the original ones. In this sense, the conventional method becomes “inefficient“ in the case where the system to be analyzed has many states.

Complex systems having many states are not unusual to encounter. Typical examples of such systems are aircraft control systems, vibration controllers for flexible structures, etc.. In fact, the motivation behind developing advanced analysis methods, such as the standard IQC analysis method, was the need to systematically analyze such systems. Furthermore, the state space of the transfer matrix  $M$  in (1.1) is the direct product of the state space of the physical system to be analyzed and the state spaces of the dynamical multipliers, if any, used in IQCs. Sometimes, in order to characterize the uncertainty in the system better, which consequentially will help to improve the accuracy of analysis, advanced dynamical multipliers which have many states are used. As a result, the resulting transfer matrix  $M$  has a large state space. See [55, 36] for more discussions on the use of dynamical multipliers to improve the accuracy of the analysis and the involvement of the states of these dynamical multipliers in IQC optimization problems.

The need for more efficient computational methods for solving SDPs derived from the

KYP lemma has been recognized a couple of years ago [87]. However, to author's knowledge, there have been no research efforts aimed at developing efficient solvers for standard IQC until very recently. There are two different approaches by which efficient solvers can be developed. One approach is to exploit the very special structure of the SDP and construct specialized interior point algorithms in which computation of directions of descent is performed efficiently. Research work in this direction includes [34, 77]. The second approach is to develop a different computational procedure such that no transformation to SDP is required for solving standard IQC problems. Along this direction, existing research work includes [61, 44].

The second part of the thesis (Chapters 6 and 7) is devoted to developing specialized computational algorithms that solve standard IQC problems in an efficient fashion. The crucial point in the development is to avoid the equivalent transformation. One of the alternative computational procedures proposed in this thesis is the cutting plane method. The development of the cutting plane method dates back to the 1950s, and a variety of cutting plane algorithms has been proposed over the years (See Chapter 14 of [11] and references therein). Three different cutting plane algorithms are considered in this thesis. We implement these algorithms and test them on a number of numerical examples. As we will demonstrate, the results of the numerical experiments indicate that the cutting plane method is much more efficient than the conventional approach of solving standard IQC problems in cases where the number of decision variables in the problem is small and the number of states in the problem is large.

Although the cutting plane algorithms appear to outperform the conventional approach in certain situations, it is well-known that they generally require many iterations to converge to a suboptimal solution with good accuracy. Furthermore, some numerical experiments indicate that the number of iterations grows substantially with the number of decision variables. Therefore, the cutting plane algorithms are expected to perform poorly when the number of decision variables in a problem is very large. This motivates us to consider another alternative, the interior point method.

The interior point method generally does not require many iterations to converge, in contrast to the cutting plane method. The development of interior point methods also dates back to the 1950s, with good early reference being [22]. Interior point methods have gained much attention and have become popular since Karmarkar introduced his famous

algorithm for solving linear programs [46], not only because his algorithm can be proved to have polynomial time worst-case complexity, but also because it works quite well practically. Another milestone in the development of interior point methods was the result by Nesterov and Nemirovsky [60]. They discovered that Karmarkar's algorithm, as well as several other polynomial time algorithms for solving linear programs, can be extended to solve a much larger class of convex optimization problems. The key element is the knowledge of a barrier function with a certain property called *self-concordance*. To be useful in practice, the barrier must be computable. Nesterov and Nemirovsky have shown that every finite dimensional convex set processes a self-concordant barrier function; however, their universal self-concordant barrier is generally *not* computable. There are only a few classes of problems for which readily computable self-concordant barrier functions are known. For the optimization problems resulting from IQC analysis, the only known computable self-concordant barrier function involves an auxiliary matrix variable which in some cases makes the computational algorithm based on this barrier function very inefficient.

We propose two new barrier functions to construct interior path-following algorithms for solving standard IQC problems. These barrier functions are readily computable: the main computation required to obtain their first and second derivatives is to solve Lyapunov equations for which efficient computational routines are widely available. Thanks to the new barrier functions, standard interior path-following algorithms can be applied to solve the IQC optimization problems without introducing auxiliary variables, which is why the interior point algorithms based on these newly introduced barrier functions are more efficient than the conventional approach. Regarding the issues of computational complexity, we have attempted to prove that the barrier functions are self-concordant in order to apply Nesterov and Nemirovsky's result. Our attempt was not quite successful, and we have not been able to determine whether the path-following algorithms using these new barrier functions are polynomial time algorithms or not. Nevertheless, we are able to show that the algorithms converge globally.

### 1.3 Contributions of This Thesis

Since mathematical models for engineering systems are inherently inaccurate, engineering designs based on such models have to be verified to ensure satisfactory performance in the

presence of internal perturbations and exogenous disturbances which are not captured by the models. Hence, system analysis becomes an important part of the engineering design process. To facilitate the design process, it is essential to have tools that can be used to analyze large-scale complex systems in an systematic, efficient, accurate, and automatic fashion. The development of the Integral Quadratic Constraint (IQC) based analysis method in the 1990s had the goal to provide such a tool. The purpose of this thesis is to further advance the IQC analysis framework. In the first part of this thesis, robustness analysis of a class of systems which appear in many important engineering applications - Linear Periodically Time-Varying (LPTV) systems - is considered. IQC-based analysis methods are developed for this class of problems. In the second part of the thesis, specialized fast computational algorithms are developed for solving the optimization problems arising from standard IQC analysis. These specialized fast solvers allow one to solve the optimization problems, and thus to verify the stability and performance of the system, in a very efficient fashion. The main contributions of the first part of this thesis are summarized as follows.

- We develop two systematic approaches for robustness analysis of LPTV systems. The development is closely related to the IQC analysis approach. These approaches are practical and flexible. In the first approach, the condition for robustness is posed as a semi-definite program, for which efficient commercial softwares exist. Therefore, checking the robustness condition can be performed in an automated and efficient manner. In the second approach, the condition for robustness is formulated as a special convex optimization problem with constraints defined in terms of parameterized operator inequalities which requires a new method to solve.
- We develop a computational algorithm to solve the convex optimization problem arising from the second approach for robustness analysis of LPTV systems. The development is based on a non-differentiable optimization technique called the cutting plane method. This computational algorithm allows one to check the robustness condition in an efficient fashion.
- We demonstrate the two approaches by applying them to a number of robustness analysis problems which are related to practical engineering applications. By these case studies, we compare the accuracy of the two approaches.



The contributions here are more at the implementation level and are important for practical engineering purposes. Using the ideas presented in the thesis, analysis of LPTV systems can be performed in a systematic fashion, and conditions for robustness can be verified automatically by computers.

The second part of the thesis is devoted to developing specialized fast solvers for standard IQC problems. The main contributions of this part of the thesis are summarized as follows.

- We develop a number of computational algorithms that are potentially much more efficient than the conventional approach for solving standard IQC problems. Specifically, three different cutting plane algorithms and two interior path-following algorithms are developed.
- We study the performance of these algorithms, as measured by the time the algorithm spends to solve an IQC problem, by applying them to a number of practical engineering problems. We also explain why the proposed algorithms are more efficient than the conventional approach from the point of view of worst-case computational complexity.
- A user friendly, MATLAB-based software for robustness analysis of large scale, complex dynamical systems is developed as a consequence of this part of research. This software, called the *IQC Toolbox*, has been used for the research purpose by several leading control groups around the world (at MIT, Caltech, Lund Institute of Technology, Royal Institute of Technology, and Stanford University). The toolbox is open to the public and available at <http://web.mit.edu/cykao/www/>.

## 1.4 Thesis Organization

The thesis is organized as follows: in the next chapter, we present mathematical concepts and terminology that will be used throughout the rest of the thesis. This chapter is also used to establish standard notations.

The first part of the thesis consists of Chapters 3, 4, and 5, where the topic of robustness analysis of periodic systems is discussed. In Chapter 3, two new techniques for robustness analysis of linear periodically time-varying systems are developed. In both approaches, Integral Quadratic Constraints (IQCs) are used to characterize the uncertainties and/or nonlinearities in the model. In Chapter 4, we address conditions for existence and stabil-

ity of stationary periodic solutions to periodic systems subject to periodic excitations, as well as conditions for certain harmonic performance of such systems. Integral Quadratic Constraints (IQCs) are used for developing these conditions. Certain infinite dimensional, convex optimization problems arise from the analysis in Chapters 3 and 4. In Chapter 5, a cutting plane algorithm is developed for solving such optimization problems. Examples to illustrate the robustness analysis techniques developed in Chapters 3 and 4 are also given in Chapter 5.

The second part of the thesis consists of Chapter 6 and Chapter 7, where specialized fast solvers for standard IQC problems are developed. In Chapter 6, three different cutting plane algorithms are implemented to solve standard IQC problems. The performance of these cutting plane algorithms are studied via a number of numerical examples and compared to the performance of the conventional approach of solving IQC problems. In Chapter 7, two new barrier functions are proposed to construct interior point algorithms for solving IQC optimization problems. The algorithms are also tested on numerical examples to evaluate their efficiency.

Finally, conclusions and remarks regarding future research direction are presented in Chapter 8. Some of the research work presented in this thesis has already been published in various journals and conference proceedings [37, 43, 44, 45, 40].

## Chapter 2

# Mathematical Preliminaries

The purpose of this chapter is to introduce notations and to overview mathematical concepts and some techniques that will be used throughout the thesis. Most of the notations and mathematical concepts defined in this chapter are more or less standard, and can be found in system theory or functional analysis textbooks (See, for instance [49]).

### 2.1 Notations and Standard Concepts

We start with a set of notations that will be used throughout the thesis. Let  $\mathbf{C}$  be the set of all complex numbers,  $\mathbf{C}^n$  be the set of  $n \times 1$  complex vectors, and  $\mathbf{C}^{n \times m}$  be the set of  $n \times m$  matrices whose elements are in  $\mathbf{C}$ . The notations  $\mathbf{R}$ ,  $\mathbf{R}^n$ ,  $\mathbf{R}^{n \times m}$  are used to denote the corresponding spaces where the elements are real numbers. We denote the set of integers by  $\mathbf{Z}$  and the set of non-negative real numbers by  $\mathbf{R}^+$ . Given a complex number  $a \in \mathbf{C}$ , we use  $\bar{a}$  to denote the complex conjugate of  $a$ .

We use  $I_n$  to denote the  $n \times n$  *identity* matrix. Sometimes the subscript  $n$  is dropped when the dimension of  $I_n$  is obvious from the context. Given a matrix  $M$ , the transposition and the conjugate transposition are denoted by  $M'$  and  $M^*$ , respectively. A matrix  $M$  is *symmetric* if  $M = M'$  and is *hermitian* if  $M = M^*$ . Let  $S^{n \times n}$  be the set of all  $n \times n$  real symmetric matrices and  $H^{n \times n}$  be the set of all  $n \times n$  hermitian matrices. Note that  $S^{n \times n}$  is a subset of  $H^{n \times n}$ . A matrix  $M$  is called *positive definite* if  $M$  belongs to  $H^{n \times n}$  and  $x'Mx > 0$  for all  $x \in \mathbf{C}^n, x \neq 0$ . The notations  $M > 0$  is used to denote positive definiteness. The positive semi-definiteness, negative definiteness, and negative semi-definiteness have similar definitions except that the “>” is replaced by “ $\geq$ ”, “<”, and “ $\leq$ ”, respectively. A matrix

$M$  is called *Hurwitz* if all its eigenvalues have strictly negative real part. Given a square matrix  $M$ , the notation  $\text{tr}(M)$  denotes the trace of  $M$ . The *Frobenius norm* of a square matrix  $M$  is defined as  $\|M\|_F := \sqrt{\text{tr}(M)}$ .

### 2.1.1 Normed Vector Spaces

A normed vector space is a vector space  $V$  over a scalar field  $F$  with a norm defined on it.

A norm is function  $\|\cdot\| : V \rightarrow \mathbf{R}^+$  such that the following axioms are satisfied

$$\|v\| \geq 0, \quad \forall v \in V, \quad \text{and} \quad \|v\| = 0 \iff v = 0.$$

$$\|\alpha v\| = |\alpha| \|v\|, \quad \forall v \in V, \quad \forall \alpha \in F.$$

$$\|v_1 + v_2\| \leq \|v_1\| + \|v_2\|, \quad \forall v_1, v_2 \in V.$$

The scalar field  $F$  considered in this thesis is either  $\mathbf{R}$  or  $\mathbf{C}$ . One example of a normed vector space is  $\mathbf{C}^n$ . We will equip  $\mathbf{C}^n$  three different norms

$$\|x\|_1 = \sum_{i=1}^n |x_i|, \quad \|x\|_2 = \left( \sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}}, \quad \|x\|_\infty = \max_i |x_i|.$$

We will simply write  $\|x\|$  for  $\|x\|_2$  unless state otherwise. In the thesis, we also consider norm vector spaces consisting of functions that map an infinite "time axis"  $\mathcal{T}$  into  $\mathbf{R}^n$  and that are square summable, where  $\mathcal{T}$  can be either  $(-\infty, \infty)$  or  $[0, \infty)$ . This space will be denoted by  $\mathbf{L}_2^n(-\infty, \infty)$  or  $\mathbf{L}_2^n[0, \infty)$  to explicitly distinguish what time axis is used. We usually drop the superscript  $n$  unless we want to emphasize the dimension of the signals in the space. The norm for the  $\mathbf{L}_2[0, \infty)$  space is defined as

$$\|f\|_{\mathbf{L}_2[0, \infty)} = \left( \int_0^\infty \|f(t)\|^2 dt \right)^{\frac{1}{2}},$$

The norm for the  $\mathbf{L}_2(-\infty, \infty)$  space is defined similarly. Another normed vector space to be considered in this thesis is the space of locally square integrable,  $T_0$ -periodic functions. We denote this space by  $\mathbf{L}_2^n(T_0)$ . Again, the superscript  $n$  is usually dropped unless the dimension of the signals is to be emphasized. The norm of this space is defined as

$$\|f\|_{\mathbf{L}_2(T_0)} = \left( \frac{1}{T_0} \int_0^{T_0} \|f(t)\|^2 dt \right)^{\frac{1}{2}}.$$

All the normed spaces mentioned above are complete; that is, all of their Cauchy sequences converge. Complete normed spaces are also called **Banach** spaces. However, we will not exploit the completeness property in this thesis.

## Extended Spaces

The extended space of a normed space consists of signals that may not be bounded in the norm of the vector space but any truncation to a finite time interval is bounded. To formalize the definition, let us consider the truncation operator  $P_T$  defined as

$$(P_T f)(t) = \begin{cases} f(t), & t \leq T \\ 0, & t > T \end{cases}$$

The extended space of a normed space  $V$  (with norm  $\|\cdot\|_V$ ) is then defined as

$$V_e = \{f \mid \|P_T f\|_V < \infty, \forall T \geq 0\}.$$

In this thesis, we only consider the extended spaces of  $\mathbf{L}_2$ , denoted by  $\mathbf{L}_{2e}$ . Furthermore, we notice that the norms of  $\mathbf{L}_2$  spaces satisfy

1. For every  $f \in \mathbf{L}_{2e}$ , if  $T_1 \leq T_2$ , then  $\|P_{T_1} f\|_{\mathbf{L}_2} \leq \|P_{T_2} f\|_{\mathbf{L}_2}$ .
2. For every  $f \in \mathbf{L}_{2e}$ , we have  $\|P_T f\|_{\mathbf{L}_2} \rightarrow \|f\|_{\mathbf{L}_2}$  as  $T \rightarrow \infty$ .

### 2.1.2 Inner Product Spaces

An inner product space is a vector space  $V$  over a field  $F$  with an inner product defined on it. An inner product is a function  $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbf{R}$ , which satisfies the following

$$\begin{aligned} \langle v, v \rangle &= 0 \iff v = 0. \\ \langle v_1, v_2 \rangle &= \overline{\langle v_2, v_1 \rangle}, \quad \forall v_1, v_2 \in V. \\ \langle \alpha v_1, v_2 \rangle &= \alpha \langle v_1, v_2 \rangle, \quad \forall v_1, v_2 \in V, \forall \alpha \in F. \\ \langle v_1 + v_2, v_3 \rangle &= \langle v_1, v_3 \rangle + \langle v_2, v_3 \rangle, \quad \forall v_1, v_2, v_3 \in V. \end{aligned}$$

Again, in this thesis, the field  $F$  can be  $\mathbf{R}$  or  $\mathbf{C}$ . The norm of an inner product space can be defined as  $\|v\| = \sqrt{\langle v, v \rangle}$ . Complete inner product spaces are called *Hilbert spaces*.

We equip  $\mathbf{C}^n(\mathbf{R}^n)$  with the inner product  $\langle x, y \rangle = x^*y = \sum_{i=1}^n \bar{x}_i y_i$ . The inner products of  $\mathbf{L}_2^n[0, \infty)$  spaces are defined as

$$\langle f, g \rangle = \int_0^\infty f(t)'g(t) dt.$$

By Plancherel formula, we have

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

where  $\hat{f}(j\omega)$ ,  $\hat{g}(j\omega)$  are the Fourier transforms of  $f(t)$  and  $g(t)$ , defined as

$$\hat{f}(j\omega) = \lim_{T \rightarrow \infty} \int_0^T f(t)e^{-j\omega t} dt, \quad \hat{g}(j\omega) = \lim_{T \rightarrow \infty} \int_0^T g(t)e^{-j\omega t} dt, \quad \omega \in \mathbf{R}.$$

The inner product for the bi-infinite case can be defined analogously. On the space  $\mathbf{L}_2^n(T_0)$ , the inner product is defined as

$$\langle f, g \rangle = \frac{1}{T_0} \int_0^{T_0} f(t)'g(t) dt.$$

similarly, if we let the Fourier coefficients of  $f \in \mathbf{L}_2^n(T_0)$  be defined as

$$\hat{f}_k = \frac{1}{T_0} \int_0^{T_0} u(t)e^{-j\omega_0 k t} dt, \quad k \in \mathbf{Z},$$

where  $\omega_0 = \frac{2\pi}{T_0}$ , then we have

$$\langle f, g \rangle = \sum_{k=-\infty}^\infty \hat{f}_k^* \hat{g}_k.$$

It can be verified that  $\mathbf{R}^n$ ,  $\mathbf{C}^n$ ,  $\mathbf{L}_2[0, \infty)$ ,  $\mathbf{L}_2(-\infty, \infty)$ , and  $\mathbf{L}_2(T_0)$  with the inner products described above are Hilbert spaces.

### 2.1.3 Operators and Induced Norms

An operator  $\Pi$  is a mapping from one normed space  $X$  into another normed space  $Y$ . In this thesis, we shall only consider the case where  $X = Y$ . An operator is *linear* if  $\Pi(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 \Pi(f_1) + \alpha_2 \Pi(f_2)$ . When an operator  $\Pi$  is linear, we will use the shorthand the notation  $\Pi f$  to denote  $\Pi(f)$ . In the thesis, we assume that all operators have

0 offset value; that is  $\Pi(0) = 0$ . Given an operator  $\Pi : X \rightarrow X$ , the *induced norm* of  $\Pi$  is defined as

$$\|\Pi\|_{ind} = \sup_{f \in X, \|f\|_X=1} \|\Pi(f)\|_X,$$

where  $\|\cdot\|_X$  denotes the norm of vector space  $X$ . We usually refer to the induced norm of an operator as the "gain" of the operator. An operator is bounded if it has finite gain; i.e., its induced norm is finite. Given operators  $\Pi_1, \Pi_2$ , it is easily to verify that  $\|\Pi_1\Pi_2\|_{ind} \leq \|\Pi_1\|_{ind} \cdot \|\Pi_2\|_{ind}$ .

An operator  $\Pi$  is called *causal* if  $P_T\Pi P_T = P_T\Pi$  for all  $T \geq 0$ , where  $P_T$  is the truncation operator defined in the previous section. Physically, this means that the current value of the output at the operator does not depend on the future values of the input. In other words, when considering the output of the current time instant, it does not matter if we truncate the future of the input. An operator is called *noncausal* if it is not causal.

Given a set of operators  $\Pi_i : V_i \rightarrow U_i, i = 1, \dots, n$ , the notation  $\Pi = \text{diag}(\Pi_1, \dots, \Pi_n)$  defines the operator  $\Pi$  which maps elements of the product space  $V_1 \times \dots \times V_n$  to those of the product space  $U_1 \times \dots \times U_n$ : let  $v_i \in V_i, i = 1, \dots, n$ . Then  $\Pi = \text{diag}(\Pi_1, \dots, \Pi_n)$  defines the operator

$$\Pi(v) := \begin{bmatrix} \Pi_1(v_1) \\ \vdots \\ \Pi_n(v_n) \end{bmatrix}, \quad \text{where } v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}.$$

To give an example, let us consider the special case where  $\Pi_1, \dots, \Pi_n$  are square matrices. Then  $\Pi = \text{diag}(\Pi_1, \dots, \Pi_n)$  defines the block diagonal matrix

$$\Pi = \begin{bmatrix} \Pi_1 & & \\ & \ddots & \\ & & \Pi_n \end{bmatrix}.$$

### 2.1.4 Adjoint Operators and Quadratic Forms

Let  $\Pi$  be a bounded linear operator from a Hilbert space  $X$  into itself. Then one can define an adjoint operator  $\Pi^* : X \rightarrow X$  associated with  $\Pi$  such that

$$\langle \Pi f, g \rangle = \langle f, \Pi^* g \rangle, \quad \forall f, g \in X.$$

An operator  $\Pi$  is called *self-adjoint* if  $\Pi = \Pi^*$ . A self-adjoint operator  $\Pi$  is called *positive semi-definite*, denoted by  $\Pi \geq 0$ , if  $\langle \Pi f, f \rangle \geq 0$  for all  $f \in X$ . It is called *positive definite*, denoted by  $\Pi > 0$ , if there exists an  $\varepsilon > 0$  such that  $\langle \Pi f, f \rangle \geq \varepsilon \|f\|^2$  for all  $f \in X$ . *Negative semi-definite* and *negative definite* self-adjoint operators are defined similarly. A bounded self-adjoint operator  $\Pi$  on a Hilbert space  $X$  defines a quadratic form  $\sigma : X \rightarrow \mathbf{R}$  by  $\sigma = \langle \Pi f, f \rangle$ . Again, we say a quadratic form  $\sigma$  is *positive semi-definite* if  $\sigma(f) \geq 0$  for all  $f \in X$  and *positive definite* if there exists an  $\varepsilon > 0$  such that  $\sigma(f) \geq \varepsilon \|f\|^2$  for all  $f \in X$ . *Negative semi-definite* and *negative definite* quadratic forms are defined analogously. Obviously,  $\sigma$  is positive (semi)-definite or negative (semi)-definite if and only if  $\Pi$  is positive (semi)-definite or negative (semi)-definite.

## 2.2 Linear Matrix Inequalities and Semi-Definite Programs

A *Linear Matrix Inequality* (LMI) has the form

$$F(x) := F_0 + \sum_{i=1}^n x_i F_i > 0, \quad (2.1)$$

where  $x \in \mathbf{R}^n$  is the variable and the symmetric matrices  $F_i \in \mathbf{R}^{n \times n}$ ,  $i = 0, 1, \dots, n$ , are given. A *Semi-Definite Program* (SDP) is a feasibility or linear objective optimization problem over a set of linear matrix inequalities. A semi-definite program is a convex optimization problem since its objective and constraint are convex. Although the LMI (2.1) may seem to have a specialized form, it represents a wide variety of convex constraints on  $x$ . Many convex optimization problems, including linear programming and (convex) quadratically constrained quadratic programming, can be cast as SDPs.

SDPs have been extensively studied. It is shown that SDPs can be solved very efficiently, both in theory and in practice. Commercial softwares for solving SDPs are available. For



more information on SDPs, the reader is referred to [9, 75].

## 2.3 S-procedure

In system analysis, we often have to check whether a quadratic form is negative semi-definite when other quadratic forms are all positive semi-definite. More precisely, given a Hilbert space  $X$  and quadratic forms  $\sigma_i$ ,  $i = 0, 1, \dots, n$ , we want to check whether

$$\sigma_0(f) \leq 0, \quad \text{for all } f \in X \text{ such that } \sigma_i(f) \geq 0, \quad i = 1, \dots, n. \quad (2.2)$$

A simple idea to check (2.2), which is now usually called the *S-procedure* in the systems and control community, is to solve the following problem:

$$\text{find } \tau_i \geq 0, \quad i = 1, \dots, n, \quad \text{such that } \sigma_0(f) + \sum_{i=1}^n \tau_i \sigma_i(f) \leq 0, \quad \forall f \in X. \quad (2.3)$$

Quite obviously, if we can find a set of positive  $\tau_i$  such that the inequality in (2.3) is satisfied, then (2.2) is true; i.e., feasibility of problem (2.3) is a sufficient condition for (2.2). In general, condition (2.2) does not imply feasibility of problem (2.3), except in some special cases. In the case where condition (2.2) and feasibility of problem (2.3) are equivalent, we call the S-procedure *lossless*. It is shown in [86] that the S-procedure is lossless in cases where

$$X = \mathbf{R}^n \text{ and } n = 1.$$

$$X = \mathbf{C}^n \text{ and } n = 2.$$

In the case of  $\mathbf{L}_2$ , Megretski and Treils [56] showed that the S-procedure is lossless for *any finite number* of quadratic forms that are defined by time invariant operators. The result was further generalized by Yakubovich in [82]. Here we state a version of the S-procedure losslessness theorem without proof.

**Theorem 2.1.** *Let  $X$  be a Hilbert space and  $\sigma_i : X \rightarrow \mathbf{R}$ ,  $i = 0, \dots, n$ , be quadratic forms defined as  $\sigma_i(f) = \langle \Pi_i f, f \rangle$ , where  $\Pi_i$  are bounded self-adjoint operators. Let  $S_\lambda$  be the shift operator on  $X$  defined as  $S_\lambda(f) = f(t - \lambda)$ . Suppose that the following conditions hold*

1. *If  $f \in X$  then  $S_\lambda f \in X$  for all  $\lambda \geq 0$ .*

2. For  $i = 0, \dots, n$ , and  $f_1, f_2 \in X$ , we have  $\langle \Pi_i S_\lambda f_1, f_2 \rangle \rightarrow 0$  as  $\lambda \rightarrow \infty$ .

3. For  $i = 0, \dots, n$ , and  $f_1, f_2 \in X$ , we have  $\langle \Pi_i f_1, S_\lambda f_2 \rangle \rightarrow 0$  as  $\lambda \rightarrow \infty$ .

4.  $\sigma_i(S_\lambda f) = \sigma_i(f)$  for all  $\lambda \geq 0$  and all  $f \in X$ .

Furthermore, there exists  $f^* \in X$  such that  $\sigma_i(f^*) > 0$  for  $i = 1, \dots, n$ . Then the following statements are equivalent

1.  $\sigma_0(f) \leq 0$ , for all  $f \in X$  such that  $\sigma_i(f) \geq 0$ ,  $i = 1, \dots, n$ .

2. There exist  $\tau_i \geq 0$ ,  $i = 1, \dots, n$ , such that  $\sigma_0(f) + \sum_{i=1}^n \tau_i \sigma_i(f) \leq 0$  for all  $f \in X$ .

## Chapter 3

# Robustness Analysis of Periodic Systems

Linear periodically time-varying systems appear in a number of engineering applications ranging from the general problem of stability analysis of systems subject to periodic excitation to the more specific problems arising in helicopter rotor control and study of stability of satellite orbits. In all cases, there are various sources of uncertainty in the linear periodic model. In this chapter, two new techniques for robustness analysis of linear periodically time-varying systems are developed. In both approaches, Integral Quadratic Constraints (IQCs) are used to characterize the uncertainties in the model. The condition for robustness is formulated as an equivalent problem of checking the feasibility of a particular affinely parameterized operator inequality.

### 3.1 Introduction

Linear Periodically Time-Varying (LPTV) systems appear in many important engineering applications. A typical example where LPTV systems naturally occur is the problem of robustness analysis of forced or unforced oscillations in a nonlinear system. Oscillatory behavior can be observed in many engineering systems, including delta-sigma modulators, automatic tuning of PID controllers, rotor motion of a magnetic bearing or a helicopter, or any machine which has a rotating part [19, 41]. To study the effect of perturbations on the oscillatory behavior of a nonlinear system, one can often linearize the perturbed system along the periodic trajectory of interest and perform analysis on the linearized system

[39]. This approach results in an uncertain linear periodically time-varying system. Other engineering applications in which one would encounter LPTV systems include analysis of the horizontal plane motion of ships in waves [73], study of the stability of satellite orbits [74], and stability analysis of beams and plates subject to periodic excitation [18] (See also survey paper [8] for more examples, motivation, and an overview of the status of the field).

Stability analysis of LPTV systems is often performed based on Floquet theory (See [12, 21, 8]). However, this technique is essentially restricted to the case where the uncertainty in the LPTV model is a perturbation of trivial size. Uncertain LPTV systems can also be analyzed to some extent by using the Nyquist Theorem developed in [33] (See also [8] for a related discussion). Another approach would be to use lifting techniques to obtain an equivalent infinite-dimensional but time-invariant system in discrete time [6, 5, 20, 8]. This approach has been used mainly to obtain  $H_\infty$  and  $H_2$  optimal controllers for discrete time LPTV systems [13, 76], and for rigorous analysis of sampled-data systems [5, 20, 57]. It can also be used for analysis of general continuous time uncertain LPTV systems, but not much work has been presented on this topic, to the author's knowledge. Absolute stability results for periodic systems have been developed by Yakubovich [81], where LPTV systems interconnected with simple nonlinearities were considered.

The objective of this chapter is to develop new techniques for robustness analysis of LPTV systems. Two different approaches are proposed in this chapter. In the first approach, the harmonic terms of the nominal LPTV model are extracted, and the time-varying coefficients are lumped into the feedback loop via a simple loop transformation; the system to be analyzed is thus transformed into the setup suited for standard robustness analysis methods - a Linear Time-Invariant (LTI) nominal subsystem in feedback interconnection with a bounded operator. Robustness analysis is then performed on the transformed system based on standard IQC analysis [55]. The criterion for robustness is formulated as an equivalent problem of checking feasibility of certain Linear Matrix Inequalities (LMIs). We refer to this approach as the *Fourier Series Expansion Method*. A new IQC is derived to give better characterization of the harmonic terms than the IQC in [55] does.

In the second approach, we allow the nominal system to remain periodic, and we extend the standard IQC theory to include the case where the nominal system is periodically time-varying. We also show how *periodic IQCs* can be used in the analysis. The development here is completely analogous to the time-invariant case in [55], with the exception that the

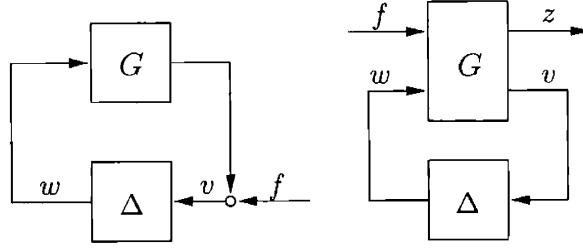


Figure 3-1: Systems for robust stability and performance analysis.

nominal system and the quadratic forms that define the IQC are allowed to be periodically time-varying. Furthermore, as opposed to LMIs in the Fourier Series Expansion Method, the stability condition of this new approach is expressed as a semi-infinite convex feasibility problem which involves operators defined on infinite dimensional Hilbert spaces. This approach is referred to as the *Periodic IQC Approach*. We will also briefly discuss how the semi-infinite feasibility problem can be solved using a non-differentiable optimization technique called the cutting plane method.

Much of the theoretical foundation of the techniques used in this chapter was laid down by previous work of other researchers. The main contribution here is to put together existing results and develop systematic approaches for robustness analysis of LPTV systems.

The organization of this chapter is as follows: in Section 3.2, we describe the uncertain LTV systems under consideration, as well as the type of analysis problems we want to study in this chapter. Next, two different approaches for robustness analysis of LPTV systems are presented. The Fourier Series Expansion Method is described in Section 3.3, and the Periodic IQC Approach is described in Section 3.4. Periodic IQCs for several uncertain operators are discussed in Section 3.5. Finally, we summarize the material of this chapter in Section 3.6.

## 3.2 Systems Under Consideration and Problem Formulation

In robustness analysis, systems under consideration are often represented in the forms shown in Figure 3-1. For robust stability we consider the system on the left-hand side

$$\begin{aligned} v &= Gw + f, \\ w &= \Delta(v), \end{aligned} \tag{3.1}$$

and for robust performance we consider the system on the right-hand side

$$\begin{bmatrix} z \\ v \end{bmatrix} = G \begin{bmatrix} f \\ w \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} f \\ w \end{bmatrix}, \quad (3.2)$$

$$w = \Delta(v),$$

where  $G$  is referred to as the nominal subsystem and  $\Delta$  is the collection of "trouble making" (nonlinear, time-varying, uncertain) components in the system.

In this chapter, we consider systems which have linear, bounded, and  $T_0$ -periodic nominal subsystems. The nominal subsystem has a state space realization

$$\begin{aligned} \dot{x}_G &= A_G(t)x_G + B_G(t)u, & x(0) &= 0, \\ y &= C_G(t)x_G + D_G(t)u, \end{aligned} \quad (3.3)$$

where all matrices are assumed to be real-valued,  $T_0$ -periodic, and continuous. Furthermore,  $A_G$  is assumed to be exponentially stable; i.e., the solution of  $\dot{\Phi}(t) = A_G(t)\Phi(t)$ ,  $\Phi(0) = I$ , evaluated at  $T_0$  has all eigenvalues inside the unit disc in the complex plane. Exponential stability of  $A_G$  ensures that the boundedness assumption on the nominal subsystem holds. The operator  $\Delta$  which can be either linear or non-linear is assumed to be bounded and causal on  $\mathbf{L}_{2e}[0, \infty)$ .

For robust stability analysis, we consider  $\mathbf{L}_2$ -stability in the sense that for any  $\mathbf{L}_2$ -input  $f$ , the corresponding signals  $v$  and  $w$  in the system are also in  $\mathbf{L}_2$ . The formal definition of stability is given as follows.

**Definition 3.1 (Well-posedness).** The feedback interconnection of  $G$  and  $\Delta$  as defined in equation (3.1) is said to be well-posed if the map  $(v, w) \mapsto f$  has a causal inverse on  $\mathbf{L}_{2e}[0, \infty)$ . That is, for any  $f \in \mathbf{L}_{2e}[0, \infty)$ , there exists a solution  $(v, w) \in \mathbf{L}_{2e}[0, \infty)$  which depends causally on  $f$ .

In other words, if the well-posedness assumption on the system is made, the map  $f \mapsto (v, w)$  is assumed to be well-defined and causal. If additionally the map is bounded, we say that the system is stable.

**Definition 3.2 (Stability).** The system in (3.1) is stable if it is well-posed and there exists a positive constant  $c$  such that  $\|P_T w\| + \|P_T v\| \leq c\|P_T f\|$ , for all  $T \geq 0$  and  $f \in \mathbf{L}_{2e}[0, \infty)$ .

For robust performance analysis, we are interested in verifying whether the signal pair  $(z, f)$  satisfies a certain performance measure. We formally define robust performance as follows.

**Definition 3.3 (Performance).** A performance measure  $\sigma_\Psi(z, f)$  is a quadratic functional which has the form

$$\sigma_\Psi(z, f) = \left\langle \begin{bmatrix} z \\ f \end{bmatrix}, \Psi \begin{bmatrix} z \\ f \end{bmatrix} \right\rangle_{\mathbf{L}_2[0, \infty)},$$

where  $\Psi$  is a self-adjoint operator on  $\mathbf{L}_2(-\infty, \infty)$ . The system in (3.2) is said to satisfy the performance criterion with respect to  $\sigma_\Psi$  if

- (i) the interconnection of  $(G_{22}, \Delta)$  is stable in the sense of Definition 3.2.
- (ii) the performance condition,  $\sigma_\Psi(z, f) \leq 0$ , holds for all pairs  $(z, f)$  from the set  $\{(z, f) \mid z = G_\Delta(f), f \in \mathbf{L}_2[0, \infty)\}$ . Here  $G_\Delta$  denotes the closed-loop map which is bounded because of assumption (i).

**Example 3.1.** As a simple example, take the quadratic functional  $\sigma_\Psi(z, f) = \|z\|^2 - \gamma^2 \|f\|^2$ . Then the performance criterion  $\sigma_\Psi(z, f) \leq 0$  simply implies that the worst case  $L_2$ -gain of the system  $z = G_\Delta(f)$  is bounded by  $\gamma$ .

In this chapter, we will discuss two alternative approaches to use IQCs for robust stability and performance analysis of the systems described in Figure 3-1.

### 3.3 Fourier Series Expansion Method

In this section, we propose the first method for robustness analysis of linear periodically time-varying systems. In the first approach, the system matrices are expanded into Fourier series. Then, the time-varying terms in the Fourier series are lumped into the feedback loop via a simple loop transformation, and the system to be analyzed is transformed into the standard setup for many existing robustness analysis methods - a Linear Time-Invariant (LTI) nominal subsystem in feedback interconnection with a bounded operator. Robustness analysis is then performed on the transformed system based on the standard IQC method [55].

Let us expand the matrices in the state space realization (3.3) into Fourier series

$$\begin{aligned} A_G(t) &= A_0 + \sum_{k \in I_A} (A_k^c \cos(k\omega_0 t) + A_k^s \sin(k\omega_0 t)) + \Delta_A(t), \\ B_G(t) &= B_0 + \sum_{k \in I_B} (B_k^c \cos(k\omega_0 t) + B_k^s \sin(k\omega_0 t)) + \Delta_B(t), \\ C_G(t) &= C_0 + \sum_{k \in I_C} (C_k^c \cos(k\omega_0 t) + C_k^s \sin(k\omega_0 t)) + \Delta_C(t), \\ D_G(t) &= D_0 + \sum_{k \in I_D} (D_k^c \cos(k\omega_0 t) + D_k^s \sin(k\omega_0 t)) + \Delta_D(t), \end{aligned}$$

where  $\omega_0 = 2\pi/T_0$ , and  $\Delta_A(t)$  to  $\Delta_D(t)$  represent the remaining terms in the Fourier series.

We assume that  $\Delta_A(t)$  is bounded in the sense

$$\|\Delta_A(t)\| \leq \gamma_A, \quad \forall t,$$

where  $\gamma_A$  is a constant and  $\|\cdot\|$  denotes the 2-induced matrix norm. Similar assumptions are made on  $\Delta_B(t)$ ,  $\Delta_C(t)$ , and  $\Delta_D(t)$ . The index sets  $I_A$  to  $I_D$  define the dominating harmonics of the Fourier expansion of  $A_G$  to  $D_G$  respectively. Let  $I_x \equiv I_A \cup I_C$  and  $I_u \equiv I_B \cup I_D$ . We assume that  $I_x$  is ordered; i.e.,  $I_x = \{k_1, k_2, \dots, k_N\}$ ,  $k_1 < k_2 < \dots < k_N$ , and so is  $I_u$ .

The system can now be transformed into a LTI system  $\tilde{G}$  interconnected with an operator

$$\Delta_{Tot} = \text{diag} (\Delta, H_x, \Delta_A, \Delta_C, H_u, \Delta_B, \Delta_D)$$

via a simple loop transformation. The LTI system  $\tilde{G}$  has a state space realization defined in terms of the Fourier coefficients. The operator  $H_x : \mathbf{L}_2^n[0, \infty) \rightarrow \mathbf{L}_2^{2nN}[0, \infty)$  is defined as multiplication in the time domain by the time-varying coefficients from the Fourier series expansion; i.e.,

$$H_x(v)(t) = \begin{bmatrix} \cos(k_1\omega_0 t)I \\ \sin(k_1\omega_0 t)I \\ \vdots \\ \cos(k_N\omega_0 t)I \\ \sin(k_N\omega_0 t)I \end{bmatrix} v(t).$$

$H_u$  is defined similarly. See Figure 3-2 for an illustration. The transformed system is now



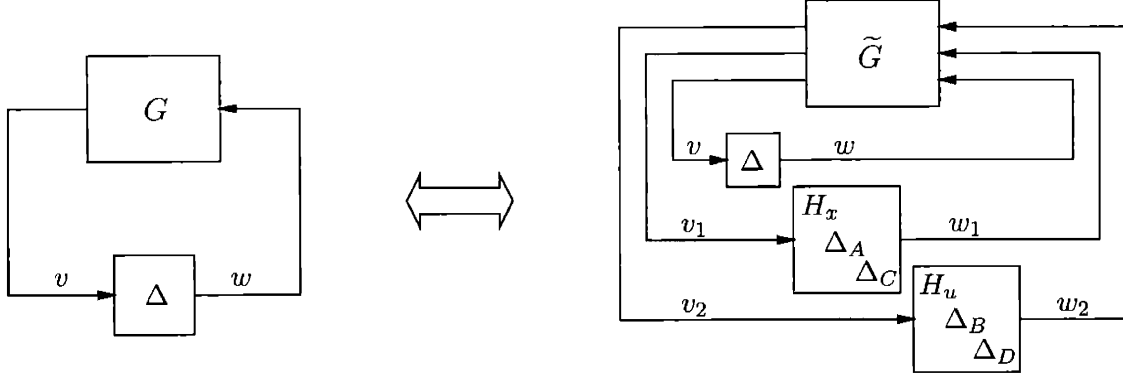


Figure 3-2: Illustration of the Fourier Series Expansion Method. Figure on the left-hand-side: the original system has a LPTV nominal subsystem  $G$  and a perturbation term  $\Delta$ . Figure on the right-hand-side: the transformed system has a LTI nominal subsystem  $\tilde{G}$ . The time-varying part of system  $G$  is extracted and lumped into the feedback loop. Let  $x$  denotes the states of  $\tilde{G}$ . Then signal  $v_1$  is equal to  $[x' \ x' \ x']'$ , while signal  $v_2$  is equal to  $[w' \ w' \ w']'$ .

in the form suitable for standard IQC analysis developed in [55]. Several IQCs that can be used to characterize the operators in  $\Delta_{\text{Tot}}$  are given in that paper. The next proposition, due to U. Jönsson, gives a new IQC for the operator  $H_x$ .

**Proposition 3.1.** *The operator  $H_x$  satisfies the IQC defined by the quadratic form*

$$\sigma(v, w) := \int_0^\infty \begin{bmatrix} v \\ w \end{bmatrix}' \begin{bmatrix} M_{11} & M_{12} \\ M'_{12} & M_{22} \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} dt, \quad (3.4)$$

where

$$\begin{bmatrix} M_{11} & M_{12} \\ M'_{12} & M_{22} \end{bmatrix} := \mathbf{Re}(U^* K U), \quad U = \begin{bmatrix} I & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & I & iI & 0 & \dots & 0 & 0 & 0 \\ & & & & \vdots & & & \\ 0 & 0 & 0 & 0 & \dots & 0 & I & iI \end{bmatrix},$$

and  $K = K^* \in \mathbf{C}^{(N+1)n \times (N+1)n}$  satisfies the LMI

$$\begin{bmatrix} A'PA - P & A'PB \\ B'PA & B'PB \end{bmatrix} + \begin{bmatrix} C' \\ D' \end{bmatrix} K \begin{bmatrix} C & D \end{bmatrix} \geq 0, \quad (3.5)$$

for some  $P = P^* \in \mathbf{C}^{nk_N \times nk_N}$ . Here  $A, B, C, D$  is any controllable realization of

$$H(z) = \frac{1}{(z+a)^{k_N}} \begin{bmatrix} I \\ z^{k_1} I \\ \vdots \\ z^{k_N} I \end{bmatrix}, \quad (3.6)$$

where  $a \neq 1$  is a real number.

*Remark 3.1.* The additional constraints,  $M_{11} \geq 0$  and  $M_{22} \leq 0$ , can be introduced to ensure the homotopy condition in the main theorem of [55].

*Proof.* The proof can be found in [37]. It is included here for the sake of completion.

The proposition is a consequence of the Kalman-Yakubovich-Popov (KYP) Lemma. To see this we first notice that the condition

$$\begin{bmatrix} I \\ z^{k_1} I \\ \vdots \\ z^{k_N} I \end{bmatrix}^* K \begin{bmatrix} I \\ z^{k_1} I \\ \vdots \\ z^{k_N} I \end{bmatrix} \geq 0, \quad \forall |z| = 1, \quad (3.7)$$

is equivalent to the condition  $H(z)^* K H(z) \geq 0$ , for all  $|z| = 1$ , which according to the KYP lemma is equivalent to the existence of  $P = P^* \in \mathbf{C}^{nk_N \times nk_N}$  such that (3.5) holds. Next, we notice that (3.7) implies that the following inequality holds

$$\begin{bmatrix} I \\ H_x \end{bmatrix}' U^* K U \begin{bmatrix} I \\ H_x \end{bmatrix} \geq 0, \quad \forall t. \quad (3.8)$$

To see this, notice that

$$U \begin{bmatrix} I \\ H_x \end{bmatrix} = \begin{bmatrix} I \\ e^{ik_1 \omega_0 t} I \\ \vdots \\ e^{ik_N \omega_0 t} I \end{bmatrix}. \quad (3.9)$$

Now, for any fixed  $t$ , let  $z = e^{i\omega_0 t}$ , and we immediately see that inequality (3.7) implies

inequality (3.8). Finally, since  $\mathbf{Im}(U^*KU)$  is skew-symmetric, therefore we have

$$\int_0^\infty \begin{bmatrix} v \\ H_x(v) \end{bmatrix}' \mathbf{Im}(U^*KU) \begin{bmatrix} v \\ H_x(v) \end{bmatrix} dt = 0. \quad (3.10)$$

Hence, inequality (3.8) implies the IQC defined by the quadratic form (3.4).  $\square$

Examples of using Fourier Series Expansion Method to analyze robust stability and performance of LPTV systems are presented in Section 5.5. In the next section, we present the second approach for robustness analysis for linear periodic systems.

### 3.4 Periodic IQC Approach

In this section, we consider another approach for robustness analysis of periodic systems (3.1) and (3.2). In this approach, the nominal subsystem  $G$  is allowed to remain linear periodically time-varying and *periodic IQCs* in the time domain are used to characterize the perturbation  $\Delta$ . By introducing a convex parameterization of the IQCs, we obtain robustness condition in terms of a convex optimization problem that involves state space realizations of the IQCs and the nominal subsystem  $G$ .

**Definition 3.4 (Periodic IQC).** Let the quadratic form  $\sigma_\Pi : \mathbf{L}_2[0, \infty) \times \mathbf{L}_2[0, \infty) \rightarrow \mathbf{R}$  be defined as

$$\sigma_\Pi(v, w) = \left\langle \begin{bmatrix} v \\ w \end{bmatrix}, \Pi \begin{bmatrix} v \\ w \end{bmatrix} \right\rangle_{\mathbf{L}_2[0, \infty)}, \quad (3.11)$$

where the operator  $\Pi$  is bounded, self-adjoint (on  $\mathbf{L}_2(-\infty, \infty)$ ) and  $T_0$ -periodic; i.e., if  $\eta_k(t) = \eta(t - kT_0)$ , then  $(\Pi\eta_k)(t) = (\Pi\eta_0)(t - kT_0)$  for all  $t \in \mathbf{R}$  and integers  $k$ . We say that the operator  $\Delta$  satisfies the IQC defined by  $\sigma_\Pi$  (Notation :  $\Delta \in \text{IQC}(\sigma_\Pi)$ ) if  $\sigma_\Pi(v, \Delta(v)) \geq 0$  for all  $v \in \mathbf{L}_2[0, \infty)$ . We refer to  $\Pi$  as the multiplier of the IQC.

**Example 3.2.** Consider an operator defined as multiplication by a  $T_0$ -periodically time-varying scalar  $\delta(t)$ ; i.e.,  $\Delta(v(t)) := \delta(t)v(t)$ , where  $\delta(t)$  satisfies  $\delta(t) = \delta(t + kT_0)$ , for all  $t \in \mathbf{R}$  and integers  $k$ . Suppose that  $|\delta(t)| \leq r(t)$  for some  $T_0$ -periodic and strictly positive

function  $r(t)$ . Then it is obvious that

$$r(t)^2 v(t)^2 - \Delta(v(t))^2 = (r(t)^2 - \delta(t)^2) v(t)^2 \geq 0, \quad \forall v(t).$$

Let the quadratic form  $\sigma_{\Pi}(v, w)$  be

$$\sigma_{\Pi}(v, w) := \int_0^{\infty} r(t)^2 v(t)^2 - w(t)^2 dt.$$

It is clear that  $\sigma_{\Pi}(v, \Delta(v)) \geq 0$  for all  $v \in \mathbf{L}_2$ . Therefore, the operator  $\Delta$  satisfies the IQC defined by  $\sigma_{\Pi}$ . The multiplier of this IQC is

$$\Pi = \begin{bmatrix} r(t)^2 & 0 \\ 0 & -1 \end{bmatrix}.$$

IQCs of the form given in Definition 3.4 will be used to obtain conditions for robust stability and performance of the systems (3.1) and (3.2). The periodic case is completely analogous to the time-invariant case in [55], and the main stability result of [55] also holds for the periodic case.

**Proposition 3.2 (Robust Stability).** *Consider system (3.1) where  $G$  is a stable LPTV system and  $\Delta$  is a bounded causal operator. Assume that*

- (i) *for all  $\tau \in [0, 1]$ , the interconnection of  $G$  and  $\tau\Delta$  is well-posed;*
- (ii) *for all  $\tau \in [0, 1]$ ,  $\tau\Delta$  satisfies the IQC defined by  $\sigma_{\Pi}$ ;*
- (iii) *there exists an  $\varepsilon > 0$  such that*

$$\sigma_{\Pi}(Gw, w) \leq -\varepsilon \|w\|^2, \quad \forall w \in \mathbf{L}_2[0, \infty).$$

*Then the system is stable.*

*Proof.* This proposition can be proven by the arguments similar to those in the proofs of the main results in [55] and [66]. □

The next proposition gives the condition for robust performance of system (3.2). It follows from the  $S$ -procedure (See Chapter 2).

**Proposition 3.3 (Robust Performance).** *Consider system (3.2). Let  $\sigma_\Psi$  correspond to the performance measure, and let  $\sigma_\Pi$  be the quadratic functional of the form in (3.11). Assume that*

- (i)  $\sigma_\Psi(z, 0) \geq 0, \forall z \in \mathbf{L}_2[0, \infty)$ ;
- (ii) for all  $\tau \in [0, 1]$ , the interconnection of  $(G_{22}, \tau\Delta)$  is well-posed;
- (iii) for all  $\tau \in [0, 1]$ ,  $\tau\Delta$  satisfies IQC defined by  $\sigma_\Pi$ ;
- (iv) there exists an  $\varepsilon > 0$  such that

$$\sigma_\Psi(z, f) + \sigma_\Pi(v, w) \leq -\varepsilon(\|f\|^2 + \|w\|^2), \quad \forall f, w \in \mathbf{L}_2[0, \infty).$$

where  $(z, v, f, w)$  satisfies system equations in (3.2).

Then the system in (3.2) satisfies the performance criterion  $\sigma_\Psi \leq 0$ .

*Proof.* Let  $(f, w) := (0, w)$ . Then conditions (i) and (iv) imply that there exists an  $\varepsilon > 0$  such that

$$\sigma_\Pi(G_{22}w, w) \leq -\varepsilon\|w\|^2, \quad \forall w \in \mathbf{L}_2[0, \infty).$$

Therefore, the interconnection of  $(G_{22}, \Delta)$  satisfies all three conditions in Proposition 3.2 and is thus stable. Finally, the performance condition,  $\sigma_\Psi(z, f) \leq 0$  for all  $(z, f)$  such that  $z = G_\Delta(f)$  and  $f \in \mathbf{L}_2[0, \infty)$ , is obtained by applying the  $S$ -procedure.  $\square$

In applications, it is important to have several IQCs that characterize the uncertainty in order to obtain more accurate analysis results; the more IQCs we have to describe the uncertainty, the better our chance to find one such that condition (iii) in Proposition 3.2 or condition (iv) in Proposition 3.3 is satisfied.

In the rest of this section, we show that it is possible to parameterize IQCs and that checking conditions for robustness becomes finding values of the parameters in the parameterized IQCs such that condition (iii) in Proposition 3.2 or condition (iv) in Proposition 3.3 is satisfied. As a result, the conditions for robust stability and performance are expressed as infinite dimensional convex feasibility problems. We first introduce a state space realization for the operator  $\Pi$  in Definition 3.4.

### 3.4.1 The State-space Realization of the Multiplier $\Pi$

Consider multipliers of the form  $\Pi = \Gamma^* \Sigma(t) \Gamma$ , where  $\Sigma(t)$  is a real symmetric, continuous, and  $T_0$ -periodic matrix. The operator  $\Gamma$  is defined as

$$(\Gamma\eta)(t) = \begin{bmatrix} \int_0^t \Phi_\Pi(t, s) B_\Pi(s) \eta(s) ds \\ \eta(t) \end{bmatrix}, \quad \eta(t) = \begin{bmatrix} v(t) \\ w(t) \end{bmatrix}, \quad (3.12)$$

where  $B_\Pi(t) = \begin{bmatrix} B_{\Pi, v}(t) & B_{\Pi, w}(t) \end{bmatrix}$ , and  $\Phi_\Pi$  is the transition matrix defined as

$$\dot{\Phi}_\Pi(t, s) = A_\Pi(t) \Phi_\Pi(t, s), \quad \Phi_\Pi(s, s) = I.$$

Matrices  $A_\Pi$ ,  $B_\Pi$  are  $T_0$ -periodic, real-valued, and continuous. Furthermore,  $A_\Pi$  is exponentially stable. Now, the quadratic forms defined by  $\Pi$  can be represented in the time domain as

$$\sigma_\Pi(v, w) = \int_0^\infty \begin{bmatrix} x_\Pi(t) \\ \eta(t) \end{bmatrix}' \Sigma(t) \begin{bmatrix} x_\Pi(t) \\ \eta(t) \end{bmatrix} dt, \quad (3.13)$$

where  $x_\Pi(t)$  satisfies the dynamical equation

$$\dot{x}_\Pi(t) = A_\Pi(t) x_\Pi(t) + B_{\Pi, v}(t) v(t) + B_{\Pi, w}(t) w(t), \quad x_\Pi(0) = 0.$$

### 3.4.2 Parameterizing and Optimizing IQCs

Suppose that  $\tau\Delta$  satisfies the IQCs with multipliers  $\Pi_0, \Pi_1, \dots, \Pi_n$ . Then it is straightforward to see that  $\tau\Delta$  also satisfies any IQC with multiplier

$$\Pi(\lambda) = \Pi_0 + \sum_{i=1}^n \lambda_i \Pi_i, \quad \text{where } \lambda_i \geq 0, \quad i = 1, \dots, n. \quad (3.14)$$

In other words,  $\tau\Delta$  satisfies a set of IQCs that is parameterized by  $\lambda$ , where  $\lambda$  belongs to the positive orthant of  $\mathbf{R}^n$ . The stability analysis problem now becomes the problem of finding a positive  $\lambda$  such that criterion (iii) in Proposition 3.2 is satisfied. Hence, it can be

formulated as a convex feasibility problem:

$$\text{Find } \lambda \geq 0, \varepsilon > 0, \text{ such that } \sigma_{\Pi}(Gw, w, \lambda) \leq -\varepsilon\|w\|^2, \quad \forall w \in \mathbf{L}_2[0, \infty). \quad (3.15)$$

Here the notation  $\sigma_{\Pi}(\cdot, \cdot, \lambda)$  is used to emphasize that the quadratic form is parameterized by  $\lambda$ . In applications, depending on the properties of  $\Delta$ , the constraints on parameter  $\lambda$  could be replaced by  $\lambda \in \Lambda$  where  $\Lambda$  is a convex set of  $\mathbf{R}^n$ . We will give several examples of parameterization of IQCs in Section 3.5.

The condition for robust performance can be treated similarly. There will, however, be a distinctive difference. In the robust performance analysis, we normally want to optimize a certain parameter in the performance criterion. Analogous to the stability analysis case, let us assume that  $\tau\Delta$  satisfies an  $\eta$ -parameterized IQCs; i.e., the IQC has an  $\eta$ -parameterized multipliers  $\bar{\Pi}(\eta) := \sum_{i=1}^n \eta_i \Pi_i$ , where  $\eta$  belongs to a convex set  $\mathcal{C}_1 \subset \mathbf{R}^n$ . Furthermore, let the performance measure  $\sigma_{\Psi}$  has the multiplier  $\Psi(y; b'y)$ , where  $y \in \mathcal{C}_2$  denotes the parameters in the performance multiplier, and  $b'y$  denotes the performance objective we want to optimized. The set  $\mathcal{C}_2$  is a convex set in  $\mathbf{R}^m$ . As an example, let us consider the performance measure  $\sigma_{\Psi}(z, f) = \|z\|^2 - \gamma^2\|f\|^2$ . If the performance criterion holds, then the system has  $\mathbf{L}_2$ -gain no larger than  $\gamma$ ; i.e.,  $\gamma$  is an upper bound of the  $\mathbf{L}_2$ -gain of the system. In order to have a better estimate of the  $\mathbf{L}_2$ -gain, we usually want to minimize  $\gamma$ . In this case, define  $y := \gamma^2$ . Then we have one parameter  $y$  in  $\sigma_{\Psi}$ . The set  $\mathcal{C}_2$  is equal to  $\mathbf{R}^+$ , and the vector  $b$  is equal to 1.

