

**NEW BARRIER FUNCTIONS AND INTERIOR  
PATH-FOLLOWING ALGORITHMS FOR  
IQC OPTIMIZATION PROBLEMS**

C.-Y. KAO

REPORT No. 24, 2002/2003, spring

ISSN 1103-467X

ISRN IML-R- -24-02/03- -SE+spring



**INSTITUT MITTAG-LEFFLER**  
THE ROYAL SWEDISH ACADEMY OF SCIENCES

# New Barrier Functions and Interior Path-following Algorithms for IQC Optimization Problems

Chung-Yao Kao\*<sup>†</sup>  
cykao@alum.mit.edu

June 15, 2003

## Abstract

We propose 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. The conventional approach to solve IQC optimization problems requires introducing auxiliary decision variables. The path-following algorithms based on the new barrier functions do not have this requirement. As a result, the algorithm can solve IQC optimization problems in a more efficient fashion. Numerical examples are used to evaluate the efficiency of the proposed algorithm.

**Keywords:** interior point method, barrier function, robustness analysis, integral quadratic constraint.

**AMS classification:** 93D09

## 1 Introduction

The idea of utilizing Integral Quadratic Constraints (IQC) to characterize uncertainties and/or nonlinearities in the system to be analyzed, offer flexible frameworks to analyze large and complex systems [15, 5, 19, 18, 12]. In the 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) [12], 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 auxiliary decision variables is often much larger than the number of original decision variables. The additional decision variables hence become the major computational burden, and most of computational effort is inefficiently spent on computing something auxiliary. In these cases, more efficient algorithms which achieve fast computation are very desirable.

To improve efficiency of the conventional approach, some specialized algorithms have been proposed recently [6, 7, 17]. The basic idea behind these algorithms is to explore and exploit the special structure of the IQC problems and construct specialized interior point algorithms in which computation

---

\*Chung-Yao Kao received his Sc.D. degree from M.I.T. in August 2002. He spent the spring semester of 2003 in the Mittag-Leffler Institute as a postdoctoral researcher. He is currently a postdoctoral research associate with the Division of Optimization and Systems Theory, Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden.

<sup>†</sup>This research was supported by grants provided by the National Science Foundations (US), the Air Force Office of Scientific Research (US), the Defense Advanced Research Projects Agency (US), and by the Mittag-Leffler Institute (Sweden).

of descent directions is performed efficiently. In [6, 7], it is reported that such a specialized algorithm can solve an IQC optimization problem with about 5000 decision variables in about 10 minutes on a laptop computer. It is much faster than the general purpose SDP solver of the MATLAB LMI Control Toolbox.

On the other hand, 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, several cutting plane algorithms were proposed to solve IQC optimization problems in a more efficient fashion [8, 9, 14]. 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, numerical experience indicates that the number of iterations grows substantially as the number of decision variables becomes larger.

In contrast to the cutting plane method, the interior point method generally does not require many iterations to converge. The development of interior point methods dates back to the 1950s, and it has gained much attention and become popular since Karmarkar introduced his famous, polynomial-time algorithm for solving linear programs [10]. Another milestone in the development of interior point methods was the result by Nesterov and Nemirovski [13]. 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 Nemirovski 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. In the case of IQC optimization problems, 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 paper, new barrier functions for solving IQC optimization problems are proposed. The 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. More importantly, there is no auxiliary decision variables involved in the barrier functions. Based on the new barrier functions, we develop interior path-following algorithms to solve IQC optimization problems. Results of numerical experiments indicate that these interior path-following algorithms are able to solve IQC optimization problems in a much more efficient fashion. Regarding the issues of convergence and computational complexity, it can be shown that the algorithm converges, but the polynomial-time complexity is not yet determined. We can show that one of the the barrier functions proposed is indeed self-concordant. However, the self-concordant coefficient appears to be dependent on the problem data, and the exact dependency is still open to us. Hence, we are not able to apply Nesterov and Nemirovski's results for constructing a polynomial-time algorithm.

The paper is organized as follows: following the introduction section, in Section 2 we define the notations to be used and formally state the IQC optimization problem under consideration. In Section 3, we briefly review the bases of the interior path-following method. In Section 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 Hessians at given points. We also prove certain properties of the barrier function, which ensure that the interior path-following algorithm based on the proposed barrier function converges globally. The algorithms based on the new barrier functions are summarized in Section 5. In Section 6, we explain why the proposed algorithm is 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 a comparison with the conventional approach are presented

in Section 7. Finally, the Appendix section collects the proofs of some technical results. These proofs are put in the end of the paper for the sake of a smooth presentation of the material.

## 2 Notations and Problems Under Consideration

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  $\nabla_i F(\lambda)$ . The second partial derivative of  $F(\lambda)$  with respect to the  $i^{\text{th}}$  and  $j^{\text{th}}$  components of  $\lambda$  is denoted by  $\nabla_{ij}^2 F(\lambda)$ . 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  $h_i \in \mathbf{R}^n$ .

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  $M = M^*$ . A matrix  $M$  is called *positive definite* if  $x'Mx > 0$  for all  $x \neq 0$ . 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. The notation  $\text{tr}(M)$  denotes the trace of  $M$ . The *Frobenius norm* of a square matrix  $M$  is defined as  $\|M\|_F := \sqrt{\text{tr}(M)}$ . Let  $M_1, \dots, M_n$  be square matrices. Then  $M = \text{diag}(M_1, \dots, M_n)$  defines the block diagonal matrix

$$M = \begin{bmatrix} M_1 & & \\ & \ddots & \\ & & M_n \end{bmatrix}.$$

### IQC Optimization Problems

The standard IQC optimization problem can be formulated as (See [12])

$$\inf_{\lambda} c'\lambda, \quad \mathbf{H}(\omega, \lambda) > 0, \quad \forall \omega \in [0, \infty], \quad (1)$$

where  $\mathbf{H}(\omega, \lambda)$  is 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

$$\mathbf{H}_i(\omega) = \begin{bmatrix} (j\omega I_m - A)^{-1} B \\ I_r \end{bmatrix}^* \begin{bmatrix} Q_i & F_i \\ F_i' & R_i \end{bmatrix} \begin{bmatrix} (j\omega I_m - A)^{-1} B \\ I_r \end{bmatrix},$$

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

$$\Sigma_i = \begin{bmatrix} Q_i & F_i \\ F_i' & R_i \end{bmatrix}, \quad i = 1, \dots, n. \quad (2)$$

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

$$Q_0 + \sum_{i=1}^n \lambda_i Q_i, \quad F_0 + \sum_{