Let  $\bar{\Pi}(\eta)$  and  $\Psi(y)$  be partitioned as

$$\bar{\Pi}(\eta) = \begin{bmatrix} \bar{\Pi}_{11}(\eta) & \bar{\Pi}_{12}(\eta) \\ \bar{\Pi}_{12}^*(\eta) & \bar{\Pi}_{22}(\eta) \end{bmatrix}, \quad \Psi(y) = \begin{bmatrix} \Psi_{11}(y; b'y) & \Psi_{12}(y; b'y) \\ \Psi_{12}^*(y; b'y) & \Psi_{22}(y; b'y) \end{bmatrix},$$

where the dimensions of  $\bar{\Pi}_{11}(\eta)$  and  $\Psi_{11}(y)$  correspond to the size of signal  $v$  and signal  $z$ , respectively. Let  $\lambda' = \begin{bmatrix} \eta' & y' \end{bmatrix}$ , and

$$\Pi(\lambda) = \begin{bmatrix} \Psi_{11}(y; b'y) & 0 & \Psi_{12}(y; b'y) & 0 \\ 0 & \bar{\Pi}_{11}(\eta) & 0 & \bar{\Pi}_{12}(\eta) \\ \Psi_{12}^*(y; b'y) & 0 & \Psi_{22}(y; b'y) & 0 \\ 0 & \bar{\Pi}_{12}^*(\eta) & 0 & \bar{\Pi}_{22}(\eta) \end{bmatrix}.$$

Then the quadratic form in condition (iv) can be represented as

$$\sigma(\tilde{v}, \tilde{w}, \lambda) := \sigma_{\Psi}(z, f, y) + \sigma_{\Pi}(v, w, \eta) = \left\langle \begin{bmatrix} \tilde{v} \\ \tilde{w} \end{bmatrix}, \Pi(\lambda) \begin{bmatrix} \tilde{v} \\ \tilde{w} \end{bmatrix} \right\rangle_{\mathbf{L}_2[0, \infty)}, \quad (3.16)$$

where  $\tilde{v}' = \begin{bmatrix} z' & v' \end{bmatrix}$  and  $\tilde{w}' = \begin{bmatrix} f' & w' \end{bmatrix}$ . Criterion (iv) in Proposition 3.3 can now be formulated as

$$\inf_{\lambda} c' \lambda, \quad \text{subj. to} \quad \begin{cases} \sigma(G\tilde{w}, \tilde{w}, \lambda) \leq -\varepsilon \|\tilde{w}\|^2, \quad \forall \tilde{w} \in \mathbf{L}_2[0, \infty) \\ \lambda \in \Lambda, \quad \varepsilon > 0 \end{cases}. \quad (3.17)$$

where  $c' = \begin{bmatrix} 0 & \dots & 0 & b' \end{bmatrix}$ , and  $\Lambda := \mathcal{C}_1 \times \mathcal{C}_2 \in \mathbf{R}^{n+m}$ .

Suppose that each multiplier  $\Pi_i$  is in the form  $\Gamma^* \Sigma_i(t) \Gamma$  where  $\Gamma$  is defined as (3.12) in Section 3.4.1. Using the state space realizations of  $G$  and  $\Gamma$ , we can equivalently represent the integral quadratic inequalities in (3.15) and (3.17) as: there exists an  $\varepsilon > 0$  such that

$$\int_0^{\infty} (x' Q(t, \lambda) x + 2x' F(t, \lambda) w + w' R(t, \lambda) w) dt \geq \varepsilon (\|x\|^2 + \|w\|^2), \quad \forall (x, w) \in \mathcal{L}, \quad (3.18)$$

where  $x' = (x'_{\Pi}, x'_G)$ ,

$$\mathcal{L} = \{(x, w) \mid \dot{x} = A(t)x + B(t)w, x(0) = 0, w \in \mathbf{L}_2[0, \infty)\}, \quad (3.19)$$

$$A(t) = \begin{bmatrix} A_{\Pi}(t) & B_{\Pi, v}(t) C_G(t) \\ 0 & A_G(t) \end{bmatrix}, \quad B(t) = \begin{bmatrix} B_{\Pi, v}(t) D_G(t) + B_{\Pi, w}(t) \\ B_G(t) \end{bmatrix},$$

and

$$\begin{bmatrix} Q(t, \lambda) & F(t, \lambda) \\ F(t, \lambda)' & R(t, \lambda) \end{bmatrix} = -N(t)' \left( \Sigma_0(t) + \sum_{i=1}^n \lambda_i \Sigma_i(t) \right) N(t), \quad N(t) = \left[ \begin{array}{cc|c} I & 0 & 0 \\ 0 & C_G(t) & D_G(t) \\ \hline 0 & 0 & I \end{array} \right].$$

Note that  $A(t)$  is exponentially stable because  $A_G(t)$  and  $A_{\Pi}(t)$  are. Furthermore,  $\mathcal{L} \subset \mathbf{L}_2[0, \infty)$  is a Hilbert space. This follows from the closed graph theorem since  $A(t)$  is exponentially stable. Also note that  $Q(t, \lambda)$ ,  $R(t, \lambda)$  are symmetric and that  $Q(t, \lambda)$ ,  $F(t, \lambda)$



$R(t, \lambda)$  all depend affinely on  $\lambda$ . Now, if we define

$$S_\lambda := \begin{bmatrix} Q(t, \lambda) & F(t, \lambda) \\ F(t, \lambda)' & R(t, \lambda) \end{bmatrix} \quad (3.20)$$

and view  $S_\lambda$  as a self-adjoint operator defined on  $\mathcal{L}$ , then (3.18) can be equivalently expressed as  $S_\lambda > 0$ . The problem of robust stability and performance analysis problems now can be formulated as feasibility/optimization problems that involve the positivity of  $S_\lambda$ .

**Proposition 3.4.** *Consider system (3.1). Assume that, for all  $\tau \in [0, 1]$ , the interconnection of  $G$  and  $\tau\Delta$  is well-posed and  $\tau\Delta$  satisfies  $\lambda$ -parameterized IQC which has a multiplier in the form of  $\Pi(\lambda) = \Pi_0 + \sum_{i=1}^n \lambda_i \Pi_i$ ,  $\lambda \in \Lambda$ . Furthermore, assume that each  $\Pi_i$  is in the form of  $\Pi_i = \Gamma^* \Sigma_i(t) \Gamma$ , where  $\Gamma$  is defined as in (3.12). Then system (3.1) is stable if the following convex feasibility problem has a solution:*

$$\text{find } \lambda \in \Lambda, \quad \text{such that } S_\lambda > 0, \quad (3.21)$$

where  $S_\lambda$  is defined according to state space representations of the nominal subsystem  $G$  and the multiplier in the IQC.

**Proposition 3.5.** *Consider system (3.2). Assume that, for all  $\tau \in [0, 1]$ , the interconnection of  $G_{22}$  and  $\tau\Delta$  is well-posed and  $\tau\Delta$  satisfies  $\lambda$ -parameterized IQC which has a multiplier in the form of  $\Pi(\lambda) = \Pi_0 + \sum_{i=1}^n \lambda_i \Pi_i$ ,  $\lambda \in \Lambda$ . Furthermore, assume that each  $\Pi_i$  is in the form of  $\Pi_i = \Gamma^* \Sigma_i(t) \Gamma$ , where  $\Gamma$  is defined as in (3.12). Then the robust performance problem can be formulated as an optimization problem*

$$\inf_{\lambda} c' \lambda, \quad \text{subj. to } \lambda \in \Lambda, \quad S_\lambda > 0, \quad (3.22)$$

where  $S(\lambda)$  is defined according to state space representations of the nominal subsystem  $G$  and the multiplier in the IQC.

Feasibility problem (3.21) and optimization problem (3.22) can be addressed using methods from non-differentiable optimization. For example, various cutting plane algorithms [11, 28, 44] can be applied to solve problems (3.21) and (3.22). In Chapter 5, a specific cutting plane algorithm is developed to solve feasibility and optimization problems arising in this chapter.

In Section 5.5, we illustrate and compare the two different approaches for robustness analysis of periodic systems by a couple of examples. As we will show, the Periodic IQC Approach is generally more accurate than the Fourier Series Expansion Method.

### 3.5 Parameterizations of Some IQCs

In this section, we give several examples of parameterizing IQCs.

#### 3.5.1 Periodically Varying Parameter

Consider an operator defined as multiplication by a periodically time-varying scalar  $\delta(t)$ . Suppose that  $|\delta(t)| \leq r(t)$  for some  $T_0$ -periodic and strictly positive function  $r(t)$ . Let

$$\sigma(v, w, \lambda) = \int_0^\infty \lambda(v(t)^2 - (w(t)/r(t))^2) dt,$$

where  $\lambda > 0$ . It can be readily verified that  $\sigma(v, \delta v, \lambda) \geq 0$ . The multiplier of this IQC is  $\text{diag}(\lambda, -\frac{\lambda}{r(t)})$ .

#### 3.5.2 Time-varying uncertainty

Consider an operator defined as multiplication by a time-varying matrix  $\Delta(t)$ , where  $\Delta(t)$  belongs to a convex polytope; i.e.,

$$\Delta(t) \in \left\{ \Delta \mid \Delta = \sum_{i=1}^N \eta_i \Delta_i, \quad \eta_i \geq 0, \quad \sum_{i=1}^N \eta_i = 1 \right\} \subset \mathbf{R}^{m \times n},$$

where  $\Delta_1, \dots, \Delta_N$  are fixed matrices. Let

$$\mathcal{M} = \left\{ M = M' : \begin{bmatrix} I \\ \Delta_i \end{bmatrix}' M \begin{bmatrix} I \\ \Delta_i \end{bmatrix} \geq 0, \quad i = 1, \dots, N \right\}$$

and

$$\Sigma(t, M_0, \dots, M_r) = M_0 + \sum_{k=1}^r \rho_k(t) M_k,$$

where  $M_k \in \mathcal{M}$ , and  $\rho_k(t) \geq 0$  are  $T_0$ -periodic and continuous. Let  $\lambda$  represent the coefficients of the matrices  $M_k$ . Then  $\Sigma(t, M_0, \dots, M_r)$  can be represented as

$$\Sigma(t, \lambda) = \Sigma_0(t) + \sum_{i=1}^s \lambda_i \Sigma_i(t). \quad (3.23)$$

Notice that the conditions  $M_k \in \mathcal{M}$  restrict  $\lambda$  to a convex set  $\Lambda$  which is defined in terms of linear matrix inequalities. Finally, let

$$\sigma(v, w, \lambda) = \int_0^\infty \begin{bmatrix} v(t) \\ w(t) \end{bmatrix}' \Sigma(t) \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} dt.$$

It can be easily verified that  $\sigma(v, \Delta v, \lambda) \geq 0$  for every  $\lambda \in \Lambda$  and for all  $v(t) \in \mathbf{L}_2[0, \infty)$ .

### 3.5.3 Uncertain parameter

Here we consider the operator defined as multiplication in the time domain by an uncertain parameter  $\delta \in [-1, 1]$ . It is easy to verify that it satisfies the IQC defined by the quadratic form

$$\sigma_1(v, w) := \left\langle \begin{bmatrix} v \\ w \end{bmatrix}, \Pi_1 \begin{bmatrix} v \\ w \end{bmatrix} \right\rangle_{\mathbf{L}_2[0, \infty)}, \quad \Pi_1 = \begin{bmatrix} X & 0 \\ 0 & -X \end{bmatrix},$$

where  $X = X^* \geq 0$  is a  $T_0$ -periodic operator. In order to obtain a state space realization we let  $X = X_0 + X_0^*$  and

$$X_0 = \sum_{k=0}^K N_k \psi_k,$$

where<sup>1</sup>  $N_0, \dots, N_K$  are  $n \times n$  matrices, and  $\psi_k$  are  $T_0$ -periodic operators. For example, we can take  $\psi_0(t)$  to be a continuous, real-valued,  $T_0$ -periodic function and  $\psi_k$ ,  $k = 1, \dots, K$ , to be the operators with first order state space realizations<sup>2</sup>

$$\begin{aligned} \dot{x}_k &= a_k(t)x_k + b_k(t)u, \\ y &= c_k(t)x_k, \end{aligned}$$

<sup>1</sup> $n$  is the size of the signal that  $\delta$  multiplies.

<sup>2</sup>We need to interpret  $N_k \psi_k$  as  $N_k(\psi_k I)$ , i.e.,  $\psi_k$  is a diagonal operator.

where  $a_k, b_k, c_k$  are continuous, real-valued, and  $T_0$ -periodic functions. This way, we can write  $X = \Psi_0^* \Sigma_0 \Psi_0$ , where the operator  $\Psi_0$  is of the form (3.12). The corresponding matrices  $A_\Pi$  and  $B_\Pi$  are defined in terms of  $a_1, \dots, a_K$ , and  $b_1, \dots, b_K$ , while the matrix  $\Sigma_0$  is defined in terms of  $N_0, \dots, N_K, \psi_0$ , and  $c_1, \dots, c_K$ . Let  $\lambda_1 \in \mathbf{R}^{(K+1)n^2}$  correspond to the coefficients in  $N_0, \dots, N_K$ . Then we can represent  $\Sigma_0(t, \lambda_1)$  in the form given in (3.23).

The parameters in  $\lambda_1$  are constrained by the condition that  $X \geq 0$ . This can be written as a convex IQC constraint in the following way: let  $\sigma_0(v, \lambda_1) = \langle \Psi_0 v, \Sigma_0(t, \lambda_1) \Psi_0 v \rangle$ , then  $\lambda_1 \in \Lambda_1 := \{\lambda : \sigma_0(v, \lambda) \geq 0, \forall v \in \mathbf{L}_2[0, \infty)\}$ . Now, let the state space realization of  $\Psi_0$  be

$$\left[ \begin{array}{c|c} A_0 & B_0 \\ \hline I & 0 \\ 0 & I \end{array} \right].$$

and let  $\Sigma_0$  be partitioned as

$$\Sigma_0 = \begin{bmatrix} \Sigma_{0,11} & \Sigma_{0,12} \\ \Sigma_{0,21} & \Sigma_{0,22} \end{bmatrix},$$

where matrix  $\Sigma_{0,11}$  has the same dimension as matrix  $A_0$ . Then, it can be readily verified that  $\Pi_1$  has the following representation

$$\Pi_1 = \Psi_1^* \Sigma_1 \Psi_1,$$

where  $\Psi_1$  has state space realization

$$\Psi_1 = \left[ \begin{array}{cc|cc} A_0 & 0 & B_0 & 0 \\ 0 & A_0 & 0 & B_0 \\ \hline I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{array} \right],$$

and

$$\Sigma_1 = \begin{bmatrix} \Sigma_{0,11} & 0 & \Sigma_{0,12} & 0 \\ 0 & -\Sigma_{0,11} & 0 & -\Sigma_{0,12} \\ \Sigma'_{0,12} & 0 & \Sigma_{0,22} & \\ 0 & -\Sigma'_{0,12} & 0 & -\Sigma_{0,22} \end{bmatrix}.$$

We have  $\sigma_1(v, \delta v, \lambda_1) \geq 0$ , for all  $\lambda_1 \in \Lambda_1$  and for all  $v \in \mathbf{L}_2[0, \infty)$ .

We can also use the IQC defined by the quadratic form

$$\sigma_2(v, w) := \left\langle \begin{bmatrix} v \\ w \end{bmatrix}, \Pi_2 \begin{bmatrix} v \\ w \end{bmatrix} \right\rangle_{\mathbf{L}_2[0, \infty)}, \quad \Pi_2 = \begin{bmatrix} 0 & Y \\ Y^* & 0 \end{bmatrix},$$

where  $Y = -Y^*$  is a  $T_0$ -periodic operator. In order to obtain a state space realization we let  $Y = Y_0 - Y_0^*$  and

$$Y_0 = \sum_{k=1}^K N_k \psi_k,$$

where the  $\psi_k$ ,  $k = 1, \dots, K$ , are  $T_0$ -periodic operators of the same type as in the definition of the quadratic form  $\sigma_1$  discussed above<sup>3</sup>. It is easy to derive a representation  $\Pi_2 = \Psi_2^* \Sigma_2 \Psi_2$ , where  $\Sigma_2(t, \lambda_2)$  depends affinely on  $\lambda_2$ . Here  $\lambda_2 \in \mathbf{R}^{Kn^2}$  denotes the coefficients in  $N_1, \dots, N_K$ . We have  $\sigma_2(v, \delta v, \lambda_2) \geq 0$ , for all  $\lambda_2$  and for all  $v \in \mathbf{L}_2[0, \infty)$ . Note that there is no constraint on  $\lambda_2$ .

The two IQCs defined by quadratic forms  $\sigma_1$  and  $\sigma_2$  can be combined to a single IQC, which is defined by the quadratic form  $\sigma = \sigma_1 + \sigma_2$  with the multiplier  $\Pi = \Psi^* \Sigma(\lambda) \Psi$ , where  $\Psi$  has the realization in the form of (3.12), and  $\lambda \in \Lambda = \{\lambda = (\lambda_1, \lambda_2) : \lambda_1 \in \Lambda_1\}$ . To be more precise, if  $\sigma_i$  is defined by  $\Pi_i = \Psi_i^* \Sigma_i \Psi_i$ , where  $\Psi_i$  has state space realization

$$\left[ \begin{array}{c|c} A_i & B_i \\ \hline I & 0 \\ 0 & I \end{array} \right],$$

---

<sup>3</sup>The matrices  $N_1, \dots, N_K$  and the  $\psi_k$  need not be identical to the ones that defined  $\sigma_1$ .

and  $\Sigma_i$  is partitioned as

$$\Sigma_i = \begin{bmatrix} \Sigma_{i,11} & \Sigma_{i,12} \\ \Sigma'_{i,21} & \Sigma_{i,22} \end{bmatrix}.$$

Then,  $\Psi$  has state space realization

$$\left[ \begin{array}{cc|c} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ \hline I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{array} \right],$$

and

$$\Sigma = \begin{bmatrix} \Sigma_{1,11} & 0 & \Sigma_{1,12} \\ 0 & \Sigma_{2,12} & \Sigma_{2,12} \\ \Sigma'_{1,12} & \Sigma'_{2,12} & \Sigma_{1,22} + \Sigma_{2,22} \end{bmatrix}.$$

We have  $\sigma(v, \delta v, \lambda) \geq 0$ , for all  $\lambda \in \Lambda$  and for all  $v \in \mathbf{L}_2[0, \infty)$ .

### 3.5.4 Dynamic Uncertainty

In this example we consider a LTI dynamic uncertainty  $\Delta$  which has  $\mathbf{L}_2$ -gain less than or equal to 1. This class of uncertainties satisfies the IQC with multiplier

$$\Pi = \begin{bmatrix} X & 0 \\ 0 & -X \end{bmatrix},$$

where  $X \in \mathbf{RL}_\infty$  satisfies  $X = X^* \geq 0$ . Again, a finite dimensional parameterization is obtained by the same way we discussed in Example 3.5.3. We get  $X = \Psi_0^* \Sigma_0 \Psi_0$ , where  $\Sigma_0(\lambda)$  is a symmetric matrix with coefficients defined in terms of the parameter  $\lambda$ , and  $\Psi_0$  has the form

$$\Psi_0(s) = \begin{bmatrix} (j\omega I - A)^{-1} B \\ I \end{bmatrix} \quad (3.24)$$

for some Hurwitz matrix  $A$ . The parameter vector is constrained as

$$\lambda \in \Lambda := \{\lambda : \Psi_0(j\omega)^* \Sigma_0(\lambda) \Psi_0(\omega) \geq 0\}.$$

We can, as in Example 3.5.3, find  $\Sigma(\lambda)$  and an operator  $\Psi$  with state space realization in the form of (3.12) such that  $\Pi = \Psi^* \Sigma(\lambda) \Psi$ .

### 3.6 Summary

We have presented two new techniques for robust stability and performance analysis of Linear Periodically Time-Varying (LPTV) systems with structured uncertainties. In the first approach, the harmonic terms of the nominal LPTV model are extracted, and the time-varying coefficients are lumped into the feedback loop via a simple loop transformation; the system to be analyzed is thus transformed into the setup suited for standard robustness analysis methods. Robustness analysis is then performed on the transformed system using the standard Integral Quadratic Constraint (IQC) method.

In the second approach, we allow the nominal subsystem to remain periodic. We develop a framework for stability and performance analysis which is analogous to the standard IQC method. In this approach, the uncertainties in the system are characterized by convex combinations of IQCs defined by quadratic forms which are allowed to be periodically time-varying. Conditions for robustness is formulated as infinite dimensional convex feasibility problems. This type of problems can be solved using the cutting plane method. A particular cutting plane algorithm will be developed in Chapter 5, where we will also illustrate and compare the two different approaches for robustness analysis of periodic systems by a couple of examples.





## Chapter 4

# Analysis of Periodically Forced Uncertain Feedback Systems

This chapter addresses conditions for existence and stability of stationary periodic solutions to uncertain periodic systems, as well as analysis of harmonics in these solutions. Existence of stationary periodic solutions is studied and harmonic performance analysis is carried out using Integral Quadratic Constraints (IQCs) defined on the space of square integrable periodic functions. Stability of the stationary periodic solutions is investigated using IQCs defined on the usual space of square integrable functions. Conditions for existence of a unique solution, stability, and performance are formulated as problems of checking feasibility of certain affinely parameterized operator inequalities<sup>1</sup>.

### 4.1 Introduction

In chapter 3, analysis of periodic systems is performed on an infinite time horizon. This is necessary when investigating stability but is less suitable when considering other important system properties, such as existence of stationary periodic solutions in the system and amplification of the harmonics in these solutions. In this chapter, we develop a framework, similar to the Periodic IQC Approach in the previous chapter, for analyzing these properties of an uncertain periodic system.

Several problems will be considered in this chapter. First, we consider the problem

---

<sup>1</sup>The contents of this chapter are adapted from technical report [40], which will soon appear in IEEE Transactions on Circuits and Systems I. The report is the result of a joint work of Dr. U. Jönsson at the Royal Institute of Technology, Sweden and the author of this thesis.

of existence of stationary periodic solutions to periodically forced systems. It is not at all obvious that a periodic input signal to a dynamic system is mapped to a periodic output signal. For example, an integrator does not map a periodic input to a periodic output if the input does not have zero DC value. Incremental IQCs will be used to derive an existence theorem that allows more flexible treatment of uncertain systems than the methods previously proposed in the literature [85, 21, 14].

The condition for existence of a stationary periodic solution does not reveal any information concerning stability of the solution. Here “stability“ means that the solution of the system equations converges to the stationary periodic solution even if there are small disturbances in the input signal or changes in the initial condition. We use incremental IQCs defined on the usual space (i.e.,  $\mathbf{L}_{2e}[0, \infty)$ ) for robust stability analysis. A sufficient condition for stability of stationary solutions is obtained in a fashion similar to the way by which the condition for stability of equilibrium points is derived in [55].

Finally we consider the so-called *harmonic performance analysis*. Analysis of harmonics has various important engineering applications [58]. There are many different performance criteria one can consider. One particular criterion will be addressed in this chapter: the criterion corresponds to the problem of estimating of the amplification of periodic signals in an uncertain periodic system. This problem is relevant to robustness analysis of periodic trajectories [38].

The analysis in this chapter results in optimization problems of the form

$$\inf_{\lambda} c' \lambda, \quad \text{such that } S_{\lambda} > 0 \text{ and } \lambda \in \Lambda, \quad (4.1)$$

where  $\Lambda \subset \mathbf{R}^n$  is a convex set of parameters and  $S_{\lambda}$  is a  $\lambda$ -parameterized, self-adjoint operator on the space of square integrable periodic functions. This type of problems is very similar to the one arising from the Periodic IQC Approach for robustness analysis of periodic systems. We will comment on the similarity between the two types of problems. The computational algorithms which allow us to numerically solve the type of problems arising in this chapter are almost identical to those for the optimization problems arising from the previous chapter. These computational algorithms will be developed in Chapter 5.

The chapter is organized as follows: the uncertain periodic system under consideration is described in Section 4.2. The problem of whether an uncertain periodic system subject to a

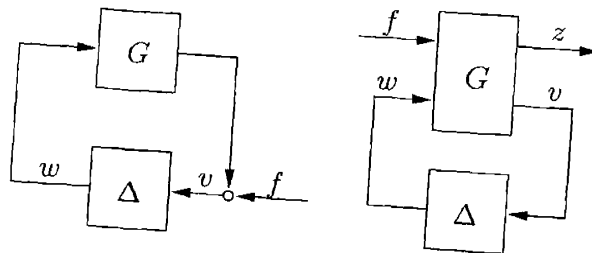


Figure 4-1: Systems for robust stability and performance analysis.

periodic input has a periodic output is discussed in Section 4.3, and the problem of whether the periodic output is stable is addressed in Section 4.4. In Section 4.5, we discuss the so-called *harmonic performance analysis* of an uncertain periodic system. The conditions of these analysis problems give rise to a class of optimization problems. We briefly discuss the idea of how the optimization problems can be solved in Section 4.6. Finally, a briefly summary of the results presented in this chapter is given in Section 4.7.

## 4.2 Systems Under Consideration

Again, we will consider the system on the left-hand side of Figure 4-1 for stability analysis and the system on the right-hand side for performance analysis. The nominal subsystem  $G$  is assumed to be a linear, bounded,  $T_0$ -periodic operator on  $L_2(T_0)$  and has a state space representation

$$\begin{aligned} \dot{x}_G &= A_G(t)x_G + B_G(t)u, \\ y &= C_G(t)x_G + D_G(t)u, \end{aligned} \quad (4.2)$$

where all matrices are assumed to be real-valued,  $T_0$ -periodic, and continuous. The transition matrix of system  $G$  is defined as the solution to the system

$$\dot{M}(t) = A_G(t)M(t), \quad M(0) = I.$$

It is well-known that if  $1 \notin \text{eig}(M(T_0))$  and if the system is initialized appropriately, the output  $y$  is  $T_0$ -periodic for any input  $u \in L(T_0)$  (See, for instance, [21]). Throughout this chapter, we assume that 1 is not an eigenvalue of the transition matrix of system  $G$ .

### 4.3 Existence and Uniqueness of Stationary Solutions

Consider the system on the left-hand-side of Figure 4-1

$$\begin{aligned} v &= Gw + f, \\ w &= \Delta(v), \end{aligned} \tag{4.3}$$

where  $\Delta$  is a Lipschitz continuous (possibly nonlinear) bounded operator on  $\mathbf{L}_2(T_0)$ . In this section, we consider the problem of whether the system has a unique  $T_0$ -periodic solution when it is subject to a  $T_0$ -periodic excitation. We will use incremental IQCs to derive a sufficient condition.

**Definition 4.1 (Incremental IQC on  $\mathbf{L}_2(T_0)$ ).** Let the quadratic form  $\sigma_\Pi : \mathbf{L}_2(T_0) \times \mathbf{L}_2(T_0) \rightarrow \mathbf{R}$  be defined as

$$\sigma_\Pi(e_1, e_2) = \left\langle \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \Pi \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \right\rangle_{\mathbf{L}_2(T_0)}, \tag{4.4}$$

where  $\Pi$  is a bounded, linear, self-adjoint operator on  $\mathbf{L}_2(T_0)$ . Then we say that  $\Delta : \mathbf{L}_2(T_0) \rightarrow \mathbf{L}_2(T_0)$  satisfies the incremental IQC defined by  $\sigma_\Pi$  if  $\sigma_\Pi(v_1 - v_2, \Delta(v_1) - \Delta(v_2)) \geq 0$  for all  $v_1, v_2 \in \mathbf{L}_2(T_0)$ .

**Example 4.1.** Consider an uncertain modulation operator  $(\Delta(v))(t) = r \cos(\omega_0 t + \theta)v(t)$ , where  $r \geq 0$ ,  $\theta \in [0, 2\pi]$ , and  $\omega = \frac{2\pi}{T_0}$ . Then  $\Delta$  satisfies the incremental IQCs defined by the  $\lambda$ -parameterized quadratic form

$$\sigma_\Pi(v, w, \lambda) := \int_0^{T_0} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix}' \Pi(t, \lambda) \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} dt,$$

where

$$\Pi(t, \lambda) = \begin{bmatrix} 0 & m(t) \\ m(t) & 0 \end{bmatrix}, \quad m(t) = \lambda_0 + \sum_{k \neq 1} \lambda_k \cos(\omega_0 k t + \theta_k), \quad \lambda_k \geq 0,$$

and  $\theta_k \in [0, 2\pi]$ .

The following theorem concerns existence and uniqueness of a stationary periodic solution of an uncertain periodic system subject to a periodic excitation.

**Theorem 4.1.** Consider system (4.3). Let  $\sigma_{\Pi}$  be a quadratic form as defined in (4.4).

Assume that

(i) for all  $\tau \in [0, 1]$ ,  $\tau\Delta$  satisfies the incremental IQC defined by  $\sigma_{\Pi}$ ;

(ii) there exists an  $\varepsilon > 0$  such that

$$\sigma_{\Pi}(Gw, w) \leq -\varepsilon\|w\|^2, \quad \forall w \in \mathbf{L}_2(T_0).$$

Then the system (4.3) has a unique solution  $(v, w) \in \mathbf{L}_2(T_0)$  for every excitation  $f \in \mathbf{L}_2(T_0)$ .

Furthermore, there exists a constant  $c > 0$  such that  $\|v\| + \|w\| \leq c\|f\|$ .

*Proof.* The proof of this theorem is given in the technical report [40]. □

Note that Theorem 4.1 does not reveal any information concerning the stability of the periodic solution  $v = (I - G\Delta)^{-1}(f)$ . In other words, if the input is perturbed by an arbitrary small function  $\delta f \in \mathbf{L}_2(-\infty, \infty)$ , then the solution  $\tilde{v} = (I - G\Delta)^{-1}(f + \delta f)$  may not converge to the stationary solution  $v$ . We will discuss the issue of stability in the next section.

## 4.4 Stability of the Stationary Solution

Assume now that there exists a unique periodic solution  $(v_0, w_0) \in \mathbf{L}_2(T_0)$  of system (4.3) for any  $f_0 \in \mathbf{L}_2(T_0)$ . Stability of such stationary solutions means that when there is perturbation on the input and/or initial conditions<sup>2</sup> of the system, the solution of the system equations converges to the stationary solution. In this section, we will derive a sufficient condition for stability of the stationary solution. Let us consider the perturbed system

$$\begin{aligned} v &= Gw + f_0 + \delta f_1, \\ w &= \Delta(v) + \delta f_2, \end{aligned} \tag{4.5}$$

where, in addition to the assumptions made in the previous section,  $G$  and  $\Delta$  are assumed to be causal, bounded, and Lipschitz continuous on  $\mathbf{L}_{2e}[0, \infty)$ . For  $G$ , this is true if the state space realization in (4.2) is exponentially stable. To ensure the exponential stability

---

<sup>2</sup>Note that the effect of perturbation on the initial conditions can be captured by introducing perturbation signals to the system.

of  $G$ , we assume that the transition matrix of the system,  $M(t)$ , satisfies the condition that all eigenvalues of  $M(T_0)$  are located strictly inside the unit disk of the complex plane.

If  $\delta f_1 = \delta f_2 = 0$ , then the system has the periodic solution  $(v_0, w_0)$ . Consider the difference between this solution and the solution  $(v, w)$  obtained when  $(\delta f_1, \delta f_2) \in \mathbf{L}_2[0, \infty)$ .

We have

$$\begin{aligned}\delta v &= G\delta w + \delta f_1, \\ \delta w &= \tilde{\Delta}(\delta v) + \delta f_2,\end{aligned}\tag{4.6}$$

where  $\tilde{\Delta}(\delta v) = \Delta(\delta v + v_0) - \Delta(v_0)$ . The stability of a stationary periodic solution is formally defined as follows.

**Definition 4.2.** A periodic solution  $(v_0, w_0)$  of system (4.3) is stable if system (4.6) is stable in the sense of Definition 3.2 (See Chapter 3). System (4.3) is uniformly stable if the system has a stable periodic solution  $(v_0, w_0)$  for every input  $f_0 \in \mathbf{L}_2(T_0)$ .

Again, we will use incremental IQCs to derive a sufficient condition for stability of a stationary periodic solution.

**Definition 4.3 (Incremental IQC on  $\mathbf{L}_2[0, \infty)$ ).** Let the quadratic form  $\sigma_\Pi : \mathbf{L}_2[0, \infty) \times \mathbf{L}_2[0, \infty) \rightarrow \mathbf{R}$  be

$$\sigma_\Pi(e_1, e_2) = \left\langle \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \Pi \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \right\rangle_{\mathbf{L}_2[0, \infty)},\tag{4.7}$$

where  $\Pi$  is a linear, bounded, and self-adjoint operator on  $\mathbf{L}_2(-\infty, \infty)$ . Then an operator  $\Delta : \mathbf{L}_{2e}[0, \infty) \rightarrow \mathbf{L}_{2e}[0, \infty)$  satisfies the incremental IQC defined by  $\sigma_\Pi$  if  $\sigma_\Pi(v_1 - v_2, \Delta(v_1) - \Delta(v_2)) \geq 0$  for all  $v_1, v_2 \in \mathbf{L}_{2e}[0, \infty)$  such that  $v_1 - v_2 \in \mathbf{L}_2[0, \infty)$ .

Notice that if  $\Delta$  satisfies an incremental IQC defined by  $\sigma_\Pi$ , then  $\tilde{\Delta}$  satisfies the IQC defined by  $\sigma_\Pi$ . The following stability theorem follows the main result in [55].

**Theorem 4.2.** Consider system (4.3). Assume that the system has a  $T_0$ -periodic solution for any input  $f \in \mathbf{L}_2(T_0)$  and that  $G$  and  $\Delta$  are causal, Lipschitz continuous on  $\mathbf{L}_{2e}[0, \infty)$ . Let  $\sigma_\Pi$  be a quadratic form as defined in (4.7). If

- (i) for all  $\tau \in [0, 1]$ ,  $\tau\Delta$  satisfies an incremental IQC defined by  $\sigma_\Pi$ ;

(ii) there exists an  $\varepsilon > 0$  such that

$$\sigma_{\Pi}(Gw, w) \leq -\varepsilon\|w\|^2, \quad \forall w \in \mathbf{L}_2[0, \infty),$$

then system (4.3) is uniformly stable.

*Proof.* Condition (i) implies that  $\tau\tilde{\Delta}$  satisfies the IQC defined by  $\sigma_{\Pi}$  for all  $\tau \in [0, 1]$ . Therefore, stability of system (4.6) follows the main result of [55]. Note that the well-posedness condition is automatically satisfied in our setup.  $\square$

It might seem that conditions (i) and (ii) in Theorem 4.2 are identical to the conditions in Theorem 4.1, and the conditions for existence of a unique solution automatically ensure the stability of the solution. However, this is not true. Notice that the strict negative definiteness in (ii) of Theorem 4.2 is different from the one in (ii) of Theorem 4.1 since the function spaces considered in Theorems 4.2 and 4.1 are different. The  $\mathbf{L}_2[0, \infty)$  case in the above theorem is the same as condition (iii) of Proposition 3.2 in Chapter 3. We will discuss how to check such a condition in Chapter 5. If  $G$  and  $\Pi$  have time-invariant state-space realizations, then the situation is even simpler. In this case, the condition in (ii) can be formulated as Linear Matrix Inequalities (LMIs).

## 4.5 Harmonic Performance Analysis

In this section, we consider performance analysis of stationary periodic solutions. Let us consider the system on the right-hand-side of Figure 4-1

$$\begin{aligned} \begin{bmatrix} z \\ v \end{bmatrix} &= G \begin{bmatrix} f \\ w \end{bmatrix}, \\ w &= \Delta(v), \end{aligned} \tag{4.8}$$

which has a special output  $z$  on which we want to perform certain performance analysis. Here  $G$  and  $\Delta$  are defined as in Section 4.2, and  $\Delta$  is assumed to have 0 offset value; i.e.,  $\Delta(0) = 0$ . Several objectives are possible. For example, the worst-case magnitude for given sets of periodic inputs can be investigated. Another example is to study the energy relation between given sets of harmonics in the output and input. This was done using

other methods in [67]. We define robust performance for the closed-loop system (4.8) as follows.

**Definition 4.4.** Let  $\Psi$  be a linear, bounded, self-adjoint operator on  $\mathbf{L}_2(T_0)$  and  $\sigma_\Psi : \mathbf{L}_2(T_0) \times \mathbf{L}_2(T_0) \rightarrow \mathbf{R}$  be the quadratic form

$$\sigma_\Psi(z, f) \equiv \left\langle \begin{bmatrix} z \\ f \end{bmatrix}, \Psi \begin{bmatrix} z \\ f \end{bmatrix} \right\rangle_{\mathbf{L}_2(T_0)}. \quad (4.9)$$

We said that system (4.8) satisfies robust performance with respect to  $\sigma_\Psi$  if

- (a) there exists a unique solution  $(z, v, w) \in \mathbf{L}_2(T_0)$  for every  $f \in \mathbf{L}_2(T_0)$ ; i.e., there exists a stationary periodic solution for all periodic inputs, and
- (b) we have  $\sigma_\Psi(z, f) \leq 0$  for all solution pairs  $(z, f) \in \mathbf{L}_2(T_0)$  of the closed-loop system.

We refer to the quadratic form  $\sigma_\Psi$  as the *performance measure*.

We have the following theorem for robust performance.

**Theorem 4.3.** Consider system (4.8). Let  $\sigma_\Psi$  be a quadratic form as defined in (4.9) and  $\sigma_\Pi$  be another quadratic form as defined in (4.4). Assume that

- (i)  $\sigma_\Psi(z, 0) \geq 0$ , for all  $z \in \mathbf{L}_2(T_0)$ ;
- (ii) for all  $\tau \in [0, 1]$ ,  $\tau\Delta$  satisfies incremental IQC defined by  $\sigma_\Pi$ ;
- (iii) there exists an  $\varepsilon > 0$  such that

$$\sigma_\Psi(z, f) + \sigma_\Pi(v, w) \leq -\varepsilon(\|f\|^2 + \|w\|^2), \quad \forall f, w \in \mathbf{L}_2(T_0),$$

where  $(z, v, f, w)$  satisfies system equations in (4.8).

Then system (4.8) satisfies robust performance  $\sigma_\Psi \leq 0$ .

*Proof.* By taking  $f = 0$ , we see conditions (i) and (iii) imply that there exists an  $\varepsilon > 0$  such that

$$\sigma_\Pi(G_{22}w, w) \leq -\varepsilon\|w\|^2, \quad \forall w \in \mathbf{L}_2(T_0).$$



This, together with condition (ii), implies that there exists a unique solution  $(z, v, w) \in \mathbf{L}_2(T_0)$  to (4.8) for every  $f \in \mathbf{L}_2(T_0)$ . Then condition (iii) implies robust performance, since for all  $w = \Delta(v)$ ,

$$\sigma_{\Pi}(v, w) = \sigma_{\Pi}(v, \Delta(v)) = \sigma_{\Pi}(v - 0, \Delta(v) - \Delta(0)) \geq 0.$$

This concludes the proof. □

### 4.5.1 Supremum Norm of Periodic Output

One problem of performance analysis was discussed in [38], where an estimate of the magnitude (supremum norm) of the output was obtained using a special performance IQC. Here we state a stronger version of that result for the case where the signal  $z$  is scalar. If signal  $z$  is vector-valued, then one can apply the proposition to each component of  $z$  to obtain an estimate of the supremum norm.

**Proposition 4.1.** *Let  $z(t) \in \mathbf{L}_2(T_0)$  be scalar function with absolutely summable Fourier series. Then*

$$\max_{t \in [0, T_0]} |z(t)| = \inf_{\Xi, \gamma_k, \xi_k} \left( \sum_{k=-\infty}^{\infty} \gamma_k \right)^{1/2} \quad \text{subject to} \quad \begin{cases} \langle z, \Xi z \rangle_{\mathbf{L}_2(T_0)} \leq 1 \\ \begin{bmatrix} \gamma_k & 1 \\ 1 & \xi_k \end{bmatrix} \geq 0 \end{cases} \quad (4.10)$$

where  $\gamma_k, \xi_k$  are real numbers, and  $\Xi$  is defined by  $(\widehat{\Xi z})_k = \xi_k \widehat{z}_k$ .

*Remark 4.1.* It is possible to obtain state space formulations of this result in a form similar to our related result in [38].

*Proof.* The absolute summability assumption implies that  $z(t)$  is uniformly continuous on  $[0, T_0]$ , and hence the maximum is achieved at some particular time  $\hat{t} \in [0, T_0]$ . Let  $\xi_k$  be

nonzero constant, and let  $\Psi_k \in \mathbf{C}$  be such that  $\xi_k = |\Psi_k|^2$ . Then we have

$$\begin{aligned} |z(\hat{t})| &= \left| \sum_{k=-\infty}^{\infty} \hat{z}_k e^{j\omega_0 k \hat{t}} \right| = \left| \sum_{k=-\infty}^{\infty} \Psi_k^{-1} \Psi_k \hat{z}_k e^{j\omega_0 k \hat{t}} \right| \leq \sqrt{\sum_{k=-\infty}^{\infty} |\Psi_k|^{-2}} \sqrt{\sum_{k=-\infty}^{\infty} \hat{z}_k^* |\Psi_k|^2 \hat{z}_k} \\ &\leq \left( \sqrt{\sum_{k=-\infty}^{\infty} \gamma_k} \right) \langle z, \Xi z \rangle_{\mathbf{L}_2(T_0)} \leq \sqrt{\sum_{k=-\infty}^{\infty} \gamma_k} \end{aligned}$$

where we used the Cauchy-Schwartz inequality to obtain the first inequality and the two constraints of the optimization problem in order to obtain the next two inequalities. This shows that the objective value of the right-hand-side of (4.10) is an upper bound for  $\sup_{t \in [0, T_0]} |z(t)|$ . To see that it is exact, let us first assume  $\hat{z}_k \neq 0$  for all  $k$ . Then the Cauchy-Schwartz inequality becomes an equality if we use  $\Psi_k^{-2} = \alpha \hat{z}_k e^{j\omega_0 k \hat{t}}$  for all  $k$  and for some  $\alpha \in \mathbf{R}$ . Now, let  $\xi_k = |\Psi_k|^2$  and  $\gamma_k = 1/\xi_k$ . Furthermore, scale  $\alpha$  such that  $\langle z, \Xi z \rangle_{\mathbf{L}_2(T_0)} = 1$ . Then the last two inequalities also become equalities, and we conclude that  $\sup_{t \in [0, T_0]} |z(t)|$  is indeed equal to the optimal objective value of the optimization problem.

If  $\hat{z}_k = 0$  then we would have a problem since  $\Psi_k^{-2} = \alpha \hat{z}_k e^{j\omega_0 k \hat{t}} = 0$  would imply that  $\xi_k = |\Psi_k|^2 = \infty$ . This problem can be overcome by considering a suitable sequence of  $\xi_k^{(n)} \nearrow \infty$  (as  $n \rightarrow \infty$ ) in order for the objective of the optimization problem to converge to the infimum.  $\square$

If the system in (4.8) is such that  $z, \dot{z} \in \mathbf{L}_2(T_0)$  for all inputs  $f \in \mathbf{L}_2(T_0)$ , then  $z$  has an absolutely summable Fourier series. Suppose that we are interested in knowing how the amplitude of  $z$  corresponds to the input signal  $f$ . We can apply Theorem 4.3 with a performance measure  $\sigma_\Omega$  where

$$\Omega = \begin{bmatrix} \Xi & 0 \\ 0 & -I \end{bmatrix}.$$

Suppose that all the conditions in Theorem 4.3 are satisfied. Then we conclude that

$$\langle z, \Xi z \rangle_{\mathbf{L}_2(T_0)} \leq \langle f, f \rangle_{\mathbf{L}_2(T_0)}.$$

As a result of Proposition 4.1, we now get a bound on the amplitude of  $z$ :  $\max_{t \in [0, T_0]} |z(t)| \leq$

$\gamma \|f\|_{L_2(T_0)}$ , where  $\gamma = (\sum_{k=-\infty}^{\infty} \gamma_k)^{1/2}$ . Furthermore, we can try to optimize over  $\Xi$  in order to get a tighter upper bound.

In the next section, we show that it is possible to parameterize IQCs and that conditions for existence of a unique solution, stability, and performance can be formulated as problems of checking feasibility of certain affinely parameterized operator inequalities.

## 4.6 Parameterizing and Optimizing IQCs

We have seen in Chapter 3 that if  $\Pi$  satisfies the IQCs defined by  $\Pi_0, \Pi_1, \dots, \Pi_n$ , then for any  $\lambda_i \geq 0, i = 1, \dots, n$ ,  $\Delta$  also satisfies the IQC defined by  $\Pi(\lambda)$ , where

$$\Pi(\lambda) = \Pi_0 + \sum_{i=1}^n \lambda_i \Pi_i.$$

The same is true for the case of incremental IQCs. Therefore, the parameterization discussed in Section 3.4.2 also applies here. Hence, following the idea discussed in Section 3.4.2, we can formulate condition (ii) in Theorem 4.1 and its counterpart in Theorem 4.2 as feasibility problems:

$$\text{find } \lambda, \quad \text{such that } \lambda \in \Lambda, \quad S_\lambda > 0. \quad (4.11)$$

and condition (iii) in Theorem 4.3 as an optimization problem:

$$\inf_{\lambda} c' \lambda, \quad \text{subj. to } \lambda \in \Lambda, \quad S_\lambda > 0. \quad (4.12)$$

Set  $\Lambda$  in (4.11) and (4.12) is a bounded and convex subset of  $\mathbf{R}^n$ . It is defined by time-dependent and/or time-independent linear matrix inequalities.  $S_\lambda$  is a  $\lambda$ -parameterized, self-adjoint operator on a particular Hilbert space. For the feasibility problems arising from condition (ii) of Theorem 4.2, the Hilbert space is of the same form as the set  $\mathcal{L}$  defined in (3.19). Therefore, the feasibility problems arising from Theorem 4.2 are of the same type as (3.21).

On the other hand, the operator  $S_\lambda$  in feasibility/optimization problems arising from Theorem 4.1 and Theorem 4.3 is defined on a Hilbert space different from the set  $\mathcal{L}$  defined

in (3.19). In these problems, the Hilbert spaces are in the form of

$$\mathcal{L}_d := \{(x, w) \mid \dot{x} = A(t)x + B(t)w, x(0) = x(T_0), w \in \mathbf{L}_2(T_0)\},$$

where matrices  $A(t)$ ,  $B(t)$ ,  $Q(t, \lambda)$ ,  $F(t, \lambda)$ , and  $R(t, \lambda)$  are continuous and  $T_0$ -periodic. Matrix  $A(t)$  is exponentially stable. Matrices  $Q(t, \lambda)$ ,  $F(t, \lambda)$  and  $R(t, \lambda)$  depend affinely on  $\lambda$ . Therefore, the constraint  $S_\lambda > 0$  in these problems is different from the one in problems arising from Theorem 4.2.

The feasibility and optimization problems resulted in this chapter can be solved using non-differentiable optimization techniques. The computational algorithm for solving these problems will be developed in Chapter 5. In Section 5.5, we demonstrate the analysis framework presented in this chapter by an example, where the problem of estimating the amplification of a periodic load disturbance in a uncertain periodic system is considered.

## 4.7 Summary

We have derived conditions for existence, stability, and various performance criteria of stationary periodic signals. A new class of integral quadratic constraints that is particularly suited for analysis of periodic signals has been introduced for the analysis. We also show that it is possible to parameterize IQCs and that conditions for existence of a unique solution, stability, and performance can be formulated as problems of checking feasibility of certain affinely parameterized operator inequalities.

In the next chapter, a computational algorithm for solving will be developed optimization problems arising from this and previous chapters.

## Chapter 5

# Cutting Plane Algorithms for Analysis of Periodic Systems

In Chapters 3 and 4, certain infinite dimensional convex optimization problems were found to arise from the analysis of uncertain periodic systems. In this chapter, we develop a cutting plane algorithm for solving these problems. Furthermore, several case studies are performed in this chapter to demonstrate and compare the analysis techniques proposed in Chapters 3 and 4.

### 5.1 Introduction

In Chapter 3, a new framework for robust stability and performance analysis of uncertain periodic systems called the *Periodic IQC Approach* was proposed. In this approach, conditions for robust stability and performance can be expressed as infinite dimensional feasibility and optimization problems of the following forms

$$\text{(F) find } \lambda, \text{ such that } S_\lambda > 0$$

$$\text{(O) } \inf_{\lambda} c'\lambda, \text{ subj. to } S_\lambda > 0.$$

where  $S_\lambda := S_0 + \sum_{i=1}^n \lambda_i S_i$ , and each  $S_i$  is a self-adjoint operator defined on a particular Hilbert space. In Chapter 4, we developed techniques for analyzing certain properties of periodically forced uncertain feedback systems. Conditions that ensure the system has certain desired properties are also expressed as feasibility and/or optimization problems in

the forms of (F) and (O), except that the  $\lambda$ -parameterized operator  $S_\lambda$  is defined on a different Hilbert space. In this chapter, we discuss how these problems can be solved using a non-differentiable optimization technique called the cutting plane method.

In the cutting plane method, a mechanism that determines whether a given point is feasible to the constraint of the problem to be solved is required. Furthermore, if the point is infeasible, the mechanism needs to be able to generate at least one hyperplane which separates the point from the feasible set. Such a mechanism is often called the *oracle*. Given a sequence of trial points, the oracle produces hyperplanes which are used to construct a sequence of outer approximations of the original problem. Every time a trial point is tested, the outer approximation is refined. As more points are tested, the outer approximation becomes successively closer to the original problem.

In this chapter, we develop a cutting plane algorithm to solve the optimization problems arising from Chapters 3 and 4. The algorithm we propose follows the basic principles of Kelley's cutting plane algorithm [47, 11]. There are, however, some slight differences between our cutting plane algorithm and Kelley's. In particular, in our algorithm, the constrained optimization problem (O) is solved via two unconstrained spectral value maximization problems. This is different from Kelley's algorithm for solving constrained convex optimization problems. Furthermore, evaluation of the function to be optimized and computation of a subgradient of the function at each trial point - the two basic steps of the Kelley's cutting plane algorithm - are replaced by slightly different procedures. We will specifically address these differences when our algorithm is presented. The essential part of the cutting plane algorithm is the oracle. For the optimization problems arising from Chapters 3 and 4, the oracles can be constructed using existing results. We will use results by Yakubovich [79, 80, 81] to construct the oracle for the optimization problems arising from Chapter 3. For the optimization problems arising from Chapter 4, the oracle is constructed based on a frequency theorem developed in [40].

Most of the material presented in this chapter is either known, or a direct consequence of well-established results. The contributions of the work here are more at the implementation level: to show how the infinite dimensional optimization problems arising from Chapters 3 and 4 can be solved using known results, to develop computational procedures using these results, and to construct a usable software for robustness analysis of uncertain periodic systems. This is important work for practical engineering purposes; after all, the analysis

frameworks presented in Chapters 3 and 4 would be useless if there is no practical way to check conditions for robustness. Furthermore, we also conduct several case studies to demonstrate and compare the analysis techniques proposed in Chapters 3 and 4.

The remainder of this chapter is organized as follows: in section 5.2, following a brief introduction of Kelley’s cutting plane algorithm for solving general convex optimization problems, an algorithm slightly different from Kelley’s is developed to solve the problems arising from Chapter 3 and Chapter 4. In this section, we also address the differences between Kelley’s algorithm and our algorithm. In Section 5.3, we present the oracle for optimization problems arising from Chapter 3, while the oracle for optimization problems arising from Chapter 4 is presented in Section 5.4. Several examples are presented in Section 5.5 to demonstrate and compare the techniques proposed in Chapters 3 and 4 for robustness analysis of periodic systems. Finally, a technical proof is presented in Section 5.6, and we draw some concluding remarks in Section 5.7.

## 5.2 A Kelley Type Cutting Plane Algorithm

In this section, we present a Kelley type cutting plane algorithm for solving feasibility problem (F) and optimization problem (O). The algorithm follows the basic principles of Kelley’s cutting plane algorithm [47] but is slightly different from Kelley’s. We start with a brief introduction of Kelley’s cutting plane algorithm. The material presented in the next subsection is adapted from [11].

### 5.2.1 Kelley’s Cutting Plane Algorithm

Given a concave function  $\phi(x) : \mathbf{R}^n \rightarrow \mathbf{R}$ , we say that  $g \in \mathbf{R}^n$  is a subgradient of  $\phi$  at  $x$  if  $\phi(z) \leq \phi(x) + g'(z - x)$ ,  $\forall z$ . We denote the set of subgradients of  $\phi$  at  $x$  by  $\partial\phi(x)$ . Now, let us consider the unconstrained maximization problem

$$\max_x \phi(x). \tag{5.1}$$

Suppose that we have computed function values and at least one subgradient at  $x_1, \dots, x_k$ :

$$\phi(x_1), \dots, \phi(x_k), \quad g_1 \in \partial\phi(x_1), \dots, g_k \in \partial\phi(x_k).$$

We then have

$$\phi(z) \leq \phi(x_i) + g'_i(z - x_i), \quad \forall z, i = 1, \dots, k,$$

and hence

$$\phi(z) \leq \phi_k^{ub}(z) := \min_{1 \leq i \leq k} (\phi(x_i) + g'_i(z - x_i)).$$

Here  $\phi_k^{ub}(z)$  is a piece-wise linear concave function that is everywhere greater than or equal to  $\phi(z)$ . Let  $\phi^*$  denote the optimal objective of (5.1). We can now get an upper bound of  $\phi^*$  by solving

$$\max_z \phi_k^{ub}(z).$$

This maximization problem can be solved via linear programming

$$\max_{U, z} U, \quad \text{subj. to } U \leq \phi(x_i) + g'_i(z - x_i), \quad i = 1, \dots, k.$$

Let  $U_k$  be the optimal objective of the above linear program. The function  $\phi_k^{ub}(z)$  can be unbounded above so that  $U_k = \infty$ . This is not a useful bound. We can avoid this situation by explicitly specifying bounds on the variables. In other words, we change the original unconstrained problem to

$$\max_{x_{min} \leq x \leq x_{max}} \phi(x).$$

In this case, we have the upper bound

$$U_k := \max_{U, z} U, \quad \text{subj. to } \begin{cases} U \leq \phi(x_i) + g'_i(z - x_i), & i = 1, \dots, k \\ x_{min} \leq z \leq x_{max} \end{cases} \quad (5.2)$$

In practice, any computer system can only deal with numbers up to a certain digit, so we can let  $(x_{min}, x_{max}) = (-B, B)$  where  $B$  is the maximum number that a computer system can have. Thus, the additional constraint  $x_{min} \leq x \leq x_{max}$  is justifiable and not practically restrictive.



Finally, a lower bound of  $\phi^*$  can be found by solving

$$L_k := \max\{\phi(x_1), \dots, \phi(x_k)\}, \quad (5.3)$$

which is the maximum objective value obtained so far. If an objective value and a subgradient are evaluated at another point  $x_{k+1}$ , the new lower and upper bounds  $L_{k+1}$  and  $U_{k+1}$  are improved:

$$L_k \leq L_{k+1} \leq \phi^* \leq U_{k+1} \leq U_k.$$

Kelley's cutting plane algorithm is built based on the idea described above. The algorithm goes as follows.

**Initialization:** let the initial box that contains the minimizer be  $B := [x_{min}, x_{max}]^n$ .

Let  $x_1 \in B$  and  $k = 1$ . Compute  $\phi(x_1)$  and any  $g_1 \in \partial\phi(x_1)$ .

**Repeat**

- (a) Solve the linear program in (5.2) to find  $U_k$ . Let the maximizer of the linear program be  $x_k^*$ .
- (b) Compute  $L_k$  according to (5.3).
- (c) Set  $x_{k+1} := x_k^*$ .
- (d) Compute  $\phi(x_{k+1})$  and any  $g_{k+1} \in \partial\phi(x_{k+1})$ .
- (e) Set  $k := k + 1$ .

**Until**  $U_k - L_k \leq \epsilon$ .

It can be shown that Kelley's cutting plane algorithm converges. The proof of convergence can be found in [47, 11]. The above algorithm can be modified to handle constrained optimization problems. See Chapter 14 of [11] for the details.

## 5.2.2 The Equivalent Eigenvalue Maximization Problem

In this section, we show that feasibility problem (F) and optimization problem (O) can be solved via spectral value maximization problems. Because of this, both feasibility and optimization problems can be solved using the same algorithm which we are going to present.

It is obvious that the feasibility problem (F) has a solution if and only if the objective value of the spectral value maximization problem

$$\sup_{\lambda, y} y \quad \text{subj to} \quad \begin{cases} S_\lambda - yI > 0 \\ y < 1 \end{cases} \quad (5.4)$$

is strictly positive, where  $I$  denotes the identity operator of the Hilbert space which the  $S_\lambda$  is defined on. Any suboptimal solution  $(\tilde{\lambda}, \tilde{y})$  with  $\tilde{y} > 0$  gives a feasible solution (i.e.,  $\tilde{\lambda}$ ) to problem (F). On the other hand, if the optimal objective of (5.4) is less than or equal to zero, then problem (F) does not have any feasible solution. Therefore, problem (F) is equivalent to problem (5.4).

It turns out that one can also solve optimization problem (O) via solving spectral value maximization problems: first, one tries to find a point  $\eta^{(0)} \in \mathbf{R}^n$  such that  $S_{\eta^{(0)}} > 0$ . Such a point can be found by solving (5.4). If a feasible solution  $\eta^{(0)}$  is found, then one solves the optimization problem

$$\sup_{\eta, y} y \quad \text{subj to} \quad \begin{cases} S_\eta - yS_{\eta^{(0)}} > 0 \\ c'(\eta^{(0)} - \eta) - y > 0 \quad y \in [0, 1] \end{cases} \quad (5.5)$$

The next theorem shows that any suboptimal solution of problem (5.5) can be used to construct a suboptimal solution of problem (O) in such a way that convergence to the true optimum is guaranteed.

**Theorem 5.1.** *Let  $y_{opt}$  denote the optimal objective of (5.5). If  $(\eta^i, y^i)$  is a feasible solution of problem (5.5) for each  $i$  such that  $y^i \rightarrow y_{opt}$ , then*

$$\lambda^i := \frac{1}{1 - y^i} (\eta^i - y^i \eta^{(0)}), \quad (5.6)$$

are feasible solutions of problem (O) and

$$c' \lambda^i \rightarrow c' \eta^{(0)} - \frac{y_{opt}}{1 - y_{opt}} = \inf_{S_\lambda > 0} c' \lambda. \quad (5.7)$$

Note that if  $y_{opt} = 1$ , then the optimal objective of problem (O) is unbounded.

*Proof.* Suppose  $(\eta^i, y^i)$  are feasible solutions of problem (5.5) such that  $y^i \rightarrow y_{opt}$ . Let

$\lambda^i = \frac{1}{1-y^i}(\eta^i - y^i\eta^{(0)})$ . Then we have

$$S_{\lambda^i} = \frac{1}{1-y^i}(S_{\eta^i} - y^i S_{\eta^{(0)}}) > 0,$$

which implies that  $\lambda^i$  is a feasible solution of problem (O). Here we used the fact that the parameterized operator  $S$  depends affinely on the parameter (Recall:  $S_\lambda$  is of the form  $S_0 + \sum_{i=1}^n \lambda_i S_i$ , where  $\lambda_i$  denotes the  $i^{\text{th}}$  component of the parameter  $\lambda$ ). Furthermore, we have

$$c'\lambda^i = c'\eta^{(0)} + \frac{1}{1-y^i}c'(\eta^i - \eta^{(0)}) < c'\eta^{(0)} - \frac{y^i}{1-y^i},$$

i.e., the objective of problem (O) at  $\lambda^i$  is smaller than the objective at  $\eta^{(0)}$  by at least  $\frac{y^i}{1-y^i}$ . Now, We will prove the last equality of (5.7) by contradiction. Suppose

$$\inf_{S_\lambda > 0} c'\lambda < c'\eta^{(0)} - \frac{y_{opt}}{1-y_{opt}}.$$

Then there exist a  $\tilde{y}$  and a feasible solution  $\tilde{\lambda}$  of problem (O) such that

$$c'\tilde{\lambda} < c'\eta^{(0)} - \frac{\tilde{y}}{1-\tilde{y}} < c'\eta^{(0)} - \frac{y_{opt}}{1-y_{opt}}. \quad (5.8)$$

Since  $\frac{y}{1-y}$  is monotonically increasing, we have  $y_{opt} < \tilde{y} < 1$ . Now, let  $\bar{\eta} = (1-\tilde{y})\tilde{\lambda} + \tilde{y}\eta^{(0)}$ . By (5.8) and the fact that the parameterized operator  $S$  depends affinely on the parameter, we have

$$\begin{aligned} S_{\bar{\eta}} - \tilde{y}S_{\eta^{(0)}} &= (1-\tilde{y})S_{\tilde{\lambda}} > 0, \\ c'(\eta^{(0)} - \bar{\eta}) - \tilde{y} &= (1-\tilde{y})(c'(\eta^{(0)} - \tilde{\lambda}) - \tilde{y}/(1-\tilde{y})) > 0, \end{aligned}$$

which imply that  $(\bar{\eta}, \tilde{y})$  is a feasible solution of problem (5.5). However, this is a contradiction since  $\tilde{y} > y_{opt}$ . Hence, we have

$$c'\lambda^i < c'\eta^{(0)} - \frac{y^i}{1-y^i} \rightarrow c'\eta^{(0)} - \frac{y_{opt}}{1-y_{opt}} = \inf_{S_\lambda > 0} c'\lambda < c'\lambda_i,$$

which concludes the proof. □

Note that the optimization problems (5.4) and (5.5) are of the same type. Indeed, the constraint  $S_\eta - yS_{\eta^{(0)}} > 0$  can equivalently be formulated as  $\tilde{S}_\eta - yI > 0$  where  $\tilde{S}_\eta = S_0^{-1/2}S_\eta S_0^{-1/2}$  and  $S_0 = S_{\eta^{(0)}}$ . If we let  $\hat{S}_\eta = \text{diag}(\tilde{S}_\eta, c(\eta^{(0)} - \eta))$ , then the constrained optimization problem (5.5) can be formulated as

$$\sup_{\eta, y} y \quad \text{subj to} \quad \begin{cases} \hat{S}_\eta - yI > 0 \\ 0 \leq y < 1 \end{cases}.$$

Therefore, all we need for solving feasibility problem **(F)** and optimization problem **(O)** is an algorithm for solving the spectral value maximization problems of the form in (5.4).

### 5.2.3 A Kelley Type Cutting Plane Algorithm

Consider the optimization problem (5.4). Let  $q(\lambda)$  be the function defined as  $q(\lambda) = \sup \{y : S_\lambda - yI > 0, y < 1\}$ . Then  $q(\lambda)$  is a concave function that defines the boundary of the feasible set of problem (5.4). It is clear that the optimization problem (5.4) is equivalent to the unconstrained optimization problem

$$y_{opt} = \sup_{\lambda} q(\lambda). \tag{5.9}$$

We can use Kelley's cutting plane algorithm described in Section 5.2.1 to solve (5.9) as long as we know how to evaluate and compute one of the subgradients of the function at any given  $\lambda$ . Since  $q(\lambda)$  is defined in terms of spectral values of an infinite dimensional operator, it has no analytical form, and therefore, computing the exact value and subgradients of  $q(\lambda)$  at a given  $\lambda$  is a difficult task. Thus, we slightly modify the algorithm.

Our algorithm starts with computing a lower bound  $y_l$  of the optimal objective of (5.4). A lower bound  $y_l$  can be obtained by fixing a particular  $\lambda^{(0)}$  and searching for a  $y^{(0)}$  such that  $S_{\lambda^{(0)}} - y^{(0)}I > 0$  is satisfied. We will discuss how the inequality  $S_{\lambda^{(0)}} - y^{(0)}I > 0$  can be checked in Section 5.3. For now, let us assume that there exists a mechanism called *oracle* which can be used to check whether  $S_{\lambda^{(k)}} - y^{(k)}I > 0$  is satisfied for a given  $(\lambda^{(k)}, y^{(k)})$ . Furthermore, if  $S_{\lambda^{(k)}} - y^{(k)}I > 0$  is not satisfied, then the oracle returns at least one hyperplane  $a\lambda - b - y = 0$  such that  $(\lambda^{(k)}, y^{(k)})$  satisfies  $a\lambda_k - b - y_k \leq 0$  and all  $(\lambda, y)$  which satisfy  $S_\lambda - yI > 0$  also satisfy  $a\lambda - b - y > 0$ . Let  $a_i\lambda - b_i - y = 0, i = 1, \dots, k$ ,

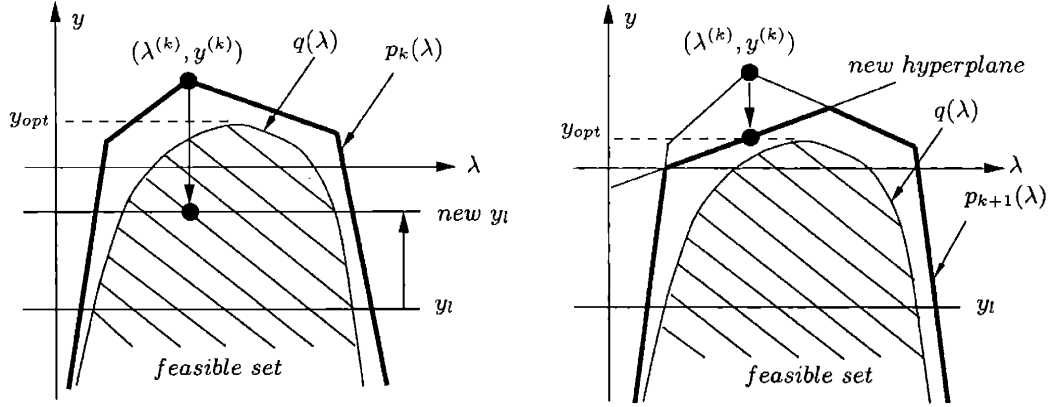


Figure 5-1: The idea of the computational algorithm is to generate a sequence of piece-wise linear functions that approximate  $q(\lambda)$  around its maxima. Figure on the left-hand-side: the test point satisfies  $S_\lambda > yI$ , and an improved lower bound of  $y_{opt}$  is obtained. Figure on the right-hand-side: the test point does not satisfy  $S_\lambda > yI$ , and a hyperplane is returned by the oracle to improve the piece-wise linear approximation of  $q(\lambda)$ . Eventually, the maximum values of these piece-wise linear functions converge to  $y_{opt}$ .

be the hyperplanes returned by the oracle. Then, these hyperplanes provide a piece-wise linear function which upper bounds  $q(\lambda)$ ; i.e.,

$$q(\lambda) \leq p_k(\lambda) := \min_{1 \leq i \leq k} \{a_i \lambda - b_i\}.$$

At the  $k^{th}$  iteration of our algorithm, the linear program

$$\max_{\lambda_{min} \leq \lambda \leq \lambda_{max}} p_k(\lambda) \tag{5.10}$$

is solved to obtain an upper bound  $y_u^{(k)}$  of  $y_{opt}$ . However, instead of evaluating  $q(\lambda^{(k)})$  and computing a subgradient of  $q(\lambda)$  at  $\lambda^{(k)}$  (Here  $\lambda^{(k)}$  denotes the maximizer of (5.10)) - an important step in the standard Kelley's cutting plane algorithm - our algorithm does the following: the algorithm computes  $\bar{y}^{(k)} = \alpha y^{(k)} + (1 - \alpha)y_l$ , where  $\alpha \in (0, 1)$ . Then, the algorithm calls the oracle to check whether  $S_{\lambda^{(k)}} - \bar{y}^{(k)}I$  is positive definite. If it is, a new lower bound of  $y_{opt}$  is obtained and  $y_l$  is updated to be  $\bar{y}^{(k)}$ . If not, then the oracle returns at least one hyperplane which can be used to improve the piece-wise linear approximation of  $q(\lambda)$ . Figure 5-1 illustrates the idea of our cutting plane algorithm, and the algorithm is summarized as follows:

1. Initialization: set the error tolerance  $\varepsilon$  and a constant  $\alpha \in (0, 1)$ . Find  $y_l$ , a lower bound of the optimal objective of (5.4). Set  $\mathcal{P}_0 = \{(\lambda, y) \mid \lambda \in [\lambda_{min}, \lambda_{max}]^n, y \leq 1\}$ .
2. At the  $k^{th}$  iteration, solve linear program  $y^{(k)} = \max \{y : (\lambda, y) \in \mathcal{P}_k\}$ .
3. Let  $\bar{y}^{(k)} = \alpha y^{(k)} + (1 - \alpha)y_l$  and check whether  $S_{\lambda^{(k)}} - \bar{y}^{(k)}I > 0$ . If the inequality is satisfied, then update the lower bound  $y_l$  to be  $\bar{y}^{(k)}$ . If not, then derive a set of linear constraints  $a'_i \lambda - b_i \geq y$ ,  $i = 1, \dots, l$  to improve the piece-wise linear approximation of  $q(\lambda)$ . Let  $\mathcal{P}_{k+1} = \mathcal{P}_k \cap \mathcal{C}_k$ , where  $\mathcal{C}_k := \{(\lambda, y) : a'_i \lambda - b_i \geq y, i = 1, \dots, l\}$ . Repeat from Step 2.
4. Stopping criterion: check whether the difference between the lower bound  $y_l$  and the current objective  $y^{(k)}$  is less than  $\varepsilon$ ; i.e., check whether  $|y^{(k)} - y_l| < \varepsilon$ . If it is true, then stop the program. Otherwise, repeat from Step 3.

Convergence of the algorithm described above can be proven. The proof follows the same arguments for proving convergence of Kelley's cutting plane algorithm [11].

Before we end this section, we briefly discuss how the cutting plane algorithm should be adapted to solve problem (5.5). With  $\lambda$  replaced by  $\eta$ , we need to make the following changes:

1. The lower bound is  $y_l = 0$ .
2. The inequality  $S_{\eta^{(k)}} - y^{(k)}S_{\eta^{(0)}} > 0$  should be verified in each iteration, and the computation of new hyperplanes should be based on this inequality.
3. After the algorithm stops, a suboptimal objective of the original problem, i.e., optimization problem (O), can be obtained by

$$c' \eta^{(0)} - \frac{y_s}{1 - y_s},$$

where  $y_s$  is the solution of problem (5.5) generated by the algorithm.

The differences between Kelley's cutting plane algorithm and the algorithm proposed in this chapter are summarized as follows.

1. Denote the optimal objective and argument of maximum of (5.10) by  $y^{(k)}$  and  $\lambda^{(k)}$ , respectively. In each iteration, the algorithm proposed in this chapter checks whether

$S_{\lambda^{(k)}} - \bar{y}^{(k)}I$  is positive definite, where  $\bar{y}^{(k)} = \alpha y^{(k)} + (1 - \alpha)y_l$  and  $\alpha \in (0, 1)$ . Kelley's cutting plane algorithm would compute the value and a subgradient of  $q(\lambda)$  at  $\lambda^{(k)}$ .

2. In the algorithm proposed in this chapter, the optimization problem **(O)** is solved in two steps via two unconstrained spectral value maximization problems. This is different from the way by which Kelley's cutting plane algorithm handles constrained optimization problems [11].

The most essential part in the algorithm is to check whether the operator inequality  $S_\lambda > yI$  holds for a given  $(\lambda, y)$ . In Section 5.3 and Section 5.4, we will show how to check the operator inequalities in the problems arising from Chapters 3 and 4, respectively.

### 5.3 The Oracle for Problems Arising from Chapter 3

Recall Chapter 3: at a given  $(\lambda, y)$ , the operator inequality  $S_\lambda > yI$  holds if and only if

1. Linear matrix inequality

$$E_1(\lambda, y) := E_{10} + \sum_{k=1}^n \lambda_k E_{1k} - yI > 0 \quad (5.11)$$

holds, where  $E_{1k}$  are real symmetric matrices for  $k = 0, 1, \dots, n$ .

2. Linear time-dependent matrix inequality

$$E_2(t, \lambda, y) := E_{20}(t) + \sum_{k=1}^n \lambda_k E_{2k}(t) - yI > 0 \quad (5.12)$$

holds for all  $t \in [0, T_0]$ , where  $E_{2k}$  are  $T_0$ -periodic, continuous, real symmetric matrix-valued functions for  $k = 0, 1, \dots, n$ .

3. The following integral inequality is satisfied:

there exists an  $\varepsilon > 0$ , such that

$$\int_0^\infty x'Q(t, \lambda, y)x + 2x'F(t, \lambda)w + w'R(t, \lambda, y)w dt \geq \varepsilon \int_0^\infty \|x\|^2 + \|w\|^2 dt, \quad (5.13)$$

$$\forall (x, w) \in \mathcal{L}_1,$$

where  $\mathcal{L}_1 := \{(x, w) \mid \dot{x} = A(t)x + B(t)w, x(0) = 0, w \in \mathbf{L}_2[0, \infty)\}$ . Matrices  $A(t)$ ,  $B(t)$ ,  $Q(t, \lambda, y)$ ,  $F(t, \lambda)$ , and  $R(t, \lambda, y)$  are continuous and  $T_0$ -periodically time-varying. Matrix  $A(t)$  is exponentially stable. Matrices  $Q(t, \lambda, y)$ ,  $F(t, \lambda)$  and  $R(t, \lambda, y)$  depend affinely on  $\lambda$  and  $y$ :  $Q(t, \lambda, y) := Q_0(t) + \sum_{k=1}^n \lambda_k Q_k(t) - yI$ ,  $F(t, \lambda) := F_0(t) + \sum_{k=1}^n \lambda_k F_k(t)$ ,  $R(t, \lambda, y) := R_0(t) + \sum_{k=1}^n \lambda_k R_k(t) - yI$ .

In this section, we will discuss how to check whether a given pair  $(\tilde{\lambda}, \tilde{y})$  satisfies inequalities (5.11), (5.12), (5.13) and, if not, how to derive hyperplanes which separate  $(\tilde{\lambda}, \tilde{y})$  from the set of  $(\lambda, y)$  pairs that satisfy  $S_\lambda > yI$ . Let  $\Omega$  denote this set.

### 5.3.1 Verification of Linear Matrix Inequalities

Given  $(\tilde{\lambda}, \tilde{y})$ , to check whether a  $T_0$ -periodic matrix  $E_2(t, \tilde{\lambda}, \tilde{y})$  is positive definite for all  $t \in [0, T_0]$ , we compute the eigenvalues of  $E_2(t, \tilde{\lambda}, \tilde{y})$  for all  $t \in [0, T_0]$  and check whether all eigenvalues are positive. If there exists  $t_0 \in [0, T_0]$  such that  $E_2(t, \tilde{\lambda}, \tilde{y})$  is not positive definite, then there exists at least one vector  $v$  with  $\|v\| = 1$  such that

$$v' E_2(t_0, \tilde{\lambda}, \tilde{y}) v \leq 0.$$

Since  $E_2(t, \lambda, y)$  depends affinely on  $(\lambda, y)$ , we have  $v' E_2(t_0, \lambda, y) v = a' \lambda - cy - b$  for some vector  $a$  and numbers  $b, c$ . More specifically,  $c = \|v\| = 1$ ,  $b = -v' E_{20}(t_0) v$  and  $a_i = v' E_{2i}(t_0) v$ , where  $a_i$  denotes the  $i^{\text{th}}$  component of  $a$ . Since any feasible point has to satisfy  $a' \lambda - b - y > 0$ , therefore  $a' \lambda - b - y = 0$  is one of the hyperplanes that separates  $(\tilde{\lambda}, \tilde{y})$  from the feasible set  $\Omega$ . Verification of positive definiteness of the time independent matrix  $E_1(\tilde{\lambda}, \tilde{y})$  is performed analogously.

### 5.3.2 Verification of the Integral Inequality

In this subsection, we will discuss how to check whether (5.13) holds for a given  $(\tilde{\lambda}, \tilde{y})$  and how to derive separating hyperplanes if it does not. We will use a “frequency theorem“ by Yakubovich [79]. To introduce the theorem, we start with defining several terminologies.

**Definition 5.1.** A system  $\dot{x}(t) = A_0(t)x(t) + B_0(t)w(t)$ ,  $x(0) = x_0$  is said to be  $\mathbf{L}_2$ -stabilizable if for any vector  $x_0 \in \mathbf{R}^n$ , there exists a function  $w(t) \in \mathbf{L}_2[0, \infty)$  such that the solution  $x(t)$  of the system is in  $\mathbf{L}_2[0, \infty)$ .



Note that in Section 3.4.2, matrices  $A_G(t)$  and  $A_\Pi(t)$  are assumed to be exponentially stable, and therefore so is  $A(t)$ . Thus the system  $\dot{x} = A(t)x + B(t)w$  in the definition of  $\mathcal{L}_1$  is  $L_2$ -stabilizable.

**Definition 5.2.** The *Hamiltonian* system corresponding to (5.13) is defined as<sup>1</sup>

$$\begin{aligned} \dot{z}(t) &= \mathcal{H}(\lambda)z(t), \quad \text{where} & (5.14) \\ z(t) &= \begin{bmatrix} x(t) \\ \psi(t) \end{bmatrix}, \quad \mathcal{H}(\lambda) = \begin{bmatrix} A - BR(\lambda, y)^{-1}F(\lambda)' & BR(\lambda, y)^{-1}B' \\ Q(\lambda, y) - F(\lambda)R(\lambda, y)^{-1}F(\lambda)' & -A' + F(\lambda)R(\lambda, y)^{-1}B' \end{bmatrix}. \end{aligned}$$

Let  $n_x$  denote the dimension of  $A$ . Let  $M(t)$  be the evolution matrix of the system (5.14):  $\dot{M}(t) = \mathcal{H}(\lambda)M(t)$ , with  $M(0) = I$ . Note that  $M(t)$  is  $2n_x \times 2n_x$ . To emphasize the  $(\lambda, y)$  dependence of  $M(t)$ , we sometimes write  $M(t, \lambda, y)$ . It can be shown that  $M(t)$  is symplectic; i.e.,

$$M(t)^T J M(t) = J, \quad \text{where } J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}.$$

This means that the eigenvalues of  $M(T_0)$  will be symmetric with respect to the unit circle. Hence, if  $M(T_0)$  satisfies the condition

$$\det(M(T_0) - e^{i\omega}I) \neq 0, \quad \forall \omega \in [0, 2\pi), \quad (5.15)$$

then there are  $n_x$  eigenvalues outside the unit circle, and there are  $n_x$  eigenvalues inside. As a result, there exists a set of  $n_x$  independent real vectors,

$$Z_s = [z_1, z_2, \dots, z_{n_x}], \quad (5.16)$$

which form a basis for the stable subspace of  $M(T_0)$ . A stable solution basis of  $z(t)$  and  $x(t)$  in (5.14) can now be defined as

$$Z(t) = M(t)Z_s, \quad (5.17)$$

$$X(t) = \begin{bmatrix} I & 0 \end{bmatrix} M(t)Z_s. \quad (5.18)$$

---

<sup>1</sup>To simplify expression, the time dependence of  $A, B, Q, F, R, \mathcal{H}$  is suppressed.

Yakubovich showed that (5.13) is equivalent to several conditions. We will use two of those conditions in our algorithm. The following theorem follows the main result in Yakubovich's paper [79].

**Theorem 5.2.** Consider (5.13). For a given  $(\tilde{\lambda}, \tilde{y})$ , let  ${}^2 \tilde{Q} = Q(\tilde{\lambda}, \tilde{y})$ ,  $\tilde{F} = F(\tilde{\lambda})$ , and  $\tilde{R} = R(\tilde{\lambda}, \tilde{y})$ . Assume that  $(A(t), B(t))$  is  $\mathbf{L}_2$ -stabilizable. Then the following statements are equivalent:

(a) There exists an  $\varepsilon > 0$ , such that

$$\int_0^\infty x' \tilde{Q} x + 2x' \tilde{F} w + w' \tilde{R} w dt \geq \varepsilon \int_0^\infty \|x\|^2 + \|w\|^2 dt, \quad \forall (x, w) \in \mathcal{L}_1. \quad (5.19)$$

(b) There exists an  $\varepsilon > 0$  such that for any non-zero complex-valued vector functions  $x(t) \in \mathbf{L}_2(T_0)$ ,  $w(t) \in \mathbf{L}_2(T_0)$ , and constant  $\omega \in [0, 2\pi)$  which satisfy

$$\dot{x}(t) = A(t)x(t) + B(t)w(t), \quad x(T_0) = e^{i\omega} x(0),$$

we have the inequality

$$\int_0^{T_0} x^* \tilde{Q} x + 2x^* \tilde{F} w + w^* \tilde{R} w dt \geq \varepsilon \int_0^{T_0} \|x\|^2 + \|w\|^2 dt. \quad (5.20)$$

(c) The Hamiltonian system (5.14) satisfies

$$(ci) \quad \tilde{R} > 0, \quad \forall t \in [0, T_0],$$

$$(cii) \quad \det(M(T_0) - e^{i\omega} I) \neq 0, \quad \forall \omega \in [0, 2\pi),$$

$$(ciii) \quad \det(X(t)) \neq 0, \quad \forall t \in [0, T_0].$$

□

*Proof.* See [79].

Using Theorem 5.2, we can verify (5.13) by first checking whether  $\tilde{R} > 0, \forall t \in [0, T_0]$ . This is performed in exactly the same way discussed in Section 5.3.1. Then we compute the evolution matrix  $M(T_0, \tilde{\lambda})$  of Hamiltonian system (5.14) and check conditions (cii) and (ciii). If either condition (cii) or (ciii) is not satisfied, separating hyperplanes can be derived according to the following propositions.

---

<sup>2</sup>Again, the time dependency of  $Q, F, R$ , and  $\tilde{Q}, \tilde{F}, \tilde{R}$  are suppressed for the sake of simplifying the notation.

**Proposition 5.1.** *Suppose that  $M(T_0, \tilde{\lambda}, \tilde{y})$  has eigenvalues on the unit circle. Then, there exists a pair of functions  $x_c(t) \in \mathbf{L}_2(T_0)$ ,  $w_c(t) \in \mathbf{L}_2(T_0)$  such that*

$$\int_0^{T_0} x_c^* \tilde{Q} x_c + 2x_c^* \tilde{F} w_c + w_c^* \tilde{R} w_c dt = 0.$$

*Proof.* Let  $z \neq 0$  and  $\omega \in [0, 2\pi)$  be such that  $M(T_0, \tilde{\lambda})z = e^{i\omega}z$ . Define

$$\begin{aligned} x_c(t) &= \begin{bmatrix} I & 0 \end{bmatrix} M(t, \tilde{\lambda})z, \\ \psi_c(t) &= \begin{bmatrix} 0 & I \end{bmatrix} M(t, \tilde{\lambda})z, \\ w_c(t) &= \tilde{R}^{-1}(B'\psi_c(t) - \tilde{F}'x_c(t)). \end{aligned}$$

Then it can be readily verified that

$$x_c^* \tilde{Q} x_c + 2x_c^* \tilde{F} w_c + w_c^* \tilde{R} w_c = x_c^* (\tilde{Q} - \tilde{F} \tilde{R}^{-1} \tilde{F}') x_c + \psi_c^* B \tilde{R}^{-1} B' \psi_c = x_c^* \dot{\psi}_c + \dot{x}_c^* \psi_c,$$

which in turn implies

$$\begin{aligned} \int_0^{T_0} x_c^* \tilde{Q} x_c + 2x_c^* \tilde{F} w_c + w_c^* \tilde{R} w_c dt &= \int_0^{T_0} x_c^* \dot{\psi}_c + \dot{x}_c^* \psi_c dt \\ &= x_c(T_0)^* \psi_c(T_0) - x_c(0)^* \psi_c(0) = (|e^{i\omega}|^2 - 1)x_c(0)^* \psi_c(0) = 0. \end{aligned} \tag{5.21}$$

This concludes the proof. □

Hence, violation of condition (ciii) leads to a pair of non-zero functions that violate inequality (5.20). Therefore, statement (b) in Theorem 5.2 is not true, and neither is (5.13). Since  $Q(t, \lambda, y)$ ,  $F(t, \lambda)$ , and  $R(t, \lambda, y)$  depend affinely on  $(\lambda, y)$ , we have

$$\int_0^{T_0} x_c^* Q(t, \lambda, y) x_c + 2x_c^* F(t, \lambda) w_c + w_c^* R(t, \lambda, y) w_c dt = a'\lambda - b - cy,$$

for some vector  $a$  and scalars  $b, c$ . Any point in the feasible set  $\Omega$  must satisfy  $a'\lambda - b - cy > 0$ ; therefore, the linear inequality  $a'\lambda - b - cy > 0$  should be used to exclude  $(\tilde{\lambda}, \tilde{y})$  from the feasible set  $\Omega$ . Note that there may be more than one eigenvalue of  $M(T_0)$  on the unit circle. All of them can be used to derive constraints. We next consider the case where condition (ciii) is violated.

**Proposition 5.2.** Suppose that <sup>3</sup>  $X(t_0, \tilde{\lambda}, \tilde{y})$  defined in (5.18) is singular for some  $t_0 \in [0, T_0]$ . Then there exist a pair of functions  $x_c(t) \in \mathbf{L}_2[0, \infty)$ ,  $w_c(t) \in \mathbf{L}_2[0, \infty)$  such that

$$\int_0^\infty x_c^* \tilde{Q} x_c + 2x_c^* \tilde{F} w_c + w_c^* \tilde{R} w_c dt = 0.$$

*Proof.* Let  $v_a \neq 0$  be one of the zero eigenvectors of  $X(t_0, \tilde{\lambda}, \tilde{y})$ . Define

$$\begin{aligned} x_a(t) &= X(t, \tilde{\lambda}, \tilde{y}) v_a, \\ \psi_a(t) &= \begin{bmatrix} 0 & I \end{bmatrix} Z(t, \tilde{\lambda}, \tilde{y}) v_a \\ w_a(t) &= \tilde{R}^{-1} (B' \psi_a(t) - \tilde{F}' x_a(t)). \end{aligned}$$

$(x_a(t), w_a(t))$  is then a pair of non-zero functions that belong to  $\mathbf{L}_2[0, \infty)$ . Note that  $x_a(t_0) = 0$  because  $X(t_0, \tilde{\lambda}) v_a = 0$ . Now define

$$w_c(t) = \begin{cases} 0 & t \in [0, t_0] \\ w_a(t) & t \in [t_0, \infty) \end{cases}$$

and let  $x_c(t)$  denote the solution of  $\dot{x}(t) = A(t)x(t) + B(t)w_c(t)$ ,  $x(0) = 0$ . We have

$$x_c(t) = \begin{cases} 0 & t \in [0, t_0] \\ x_a(t) & t \in [t_0, \infty) \end{cases}.$$

If we use  $(x_c, w_c)$  in (5.19) and recall the derivation in (5.21), we have

$$\int_0^\infty (x_c^* \tilde{Q} x_c + 2x_c^* \tilde{F} w_c + w_c^* \tilde{R} w_c) dt = \lim_{t \rightarrow \infty} x_a^T(t) \psi_a(t) - x_a^T(t_0) \psi_a(t_0) = 0.$$

The last equality follows from the fact that  $x_a(t)$  and  $\psi_a(t)$  are stable solutions of the Hamiltonian system and  $x_a(t_0) = 0$ .  $\square$

Hence, if condition (ciii) does not hold, we can construct a pair of non-zero functions  $(x_c, w_c)$  such that inequality (5.13) does not hold. Again, since  $Q(t, \lambda, y)$ ,  $F(t, \lambda)$ ,  $R(t, \lambda, y)$

---

<sup>3</sup>Here we explicitly indicate the fact that  $X(t)$  in (5.18) depends on  $(\tilde{\lambda}, \tilde{y})$ , since  $M(t)$  does.

depend affinely on  $(\lambda, y)$ , we have

$$\begin{aligned} & \int_0^\infty x_c^* Q(t, \lambda, y) x_c + 2x_c^* F(t, \lambda) w_c + w_c^* R(t, \lambda, y) w_c dt \\ &= \int_{t_0}^\infty x_c^* Q(t, \lambda, y) x_c + 2x_c^* F(t, \lambda) w_c + w_c^* R(t, \lambda, y) w_c dt = a' \lambda - b - cy, \end{aligned} \quad (5.22)$$

for some vector  $a$  and scalars  $b, c$ . Then  $a' \lambda - b - cy = 0$  is the separating hyperplane and the inequality  $a' \lambda - b - cy > 0$  shall be used to exclude  $(\tilde{\lambda}, \tilde{y})$  from the feasible set. Note that there may be more than one  $t_0$  such that  $X(t_0, \tilde{\lambda}, \tilde{y})$  is singular. Also the null space of  $X(t_0, \tilde{\lambda}, \tilde{y})$  could be of dimension higher than one. In these cases, we should use all the information to derive as many separating hyperplanes as possible.

### 5.3.3 Some Issues Regarding Numerical Computation

In most applications, all matrices are given as numerical data and all computations are performed numerically. In the following, we discuss several issues regarding the numerical computation.

#### Computation of the integral in (5.22)

Recall that in (5.22) we had to compute

$$a\lambda - b - cy := \int_{t_0}^\infty x_c^* Q(t, \lambda, y) x_c + 2x_c^* F(t, \lambda) w_c + w_c^* R(t, \lambda, y) w_c dt \quad (5.23)$$

where

$$\begin{bmatrix} Q(t, \lambda, y) & F(t, \lambda) \\ F(t, \lambda)' & R(t, \lambda, y) \end{bmatrix} = \begin{bmatrix} Q_0(t) & F_0(t) \\ F_0(t)' & R_0(t) \end{bmatrix} + \sum_{j=1}^n \lambda_j \begin{bmatrix} Q_j(t) & F_j(t) \\ F_j(t)' & R_j(t) \end{bmatrix} - y \begin{bmatrix} Q_{n+1}(t) & 0 \\ 0 & R_{n+1}(t) \end{bmatrix}. \quad (5.24)$$

$Q_{n+1}(t)$  and  $R_{n+1}(t)$  are identity matrices. The following proposition shows that we can avoid an infinite interval integration (5.23) by solving Lyapunov equations.

**Proposition 5.3.** *Let  $M(t)$  be the evolution matrix of the system (5.14) with  $\lambda = \tilde{\lambda}$  and  $y = \tilde{y}$ . Let  $Z_s$  and  $X(t)$  be defined as in (5.16), (5.18) respectively. Suppose that  $X(t)$  is singular at  $t_0$  and  $v_a$  is a zero eigenvector of  $X(t_0)$ . Then  $b$ , the  $j^{\text{th}}$  component of  $a$ , and  $c$  can be computed as  $v_a' P_j v_a$ ,  $j = 0, 1, 2, \dots, n+1$ , respectively. Each  $P_j$  is the solution of*

the Lyapunov equation

$$\Gamma^T P_j \Gamma - P_j = \Sigma_j, \quad (5.25)$$

where

$$\begin{aligned} \Sigma_j &= Z_s' \left( \int_{t_0}^{t_0+T_0} M(\tau)' \Psi_j(\tau) M(\tau) d\tau \right) Z_s, \\ \Psi_j &= \begin{bmatrix} I & 0 \\ -R_j^{-1} F_j' & R_j^{-1} B' \end{bmatrix}' \begin{bmatrix} Q_j & F_j \\ F_j' & R_j \end{bmatrix} \begin{bmatrix} I & 0 \\ -R_j^{-1} F_j' & R_j^{-1} B' \end{bmatrix}, \end{aligned}$$

and  $Q_j, F_j, R_j, j = 0, 1, \dots, n+1$  are defined as in (5.24). Finally,  $\Gamma$  is a real square matrix that corresponds to the stable eigenvalues of  $M(T_0)$ ; i.e.,  $M(T_0)Z_s = Z_s\Gamma$ .

*Proof.* To see how to get (5.25), we restrict the integration of the  $j^{\text{th}}$  component of  $a$  in (5.22) to one period.

$$\int_{t_0+kT_0}^{t_0+(k+1)T_0} x_a^T Q_j x_a + 2x_a^T F_j w_a + w_a^T R_j w_a dt = \int_{t_0+kT_0}^{t_0+(k+1)T_0} \begin{bmatrix} x_a(t) \\ \psi_a(t) \end{bmatrix}' \Psi_j(t) \begin{bmatrix} x_a(t) \\ \psi_a(t) \end{bmatrix} dt.$$

Let  $\tau = t - kT_0$ , then the integral can equivalently be written as

$$\int_{t_0}^{t_0+T_0} \begin{bmatrix} x_a(\tau + kT_0) \\ \psi_a(\tau + kT_0) \end{bmatrix}' \Psi_j(\tau + kT_0) \begin{bmatrix} x_a(\tau + kT_0) \\ \psi_a(\tau + kT_0) \end{bmatrix} d\tau. \quad (5.26)$$

Notice that

$$\begin{bmatrix} x_a(\tau + kT_0) \\ \psi_a(\tau + kT_0) \end{bmatrix} = M(\tau + kT_0) Z_s v_a = M(\tau) M(T_0)^k Z_s v_a = M(\tau) \Gamma^k v_a, \quad (5.27)$$

where  $\Gamma^k = M(T_0)^k Z_s$ . By (5.27) and the periodicity of  $\Psi_j$ , integral (5.26) is equal to

$$(\Gamma^k v_a)^T \left( \int_{t_0}^{t_0+T_0} M(\tau)^T \Psi_j(\tau) M(\tau) d\tau \right) \Gamma^k v_a = v_a^T (\Gamma^k)^T \Sigma_j \Gamma^k v_a.$$

Now it is easy to see that the integral corresponding to the  $j^{\text{th}}$  coefficient in (5.22) can be

expressed as

$$v_a^T \left( \Sigma_j + \Gamma^T \Sigma_j \Gamma + \dots + (\Gamma^k)^T \Sigma_j \Gamma^k + \dots \right) v_a = v_a^T P_j v_a,$$

where  $P_j$  satisfies the Lyapunov equation (5.25). □

### Computation of the evolution matrix $M(t)$

Since the evolution matrix  $M(t)$  of the Hamiltonian system (5.14) is symplectic, it is important to choose a numerical method that can preserve the symplectic structure when numerically computing  $M(t)$ . Let  $T_s$  denote the time step of the numerical integration scheme. The forward Euler formula,

$$M((k+1)T_s) = (I + T_s H(kT_s))M(kT_s), \quad M(0) = I$$

does not ensure the symplectic structure of  $M$ , while the integration method

$$\begin{aligned} & M((k+1)T_s) \\ &= \left( I - \frac{T_s}{2} H(kT_s) \right)^{-1} \left( I + \frac{T_s}{2} H(kT_s) \right) M(kT_s), \end{aligned}$$

guarantees that  $M$  is symplectic.

Let us briefly summarize the material presented in this section: we discuss how to construct the oracle for optimization problems arising from Chapter 3. The oracle has to check three type of inequalities: (5.11), (5.12) and (5.13). Inequalities (5.11) and (5.12) can be checked in the manner described in Section 5.3.1. One way to check inequality (5.13) is discussed in Section 5.3.2. Several issues regarding numerical computation is discussed in Section 5.3.3.

In the next section, we discuss how to check the operator inequality arising from the feasibility and optimization problems discussed in Chapter 4.

## 5.4 The Oracle for Problems Arising from Chapter 4

In Chapter 4, we developed techniques for analyzing certain properties of periodically forced uncertain periodic systems. Conditions that ensure the system has certain desired properties

are expressed as feasibility and/or constrained optimization problems:

$$\text{(F) find } \lambda, \text{ such that } S_\lambda > 0$$

$$\text{(O) } \inf_{\lambda} c' \lambda, \text{ subj. to } S_\lambda > 0.$$

where  $S_\lambda := S_0 + \sum_{i=1}^n \lambda_i S_i$ , and each  $S_i$  is a self-adjoint operator defined on a particular Hilbert space. For a given  $\lambda$ , the operator inequality  $S_\lambda > 0$  is satisfied if and only if the following inequalities hold.

1. Linear matrix inequalities in the following forms

$$E_1(\lambda) := E_{10} + \sum_{k=1}^n \lambda_k E_{1k} > 0, \quad (5.28)$$

$$E_2(t, \lambda) := E_{20}(t) + \sum_{k=1}^n \lambda_k E_{2k}(t) > 0, \quad \forall t \in [0, T_0], \quad (5.29)$$

where  $E_{1k}$  are real symmetric matrices and  $E_{2k}$  are  $T_0$ -periodic, continuous, real symmetric matrix-valued functions for  $k = 0, 1, \dots, n$ .

2. Integral inequality in the form of

there exists an  $\varepsilon > 0$ , such that

$$\int_0^{T_0} x' Q(t, \lambda) x + 2x' F(t, \lambda) w + w' R(t, \lambda) w \, dt \geq \varepsilon \int_0^{T_0} (\|x\|^2 + \|w\|^2) \, dt, \quad (5.30)$$

$$\forall (x, w) \in \mathcal{L}_2,$$

where  $\mathcal{L}_2 := \{(x, w) \mid \dot{x} = A(t)x + B(t)w, x(0) = x(T_0), w \in \mathbf{L}_2[0, T_0]\}$ . Matrices  $A(t)$ ,  $B(t)$ ,  $Q(t, \lambda)$ ,  $F(t, \lambda)$ , and  $R(t, \lambda)$  are continuous and  $T_0$ -periodically time-varying. Matrix  $A(t)$  is exponentially stable. Matrices  $Q(t, \lambda)$ ,  $F(t, \lambda)$  and  $R(t, \lambda)$  depend affinely on  $\lambda$ :  $Q(t, \lambda) := Q_0(t) + \sum_{k=1}^n \lambda_k Q_k(t)$ ,  $F(t, \lambda) := F_0(t) + \sum_{k=1}^n \lambda_k F_k(t)$ ,  $R(t, \lambda) := R_0(t) + \sum_{k=1}^n \lambda_k R_k(t)$ .

Therefore, given  $(\tilde{\lambda}, \tilde{y})$ , one component of checking whether  $S_{\tilde{\lambda}} - \tilde{y}I$  is to check whether (5.28) and (5.29) hold. This can be done in the manner discussed in Section 5.3.1. Another



component is to check whether

there exists an  $\varepsilon > 0$ , such that

$$\int_0^{T_0} x' \tilde{Q}(t)x + 2x' \tilde{F}(t)w + w' \tilde{R}(t)w dt \geq \varepsilon \int_0^{T_0} \|x\|^2 + \|w\|^2 dt, \quad \forall (x, w) \in \mathcal{L}_2 \quad (5.31)$$

where  $\tilde{Q}(t) = Q(t, \tilde{\lambda}) - \tilde{y}I$ ,  $\tilde{F}(t) = F(t, \tilde{\lambda})$ , and  $\tilde{R}(t) = R(t, \tilde{\lambda}) - \tilde{y}I$ . In this section, we will discuss how to check whether (5.31) holds and, if it does not hold, how to derive hyperplanes which separate  $(\tilde{\lambda}, \tilde{y})$  from the set of points which satisfy (5.31). We start with the following definition.

**Definition 5.3.** The pair  $(A, B)$  is completely controllable if any of the following equivalent conditions hold (here  $t_1, t_0 \in [0, T_0]$ )

1. For any  $x_1, x_0$  and  $t_1 > t_0$  there exists a  $u \in \mathbf{L}_2[t_0, t_1]$  such that  $\dot{x} = Ax + Bu$ ,  $x(t_0) = x_0$  gives  $x(t_1) = x_1$ .
2.  $W(t_0, t_1) = \int_{t_0}^{t_1} \Phi(t_0, \tau)B(\tau)B(\tau)'\Phi(t_0, \tau)'d\tau > 0$ , for all  $t_1 > t_0$ , where

$$\frac{d}{d\tau} \Phi(t_0, \tau) = A(\tau)\Phi(t_0, \tau), \quad \Phi(t_0, t_0) = I.$$

3. For any interval  $[t_0, t_1] \subset [0, T_0]$  we have the implication

$$\begin{cases} \frac{d\psi}{dt} = -A'\psi \\ \psi'B = 0 \end{cases} \Rightarrow \psi(t) \equiv 0 \text{ on } [t_0, t_1].$$

Similar to the one in Definition 5.2, the Hamiltonian system associated with (5.31) is defined as

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{\psi}(t) \end{bmatrix} &= \mathcal{H}(t) \begin{bmatrix} x(t) \\ \psi(t) \end{bmatrix}, \\ \mathcal{H}(t) &= \begin{bmatrix} A(t) - B(t)\tilde{R}(t)^{-1}\tilde{F}(t)' & B(t)\tilde{R}(t)^{-1}B(t)' \\ \tilde{Q}(t) - \tilde{F}(t)\tilde{R}(t)^{-1}\tilde{F}(t)' & -A(t)' + \tilde{F}(t)\tilde{R}(t)^{-1}B(t)' \end{bmatrix}, \end{aligned} \quad (5.32)$$

The transition matrix of system (5.32) is defined by  $\frac{d}{dt}M(t, t_0) = \mathcal{H}(t)M(t, t_0)$ ,  $M(t_0, t_0) = I$ . We will use the simplified notation  $M(t) := M(t, 0)$ . Let  $M(t)$  be partitioned as

$$M(t) = \begin{bmatrix} M_{11}(t) & M_{12}(t) \\ M_{21}(t) & M_{22}(t) \end{bmatrix}, \quad (5.33)$$

where the size of each block is the same as the corresponding block in  $\mathcal{H}(t)$ .

#### 5.4.1 A Frequency Theorem

We now formulate a so-called frequency theorem, which gives necessary and sufficient conditions for (5.31) to be strictly positive definite.

**Theorem 5.3 (Frequency Theorem).** *Consider (5.31) and the corresponding Hamiltonian system. Let  $M(t)$  be the transition matrix of the Hamiltonian system. Assume that  $(A, B)$  are completely controllable. Then the following two statements are equivalent*

(a) *There exists an  $\varepsilon > 0$  such that*

$$\int_0^{T_0} x' \tilde{Q}(t)x + 2x' \tilde{F}(t)w + w' \tilde{R}(t)w dt \geq \varepsilon \int_0^{T_0} \|x\|^2 + \|u\|^2 dt, \quad \forall (x, w) \in \mathcal{L}_2.$$

(b) *We have*

$$(bi) \quad \exists \varepsilon > 0 \text{ such that } \tilde{R}(t) \geq \varepsilon I, \quad \forall t \in [0, T_0],$$

$$(bii) \quad \det(M_{12}(t)) \neq 0, \quad \forall t \in (0, T_0],$$

$$(biii) \quad \Xi = \frac{1}{2}(N + N') > 0,$$

$$\text{where } N = M_{21}(T_0) + (M_{22}(T_0) - I)M_{12}(T_0)^{-1}(I - M_{11}(T_0)).$$

*Proof.* The theorem is adapted from [40]. We will show in Propositions 5.4 and 5.5 that conditions (bii) and (biii) are necessary for Statement (a). The full version of the proof is included in Section 5.6.  $\square$

*Remark 5.1.* Note that  $M_{12}(0) = 0$  so the time interval in (bii) cannot include  $t = 0$ .

The theorem provides a simple way to check whether (5.31) holds for a given  $(\tilde{\lambda}, \tilde{y})$ . To do so, one first checks condition (bi) of Theorem 5.3. If it is satisfied, one then computes the solution of the Hamiltonian system (5.32) and checks conditions (bii) and (biii). Violation

of either condition will lead to one or more hyperplanes which can be used to separate  $(\tilde{\lambda}, \tilde{y})$  and the set of points which satisfy (5.31).

### 5.4.2 Generate Separating Hyperplanes

Suppose that condition (bi) is not satisfied; that is, there exists  $t_0 \in [0, T_0]$  such that  $R(t_0, \tilde{\lambda})$  is not positive definite. Then we can derive separating hyperplanes in exactly the same way as we discussed in Section 5.3.1. If condition (bi) is satisfied but condition (bii) is not, then the following proposition provides a way to derive separating hyperplanes.

**Proposition 5.4.** *Let  $M(t)$  be the transition matrix of the Hamiltonian system defined in (5.32). Suppose that  $M_{12}(t)$  is singular for some  $t \in (0, T_0]$ . Then there exists a pair of non-zero functions  $(z(t), u(t))$  such that*

$$\int_0^{T_0} z' \tilde{Q}(t) z + 2z' \tilde{F}(t) u + u' \tilde{R}(t) u \, dt = 0.$$

*Proof.* Since  $M_{12}(t)$  is singular for some  $t \in (0, T_0]$ , there exist at least one nonzero  $t_* \in (0, T_0]$  and one nonzero  $\psi_0 \in \mathbf{R}^n$  such that  $M_{12}(t_*)\psi_0 = 0$ . Then let

$$\begin{bmatrix} z(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} M_{12}(t) \\ M_{22}(t) \end{bmatrix} \psi_0 \text{ for } t \in [0, t_*], \quad \text{and} \quad \begin{bmatrix} z(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} 0 \\ \diamond \end{bmatrix} \text{ for } t \in (t_*, T_0]. \quad (5.34)$$

where  $\diamond$  denotes a function we don't need to bother about. Using this choice of  $z(t)$  and

$$u = \begin{cases} -\tilde{R}^{-1}(\tilde{F}'z - B'\psi), & t \in [0, t_*], \\ 0, & t \in (t_*, T_0], \end{cases} \quad (5.35)$$

we get

$$\begin{aligned} \int_0^{T_0} z' \tilde{Q}(t) z + 2z' \tilde{F}(t) u + u' \tilde{R}(t) u \, dt &= \int_0^{t_*} (z'(\tilde{Q} - \tilde{F} \tilde{R}^{-1} \tilde{F}') z + \psi' B \tilde{R}^{-1} B' \psi) dt \quad (5.36) \\ &= \int_0^{t_*} (z' \dot{\psi} + \dot{z}' \psi) dt = z(t_*)' \psi(t_*) - z(0)' \psi_0 \\ &= 0. \end{aligned}$$

The first and second equalities can be readily verified by substituting the Hamiltonian system defined in (5.32) into the left-hand-side of (5.36), while the last equality follows

from  $z(t_*) = z(0) = 0$ . □

Suppose that  $\tilde{R}(t)$  is invertible for all  $t \in [0, T_0]$  and condition (bii) is violated. This leads to at least one pair of nonzero functions  $(z(t), u(t))$  defined in (5.34) and (5.35) such that

$$\int_0^{T_0} z'Q(t, \tilde{\lambda}, \tilde{y})z + 2z'F(t, \tilde{\lambda})u + u'R(t, \tilde{\lambda}, \tilde{y})u dt = a'\tilde{\lambda} - b - c\tilde{y} \equiv 0,$$

where  $a$  is a vector in  $\mathbf{R}^n$  and  $b, c$  are constants. Since any  $(\lambda, y)$  which satisfies (5.31) has to satisfy  $a'\lambda - b - cy > 0$ , therefore the inequality  $a'\lambda - b - cy > 0$  excludes  $(\tilde{\lambda}, \tilde{y})$  from the set of points which satisfy (5.31).

Finally, suppose that condition (bi) and (bii) are satisfied but condition (biii) is not. The following proposition shows that we can find a pair of functions which violate Statement (a).

**Proposition 5.5.** *Let  $M(t)$  be the transition matrix of the Hamiltonian system defined in (5.32). Let  $\Xi = \frac{1}{2}(N + N')$ , where  $N = M_{21}(T_0) + (M_{22}(T_0) - I)M_{12}(T_0)^{-1}(I - M_{11}(T_0))$ . Suppose that  $\Xi$  is singular. Then there exists a pair of non-zero functions  $(z(t), u(t))$  such that*

$$\int_0^{T_0} z'\tilde{Q}(t)z + 2z'\tilde{F}(t)u + u'\tilde{R}(t)u dt = 0.$$

*Proof.* Since  $\Xi$  is singular, there exists at least one vector  $z_0$ ,  $\|z_0\| = 1$ , such that  $z_0'\Xi z_0 \leq 0$ . Consider now a candidate extremal from the Hamiltonian system (5.32)

$$\begin{bmatrix} z_*(t) \\ \psi_*(t) \end{bmatrix} = \begin{bmatrix} M_{11}(t) & M_{12}(t) \\ M_{21}(t) & M_{22}(t) \end{bmatrix} \begin{bmatrix} z_0 \\ M_{12}(T_0)^{-1}(I - M_{11}(T_0))z_0 \end{bmatrix}, \quad (5.37)$$

and let  $u_*(t) = -\tilde{R}^{-1}(t)\tilde{F}(t)'z_*(t) + \tilde{R}^{-1}(t)B(t)'\psi_*(t)$ . Notice that the choice

$$\psi_0 = M_{12}(T_0)^{-1}(I - M_{11}(T_0))z_0$$

makes  $z(0) = z(T_0) = z_0$ . Therefore,  $(z_*(t), u_*(t)) \in \mathcal{L}_2$ . It can be readily verified that

$$\int_0^{T_0} z'_*\tilde{Q}(t)z_* + 2z'_*\tilde{F}(t)u_* + u'_*\tilde{R}(t)u_* dt = z(T_0)'\psi(T_0) - z_0'\psi_0 = z_0'\Xi z_0 = 0$$

which concludes the proof. □

Since  $Q(t, \lambda, y)$ ,  $F(t, \lambda)$ ,  $R(t, \lambda, y)$  depend affinely on  $(\lambda, y)$ , we have

$$\int_0^{T_0} z'_* Q(t, \lambda, y) z_* + 2z'_* F(t, \lambda) u_* + u'_* R(t, \lambda, y) u_* dt, = a\lambda - b - cy$$

where  $a \in \mathbf{R}^n$  is a vector and  $b, c$  are constants. The proof of Proposition 5.5 implies that  $a\tilde{\lambda} - b - c\tilde{y} \leq 0$ . On the other hand, any  $(\lambda, y)$  which satisfies (5.31) has to satisfy  $a\lambda - b - cy > 0$ . Therefore,  $a\lambda - b - cy = 0$  is a hyperplane which separates  $(\tilde{\lambda}, \tilde{y})$  from the set of points that satisfy (5.31). Note that  $\Xi$  may have more than one eigenvalue less than or equal to 0. In this case, we can use all of them to derive separating hyperplanes.

## 5.5 Examples

In this section, we present several examples to demonstrate the techniques proposed in Chapters 3 and 4 for analysis of periodic systems.

**Example 5.1 (An Academic Example).** Consider the robust performance problem (3.2). Let  $G$  have the state space realization

$$\dot{x} = A(t)x + B \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \quad \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = Cx,$$

where  $B, C$  are constant matrices, and matrix-valued function  $A(t)$  is in the form of

$$A(t) = A_0 + \sum_{k=1,2} A_k^c \cos(k\omega_0 t) + \sum_{k=1,2} A_k^s \sin(k\omega_0 t).$$

Let  $\Delta$  be an unstructured uncertainty with induced  $L_2$ -norm less than  $\mu$ .

Since the  $L_2$ -gain of  $\Delta$  is less than  $\mu$ ,  $\Delta$  is characterized by the IQC defined by the  $\lambda_2$ -parameterized quadratic form

$$\sigma_2(w_2, v_2, \lambda_2) := \lambda_2(\|v_2\|^2 - (\|w_2\|/\mu)^2),$$

for any  $\lambda_2 > 0$ . In this example, we are interested in estimating the induced  $L_2$ -gain from

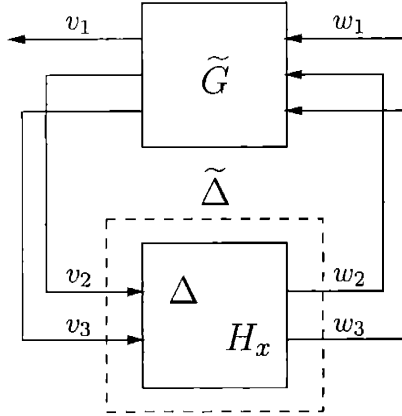


Figure 5-2: Setup for applying Fourier Series Expansion Method to estimate  $L_2$ -gain of the system in Example 5.1.

$w_1$  to  $v_1$ . Therefore, the performance measure we should use is

$$\sigma_{\Phi}(v_1, w_1) := \|v_1\|^2 - \lambda_1 \|w_1\|^2.$$

### Fourier Series Expansion Method

Let  $v_3 = x$  and define

$$w_3 := \begin{bmatrix} w_{31} \\ w_{32} \\ w_{33} \\ w_{34} \end{bmatrix} = H_x(v_3) := \begin{bmatrix} \cos(\omega_0 t)I \\ \sin(\omega_0 t)I \\ \cos(2\omega_0 t)I \\ \sin(2\omega_0 t)I \end{bmatrix} v_3.$$

Let  $H_{x1}(\cdot)$  be the operator which maps  $v_3$  to  $w_{31}$ ; i.e.,  $w_{31} = H_{x1}(v_3) := \cos(\omega_0 t)v_3$ . Define operators  $H_{xi}(\cdot)$ ,  $i = 2, 3, 4$ , in a similar fashion.

To apply the Fourier Series Expansion Method, we transform the original system into the interconnection of  $(\tilde{G}, \tilde{\Delta})$  as shown in Figure 5-2. Here, operator  $\tilde{G}$  is a LTI system with state space realization  $(A_0, B_0, C_0, 0)$ . Matrices  $B_0$  and  $C_0$  are defined as

$$B_0 = \begin{bmatrix} B & A_1^c & A_1^s & A_2^c & A_2^s \end{bmatrix}, \quad C_0 = \begin{bmatrix} C \\ I \end{bmatrix}.$$

Operator  $\tilde{\Delta}$  is block diagonal. The (1,1) block of  $\tilde{\Delta}$  is  $\Delta$ , while the (2,2) block is  $H_x(\cdot)$ . The transformed system is now in the form suitable for IQC analysis developed in [55].

In this example, we use the IQC described in Proposition 3.1 to characterize the operator  $H_x(\cdot)$ . Furthermore, we characterize operator  $H_{x_i}(\cdot)$  using the IQC defined by the parameterized quadratic form

$$\sigma_i(v_3, w_{3i}, X_i) := \int_{-\infty}^{\infty} \hat{v}_3^*(\Pi_i(j\omega - j\omega_0) + \Pi_i(j\omega + j\omega_0))\hat{v}_3 - 2\hat{w}_{3i}^*\Pi_i(j\omega)\hat{w}_{3i} d\omega, \quad (5.38)$$

where  $\hat{v}_3$  and  $\hat{w}_{3i}$  denote the Fourier transforms of  $v_3(t)$  and  $w_{3i}(t)$ , respectively. The frequency dependent matrix  $\Pi_i(j\omega)$  is of the form

$$\Pi_i(j\omega) = \frac{1}{j\omega + 1}X_i + \frac{1}{-j\omega + 1}X_i',$$

where the parameter  $X_i$  has to be constrained such that  $\Pi_i(j\omega) > 0 \forall \omega \in [0, \infty]$ . To see the quadratic form (5.38) defines a set of IQCs for operator  $H_{x_i}(\cdot)$ , readers are referred to [55].

Apply the standard IQC analysis, the  $L_2$ -gain estimation problem can be formulated as a SDP. Here we omit the details of the problem formulations.

### Periodic IQC Approach

In the Fourier Series Expansion Method, we have to transform the original system into a new system where the nominal subsystem is LTI. In contrast, we do not perform any transformation in the Periodic IQC Approach. Instead, we apply Proposition 3.3, and the  $L_2$ -gain estimation problem can be formulated as an optimization problem

$$\inf_{\lambda_1 > 0, \lambda_2 > 0} \lambda_1, \quad \text{such that } S_\lambda > 0,$$

where inequality  $S_\lambda > 0$  holds if and only if there exists an  $\epsilon > 0$  such that

$$\int_0^\infty x'Q(\lambda)x + 2x'F(\lambda)w + w'R(\lambda)w dt \geq \epsilon(\|x\|^2 + \|w\|^2), \quad \forall (x, w) \in \mathcal{L}.$$

$\mathcal{L}$  is defined as  $\{(x, w) \mid \dot{x} = A(t)x + Bw, x(0) = 0, w \in \mathbf{L}_2\}$ . Matrices  $Q(\lambda)$ ,  $F(\lambda)$ ,  $R(\lambda)$  are defined as

$$\begin{bmatrix} Q(\lambda) & F(\lambda) \\ F'(\lambda) & R(\lambda) \end{bmatrix} := \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}' \left[ \begin{array}{cc|cc} -1 & 0 & 0 & 0 \\ 0 & -\lambda_2 & 0 & 0 \\ \hline 0 & 0 & \lambda_1 & 0 \\ 0 & 0 & 0 & \lambda_2 \end{array} \right] \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}.$$

## Numerical Results

Let  $\omega_0 = 3.9641$  and

$$\begin{aligned} A_0 &= \begin{bmatrix} -4.1570 & -0.4687 \\ 0.8592 & -2.9100 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \\ A_1^c &= \begin{bmatrix} -0.8744 & 0.8335 \\ 0.8631 & -0.9388 \end{bmatrix}, \quad A_1^s = \begin{bmatrix} -0.0762 & 0.5442 \\ 1.6971 & -1.5792 \end{bmatrix}, \\ A_2^c &= \begin{bmatrix} -0.1985 & 0.1715 \\ -0.0817 & 0.2795 \end{bmatrix}, \quad A_2^s = \begin{bmatrix} 0.5028 & 0.0133 \\ 0.0624 & 0.4189 \end{bmatrix}. \end{aligned}$$

We solve the SDP arising from the Fourier Series Expansion Method using the MATLAB LMI Control Toolbox and the semi-infinite optimization problem using the algorithm presented in Section 5.2. The results are listed in Table 5.1.

As we can see from Table 5.1, the Fourier Series Expansion Method is much more conservative compared to the Periodic IQC Approach. The Fourier Series Expansion Method can only verify stability of the system up to  $\mu = 1.729$ . The Periodic IQC Approach is able to prove stability up to  $\mu = 2.67$ , a much larger stability margin. Furthermore, given a  $\mu$  for which stability can be verified by both approaches, the Periodic IQC Approach obtains an estimated  $\mathbf{L}_2$ -gain much smaller than the estimate obtained by the Fourier Series Expansion Method.

Recall that in the Fourier Series Expansion Method, the system is transformed in such a way that the harmonic terms in the nominal subsystem are extracted and lumped into the feedback loop. These periodically time-varying coefficients are then treated as ‘‘uncertainties’’ and characterized (in other words, approximated) using IQCs. On the other hand, in



Fourier Series Expansion Method		Periodic IQC Approach	
$\mu$	estimated $L_2$ -gain	$\mu$	estimated $L_2$ -gain
0.20	0.2879	0.20	0.2479
0.40	0.3114	0.40	0.2639
0.60	0.3409	0.60	0.3138
0.80	0.3801	0.80	0.3749
1.00	0.9227	1.00	0.4623
1.50	0.9227	1.50	0.3731
1.70	5.4962	2.00	0.4629
1.72	15.6844	2.30	0.5424
1.725	30.8058	2.60	2.5032
1.729	151.9042	2.67	564.1552
1.730	Infeasible	2.68	infeasible

Table 5.1: List of  $\mu$  and estimated  $L_2$ -gain.

the Periodic IQC Approach, the harmonic terms are kept in the nominal subsystem, and no approximation is introduced for these harmonic terms. Therefore, it is expected that the Fourier Series Expansion Method is more conservative than the Periodic IQC Approach. The price paid for more accurate analysis is the computational effort. In order to verify the robustness condition in the Periodic IQC Approach, one has to solve a convex optimization problem with constraints in the form of quadratic integration, which is more difficult than solving the LMIs arising from the Fourier Series Expansion Method.

**Example 5.2 (Coupled Mathieu Equations).** The problem of lateral bending-torsion vibrations of a thin beam subject to vertical motion of the base is extensively studied. See, for example, [18] and references therein. Usually, the deflection of the beam is approximated by several vibration modes, and the partial differential equations of motion are reduced to a set of coupled Mathieu equations

$$\ddot{X} + D\dot{X} + FX - \omega_F^2 \cos(\omega_F t) EX = 0 \quad (5.39)$$

where  $E$  is a real symmetric matrix, and  $D, F$  are diagonal matrices. Stability analysis is then performed based on (5.39). To be rigorous, the effects of the unmodeled vibration modes should be treated as uncertainties and included in the Mathieu equations. The

Mathieu equations then become

$$\ddot{X} + D\dot{X} + FX - \omega_F^2 \cos(\omega_F t) (EX + \Delta_m(X)) = 0, \quad (5.40)$$

where  $\Delta_m(X)$  represents the effects of the unmodelled vibration modes.

Suppose that  $X$  is a  $3 \times 1$  vector. In this example, we will investigate whether 0 is a stable equilibrium of (5.40) when the unmodelled dynamics and uncertainties in the damping matrix  $D$  are taken into account. Let  $D := D_0 + \text{diag}(\delta_1, \delta_2, \delta_3)$ , where  $\delta_i$  are unknown constants. Then equation (5.40) can be represented as the  $(G, \Delta)$  interconnection shown on the left-hand-side of Figure 3-1 (with the external excitation  $f$  equal to 0). Let  $x' = [X_1 \ X_2 \ X_3 \ \dot{X}_1 \ \dot{X}_2 \ \dot{X}_3]$ . Then operator  $G$  has state space representation

$$\dot{x} = A(t)x + B(t) \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \quad \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = Cx,$$

where  $v_1, v_2, w_1, w_2$  are  $3 \times 1$  vector signals, and

$$A(t) = \begin{bmatrix} 0 & I_3 \\ \omega_F^2 \cos(\omega_F t)E - F & -D_0 \end{bmatrix}, \quad B(t) = \begin{bmatrix} 0 & 0 \\ I_3 & \omega_F^2 \cos(\omega_F t)I_3 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix}. \quad (5.41)$$

Operator  $\Delta$  is in the following form

$$\Delta = \begin{bmatrix} \Delta_\delta & 0 \\ 0 & \Delta_m \end{bmatrix},$$

where  $\Delta_\delta = \text{diag}(\delta_1, \delta_2, \delta_3)$ . We assume that operator  $\Delta_m$  has induced  $\mathbf{L}_2$ -gain less than or equal to  $\gamma_2$  and each uncertain constant  $\delta_i$  satisfies  $|\delta_i| \leq \gamma_1$ .

### Fourier Series Expansion Method

Again, to apply the Fourier Series Expansion Method, we first transform the system such that the nominal subsystem becomes linear time-invariant. To do so, we introduce  $3 \times 1$

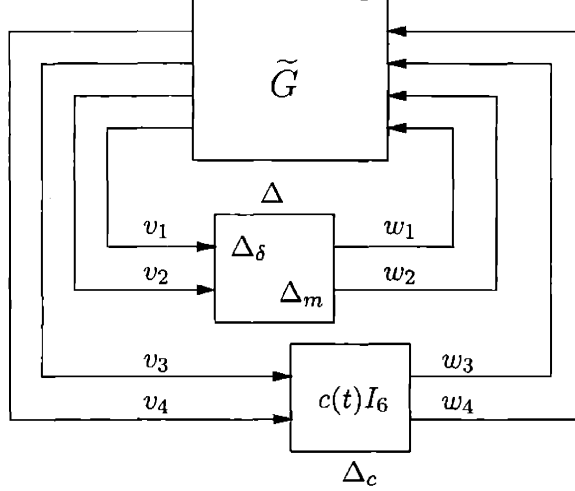


Figure 5-3: Setup for robust stability analysis of the uncertain coupled Mathieu equations (5.40) using Fourier Series Expansion Method. Here  $c(t) = \cos(\omega_F t)$ .

vector signals  $v_3, v_4, w_3, w_4$  defined as

$$v_3 = x, \quad v_4 = w_2, \quad w_3 = \cos(\omega_F t)v_3, \quad w_4 = \cos(\omega_F t)v_4.$$

It can be readily verified that the coupled Mathieu equation (5.40) is equivalent to the system shown in Figure 5-3, where the LTI system  $\tilde{G}$  has state space realization

$$\dot{x} = \begin{bmatrix} 0 & I \\ -F & -D_0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 & 0 \\ -I & 0 & \omega_F^2 E & \omega_F^2 I \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_3 \end{bmatrix}, \quad \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 & I \\ I & 0 \\ I & 0 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_3 \end{bmatrix}.$$

Standard IQC analysis developed in [55] now can be applied to the transformed system. In this example, operator  $\Delta$  is characterized using the IQC defined by  $\lambda$ -parameterized quadratic form

$$\sigma_1(y_1, f_1, \lambda) := \int_0^\infty y_1' D_1(\lambda) y_1 + f_1' D_2(\lambda) f_1 dt, \quad (5.42)$$

where  $y_1$  and  $f_1$  are  $6 \times 1$  vector signals in  $L_2[0, \infty)$ .  $\lambda$ -parameterized matrices  $D_1(\lambda)$  and  $D_2(\lambda)$  are in the form of:  $D_1(\lambda) := \text{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4 I_3)$  and  $D_2(\lambda) := -\text{diag}(\frac{\lambda_1}{\gamma_1}, \frac{\lambda_2}{\gamma_1}, \frac{\lambda_3}{\gamma_1}, \frac{\lambda_4}{\gamma_2} I_3)$ , where parameters  $\lambda_i, i = 1, \dots, 4$ , have to be positive. Operator  $\Delta_c$  is characterized using

the IQC defined by the  $X$ -parameterized quadratic form similar to (5.38)

$$\sigma_2(y_2, f_2, X) := \int_{-\infty}^{\infty} \hat{y}_2^*(\Pi(j\omega - j\omega_0) + \Pi(j\omega + j\omega_0))\hat{y}_2 - 2\hat{f}_2^*\Pi(j\omega)\hat{f}_2 d\omega,$$

where  $y_2(t)$ ,  $f_2(t)$  are  $6 \times 1$  vector signals in  $\mathbf{L}_2[0, \infty)$ , and  $\hat{y}_2$  and  $\hat{f}_2$  denote the Fourier transforms of  $y_2(t)$  and  $f_2(t)$ , respectively. The frequency dependent matrix  $\Pi(j\omega)$  is of the form

$$\Pi(j\omega) = \frac{1}{j\omega + 2}X + \frac{1}{-j\omega + 2}X',$$

where the parameter  $X$  has to be constrained such that  $\Pi(j\omega) > 0 \forall \omega \in [0, \infty]$ . Apply the standard IQC analysis, the condition for stability can be formulated as checking feasibility of a set of linear matrix inequality. Here we omit the detailed formulation.

### Periodic IQC Approach

If the coupled Mathieu equation (5.40) is analyzed using the Periodic IQC Approach, the nominal linear system is kept to be periodic time-varying, and no system transformation is required. Using the IQC defined by the  $\lambda$ -parameterized quadratic form (5.42) to characterize  $\Delta$ , we can formulate the condition for stability as checking feasibility of the following problem

$$\text{find } \lambda_i > 0, i = 1, \dots, 4, \quad \text{such that } S_\lambda > 0,$$

where  $S_\lambda$  is a  $\lambda$ -parameterized self-adjoint operator defined on a certain Hilbert space. Inequality  $S_\lambda > 0$  holds if and only if there exist an  $\epsilon > 0$  such that

$$\int_0^\infty -x'C'D_1Cx - w'D_2w dt \geq \epsilon \int_0^\infty \|x\|^2 + \|w\|^2 dt, \quad \forall (x, w) \in \mathcal{L},$$

$$\mathcal{L} := \{(x, w) \mid \dot{x} = A(t)x + Bw, x(0) = 0, w \in \mathbf{L}_2\}.$$

Matrices  $A(t)$ ,  $B(t)$ ,  $C$  are defined as in (5.41).

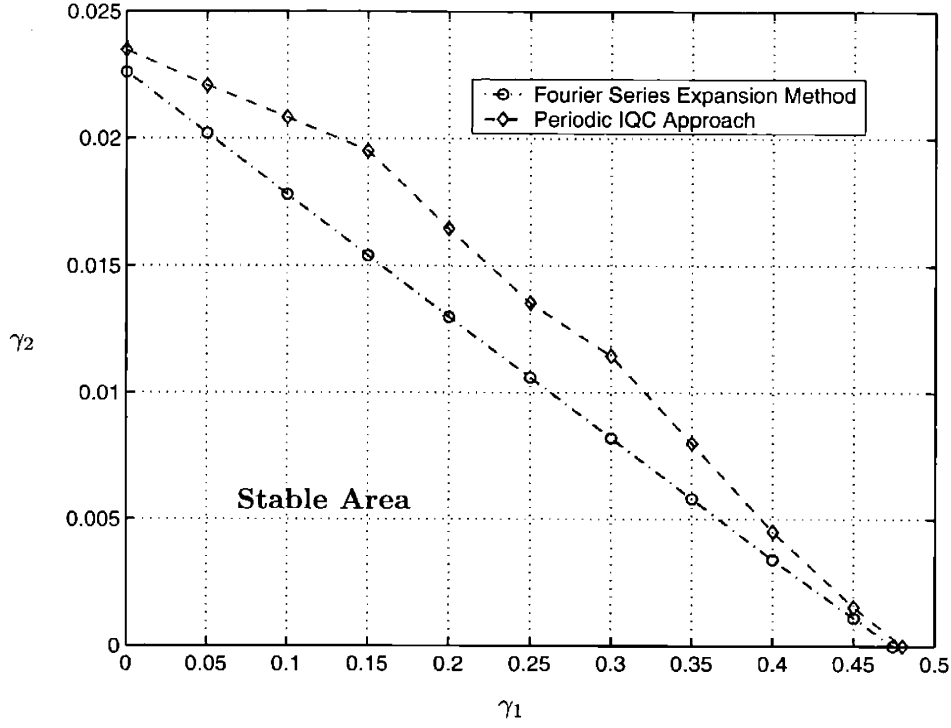


Figure 5-4: Stability region for the uncertain coupled Mathieu equations.

### Numerical Results

Let  $\omega_F = 12$  and

$$D_0 = \begin{bmatrix} 0.48 & 0 & 0 \\ 0 & 1.31 & 0 \\ 0 & 0 & 2.15 \end{bmatrix}, \quad F = \begin{bmatrix} 47.5 & 0 & 0 \\ 0 & 1902 & 0 \\ 0 & 0 & 11481 \end{bmatrix}, \quad \omega_F^2 E = \begin{bmatrix} 0.059 & -0.249 & 2.505 \\ -0.249 & 1.035 & -9.570 \\ 2.505 & -9.570 & -1.095 \end{bmatrix}.$$

This set of data is obtained from [18].

Using the Fourier Series Expansion Method and the Periodic IQC Approach, we identify the set of  $(\gamma_1, \gamma_2)$  for which the uncertain coupled Mathieu equations are stable. The stability regions obtained by both method are shown in Figure 5-4. We see that the stability region obtained by the Fourier Series Expansion Method is strictly contained in the region obtained by Periodic IQC Approach. This implies that Fourier Series Expansion Method is more conservative than the Periodic IQC Approach. The region obtained by Periodic IQC Approach is about 15% larger the one obtained by the Fourier Series Expansion Method.

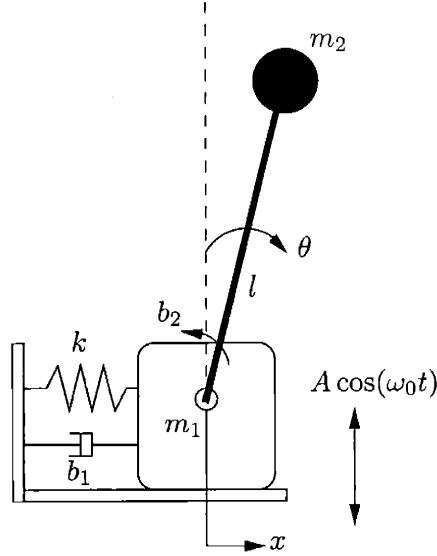


Figure 5-5: An inverted pendulum on a vertically vibrating platform.

**Example 5.3 (An inverted Pendulum on a Vertically Vibrating Platform).** Consider the inverted pendulum system shown in Figure 5-5, where the notations have their usual meanings. The system is subject to a vertically oscillation which is described as  $A \cos(\omega t)$ . The equations of motion of this system can be derived as

$$m_2 l^2 \ddot{\theta} + m_2 l \cos(\theta) \ddot{x} + b_2 \dot{\theta} = m_2 (g - \omega_0^2 A \cos(\omega_0 t)) l \sin(\theta), \quad (5.43)$$

$$(m_1 + m_2) \ddot{x} + m_2 l \cos(\theta) \ddot{\theta} + b_1 \dot{x} + kx = m_2 l \sin(\theta) \dot{\theta}^2. \quad (5.44)$$

It is well-known that the origin of this system is a locally asymptotically stable equilibrium if the amplitude and the frequency of the oscillation are selected from a specific range. Let  $m_1 = 0.2$ ,  $m_2 = l = 1$ ,  $b_1 = b_2 = 2$ ,  $k = 5$ , and  $g = 10$ . In this case, it can be easily verified that the origin of the the system is locally asymptotically stable for the selection of  $A = 0.111$  and  $\omega_0 = 30$ .

In this example, we are interested in investigating the locally asymptotic stability of the origin when the damping coefficient  $b_2$  and the amplitude of the oscillation  $A$  are perturbed. More specifically, let  $A = 0.111(1 + \delta_1)$  and  $b_2 = 2(1 + \delta_2)$ . We would like to estimate the ranges of  $\delta_1$  and  $\delta_2$  such that the origin of the system remains locally asymptotically stable. Linearizing equations of motion (5.43) and (5.44) around the origin and substituting the

coefficients with the numerical values, we obtain the following dynamical equations

$$\ddot{\theta} + \ddot{x} + 2(1 + \delta_1)\dot{\theta} = (10 - 99.9(1 + \delta_2) \cos(30t))\theta, \quad (5.45)$$

$$1.2\ddot{x} + \ddot{\theta} + 2\dot{x} + 5x = 0. \quad (5.46)$$

It is well-known that the locally asymptotic stability of the origin is established if one can show that the linearized system defined in (5.45) and (5.46) is asymptotically stable.

### Fourier Series Expansion Method

To check whether the system defined in (5.45) and (5.46) is asymptotically stable by the Fourier Series Expansion Method, we arrange the system in the standard setup for robustness analysis. In this example, the linear time-invariant part of the system corresponds to the dynamical equations in the form of

$$\ddot{y} + \ddot{z} + 2\dot{y} - 10y = 2w_1 - 99.9w_2 + 99.9w_3,$$

$$1.2\ddot{z} + \ddot{y} + 2\dot{z} + 5z = 0.$$

It can be readily verified that the above LTI dynamical system is *unstable*. Since in any standard robustness analysis method, the system to be analyzed is assumed to have a *stable* linear time-invariant part, therefore, no standard robustness analysis method can be applied to check whether the system defined in (5.45) and (5.46) is stable or not. Hence, in this example, the Fourier Series Expansion Method fails to provide any stability margin!

*Remark 5.2.* That the Fourier Series Expansion Method fails to provide any stability margin can be interpreted physically as follows: suppose that the amplitude of the vertical oscillation is zero; i.e., the inverted pendulum system is not subject to any vertical oscillation. In this case, the origin of the system is indeed *unstable*. Hence, stability of the vibrating inverted pendulum system is due to the vertical oscillation, and the harmonic term in equations (5.45) and (5.46) is essential for proving that the system is stable. In the Fourier Series Expansion Method, the harmonic term is treated as uncertainty, which deteriorates stability instead of providing it. Therefore, this method is not able to provide any positive result regarding stability of the system.

## Periodic IQC Approach

To check whether the system defined in (5.45) and (5.46) is asymptotically stable by the Periodic IQC Approach, we arrange the system in the  $(G, \Delta)$  interconnection as shown in the left-hand-side of Figure 3-1. The nominal subsystem  $G$  has the state space representation

$$\dot{y} = A(t)y + B \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \quad \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = C(t)y,$$

where

$$A(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 60 - 599.4 \cos(30t) & -12 & 25 & 10 \\ 0 & 0 & 0 & 1 \\ -50 + 499.5 \cos(30t) & 10 & -25 & -10 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ -599.4 & -12 \\ 0 & 0 \\ 499.5 & 10 \end{bmatrix},$$

$$C(t) = \begin{bmatrix} \cos(30t) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Operator  $\Delta$  is a diagonal matrix which has two components  $\delta_1$  and  $\delta_2$ , each of which represents an unknown constant. Signals  $v_1, v_2, w_1, w_2$  are related as follows

$$w_1 = \delta_1 v_1, \quad w_2 = \delta_2 v_2.$$

We assume that  $\delta_1, \delta_2$  satisfy  $|\delta_1| \leq \gamma_1$  and  $|\delta_2| \leq \gamma_2$ , respectively.

In this example, we characterize the relationship  $w_1 = \delta_1 v_1$  using the IQC defined by the  $\lambda_1$ -parameterized quadratic form

$$\sigma_2(v_1, w_1, \lambda_1) := \int_{-\infty}^{\infty} \hat{v}_1^*(\lambda_1 + H(j\omega)^* H(j\omega)) \hat{v}_1 - \frac{1}{\gamma_1^2} \hat{w}_1^*(\lambda_1 + H(j\omega)^* H(j\omega)) \hat{w}_1 \, d\omega, \quad \lambda_1 > 0,$$

where  $H(s) = \frac{32s}{s^2 + 57s + 800}$ . Notations  $\hat{v}_1, \hat{w}_1$  denote the Fourier transformations of signals  $v_1$  and  $w_1$ , respectively. The relationship  $w_2 = \delta_2 v_2$  is characterized using the IQC defined by the  $\lambda_2$ -parameterized quadratic form

$$\sigma_2(v_2, w_2, \lambda_2) := \int_{-\infty}^{\infty} \lambda_2 (|\hat{v}_2|^2 - \frac{1}{\gamma_2^2} |\hat{w}_2|^2) \, d\omega, \quad \lambda_2 > 0,$$



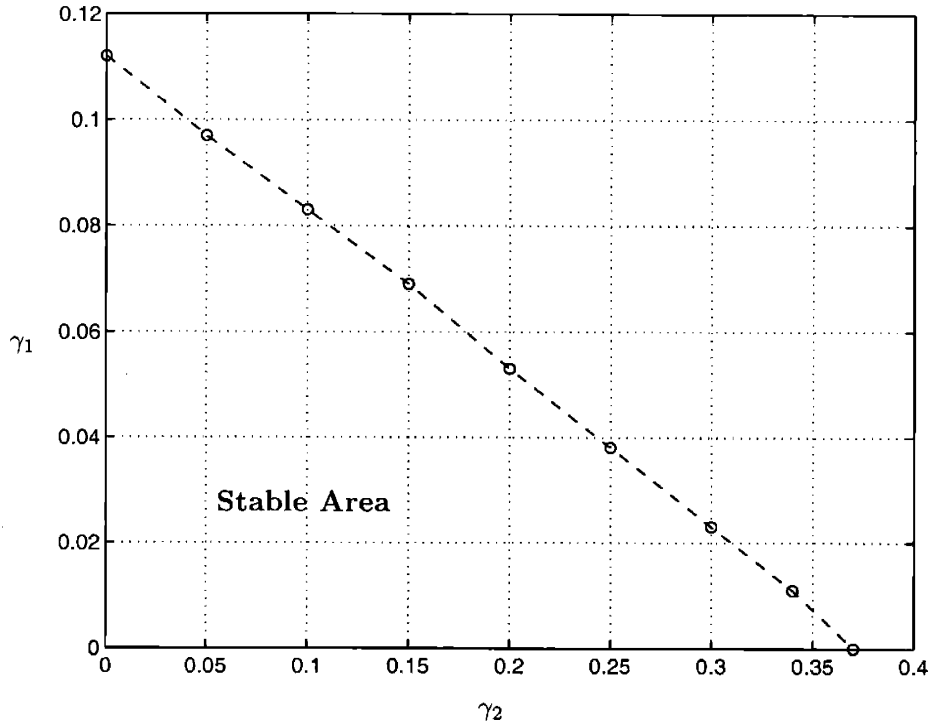


Figure 5-6: Stability region for the inverted pendulum system.

where  $\hat{v}_2$ ,  $\hat{w}_2$  denote the Fourier transformations of signals  $v_2$  and  $w_2$ , respectively. Applying Proposition 3.2 with these IQCs, we can formulate the condition for stability as finding parameters  $\lambda_1$  and  $\lambda_2$  to satisfy a particular parameterized operator inequality. Here we omit the detailed formulation.

Using the computational algorithm developed in this chapter, we obtain a region in the  $(\gamma_1, \gamma_2)$  space such that the system defined in (5.45) and (5.46) is asymptotically stable. The region is shown in Figure 5-6.

**Example 5.4.** Consider again the uncertain coupled Mathieu equations (5.40) in Example 5.2, but now let us add a periodic forcing function  $f(t)$ :

$$\ddot{X} + (D_0 + \Delta_\delta)\dot{X} + FX - \omega_F^2 \cos(\omega_F t) (EX + \Delta_m(X)) = f(t), \quad (5.47)$$

where  $\Delta_\delta$  is a diagonal matrix with three components  $\delta_1$ ,  $\delta_2$ ,  $\delta_3$ . Each of them represents the uncertainty in a damping coefficient.  $\Delta_m(\cdot)$  denotes an uncertain LTI operator which is used to represent the effects of the unmodelled vibration modes. We assume that operator

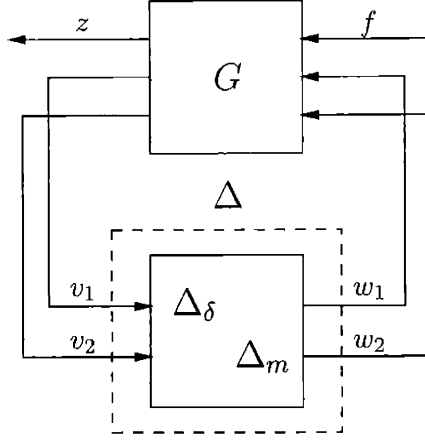


Figure 5-7: Setup for robustness analysis of periodically forced uncertain coupled Mathieu equations.

$\Delta_m$  has induced  $\mathbf{L}_2$ -gain less than or equal to  $\gamma_2$  and each uncertain constant  $\delta_i$  has absolute value less than or equal to  $\gamma_1$ ; i.e.,  $|\delta_i| \leq \gamma_1, i = 1, \dots, 3$ . The period of  $f(t)$  is  $T := \frac{2\pi}{\omega_F}$ .

We can represent (5.47) in the standard form for robustness analysis (See Figure 5-7).

The nominal subsystem  $G$  is periodic and has state space representation

$$\dot{x} = A(t)x + B(t) \begin{bmatrix} w_1 \\ w_2 \\ f \end{bmatrix}, \quad \begin{bmatrix} z \\ v_1 \\ v_2 \end{bmatrix} = Cx$$

where  $v_1, v_2, w_1, w_2, f$  are  $3 \times 1$  vector signals, and  $z = x_1$ . Let  $c(t) = \omega_F^2 \cos(\omega_F t)$ . Matrices  $A(t), B(t)$  and  $C$  are defined as followings

$$A(t) = \begin{bmatrix} 0 & I_3 \\ c(t)E - F & D_0 \end{bmatrix}, \quad B(t) = \begin{bmatrix} B_1(t) & B_2 \end{bmatrix}, \quad C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}, \quad (5.48)$$

$$C_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix}, \quad B_1(t) = \begin{bmatrix} 0 & 0 \\ I_3 & c(t)I_3 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ I_3 \end{bmatrix}.$$

In this example, we are interested in checking whether system (5.47) has a unique stationary periodic solution for any periodic input  $f(t) \in \mathbf{L}_2(T_0)$ . Furthermore, we also want to estimate a bound on the amplitude of  $z(t)$  relative to the amplitude of  $f(t)$ .

Since  $\Delta_\delta$  and  $\Delta_m$  are linear, and we know that for linear operators each IQC is also an incremental IQC, therefore the  $\lambda$ -parameterized quadratic form (5.42) also defines an

incremental IQC for  $\Delta$ .

To check whether the solution of (5.47) is periodic, we apply Theorem 4.1. Using the incremental IQC defined by the parameterized quadratic form (5.42), we can formulate the condition for existence of a unique periodic solution as checking whether the following semi-infinite feasibility problem has a solution

$$\text{find } \lambda_i > 0, \quad i = 1, \dots, 4, \quad \text{such that } S_\lambda > 0,$$

where  $S_\lambda$  is a  $\lambda$ -parameterized self-adjoint operator defined on a certain Hilbert space. Inequality  $S_\lambda > 0$  holds if and only if there exists an  $\epsilon > 0$  such that

$$\int_0^T -x' C_2' D_1 C_2 x - w' D_2 w \, dt \geq \epsilon(\|x\|^2 + \|w\|^2), \quad \forall (x, w) \in \mathcal{L},$$

$$\mathcal{L} := \{(x, w) \mid \dot{x} = A(t)x + B_1(t)w, \quad x(0) = x(T), \quad w \in \mathbf{L}_2(T)\}.$$

Matrices  $A(t)$ ,  $B_1(t)$ , and  $C_2$  are defined as in (5.48).

To estimate the amplitude of  $z$ , we apply Theorem 4.3. The following performance measure is chosen

$$\sigma_\Omega(z, f) = \int_0^T |\dot{z} + z|^2 - \lambda_5 \|f\|^2 \, dt,$$

where the corresponding  $\Omega$  is

$$\Omega = \begin{bmatrix} \Psi^* \Psi & 0 \\ 0 & -\lambda_5 I_3 \end{bmatrix}, \quad \lambda_5 > 0,$$

and the operator  $\Psi$  is defined as  $\Psi(z) = (1 + \frac{d}{dt})z$ . Now, to estimate the magnitude of  $z$ , we solve the following optimization problem

$$\inf_{\lambda_i > 0, i=1, \dots, 5} \lambda_5, \quad \text{such that } S_\lambda > 0, \quad (5.49)$$

where inequality  $S_\lambda > 0$  holds if and only if there exists an  $\epsilon > 0$  such that

$$\int_0^T -|\Psi C_1 x|^2 - x' C_2' D_1 C_2 x + w' D_2 w + \lambda_5 \|f\|^2 dt \geq \epsilon(\|x\|^2 + \|w\|^2), \quad \forall (x, w) \in \mathcal{L},$$

$$\mathcal{L} := \{(x, w) \mid \dot{x} = A(t)x + B_1(t)w + B_2 f, x(0) = x(T), (w, f) \in \mathbf{L}_2(T)\}.$$

Matrices  $A(t)$ ,  $B_1(t)$ ,  $B_2$ ,  $C_1$ , and  $C_2$  are defined as in (5.48).

## Numerical Results

Using the computational algorithm developed in Section 5.2, we were able to verified that if  $\|\Delta_m\| \leq 0.0165$  and  $|\delta_i| \leq 0.2$ ,  $i = 1, \dots, 3$ , then the system defined in (5.47) has a unique stationary periodic solution when subjected to periodic excitation. Furthermore, the objective of optimization problem (5.49) is found to be 38.6096. Then, applying Proposition 4.1, we have

$$\max_{t \in [0, T]} |z(t)| \leq \gamma \|f\|_{\mathbf{L}_2(T_0)},$$

where

$$\gamma = \left( \sum_{k=-\infty}^{\infty} \frac{\lambda_5}{1 + k^2 \omega_F^2} \right)^{\frac{1}{2}} = \frac{\sqrt{\lambda_5}}{(1 - e^{-T})} \sqrt{\frac{1 - e^{-2T}}{2}} = 4.4779.$$

Thus, we conclude that the amplitude of  $z(t)$  is bounded by  $4.4779 \|f\|_{\mathbf{L}_2(T_0)}$ .

## 5.6 Proof of Theorem 5.3

The following proof is adapted from [40]. The original version of the proof was due to U. Jönsson and A. Megretski.

The proof can be divided into two main steps.

**S1:** We first show that (bi) and (bii) are necessary for the following optimization problem to be strictly convex

$$\inf_{(z, u) \in \mathcal{L}(a, b)} \int_0^{T_0} \sigma(t, z, u) dt,$$

where  $\sigma(t, z, u) = z' \bar{Q} z + 2z' \bar{F} u + u' \bar{R} u$  and  $\mathcal{L}(a, b) = \{(z, u) : \dot{z} = Az + Bu, z(0) =$

$a, z(T_0) = b, u \in \mathbf{L}_2[0, T_0]\}$ . From the proof it also follows that  $(a) \Rightarrow (bi)$  and  $(bii)$ .

**S2:** From the previous step we know that  $(bi)$  and  $(bii)$  imply that

$$\inf_{(z,u) \in \mathcal{L}(z_0, z_0)} \int_0^{T_0} \sigma(t, z, u) dt, \quad (5.50)$$

has a unique minimum for all  $z_0 \in \mathbf{R}^n$ . We will show, using the extremal from the Hamiltonian system, that

$$\min_{(z,u) \in \mathcal{L}(z_0, z_0)} \int_0^{T_0} \sigma(t, z, u) dt = z_0' N z_0.$$

This shows that  $(biii)$  is also necessary. A simple calculation will show the sufficiency, which gives the implication  $(b) \Rightarrow (a)$ . A proof of **S2** is previously given in [51].

We will start with **S1** which is the main part of the proof. Let  $\tilde{z}$  and  $\tilde{u} \in \mathbf{L}_2(T_0)$  be such that  $\tilde{z} = A\tilde{z} + B\tilde{u}$ ,  $\tilde{z}(0) = a$ ,  $\tilde{z}(T_0) = b$ . This means that  $\mathcal{L}(a, b) = (\tilde{z}, \tilde{u}) + \mathcal{L}_0$ , where  $\mathcal{L}_0 = \mathcal{L}(0, 0)$ . If we use the notation  $v = (z, u) \in \mathcal{L}(a, b)$ ,  $\tilde{v} = (\tilde{z}, \tilde{u})$ , and  $v_0 = (z_0, u_0) \in \mathcal{L}_0$ , and  $\langle v, v \rangle = \int_0^{T_0} \sigma(t, z, u) dt$ , then we have

$$\inf_{v \in \mathcal{L}(a, b)} \langle v, v \rangle = \inf_{v_0 \in \mathcal{L}_0} \langle v_0, v_0 \rangle + 2 \langle v_0, \tilde{v} \rangle + \langle \tilde{v}, \tilde{v} \rangle.$$

Thus, the optimization problem is strictly convex if and only if the problem

$$\inf_{v_0 \in \mathcal{L}_0} \langle v_0, v_0 \rangle$$

is strictly convex. This means that we need to prove that

$$\int_0^{T_0} \sigma(t, z, u) dt \geq \varepsilon \int_0^{T_0} (\|z\|^2 + \|u\|^2) dt, \quad \forall (z, u) \in \mathcal{L}_0 \quad (5.51)$$

We will show that the following are equivalent, which also proves  $(a) \Rightarrow (bi)$  and  $(bii)$ .

(I) There exists an  $\varepsilon > 0$  such that

$$\int_0^{T_0} \sigma(t, z, u) dt \geq \varepsilon \int_0^{T_0} (\|z\|^2 + \|u\|^2) dt, \quad \forall (z, u) \in \mathcal{L}_0.$$

(II) We have

$$(IIa) \exists \epsilon > 0 \text{ such that } \tilde{R}(t) \geq \epsilon I, \forall t \in [0, T_0],$$

$$(IIb) \det(M_{12}(t)) \neq 0, \forall t \in (0, T_0].$$

*Proof of (I)  $\Rightarrow$  (II):* It is easy to see that (IIa) is necessary, see for example [35] for a similar proof. The proof of necessity of (IIb) follows arguments similar to those in [35]. Indeed, assume that there exist  $t_*$  and nonzero  $\psi_0 \in \mathbf{R}^n$  such that  $M_{12}(t_*)\psi_0 = 0$ . Then let

$$\begin{bmatrix} z(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} M_{12}(t) \\ M_{22}(t) \end{bmatrix} \psi_0 \text{ for } t \in [0, t_*], \text{ and } \begin{bmatrix} z(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} 0 \\ \diamond \end{bmatrix} \text{ for } t \in (t_*, T_0]. \quad (5.52)$$

where  $\diamond$  is a function we don't need to bother about. Using this choice of  $z(t)$  and the control

$$u = \begin{cases} -\tilde{R}^{-1}(\tilde{F}'z - B'\psi), & t \in [0, t_*] \\ 0, & t \in (t_*, T_0], \end{cases} \quad (5.53)$$

we get

$$\begin{aligned} \int_0^{T_0} \sigma(t, z, u) dt &= \int_0^{t_*} (z'(\tilde{Q} - \tilde{F}\tilde{R}^{-1}\tilde{F}')z + \psi' B\tilde{R}^{-1}B'\psi) dt \\ &= \int_0^{t_*} (z'\dot{\psi} + \dot{z}'\psi) dt = z(t_*)'\psi(t_*) - z(0)'\psi_0 = 0. \end{aligned} \quad (5.54)$$

The first and second equalities can be readily verified by substituting the Hamiltonian system defined in (5.32), while the last equality follows from  $z(t_*) = z(0) = 0$ . The last equality in (5.54) would contradict (I) provided that the solution  $(z, u)$  is nontrivial. We will show the non-triviality by contradiction: the differential equation corresponding to the solution (5.52) over  $[0, t_*]$  can be formulated as

$$\begin{aligned} \dot{z} &= Az + Bu, \quad z(0) = 0, \\ \dot{\psi} &= -A'\psi + \tilde{Q}z + \tilde{F}'u, \quad \psi(0) = \psi_0, \\ u &= -\tilde{R}^{-1}(\tilde{F}'z - B'\psi). \end{aligned}$$

If we assume that  $u \equiv 0$  then  $z \equiv 0$ . This gives  $\dot{\psi} = -A'\psi$ ,  $\psi'B = (\tilde{R}u + \tilde{F}'z)' \equiv 0$ , but

$\psi \neq 0$  on  $[0, t_*]$ , since  $\psi(0) = \psi_0$ . This contradicts the strong controllability assumption on  $(A, B)$ . Therefore, the pair of functions,  $(z(t), u(t))$ , defined in (5.52) and (5.53) are non-zero solutions, and we conclude that violation of (IIb) leads to a pair of non-zero  $(z, u)$  which do not satisfy inequality (I). Thus (I) implies (II).

*Proof of (II)  $\Rightarrow$  (I):* Here we will use standard properties of the Riccati equation that corresponds to the Hamiltonian system. Let  $Z_0 \in \mathbf{R}^{2n \times n}$  and

$$\begin{bmatrix} Z(t) \\ \Psi(t) \end{bmatrix} = M(t)Z_0, \quad t \in [0, T_0].$$

If  $Z(t)$  is nonsingular, then  $P(t) = -\Psi(t)Z(t)^{-1}$  satisfies the following properties (for  $t \in [0, T_0]$ )

- $P(t) = P(t)'$ .
- $\frac{dP}{dt} + PA + A'P + \tilde{Q} = (PB + \tilde{F})\tilde{R}^{-1}(PB + \tilde{F})'$ . (We refer to this equation as the *Riccati equation*).
- If  $\dot{z} = Az + Bu$  and  $K = -\tilde{R}^{-1}(PB + \tilde{F})'$ , we have

$$\frac{d}{dt}(z'Pz) + \sigma(t, z, u) = (u - Kz)' \tilde{R}(u - Kz). \quad (5.55)$$

The choice  $Z_0 = \begin{bmatrix} 0 & I \end{bmatrix}'$  gives  $P_0(t) = -M_{22}(t)M_{12}(t)^{-1}$ , which is defined everywhere except at  $t = 0$ . We will construct a bounded solution to the Riccati equation by perturbing the final value of  $P_0$ . Let  $P(t) = -\Psi(t)Z(t)^{-1}$ , where

$$\begin{bmatrix} Z(t) \\ \Psi(t) \end{bmatrix} = M(t, T_0) \begin{bmatrix} I \\ -P(T_0) \end{bmatrix}, \quad t \in [0, T_0], \quad (5.56)$$

and  $P(T_0) = P_0(T_0) + \varepsilon I$  for some  $\varepsilon > 0$ . We will show below that  $P(t)$  is well defined and bounded for all  $t \in [0, T_0]$ . Then integration of (5.55) gives

$$\int_0^{T_0} \sigma(t, z, u) dt = \int_0^{T_0} (u - Kz)' \tilde{R}(u - Kz) dt. \quad (5.57)$$

The first term of the left hand side of (5.55) disappears because  $z(T_0) = z(0) = 0$  and  $P$  is bounded. The next lemma shows that (5.57) implies (I).

**Lemma 5.1.** *There exists an  $\varepsilon > 0$  such that*

$$\int_0^{T_0} (u - Kz)' \tilde{R}(u - Kz) dt \geq \varepsilon \int_0^{T_0} (\|z\|^2 + \|u\|^2) dt \quad (5.58)$$

for all  $u \in \mathbf{L}_2(T_0)$  such that  $\dot{z} = Az + Bu$ ,  $z(0) = 0$  satisfies  $z(T_0) = 0$ .

*Proof.* Let the subspace  $\mathcal{U} = \{u \in \mathbf{L}_2(T_0) : \dot{z} = Az + Bu, z(0) = 0, \text{ and } z(T_0) = 0\}$  be equipped with the norm

$$\|u\|^2 = \int_0^{T_0} u' \tilde{R} u dt$$

and let  $L : \mathcal{U} \rightarrow \mathbf{L}_2(T_0)$  be defined as  $Lu = u - Kz$ . We will show that the kernel of  $L$  is  $\{0\}$  and the co-dimension of  $\text{Im}(L)$  is finite, where  $\text{Im}(L)$  denotes the range space of  $L$ . It then follows from a version of Banachs' isomorphism theorem that there exists a  $c > 0$  such that  $\|L\| \geq c$ . This gives

$$\int_0^{T_0} (u - Kz)' \tilde{R}(u - Kz) dt \geq \frac{c^2 \varepsilon}{1 + c_1^2} \int_0^{T_0} (\|z\|^2 + \|u\|^2) dt. \quad (5.59)$$

Here we used the fact that  $R(t) \geq \varepsilon I$ . Constant  $c_1$  denotes any bound on the induced norm of the operator  $\mathcal{U} \rightarrow \mathbf{L}_2(T_0)$  defined as  $u \mapsto z$ , where  $\dot{z} = Az + Bu$ .

To see that the kernel of  $L$  is  $\{0\}$ , we note that  $Lu = 0$  implies that  $u = Kz$ . Use of this in the state equation gives  $\dot{z} = (A + BK)z$ ,  $z(0) = 0$ . This means that  $z \equiv 0$  and thus  $u \equiv 0$ .

To prove that the co-dimension of  $\text{Im}(L)$  is finite, we consider an arbitrary vector  $v \in \text{Im}(L)$ . This means that  $v = u - Kz$  for some  $u \in \mathcal{U}$ , which in turn implies that  $\dot{z} = (A + BK)z + Bv$ . Let  $\Phi$  be the transition matrix for  $A + BK$ . Then we have the following constraint on  $v$ :

$$z(T_0) = \int_0^{T_0} \Phi(T_0, t) B(t) v(t) dt = 0.$$

Let  $h_i(t) = B(t)' \Phi(T_0, t)' e_i$ , where  $e_i$  is the  $i^{\text{th}}$  unit vector. It follows that  $\text{Im}(L) = \{v \in \mathbf{L}_2(T_0) : \langle h_i, v \rangle = 0\}$ , which is the intersection of at most  $n$  hyperplanes. This shows that the co-dimension of  $\text{Im}(L)$  is less than or equal to  $n$ .  $\square$

Therefore, we conclude that (II) implies equality (5.57), which in turn implies (I). To



complete the proof, we need to show that  $P(t) = -\Psi(t)Z(t)^{-1}$  is well defined (i.e., bounded over  $t \in [0, T_0]$ ). To do this we use the following two conditions on a Riccati equation

**Strict monotonicity:** From (5.55) it follows that

$$\min_{u \in \mathbf{L}_2[t, T_0]} \left\{ \int_t^{T_0} \sigma(t, z, u) dt + z(T_0)' P(T_0) z(T_0) \right\} = z(t)' P(t) z(t).$$

Hence, if  $P_1$  and  $P_2$  are solutions of the Riccati equation with  $P_1(T_0) \geq P_2(T_0)$  then  $P_1(t) \geq P_2(t)$ , since  $z(t)$  can be chosen arbitrarily.

**Extension of solution:** If  $P(t)$  is defined on  $(t_0, T_0]$  but cannot be extended to  $[t_0 - \varepsilon, T_0]$  for any  $\varepsilon > 0$ , then  $\lambda_{\min}(P(t)) \rightarrow -\infty$  as  $t \rightarrow t_0$ . To see this note that (5.55) gives

$$z(t_0 - \varepsilon)' P(t_0 - \varepsilon) z(t_0 - \varepsilon) = \min_{u \in \mathbf{L}_2[t_0 - \varepsilon, t_0]} \left\{ \int_{t_0 - \varepsilon}^{t_0} \sigma(t, z, u) dt + z(t_0)' P(t_0) z(t_0) \right\}.$$

If  $P(t_0)$  is bounded then there exists  $\varepsilon > 0$  such that the optimization problem on the right hand side is bounded. Note also that  $z(t_0 - \varepsilon)$  can be chosen arbitrarily. This means that  $P(t_0 - \varepsilon)$  must be bounded.

From the above properties we see that our choice of  $P(t)$  satisfies  $P(t) \geq P_0(t)$  and it is defined at least on  $(0, T_0]$ . We will see next that it is defined also at  $t = 0$ . Assume on the contrary that there exists a nonzero  $v_0$  such that  $Z(0)v_0 = 0$ . It then follows from (5.56) that

$$\begin{bmatrix} I \\ -P(T_0) \end{bmatrix} v_0 = M(T_0, 0) \begin{bmatrix} 0 \\ \Psi(0) \end{bmatrix} v_0. \quad (5.60)$$

We also have

$$\begin{bmatrix} I \\ -P_0(T_0) \end{bmatrix} v_0 = M(T_0, 0) \begin{bmatrix} 0 \\ M_{12}(T_0)^{-1} \end{bmatrix} v_0. \quad (5.61)$$

It can be seen that (5.60) and (5.61) imply that  $M_{12}(T_0)^{-1} v_0 = \Psi(0) v_0$ , which in turn implies that  $v_0'(P_0(T_0) - P(T_0))v_0 = 0$ . This is a contradiction. Hence, we have shown that  $P(t)$  must be bounded. We have thus proven (I)  $\Leftrightarrow$  (II).

**Step 2:** Condition (a) holds if and only if the following optimization problem has a strictly

positive objective

$$\inf_{z_0 \in \mathbf{R}^n} \inf_{(z,u) \in \mathcal{L}(z_0, z_0)} \int_0^{T_0} \sigma(t, z(t), u(t)) dt \quad (5.62)$$

We have shown that conditions (bi) and (bii) imply that the inner optimization is strictly convex. Hence, the extremal candidates suggested by the Hamiltonian system are indeed optimal for the inner optimization problem. Let  $z(0) = z_0$  be given. Integration of the Hamiltonian system gives

$$\begin{bmatrix} z(T_0) \\ \psi(T_0) \end{bmatrix} = \begin{bmatrix} M_{11}(T_0) & M_{12}(T_0) \\ M_{21}(T_0) & M_{22}(T_0) \end{bmatrix} \begin{bmatrix} z_0 \\ \psi_0 \end{bmatrix}.$$

The constraint  $z(T_0) = z_0$  implies that  $\psi_0 = M_{12}(T_0)^{-1}(I - M_{11}(T_0))z_0$ . Hence

$$\min_{(z,u) \in \mathcal{L}(z_0, z_0)} \int_0^{T_0} \sigma(t, z, u) dt = z(T_0)' \psi(T_0) - z_0' \psi_0 = z_0' N z_0, \quad (5.63)$$

where  $N$  is defined in (biii) of the theorem statement. We immediately see that that  $N + N' > 0$  is a necessary and sufficient condition for the optimal objective of (5.63) to be strictly positive. Thus we conclude that (biii)  $\Leftrightarrow$  (a).

## 5.7 Summary

In this chapter, a cutting plane algorithm is proposed to solve the feasibility and optimization problems arising from Chapters 3 and 4. The algorithm follows the basic principles of Kelley's cutting plane algorithm. There are, however, two distinctions. One is that, in our algorithm, the constrained optimization problems are solved in two steps via two unconstrained spectral value maximization problems. This is different from Kelley's algorithm for solving constrained convex optimization problems. The other difference is that the two basic steps of the Kelley's cutting plane algorithm, namely, evaluation of the function to be optimized and computation of a subgradient of the function at each trial point, are replaced by slightly different procedures in our algorithm.

The essential part of the algorithm is the oracle, which checks whether a given trial point satisfies the constraint in an optimization problem. We use one of Yakubovich's results to construct the oracle for the problems arising from Chapter 3. A new frequency theorem

by Jönsson and Megretski is used to construct the oracle for the problems arising from Chapter 4.

Several examples are presented in this chapter to demonstrate the techniques proposed in Chapters 3 and 4 for analysis of periodic systems. The results of the examples indicate that between the two methods proposed in Chapter 3 for robustness analysis of periodic systems, Periodic IQC Approach is more accurate than the Fourier Series Expansion Method. This is because the periodically time-varying part of the nominal system is treated approximately using IQCs in the Fourier Series Expansion Method, while it is treated exactly in the Periodic IQC Approach.



## Chapter 6

# Cutting Plane Algorithms for Standard IQC Problems

The conventional way to treat standard Integral Quadratic Constraint (IQC) problems is to transform them into Semi-Definite Programs (SDPs). SDPs can be solved using interior point methods which have been proven efficient. This approach, however, is not the *most* efficient since it introduces additional decision variables to the SDP, and the additional decision variables sometimes largely increase the complexity of the problem. In this chapter, the Kelley type cutting plane algorithm presented in the previous chapter, the ellipsoid algorithm, and the analytical center cutting plane method are implemented to solve standard IQC problems. We will demonstrate that in certain cases these cutting plane algorithms can solve IQC problems *much* faster than the conventional approach. Numerical examples, as well as some explanations from the point of view of computational complexity, are provided to support our point.

### 6.1 Introduction

Developing systematic methods for analyzing stability and performance robustness of a system has been a dominating research topic in the area of systems and control theory because of its practical importance. Many different approaches have been developed over the past several decades. A class of methods, based on the idea of utilizing Integral Quadratic Constraints (IQC) to characterize uncertainties and/or nonlinearities in the system to be analyzed, offers flexible frameworks to analyze large and complex systems [69, 16, 55]. The

robustness conditions given by these methods can be expressed as feasibility problems:

$$\begin{aligned} & \text{find } \lambda_i, \text{ such that} \\ & \begin{bmatrix} G(j\omega) \\ I \end{bmatrix}^* (\Sigma_0 + \sum_{i=1}^n \lambda_i \Sigma_i) \begin{bmatrix} G(j\omega) \\ I \end{bmatrix} > 0 \quad \forall \omega \in [0, \infty], \end{aligned} \quad (6.1)$$

or optimization problems:

$$\inf_{\lambda} c' \lambda, \quad \text{subject to the constraint in (6.1),} \quad (6.2)$$

where  $G(s) := (sI - A)^{-1}B$  is a rational transfer matrix. We will refer to (6.1) as IQC feasibility problems and (6.2) as IQC optimization problems due to their close connection to integral quadratic constraints.

The conventional way to treat standard IQC problems is to transform the frequency dependent matrix inequality in (6.1) into a non-frequency dependent one. The Kalman-Yakubovich-Popov lemma states that the frequency dependent matrix inequality holds if and only if there exists a symmetric matrix  $P$  such that

$$\begin{bmatrix} PA + A'P & PB \\ B'P & 0 \end{bmatrix} + \Sigma_0 + \sum_{i=1}^n \lambda_i \Sigma_i > 0. \quad (6.3)$$

Therefore, the inequalities in (6.1) and (6.2) can be equivalently expressed as (6.3). This transforms the IQC problems (6.1) and (6.2) into Semi-Definite Programs (SDPs) at the price of adding additional decision variables (i.e., the components of the matrix variable  $P$ ). Semi-definite programs can be viewed as generalizations of linear programs. Recently, many interior-point algorithms for solving linear programs have been extended to solve semi-definite programs [60, 75].

The number of additional decision variables which result from transforming the frequency dependent inequality is proportional to the square of the number of states in the system, or equivalently, the dimension of the matrix  $A$ . Therefore, when the number of states is substantially larger than the square root of the number of original decision variables, the additional decision variables play a dominant role in the equivalent SDP and become the major computational burden. Thus, most of the computational effort is spent on computing the auxiliary decision variables rather than the original ones. In this sense,

the conventional method becomes “inefficient“ in the case where the system to be analyzed has many states.

Complex systems having many states are not unusual to encounter. Typical examples of such systems are aircraft control systems, vibration controllers for flexible structures, etc.. In fact, the motivation behind developing advanced analysis methods, such as the standard IQC analysis method, was the need to systematically analyze such systems. Furthermore, the state space of the transfer matrix  $G(s)$  in (6.1) is the direct product of the state space of the physical system to be analyzed and the state spaces of the dynamical multipliers, if any, used in IQCs. Sometimes, in order to characterize the uncertainty in the system better, which consequentially will help to improve the accuracy of analysis, advanced dynamical multipliers which have many states are used. As a result, the resulting transfer matrix  $G(s)$  has a large state space. See [55, 36] for more discussions on the use of dynamical multipliers to improve the accuracy of the analysis and the involvement of the states of these dynamical multipliers in IQC optimization problems.

To improve the efficiency of the conventional approach, some specialized algorithms have been proposed recently [34, 77]. The basic idea of these algorithms is to explore and exploit the very special structure of the IQC problems and construct specialized interior point algorithms in which computation of directions of descent is performed efficiently. In [34], it is reported that such a specialized algorithm can solve an IQC optimization problem with about 5000 decision variables in 10 minutes. It is much faster than the general purpose SDP solver of the MATLAB LMI Control Toolbox.

On the other hand, since the inefficiency of the conventional method is due to the existence of auxiliary decision variables, another way to improve the efficiency is to avoid introducing the matrix variable  $P$ . Along this line of thought, the cutting plane method is one of the prospective alternatives. In this chapter, several cutting plane algorithms are implemented to solve standard IQC problems. A cutting plane algorithm generally has three major steps. First, an outer approximation of the original problem is chosen. A trial point is then selected by applying certain operations to the outer approximation. Finally, the trial point is used by a mechanism, often called the *oracle*, to produce cutting hyperplanes to improve the outer approximation. A cutting plane algorithm iterates these three steps until a suboptimal solution with certain desired accuracy is obtained.

The first cutting plane algorithm we implement to solve the standard IQC problems is

the Kelley type cutting plane algorithm presented in Chapter 5. Numerical experiments show that the algorithm works very well in some cases but suffers from slow convergence from time to time. This motivates us to consider another type of cutting plane algorithm - the centering method. Performance of the centering method generally does not vary too much when it is applied to solve a set of problems of the same size.

Many different centering methods have been proposed. One of the well-known centering methods is the ellipsoid method [1, 32] where ellipsoids are used to outer-approximate the feasible set of the original problem, and centers of ellipsoids are selected as the trial points. The advantage of the ellipsoid method is that it takes very little computational effort to generate the trial points. However, practical experience shows that it generally requires many iterations to converge. The recently developed Analytical Center Cutting Plane Method (ACCPM) [72, 27, 24, 2, 59] uses polyhedrons as outer approximations of the feasible set of the original problem and analytical centers of polyhedrons as the trial points. Although the ACCPM requires more computational effort to generate trial points and its estimated worst-case complexity is higher than the complexity of the ellipsoid algorithm, it appears to be an efficient algorithm according to practical experience. We have implemented an ellipsoid algorithm, which is slightly different from the original version, and a version of ACCPM to solve standard IQC problems.

The most essential part of a cutting plane algorithm is the oracle which is used to determine whether a given point satisfies the constraint in (6.1) and, if not, produces a hyperplane to separate the point from the feasible set. We will show the oracle for standard IQC problems can be constructed using a well-known result from the systems and control literature. The oracle requires low computational effort. This is the main reason why cutting plane algorithms are able to outperform the conventional method. Using numerical examples, we compare the three cutting plane algorithms with the conventional approach. As expected, the results of numerical experiments indicate that when the size of the matrix  $A$  is significantly larger than the number of decision variables, the cutting plane algorithms clearly outperform the conventional method. Previous work in the direction of constructing an efficient algorithm for solving standard IQC problems via avoiding the introduction of  $P$  matrix includes [61, 44].

The chapter is organized as follows: following the introduction section, we present a motivating example in Section 6.2 to show that more efficient computational methods to



solve IQC optimization problems are in need. In Section 6.3, we discuss how to use the Kalman-Yakubovich-Popov lemma [92] to construct the oracle which is required for applying cutting plane algorithms to solve IQC optimization problems. In Section 6.4, the results of applying Kelley type cutting plane algorithm to solve the problem in Section 6.2 is demonstrated. The Kelley type cutting plane algorithm outperforms the conventional approach in many cases, but its performance is not very consistent. This motivates us to consider the ellipsoid algorithm presented in Section 6.5 and the analytical center cutting plane method presented in Section 6.6. The cutting planes algorithms are tested on several numerical examples. The results are presented in Section 6.7. In Section 6.8, we discuss the computational complexities of these cutting plane algorithms and explain why the cutting plane method is much more efficient than the conventional method in certain cases from the point of view of the theoretical worst-case complexity. Finally, some remarks are given in Section 6.9 to conclude this chapter.

In the rest of this section, we introduce a couple of notations which will be used throughout this chapter and formally state the standard IQC problem for which we would like to find efficient algorithms.

## Notations and Problem Formulation

The feasibility problems arising from IQC analysis are of the forms

$$\text{find } \lambda \quad \text{such that} \quad \mathbf{H}(\omega, x) := \mathbf{H}_0(\omega) + \sum_{i=1}^n \lambda_i \mathbf{H}_i(\omega) > 0, \quad \forall \omega \in [0, \infty], \quad (6.4)$$

while the optimization problems are of the form

$$\inf_{\lambda} c' \lambda \quad \text{subj. to} \quad \mathbf{H}(\omega, \lambda) := \mathbf{H}_0(\omega) + \sum_{i=1}^n \lambda_i \mathbf{H}_i(\omega) > 0, \quad \forall \omega \in [0, \infty], \quad (6.5)$$

where each  $\mathbf{H}_i(\omega)$  is a self-adjoint, rational transfer matrix in the following form

$$\mathbf{H}_i(\omega) = \begin{bmatrix} (j\omega I - A)^{-1} B \\ I \end{bmatrix}^* \begin{bmatrix} Q_i & F_i \\ F_i' & R_i \end{bmatrix} \begin{bmatrix} (j\omega I - A)^{-1} B \\ I \end{bmatrix}, \quad i = 0, 1, \dots, n. \quad (6.6)$$

Matrices  $Q_i$  and  $R_i$  in (6.6) are symmetric. Matrix  $A$  is a Hurwitz matrix; i.e., none of its eigenvalues is in the closed right-half complex plane. Let

$$\Sigma_i = \begin{bmatrix} Q_i & F_i \\ F_i' & R_i \end{bmatrix}, \quad i = 1, \dots, n.$$

We assume that  $\Sigma_i$  are linearly independent. This ensures that none of the decision variable  $\lambda_i$  can be removed. The notations  $Q(\lambda)$ ,  $F(\lambda)$ , and  $R(\lambda)$  are used to denote

$$Q_0 + \sum_{i=1}^n \lambda_i Q_i, \quad F_0 + \sum_{i=1}^n \lambda_i F_i, \quad R_0 + \sum_{i=1}^n \lambda_i R_i, \quad (6.7)$$

respectively. The feasible set of the IQC problems is denoted by  $\Omega$ ; i.e.,

$$\Omega = \{\lambda \mid \mathbf{H}(\omega, \lambda) > 0, \forall \omega \in [0, \infty]\}. \quad (6.8)$$

Note that feasibility problem (6.4) can be solved as a spectral value maximization problem

$$\sup_{(\lambda, y)} y, \quad \text{subj. to } \mathbf{H}(\omega, \lambda) - yI > 0 \quad \forall \omega \in [0, \infty].$$

Any suboptimal solution  $(\bar{\lambda}, \tilde{y})$  with  $\tilde{y} > 0$  will give us a solution  $\bar{\lambda}$  to problem (6.4). Notice that the above problem has exactly the same setup as the optimization problem (6.5). Therefore, without loss of generality, we will only consider problem (6.5) through the rest of the paper.

In this chapter, we will present a number of computation algorithms that can be used to solve problems (6.4) and (6.5) in an efficient fashion.

## 6.2 A Motivating Example

In this section, we present an example to show that more efficient computational methods than the conventional approach of solving IQC optimization problems are in need.

Consider the standard block diagram for robustness analysis in Figure 6-1. The nominal

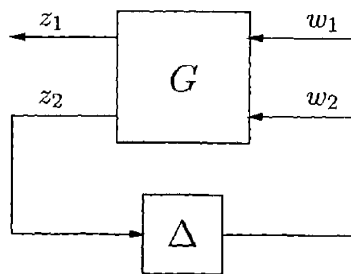


Figure 6-1: Standard block diagram for robustness analysis.

system  $G$  is linear time-invariant and has a state space representation

$$\begin{aligned} \dot{x} &= Ax + B_1 w_1 + B_2 w_2, \\ z_1 &= C_1 x + D_{11} w_1 + D_{22} w_2, \\ z_2 &= C_2 x + D_{21} w_1 + D_{22} w_2, \end{aligned}$$

where  $A$  is a  $n_s \times n_s$  Hurwitz matrix, and  $w_1$ ,  $w_2$ ,  $z_1$ ,  $z_2$  are vector-valued signals. Each of them has  $n$  components. The uncertainty  $\Delta$  corresponds to a diagonal, gain bounded, linear time-varying operator. That is, if  $z_{2i}$  and  $w_{2i}$  denote the  $i^{\text{th}}$  components of signals  $z_2$  and  $w_2$  respectively, then  $z_{2i} = \delta_i(t)w_{2i}$ , where  $|\delta_i(t)| \leq 1$  for all  $t$ . We note that the uncertain system described above captures a large class of practical problems [4].

In this example, we would like to compute an upper bound of the  $L_2$ -gain of the system in Figure 6-1. By the standard IQC analysis, an upper bound of the  $L_2$ -gain can be found by solving

$$\begin{aligned} \inf_{\lambda} \quad & \lambda_{n+1}, \\ \text{subj to} \quad & H(\omega, \lambda) > 0, \quad \forall \omega \in [0, \infty], \\ & \lambda_i > 0, \quad i = 1, \dots, n+1, \end{aligned} \tag{6.9}$$

where

$$H(\omega, \lambda) := \begin{bmatrix} G_{11}(j\omega) & G_{12}(j\omega) \\ G_{21}(j\omega) & G_{22}(j\omega) \\ I & 0 \\ 0 & I \end{bmatrix}^* \begin{bmatrix} -I & 0 & 0 & 0 \\ 0 & -\Lambda & 0 & 0 \\ 0 & 0 & \lambda_{n+1}I & 0 \\ 0 & 0 & 0 & \Lambda \end{bmatrix} \begin{bmatrix} G_{11}(j\omega) & G_{12}(j\omega) \\ G_{21}(j\omega) & G_{22}(j\omega) \\ I & 0 \\ 0 & I \end{bmatrix},$$

$G_{rs}(j\omega) = C_r(j\omega I - A)^{-1}B_s + D_{rs}$ , and  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ . Conventionally, one solves problem (6.9) by transforming the problem into its equivalent SDP formulation and then solving the SDP using the interior point method. The equivalent SDP of problem (6.9) can be expressed as

$$\begin{aligned} \inf_{P, \lambda} \quad & \lambda_{n+1}, \\ \text{subj to} \quad & S(P, \lambda) > 0, \\ & P = P', \quad \lambda_i > 0, \quad i = 1, \dots, n+1, \end{aligned} \tag{6.10}$$

where  $P$  is a matrix variable, and

$$S(P, \lambda) := \begin{bmatrix} PA + A'P & PB \\ B'P & 0 \end{bmatrix} + \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}' \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}.$$

Matrices  $B$ ,  $C$ ,  $D$ ,  $M_1$ , and  $M_2$  are defined as follows

$$B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}, \quad C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}, \quad D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}, \quad M_1 = \begin{bmatrix} -I & 0 \\ 0 & -\Lambda \end{bmatrix}, \quad M_2 = \begin{bmatrix} \lambda_{n+1}I & 0 \\ 0 & \Lambda \end{bmatrix}.$$

Let  $n = 10$  and  $n_s = 10, 20, \dots, 70$ . For each pair of  $(n, n_s)$ , five problems of the form in (6.10) are randomly generated and solved using the MATLAB LMI Control Toolbox. The average time spent on solving each set of problems is shown in Figure 6-2. These problems were solved on a Pentium III 800MHz machine with 256MB of memory. The operating system of the machine is LINUX and the version of MATLAB is 5.3.1.

As we can see from Figure 6-2, the time that LMI Control Toolbox took to solve a problem grows *rapidly* as the number of states (i.e.,  $n_s$ ) grows. This can be expected since the number of decision variables in  $P$  is proportional to  $n_s^2$ . When  $n_s = 70$ , the optimization problem has totally 2496 decision variables, and 2485 of them are from the matrix variable  $P$ . The largest problem we have ever tested has 100 states (totally 5061 decision variables). The problem took the LMI Control Toolbox more than 10 hours to solve.

Robustness analysis problems which have large state spaces are not unusual to encounter. An aircraft control system, for example, can easily have 30 to 40 states [4]. Furthermore, the state space of  $G$  is the direct product of the state space of the physical system to be analyzed and the state spaces of the multipliers used to defined the IQCs. If the multipliers have

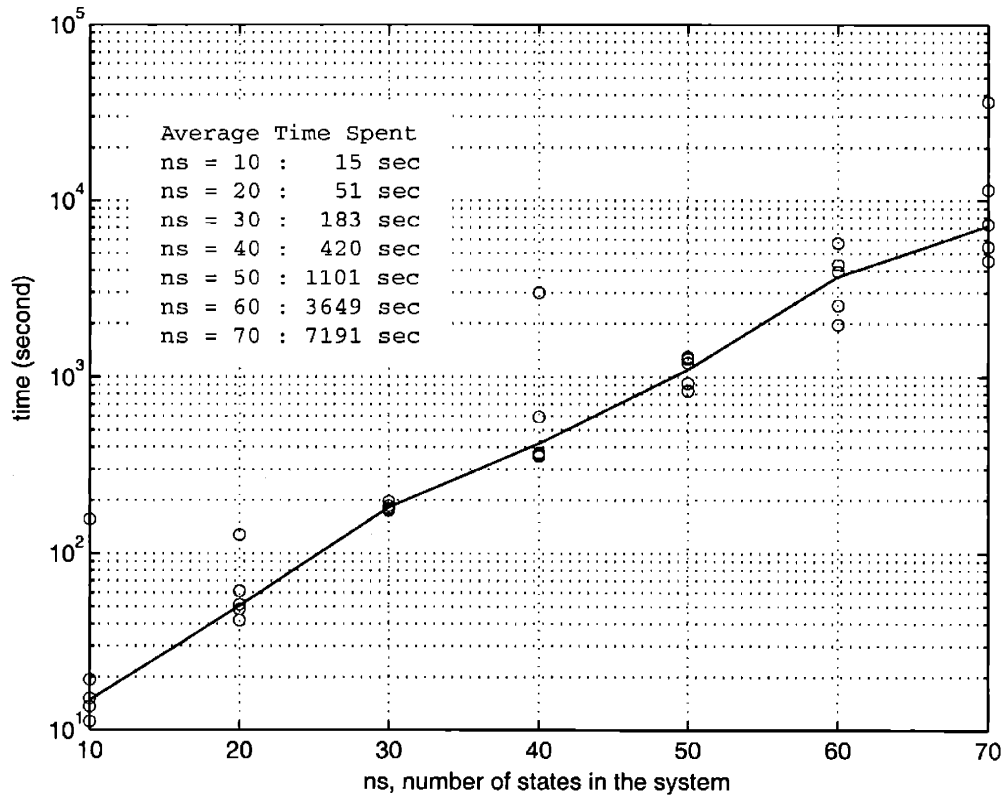


Figure 6-2: The figure shows the amount of time that the MATLAB LMI Control Toolbox took to solve SDP (6.10). For each  $n_s$ , five testing problems were randomly generated. As we can see, the amount of time that MATLAB LMI Control Toolbox took to solve a problem increases almost exponentially as  $n_s$  grows. It took approximately two hours for the LMI Control Toolbox to solve a problem with 70 states.

non-trivial state spaces, then the number of states in  $G$  is the sum of states in the physical system and states in the multipliers. Sometimes, in order to characterize the uncertainty in the system better, which consequentially will help to improve the accuracy of analysis, advanced dynamical multipliers which have many states are used. As a result, the resulting transfer matrix  $G$  has a large state space, and the corresponding SDP have a huge number of decision variables, most of which are from the auxiliary matrix variable  $P$ . We can infer from the results of the numerical experiments presented above that in this case it requires substantial computational capacity and takes a long time to solve the corresponding SDP. Furthermore, the computational effort is mostly spent on computing the auxiliary decision variables. In this sense, the conventional approach is very “inefficient”.

Since the inefficiency of the conventional approach is mainly due to the introduction of matrix variable  $P$ , an obvious approach to improve efficiency is to avoid introducing it. Along this line of thought, the cutting plane method is a prospective candidate. Take the Kelley type cutting plane algorithm presented in the previous chapter for example. In order to apply the cutting plane algorithm to solve IQC optimization problems, all we need is an oracle which can check whether a given point satisfies the frequency dependent matrix inequality in (6.5) or not and generates cutting hyperplanes accordingly. As long as no auxiliary decision variable is required for the computational procedures involved in the oracle, the cutting plane algorithm should be more efficient than the conventional method, at least when solving a problem which has many states.

In the next section, we show that an efficiently computable oracle for IQC optimization problem (6.5) can be constructed using a well-known result in the field of systems and control.

### 6.3 The Oracle for the Standard IQC Optimization Problem

To solve IQC optimization problem (6.5) using the Kelley type cutting plane algorithm presented in the previous chapter, one requires an oracle which checks whether a given  $(\tilde{\lambda}, \tilde{y})$  satisfies

$$\mathbf{H}(\omega, \tilde{\lambda}) > \tilde{y}I, \quad \forall \omega \in [0, \infty], \quad (6.11)$$

and, if not, generates cutting hyperplanes to separate  $(\tilde{\lambda}, \tilde{y})$  from the set of points which satisfy (6.11). The following well-known result can be used to construct such an oracle.

**Lemma 6.1 (Kalman, Yakubovich, Popov, ...).** *Assume the pair of matrices  $(A, B)$  is stabilizable<sup>1</sup>. For a given point  $\tilde{\lambda}$ , let  $\tilde{Q} = Q(\tilde{\lambda})$ ,  $\tilde{F} = F(\tilde{\lambda})$ , and  $\tilde{R} = R(\tilde{\lambda})$ , where  $Q(\lambda)$ ,  $F(\lambda)$ ,  $R(\lambda)$  are defined in (6.7). The following statements are equivalent:*

1.  $\mathbf{H}(\omega, \tilde{\lambda}) > 0$  for all  $\omega \in [0, \infty]$ .

2. There exists a  $P = P'$  such that

$$\begin{bmatrix} PA + A'P & PB \\ B'P & 0 \end{bmatrix} + \begin{bmatrix} \tilde{Q} & \tilde{F} \\ \tilde{F}' & \tilde{R} \end{bmatrix} > 0.$$

3.  $\tilde{R} > 0$ , and the Hamiltonian  $\mathcal{H}$  defined below has no eigenvalues on the imaginary axis.

$$\mathcal{H} = \begin{bmatrix} A - B\tilde{R}^{-1}\tilde{F}' & B\tilde{R}^{-1}B' \\ \tilde{Q} - \tilde{F}\tilde{R}^{-1}\tilde{F}' & -A' + \tilde{F}\tilde{R}^{-1}B' \end{bmatrix}. \quad (6.12)$$

Furthermore, if the Hamiltonian matrix  $\mathcal{H}$  has eigenvalues on the imaginary axis, for example  $\pm j\omega_1, \dots, \pm j\omega_m$ , then  $\mathbf{H}(\omega, \tilde{\lambda})$  is singular at frequency  $\omega_1, \dots, \omega_m$ .

*Proof.* The proof can be found in a number of control theory textbooks. See, for example, [92].  $\square$

To check whether inequality (6.11) holds or not, one can use the third statement of Lemma 6.1. That is, one first checks whether  $\tilde{R} > \tilde{y}I$ . This can be performed by computing the eigenvalues of  $\tilde{R} - \tilde{y}I$  and checking whether all the eigenvalues are strictly positive. If some of the eigenvalues are not strictly positive, then  $\tilde{R} - \tilde{y}I$  is not strictly positive definite, which in turn implies that (6.11) is not satisfied. If  $\tilde{R} > \tilde{y}I$ , then one forms a Hamiltonian matrix  $\mathcal{H}$  which is defined as in (6.12) except that  $\tilde{R}$  is replaced by  $\tilde{R} - \tilde{y}I$  and computes the eigenvalues of  $\mathcal{H}$ . If none of the eigenvalues of  $\mathcal{H}$  is on the imaginary axis, then (6.11) is satisfied. Otherwise, the inequality does not hold.

<sup>1</sup>In any IQC optimization problem, matrix  $A$  is assumed to be a Hurwitz matrix. Therefore, the stabilizability condition is automatically satisfied.

If  $(\tilde{\lambda}, \tilde{y})$  does not satisfy inequality (6.11), cutting hyperplanes which separate  $(\tilde{\lambda}, \tilde{y})$  from the set of points that satisfies (6.11) can be derived as follows.

### 6.3.1 Generate Separating Hyperplanes

Suppose that  $\tilde{R} > \tilde{y}I$  is not satisfied. Then there exists at least one vector  $v$  with  $\|v\| = 1$ , such that

$$v' \tilde{R} v - \tilde{y} \leq 0. \quad (6.13)$$

Since  $R(\lambda)$  is affine in  $\lambda$ , we have  $v'R(\lambda)v = a'\lambda - b$  for some vector  $a$  and number  $b$ . More specifically, let  $a_i$  be the  $i^{\text{th}}$  component of  $a$ . Then  $a_i = v'R_i v$  and  $b = -v'R_0 v$ . Since any  $(\lambda, y)$  which satisfies (6.11) must also satisfy  $a'\lambda - b - y > 0$ , therefore the polyhedral constraint to be introduced is  $a'\lambda - b - y > 0$ .

Now suppose that  $\tilde{R} > \tilde{y}I$ , but the Hamiltonian matrix  $\mathcal{H}$  has eigenvalues on the imaginary axis, say  $\pm j\omega_1, \dots, \pm j\omega_m$ . This means that  $\mathbf{H}(\omega, \tilde{\lambda}) > \tilde{y}I$  is violated at each one of  $\omega_i, i = 1, \dots, m$ ; i.e.,  $\mathbf{H}(\omega_i, \tilde{\lambda}) - \tilde{y}I$  is not strictly positive definite. Then the separating hyperplanes can be derived as in the case when  $\tilde{R} > \tilde{y}I$  is not satisfied.

Usually, there may be more than one eigenvalue of  $\tilde{R} - \tilde{y}I$  which are not strictly positive, or more than one pair of eigenvalues of  $\mathcal{H}$  which are on the imaginary axis. In either of these cases, we should use all the information available to obtain as many separating hyperplanes as possible.

The oracle described in this section can be used to construct various cutting plane algorithms for solving IQC optimization problem (6.5). Note that the major computation involved in the oracle is the eigenvalue decomposition, for which efficient computational routines are widely available.

## 6.4 The Results of Solving Problem (6.9) Using the Kelley Type Cutting Plane Algorithm

The first cutting plane algorithm we implemented to solve the IQC optimization problems is the Kelley type of cutting plane algorithm. The algorithm is exactly the same as the one presented in Section 5.2 except that the oracle is replaced by the one suitable for



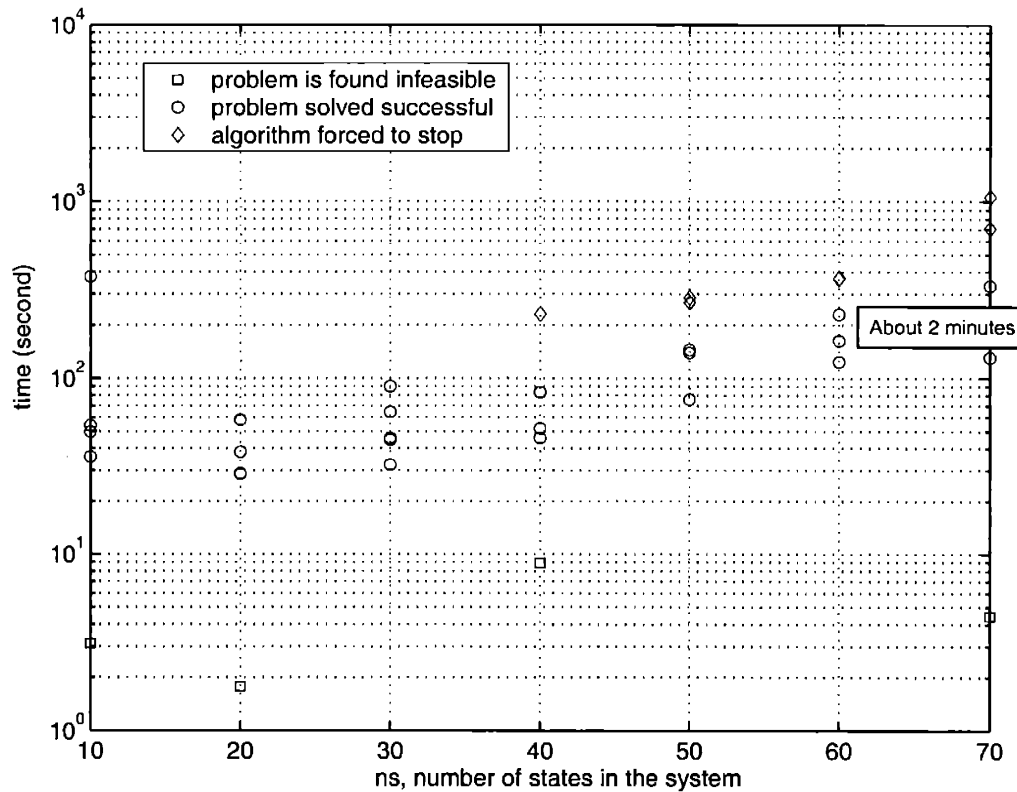


Figure 6-3: The figure shows the amount of time that the Kelley type cutting plane algorithm took to solve problem (6.9). For each  $n_s$ , five testing problems were randomly generated. Compared with Figure 6-2, we see that the amount of time that the Kelley type cutting plane algorithm took to solve a problem is less than the conventional method. Furthermore, as the number of states in a problem increases, the difference in speed becomes more significant. For  $n_s = 70$ , the cutting plane algorithm took only a few minutes to solve a problem which the conventional method took approximately two hours to solve.

problem (6.5). We tested the algorithm on the  $L_2$ -gain estimation problem in Section 6.2. Again, a set of problems is randomly generated for each pair of  $(n, n_s)$ , where  $n = 10$  and  $n_s = 10, 20, \dots, 70$ . The amount of time the cutting algorithm spent to solve each problem is shown in Figure 6-3.

Compared with Figure 6-2, we see that the cutting plane algorithm is significantly faster than the conventional method when  $n_s$  (the number of states in the problem) is larger than 30. Furthermore, as  $n_s$  becomes larger, the difference in speed becomes more significant. When  $n_s = 50$ , the cutting plane algorithm is roughly 10 times faster than the conventional method. When  $n_s = 70$ , it took the cutting plane algorithm only a couple of minutes to

solve the problems which the conventional method took over an hour to solve. We also notice that the amount of time required for the cutting plane algorithm to solve a problem increase fairly slowly as the number of states in the problem grows.

On the other hand, Figure 6-3 also indicates that the performance of the Kelley type cutting plane algorithm is not quite consistent. For instance, at  $n_s = 10$ , there is one problem for which the algorithm took longer time to solve. In addition, at  $n_s = 40, 50, 60, 70$ , there are cases where the algorithm converges very slowly so that the algorithm was forced to terminate before it found the solution with desired accuracy.

Slow convergence is a known problem for all cutting plane algorithms. In the case of the Kelley type of cutting plane algorithm, the estimated number of iterations for the algorithm to converge to an  $\varepsilon$ -accurate suboptimal solution (i.e., the difference between the objective at the suboptimal solution and the optimal objective is less than  $\varepsilon$ .) is  $\mathbf{O}(1/\varepsilon^n)$  [11], where  $n$  is the number of decision variables. Therefore, from the worst-case complexity point of view, the Kelley type of cutting plane algorithm potentially could work very poorly. Some of our numerical experiments agree with this point.

An heuristic argument for slow convergence of the Kelley type of cutting plane algorithm is as follows: the algorithm uses extreme points of polyhedral outer approximations as the trial points. The extreme point is a good choice when the outer approximation is very close to the original problem. However, the extreme point could also turn out to be almost irrelevant when the outer approximation is far from accurate. In this case, the improvement on the outer approximation is usually little, and the Kelley type cutting plane algorithm converges slowly. On the contrary, cutting plane algorithms which use certain centers as trial points generally have no such problem. The reason is that centers are generally less sensitive to the introduction of cutting planes, and therefore, using them as trial points usually results in steady improvement on the outer approximations. Hence, when the approximation is far from accurate, this class of cutting plane algorithms converges more rapidly than the Kelley type cutting plane algorithm. This motivates us to consider the centering methods presented in the next two sections, the ellipsoid algorithm and the Analytical Center Cutting Plane Method (ACCPM).

## 6.5 The Ellipsoid Algorithm

One of the best known cutting plane algorithms which uses centers as trial points is the ellipsoid algorithm. The ellipsoid algorithm was first developed in the 1970s in the former Soviet Union by Shor, Yudin, and Nemirovsky. A detailed history of its development can be found in [1]. The algorithm was used by Khachiyan [48] in his famous proof that linear programs can be solved in polynomial time. The algorithm and its variations have been well studied. A good reference of this subject is [32]. In [11], it was shown that many control related engineering problems can be solved using the cutting plane method, and the ellipsoid algorithm is specifically discussed. In the next subsection, we briefly describe the basic principles of the algorithm. The material is adapted from Chapter 14 of [11].

### 6.5.1 The Basic Algorithm

Consider the constrained minimization problem

$$\inf c'\lambda, \quad \text{subj. to } \lambda \in \Lambda \quad (6.14)$$

where  $\lambda \in \mathbf{R}^n$  and  $\Lambda \subset \mathbf{R}^n$  is an open convex set. We assume the convex set  $\Lambda$  is described by a separating oracle. That is, there exists a mechanism which can be used to test whether a given  $\tilde{\lambda}$  belongs to  $\Lambda$  or not. Furthermore if  $\tilde{\lambda}$  does not belong to  $\Lambda$ , the mechanism generates a hyperplane which separates  $\tilde{\lambda}$  from  $\Lambda$ .

The basic idea of the ellipsoid algorithm goes as follows. Suppose that we have an ellipsoid  $E_0$  that is guaranteed to contain the optimal solution of (6.14). In the algorithm, the center (denoted by  $\lambda_0$ ) of  $E_0$  is submitted to the oracle for feasibility test. If  $\lambda_0$  is feasible, then obviously the sliced half ellipsoid  $E_0 \cap \{\lambda \mid c'\lambda \leq c'\lambda_0\}$  contains the optimal solution of (6.14). We then compute the ellipsoid  $E_1$  of minimum volume that contains the sliced half ellipsoid; we call this an *objective* iteration and refer to the hyperplane  $c'\lambda = c'\lambda_0$  as an *optimality* cut. If  $\lambda_0$  is not feasible, the oracle returns a hyperplane  $a'\lambda = b$  such that the feasible set  $\Lambda$  is contained in the half ellipsoid  $E_0 \cap \{\lambda \mid a'\lambda \leq b\}$ . Then, again, the ellipsoid  $E_1$  of minimum volume that contains the sliced ellipsoid is computed. We call this a *constraint* iteration and refer to the hyperplane  $a'\lambda = b$  as a *feasibility* cut. In a constraint iteration, the points discarded are all infeasible. In an objective iteration, the objective values at all the points discarded are greater than the objective value at the current feasible

point. Thus, in each case, we do not discard any optimal solutions, and  $E_1$  always contains all optimal solutions that are in  $E_0$ . The ellipsoid algorithm iterates this procedure and generates a sequence of ellipsoids which guarantee to contain an optimal solution of (6.14). Furthermore, the ellipsoid generated in an iteration is guaranteed to be smaller than its counterpart in the previous iteration by a factor. As the ellipsoids become smaller and smaller, the algorithm eventually finds a suboptimal solution with certain desired accuracy. The basic ellipsoid algorithm is summarized as below.

**Initialization:** select  $\lambda_0$  and  $E_0$ , where  $E_0$  is any initial ellipsoid that contains  $\Lambda$  (if  $\Lambda$  is non-empty), and  $\lambda_0$  is the center of  $E_0$ . Set  $U_0$ , the upper bound of the optimal objective, to be an empty set. Set  $k := 0$ .

**Repeat**

Submit  $\lambda_k$  to the oracle for feasibility checking.

If  $\lambda_k \in \Lambda$ ,

1. Compute  $(\lambda_{k+1}, E_{k+1})$ , where  $E_{k+1}$  is the minimum volume ellipsoid which contains  $E_k \cap \{\lambda \mid c'\lambda \leq c'\lambda_k\}$ , and  $\lambda_{k+1}$  is the center of  $E_{k+1}$ .
2. Set  $U_{k+1} := c'\lambda_k$ .

Else

1. The oracle returns a hyperplane  $a'\lambda = b$  such that the feasible solutions satisfy  $a'\lambda < b$  and the center  $\lambda_k$  satisfies  $a'\lambda_k \geq b$ .
2. Compute  $(\lambda_{k+1}, E_{k+1})$ , where  $E_{k+1}$  is the minimum volume ellipsoid which contains  $E_k \cap \{\lambda \mid a'\lambda \leq b\}$ , and  $\lambda_{k+1}$  is the center of  $E_{k+1}$ .
3. Set  $U_{k+1} := U_k$ .

Endif

Set  $k := k + 1$ .

**Until** (the stopping criterion is satisfied).

### Generating New Ellipsoids

Let  $E$  be an ellipsoid defined as  $\{\lambda \mid (\lambda - \lambda_0)'A^{-1}(\lambda - \lambda_0) \leq 1\}$  and  $P$  be a half plane  $\{\lambda \mid a'\lambda \leq b\}$ . The minimum volume ellipsoid which contains  $E \cap P$  can be found using the

following proposition. The proposition is well-known and can be found in the programming literature. For instance, see Chapter 3 of [32].

**Proposition 6.1.** *Let*

$$\alpha = \frac{a'\lambda_0 - b}{\sqrt{a'Aa}}, \quad \beta = \frac{Aa}{\sqrt{a'Aa}}.$$

*Let the minimum-volume ellipsoid that contains  $E \cap P$  be  $\hat{E}$ . Then  $\hat{E}$  can be determined according to the value of  $\alpha$ .*

1. *If  $\alpha > 1$ , then the intersection of  $E$  and  $P$  is empty.*
2. *If  $\alpha = 1$ , then the intersection of  $E$  and  $P$  is a point, given by  $\lambda_0 - \beta$ .*
3. *If  $\alpha \leq -\frac{1}{n}$ , then  $\hat{E} = E$ .*
4. *If  $-\frac{1}{n} < \alpha < 1$ , then  $\hat{E} = \{\lambda \mid (\lambda - \hat{\lambda}_0)'\hat{A}^{-1}(\lambda - \hat{\lambda}_0)\}$ , where*

$$\hat{\lambda}_0 = \lambda_0 - \frac{1 + \alpha n}{1 + n}\beta,$$

$$\hat{A} = \frac{n^2}{n^2 - 1}(1 - \alpha^2) \left( A - \frac{2(1 + \alpha n)}{(n + 1)(1 + \alpha)}\beta\beta' \right).$$

*Proof.* See Chapter 3 of [32] for a complete proof of the proposition. □

In each iteration of the ellipsoid algorithm, we first compute the corresponding  $\alpha$  and then produce the minimum volume ellipsoid according to the value of  $\alpha$ . If  $\alpha > 1$ , then the intersection of the separating hyperplane and  $E_k$  is empty. In this case, we stop the algorithm immediately. Notice that if the problem is feasible, this case is not supposed to happen since we assume that the initial ellipsoid  $E_0$  contains all optimal solutions. The case of  $\alpha = 1$  is very rare, and virtually impossible to encounter. If it does happen, we find the exact optimal solution.

The third case of Proposition 6.1 never happens in the ellipsoid algorithm. To see this, suppose that the algorithm is in an objective iteration; i.e.,  $\lambda_k$ , the center of ellipsoid  $E_k$ , is feasible. Then the cutting hyperplane for computing the next ellipsoid is  $c'\lambda = c'\lambda_k$ , and the corresponding  $\alpha$  is 0. On the other hand, suppose that algorithm is in a constraint iteration; i.e.,  $\lambda_k$  is not feasible. In this case, the cutting hyperplane  $a'\lambda = b$  satisfies  $a'\lambda_k \geq b$  which in turn implies the corresponding  $\alpha$  is greater than or equal to 0.

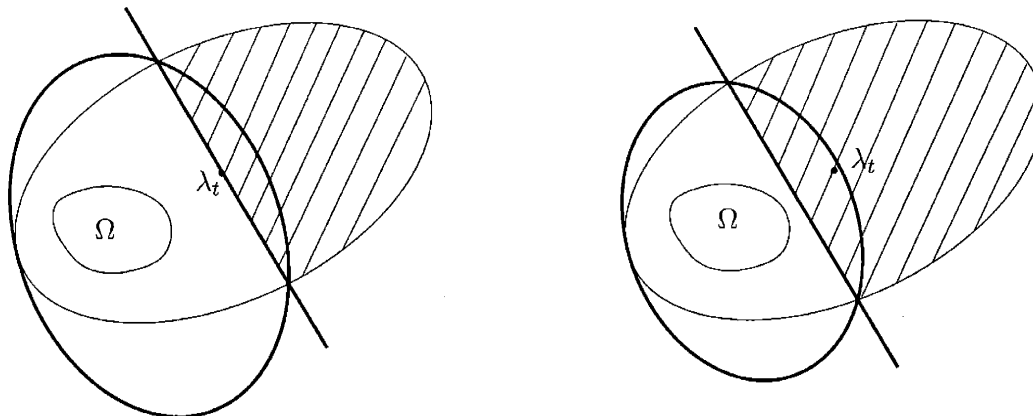


Figure 6-4: Illustration of central cuts and deep cuts. Let  $\Omega$  be the feasible set and  $\lambda_t$  be a trial point. The shaded regions are the parts of ellipsoids removed by the cutting planes. Figure on the left-hand-side: a central cut is a cutting hyperplane that passes through the test point  $\lambda_t$ . Figure on the right-hand-side: a deep cut passes between the test point and the feasible set and cuts off a larger piece of the outbound set.

The last case in Proposition 6.1 is what usually happens in most of the iterations of the ellipsoid algorithm. As discussed above, the value of  $\alpha$  is always greater than or equal to 0. If  $\alpha = 0$ , which implies that the cutting hyperplane passing through the center of the ellipsoid, we call the cutting hyperplane a *central cut*. Notice that in any objective iterations, the cut is central. If  $\alpha > 0$ , we refer to the hyperplane as a *deep cut*. In contrast to the *central cut* which passes through the test point, a deep cut *strictly* separates the test point and the feasible set. See Figure 6-4 for an illustration. We see from Figure 6-4 that a central cut cuts exactly half of the ellipsoid away, while a deep cut cuts more than a half.

### The Stopping Criterion

The following assumptions on the feasible set  $\Lambda$  is made:  $\Lambda$  is either bounded and of full dimension, or empty. Furthermore, if  $\Lambda$  is not empty, then it contains a ball of radius  $\varepsilon$ . These assumptions imply that if  $\Lambda$  is non-empty, then the volume of  $\Lambda$  is lower bounded by  $\pi_n \varepsilon^n$ , where  $\pi_n$  denotes the volume of the unit ball in  $\mathbf{R}^n$ .

Suppose that  $\Lambda$  is non-empty. Since we assume that an optimal solution is contained in  $E_0$ , we know that there always exists an optimal solution  $\lambda^* \in E_k$  for all  $k$ . Therefore, we

have

$$c'\lambda^* \geq \min_{\lambda \in E_k} c'\lambda. \quad (6.15)$$

Thus a lower bound on the optimal objective of (6.14) can be obtained by solving the minimization problem on the right-hand-side of (6.15). Let  $E_k := \{\lambda \mid (\lambda - \lambda_k)' A_k^{-1} (\lambda - \lambda_k) \leq 1\}$ . Then the right hand side of (6.15) is equal to  $c'\lambda_0 - \sqrt{c' A_k c}$ . Hence, the simple stopping criterion

$$\text{Until } (U_k - (c'\lambda_0 - \sqrt{c' A_k c}) < \epsilon) \quad (6.16)$$

guarantees that on exit, we find a suboptimal solution  $\tilde{\lambda}$  such that the difference between  $c'\tilde{\lambda}$  and the optimal objective is less than or equal to  $\epsilon$ .

Suppose that up to the  $k^{\text{th}}$  iteration, there is no feasible point found. Then the volume of  $E_k$  is checked to determine whether the algorithm is to be terminated. Since we assume that the volume of  $\Lambda$  is at least  $\pi_n \epsilon^n$  if it is not empty, and ellipsoid  $E_k$  contains  $\Lambda$ , therefore, the volume of  $E_k$  has to be at least  $\pi_n \epsilon^n$  if  $\Lambda$  is not empty. Thus if the volume of  $E_k$  is smaller than  $\pi_n \epsilon^n$ , then  $\Lambda$  is empty. The volume of  $E_k$  is equal to  $\pi_n \sqrt{\det(A_k)}$ , where  $\det(A_k)$  denotes the determinate of  $A_k$ . Hence, the criterion for terminating the algorithm is

$$\text{Until } (\sqrt{\det(A_k)} < \epsilon^n). \quad (6.17)$$

As soon as the inequality in (6.17) is satisfied and there is no feasible point found, the algorithm declares the problem is infeasible and stops.

To summarize, the ellipsoid algorithm uses (6.17) as a stopping criterion until a feasible point is found. When a feasible point is found, the algorithm starts to use (6.16) as the stopping criterion. On exit, criterion (6.16) guarantees an  $\epsilon$ -accurate suboptimal solution if the problem is feasible.

### Convergence and Complexity

It is well-known that the ellipsoid algorithm converges in polynomial time. More specifically, the algorithm terminates in  $\mathcal{O}(n^2 |\log \epsilon|)$  iterations with an  $\epsilon$ -accurate suboptimal solution

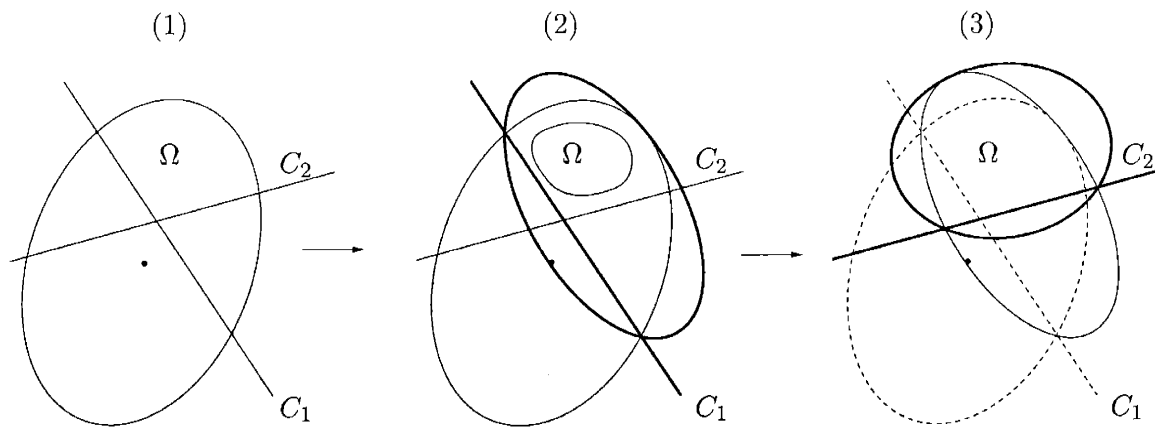


Figure 6-5: Illustration of the idea of using multiple cutting planes. Here,  $\Omega$  denotes the feasible set. (1) Cutting planes  $C_1$  and  $C_2$  are placed. (2) Cutting plane  $C_1$  is used to remove a part of the original ellipsoid and generate a new ellipsoid of smaller volume. (3) Cutting plane  $C_2$  is then used to remove a part of the newly generated ellipsoid.

found or with the declaration that  $\Lambda$  is empty. The proof is available in the literature. See for example, Chapter 14 of [11] or Chapter 3 of [32].

### 6.5.2 The Ellipsoid Algorithm for Solving IQC Optimization Problems

In this subsection, the ellipsoid algorithm we implement to solve IQC optimization problem (6.5) is presented. Our implementation follows the basic principles of the ellipsoid algorithm described in Section 6.5.1. However, there are a few small variations. The modifications are due to certain features of the oracle for the IQC optimization problem (6.5). As we have mentioned in Section 6.3, the oracle for IQC optimization problems is capable of generating more than one separating hyperplanes on a single inquiry. We take advantage of this feature and manage to use all cutting hyperplanes to produce the volume-reduced ellipsoid for the next iteration. The idea is as follows: suppose that the algorithm is in the  $k^{\text{th}}$  iteration and the oracle returns  $m$  hyperplanes  $a'_i \lambda = b_i$ ,  $i = 1, \dots, m$ . The optimal solutions is contained in  $E_k \cap P_1 \cap P_2 \cdots \cap P_m$ , where  $P_i = \{\lambda \mid a'_i \lambda \leq b_i\}$ . We first compute the ellipsoid of minimum volume which contains  $E_k \cap P_1$ . Denote this ellipsoid by  $\tilde{E}_1$ . Then we use the cutting hyperplane  $a'_2 \lambda = b_2$  to further reduce  $\tilde{E}_1$  and compute the ellipsoid of minimum volume which contains  $\tilde{E}_1 \cap P_2$ . Denote this ellipsoid by  $\tilde{E}_2$ . Then we reduce  $\tilde{E}_2$  using hyperplane  $a'_3 \lambda = b_3, \dots$ , and so on. Finally, the test point for the next iteration is the center of the minimum volume ellipsoid which contains  $\tilde{E}_{m-1} \cap P_m$ . Figure 6-5 illustrates the idea.



Another modification we made is that, the linear inequalities returned by the oracle are collected, and any new candidate for the optimal solution is subject to feasibility test with respect to these inequalities before submitted to the oracle. Since the optimal solution has to be feasible to all of these inequalities, a candidate which violates any of these inequalities can not be the optimal solution and should be disqualified immediately. The inequalities which the candidate violates can be immediately used as cuts to reduce the ellipsoid. Although our experience shows that trial points are often feasible to these inequalities, it is still worthwhile to perform the checking. The reason is that, compared to the computational complexity of the oracle, the complexity of checking feasibility with respect to these linear inequalities is very low. Therefore, if a trial point does not satisfy any of these inequalities, we obtain a cutting plane with very little computational effort spent.

It turns out that these modifications substantially speed up the algorithm. We have compared the modified ellipsoid algorithm with the basic one; i.e., the algorithm described in Section 6.5.1. The modified version is approximately three times faster than the basic ellipsoid algorithm.

We summarize our modified ellipsoid algorithm for solving IQC optimization problem (6.5) as follows.

1. Initialization: set the initial outbound ellipsoid  $E_0$  to be

$$E_0 := \{\lambda \in \mathbf{R}^n : \|\lambda\| \leq r\},$$

where  $r$  is a positive constant, say  $10^9$ . Set the first test point  $\lambda^{(0)} = 0$ . Set thresholds for stopping criteria. Let  $P_A$  be an empty set.

2. At the  $k^{\text{th}}$  iteration, submit  $\lambda^{(k)}$ , the candidate for the optimal solution, to the oracle for feasibility checking. If it is not feasible to the inequality in (6.5), the oracle returns a set of inequalities  $P_h = \{a'_i \lambda \leq b_i, i = 1, \dots, N_k\}$ . If it is feasible, let  $P_h$  be  $\{c' \lambda \leq c' \lambda^{(k)}\}$ . Attach  $P_h$  to  $P_A$ ; i.e., set  $P_A := P_A \cup P_h$ .
3. Produce a new, volume-reduced ellipsoid: let  $N_k$  be the number of constraints in  $P_h$ , and perform the following procedure.

Set  $\tilde{E}_0 = E_k$  and  $l = 1$ .

**While**  $l \leq N_k$

- (a) Let  $\tilde{P}_l = \{\lambda : a'_l \lambda \leq b_l\}$ ; i.e., the half-plane defined by the  $l^{\text{th}}$  inequality in the set  $P_h$ . Compute  $\tilde{E}_l$ , the minimum volume ellipsoid that contains  $\tilde{E}_{l-1} \cap \tilde{P}_l$ .
- (b) Compute  $\tilde{\lambda}^{(l)}$ , the center of  $\tilde{E}_l$ .
- (c) Set  $l := l + 1$ .

**until**  $l = N_k$

4. Check whether  $\tilde{\lambda}^{(N_k)}$  satisfies all inequalities in  $P_A$ . If it does not, then let  $P_h$  be the set of inequalities which  $\tilde{\lambda}^{(N_k)}$  violates, set  $E_k = \tilde{E}_{N_k}$ , and repeat from Step 3. If  $\tilde{\lambda}^{(N_k)}$  satisfies all inequalities in  $P_A$ , set  $E_{k+1} = \tilde{E}_{N_k}$  and  $\lambda^{(k+1)} = \tilde{\lambda}^{(N_k)}$ .
5. Check stopping criteria. If any stopping criterion is met, stop the program. Otherwise repeat from Step 2.

The most essential part of the algorithm is the oracle. The oracle can be constructed using the Kalman-Yakuvich-Popov lemma, which we have already presented in Section 6.3. In Step 5, the stopping criteria are exactly as what we discussed in Section 6.5.1. The modified ellipsoid algorithm described above has been implemented in MATLAB and tested on several numerical examples, including the one presented in Section 6.2. These results are presented in Section 6.7.

The most favorable aspect of the ellipsoid algorithm is that it takes very little computational effort to generate trial points. However, the algorithm usually requires many iterations to converge to a suboptimal solution with moderate accuracy. This motivated us to consider another centering method, the Analytical Center Cutting Plane Method (ACCPM). Practical experience shows that this particular centering method has very good performance in terms of rate of convergence. We present this algorithm in the next section.

## 6.6 Analytical Center Cutting Plane Method

The concept of analytical center was introduced by Sonnevend [72] who also alluded to its use in the cutting plane methods. The Analytical Center Cutting Plane Method (ACCPM) and its implementation were then proposed by Goffin, Haurie, and Vial [24], and Ye [88]. After that, theory underlying the ACCPM has been studied in depth by several authors, who provided estimates of complexity for the basic method and several of its enhancements

[25, 89, 26, 29, 31]. The method has been implemented in C/C++. The package is available at <http://ecolu-info.unige.ch/logilab> with a tutorial guide [62]. ACCPM has been applied to solve various type of programming problems. Judging from the practical experience, it appears that the performance of the algorithm is very good [23, 3]. An article by Goffin and Vial [28] gives a detailed survey of the recent development of ACCPM.

In the next subsection, we briefly introduce the analytical center cutting plane method. The contents presented in the next subsection is adapted and summarized from the material in the references mentioned above.

### 6.6.1 The Algorithm

Consider again the minimization problem (6.14)

$$\inf c'\lambda, \quad \lambda \in \Lambda.$$

The underlying principles of the analytical center cutting plane method are exactly the same as those of the ellipsoid algorithm: the algorithm first selects an outer approximation of the feasible set  $\Lambda$ ; i.e., a set  $\tilde{\Lambda}$  which contains  $\Lambda$ . Then a trial point inside  $\tilde{\Lambda}$  is selected for feasibility checking. Feasibility checking is performed by the oracle, and an optimality or a feasibility cut is returned to reduce the approximation  $\tilde{\Lambda}$ . The algorithm iterates this procedure until a suboptimal solution with certain desired accuracy is obtained, or the algorithm determines that the problem is infeasible.

In the ellipsoid algorithm, the outer approximations are ellipsoids, and the trial points are the centers of these ellipsoids. In the analytical center cutting plane method, the outer approximations are polyhedrons defined by linear inequalities

$$P_k := \{\lambda \mid a'_i\lambda - b_i \geq 0, \quad \|a_i\| = 1, \quad i = 1, \dots, N_k\},$$

and the trial points are  $\varepsilon$ -approximate analytical centers of these polyhedrons. The analytical center of  $P_k$  is defined as the unique minimizer of the logarithmic function

$$F_k(\lambda) = - \sum_{i=1}^{N_k} \log(a'_i\lambda - b_i), \quad \lambda \in \text{Int}(P_k),$$

where  $\text{Int}(P_k)$  denotes the interior of  $P_k$ . Given an  $\varepsilon \in (0, 1)$ , an  $\varepsilon$ -approximation of the

analytical center, or  $\varepsilon$ -center for short, is a point  $\tilde{\lambda}$  which satisfies

$$\rho_k(\tilde{\lambda}) := \left( \nabla F_k(\tilde{\lambda})' \nabla^2 F_k(\tilde{\lambda}) \nabla F_k(\tilde{\lambda}) \right)^{\frac{1}{2}} \leq \varepsilon, \quad (6.18)$$

where  $\nabla F_k(\lambda)$ ,  $\nabla^2 F_k(\lambda)$  denote the gradient and Hessian of  $F_k(\lambda)$ , respectively. The algorithm is summarized as follows.

**Initialization:** select  $\lambda_0$  and  $P_0$ , where  $P_0$  is any initial polyhedron that contains  $\Lambda$  (if it is non-empty), and  $\lambda_0$  is an  $\varepsilon$ -center of  $P_0$ . Set  $U_0$ , the upper bound of the optimal objective, to be an empty set. Set  $k := 0$ .

**Repeat**

Submit  $\lambda_k$  to the oracle for feasibility checking.

If  $\lambda_k \in \Lambda$ ,

1. Let  $P_{k+1} := P_k \cap \{\lambda \mid c'\lambda \leq c'\lambda_k\}$ .
2. Compute  $\lambda_{k+1}$ , an  $\varepsilon$ -center of  $P_{k+1}$ .
3. Set  $U_{k+1} := c'\lambda_k$ .

Else

1. The oracle returns a number of separating hyperplanes  $a'_i \lambda = b_i$ ,  $i = 1, \dots, m$ , such that the feasible solutions satisfy  $a'_i \lambda > b_i$ ,  $i = 1, \dots, m$ , and  $\lambda_k$  satisfies  $a'_i \lambda_k \leq b_i$ ,  $i = 1, \dots, m$ .
2. Let  $P_{k+1} := P_k \cap \{\lambda \mid a'_i \lambda \geq b_i, i = 1, \dots, m\}$ .
3. Compute  $\lambda_{k+1}$ , an  $\varepsilon$ -center of  $P_{k+1}$ .
4. Set  $U_{k+1} := U_k$ .

Endif

Set  $k := k + 1$ .

**Until** (the stopping criterion is satisfied).

### Generate an $\varepsilon$ -Center

In each iteration of the analytical center cutting plane method, an  $\varepsilon$ -center of polyhedron  $P_k$  has to be computed. Algorithms for finding such an approximate center have been well-studied for decades. Various algorithms based on the Newton's method have been developed

and can be found in the mathematical programming literature. These algorithms are proven to be efficient (converge in polynomial-time). The book by Ye [90] is a good reference of this subject.

Furthermore, the Newton's method for finding an  $\varepsilon$ -center of  $P_k$  would converge very rapidly if it starts with an appropriately selected point in  $P_k$ . Therefore, some research efforts have focused on the issue of how to select a good starting point. In the ACCPM where one central cutting plane is introduced in each iteration, the paper by Goffin, Luo and Ye [25] proposes a way of selecting starting points such that the Newton's method restores an  $\varepsilon$ -center of  $P_k$  in  $\mathbf{O}(1)$  iterations. The case of multiple cuts was studied in [31]. It is shown that the restoration of an  $\varepsilon$ -center can be done in  $\mathbf{O}(p \log(p + 1))$  Newton's iterations, where  $p$  is the number of new cuts introduced by the oracle.

### The Stopping Criterion

Again, we assume that set  $\Lambda$  is bounded, of full dimension, and contained in the initial outer approximation  $P_0$ . We also assume that  $\Lambda$  contains a ball of radius  $\kappa$  if it is non-empty.

Suppose that  $\Lambda$  is not empty. Since  $P_0$  contains  $\Lambda$ , therefore, at least an optimal solution  $\lambda^*$  is contained in  $P_k$  for all  $k$ . Therefore, we have

$$c'\lambda^* \geq \min_{\lambda \in P_k} c'\lambda := \min_{A'\lambda \geq b} c'\lambda, \quad (6.19)$$

where  $A = [a_1, \dots, a_{N_k}]$  and  $b' = [b_1, \dots, b_{N_k}]$ . Thus a lower bound on the optimal objective of (6.14) can be obtained by solving the minimization problem on the right-hand-side of (6.19), which is a linear program.

Another approach to obtain a lower bound of  $c'\lambda^*$  is to solve the dual problem of the linear program in (6.19)

$$\max b'y, \quad \text{subj. to } Ay = c, \quad y \geq 0. \quad (6.20)$$

This approach is suggested in [62]. It is well known that the optimal objective of (6.20) is equal to the optimal objective of the linear program in (6.19). Therefore, any suboptimal objective of (6.20) is a lower bound of  $c'\lambda^*$ . Hence, the advantage of this approach is that a lower bound of  $c'\lambda^*$  is obtained whenever a feasible solution of (6.20) is found and we do

not have to solve (6.20) exactly.

Let  $L_k$  be a lower bound of  $c'\lambda^*$  obtained by solving either the linear program in (6.19) or the linear program (6.20). Then, the simple stopping criterion

$$\text{Until } (U_k - L_k < \kappa) \tag{6.21}$$

guarantees that on exit, we find a suboptimal solution  $\bar{\lambda}$  such that the difference between  $c'\bar{\lambda}$  and the optimal objective is less than or equal to  $\kappa$ .

Suppose that there is no feasible solution found up to the  $k^{\text{th}}$  iteration. Then the following stopping criterion is used to determine whether the algorithm is to be stopped with a declaration that there is no feasible solution:

$$\text{Until } (F_k(\lambda_k) > -N_k \ln(\kappa) + \frac{\varepsilon^2}{1 - \varepsilon^2}), \tag{6.22}$$

where  $\lambda_k$  is the  $\varepsilon$ -center of  $P_k$ ,  $N_k$  is the number of linear inequalities which define  $P_k$ , and  $\kappa$  is the radius of the ball contained in  $\Lambda$ . The idea behind this criterion is as follows [25]: assume that  $\Lambda$  is non-empty. Then  $\Lambda$  contains a ball of radius  $\kappa$ . Let  $\bar{\lambda}$  denote the center of the ball and  $\lambda_a$  denote the analytical center of  $P_k$ . Since  $P_k$  contains  $\Lambda$ , therefore  $P_k$  contains the ball. Note that  $\|a_i\| = 1$  for all  $i$ . Hence,  $a_i'\bar{\lambda} - b_i \geq \kappa$  for  $i = 1, \dots, N_k$ , and the following inequalities hold

$$-F_k(\lambda_a) := \sum_{i=1}^{N_k} \ln(a_i'\lambda_a - b_i) \geq \sum_{i=1}^{N_k} \ln(a_i'\bar{\lambda} - b_i) \geq N_k \ln(\kappa).$$

Finally, it is well-known that if  $\lambda_k$  is an  $\varepsilon$ -center, then

$$-F_k(\lambda_k) + \frac{\varepsilon^2}{1 - \varepsilon^2} \geq -F_k(\lambda_a).$$

Hence if  $\Lambda$  is non-empty, the following inequality has to hold for all  $k$

$$F_k(\lambda_k) \leq -N_k \ln(\kappa) + \frac{\varepsilon^2}{1 - \varepsilon^2}.$$

To summarize, the analytical center cutting plane algorithm uses (6.22) as a stopping criterion until a feasible point is found. When a feasible point is found, the algorithm starts

to use (6.21) as the stopping criterion. On exit, criterion (6.21) guarantees a  $\kappa$ -accurate suboptimal solution if the problem is feasible.

### Convergence and Complexity

If the total number of cuts generated by the oracle before the algorithm stops is assumed to be known a priori, then it is easy to show that the number of iterations of ACCPM is polynomial in this number. This assumption is of course not tenable in general, and the challenge is then to prove a complexity bound that depends on the number of decision variables but not on the number of cutting hyperplanes generated. Goffin, Luo, and Ye [25] showed that, the basic setup of the algorithm - the algorithm adds one single central cut in each iteration - has a complexity estimation of  $\mathbf{O}^*(\frac{n^2}{\epsilon^2})$  for the total number of iterations, where  $n$  is the number of decision variables,  $\epsilon$  is the index of accuracy used in the stopping criterion, and the  $\mathbf{O}^*$  notation means that the lower order terms are ignored. The extensions of this result to the case of deep cuts were given in [26, 29].

In practice, it often occurs that the oracle in the cutting plane algorithm can generate more than one cutting hyperplane upon a single inquiry. In [30], the case of two central cuts was analyzed, and it is shown that the ACCPM in which two central cuts are placed in each iteration also converges in  $\mathbf{O}^*(\frac{n^2}{\epsilon^2})$  iterations. The case of multiple central cuts was analyzed in [89]. It is shown that the algorithm converges after  $\mathbf{O}^*(\frac{\bar{p}^2 n^2}{\epsilon^2})$  cutting planes have been generated, where  $\bar{p}$  is the maximum number of cuts generated at any given iteration.

*Remark 6.1.* Our numerical experiments suggest that the  $\mathbf{O}^*(\frac{n^2}{\epsilon^2})$  bound on the number of iterations is a very conservative estimation. From our experience, the ACCPM usually terminates long before the bound is reached.

### 6.6.2 The ACCPM for Solving IQC Optimization Problems

We have implemented ACCPM in MATLAB to solve IQC optimization problems (6.5). Our implementation follows exactly the algorithm described in the previous subsection, except for the stopping criterion.

### An Alternative Stopping Criterion

As mentioned in the previous subsection, at the  $k^{\text{th}}$  iteration of the ACCPM, a lower bound of the optimal objective of the problem to be solved can be obtained by minimizing the

linear objective over  $P_k$ ; i.e., by solving a linear program. However, this is often too much computational effort for the purpose. A rough lower bound can be obtained by a much less expensive way. The idea is to find an ellipsoid  $\tilde{E}_k$  which contains  $P_k$ . Then a lower bound of the optimal objective can be obtained by minimizing the linear objective over  $\tilde{E}_k$ , for which an analytical solution is available.

Let  $A > 0$  be an  $n \times n$  symmetric matrix and  $\lambda_0$  be a column vector in  $\mathbf{R}^n$ . Let  $E(A, \lambda_0, \gamma)$  denote the ellipsoid  $\{\lambda \mid (\lambda - \lambda_0)'A(\lambda - \lambda_0) \leq \gamma^2\}$ . Let  $\lambda_k^*$  be the analytical center of  $P_k$  and  $\lambda_k$  be an  $\varepsilon$ -approximation of  $\lambda_k^*$ . The following properties are well known. The proofs can be found in [60, 2].

**Proposition 6.2.** *If  $\rho_k(\lambda_k) < 1$ , then analytical center  $\lambda_k^*$  belongs to  $E(\nabla^2 F_k(\lambda_k), \lambda_k, \kappa)$ , where  $\rho_k(\cdot)$  is defined in (6.18) and  $\kappa = \frac{\rho_k(\lambda_k)}{1 - \rho_k(\lambda_k)}$ .*

**Proposition 6.3.** *Let  $z$  belong to the interior of  $P_k$ , and  $0 < \alpha < 1$ . Then  $E(\nabla^2 F_k(z), z, \alpha)$  is contained in  $P_k$ . Furthermore, for any  $y \in E(\nabla^2 F_k(z), z, \alpha)$ , the following inequality holds*

$$\frac{1}{(1 + \alpha)^2} \nabla^2 F_k(z) \leq \nabla^2 F_k(y) \leq \frac{1}{(1 - \alpha)^2} \nabla^2 F_k(z).$$

**Proposition 6.4.** *The polyhedron  $P_k$  is contained in the ellipsoid  $E(\nabla^2 F(\lambda_k^*), \lambda_k^*, N_k)$ , where  $N_k$  is the number of linear inequalities that define  $P_k$ .*

The following lemma shows that the polyhedron  $P_k$  is contained in a particular ellipsoid centered at an  $\varepsilon$ -center of  $P_k$ .

**Lemma 6.2.** *Let  $\lambda_k$  be an  $\varepsilon$ -center of  $P_k$  and  $\varepsilon < \frac{1}{2}$ . Then  $P_k$  is contained in the ellipsoid  $E(\nabla^2 F(\lambda_k), \lambda_k, \kappa)$ , where*

$$\kappa = \frac{N_k}{1 - \rho_k(\lambda_k)} + \frac{\rho_k(\lambda_k)}{1 - \rho_k(\lambda_k)},$$

and  $N_k$  is the the number of linear inequalities that define  $P_k$ .

*Proof.* Since  $\lambda_k$  is an  $\varepsilon$ -approximation of the analytical center and  $\varepsilon < \frac{1}{2}$ ,  $\rho_k(\lambda_k)$  must satisfy

$$\frac{\rho_k(\lambda_k)}{1 - \rho_k(\lambda_k)} < 1,$$



Therefore, by Proposition 6.2 and Proposition 6.3, we have

$$(\lambda_k^* - \lambda_k)' \nabla^2 F_k(\lambda_k) (\lambda_k^* - \lambda_k) \leq \left( \frac{\rho_k(\lambda_k)}{1 - \rho_k(\lambda_k)} \right)^2 < 1,$$

and

$$\nabla^2 F_k(\lambda_k) \leq \left( 1 + \frac{\rho_k(\lambda_k)}{1 - \rho_k(\lambda_k)} \right)^2 \nabla^2 F_k(\lambda_k^*) = \frac{1}{(1 - \rho_k(\lambda_k))^2} \nabla^2 F_k(\lambda_k^*). \quad (6.23)$$

Now, let  $z \in P_k$ . We have

$$\begin{aligned} (z - \lambda_k)' \nabla^2 F_k(\lambda_k) (z - \lambda_k) &= (z - \lambda_k^* + \lambda_k^* - \lambda_k)' \nabla^2 F_k(\lambda_k) (z - \lambda_k^* + \lambda_k^* - \lambda_k) \\ &\leq \left( \sqrt{(z - \lambda_k^*)' \nabla^2 F_k(\lambda_k) (z - \lambda_k^*)} + \sqrt{(\lambda_k^* - \lambda_k)' \nabla^2 F_k(\lambda_k) (\lambda_k^* - \lambda_k)} \right)^2 \\ &\leq \left( \frac{1}{1 - \rho_k(\lambda_k)} \sqrt{(z - \lambda_k^*)' \nabla^2 F_k(\lambda_k^*) (z - \lambda_k^*)} + \sqrt{(\lambda_k^* - \lambda_k)' \nabla^2 F_k(\lambda_k) (\lambda_k^* - \lambda_k)} \right)^2. \end{aligned}$$

To see the first inequality, we note that  $\nabla^2 F_k(\lambda_k) > 0$ , and therefore  $g(h) := (h' \nabla^2 F_k(\lambda_k) h)^{\frac{1}{2}}$  is a generalized norm of  $\mathbf{R}^n$ . Thus,  $g(z - \lambda_k^* + \lambda_k^* - \lambda_k) \leq g(z - \lambda_k^*) + g(\lambda_k^* - \lambda_k)$ . The second inequality follows inequality (6.23). Now, by Proposition 6.4 and Proposition 6.2, we obtain

$$(z - \lambda_k)' \nabla^2 F_k(\lambda_k) (z - \lambda_k) \leq \left( \frac{m}{1 - \rho_k(\lambda_k)} + \frac{\rho_k(\lambda_k)}{1 - \rho_k(\lambda_k)} \right)^2. \quad (6.24)$$

This concludes the proof.  $\square$

Therefore, by computing an  $\varepsilon$ -approximation of the analytical center of  $P_k$ , we also get an ellipsoid which contains the polyhedron  $P_k$  as a byproduct. A lower bound of the optimal objective can now be easily computed as

$$c' \lambda_k - \kappa \sqrt{c' (\nabla^2 F(\lambda_k))^{-1} c}.$$

Furthermore, since the ellipsoid  $E(\nabla^2 F(\lambda_k), \lambda_k, \kappa)$  contains  $P_k$ , its volume can be used as an estimation of the size of  $P_k$ .

Under the assumption that  $\Omega$  contains a full dimension ball of radius  $\varepsilon$  if it is non-empty, we adopt the following stopping criterion: at the  $k^{\text{th}}$  iteration, if no feasible solution has

been found so far and  $\kappa^n / \sqrt{\det(\nabla^2 F(\lambda_k))} < \epsilon$ , then the algorithm stops and declares that the problem has no feasible solution. If any feasible solution is found, the algorithm stops whenever

$$U_k - (c' \lambda_k - \kappa \sqrt{c'(\nabla^2 F(\lambda_k))^{-1}c}) < \epsilon.$$

On exit, the above criterion guarantees the suboptimal found is  $\epsilon$ -accurate.

## Some Other Implementation Details

Some other implementation details about our implementation are summarized as follows:

1. Initialization: in our algorithm, we assume that the feasible set of (6.5) is contained in a box  $P_0$

$$P_0 := \{\lambda \in \mathbf{R}^n : -r \leq \lambda_i \leq r, i = 1, \dots, n\},$$

where  $r$  is a positive constant, say  $10^9$ . Notice that by this choice, 0 is the exact analytical center of  $P_0$ .

2. The oracle for the IQC optimization problems is discussed in Section 6.3. The main computational procedure involved is basically eigenvalue decomposition of certain matrices.
3. We have implemented two versions of ACCPM to solve IQC optimization problems. One adds a single central cut in each iteration; the other allows multiple central cuts. In the single central cut version, we follow [25] to implement a routine for recovering an  $\epsilon$ -center after a cut is placed. The restoration takes  $\mathbf{O}(1)$  Newton's iterations. In the multiple central cuts version, we follow [31] which suggests an approach that takes  $\mathbf{O}(p \log(p + 1))$  Newton's iterations to restore an  $\epsilon$ -center. Here  $p$  is the number of new cuts introduced by the oracle, which may vary with the iteration.

We have tested our implementation on a number of numerical examples. We present the results of these numerical experiments in the next section.

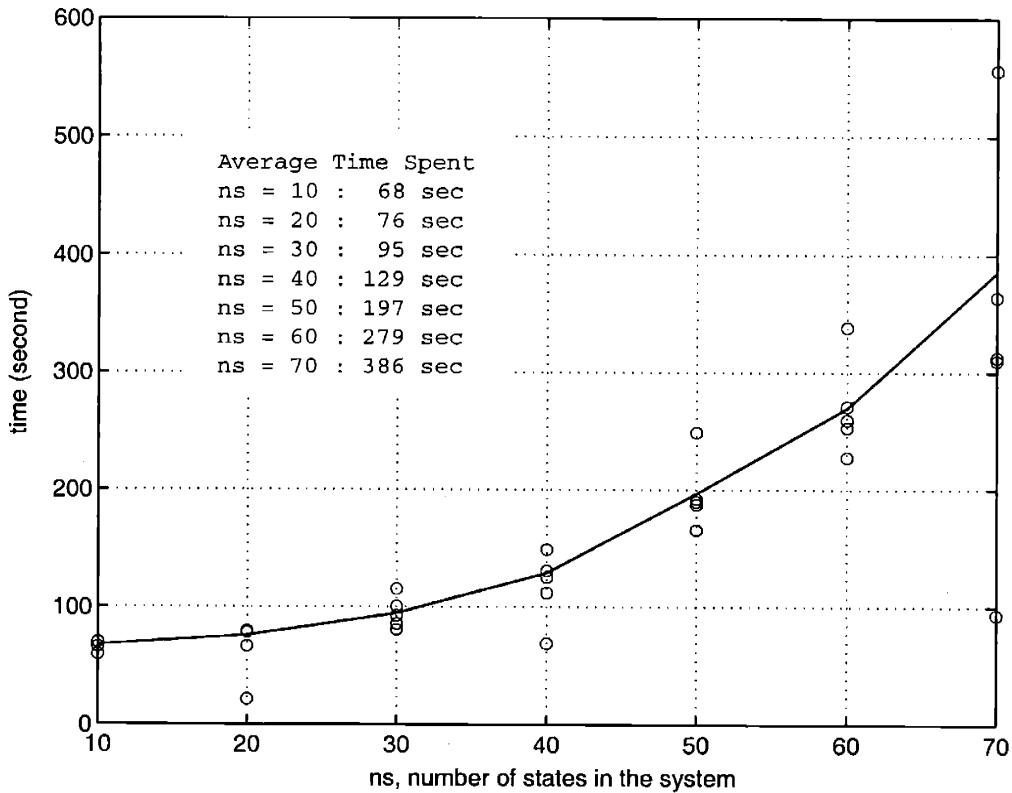


Figure 6-6: The figure shows the amount of time that the modified ellipsoid algorithm described in Section 6.5 took to solve problem (6.9). For each  $n_s$ , five test problems were randomly generated. The algorithm starts with an initial ball of radius equal to  $10^6$ . As we can see from the figure, all problems were solved in less than 10 minutes. Compared with Figure 6-2, we see that the ellipsoid algorithm is much faster than the conventional method when the problem to be solved has a large state space.

## 6.7 Examples

**Example 6.1.** Consider again the  $L_2$ -gain estimation problem in Section 6.2. The modified ellipsoid algorithm and the analytical center cutting plane method described in Sections 6.5 and 6.6 are used to solve the same set of problems. Figure 6-6 and Figure 6-7 show the amount of time which the ellipsoid algorithm and the ACCPM spent on solving these problems, respectively.

In the case of the ellipsoid algorithm, Figure 6-2 and Figure 6-6 indicate that the conventional method solved a problem faster than the ellipsoid algorithm when the number of states in the problem is small ( $n_s \leq 20$ ). The ellipsoid algorithm becomes faster than

the conventional method when the problem to be solved has more than 30 states. When  $30 \leq n_s \leq 50$ , the ellipsoid method is about 2 to 5 times faster than the conventional method. When  $n_s \geq 60$ , the difference in speed becomes more significant. The ellipsoid algorithm in these cases is more than 10 times faster than the conventional method. In all cases, the ellipsoid algorithm solved a problem in 10 minutes.

In the case of the analytical center cutting plane method, Figure 6-2 and Figure 6-7 indicate that the ACCPM is no worse than the conventional method even when the problem to be solved has a small state space. When the number of states of a problem is 10, the ACCPM and the conventional method took about the same time to solve the problem. The ACCPM took approximately half amount of the time which the conventional method spent on solving a problem with 20 states. The difference in speed becomes more and more significant as the number of states grows. When  $n_s = 70$ , the ACCPM solved a problem approximately 60 to 100 times faster than the conventional method. In all cases, the ACCPM spent less than 2 minutes to solve a problem.

Note that the initial ball constraint of the ellipsoid algorithm and the initial box constraint of the analytical center cutting plane algorithm certainly affect the amount of time which the two algorithms spend on solving a problem. For a particular optimization problem, the larger the sizes of the initial ball and box in which we assume the feasible set is contained, the longer it takes for the ellipsoid algorithm and the ACCPM to obtain sub-optimal solutions with the same accuracy. For the results presented so far, the ellipsoid algorithm starts with a ball of radius equal to  $10^6$ , and the analytical center cutting plane algorithm starts with the box  $[-10^6, 10^6]^n$ . We have tested the algorithms with different initial constraints. With the radius of the initial ball increasing to  $10^{10}$  and the initial boxes increasing to  $[-10^{10}, 10^{10}]^n$ , ellipsoid algorithm and the ACCPM still significantly outperform the conventional method when  $n_s \geq 50$ .

In contrast to the conventional method, the amount of time which the ellipsoid algorithm and the analytical center cutting plane method spent on solving the problems does not increase significantly as the number of states of the problems increases. Therefore, one can expect that the ellipsoid algorithm and the ACCPM are able to solve problems with very large state space in a reasonable time. We have tested the algorithms on problems which have up to 200 states. The results are listed in Table 6.1. Note when problems have more than 130 states, the sizes of the equivalent SDPs are so large that the MATLAB LMI solver

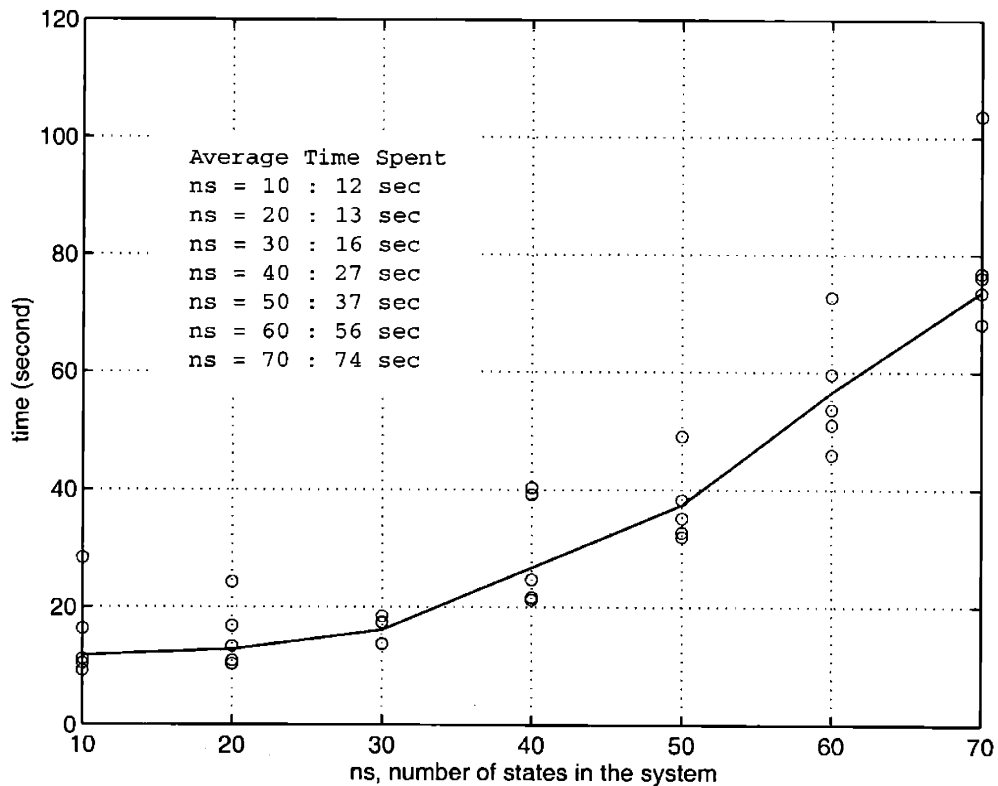


Figure 6-7: The figure shows the amount of time that the analytical center cutting plane method described in Section 6.6 took to solve problem (6.9). For each  $n_s$ , five test problems were randomly generated. The algorithm started with an initial box constraint  $\lambda_i \in [-10^6, 10^6]$ . The figure indicates that all problems were solved in less than 2 minutes. Compared with the Figure 6-2, we see that the ACCPM is significantly faster than the conventional method when the problem to be solved has a large state space. Furthermore, the ACCPM is no worse than the conventional method even when the problem has only a few states.

	$n_s = 100$			$n_s = 130$		
Method	objective	time (sec)	iterations	objective	time (sec)	iterations
LMI Tool	0.80193	> 9 hours	53	*****	****	****
ACCPM	0.80192	277	380	0.21066	414	412
Ellipsoid	0.80191	1347	1087	0.21066	2180	1001
	$n_s = 160$			$n_s = 200$		
Method	objective	time (sec)	iterations	objective	time (sec)	iterations
LMI Tool	*****	****	****	*****	****	****
ACCPM	0.08846	1473	371	0.92488	3244	402
Ellipsoid	0.08846	5773	1031	0.92487	13767	1327

Table 6.1: Results of solving (6.9) using MATLAB LMI Control Toolbox, the ellipsoid algorithm, and the ACCPM. “LMI Tool“ denotes MATLAB LMI Control Toolbox, “ACCPM“ denotes the Analytical Center Cutting Plane Method, and “Ellipsoid“ denotes the ellipsoid algorithm. In this set of testing problems, the ACCPM starts with a initial box constraint  $\{x : |x_i| \leq 10^9\}$ , and the ellipsoid algorithm starts with a ball constraints  $\{x : \|x\| \leq 10^9\}$ . The objective value is accurate up to the 4<sup>th</sup> digit. When  $n_s = 200$  (the number of states in the problem is 200), the ACCPM and the ellipsoid algorithm can still solve the problem in a reasonable period of time.

	$n_s = 10$	$n_s = 20$	$n_s = 30$	$n_s = 40$	$n_s = 50$	$n_s = 60$	$n_s = 70$
Method	iteration	iteration	iteration	iteration	iteration	iteration	iteration
ACCPM	397	388	340	374	327	360	334
Ellipsoid	1589	1582	1406	1339	1414	1388	1296

Table 6.2: Numbers of iterations that the ellipsoid algorithm (denoted by “Ellipsoid“) and the analytical center cutting plane method (denoted by “ACCPM“) took to solve (6.9).

ran out of the memory (256MB) of the machine we tested these problems on and failed to produce results.

Finally, we note that the analytical center cutting plane algorithm is approximately 5 times faster than the ellipsoid algorithm according to the numerical results presented in this example. This is somewhat contrary to the theoretical prediction. Recall the theoretical worst-case complexity analysis of the two algorithms. Counted in the number of iterations required to find an  $\epsilon$ -accurate suboptimal solution, the ACCPM has a worst-case  $\mathbf{O}^*(\frac{n^2}{\epsilon^2})$  estimate. The estimate for the ellipsoid algorithm is  $\mathbf{O}(n^2 |\log \epsilon|)$ . This suggests that the ACCPM should have taken more iterations to converge compared to the ellipsoid algorithm. However, in all test incidents, the ACCPM took fewer iterations to solve a problem. Table 6.2 shows the average number of iterations which the two algorithms took to solve these problems.

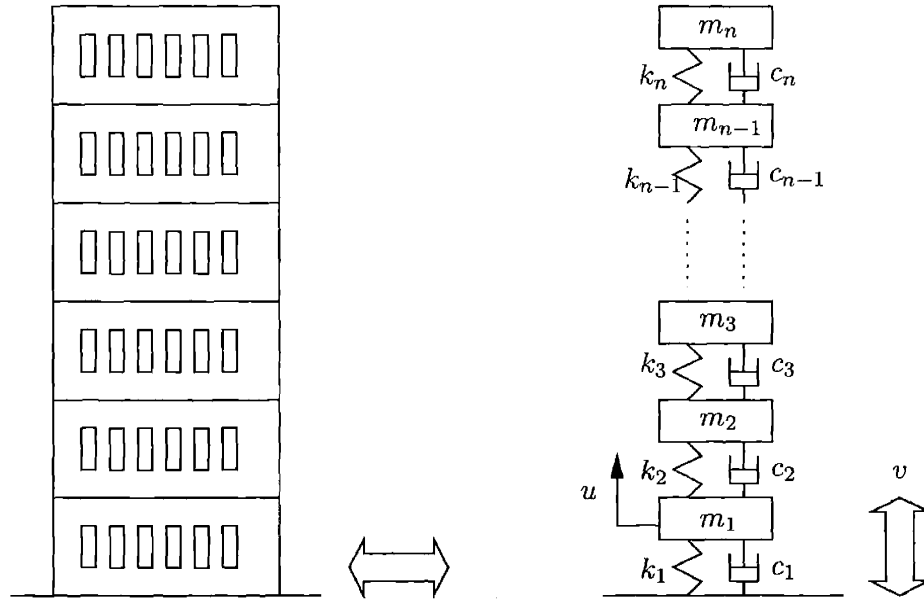


Figure 6-8: Seismic isolation control of a building. The building is modelled as a series connection of masses, springs, and dampers.

**Example 6.2 (Seismic Isolation Control).** This example is adapted from [53]. Consider the problem of seismic isolation control of an  $n$  story building. The building is modelled as a series connection of masses, springs and dampers. See Figure 6-8 for an illustration. This way, the dynamics of the building is modelled by multiple spring-damper-mass equations

$$\begin{aligned}
 m_1 \ddot{x}_1 + c_1 \dot{x}_1 + k_1 x_1 - c_2(\dot{x}_2 - \dot{x}_1) - k_2(x_2 - x_1) &= -u + v, \\
 m_r \ddot{x}_r + c_r(\dot{x}_r - \dot{x}_{r-1}) + k_r(x_r - x_{r-1}) - c_{r+1}(\dot{x}_{r+1} - \dot{x}_r) - k_{r+1}(x_{r+1} - x_r) &= 0 \\
 (\text{for } r = 2, \dots, n-1), \\
 m_n \ddot{x}_n + c_n(\dot{x}_n - \dot{x}_{n-1}) + k_n(x_n - x_{n-1}) &= 0,
 \end{aligned}$$

where  $u$  is the control force applied between the ground and the first floor of the building, and  $v$  represents the result of earthquake force applied to the ground. The frequency spectrum of  $v$  is assumed to be between 1/3 and 3 Hz.

Seismic isolation controllers are designed for buildings of 6, 7, 8, 9, 10, and 11 stories, under the assumption that acceleration sensors are available at each floor of the buildings.





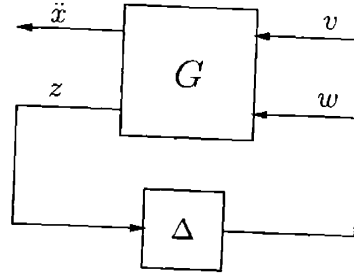


Figure 6-10: Setup for robustness analysis of the seismic control systems.

damping coefficients for floors 1 to 5, denoted by  $\bar{k}_r$  and  $\bar{c}_r$ , are represented as

$$\bar{k}_r = (1 + \delta_{kr})k_r, \quad \bar{c}_r = (1 + \delta_{cr})c_r, \quad r = 1, \dots, 5,$$

where  $\delta_{kr}$  and  $\delta_{cr}$  are unknown constants whose absolute values are less than or equal to 0.1. We are interested in checking whether the seismic control systems are still stable under the change of these parameters and estimating the  $L_2$ -gain from the earthquake input  $v$  to the acceleration vector  $\ddot{x}$ . To do this, we represent the closed-loop seismic control systems in the standard form for robustness analysis as shown in Figure 6-10, where  $G$  denotes the nominal closed-loop seismic control system, and output  $z$  is a  $10 \times 1$  vector signal defined as

$$z' = \left[ \frac{k_1 x_1}{10} \quad \frac{k_2(x_2 - x_1)}{10} \quad \dots \quad \frac{k_5(x_5 - x_4)}{10} \quad \frac{c_1 \dot{x}_1}{10} \quad \frac{c_2(\dot{x}_2 - \dot{x}_1)}{10} \quad \dots \quad \frac{c_5(\dot{x}_5 - \dot{x}_4)}{10} \right].$$

Matrix  $\Delta := \text{diag}(\delta_1, \dots, \delta_{10})$  represents the uncertainties in the system. Each  $\delta_i$  denotes an unknown constant with absolute value less than or equal to 1.

To estimate the  $L_2$ -gain from  $v$  to  $\ddot{x}$ , we apply standard IQC analysis to the system in Figure 6-10. The IQC defined by the following quadratic form

$$\int_{-\infty}^{\infty} \begin{bmatrix} \hat{z}_i(j\omega) \\ \hat{w}_i(j\omega) \end{bmatrix}^* \Pi_i(j\omega) \begin{bmatrix} \hat{z}_i(j\omega) \\ \hat{w}_i(j\omega) \end{bmatrix} d\omega \quad (6.27)$$

is used to characterize the relationship  $w_i = \delta_i z_i$ , where the multiplier  $\Pi_i(j\omega)$  is of the form

$$\Pi_i(j\omega) = \begin{bmatrix} \text{Re}(\lambda_{1i} + \lambda_{2i} \frac{1}{j\omega+1}) & \lambda_{3i}(\frac{1}{j\omega+1} - \frac{1}{-j\omega+1}) \\ \lambda_{3i}(\frac{1}{-j\omega+1} - \frac{1}{j\omega+1}) & -\text{Re}(\lambda_{1i} + \lambda_{2i} \frac{1}{j\omega+1}) \end{bmatrix}, \quad (6.28)$$

	$n = 6$				$n = 7$			
Method	$L_2$ -gain	time (sec)	iterations	var.	$L_2$ -gain	time (sec)	iterations	var.
LMI Tool	0.91807	4947	59	1169	1.48210	8700	76	1367
ACCPM	0.91805	339	1803	31	1.48210	271	1382	31
Ellipsoid	0.91805	1461	2066	31	1.48211	1988	3701	31
	$n = 8$				$n = 9$			
Method	$L_2$ -gain	time (sec)	iterations	var.	$L_2$ -gain	time (sec)	iterations	var.
LMI Tool	1.02685	11947	78	1581	1.10925	15221	82	1811
ACCPM	1.02685	290	1393	31	1.10917	208	975	31
Ellipsoid	1.02686	2163	3875	31	1.10917	1421	3178	31
	$n = 10$				$n = 11$			
Method	$L_2$ -gain	time (sec)	iterations	var.	$L_2$ -gain	time (sec)	iterations	var.
LMI Tool	0.83181	19033	77	2057	1.14860	22325	74	2319
ACCPM	0.83179	383	1560	31	1.14815	317	1190	31
Ellipsoid	0.83181	1590	1871	31	1.14816	2054	3195	31

Table 6.3: Results of solving the  $L_2$ -gain estimation problem in Example 6.2 using MATLAB LMI Control Toolbox, the ellipsoid algorithm, and the ACCPM. “LMI Tool“ denotes MATLAB LMI Control Toolbox. “ACCPM“ denotes the Analytical Center Cutting Plane Method. “Ellipsoid“ denotes the ellipsoid algorithm. The numbers in the column “var“ indicate the number of decision variables in a problem. In this set of testing problems, the ACCPM starts with an initial box constraint  $\{x : |x_i| \leq 10^6\}$ , and the ellipsoid algorithm starts with a ball constraints  $\{x : \|x\| \leq 10^6\}$ .

and parameters  $\lambda_{1i}$ ,  $\lambda_{2i}$  must satisfy  $\text{Re}(\lambda_{1i} + \lambda_{2i} \frac{1}{j\omega+1}) > 0 \forall \omega$ . Then the  $L_2$ -gain estimation problem can be formulated either as an optimization problem over frequency dependent matrix inequalities, or equivalently a SDP. Here we omit the details of the problem formulations.

We solve the optimization problems resulting from IQC analysis of seismic isolation control systems using the MATLAB LMI Control Toolbox, the ellipsoid algorithm described in Section 6.5, and the analytical center cutting plane method described in Section 6.6. The results are listed in Table 6.3. Table 6.3 indicates that both the ellipsoid algorithm and the ACCPM solved every problem faster than the conventional method using MATLAB LMI Control Toolbox. When  $n = 6$  (that is, the building has 6 floors), the equivalent SDP has 1169 decision variables and it took the LMI Control Toolbox about 80 minutes to solve the problem. For the same case, the ellipsoid algorithm took about 25 minutes and the ACCPM took only 5 minutes. As  $n$  increases from 6 to 11, the amount of time which the LMI Control Toolbox took to solve a problem increases rapidly. This is expected since the number of decision variables in the equivalent SDP increases from 1169 to 2319. On the other hand, the amount of time which the ellipsoid algorithm and the analytical center

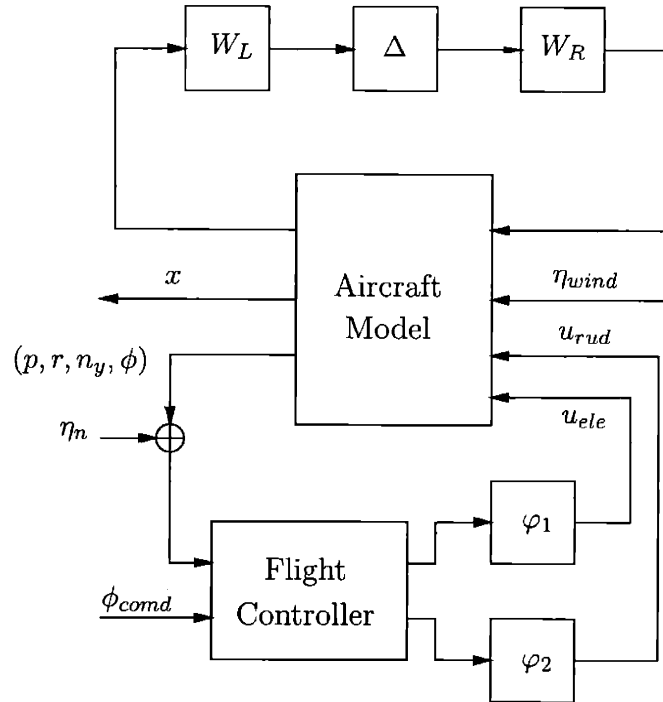


Figure 6-11: Schematic diagram of the lateral flight control system of the Space Shuttle.

cutting plane method took stays at the same level. For  $n = 11$ , the LMI Control Toolbox took more than 6 hours to solve the problem, while the ellipsoid algorithm took about half an hour and the ACCPM took only 5 minutes.

**Example 6.3 (Space Shuttle Robustness Analysis).** This example is about robustness analysis of the Space Shuttle lateral axis flight control system during the reentry. The system is a simplified model of the space shuttle, in the final stage of landing, as it transitions from supersonic to subsonic speeds. This example is adapted from [4], and the material is originally based on the paper [15].

Figure 6-11 shows the Space Shuttle lateral axis flight control system. The rigid body model for the aircraft at Mach 0.9 is a four-state system with states

$$x = \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} \text{sideslip angle} \\ \text{roll rate} \\ \text{yaw rate} \\ \text{bank angle} \end{bmatrix}.$$

There are three inputs to the aircraft. The first input is the actual angular deflection of

the elevon surface. The second input is the actual deflection of the rudder surface. Finally, the last input is a lateral wind gust disturbance input  $\eta_{wind}$ , due to the winds that occur at this altitude.

The major source of uncertainty in the aircraft model is in the aerodynamic coefficients. There are standard aerodynamic parameters which express incremental forces and torques generated by incremental changes in sideslip, elevon, and rudder angles. This is a linear relationship, expressed as

$$\begin{bmatrix} \text{side force} \\ \text{yawing moment} \\ \text{rolling moment} \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 \\ c_4 & c_5 & c_6 \\ c_7 & c_8 & c_9 \end{bmatrix} \begin{bmatrix} \beta \\ \theta_{ele} \\ \theta_{rud} \end{bmatrix}.$$

Uncertainty in these coefficients is modelled as a nominal value, plus a perturbation term

$$\begin{bmatrix} c_1 & c_2 & c_3 \\ c_4 & c_5 & c_6 \\ c_7 & c_8 & c_9 \end{bmatrix} = \bar{C} + W_r \Delta W_l,$$

where  $W_R$  and  $W_L$  are weighting matrices of dimension  $3 \times 9$  and  $9 \times 3$ , respectively.  $\Delta$  is a diagonal uncertainty; i.e.,  $\Delta = \text{diag}(\delta_1, \dots, \delta_9)$ , and the each  $\delta_i$  is assumed to be a fixed, unknown real parameter which satisfies  $|\delta_i| \leq 1$ .

The controller has four noised sensor inputs and one command input. Three of the sensor inputs are states:  $(p, r, \phi)$ . The fourth sensor input is the lateral acceleration at the pilot's location, denoted by  $n_y$ . The command input is the band-angle command. There are two outputs of the controller. One is the rudder command, and the other is the elevon command. In this example, the flight controller is identical to the one in Chapter 7 of [4], where controller design is based on weighted  $\mathcal{H}_\infty$  optimization. The controller has 28 states.

The control signals are modified by two identical nonlinear operators before they enter the aircraft model. The nonlinear operators are defined as

$$\varphi_i(x) = x - \text{dzn}_i(x), \quad \text{where } \text{dzn}_i(x) = \begin{cases} 0, & \text{if } |x| \leq 1 \\ x, & \text{otherwise} \end{cases}.$$

The purpose of these nonlinearities is to model the effect of actuator saturation. Note that

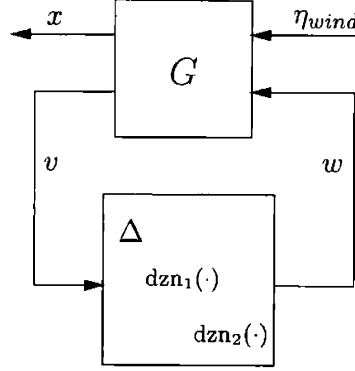


Figure 6-12: Setup for robustness analysis of the Space Shuttle lateral axis flight control system.

in [4] where this example is taken, the effect of actuator saturation is not considered. Here we add these nonlinearities to make the setup more realistic.

In this example, we are interested in verifying stability of the flight control system subject to the uncertainty in aerodynamic coefficients and actuator saturation. Furthermore, we also want to compute an upper bound of the  $\mathbf{L}_2$ -gain from the disturbance  $\eta_{wind}$  to the state vector  $x$ . To do so, we transform the flight control system into the standard robustness analysis setup, as shown in Figure 6-12 and apply standard IQC analysis. In this example, the nominal linear time-invariant system  $G$  has 58 states. The feedback part consists of 9 unknown constants and two dead zone nonlinearities. The uncertain constants  $\delta_i$ ,  $i = 1, \dots, 9$  are characterized using IQCs defined by the quadratic forms of the form in (6.27). The dead zone nonlinearities  $dzn_i(\cdot)$  ( $i = 1, 2$ ) are characterized using the so-called Zame-Falb's IQCs [55]. The IQCs are defined by the following quadratic forms

$$\int_{-\infty}^{\infty} \begin{bmatrix} \hat{v}_i(j\omega) \\ \hat{w}_i(j\omega) \end{bmatrix}^* \begin{bmatrix} 0 & \Pi_i(j\omega) \\ \Pi_i(j\omega)^* & -2\text{Re}(\Pi_i(j\omega)) \end{bmatrix} \begin{bmatrix} \hat{v}_i(j\omega) \\ \hat{w}_i(j\omega) \end{bmatrix} d\omega, \quad i = 1, 2, \quad (6.29)$$

where  $\hat{v}_i(j\omega)$ ,  $\hat{w}_i(j\omega)$  are the Fourier transforms of the input and output signals of  $dzn_i(\cdot)$ .  $\Pi_i(j\omega)$  are in the form of

$$\Pi_i(j\omega) = \lambda_{1i} \frac{j\omega}{j\omega + 1} + \lambda_{2i} \frac{-j\omega}{-j\omega + 1} + \lambda_{3i} \frac{j\omega + 2}{j\omega + 1} + \lambda_{4i} \frac{-j\omega + 2}{-j\omega + 1},$$

where  $\lambda_{ki} > 0$ ,  $k = 1, \dots, 4$ . The  $\mathbf{L}_2$ -gain estimation problem then can be formulated as an optimization problem of the form in (6.5). We solve the problem using the ellipsoid al-

Method	#. of dec. var.	Estimated $L_2$ -gain	Time (sec)	Iterations
LMI Tool	1756	66.338	29437	116
ACCPM	36	66.337	334	813
Ellipsoid	36	66.323	1022	2402

Table 6.4: Results of solving the IQC optimization problem arising from the robustness analysis of Space Shuttle lateral flight control system using MATLAB LMI Control Toolbox, the ellipsoid algorithm, and the ACCPM. “LMI Tool“ denotes MATLAB LMI Control Toolbox. “ACCPM“ denotes the Analytical Center Cutting Plane Method. “Ellipsoid“ denotes the ellipsoid algorithm. In this set of testing problems, the ACCPM starts with a initial box constraint  $\{x : |x_i| \leq 10^6\}$ , and the ellipsoid algorithm starts with a ball constraints  $\{x : \|x\| \leq 10^6\}$ .

gorithm described in Section 6.5 and the analytical center cutting plane method described in Section 6.6. The equivalent SDP of the optimization problem is solved using MATLAB LMI Control Toolbox. The results are listed in Table 6.4. Again, we see that the ellipsoid algorithm and the analytical center cutting plane method are much faster than the MATLAB LMI Control Toolbox. While the MATLAB LMI Control Toolbox spent more than 8 hours to solve the problem, the ACCPM and the ellipsoid algorithm only took about 5 minutes and 20 minutes, respectively.

## 6.8 Comparison with the Conventional Method

Recall that the IQC optimization problems are in the form of

$$\inf_{\lambda} c'\lambda, \text{ subj. to} \quad \begin{bmatrix} (j\omega I - A)^{-1}B \\ I \end{bmatrix}^* \begin{bmatrix} Q(\lambda) & F(\lambda) \\ F(\lambda)' & R(\lambda) \end{bmatrix} \begin{bmatrix} (j\omega I - A)^{-1}B \\ I \end{bmatrix} > 0 \quad \forall \omega \in [0, \infty]. \quad (6.30)$$

This problem has an equivalent SDP formulation

$$\inf_{P=P', \lambda} c'\lambda, \text{ subj. to} \quad \begin{bmatrix} PA + A'P & PB \\ B'P & 0 \end{bmatrix} + \begin{bmatrix} Q(\lambda) & F(\lambda) \\ F(\lambda)' & R(\lambda) \end{bmatrix} > 0. \quad (6.31)$$

The conventional approach of solving an IQC problem is to transform (6.30) into (6.31) and then solve (6.31) using interior point methods. In this chapter, several cutting plane

algorithms are presented to solve (6.30) directly. It turns out that in certain cases, the cutting plane algorithms can solve an IQC problem much faster than the conventional method does. In this section, we offer some explanations to this phenomenon from the point of view of computational complexity.

Let the number of decision variables in  $\lambda$  be  $n$ . Let the dimension of matrix  $A$  be  $n_x \times n_x$  and the dimension of  $R(\lambda)$  be  $n_r \times n_r$ .

In each iteration of a cutting plane algorithm, the major computation involved is to check whether a trial point is feasible or not. In the case of IQC optimization problems, the most expensive computational procedure performed in feasibility checking is eigenvalue decomposition of the matrix  $R$  and the Hamiltonian matrix  $\mathcal{H}$ . The arithmetic operations for eigenvalue decomposition of an  $m \times m$  square matrix is  $\mathbf{O}(m^3)$ . Since the dimension of the matrix  $R$  is  $n_r \times n_r$  and the dimension of the matrix  $\mathcal{H}$  is  $2n_x \times 2n_x$ , the computational complexity of checking feasibility of a trial point is bounded by  $\mathbf{O}(n_r^3 + 8n_x^3)$ . Suppose that  $n_x = \mathbf{O}(n_r)$ . Then the number of arithmetic operations required for feasibility checking is  $\mathbf{O}(n_x^3)$ . Thus, the computational complexity of each iteration of cutting plane algorithms for solving IQC optimization problems is  $\mathbf{O}(n_x^3)$ , provided  $n_x = \mathbf{O}(n_r)$ . The total complexity of an algorithm is equal to the number of iterations multiplied by the complexity of each iteration. As discussed in Sections 6.5 and 6.6, the number of iterations required for the ellipsoid algorithm and the analytical center cutting plane method to converge to an  $\varepsilon$ -accurate solution are  $\mathbf{O}(n^2 |\log \varepsilon|)$  and  $\mathbf{O}^*(\frac{n^2}{\varepsilon^2})$ , respectively. Therefore, the total computational complexities of the ellipsoid algorithm and the ACCPM are as follows:

$$\text{the ellipsoid algorithm: } \mathbf{O}(n^2 |\log \varepsilon|) \cdot \mathbf{O}(n_x^3),$$

$$\text{the ACCPM: } \mathbf{O}^*\left(\frac{n^2}{\varepsilon^2}\right) \cdot \mathbf{O}(n_x^3).$$

The computational complexity of the interior point methods for solving SDP has been well studied [75, 60]. Suppose that  $n_x = \mathbf{O}(n_r)$ . Then the computational complexity of a single iteration of the interior point method is  $\mathbf{O}(n_x^2(n_x^2 + n)^2)$ , and the total number of iterations required for the algorithm to converge to an  $\varepsilon$ -accurate solution is  $\mathbf{O}(\sqrt{n_x} |\log \varepsilon|)$ . Therefore, the total computational complexity of the interior point algorithm for solving (6.31) is  $\mathbf{O}(\sqrt{n_x} |\log \varepsilon|) \cdot \mathbf{O}(n_x^2(n_x^2 + n)^2)$ .

From the complexity analysis, we observe the fundamental difference between the cutting

plane algorithm and the interior point algorithm for solving IQC optimization problems. In a cutting plane algorithm, the complexity of each iteration is low but the algorithm requires many iterations to converge. On the contrary, the interior point algorithm requires only a few iterations to converge but the computational complexity of each iteration is much higher. The observation from the complexity analysis completely agrees with the numerical experiments presented in the previous section. Now suppose that  $n_x = \mathbf{O}(n)$  and disregard the  $\varepsilon$  term. Then the total complexity of the interior point algorithm is  $\mathbf{O}(n_x^{6.5})$ , while the total complexities of the ellipsoid algorithm and the analytical center cutting plane algorithm are both  $\mathbf{O}(n^2 n_x^3)$ . This indicates that the cutting plane algorithms shall outperform the interior point algorithm when  $n_x = \mathbf{O}(n)$ . Furthermore, the larger the ratio  $n_x/n$  is, the difference in speed is more significant. These two points also agree with what we observe from the results of our numerical experiments.

Finally, we remark that according to our numerical experiments, the total number of iterations that the analytical center cutting plane algorithm requires to find an  $\varepsilon$ -accurate suboptimal solution is closer to  $\mathbf{O}(n|\log \varepsilon|)$  instead of  $\mathbf{O}^*(\frac{n^2}{\varepsilon^2})$ . That is why the ACCPM outperforms the ellipsoid algorithm in all our experiments. However, our attempt to prove that  $\mathbf{O}(n|\log \varepsilon|)$  is indeed the number of iterations for analytical center cutting plane algorithm to converge was not successful.

## 6.9 Summary

We implement three cutting plane algorithms, the Kelley type cutting plane algorithm, the ellipsoid algorithm, and the analytical center cutting plane method to solve feasibility and optimization problems arising from standard IQC analysis. The cutting plane algorithms are potentially very efficient compared to the conventional approach for solving IQC problems. The key idea of achieving the efficiency is to avoid introducing additional decision variables, which are required in the conventional approach. These cutting plane algorithms have been tested on a number of numerical examples. The results indicate that they are indeed much faster than the conventional approach. The difference in speed is very significant when the dimension of the system matrices is large.

The cutting plane algorithms presented in this chapter are programmed in MATLAB. These special solvers for IQC optimization problems, as well as the numerical experiments



shown in this chapter, will soon be incorporated into the *IQC Toolbox*, a MATLAB package developed by the author for robustness analysis of complex dynamical systems [54].

The major disadvantage which the cutting plane algorithms share is that they usually require many iterations to converge. On the other hand, interior point algorithms generally do not have this problem. This motivates our research of constructing efficient interior point algorithms for solving IQC optimization problem. We present this research in the next chapter.



## Chapter 7

# Interior Path-following Algorithms for Standard IQC Problems

In this chapter, we propose two new barrier functions for IQC optimization problems. The new barrier functions are used to construct interior path-following algorithms which follow the basic principles of the standard path-following method. As we already discussed in the previous chapter, the conventional approach to solve IQC optimization problems requires introducing new decision variables. Thanks to the new barrier function, the path-following algorithms proposed in this chapter do not have this requirement. As a result, these algorithms can solve IQC optimization problems in a more efficient fashion. Numerical examples are used to evaluate the efficiency of the proposed algorithms.

### 7.1 Introduction

IQC analysis offers a flexible framework to analyze large complex dynamical systems with structured uncertainties [55, 66, 67]. In IQC analysis method, conditions for robust stability and performance are expressed as feasibility and optimization problems over frequency dependent linear matrix inequalities. The conventional way to treat this type of problems is to transform them into Semi-Definite Programs (SDPs), which can then be solved using interior point methods. This transformation, however, has the disadvantage of introducing additional decision variables. In certain practical problems where the systems to be analyzed have state spaces of large dimension, the number of additional decision variables resulting from the transformation is much larger than the number of the original decision variables.

The additional decision variables hence become the major computational burden, and most of computational effort is inefficiently spent on something auxiliary. In these cases, more efficient algorithms which achieve fast computation are very desirable.

Since the inefficiency is due to the existence of auxiliary decision variables, one approach to improve the efficiency is to avoid introducing these variables. Along this line of thought, we presented several cutting plane algorithms which are able to solve IQC optimization problems in a more efficient fashion in the previous chapter. These algorithms appear to work very well, especially when the number of states in the system to be analyzed is much larger than the number of decision variables. However, a disadvantage which cutting plane methods commonly have is that they generally require many iterations to converge to a suboptimal solution with good accuracy. Furthermore, our experience indicates that the number of iterations grows substantially as the number of decision variables becomes larger.

The interior point method generally does not require many iterations to converge, in contrast to the cutting plane method. The development of interior point methods dates back to the 1950s, with good early reference being [22]. Interior point methods have gained much attention and have become popular since Karmarkar introduced his famous algorithm for solving linear programs [46], not only because his algorithm can be proved to have polynomial time worst-case complexity, but also because it works quite well practically. Another milestone in the development of interior point methods was the result by Nesterov and Nemirovsky [60]. They discovered that Karmarkar's algorithm, as well as several other polynomial time algorithms for solving linear programs, can be extended to solve a much larger class of convex optimization problems. The key element is the knowledge of a barrier function with a certain property called *self-concordance*. To be useful in practice, the barrier must be computable. Nesterov and Nemirovsky have shown that every finite dimensional convex set processes a self-concordant barrier function; however, their universal self-concordant barrier is generally *not* computable. There are only a few classes of problems for which readily computable self-concordant barrier functions are known. For the optimization problems resulting from IQC analysis, the only known computable self-concordant barrier function involves an auxiliary matrix variable which in some cases makes the computational algorithm based on this barrier function very inefficient.

In this chapter, new barrier functions for solving IQC optimization problems are proposed. These barrier functions are readily computable: the main computation to obtain

their first and second derivatives is to solve Lyapunov equations for which efficient computational routines are widely available. Thanks to the new barrier functions, standard interior path-following algorithms can be applied to solve the IQC optimization problems without introducing auxiliary variables, which is why the interior point algorithms based on these newly introduced barrier functions are more efficient than the conventional approach. Regarding the issues of computational complexity, we have attempted to prove that the barrier function is self-concordant in order to apply Nesterov and Nemirovsky's result. Our attempt was not quite successful, and we have not been able to determine whether the path-following algorithms using these new barrier functions are polynomial time algorithms or not. Nevertheless, we are able to show that the algorithms converge globally.

The chapter is organized as follows: following the introduction section, in Section 7.2 we define the notations to be used and formally state the IQC optimization problem under consideration in this chapter. In Section 7.3, the notion of central path is introduced, and the interior path-following algorithm for solving general convex optimization problems is briefly presented. A short discussion of Nesterov and Nemirovsky's complexity analysis for interior path-following algorithms is also given in this section. In Section 7.4, the new barrier functions we propose for IQC optimization problems are presented. We will show how to evaluate the functions and how to compute their gradients and Hessian at a given point. We also prove certain properties of these barrier functions. These properties ensure that the interior path-following algorithms based on the proposed barrier functions converge globally. These algorithms are summarized in Section 7.5. In Section 7.6, we explain why the proposed algorithms are more efficient than the conventional approach from the point of view of computational complexity. The proposed algorithms are tested on some numerical examples. The results and comparison with the conventional approach are presented in Section 7.7. Section 7.8 collects the proofs of some technical results. These proofs are put in the end of the chapter for the sake of a smooth presentation of the material in this chapter.

## 7.2 Notations and Problem Formulation

Given a function  $F(\lambda) : \mathbf{R}^n \rightarrow \mathbf{R}$ , the notations  $\nabla F(\lambda)$  and  $\nabla^2 F(\lambda)$  are used to denote the *gradient vector* and the *Hessian matrix* of  $F(\lambda)$  (the *gradient* and the *Hessian* for short),

respectively. The partial derivative of  $F(\lambda)$  with respect to the  $i^{\text{th}}$  component of  $\lambda$  is denoted by  $\partial_i F(\lambda)$ . The second partial derivative of  $F(\lambda)$  with respect to the  $i^{\text{th}}$  and  $j^{\text{th}}$  components is denoted by  $\partial_{ij}^2 F(\lambda)$ . That is,

$$\partial_i F(\lambda) = \frac{\partial F}{\partial \lambda_i}(\lambda), \quad \partial_{ij}^2 F(\lambda) = \frac{\partial^2 F}{\partial \lambda_i \partial \lambda_j}(\lambda).$$

If  $F(\lambda)$  is at least  $k$  times differentiable, then the notation

$$\nabla^k F(\lambda)[h_1, \dots, h_k]$$

denotes the value of the  $k^{\text{th}}$  differential of  $F$  taken at  $\lambda$  along the collection of directions  $h_1, \dots, h_k$ , where  $h_i \in \mathbf{R}^n$ .

Recall Chapter 6. The standard IQC optimization problem can be formulated as

$$\inf_{\lambda} c' \lambda, \quad \mathbf{H}(\omega, \lambda) > 0, \quad \forall \omega \in [0, \infty], \quad (7.1)$$

where  $\mathbf{H}(\omega, \lambda)$  are of the form

$$\mathbf{H}(\omega, \lambda) := \mathbf{H}_0(\omega) + \sum_{i=1}^n \lambda_i \mathbf{H}_i(\omega).$$

Each  $\mathbf{H}_i(\omega)$  is a self-adjoint, rational transfer matrix of the form

$$\mathbf{H}_i(\omega) = \begin{bmatrix} (j\omega I - A)^{-1} B \\ I \end{bmatrix}^* \begin{bmatrix} Q_i & F_i \\ F_i' & R_i \end{bmatrix} \begin{bmatrix} (j\omega I - A)^{-1} B \\ I \end{bmatrix},$$

where matrices  $Q_i, R_i$  are real symmetric, and matrix  $A$  is Hurwitz; i.e., none of its eigenvalues is in the closed right-half complex plane. Let

$$\Sigma_i = \begin{bmatrix} Q_i & F_i \\ F_i' & R_i \end{bmatrix}, \quad i = 1, \dots, n. \quad (7.2)$$

We assume that  $\Sigma_i$  are linearly independent. This assumption ensures that none of the decision variable  $\lambda_i$  can be removed. The notations  $Q(\lambda)$ ,  $F(\lambda)$ , and  $R(\lambda)$  are used to

denote

$$Q_0 + \sum_{i=1}^n \lambda_i Q_i, \quad F_0 + \sum_{i=1}^n \lambda_i F_i, \quad R_0 + \sum_{i=1}^n \lambda_i R_i, \quad (7.3)$$

respectively. The feasible set of the IQC optimization problems is denoted by  $\Omega$ ; i.e.,

$$\Omega = \{\lambda \mid \mathbf{H}(\omega, \lambda) > 0, \forall \omega \in [0, \infty]\}. \quad (7.4)$$

In this chapter, we propose two new *barrier functions* for  $\Omega$ . Furthermore, standard interior path-following algorithms based on the proposed barrier functions are implemented to solve IQC optimization problem (7.1).

### 7.3 The Interior Path-Following Algorithm

Consider the general convex optimization problem

$$\inf c' \lambda, \quad \text{subj. to } \lambda \in \Lambda \quad (7.5)$$

where  $\Lambda \subset \mathbf{R}^n$  is a bounded open convex set defined by inequalities  $\phi_j(\lambda) > 0$ ,  $j = 1, 2, \dots$ . The number of inequalities could be infinity. In this section, we discuss a class of computational algorithms, called the *sequential unconstrained minimization method*, or the *interior path-following method*, for solving problem (7.5). The method is based on using smooth minimization techniques, often Newton's method, to approximately solve a sequence of smooth, unconstrained problems. The sequence of unconstrained problems is selected in such a way that the approximate solutions of these problems, denoted by  $\lambda_k^*$ , converge to the set of optimal solutions of (7.5). The term *interior point* refers to the fact that  $\lambda_k^*$  are all strictly feasible; i.e.,  $\lambda_k^*$  belongs to the interior of  $\Lambda$  for all  $k$ .

The contents in this section are well-known. The material is adapted and summarized from [10], [60], and [7].

#### 7.3.1 The Barrier Function and the Central Path

Suppose that  $\Lambda$  is non-empty and there exists a function  $B(\lambda)$  defined on  $\Lambda$  such that

- $B(\lambda)$  is convex and smooth.

- $B(\lambda) \rightarrow \infty$  as  $\lambda$  approaches the boundary of  $\Lambda$ .

We refer to  $B(\lambda)$  as a barrier function of  $\Lambda$ . Now, let  $\varphi_t(\lambda)$  be defined as

$$\varphi_t(\lambda) = tc'\lambda + B(\lambda)$$

where  $t > 0$ . Note that  $\varphi_t(\lambda)$  is a convex function since it is a combination of a linear and a convex function.

For any fixed  $t$ , consider the minimization problem

$$\inf \varphi_t(\lambda) \quad \text{subj. to } \lambda \in \Lambda. \quad (7.6)$$

Note that since  $\varphi_t(\lambda)$  is a convex function and  $\Lambda$  is a convex set, problem (7.6) has a unique optimal solution. Suppose that a point  $\lambda_0$  inside  $\Lambda$  is known and a gradient method is applied to solve (7.6) starting at  $\lambda_0$ . It is obvious that the successive iterates of the gradient method will be all strictly feasible and so is the minimizer, since  $B(\lambda)$  approaches infinity as  $\lambda$  approaches to the boundary of  $\Lambda$ . Therefore, the *inf* in (7.6) can be replaced by *min*, and the problem (7.6) is solved as if there is no constraint  $\lambda \in \Lambda$ .

We will refer to problem (7.6) as *centering*. Centering is the main step in an interior path-following algorithm. Let the minimizer of (7.6) be  $\lambda^*(t)$ ; i.e.,

$$\lambda^*(t) = \operatorname{argmin} \{ \varphi_t(\lambda) \mid \lambda \in \Lambda \}. \quad (7.7)$$

For a fixed  $t$ , the argument of minimum  $\lambda^*(t)$  is referred to as the *center* of (7.6). The curve  $\lambda^*(t)$ ,  $t > 0$ , is referred to as the *central path*.

### 7.3.2 The Path-following Algorithm

Note that  $\lambda^*(t)$  is also the minimizer of the problem

$$\min c'\lambda + \frac{1}{t}B(\lambda).$$

The term  $\frac{1}{t}B(\lambda) \rightarrow 0$  as  $t \rightarrow \infty$  for all interior point  $\lambda \in \Lambda$ ; therefore, as  $t$  approaches infinity, the barrier term in (7.6) becomes increasingly inconsequential as far as interior points are concerned. Hence, as  $t$  becomes bigger and bigger, the minimizer  $\lambda^*(t)$  becomes



successively closer to the boundary of  $\Lambda$  and eventually converges to the set of optimal solutions of (7.5).

The idea of the interior path-following algorithm is to follow the central path  $\lambda^*(t)$  to an optimal solution of (7.5). The algorithm forms and solves a sequence of unconstrained problems

$$\min t_k c' \lambda + B(\lambda),$$

where  $t_k = \mu t_{k-1}$ ,  $\mu > 1$ ,  $k = 0, 1, \dots$ . As  $t_k$  is sequentially increased to infinity, the solution  $\lambda_k^*$  becomes closer and closer to an optimal solution of (7.5). The idea of this algorithm was first proposed in the 1960s by Fiacco and McCormick, when it was called the sequential unconstrained minimization method. The basic algorithm can be summarized as follows

**Given:**  $\lambda_0^* \in \Lambda$ .

**Initialization:** set  $k := 1$ ,  $t_1 = 1$ .

**Repeat**

- (1) Centering : solve  $\lambda_k^* = \operatorname{argmin} \varphi_{t_k}(\lambda)$ , starting with  $\lambda_{k-1}^*$ .
- (2) Select a  $\mu > 1$ .
- (3) Set  $t_{k+1} := \mu t_k$ .
- (4) Update  $k := k + 1$ .

**Until**  $t_k$  is large enough.

At each iteration, except the first one, we compute the center point starting from the previously computed center point and then increase  $t$  by a factor  $\mu$ . Various unconstrained minimization methods can be applied to compute the central point in each centering step. A commonly used method is the Newton descent method. We refer to the iterations executed during the centering step as the *inner iterations*.

In the *Centering* step of the above described algorithm, we assume that the exact center point is computed. Practically, however, it is not necessary and in fact not possible to compute the exact center point. Since the central path has no significance other than that it leads to an optimal solution of the original optimization problem (7.5) when  $t$  approaches infinity, the algorithm will still produce a sequence of feasible points that converges to an optimal solution even if the centering is inexact.

The following theorem gives a weak result regarding the convergence property of the interior path-following algorithm.

**Theorem 7.1.** *Every limit point of a sequence  $\{\lambda_k^*\}$  generated by an interior path-following algorithm is a global minimum of the original constrained problem (7.5).*

*Proof.* The proof can be found in Chapter 4 of [7] □

In the algorithm, we assume that a strictly feasible point is available to start with. A feasible point of optimization problem (7.5) can be found using the method discussed in the next subsection.

### 7.3.3 Feasibility and the Phase-I Method

To find a point feasible to optimization problem (7.5), we can solve the following problem

$$\sup_{\lambda, y} y, \quad \text{subj. to } \phi_j(\lambda) - y > 0, \quad \forall j. \quad (7.8)$$

Note that this problem is always feasible. Furthermore, a feasible point of this problem can be easily obtained. For example, we can simply choose  $\lambda^{(0)} = 0$  and  $y^{(0)} < \min_i \phi_i(0)$ . Let  $\eta = [\lambda' \quad y]'$ . We can see that maximization problem (7.8) is also in the form of (7.5) with  $c' = [0 \quad 0 \quad \dots \quad -1]$ . Therefore the problem can be solved using the interior path-following algorithm described in Section 7.3.2 starting at the point  $\eta^{(0)} = [0, \dots, y^{(0)}]$ . If the optimal objective of (7.8) is larger than or equal to 0, then (7.5) has a strictly feasible point. Furthermore, if  $\eta$  is a feasible solution of (7.8) such that  $y \geq 0$ , then  $\lambda$  is a feasible solution of (7.5). This means we can terminate the algorithm anytime when we obtain a solution with a positive  $y$ . On the other hand, if the optimal value of (7.8) is less than 0, then the original optimization problem is infeasible.

There are many variations on this idea which one can adopt to obtain a feasible point. One of them, which is often called the *Phase-I* or the *Big-M method*, is quite useful because of an interesting property. The Phase-I method forms an auxiliary problem similar to (7.8) but adds an inequality  $c'\lambda < M$ :

$$\begin{aligned} \inf_{\lambda, y} \quad & -y, \quad \text{subj. to} \\ & M - c'\lambda > 0 \quad \text{and} \quad \phi_j(\lambda) - y > 0, \quad \forall j, \end{aligned} \quad (7.9)$$

where  $M$  is larger than the optimal objective of (7.5). Suppose that there exists a function  $B(\lambda, y)$  such that

- $B(\lambda, y)$  is a barrier function of the set  $\{(\lambda, y) \mid \phi_j(\lambda) - y > 0, \forall j\}$ .
- $B(\lambda, 0)$  is a barrier function of the set  $\{\lambda \mid \phi_j(\lambda) > 0, \forall j\}$ .

Let  $B(\lambda, y) - \log(M - c'\lambda)$  be the barrier function in the interior path-following algorithm used to solve problem (7.9) for finding a feasible point. Then the central path associated with problem (7.9) must satisfy

$$\nabla_i B(\lambda, y) + \frac{c_i}{M - c'\lambda} = 0, \quad i = 1, \dots, n.$$

If  $(\tilde{\lambda}, \tilde{y})$  is on the central path and  $\tilde{y} = 0$ , then  $\tilde{\lambda}$  satisfies

$$\frac{c_i}{M - c'\tilde{\lambda}} + \nabla_i B(\tilde{\lambda}, 0) = 0, \quad i = 1, \dots, n.$$

Note that the above equalities are the optimality conditions of the minimization problem

$$\min_{\lambda} c'\lambda + (M - c'\lambda)B(\lambda, 0).$$

Thus,  $\tilde{\lambda}$  is the minimizer of the above problem, which in turn implies that  $\tilde{\lambda}$  is on the central path associated with (7.5) (at  $t = 1/(M - c'\tilde{\lambda})$ ). Therefore, when the path-following algorithm returns an approximate center point of (7.9) such that  $y$  is larger than but very close to zero, we immediately obtain a feasible solution of (7.5) which is almost centered.

### 7.3.4 Convergence and Complexity Analysis

We have seen in Section 7.3.2 that the interior path-following algorithm converges to an optimal solution of (7.5). However, the theorem of convergence in Section 7.3.2 is a weak result in the sense that it does not indicate the efficiency of the path-following algorithm or how fast the algorithm converges. In general, a path-following algorithm could be very inefficient and take very long time to converge. One way to measure the efficiency of an algorithm is by its worst-case complexity. An algorithm for a class of optimization problems is called a *polynomial time algorithm* if the number of arithmetic operations required to compute an  $\epsilon$ -accurate suboptimal solution for every instance of the problem is bounded by

a polynomial function which only depends on  $\log(1/\epsilon)$  and the size of problem. Usually, an algorithm is considered to be efficient if it is a polynomial time algorithm.

The introduction of polynomial time interior point methods is one of the most remarkable events in the development of mathematical programming in the 1980s. The first method of this family was suggested for linear programming by Karmarkar [46]. Karmarkar's work is considered as a landmark since his work inspired very intensive and fruitful studies in the subject of polynomial time interior point algorithms. The connection between Karmarkar's algorithm and classical interior path-following algorithms was pointed out by Renegar [68]. He also demonstrated that in the case of a linear programming problem, the path-following algorithm associated with the standard logarithmic barrier converges in polynomial time. Another major breakthrough in this subject is the result by Nesterov and Nemirovsky [60]. They discovered that among all properties of a barrier function, only two are responsible for the polynomiality of the associated interior path-following algorithm. In this subsection, we briefly summarize Nesterov and Nemirovsky's results.

### Self-concordance and Complexity of the Interior Path-following Algorithm

Let  $X \subset \mathbf{R}^n$  be an convex open set. A smooth convex function  $F : X \rightarrow \mathbf{R}$  is called self-concordant with the parameter value  $a$  (or *a-self-concordant* for short) if there exists a constant  $a$  such that the following inequality holds for all  $x \in X$  and for all  $h \in \mathbf{R}^n$ :

$$|\nabla^2 F(x)[h, h, h]| \leq 2a^{-1/2}(\nabla^2 F(x)[h, h])^{\frac{3}{2}}. \quad (7.10)$$

$F(x)$  is called strongly  $a$ -self-concordant if furthermore  $F(x) \rightarrow \infty$  as  $x$  approaches the boundary of  $X$ .

**Example 7.1.** The function  $f(x) = -\ln x$  is a self-concordant function on  $(0, \infty)$ . It can be easily verified that  $|\ddot{f}(x)| \leq 2\dot{f}(x)^{1.5}$ . However,  $f(x)$  is not a strongly self-concordant function since  $f(x) \rightarrow -\infty$  as  $x \rightarrow \infty$ . On the other hand,  $g(x) = x - \ln x$  is a strongly self-concordant function on  $(0, \infty)$  which has the same self-concordant parameter as  $f(x)$  has.

Now, consider again the minimization problem (7.5). Let  $B(\lambda) : \mathbf{R}^n \rightarrow \mathbf{R}$  be a smooth convex function defined on  $\Lambda$  and satisfy

(P1).  $B(\lambda) \rightarrow \infty$  as  $\lambda$  approaches the boundary of  $\Lambda$ .

(P2). There exists a constant  $c_1$  such that inequality  $|\nabla^3 B(\lambda)[h, h, h]| \leq c_1(\nabla^2 B(\lambda)[h, h])^{\frac{3}{2}}$  holds for all  $\lambda \in \Lambda$ , and for all  $h \in \mathbf{R}^n$ .

(P3). There exists a constant  $c_2$  such that inequality  $|\nabla B(\lambda)[h]| \leq c_2(\nabla^2 B(\lambda)[h, h])^{\frac{1}{2}}$  holds for all  $\lambda \in \Lambda$ , and for all  $h \in \mathbf{R}^n$ .

By properties (P1) and (P2), the barrier function  $B(\lambda)$  is strongly self-concordant. Note that a scaling  $B(\lambda) \rightarrow cB(\lambda)$  updates the constants  $c_1$  and  $c_2$  as follows

$$c_1 \rightarrow c^{-1/2}c_1, \quad c_2 \rightarrow c^{1/2}c_2.$$

Therefore, one can enforce one of the constants to be a prescribed value. Nesterov and Nemirovsky refer to a barrier function which satisfies all three properties with  $c_1 = 2$  (This implies that  $B(\lambda)$  is a strongly 1-self-concordant function.) and  $c_2 = \vartheta$  as a  $\vartheta$ -self-concordant barrier.

**Example 7.2.** Let  $\Lambda$  be a bounded convex polytope  $\{\lambda \mid a'_i \lambda < b_i, 1 \leq i \leq m\}$ . Then the standard logarithmic barrier function for  $\Lambda$

$$B(\lambda) = - \sum_{i=1}^m \ln(b_i - a'_i \lambda)$$

is an  $m$ -self-concordant barrier.

Nesterov and Nemirovsky gave a complete complexity analysis of an interior path-following algorithm based on a self-concordant barrier. Their results are summarized in the following theorems. The proofs of these theorems can be found in Chapters 2 and 3 of their book [60].

**Definition 7.1.** Let  $F(x)$  be an  $a$ -self-concordant function. The *Newton decrement* of  $F$  at a point  $x$  is defined as

$$\rho(x) := \left( \frac{1}{a} \nabla^2 F(x)[e(x), e(x)] \right)^{\frac{1}{2}},$$

where  $e(x) := -\nabla^2 F(x)^{-1} \nabla F(x)$  is often referred to as the Newton descent direction at  $x$ .

**Theorem 7.2.** Let  $F(x) : X \subset \mathbf{R}^n \rightarrow \mathbf{R}$  be a strongly  $a$ -self-concordant function, and  $\rho(x)$  be the Newton decrement of  $F(x)$ .

1. If there exists  $x \in X$  such that  $\rho(x) < 1$ , then  $F(x)$  attains its minimum over  $X$ ; i.e.,  $\operatorname{argmin}_X F(x) \neq \emptyset$ .
2. Let  $\tilde{x} \in X$  such that  $\rho(\tilde{x}) < \infty$ . Let  $\rho^* = 2 - \sqrt{3}$  and consider the following Newton iteration starting at  $\tilde{x}$

$$x_0 = \tilde{x}; \quad x_{i+1} = x_i + \sigma(\rho(x_i))e(x_i)$$

where

$$\sigma(\rho(x_i)) = \begin{cases} 1, & \text{if } \rho(x_i) < \rho^* \\ 1/(1 + \rho(x_i)), & \text{if otherwise.} \end{cases}$$

Then, we have  $\rho(x_{i+1}) \leq \frac{1}{2}\rho(x_i)$ , if  $\rho(x_i) < \rho^*$ . If  $\rho(x_i) \geq \rho^*$  then  $F(x_i) - F(x_{i+1}) \geq a(\rho^* - \ln(1 + \rho^*))$ .

3. Let  $x^* \in \operatorname{argmin}_X F(x)$ , and let  $x_i$  be such that  $\rho(x_i) < 1/3$ . Then

$$F(x_i) - F(x^*) \leq \frac{a\tau^2(x_i)(1 + \tau(x_i))}{2(1 - \tau(x_i))}, \quad (7.11)$$

where  $\tau(x_i) = 1 - (1 - 3\rho(x_i))^{1/3}$ .

**Theorem 7.3.** Let  $B(\lambda)$  be a  $\vartheta$ -self-concordant barrier for problem (7.5), and  $\varphi_t(\lambda) = tc'\lambda + B(\lambda)$ . Let  $\mu$  and  $s$  be constants such that  $\mu > s > 1$  and  $\kappa = \mu/s$ .

1. Let  $\nu > 0$  and  $\bar{\lambda} \in \Lambda$ . If  $\rho_s(\bar{\lambda}) < \nu$ , and

$$\left(1 + \frac{\vartheta^{1/2}}{\nu}\right) \ln \kappa \leq 1 - \frac{\rho_s(\bar{\lambda})}{\nu},$$

then  $\rho_\mu(\bar{\lambda}) \leq \nu$ . Here  $\rho_\mu(\lambda)$  and  $\rho_s(\lambda)$  are the Newton decrements of  $\varphi_\mu(\lambda)$  and  $\varphi_s(\lambda)$ , respectively.

2. Let  $\bar{\rho} \in (0, 1)$  and  $\bar{\lambda} \in \Lambda$  be such that  $\rho_s(\bar{\lambda}) \leq \bar{\rho}$ . Then

$$\varphi_\mu(\bar{\lambda}) - \varphi_\mu(\lambda^*(\mu)) \leq \mathbf{O}(1)(|\kappa - 1|^{\vartheta^{1/2}} + 1) + \vartheta(\kappa - 1 - \ln \kappa).$$

where  $\lambda^*(\mu)$  denotes the minimizer of  $\varphi_\mu(\lambda)$ , and  $\mathbf{O}(1)$  depends only on  $\bar{\rho}$ .

**Theorem 7.4.** *Consider problem (7.5). Let  $\lambda^*(t)$  be the central path generated by a  $\vartheta$ -self-concordant barrier and  $p^*$  be the optimal objective of (7.5). Then  $c'\lambda^*(t) - p^* \leq \vartheta/t$ .*

In the first outer iteration of the interior path-following algorithm, we minimize the function

$$\varphi_1(\lambda) = c'\lambda + B(\lambda).$$

Suppose that  $B(\lambda)$  is a  $\vartheta$ -self-concordant barrier. Then,  $\varphi_1(\lambda)$  is a strongly 1-self-concordant function since the second and the third derivatives of  $\varphi_1(\lambda)$  are identical to those of  $B(\lambda)$ . In fact,  $\varphi_t(\lambda)$  is 1-self-concordant for all  $t$ . Therefore, the Newton's method described in the second statement of Theorem 7.2 allows us to compute an  $\epsilon$ -approximation of the center  $\lambda^*(1)$  in

$$\left\lceil \frac{\varphi_1(\lambda_0) - \varphi_1(\lambda^*(1))}{\delta} \right\rceil + \mathbf{O}(\ln |\epsilon|) \quad (7.12)$$

Newton iterations, where  $\delta := \rho^* - \ln(1 + \rho^*) \approx 0.0305$ , and  $\lambda_0$  is the initial point at which the Newton's method starts. To see (7.12), note that the Newton's method described in Theorem 7.2 can be divided into two stages: the first one corresponds to  $\rho_1(\lambda) \geq \rho^*$  and the second one corresponds to  $\rho_1(\lambda) < \rho^*$ , where  $\rho_1(\lambda)$  is the Newton decrement of  $\varphi_1(\lambda)$ . At the first stage,  $\varphi_1(\lambda)$  decreases in each iteration by a quantity that is guaranteed to be at least  $\rho^* - \ln(1 + \rho^*)$ . Since  $\varphi_1(\lambda)$  is bounded below, therefore, in at most

$$\left\lceil \frac{\varphi_1(\lambda_0) - \varphi_1(\lambda^*(t))}{\delta} \right\rceil \quad (7.13)$$

iterations,  $\rho_1(\lambda)$  has to become smaller than  $\rho^*$ , and the Newton's method enters the second stage. At the second stage, the quantities  $\rho_1(\lambda_i)$  decrease quadratically. Hence, an  $\epsilon$ -accurate suboptimal solution is found in  $\mathbf{O}(|\log \epsilon|)$  Newton iterations after the Newton's method enters the second stage.

In each outer iteration (except the first one), with an increased  $t$ , the path-following algorithm recomputes an approximate center point starting at the previously computed one. Theorem 7.3 relates the increment in  $t$  and the number of Newton iterations required to obtain an approximation of the new center. Suppose that the value of  $t$  is increased from  $s$  to  $s^+$ . Furthermore, suppose that an approximate center for  $\lambda^*(s)$  (denoted by  $\bar{\lambda}_s$ )

satisfies  $\rho_s(\bar{\lambda}_s) \leq \frac{1}{2}\rho^*$ . Note the inequality implies that  $\bar{\lambda}_s$  is in the quadratically convergent stage of minimizing  $\varphi_s(\lambda)$ . By the first statement of Theorem 7.3, if  $\nu := \rho^*$  and

$$1 < \kappa := \frac{s^+}{s} \leq \exp\left(\frac{\rho^* - \rho_s(\bar{\lambda}_s)}{\rho^* + \vartheta^{1/2}}\right), \quad (7.14)$$

then  $\rho_{s^+}(\bar{\lambda}_s) \leq \rho^*$ . In other words, if  $\kappa$  is selected according to (7.14), then the current solution  $\bar{\lambda}_s$  remains within the quadratically convergent stage of minimizing  $\varphi_{s^+}(\lambda)$ . Therefore, only a single Newton iteration is required to obtain an approximate center  $\bar{\lambda}_{s^+}$  for  $\lambda^*(s^+)$  such that  $\rho_{s^+}(\bar{\lambda}_{s^+}) \leq \frac{1}{2}\rho^*$ . This selection of  $\kappa$  usually referred to as the *small step strategy*. The advantage of small step strategy is that only one Newton iteration is required for re-centering. However, since  $\kappa$  in this selection is usually very close to 1, therefore, many outer iterations are required for the path-following algorithm to converge.

On the other hand, the *large step strategy* chooses a larger  $\kappa$  which might make  $\rho_{s^+}(\bar{\lambda}_s) \geq \rho^*$ . In this case,  $\bar{\lambda}_s$  is no longer in the quadratically convergent stage of minimizing  $\varphi_{s^+}(\lambda)$ , and more than one Newton step is required to restore the centrality. The second statement of Theorem 7.3 provides an upper bound on  $\varphi_{s^+}(\bar{\lambda}_s) - \varphi_{s^+}(\lambda^*(s^+))$ . This upper bound in turn gives an estimate of  $\mathcal{O}(\vartheta)$  Newton steps for restoring the centrality. Thus, by the large step strategy, we make a more rapid progress in increasing  $t$  at the price that more Newton's steps are required for re-centralization each time when the value of  $t$  is increased.

Finally, Theorem 7.4 implies that the algorithm converges to a central point which is an guaranteed  $\varepsilon$ -accurate suboptimal solution in  $\mathcal{O}(\log \frac{\vartheta}{\varepsilon})$  outer iterations.

*Remark 7.1.* We note that only properties (P1) and (P2) (i.e., the strongly self-concordance) of a barrier function are required to establish Theorem 7.2. Furthermore, the first statement of Theorem 7.3 and Theorem 7.4 are established only by property (P3).

## 7.4 New Barrier Functions for IQC Optimization Problems

Now let us consider the IQC optimization problem (7.1). It is well-known [9] that the frequency dependent matrix inequality in (7.1) holds if and only if there exists a symmetric matrix  $P$  such that

$$E(P, \lambda) := \begin{bmatrix} PA + A'P & PB \\ B'P & 0 \end{bmatrix} + \Sigma_0 + \sum_{i=1}^n \lambda_i \Sigma_i > 0. \quad (7.15)$$



Hence, problem (7.1) is equivalent to a SDP

$$\inf_{P=P', \lambda} c' \lambda, \quad E(P, \lambda) > 0. \quad (7.16)$$

The feasible set of problem (7.16) has a well-known self-concordant barrier

$$-\log \det(E(P, \lambda)), \quad (7.17)$$

where the notation  $\det(M)$  denotes the determinate of the matrix  $M$ . The barrier function (7.17), however, involves an additional matrix variable  $P$ . The decision variables in  $P$  is proportional to the square of the dimension of the matrix  $A$ . Therefore, when the dimension of the matrix  $A$  is significantly larger than the square root of the number of the original decision variables, the decision variables in  $P$  become the major computational burden for an interior point algorithm based on barrier function (7.17). In these cases, the interior point algorithm is inefficient in the sense that the computational effort is spent on solving auxiliary decision variables.

The main purpose of this chapter is to propose alternative barrier functions which can be used to construct *more* efficient interior point algorithms for solving IQC optimization problems. We propose two new barrier functions. They are described in this section.

#### 7.4.1 The First Barrier Function

We propose the following function

$$B_1(\lambda) := \frac{1}{\pi} \int_{-\infty}^{\infty} \text{tr}(\mathbf{H}(\omega, \lambda)^{-1}) \frac{d\omega}{1 + \omega^2}, \quad \lambda \in \Omega, \quad (7.18)$$

as a candidate barrier function for IQC optimization problems. It is obvious that  $B_1(\lambda)$  is smooth. The  $i^{\text{th}}$  element of the gradient and the  $(i, j)$  entry of the Hessian of  $B_1(\lambda)$  are given as follows:

$$\partial_i B_1(\lambda) = -\frac{1}{\pi} \int_{-\infty}^{\infty} \text{tr}(\mathbf{H}(\omega, \lambda)^{-1} \mathbf{H}_i(\omega) \mathbf{H}(\omega, \lambda)^{-1}) \frac{d\omega}{1 + \omega^2}, \quad (7.19)$$

$$\partial_{i,j}^2 B_1(\lambda) = \frac{2}{\pi} \int_{-\infty}^{\infty} \text{tr}(\mathbf{H}(\omega, \lambda)^{-1} \mathbf{H}_i(\omega) \mathbf{H}(\omega, \lambda)^{-1} \mathbf{H}_j(\omega) \mathbf{H}(\omega, \lambda)^{-1}) \frac{d\omega}{1 + \omega^2}. \quad (7.20)$$

Furthermore, it is not difficult to verify that  $B_1(\lambda)$  is a convex function.

**Proposition 7.1.**  $B_1(\lambda)$  is a convex function.

*Proof.* We prove  $B_1(\lambda)$  is convex by showing that the Hessian of  $B_1(\lambda)$  is strictly positive definite; i.e.,  $\nabla^2 B_1(\lambda) > 0$ , for all  $\lambda \in \Omega$ . By (7.20), we have<sup>1</sup>

$$\begin{aligned} \nabla^2 B_1(\lambda)[h, h] &= \sum_{i,j} \partial_{i,j}^2 B_1(\lambda) h_i h_j = \frac{2}{\pi} \int_{-\infty}^{\infty} \text{tr}(\mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1}) \frac{d\omega}{1 + \omega^2} \\ &= \frac{2}{\pi} \int_{-\infty}^{\infty} \left\| \mathbf{H}^{-\frac{1}{2}} \mathcal{A} \mathbf{H}^{-1} \right\|_F^2 \frac{d\omega}{1 + \omega^2}, \end{aligned}$$

where  $\mathcal{A} = \sum_{i=1}^n h_i \mathbf{H}_i(\omega)$ . Therefore  $\nabla^2 B_1(\lambda)[h, h] > 0$  for all  $\lambda \in \Omega$  and for all  $h \neq 0$ . This concludes the proof.  $\square$

Furthermore, we can show that the value of  $B_1(\lambda)$  approaches infinity as  $\lambda$  approaches the boundary of  $\Omega$ . This property, together with smoothness and convexity, makes  $B_1(\lambda)$  a natural barrier for  $\Omega$ . Before we prove this property, we first discuss how to evaluate  $B_1(\lambda)$ .

### Computation of the value, the gradient and the Hessian of $B_1(\lambda)$

Although the integration over an infinite horizon makes  $B_1(\lambda)$  seemingly difficult to evaluate at any given  $\lambda$ , the evaluation can be performed by using a rather efficient computational procedure. The main computation for evaluating  $B_1(\lambda)$  is to solve one Riccati equation and one Lyapunov equation. Moreover, efficient computation of the gradient and the Hessian of  $B_1(\lambda)$  at a given  $\lambda$  can be also performed by a similar approach. We here present these computational procedures.

Let  $\tilde{\lambda}$  belong to the feasible set  $\Omega$ . Therefore  $\mathbf{H}(\omega, \tilde{\lambda}) > 0 \forall \omega$ . Let  $\tilde{Q}$ ,  $\tilde{F}$ ,  $\tilde{R}$  denote constant matrices  $Q(\tilde{\lambda})$ ,  $F(\tilde{\lambda})$ , and  $R(\tilde{\lambda})$ , where  $Q(\lambda)$ ,  $F(\lambda)$ , and  $R(\lambda)$  are defined in (7.3). The following lemma is essential for obtaining an efficient computational procedure to evaluate  $B_1(\tilde{\lambda})$ .

**Lemma 7.1.** Assume that  $A \in \mathbf{R}^{m \times m}$  is a Hurwitz matrix. The following three statements are equivalent

$$1 \quad \mathbf{H}(\omega, \tilde{\lambda}) > 0 \quad \forall \omega \in [0, \infty].$$

<sup>1</sup>The  $\omega$ ,  $\lambda$  dependence of  $\mathbf{H}(\omega, \lambda)$  and  $\mathbf{H}_i(\omega)$  are suppressed for simplifying the notation.

2  $\mathbf{H}(\omega, \tilde{\lambda})^{-1}$  can be factorized as  $D_H + G_H(j\omega) + G_H(j\omega)^*$ , where

$$G_H(s) = C_H(sI - A_H)^{-1}B_H, \quad (7.21)$$

$$A_H = A + BC_H, \quad B_H = B\tilde{R}^{-1} + YC'_H, \quad (7.22)$$

$$C_H = -\tilde{R}^{-1}(PB + \tilde{F})', \quad D_H = \tilde{R}^{-1}, \quad (7.23)$$

and  $P, Y$  satisfy the following Riccati and Lyapunov equations, respectively

$$PA + A'P + \tilde{Q} - (PB + \tilde{F})\tilde{R}^{-1}(PB + \tilde{F})' = 0, \quad (7.24)$$

$$A_H Y + Y A'_H + B\tilde{R}^{-1}B' = 0. \quad (7.25)$$

Furthermore, matrix  $A_H$  is a Hurwitz matrix and has the same dimension of matrix  $A$ .

3  $\mathbf{H}(\omega, \tilde{\lambda})^{-1}$  can be factorized as  $\Psi(j\omega)\Psi(j\omega)^*$ , where

$$\Psi(s) = \tilde{R}^{\frac{1}{2}} + C_H(sI - A_H)^{-1}B\tilde{R}^{\frac{1}{2}}. \quad (7.26)$$

*Proof.* See [92]. □

Given any  $\tilde{\lambda} \in \Omega$ ,  $B_1(\tilde{\lambda})$  can be evaluated using the formula in the following lemma.

**Lemma 7.2.** *Let  $\tilde{\lambda} \in \Omega$ . Then  $B_1(\tilde{\lambda}) = \text{tr}(D_H) + 2\text{tr}(C_H(I - A_H)^{-1}B_H)$ , where  $A_H, B_H, C_H, D_H$  are defined in (7.22) to (7.25).*

*Proof.* First, notice that the order of the trace operator and the integral operator can be reversed; therefore by the second statement of Lemma 7.1, we have

$$\begin{aligned} B_1(\tilde{\lambda}) &= \text{tr} \left( \frac{1}{\pi} \int_{-\infty}^{\infty} (D_H + G_H(j\omega) + G_H(j\omega)^*) \frac{d\omega}{1 + \omega^2} \right) \\ &= \text{tr}(D_H) + \text{tr} \left( \frac{1}{\pi} \int_{-\infty}^{\infty} (G_H(j\omega) + G_H(j\omega)^*) \frac{d\omega}{1 + \omega^2} \right). \end{aligned}$$

Now, if we treat  $G_H(j\omega)$  as the Fourier transform of the stable causal system  $G_H(s)$  and let  $g(t)$  be the impulse response of  $G_H(s)$ , then we have

$$\frac{1}{\pi} \int_{-\infty}^{\infty} G_H(j\omega) \frac{1}{1 + \omega^2} d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} G_H(j\omega) \frac{2}{1 + \omega^2} d\omega = g(t) * e^{-|t|} \Big|_{t=0},$$

where  $*$  denotes the convolution operator. Since  $G_H(s)$  is stable and causal, we have

$$g(t) * e^{-|t|} \Big|_{t=0} = \int_0^\infty g(\tau) e^{-\tau} d\tau = G_H(1) = C_H(I - A_H)^{-1} B_H.$$

Similarly, we have

$$\frac{1}{\pi} \int_{-\infty}^\infty G_H(j\omega) * \frac{1}{1 + \omega^2} d\omega = C_H(I - A_H)^{-1} B_H.$$

Thus we conclude that  $B_1(\tilde{\lambda})$  is equal to the formula stated in the lemma.  $\square$

Therefore, computation of  $B_1(\tilde{\lambda})$  mainly involves solving Riccati equation (7.24) and Lyapunov equation (7.25). Solving Riccati equations and Lyapunov equations has been well studied, and efficient computational routines are widely available. Thus, evaluation of  $B_1(\lambda)$  at a given  $\lambda$  can be performed very efficiently.

We are now ready to show that that  $B_1(\lambda)$  approaches to infinity as  $\lambda$  approaches to the boundary of  $\Omega$ .

**Proposition 7.2.** *Let  $\{\lambda_n\}_{n=1}^\infty$  be a set of points strictly inside  $\Omega$  such that  $\lambda_n$  approaches the boundary of  $\Omega$  as  $n \rightarrow \infty$ . Then  $B_1(\lambda_n) \rightarrow \infty$ .*

*Proof.* Let  $\tilde{\lambda}$  belong to the boundary  $\Omega$ ; i.e.,  $\mathbf{H}(\omega, \tilde{\lambda})$  is only semi-positive definite and singular at the set  $\Gamma = \{\pm\omega_i, i = 1, \dots, m\}$ . It is a well-known result in the area of systems and control that, in this case, either  $\tilde{R} := R(\tilde{\lambda})$  is singular (if  $\infty \in \Gamma$ ), or  $A_H$  has pure imaginary eigenvalues  $\{\pm j\omega_i, i = 1, \dots, m\}$  (if  $\infty \notin \Gamma$ ). If  $\tilde{R}$  is singular, we see from Lemma 7.2 that  $B_1(\tilde{\lambda})$  is unbounded since  $D_H = \tilde{R}^{-1}$  and  $\tilde{R}$  is not invertible. If  $\tilde{R}$  is not singular, then  $\mathbf{H}(\omega, \tilde{\lambda})$  can be factorized as  $\Psi(j\omega)\Psi(j\omega)^*$ , where  $\Psi(s)$  is defined in (7.26).

Note that

$$\frac{1}{\pi} \int_{-\infty}^\infty \text{tr}(\mathbf{H}(\omega, \tilde{\lambda})^{-1}) \frac{d\omega}{1 + \omega^2} = \frac{1}{\pi} \int_{-\infty}^\infty \text{tr}(\Psi(j\omega)\Psi(j\omega)^*) \frac{d\omega}{1 + \omega^2} = 2\|\tilde{\Psi}(s)\|_{\mathcal{H}_2}, \quad (7.27)$$

where  $\tilde{\Psi}(s) = \frac{1}{s+1}\Psi(s)$ . Since  $A_H$  has eigenvalues on the imaginary axis,  $\tilde{\Psi}(s)$  is not a stable transfer matrix. Therefore, its  $\mathcal{H}_2$ -norm is unbounded, which in turn implies that  $B_1(\tilde{\lambda})$  is unbounded. This concludes the proof.  $\square$

Lemma 7.2 gives the following equivalent expression for  $B_1(\lambda)$

$$B_1(\lambda) = \text{tr}(D_H(\lambda) + 2C_H(\lambda)(I - A_H(\lambda))^{-1}B_H(\lambda)), \quad (7.28)$$

where  $A_H(\lambda)$ ,  $B_H(\lambda)$ ,  $C_H(\lambda)$ , and  $D_H(\lambda)$  are defined as in (7.21) to (7.25) with  $\tilde{Q}$ ,  $\tilde{F}$  and  $\tilde{R}$  replaced by  $Q(\lambda)$ ,  $F(\lambda)$ , and  $R(\lambda)$ . By partially differentiating (7.28) with respect to  $\lambda_i$ , we obtain the following expressions for the  $i^{\text{th}}$  component of the gradient of  $B_1(\lambda)$

$$\begin{aligned} \partial_i B_1(\lambda) &= \text{tr}(\partial_i D_H) + 2((\partial_i C_H)(I - A_H)^{-1}B_H - C_H(I - A_H)^{-1}(\partial_i A_H)(I - A_H)^{-1}B_H \\ &\quad + C_H(I - A_H)^{-1}(\partial_i B_H)) \\ &= \text{tr}(\partial_i D_H + 2(I + C_H(I - A_H)^{-1}B)((\partial_i C_H)(I - A_H)^{-1}B_H) \\ &\quad + 2C_H(I - A_H)^{-1}(\partial_i B_H)), \end{aligned} \quad (7.29)$$

where the second equality is obtained by noting that  $\partial_i A_H = B(\partial_i C_H)$ . It can be easily verified that the partial derivatives of  $B_H$ ,  $C_H$ , and  $D_H$  have the following expressions

$$\partial_i B_H = (\partial_i Y)C_H' + Y(\partial_i C_H)', \quad (7.30)$$

$$\partial_i C_H = R^{-1}R_i R^{-1}(PB + F)' - R^{-1}((\partial_i P)B + F_i)', \quad (7.31)$$

$$\partial_i D_H = -R^{-1}R_i R^{-1}, \quad (7.32)$$

and the partial derivatives of  $P$  and  $Y$  satisfy the following equations

$$(\partial_i P)A_H + A_H'(\partial_i P) + (Q_i + F_i C_H + C_H' F_i' + C_H' R_i C_H) = 0, \quad (7.33)$$

$$A_H(\partial_i Y) + (\partial_i Y)A_H' + (B(\partial_i C_H)Y + Y(\partial_i C_H)'B' - BR^{-1}R_i R^{-1}B') = 0. \quad (7.34)$$

For a given point  $\tilde{\lambda} \in \Omega$ , computation of  $\nabla_i B_1(\tilde{\lambda})$  can be performed as follows: first, notice that for a fixed  $\tilde{\lambda}$ , equation (7.33) is a Lyapunov equation with respect to  $\partial_i P$ . Thus, the value of  $\partial_i P$  can be obtained by solving the Lyapunov equation. Then the values of  $\partial_i D_H$ ,  $\partial_i C_H$  can be computed according to expressions (7.32) and (7.31), respectively. As soon as the value of  $\partial_i C_H$  is available, one can solve another Lyapunov equation (7.34) to obtain the value of  $\partial_i Y$  and then evaluate  $\partial_i B_H$  using expression (7.30). Finally,  $\partial_i B_1(\tilde{\lambda})$  can be evaluated using equation (7.29). Hence, the main computation for obtaining the value of

every entry of the gradient of  $B_1(\lambda)$  is to solve two Lyapunov equations.

If we further differentiate (7.29) with respect to  $\lambda_j$ , we obtain an expression for  $\partial_{ij}B_1(\lambda)$

$$\partial_{ij}B_1(\lambda) = \text{tr}(\partial_{ij}^2D_H) + 2\text{tr}(C_H(I - A_H)^{-1}(\partial_{ij}^2B_H) + (I + C_H(I - A_H)^{-1}B)T_1), \quad (7.35)$$

where

$$\begin{aligned} T_1 &= (\partial_{ij}^2C_H)(I - A_H)^{-1}B_H + (\partial_iC_H)(I - A_H)^{-1}((\partial_jB_H) + B(\partial_jC_H)(I - A_H)^{-1}B_H) \\ &\quad + (\partial_jC_H)(I - A_H)^{-1}((\partial_iB_H) + B(\partial_iC_H)(I - A_H)^{-1}B_H), \end{aligned} \quad (7.36)$$

$$\partial_{ij}^2B_H = (\partial_{ij}^2Y)C'_H + Y(\partial_{ij}^2C_H)' + (\partial_iY)(\partial_jC_H)' + (\partial_jY)(\partial_iC_H)', \quad (7.37)$$

$$\partial_{ij}^2C_H = -R^{-1}(R_j(\partial_iC_H) + R_i(\partial_jC_H) + B'(\partial_{ij}^2P)'), \quad (7.38)$$

$$\partial_{ij}^2D_H = R^{-1}R_iR^{-1}R_jR^{-1} + R^{-1}R_jR^{-1}R_iR^{-1}, \quad (7.39)$$

$\partial_{ij}^2P$  and  $\partial_{ij}^2Y$  satisfy the following Lyapunov equations

$$(\partial_{ij}^2P)A_H + A'_H(\partial_{ij}^2P) + (T_2 + T_2') = 0, \quad (7.40)$$

$$(\partial_{ij}^2Y)A_H + A'_H(\partial_{ij}^2Y) + (T_3 + T_3') = 0, \quad (7.41)$$

where  $T_2$  and  $T_3$  denote the following expressions

$$T_2 = ((\partial_iP)B + C_H R_i + F_i)(\partial_jC_H), \quad (7.42)$$

$$T_3 = B((\partial_iC_H)(\partial_jY) + (\partial_jC_H)(\partial_iY) + (\partial_{ij}^2C_H)Y) + BR^{-1}R_iR^{-1}R_jR^{-1}B'. \quad (7.43)$$

For a given  $\tilde{\lambda} \in \Omega$ , the computational procedure for evaluating  $\partial_{ij}B_1(\tilde{\lambda})$  is similar to the one for computing  $\nabla_i B_1(\tilde{\lambda})$ . Assume that the values of first partial derivatives of  $B_H$ ,  $C_H$ ,  $D_H$ ,  $P$ , and  $Y$  are available. Then,  $T_2$  can be evaluated, and Lyapunov equation (7.40) can be solved for the value of  $\partial_{ij}^2P$ . As soon as the value of  $\partial_{ij}^2P$  is available, one can evaluate  $\partial_{ij}^2B_H$ ,  $\partial_{ij}^2C_H$ , and  $\partial_{ij}^2D_H$  using expressions (7.37) to (7.39). Once the value of  $\partial_{ij}^2C_H$  is obtained, Lyapunov equation (7.41) can be solved for the value of  $\partial_{ij}^2Y$ . Finally,  $\nabla_{ij}^2B_1(\tilde{\lambda})$  can be computed according to expression (7.35). Thus, we see that the main computation for obtaining the value of every entry of the Hessian of  $B_1(\lambda)$  is to solve two Lyapunov equations.

## Properties Related to Polynomial Time Complexity

In order to apply Nesterov and Nemirovsky's results to construct a polynomial time interior path-following algorithm, the barrier function  $B_1(\lambda)$  has to satisfy properties (P1) to (P3). We have already shown that  $B_1(\lambda)$  satisfies (P1) in Proposition 7.2. However, the following counter-example demonstrates that in general,  $B_1(\lambda)$  does not satisfy property (P3).

**Example 7.3.** Let  $\mathbf{H}(\omega, \lambda) := \lambda - \frac{1}{\omega^2 + 1}$ . It can be readily verified that the set  $\Omega$  defined by  $\{\lambda \mid \mathbf{H}(\omega, \lambda) > 0, \forall \omega \in [0, \infty]\}$  is the open interval  $\lambda > 1$ . It can also be easily verified, using Lemma 7.2, that  $B_1(\lambda) = 2/\sqrt{\lambda(\lambda - 1)}$ . Therefore, the first and second derivatives of  $B_1(\lambda)$  can be computed straightforwardly:

$$\begin{aligned}\dot{B}_1(\lambda) &= -\frac{2\lambda - 1}{(\lambda(\lambda - 1))^{\frac{3}{2}}}, \\ \ddot{B}_1(\lambda) &= \frac{1}{2} \frac{8\lambda^2 - 8\lambda + 3}{(\lambda(\lambda - 1))^{\frac{5}{2}}}.\end{aligned}$$

Let

$$r_1 := \frac{\dot{B}_1(\lambda)^2}{\ddot{B}_1(\lambda)} = \frac{2(2\lambda - 1)^2}{(\lambda(\lambda - 1))^{\frac{1}{2}}(8\lambda^2 - 8\lambda + 3)}.$$

Obviously,  $r_1 \rightarrow \infty$  as  $\lambda \rightarrow 1$ . Thus, there exists no constant  $c$  such that  $|\dot{B}_1(\lambda)| \leq c \cdot (\ddot{B}_1(\lambda))^{\frac{1}{2}}, \forall \lambda \in (1, \infty)$ .

Therefore, we can not apply Nesterov and Nemirovsky's results to construct a polynomial time interior path-following algorithm using barrier function  $B_1(\lambda)$ . Nevertheless,  $B_1(\lambda)$  is still a well-defined barrier for  $\Omega$  and can be used to construct an interior path-following algorithm for solving IQC optimization problems. Furthermore, Theorem 7.1 guarantees that the interior path-following algorithm based on  $B_1(\lambda)$  converges globally.

### 7.4.2 The Second Barrier Function

In this subsection, we present the second barrier function we propose for IQC optimization problems. Consider the following function defined on  $\Omega$

$$B_2(\lambda) = \log B_1(\lambda). \tag{7.44}$$

Obviously,  $B_2(\lambda)$  is a smooth function, and  $B_2(\lambda)$  approaches infinity as  $\lambda$  approaches the boundary of  $\Omega$  since  $B_1(\lambda)$  does. The following proposition shows that  $B_2(\lambda)$  is also a convex function. Hence,  $B_2(\lambda)$  is a well-defined barrier for  $\Omega$ .

**Proposition 7.3.**  $B_2(\lambda)$  is a convex function.

*Proof.* We will show that the Hessian of  $B_2(\lambda)$  is strictly positive definite; i.e.,  $\nabla^2 B_2(\lambda) > 0$ , for all  $\lambda \in \Omega$ . It can be easily verified that

$$\nabla^2 B_2(\lambda) = B_1^{-1}(\lambda) \nabla^2 B_1(\lambda) - B_1^{-2}(\lambda) \nabla B_1(\lambda) \nabla B_1(\lambda)'$$

Since  $B_1(\lambda) > 0$  for any  $\lambda \in \Omega$ , therefore, given any  $\lambda \in \Omega$ ,  $\nabla^2 B_2(\lambda) > 0$  if and only if  $\nabla^2 B_1(\lambda) - B_1^{-1}(\lambda) \nabla B_1(\lambda) \nabla B_1(\lambda)' > 0$ . To prove  $\nabla^2 B_1(\lambda) - B_1^{-1}(\lambda) \nabla B_1(\lambda) \nabla B_1(\lambda)' > 0$ , it is sufficient to show that

$$\begin{bmatrix} \nabla^2 B_1(\lambda) & \nabla B_1(\lambda) \\ \nabla B_1(\lambda)' & B_1(\lambda) \end{bmatrix} > 0. \quad (7.45)$$

Matrix inequality (7.45) holds if  $\nabla^2 B_1(\lambda)[h, h] + 2\nabla B_1(\lambda)[h] + B_1(\lambda) > 0$ , for all  $h \in \mathbf{R}^n$ ,  $h \neq 0$ . Let  $\mathcal{A} = \sum_i^n h_i \mathbf{H}_i(\omega)$ . By (7.19) and (7.20), we have

$$\begin{aligned} \nabla^2 B_1(\lambda)[h, h] + 2\nabla B_1(\lambda)[h] + B_1(\lambda) &= \sum_{i,j} h_i h_j \partial_{ij}^2 B_1(\lambda) + 2 \sum_i h_i \partial_i B_1(\lambda) + B_1(\lambda) \\ &= \frac{1}{\pi} \int_{-\infty}^{\infty} \text{tr} \left( 2\mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} - 2\mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} + \mathbf{H}^{-1} \right) \frac{d\omega}{1 + \omega^2} \\ &= \frac{1}{\pi} \int_{-\infty}^{\infty} 2 \|\mathbf{H}^{-\frac{1}{2}} \mathcal{A} \mathbf{H}^{-1} - 0.5 \mathbf{H}^{-\frac{1}{2}}\|_F^2 + 0.5 \|\mathbf{H}^{-\frac{1}{2}}\|_F^2 \frac{d\omega}{1 + \omega^2}. \end{aligned}$$

Hence, for any  $\lambda \in \Omega$ ,  $\nabla^2 B_1(\lambda)[h, h] + 2\nabla B_1(\lambda)[h] + B_1(\lambda) > 0$  for all  $h \neq 0$ . This in turn implies that  $\nabla^2 B_2(\lambda) > 0$  for all  $\lambda \in \Omega$ , and thus  $B_2(\lambda)$  is a convex function.  $\square$

Since  $B_2(\lambda) = \log(B_1(\lambda))$ , evaluation of barrier function  $B_2(\lambda)$  is performed by a computational procedure the same as the one for evaluating  $B_1(\lambda)$ , except the logarithm of the value of  $B_1(\lambda)$  is taken in the end. Therefore, the complexity of evaluating  $B_2(\lambda)$  is almost the same as the complexity of evaluating  $B_1(\lambda)$ . As to the gradient and the Hessian of



$B_2(\lambda)$ , it can be readily verified that

$$\begin{aligned}\nabla B_2(\lambda) &= B_1^{-1}(\lambda)\nabla B_1(\lambda), \\ \nabla^2 B_2(\lambda) &= B_1^{-1}(\lambda)\nabla^2 B_1(\lambda) - B_1^{-2}(\lambda)\nabla B_1(\lambda)\nabla B_1(\lambda)'.\end{aligned}$$

Therefore, as soon as the value, the gradient, and the Hessian of  $B_1(\lambda)$  are available, the gradient and the Hessian of  $B_2(\lambda)$  can be easily computed with little extra computational effort. Hence, the computational complexities of computing the gradient and the Hessian of  $B_2(\lambda)$  are only slightly higher than those of  $B_1(\lambda)$ .

What we gain from the extra computational effort is that now the barrier function  $B_2(\lambda)$  satisfies property (P3), one of the requirements for  $B_2(\lambda)$  to be a self-concordant barrier.

**Proposition 7.4.** *The barrier function  $B_2(\lambda)$  satisfies*

$$|\nabla B_2(\lambda)[h]| \leq \nu(\nabla^2 B_2(\lambda)[h, h])^{\frac{1}{2}}, \quad \forall \lambda \in \Lambda, \quad \forall h \in \mathbf{R}^n, \quad (7.46)$$

where  $\nu$  is greater than or equal to 1.

*Proof.* Substituting the expressions of the gradient and the Hessian of  $B_2(\lambda)$  into the inequality in (7.46) and noticing that  $B_1(\lambda) > 0 \forall \lambda \in \Omega$ , we see (7.46) is equivalent to

$$(\nu^2 + 1)B_1(\lambda)^{-1}(\nabla B_1(\lambda)[h])^2 \leq \nu^2 \nabla^2 B_1(\lambda)[h, h], \quad \forall \lambda \in \Omega, \quad \forall h \in \mathbf{R}^n \quad (7.47)$$

Condition (7.47) is satisfied if the matrix

$$\begin{bmatrix} \nabla^2 B_1(\lambda) & \nabla B_1(\lambda) \\ \nabla B_1(\lambda)' & \frac{\nu^2}{\nu^2+1} B_1(\lambda) \end{bmatrix}$$

is positive definite for all  $\lambda \in \Omega$ . Therefore, what we have to show is that  $\nabla^2 B_1(\lambda)[h, h] + 2\nabla B_1(\lambda)[h] + \frac{\nu^2}{\nu^2+1} B_1(\lambda) \geq 0$  for all  $\lambda \in \Omega$  and for all  $h \in \mathbf{R}^n$ . We have

$$\begin{aligned}& \nabla^2 B_1(\lambda)[h, h] + 2\nabla B_1(\lambda)[h] + \frac{\nu^2}{\nu^2+1} B_1(\lambda) \\ &= \frac{1}{\pi} \int_{-\infty}^{\infty} \text{tr} \left( 2\mathbf{H}^{-1}\mathcal{A}\mathbf{H}^{-1}\mathcal{A}\mathbf{H}^{-1} - 2\mathbf{H}^{-1}\mathcal{A}\mathbf{H}^{-1} + \frac{\nu^2}{\nu^2+1}\mathbf{H}^{-1} \right) \frac{d\omega}{1+\omega^2} \\ &= \frac{1}{\pi} \int_{-\infty}^{\infty} 2\|\mathbf{H}^{-\frac{1}{2}}\mathcal{A}\mathbf{H}^{-1} - 0.5\mathbf{H}^{-\frac{1}{2}}\|_F^2 + \left( \frac{\nu^2}{\nu^2+1} - \frac{1}{2} \right) \|\mathbf{H}^{-\frac{1}{2}}\|_F^2 \frac{d\omega}{1+\omega^2},\end{aligned}$$

where  $\mathcal{A} = \sum_{i=1}^n h_i \mathbf{H}_i(\omega)$ . Note that  $\frac{\nu^2}{\nu^2+1}$  is monotonically increasing on  $[0, \infty)$ , and  $\frac{\nu^2}{\nu^2+1} \geq \frac{1}{2}$  if  $\nu^2 \geq 1$ . Hence, as long as  $\nu \geq 1$ , we have  $\nabla^2 B_1(\lambda)[h, h] + 2\nabla B_1(\lambda)[h] + \frac{\nu^2}{\nu^2+1} B_1(\lambda) \geq 0$  for all  $\lambda \in \Omega$  and for all  $h \in \mathbf{R}^n$ . This concludes the proof.  $\square$

Furthermore, we can also prove that property (P2) also holds for  $B_2(\lambda)$ .

**Theorem 7.5.** *There exists a constant  $c$  such that*

$$|\nabla^3 B_2(\lambda)[h, h, h]| \leq c(\nabla^2 B_2(\lambda)[h, h])^{\frac{3}{2}} \quad \forall h \in \mathbf{R}^n \quad (7.48)$$

holds for all  $\lambda \in \Omega$ .

*Proof.* The proof is included in Section 7.8.  $\square$

Thus,  $B_2(\lambda)$  satisfies all three properties (P1) to (P3) and is a strongly self-concordant barrier. However, although we are able to prove that (7.48) holds, the value of the constant  $c$  is not provided from our proof and yet to be determined. Furthermore, in order for the interior path-following algorithm based on  $B_2(\lambda)$  to be a polynomial time algorithm, the constant  $c$  can only depend on the size of a given problem but not the data of the problem. Judging from some numerical experiments, this property does not seem to hold on  $B_2(\lambda)$ . See the following example.

**Example 7.4.** Let  $\mathbf{H}(\omega, \lambda)$  be  $\frac{a}{\lambda} + \frac{1}{1-\lambda}$ ,  $a > 0$ . Note that this  $\mathbf{H}(\omega, \lambda)$  corresponds to the set  $\{\lambda \mid 0 < \lambda < 1\}$  regardless the value of  $a$ . Then

$$B_2(\lambda) = \log B_1(\lambda) = \log \left( \frac{a}{\lambda} + \frac{1}{1-\lambda} \right)$$

The second and the third derivatives of  $B_2(\lambda)$  can be easily computed. Let

$$\gamma := \frac{\ddot{B}_2(\lambda)^2}{\ddot{B}_2(\lambda)^3}. \quad (7.49)$$

Numerical computations indicate that the maximum of  $\gamma$  over  $(0, 1)$  varies with  $a$ : the maximum is equal to 4 when  $a = 1$ , is between 38 and 39 when  $a = 100$ , and is between 7634 and 7635 when  $a = 100000$ . We expect the maximum approaches infinity as  $a$  does.

Hence, we are not able to apply Nesterov and Nemirovsky's results to construct a polynomial time interior path-following algorithm based on  $B_2(\lambda)$ . Nevertheless, the path-following

algorithm based on  $B_2(\lambda)$  converges globally according to Theorem 7.1. Furthermore, for a fixed  $t$ , Theorem 7.4 shows that the center corresponds to

$$\min_{\lambda} tc'\lambda + B_2(\lambda)$$

is guaranteed to be  $\frac{1}{t}$ -accurate; i.e., if  $\lambda^*(t)$  is the minimizer of the above minimization problem, then the difference between  $c'\lambda^*(t)$  and the optimal objective of the original problem is less than  $1/t$ .

## 7.5 Path-following Algorithms for IQC Optimization Problems

Barrier functions  $B_1(\lambda)$  and  $B_2(\lambda)$  are used to construct interior path-following algorithms for solving IQC optimization problems. The algorithms follows the basic principles of the interior path-following method described in Section 7.3. The algorithm based on barrier function  $B_1(\lambda)$  can be summarized as follows.

**Given:**  $\lambda_0 \in \Omega$ .

**Initialization:** select  $t := t_0 > 0$ ,  $\mu > 1$ , and  $\epsilon > 0$ . Set  $\lambda_U := \lambda_0$  and  $U := c'\lambda_U$ .

**Repeat**

(1) Centering : find an approximate solution to the problem

$$\min_{\lambda} \varphi_t(\lambda) := tc'\lambda + B_1(\lambda)$$

using the Newton's method:

Start at  $\lambda_0$ . Set  $n = 0$ .

**Repeat**

- (a) Compute the Newton descent direction  $\delta\lambda_n = -(\nabla^2\varphi_t(\lambda_n))^{-1}\nabla\varphi_t(\lambda_n)$
- (b) Compute Newton decrement at  $\lambda_n$  :  $\rho = (\nabla\varphi_t(\lambda_n))'(\nabla^2\varphi_t(\lambda_n))^{-1}\nabla\varphi_t(\lambda_n))^{\frac{1}{2}}$ .
- (c) Line minimization: compute  $\alpha^* = \operatorname{argmin} \varphi_t(\lambda_n + \alpha \cdot \delta\lambda_n)$ .
- (d) Set  $\lambda_{n+1} := \lambda_n + \alpha^*\delta\lambda_n$  and  $n := n + 1$ .
- (e) If  $c'\lambda_{n+1} < U$ , then update  $\lambda_U := \lambda_{n+1}$  and  $U := c'\lambda_U$ .

(f) If  $\rho < \epsilon$ , then stop the loop and return  $\lambda_n$  and  $\lambda_U$ .

**End.**

(2) Update  $\lambda_0$  : set  $\lambda_0 := \lambda_n$ .

(3) Update  $t$  : set  $t := \mu t$ .

**Until** (the stopping criterion is satisfied).

The initial feasible point  $\lambda_0$  is found (or determined not to exist) using the Phase-I method described in Section 7.3.3. The algorithm for Phase-I is the same as the algorithm described above.

Since  $B_1(\lambda)$  is not a self-concordant barrier, the rigorous stopping criterion (in the sense that on exit, an  $\epsilon$ -accurate suboptimal solution is guaranteed) by Nesterov and Nemirovsky does not apply here. In the algorithm, we adopt a heuristic criterion

$$\text{Until } \left( \frac{1}{t} B_1(\lambda_U) < \epsilon \right) \quad (7.50)$$

The idea of this criterion is straightforward: if  $\frac{1}{t} B_1(\lambda_U)$  is very small, then the penalty term has very little influence, and one can expect that the objective at the solution  $\lambda_U$  should be very close to the optimal objective of the original problem.

The interior path-following algorithm based on barrier function  $B_2$  is the same as the algorithm described above, except the stopping criterion. Instead of criterion (7.50), we use

$$\text{Until } \left( \frac{1}{t} < \epsilon \right) \quad (7.51)$$

as the stopping criterion. We have already shown that barrier function  $B_2(\lambda)$  satisfies

$$|\nabla B_2(\lambda)[h]| \leq (\nabla^2 B_2(\lambda)[h, h])^{\frac{1}{2}}, \quad \forall \lambda \in \Omega, \quad \forall h \in \mathbf{R}^n.$$

Therefore, according to Theorem 7.4, the minimizer of

$$\min_{\lambda} tc'\lambda + B_2(\lambda)$$

is guaranteed to be  $\frac{1}{t}$ -accurate. Hence, if  $\frac{1}{t} < \epsilon$  and the approximation of the center is sufficiently close to the center, then one can expect that the approximation of the center

is  $\epsilon$ -accurate. If the barrier function is self-concordant, then there is a rigorous mathematical criterion to determine whether an approximate center is sufficiently close or not. The criterion, however, requires the value of the self-concordant parameter (i.e., the constant  $c$  in (7.48)). In our case, since we do not know the value of the self-concordant parameter associated with  $B_2(\lambda)$ , we are not able to apply the criterion. In our algorithm, an approximate center is considered to be close enough to the center if the Newton decrement at the approximate center is less than a pre-selected small number.

## 7.6 Comparison with the Conventional Method

In this section, we compare computational complexities of the interior path-following algorithms proposed in this chapter with the complexity of the conventional method of solving IQC problems.

The conventional approach to solve the IQC optimization problem (7.1) is to transform it into the equivalent SDP

$$\begin{aligned} & \inf_{P=P', \lambda} c' \lambda, \text{ subj. to} \\ & \begin{bmatrix} PA + A'P & PB \\ B'P & 0 \end{bmatrix} + \begin{bmatrix} Q(\lambda) & F(\lambda) \\ F(\lambda)' & R(\lambda) \end{bmatrix} > 0. \end{aligned} \quad (7.52)$$

Problem (7.52) is then solved using interior point method. Let the number of decision variables in  $\lambda$  be  $n$ . Let the dimension of matrix  $A$  be  $n_x \times n_x$  and the dimension of  $R(\lambda)$  be  $n_r \times n_r$ . Problem (7.52) has a well-known strongly  $(n_x + n_r)$ -self-concordant barrier, and the interior point algorithm for solving (7.52) can be proven to converge in  $\mathbf{O}(\sqrt{n_x + n_r} \log \frac{n_x + n_r}{\epsilon})$  Newton steps [75, 60]. Furthermore, the computational complexity of each Newton step is counted as  $\mathbf{O}(n_x^2(n_x^2 + n)^2)$ . Therefore, the total complexity is  $\mathbf{O}(\sqrt{n_x + n_r} \log \frac{n_x + n_r}{\epsilon}) \cdot \mathbf{O}(n_x^2(n_x^2 + n)^2)$ .

In the case of the interior path-following algorithms proposed in this chapter, the number of Newton steps required for the algorithms to converge is yet to be determined. However, the complexity of each Newton step of the algorithms can be estimated as follows: in each Newton step, the algorithm computes a descent direction and performs a line search to find a new point. This procedure involves one computation of the gradient, one computation of the Hessian, and  $\mathbf{O}(1)$  evaluations of the barrier function. To evaluate the barrier function,

a Riccati equation in the form of (7.24) and a Lyapunov equation in the form of (7.25) have to be solve. The computation complexities of solving (7.24) and (7.25) are both  $\mathbf{O}(n_x^3)$ . To compute each entry of the gradient and the Hessian of the barrier, two Lyapunov equations in the form of (7.25) have to be solved. Complexity of solving each of the Lyapunov function again is  $\mathbf{O}(n_x^3)$ . There are  $n$  entries in the gradient vector and  $\frac{n^2+n}{2}$  entries in the Hessian matrix. Therefore, the estimated complexity of each Newton step is  $\mathbf{O}(n^2n_x^3)$ .

Suppose that  $n_x = \mathbf{O}(n)$ . Then each Newton step of the interior point algorithm for solving SDP (7.52) requires  $\mathbf{O}(n_x^6)$  arithmetic operations, while each Newton step of interior path-following algorithms proposed in this chapter requires only  $\mathbf{O}(n^2n_x^3)$ . This is the why the algorithms proposed in this chapter are more efficient. Furthermore, we expect that when the ratio  $n_x/n$  is large enough, the algorithms proposed in this chapter will perform significantly better than the conventional method. The total complexity of an algorithm is the complexity of an iteration times the number of iterations the algorithm requires to converge. Therefore, the argument here is based on the assumption that the number of iterations which the path-following algorithms presented in this chapter require to solve an IQC problem is roughly the same as the number of iterations which the interior point algorithm requires to solve the equivalent SDP.

In the next section, we present the results of numerical tests on the interior path-following algorithms proposed in this chapter. As we will see, these results agree with the expectation from the complexity analysis.

## 7.7 Examples

**Example 7.5.** Consider the  $L_2$ -gain estimation problem in Example 6.1. Again, let  $n = 10$  and  $n_s = 10, \dots, 70$ . For each pair of  $(n, n_s)$ , the five randomly generated problems mentioned in Chapter 6 are solved using the interior path-following algorithms proposed in this chapter. Figures 7-1 and 7-2 show the amount of time the two algorithms spent on solving these problems.

First of all, we see from Figure 7-1 and 7-2 that the two interior path-following algorithms have similar performance. The amount of time they spent on solving a problem is at the same level. This is expected since, for a single iteration, the two algorithms have similar computational complexities.

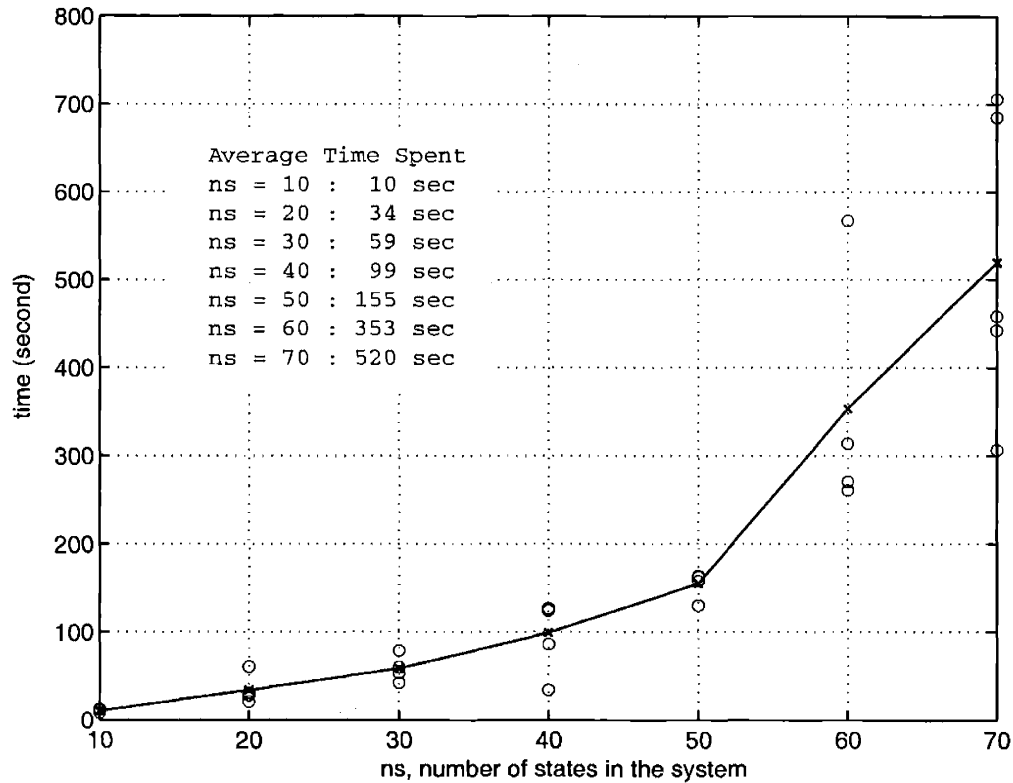


Figure 7-1: This figure shows the amount of time that the interior path-following algorithm based on the first barrier function took to solve problem (6.9). For each  $n_s$ , five test problems were randomly generated. As we can see from the figure, all problems were solved in less than 15 minutes. Compared with the Figure 7-3, we see that the proposed algorithm is significantly faster than the conventional method when the problem to be solved has a large state space. Furthermore, the proposed algorithm is no worse than the conventional method even when the problem has only a few states.

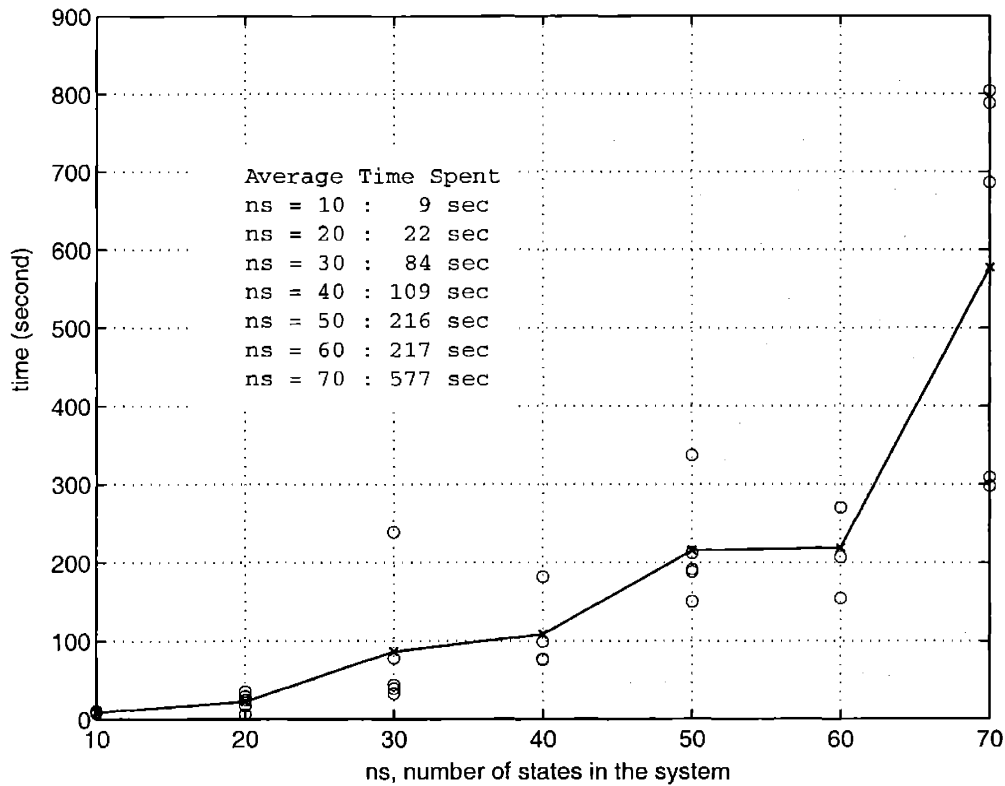


Figure 7-2: This figure shows the amount of time that the interior path-following algorithm based on the second barrier function took to solve problem (7-1). We see the performance of the interior path-following algorithm based on the second barrier function is very similar to the performance of the algorithm based on the first barrier. The amount of time they spent to solve a problem is at the same level.



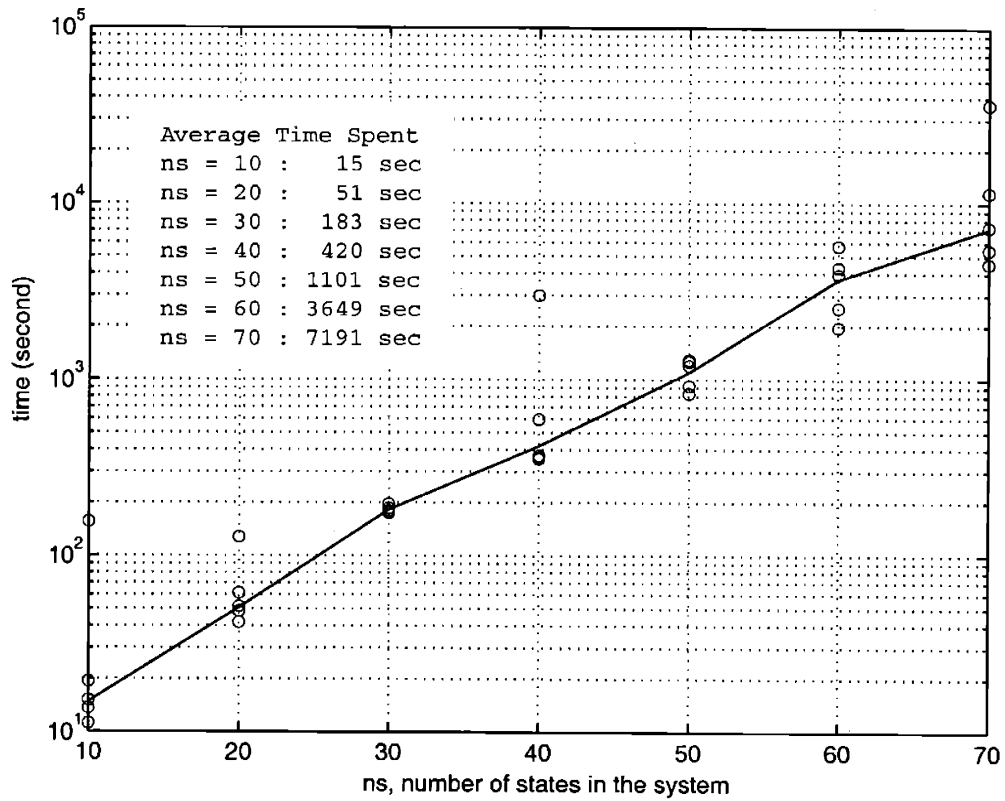


Figure 7-3: This figure (same as Figure 6-2) shows the amount of time that the MATLAB LMI Control Toolbox took to solve problem (6.10).

Method	iteration	iteration	iteration	iteration	iteration	iteration	iteration
LMI Tool	24	23	28	25	28	37	33
IPA-1	31	30	31	43	28	33	36
IPA-2	27	34	42	40	33	24	36

Table 7.1: Numbers of iterations that the MATLAB LMI Control Toolbox (denoted by “LMI Tool”), the first interior path-following algorithm (denoted by “IPA-1”), and the second interior path-following algorithm (denoted by “IPA-2”) took to solve problem (6.9).

Compared to the Figure 7-3 (which is the Figure 6-2 in Chapter 6), the performance of the two algorithms proposed in this Chapter is similar to the performance of the MATLAB LMI Control Toolbox when the problem they solve has few to moderate number of states ( $n_s \leq 20$ ). When the number of states in a problem becomes larger ( $30 \leq n_s \leq 50$ ), the performance of the two proposed algorithms becomes much better than the performance of the MATLAB LMI Control Toolbox. For a problem of 30 states, the amount of time which the two proposed algorithms spent is about 1/3 of the time which the MATLAB LMI Control Toolbox took. For a problem of 50 states, the ratio reduces to 1/5 to 1/6. As the number of states becomes larger, the difference in speed becomes more significant. When  $n_s = 70$ , the two proposed algorithms are more than 10 times faster than the MATLAB LMI Control Toolbox. This fits our expectation from the complexity analysis.

Table 7.1 shows the average number of iterations which the two path-following algorithms proposed in this chapter and the MATLAB LMI Control Toolbox took to solve a set of problems. As we can see, the number of Newton’s iterations which the proposed interior path-following algorithms took to solve these problems is similar to those the MATLAB LMI Control Toolbox took. Therefore, although we have not been able to determine whether the proposed path-following algorithms are polynomial-time algorithms or not, this example shows that at least practically the proposed algorithms can work as good as the interior point methods for SDPs.

**Example 7.6.** Consider the *Seismic Isolation Control System* (Example 6.2) described in Chapter 6. Again, we are interested in analyzing robustness properties of the seismic control systems. We assume that real values of the stiffness coefficients  $k_1, \dots, k_3$  and the damping coefficients  $c_1, \dots, c_3$  are different from the nominal values used for controller designs, but within  $\pm 5\%$  of those shown in (6.25) and (6.26); i.e., the real stiffness and damping coefficients for floors 1 to 3, denoted by  $\bar{k}_r$  and  $\bar{c}_r$ , are represented as

$$\bar{k}_r = (1 + \delta_{kr})k_r, \quad \bar{c}_r = (1 + \delta_{cr})c_r, \quad r = 1, \dots, 3,$$

where  $\delta_{kr}$  and  $\delta_{cr}$  are unknown constants whose absolute values are less than or equal to 0.05. We are interested in checking whether the seismic control systems are still stable under the change of these parameters and estimating the  $L_2$ -gain from the earthquake input  $v$  to the acceleration vector  $\ddot{x}$ . To do this, we represent the closed-loop seismic control systems in the standard form for robustness analysis as shown in Figure 6-10, where  $G$  denotes the nominal closed-loop seismic control system, and output  $z$  is a  $6 \times 1$  vector signal defined as

$$z' = \left[ \frac{k_1 x_1}{20} \quad \frac{k_2(x_2 - x_1)}{20} \quad \frac{k_3(x_3 - x_2)}{20} \quad \frac{c_1 \dot{x}_1}{20} \quad \frac{c_2(\dot{x}_2 - \dot{x}_1)}{20} \quad \frac{c_3(\dot{x}_3 - \dot{x}_2)}{20} \right].$$

Matrix  $\Delta := \text{diag}(\delta_1, \dots, \delta_6)$  represents the uncertainties in the system. Each  $\delta_i$  denotes an unknown constant with absolute value less than or equal to 1.

To estimate the  $L_2$ -gain from  $v$  to  $\ddot{x}$ , we apply standard IQC analysis to the system in Figure 6-10. The IQCs used to characterize uncertain constants  $\delta_i$ ,  $i = 1, \dots, 6$ , are similar to the IQCs described in Example 6.2. The problems were solved using the MATLAB LMI Control Toolbox and the interior path-following algorithms proposed in this chapter. The results are listed in Table 7.2.

Table 7.2 indicates that the two interior path-following algorithms solved every problem faster than the conventional method using MATLAB LMI Control Toolbox. When  $n = 5$  (that is, the building has 5 floors), the equivalent SDP has 655 decision variables and it took the LMI Control Toolbox about 12 minutes to solve the problem. The two interior path-following algorithms took only about 60% of the time to solve the same problem. As  $n$  increases from 5 to 11, the amount of time which the LMI Control Toolbox took to solve a problem increases rapidly. When  $n = 11$ , the two interior path-following algorithms are about 5 to 7 times faster than the MATLAB LMI Control Toolbox.

	$n = 5$			$n = 6$			$n = 7$		
Method	$L_2$ -gain	time (sec)	var.	$L_2$ -gain	time (sec)	var.	$L_2$ -gain	time (sec)	var.
LMI Tool	1.16309	723	655	1.11712	1169	805	0.78299	1483	971
IPA-1	1.16228	382	19	1.11547	806	19	0.78425	335	19
IPA-2	1.16242	431	19	1.11534	498	19	0.78334	467	19
	$n = 8$			$n = 9$			$n = 10$		
Method	$L_2$ -gain	time (sec)	var.	$L_2$ -gain	time (sec)	var.	$L_2$ -gain	time (sec)	var.
LMI Tool	0.82624	2424	1153	0.82851	3153	1351	0.95618	4285	1565
IPA-1	0.82611	481	19	0.82806	788	19	0.95581	597	19
IPA-2	0.82622	564	19	0.82816	788	19	0.95626	847	19
	$n = 11$								
Method	$L_2$ -gain	time (sec)	var.						
LMI Tool	0.93548	7141	1795						
IPA-1	0.93559	1017	19						
IPA-2	0.93558	1437	19						

Table 7.2: Results of solving the  $L_2$ -gain estimation problem in Example 6.2 using MATLAB LMI Control Toolbox, the path-following algorithms. “LMI Tool“ denotes MATLAB LMI Control Toolbox. “IPA-1“ denotes the interior path-following algorithm based on the first barrier function. “IPA-2“ denotes the interior path-following algorithm based on the second barrier function. The numbers in the column “var“ indicate the number of decision variables in a problem.

## 7.8 Proof of Theorem 7.5

We prove Theorem 7.5 in this section. The following two lemmas will be used in the proof of Theorem 7.5.

**Lemma 7.3.** *Let  $\omega_0$  be a real number and  $\beta \geq 2$  be a positive even integer. Let*

$$G(t, \omega) = \frac{r_0(t) + r_1(t)(\omega - \omega_0)^\beta + \cdots + r_{n-1}(t)(\omega - \omega_0)^{\beta(n-1)}}{(p(t) + (\omega - \omega_0)^\beta q(t))^n}, \quad (7.53)$$

where  $p(t)$ ,  $q(t)$ ,  $r_i(t)$ ,  $i = 0, \dots, n-1$  are polynomial functions in  $t$ . Now consider the integration

$$F(t) = \int_{\omega_0 - \epsilon}^{\omega_0 + \epsilon} G(t, \omega) d\omega. \quad (7.54)$$

Then, for any  $t^* > 0$ , there exists a sufficiently small  $\epsilon$  such that

$$F(t) \approx c(t) \cdot r_0(t) p(t)^{\frac{1}{\beta} - n} q(t)^{-\frac{1}{\beta}} \quad (7.55)$$

for all  $|t| \leq t^*$ , where  $c(t)$  is a bounded function in  $t$ .

*Proof.* Fix  $t$ . Then for sufficiently small  $\epsilon$ , the term  $r_0(t)$  dominates the numerator of  $G(t, \omega)$ . Therefore,

$$F(t) \approx \int_{\omega_0 - \epsilon}^{\omega_0 + \epsilon} \frac{r_0(t)}{(p(t) + (\omega - \omega_0)^\beta q(t))^n} d\omega.$$

Let

$$\omega = \omega_0 + \left(\frac{p(t)}{q(t)}\right)^{\frac{1}{\beta}} \tilde{\omega}.$$

We have

$$\begin{aligned} \int_{\omega_0 - \epsilon}^{\omega_0 + \epsilon} \frac{r_0(t)}{(p(t) + (\omega - \omega_0)^\beta q(t))^n} d\omega &= \int_{-\eta}^{\eta} \frac{r_0(t) p(t)^{\frac{1}{\beta}} q(t)^{-\frac{1}{\beta}}}{p(t)^n (1 + \tilde{\omega}^\beta)^n} d\tilde{\omega} \\ &= r_0(t) p(t)^{\frac{1}{\beta} - n} q(t)^{-\frac{1}{\beta}} \int_{-\eta}^{\eta} \frac{1}{(1 + \tilde{\omega}^\beta)^n} d\tilde{\omega}, \end{aligned}$$

where  $\eta = \epsilon \left(\frac{p(t)}{q(t)}\right)^{-\frac{1}{\beta}}$ . Finally, let

$$c(t) := \int_{-\eta}^{\eta} \frac{1}{(1 + \tilde{\omega}^\beta)^n} d\tilde{\omega}.$$

Since

$$\int_{-\eta}^{\eta} \frac{1}{(1 + \tilde{\omega}^\beta)^n} d\tilde{\omega} \leq \int_{-\infty}^{\infty} \frac{1}{(1 + \tilde{\omega}^\beta)^n} d\tilde{\omega} = \text{const},$$

therefore, we conclude that  $F(t) \approx c(t) r_0(t) p(t)^{\frac{1}{\beta} - n} q(t)^{-\frac{1}{\beta}}$ , where  $c(t)$  is bounded.  $\square$

**Lemma 7.4.** *Let  $\alpha \geq 1$  be an integer and  $\beta \geq 2$  be an even integer. Let*

$$G(t, \omega) = \frac{r(t)}{t^\alpha p(t) + (\omega - \omega_0)^\beta q(t)}, \quad F(t) = \int_{\omega_0 - \epsilon}^{\omega_0 + \epsilon} G(t, \omega) d\omega,$$

where  $p(t), q(t), r(t)$  are polynomial functions in  $t$ , and  $p(0) \neq 0, q(0) \neq 0$ . Then

$$F(t) \approx t^{-n} s_0(t), \quad \dot{F}(t) \approx t^{-n-1} s_1(t), \tag{7.56}$$

$$\ddot{F}(t) \approx t^{-n-2} s_2(t), \quad \dddot{F}(t) \approx t^{-n-3} s_3(t), \tag{7.57}$$

where  $n = \alpha - \frac{\alpha}{\beta}$ , and  $s_i(t)$   $i = 0, \dots, 3$  are bounded at 0.

*Proof.* Notice that  $G(t)$  is in the form of (7.53). The corresponding  $r_0(t)$ ,  $p(t)$ ,  $q(t)$ , and  $n$  are  $r(t)$ ,  $t^\alpha p(t)$ ,  $q(t)$  and 1, respectively. Therefore, by Lemma 7.3, we have

$$F(t) \approx c(t)r(t)(t^\alpha p(t))^{\frac{1}{\beta}-1}q(t)^{-\frac{1}{\beta}} = t^{-\alpha+\frac{\alpha}{\beta}}c(t)r(t)p(t)^{\frac{1}{\beta}-1}q(t)^{-\frac{1}{\beta}}$$

which is exactly of the form in (7.56) with  $s_0(t) = c(t)r(t)p(t)^{\frac{1}{\beta}-1}q(t)^{-\frac{1}{\beta}}$ . Since  $p(0) \neq 0$ ,  $q(0) \neq 0$ , therefore  $s_0(t)$  is bounded at 0.

Now, let us consider the differentiations of  $F(t)$ . We have

$$\dot{F}(t) = \int_{\omega_0-\epsilon}^{\omega_0+\epsilon} \dot{G}(t, \omega) d\omega, \quad \ddot{F}(t) = \int_{\omega_0-\epsilon}^{\omega_0+\epsilon} \ddot{G}(t, \omega) d\omega, \quad \dddot{F}(t) = \int_{\omega_0-\epsilon}^{\omega_0+\epsilon} \dddot{G}(t, \omega) d\omega,$$

where

$$\begin{aligned} \dot{G}(t, \omega) &= \frac{d_{10}(t) + d_{11}(t)(\omega - \omega_0)^\beta}{(t^\alpha p(t) + (\omega - \omega_0)^\beta q(t))^2}, \\ \ddot{G}(t, \omega) &= \frac{d_{20}(t) + d_{21}(t)(\omega - \omega_0)^\beta + d_{22}(t)(\omega - \omega_0)^{2\beta}}{(t^\alpha p(t) + (\omega - \omega_0)^\beta q(t))^3}, \\ \dddot{G}(t, \omega) &= \frac{d_{30}(t) + d_{31}(t)(\omega - \omega_0)^\beta + d_{32}(t)(\omega - \omega_0)^{2\beta} + d_{33}(t)(\omega - \omega_0)^{3\beta}}{(t^\alpha p(t) + (\omega - \omega_0)^\beta q(t))^4}, \end{aligned}$$

and

$$\begin{aligned} d_{10}(t) &= \dot{r}(t)t^\alpha p(t) - r(t)(\alpha t^{\alpha-1}p(t) + t^\alpha \dot{p}(t)) \\ &= -\alpha t^{\alpha-1}r(t)p(t) + \mathbf{O}(t^\alpha) \\ d_{20}(t) &= \dot{d}_{10}(t)t^\alpha p(t) - d_{10}(t)(\alpha t^{\alpha-1}p(t) + t^\alpha \dot{p}(t)) \\ &= \alpha(\alpha + 1)t^{2\alpha-2}r(t)p(t)^2 + \mathbf{O}(t^{2\alpha-1}) \\ d_{30}(t) &= \dot{d}_{20}(t)t^\alpha p(t) - d_{20}(t)(\alpha t^{\alpha-1}p(t) + t^\alpha \dot{p}(t)) \\ &= -\alpha(\alpha + 1)(\alpha + 2)t^{3\alpha-3}r(t)p(t)^3 + \mathbf{O}(t^{3\alpha-2}) \end{aligned}$$

We do not care for the detailed expressions of the rest of  $d_{ij}(t)$  terms. By Lemma 7.3,

$$\dot{F}(t) \approx c_1(t)d_{10}(t)t^{-2\alpha+\frac{\alpha}{\beta}}p(t)^{\frac{1}{\beta}-2}q(t)^{-\frac{1}{\beta}}. \quad (7.58)$$

Substituting the expression of  $d_{10}(t)$  into (7.58), we obtain

$$\begin{aligned}\dot{F}(t) &\approx t^{-2\alpha+\frac{\alpha}{\beta}}(-\alpha t^{\alpha-1}r(t)p(t) + \mathbf{O}(t^\alpha))c_1(t)r(t)p(t)^{\frac{1}{\beta}-2}q(t)^{-\frac{1}{\beta}} \\ &= t^{-n-1}(c_1(t)r(t)p(t)^{\frac{1}{\beta}-2}q(t)^{-\frac{1}{\beta}})(-\alpha r(t)p(t) + \mathbf{O}(t)) \\ &= t^{-n-1}s_1(t),\end{aligned}$$

where  $s_1(t)$  is bounded at 0 since  $p(0) \neq 0$  and  $q(0) \neq 0$ . Therefore, we conclude that  $\dot{F}(t)$  is in the form described in (7.56).  $\ddot{F}(t) \approx t^{-n-2}s_2(t)$  and  $\ddot{\ddot{F}}(t) \approx t^{-n-3}s_3(t)$  are obtained in a similar fashion.  $\square$

### Proof of Theorem 7.5

We are now ready to prove Theorem 7.5. Recall that

$$B_2(\lambda) = \log(B_1(\lambda)), \quad \text{and} \quad B_1(\lambda) = \left( \frac{1}{\pi} \int_{-\infty}^{\infty} \text{tr}(\mathbf{H}(\omega, \lambda)^{-1}) \frac{d\omega}{1 + \omega^2} \right).$$

Given any  $\lambda \in \Omega$  and any  $h \in \mathbf{R}^n$ , let  $T$  be the open interval  $\{t \mid \lambda + th \in \Omega\}$ . Now, define  $F(t) : T \rightarrow \mathbf{R} := B_1(\lambda + th)$  and  $E(t) : T \rightarrow \mathbf{R} := B_2(\lambda + th) = \log(F(t))$ . Let  $\gamma(t) := \frac{\ddot{E}(t)^2}{\dot{E}(t)^3}$ . To show  $B_2(\lambda)$  satisfies (7.48), we have to prove that

$$\sup_{t \in T} \gamma(t) < \infty, \tag{7.59}$$

i.e.,  $\gamma(t)$  is bounded above for all  $t \in T$ . Since  $\gamma(t)$  is a continuous function, therefore (7.59) is true if  $\gamma(t)$  is finite as  $t$  approaches any boundary point of  $T$ .

Note that

$$\frac{1}{\pi} \frac{1}{(1 + \omega^2)} \text{tr}(\mathbf{H}(\omega, \lambda + th)^{-1}) = \frac{r(\omega, t)}{s(\omega, t)},$$

where  $r(\omega, t)$  and  $s(\omega, t)$  are polynomials in  $\omega$  and  $t$ . Without loss of generality, let us assume that 0 is a boundary point of  $T$  and  $(1 + \omega^2)\mathbf{H}(\omega, \lambda + th)$  is singular at  $\omega = \omega_1, \dots, \omega_n$  at  $t = 0$ . Under these assumptions, we have

$$\frac{r(\omega, t)}{s(\omega, t)} \rightarrow G_k(t) := \frac{r_k(t)}{t^{\alpha_k} p_k(t) + (\omega - \omega_k)^{\beta_k} q_k(t)} \tag{7.60}$$

as  $\omega \rightarrow \omega_k$ , where  $\alpha_k$  is an integer greater than or equal to 1,  $\beta_k$  is an even integer greater than or equal to 2, and  $p_k(t)$ ,  $q_k(t)$ ,  $r_k(t)$  are polynomials in  $t$  such that  $p_k(0) \neq 0$ ,  $q_k(0) \neq 0$ . Let  $\epsilon$  be a small number and  $\Sigma = \bigcup_{k=1}^n [\omega_k - \epsilon, \omega_k + \epsilon]$ . We have

$$\begin{aligned} F(t) &= \int_{-\infty}^{\infty} \frac{1}{\pi} \operatorname{tr}(\mathbf{H}(\omega, \lambda + th)^{-1}) \frac{d\omega}{(1 + \omega^2)} \\ &= \sum_{k=1}^n \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} \frac{r(\omega, t)}{s(\omega, t)} d\omega + \int_{[-\infty, \infty] \setminus \Sigma} \frac{r(\omega, t)}{s(\omega, t)} d\omega. \end{aligned}$$

Since  $s(\omega, t)$  is bounded away from 0 for all  $t \in T$  and for all  $\omega \in [-\infty, \infty] \setminus \Sigma$ , therefore

$$\sup_{t \in T} \int_{[-\infty, \infty] \setminus \Sigma} \frac{1}{\pi} \operatorname{tr}(\mathbf{H}(\omega, \lambda + th)^{-1}) \frac{d\omega}{(1 + \omega^2)} \leq M, \quad (7.61)$$

where  $M$  is a constant depending on  $\epsilon$ . Thus, by (7.60) and (7.61), we see that for every  $t \in T$ ,

$$F(t) = \sum_{k=1}^n \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} G_k(\omega, t) d\omega + \mathbf{O}(1), \quad (7.62)$$

where  $\mathbf{O}(1)$  depends on  $\epsilon$  and  $t$ . Let us consider the  $\kappa^{\text{th}}$  derivative of  $F(t)$ . We have

$$\begin{aligned} \frac{d^\kappa}{dt^\kappa} F(t) &= \int_{-\infty}^{\infty} \frac{1}{\pi} \frac{d^\kappa}{dt^\kappa} \operatorname{tr}(\mathbf{H}(\omega, \lambda + th)^{-1}) \frac{d\omega}{(1 + \omega^2)} \\ &= \sum_{k=1}^n \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} \frac{d^\kappa}{dt^\kappa} \left( \frac{r(\omega, t)}{s(\omega, t)} \right) d\omega + \int_{[-\infty, \infty] \setminus \Sigma} \frac{d^\kappa}{dt^\kappa} \left( \frac{r(\omega, t)}{s(\omega, t)} \right) d\omega \\ &= \sum_{k=1}^n \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} \frac{r_\kappa(\omega, t)}{s(\omega, t)^\kappa} d\omega + \int_{[-\infty, \infty] \setminus \Sigma} \frac{r_\kappa(\omega, t)}{s(\omega, t)^\kappa} d\omega, \end{aligned}$$

where  $r_\kappa(\omega, t)$  is a polynomial in  $\omega$  and  $t$ . Since  $s(\omega, t)^\kappa$  is bounded away from 0 for all  $t \in T$  and for all  $\omega \in [-\infty, \infty] \setminus \Sigma$ , and

$$\frac{r_\kappa(\omega, t)}{s(\omega, t)^\kappa} \rightarrow \frac{d^\kappa}{dt^\kappa} G_k(\omega, t) \quad (7.63)$$

as  $\omega \rightarrow \omega_k$ , thus, we conclude that

$$\frac{d^\kappa}{dt^\kappa} F(t) = \sum_{k=1}^n \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} \frac{d^\kappa}{dt^\kappa} G_k(\omega, t) d\omega + \mathbf{O}(1). \quad (7.64)$$



Now, let

$$F_k(t) = \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} G_k(\omega, t) d\omega, \quad k = 1, \dots, n.$$

By virtue of Lemma 7.4, we conclude that

$$F_k(t) \approx t^{-m_k} s_{k0}(t), \quad (7.65)$$

$$\dot{F}_k(t) \approx t^{-m_k-1} s_{k1}(t), \quad (7.66)$$

$$\ddot{F}_k(t) \approx t^{-m_k-2} s_{k2}(t), \quad (7.67)$$

$$\dddot{F}_k(t) \approx t^{-m_k-3} s_{k3}(t), \quad (7.68)$$

where  $m_k = \alpha_k - \frac{\alpha_k}{\beta_k}$ , and  $s_{ki}(t)$ ,  $i = 0, \dots, 3$ , are bounded at 0. Without loss of generality, let us assume  $m_1 \geq m_2 \geq \dots \geq m_n$ . Then (7.62), and (7.64) to (7.68) imply that

$$F(t) \approx t^{-m_1} s_0(t), \quad (7.69)$$

$$\dot{F}(t) \approx t^{-m_1-1} s_1(t), \quad (7.70)$$

$$\ddot{F}(t) \approx t^{-m_1-2} s_2(t), \quad (7.71)$$

$$\dddot{F}(t) \approx t^{-m_1-3} s_3(t). \quad (7.72)$$

Again, in (7.69) to (7.72),  $s_i(t)$ ,  $i = 0, \dots, 3$ , are bounded at 0.

Now, consider  $E(t) = \log F(t)$ . It can be readily verified that

$$\dot{E}(t) = F(t)^{-1} \dot{F}(t),$$

$$\ddot{E}(t) = F(t)^{-1} \ddot{F}(t) - F(t)^{-2} \dot{F}(t)^2,$$

$$\dddot{E}(t) = F(t)^{-1} \dddot{F}(t) - 3F(t)^{-2} \dot{F}(t) \ddot{F}(t) + 2F(t)^{-3} \dot{F}(t)^3,$$

and

$$\gamma(t) := \frac{\ddot{E}(t)^2}{\dot{E}(t)^3} = \frac{(F(t)^2 \ddot{F}(t) - 3F(t) \dot{F}(t) \ddot{F}(t) + 2\dot{F}(t)^3)^2}{(F(t) \ddot{F}(t) - \dot{F}(t)^2)^3}. \quad (7.73)$$

Substituting (7.69) to (7.72) into (7.73), we obtain

$$\gamma(t) \approx \frac{t^{-6m_1-6}(s_0(t)^2 s_3(t) - 3s_0(t)s_1(t)s_2(t) + 2s_1(t)^3)^2}{t^{-6m_1-6}(s_0(t)s_2(t) - s_1(t)^2)^3}.$$

Therefore, as  $t \rightarrow 0$ , we have

$$\gamma(0) \approx \frac{(s_0(0)^2 s_3(0) - 3s_0(0)s_1(0)s_2(0) + 2s_1(0)^3)^2}{(s_0(0)s_2(0) - s_1(0)^2)^3},$$

which is a finite number. Thus, the supremum

$$\sup_{t \in T} \frac{\ddot{E}(t)^2}{\ddot{E}(t)^3}$$

is bounded. This in turn implies  $B_2(\lambda)$  satisfies (7.48).

## 7.9 Summary

In this chapter, we proposed two new barrier functions to construct interior path-following algorithms for solving standard IQC optimization problems. Conventionally, one transforms a standard IQC optimization problem into a SDP and then solves the SDP using interior point methods. The transformation requires an additional matrix variable which in some cases substantially increases the computational complexity. Thus, in these cases, the conventional approach to solve IQC optimization problems is very inefficient. The new barrier functions we propose do not involve any additional decision variable, and therefore, interior path-following algorithms proposed in this chapter can solve standard IQC optimization problems in a more efficient fashion.

It is known that one can construct a polynomial time interior path-following algorithm provided that the barrier function used in the algorithm is strongly self-concordant. The two barrier functions proposed in this chapter do not quite satisfy this property. One barrier function can be shown not to be a strongly self-concordant barrier. For the other barrier function, although we are able to prove that it is a self-concordant barrier, the value of self-concordant parameter is yet to be discovered. Furthermore, numerical experiments indicate that the value of the self-concordant parameter seems to depend not only on the dimension but also the data of the problem. Thus, we are not able to determine whether polynomial

time interior path-following algorithms can be constructed based on the proposed barrier functions.

Nevertheless, the interior path-following algorithms based on the proposed barrier functions can be shown to converge globally. We have tested the proposed algorithms on a number of numerical examples. The results indicate that the proposed algorithms work well and significantly outperform the conventional method of solving IQC optimization problems, especially when the number of states in an IQC problem is very large.



## Chapter 8

# Conclusions and Future Research

Motivated by the need for robustness analysis of periodic trajectories, the first part of this thesis considers the problem of robustness analysis of Linear Periodically Time-Varying (LPTV) systems. This part of thesis consists of Chapters 3 to Chapter 5.

In Chapter 3, two new techniques for robust stability and performance analysis of LPTV systems with structured uncertainties were proposed. In the first approach, the system was transformed into a setup suitable for the standard Integral Quadratic Constraint (IQC) method. Analysis is then performed on the transformed system based on standard IQC analysis. In the second approach, we extended standard IQC analysis to include the case where the nominal system is LPTV. The robustness conditions were formulated as checking feasibility of affinely parameterized operator inequalities.

In Chapter 4, we derived conditions for existence and stability of stationary periodic solutions to periodic systems subject to periodic excitations, as well as conditions for certain harmonic performance of such systems. A new class of integral quadratic constraints particularly suited for analysis of periodic signals was introduced. Checking conditions for existence of a unique solution, stability, and performance was again formulated as solving certain constrained feasibility problems with affinely parameterized operator inequalities.

In Chapter 5, a cutting plane algorithm was developed to solve the feasibility and optimization problems arising from Chapters 3 and 4. The algorithm follows the basic principles of Kelley's cutting plane algorithm. The most essential part of the algorithm is the oracle. One of Yakubovich's results was used to construct the oracle for the problems arising from Chapter 3. A new "frequency theorem" by Jönsson and Megretski was used to

derive the oracle for the problems arising from Chapter 4. Several examples were given to illustrate the analysis techniques proposed in Chapters 3 and 4.

Motivated by the need for faster solvers for standard IQC problems, a number of computational algorithms that are potentially more efficient than the conventional approach for solving standard IQC problems were developed in the second part of the thesis. This part of thesis consists of Chapters 6 and 7.

In Chapter 6, three cutting plane algorithms - the Kelley type cutting plane algorithm, the ellipsoid algorithm, and the analytical center cutting plane method - were implemented to solve standard IQC problems. We studied their performance by testing them on a number of numerical examples and comparing their performance to that of the conventional approach of solving standard IQC problems. The results of numerical experiments indicate that the cutting plane algorithms are potentially very efficient as they avoid introducing additional decision variables, which are required in the conventional approach. In many numerical experiments, the cutting plane algorithms solved a problem more than 10 times faster than the conventional approach did. This difference in time is very significant, and grows bigger as the dimension of the system matrices becomes larger.

In Chapter 7, we proposed two new barrier functions to construct interior path-following algorithms for solving standard IQC optimization problems. Conventionally, one solves a standard IQC optimization problem by transforming the problem into a SDP and then solving the SDP using interior point methods. The transformation requires an additional matrix variable which in some cases substantially increases computational complexity. Thus, in these cases, the conventional approach for solving IQC optimization problems is very inefficient. The new barrier functions we proposed do not involve any additional decision variable. Therefore, interior path-following algorithms based on these barriers can solve standard IQC optimization problems in a more efficient fashion.

## 8.1 Suggested Future Research

The research presented in this thesis raises a few interesting questions. We list some of them in the following as topics for future research:

1. Systematic approaches for robustness analysis of linear periodically time-varying systems are proposed in the first part of the thesis. We demonstrated these approaches by

a number of examples, which are simple and more or less “academic“ (as opposed to “practical“). More case studies related to practical engineering problem are required in order to investigate how well these approaches can deal with real world problems.

2. The analytical center cutting plane method described in Chapter 6 has a worst-case  $\mathbf{O}^*(\frac{n^2}{\epsilon^2})$  bound on the number of iterations required for convergence. The results of our numerical experiments suggest that the  $\mathbf{O}^*(\frac{n^2}{\epsilon^2})$  bound is very conservative and that the algorithm usually terminates long before the bound is reached. The number of iterations required seemed to be practically close to  $\mathbf{O}(n|\log \epsilon|)$ . Our attempt to prove that  $\mathbf{O}(n|\log \epsilon|)$  is indeed the number of iterations for analytical center cutting plane algorithm to converge was not successful, nor can we construct an example to show that the worst-case  $\mathbf{O}^*(\frac{n^2}{\epsilon^2})$  bound can be reached. Whether the worst-case complexity of ACCPM counted in the number of iterations is  $\mathbf{O}(n|\log \epsilon|)$  is yet to be determined. Further investigation is required to answer this question.
3. We were unable to determine whether the interior path-following algorithms proposed in Chapter 7 are polynomial time algorithms or not, and further research is required to answer this question. In Chapter 7, one of the barrier function is shown *not* to be self-concordant. Therefore, the technique used in Nesterov and Nemirovsky’s proof cannot be used to prove that the interior path-following algorithm based on that barrier function converges in polynomial time, and one has to develop a new approach for proving polynomiality. The other barrier function is proven to be self-concordant. However, the value of the self-concordant parameter (or at least a bound on it), which is required to apply Nesterov and Nemirovsky’s result, is yet to be determined. Furthermore, the value (or the bound) can depend on the size but not on the data of a problem. Otherwise, the result by Nesterov and Nemirovsky still does not apply, and one has to look for a new proof to show polynomiality.
4. For now, the specialized fast computational algorithms presented in this thesis are programmed in MATLAB. Our implementation is crude and has a lot of room for improvement. For instance, these algorithms can be implemented in C/C++. For certain computational procedures, C programs are known to perform better (in terms of speed) than MATLAB programs. In addition, more numerical tests are required to further investigate the efficiency of these algorithms.





# Bibliography

- [1] M. Akgül. *Topics in Relaxation and Ellipsoidal Methods*. Volume 97 of Research Notes in Mathematics. Pitman, 1984.
- [2] D. S. Atkinson and P. M. Vaidya. A cutting plane algorithm for convex programming that uses analytic centers. *Mathematical Programming*, 69(B):1–43, 1995.
- [3] O. Bahn, O. du Merle, J.-L. Goffin, and J.-P. Vial. A cutting plane method from analytical centers for stochastic programming. *Mathematical Programming, Series B*, 69(1):45–73, 1995.
- [4] G.J. Balas, J.C. Doyle, K. Glover, A. Packard, and R. Smith.  *$\mu$ -Analysis and Synthesis Toolbox*. The Math Works Inc, 1993.
- [5] B.A. Bamieh and J.B. Pearson. A general framework for linear periodic systems with applications to  $\mathcal{H}^\infty$  sampled-data control. *IEEE Transactions on Automatic Control*, 37(4):418–435, April 1992.
- [6] B.A. Bamieh, J.B. Pearson, B. Francis, and A. Tannenbaum. A lifting technique for linear periodic systems with applications to sampled-data control. *Systems and Control Letters*, 17:79–88, 1991.
- [7] D. P. Bertsekas. *Nonlinear Programming*. Athen Scientific, Massachusetts, 1995.
- [8] S. Bittanti and P. Colaneri. *Encyclopedia of Electrical and Electronics Engineering*, volume 16. John Wiley and Sons, 1999. chapter Periodic Control, pages 59–73.
- [9] S. Boyd, L. EL Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15. SIAM Studies in Applied Mathematics, Philadelphia, 1994.
- [10] S. Boyd and L. Vandenberghe. *Lecture Notes on Convex Optimization*.
- [11] S. P. Boyd and C. H. Barratt. *Linear Controller Design—Limits of Performance*. Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- [12] R.W. Brocket. *Finite Dimensional Linear Systems*. John Wiley and sons inc, New York, 1970.
- [13] M. A. Dahleh, P. G. Voulgaris, and L. S. Valavani. Optimal and robust controllers for periodic and multirate systems. *IEEE Transactions on Automatic Control*, 37(1):90–99, January 1992.

- [14] C.A. Desoer and M. Vidyasagar. *Feedback Systems: Input-Output Properties*. Academic Press, New York, 1975.
- [15] J. Doyle, K. Lenz, and A. Packard. Design examples using  $\mu$ -synthesis: Space shuttle lateral axis fcs during reentry. In *NATO ASI Series, Modelling, Robustness, and Sensitivity Reduction in Control Systems*, volume 34. Springer-Verlag, 1987.
- [16] J. Doyle, A. Packard, and K Zhou. Review of LFTs, LMIs, and  $\mu$ . In *Proceedings of the 30th Conference on Decision and Control*, pages 1227–1232, Brighton, England, 1991.
- [17] J. C. Doyle. Analysis of feedback systems with structured uncertainties. In *IEE Proceedings*, volume D-129, pages 242–251, 1982.
- [18] J. Dugundji and V. Mukhopadhyay. Lateral bending-torsion of a thin beam under parametric excitation. *Journal of Applied Mechanics*, 40:693–698, September 1973.
- [19] J. Dugundji and J. H. Wendell. Some analysis methods for rotating systems with periodic coefficients. *AIAA Journal*, 21(6):890–897, June 1983.
- [20] G. E. Dullerud. *Control of Uncertain Sampled-Data Systems*. Birkhäuser, Boston, 1996.
- [21] M. Farkas. *Periodic Motions*. Springer-Verlag, New York, 1994.
- [22] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, New York, 1968.
- [23] J.-L. Goffin, J. Gondzio, R. Sarkissian, and J.-P. Vial. Solving nonlinear multicommodity flows problems by the analytical center cutting plane method. *Mathematical Programming*, 76:131–154, 1997.
- [24] J.-L. Goffin, A. Haurie, and J.-P. Vial. Decomposition and nondifferentiable optimization with the projective algorithm. *Management Science*, 38(2):284–302, 1992.
- [25] J.-L. Goffin, Z.-Q. Luo, and Y. Ye. Complexity analysis of an interior cutting plane method for convex feasibility problems. *SIAM J. of Optimization*, 6(3):638–652, August 1996.
- [26] J.-L. Goffin and F. Sharifi Mokhtarian. Using the primal dual infeasible Newton method in the analytical method for problems defined by deep cutting planes. *Journal of Optimization Theory and Applications*, 101:35–58, 1999.
- [27] J.-L. Goffin and J.-P. Vial. Cutting planes and column generation techniques with the projective algorithm. *Journal of Optimization Theory and Applications*, 65:409–429, 1989.
- [28] J.-L. Goffin and J.-P. Vial. Convex nondifferentiable optimization: A survey focussed on the analytical center cutting plane method. Technical Report Technical Report 99.02, HEC/Logilab, 1999.
- [29] J.-L. Goffin and J.-P. Vial. Shallow, deep and very deep cuts in the analytical center cutting plane method. *Mathematical Programming*, 84:89–103, 1999.

- [30] J.-L. Goffin and J.-P. Vial. A two-cuts approach in the analytical center cutting plane method. *Mathematical Methods of Operations Research*, 49(1):149–169, 1999.
- [31] J.-L. Goffin and J.-P. Vial. Multiple cuts in the analytical center cutting plane method. *SIAM Journal of Optimization*, 11(1):266–288, 2000.
- [32] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, New York, 1993.
- [33] S.R. Hall and N.M. Wereley. Generalized Nyquist stability criterion for linear time periodic systems. In *Proceedings of the American Control Conference*, pages 1518–1524, San Diego, CA, May 1990.
- [34] A. Hansson and L. Vandenberghe. Efficient solution of linear matrix inequalities for integral quadratic constraints. In *Proceedings of the 39th IEEE Conference on Decision and Control*, December 2000.
- [35] D. H. Jacobson, D. H. Martin, M. Pachter, and T. Geveci. *Extensions of Linear-Quadratic Control Theory*. Lecture Notes in Control and Information Sciences. Springer-Verlag, 1980.
- [36] U. Jönsson. *Robustness Analysis of Uncertain and Nonlinear Systems*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, 1996.
- [37] U. Jönsson, C.-Y. Kao, and A. Megretski. Robustness analysis of periodic systems. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 2, Phoenix, Arizona, USA, December 1999.
- [38] U. Jönsson, C.-Y. Kao, and A. Megretski. Robustness analysis of periodic trajectories. Technical Report TRITA/MAT-99-S5, Department of Mathematics, Royal Institute of Technology, Sweden, 1999.
- [39] U. Jönsson, C.-Y. Kao, and A. Megretski. Robustness of periodic trajectories. In *Proceedings of the American Control Conference 2000*, pages 1307–1311, Chicago, Illinois, USA, June 2000.
- [40] U. Jönsson, C.-Y. Kao, and A. Megretski. Analysis of uncertain periodically forced feedback systems. Technical Report TRITA/MAT-01-OS08, Department of Mathematics, Royal Institute of Technology, Sweden, 2001.
- [41] R. M. Mckillip Jr. Periodic control of the individual-blade-control helicopter rotor. *Vertica*, 9(2):199–225, 1985.
- [42] R.E. Kalman. Lyapunov functions for the problem of Lure in automatic control. *Proceedings of the National Academy of Sciences*, 49(2):201–205, 1963.
- [43] C.-Y. Kao, U. Jönsson, and A. Megretski. A cutting plane algorithm for robustness analysis of periodic systems. In *Proceedings of the 1999 American Control Conference*, pages 2365–2369, San Diego, CA, USA, June 1999.
- [44] C.-Y. Kao, A. Megretski, and U. Jönsson. An algorithm for solving special frequency dependent LMIs. In *Proceedings of the American Control Conference 2000*, pages 307–311, Chicago, Illinois, USA, June 2000.

- [45] C.-Y. Kao, A. Megretski, and U. Jönsson. A cutting plane algorithm for robustness analysis of periodically time-varying systems. *IEEE Transactions on Automatic Control*, 46(4):579–592, April 2001.
- [46] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.
- [47] J. E. Kelley. The cutting-plane method for solving convex problems. *Journal of the SIAM*, 8(4):703–712, 1960.
- [48] L. G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.
- [49] E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley & Sons, New York, 1978.
- [50] A.I. Lure and V.N. Postnikov. On the theory of stability of control systems. *Prikl. Mat. i Mekh*, 8:3–13, 1944.
- [51] A. Megretski. Fixed points in systems under an integral quadratic constraint. *Vestnik Leningradskogo Universiteta. Matematika*, 23(3):30–34, 1990.
- [52] A. Megretski. Power distribution approach in robust control. In *Proceedings of the IFAC Congress*, pages 399–402, Sydney, Australia, 1993.
- [53] A. Megretski. *Lecture Notes on Multivariable Control Systems*. MIT, Cambridge, MA, 2000.
- [54] A. Megretski, C.-Y. Kao, U. Jönsson, and A. Rantzer. A guide to IQC $\beta$ : Software for robust analysis. August 1999. A draft of the manual of IQC Toolbox. See <http://web.mit.edu/cykao/www/index.html>.
- [55] A. Megretski and A. Rantzer. System analysis via Integral Quadratic Constraints. *IEEE Transactions on Automatic Control*, 42(6):819–830, June 1997.
- [56] A. Megretski and S. Treil. Power distribution inequalities in optimization and robustness of uncertain systems. *Journal of Mathematical Systems, Estimation, and Control*, 3(3):301–319, 1993.
- [57] L. Mirkin and Z. J. Palmor. A new representation of the parameters of lifted systems. *IEEE Transactions on Automatic Control*, 44(4):833–840, April 1999.
- [58] Erik Möllerstedt. *Dynamic Analysis of Harmonics in Electrical Systems*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, 2000.
- [59] Y. Nesterov. Cutting plane algorithms from analytical centers: Efficient estimates. *Mathematical Programming*, 69(B):149–176, 1995.
- [60] Y. Nesterov and A. Nemirovski. *Interior Point Polynomial Methods in Convex Programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, 1994.
- [61] Pablo A Parrilo. On the numerical solution of LMIs derived from the KYP lemma. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 3, Phoenix, Arizona, USA, December 1999.

- [62] O. Péton and J.-P. Vial. *A tutorial on ACCPM*. available at <http://ecolu-info.unige.ch/logilab>.
- [63] V.M. Popov. Absolute stability of nonlinear systems of automatic control. *Automation and Remote Control*, 22:857–875, 1961.
- [64] A. Rantzer and A. Megretski. System analysis via Integral Quadratic Constraints. In *Proceedings of the IEEE Conference of Decision and Control*, volume 3, pages 3062–3067, Lake Buena Vista, Florida, 1994.
- [65] A. Rantzer and A. Megretski. Stability criteria based on Integral Quadratic Constraints. In *Proceedings of the IEEE Conference of Decision and Control*, volume 1, pages 215–220, Kobe, Japan, 1996.
- [66] A. Rantzer and A. Megretski. System analysis via Integral Quadratic Constraints, Part II. Technical Report TFTR-7559, Department of Automatic Control, Lund Institute of Technology, 1997.
- [67] A. Rantzer and A. Megretski. Analysis of rate limiters using Integral Quadratic Constraints. In *Proceedings of IFAC Nonlinear Control Systems Design Symposium*, pages 696–700, Enschede, Netherlands, July 1998.
- [68] J. Renegar. A polynomial-time algorithm, based on Newton’s method, for linear programming. *Mathematical Programming*, 40:59–93, 1988.
- [69] M.G. Safonov and M. Athans. A multi-loop generalization of the circle criterion for stability margin analysis. *IEEE Transactions on Automatic Control*, 26:415–422, 1981.
- [70] I.W. Sandberg. An observation concerning the application of the contraction mapping fixed-point theorem and a result concerning the norm boundedness of solutions of nonlinear functional equations. *Bell System Technical Journal*, 44, 1965.
- [71] I.W. Sandberg. Some results on the theory of physical systems governed by nonlinear functional equations. *Bell System Technical Journal*, 44:871–898, May–June 1965.
- [72] G. Sonnevand. New algorithms in convex programming based on a notion of ‘centre’ (for systems of analytical inequalities) and on rational extrapolation. In *Trends in Mathematical Optimization: Proceedings of the 4th French-German Conference on Optimization*, pages 311–327, Irsee West-Germany, April 1986. K. H. Hoffmann, J. B. Hiriart-Urruty, C. Lemaréchal, and J. Zowe, eds.
- [73] K. J. Spyrou. Dynamic instability in quartering seas - Part (III): Nonlinear effects on periodic motions. *Journal of Ship Research*, 41(3):210–223, November 1997.
- [74] P. M. Bara T. R. Kane. Attitude stability of a spinning satellite in an elliptical orbit. *Journal of Applied Mechanics*, 33:402–405, 1966.
- [75] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, March 1996.
- [76] P. G. Voulgaris, M. A. Dahleh, and L. S. Valavani.  $H_\infty$  and  $H_2$  optimal controllers for periodic and multirate systems. *Automatica*, 30(2):252–263, 1994.

- [77] R. Wallin, A. Hansson, and L. Vandenberghe. Efficient implementations of interior-point methods for integral quadratic constraints. In *Fourth SIAM Conference on Linear Algebra in Signals, Systems and Control*, Cambridge, MA, August 2001.
- [78] J.C. Willems. Dissipative dynamical system, Part I: General theory. *Archive for Rational Mechanics and Analysis*, 45:321–353, 1972.
- [79] V. A. Yakubovich. A linear-quadratic optimization problem and the frequency theorem for nonperiodic systems. *Translation from Sibirskii Matematicheskii Zhurnal*, 27(4):181–200, July-August 1986.
- [80] V. A. Yakubovich. Frequency theorem for the periodic systems and the theory of the analytical design of regulators. In *Lyapunov Function Method in Analysis of Systems Dynamics*, pages 281–290, 1987.
- [81] V. A. Yakubovich. Absolute stability of nonlinear systems with periodically nonstationary linear part. *DAN SSSR*, 298(2):299–303, 1988.
- [82] V. A. Yakubovich. Nonconvex optimization problem: The infinite-horizon linear quadratic control problem with quadratic constraints. *System and Control Letters*, 19:13–22, 1992.
- [83] V.A. Yakubovich. Solution of certain matrix inequalities occurring in the theory of automatic control. *Docl. Acad. Nauk. SSSR*, 143:1304–1307, 1962.
- [84] V.A. Yakubovich. Frequency conditions for absolute stability of control systems with hysteresis nonlinearities. *Dokl. Akad. Nauk SSSR*, 149(2):288–291, 1963.
- [85] V.A. Yakubovich. The method of matrix inequalities in the theory of stability of nonlinear controlled systems. Part I. *Avtomatica i Telemekhanika*, 25(7):1017–1029, 1964.
- [86] V.A. Yakubovich. S-procedure in nonlinear control theory. *Vestnik Leningrad University*, pages 62–77, 1971. (English translation in *Vestnik Leningrad Univ. Math.* 4:73–93, 1977.).
- [87] K. Y. Yang. *Efficient Design of Robust Controllers for  $\mathcal{H}_2$  Performance*. PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, 1997.
- [88] Y. Ye. A potential reduction algorithm allowing column generation. *SIAM Journal of Optimization*, 2:7–20, 1992.
- [89] Y. Ye. Complexity analysis of the analytical center cutting plane method that uses multiple cuts. *Mathematical Programming*, 78:85–104, 1997.
- [90] Y. Ye. *Inter Point Algorithms, Theory and Analysis*. John Wiley and Sons, New York, 1997.
- [91] G Zames. On the input-output stability of nonlinear time-varying feedback systems — Part I: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Transactions on Automatic Control*, 11:228–238, April 1966.

- [92] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, New Jersey, 1996.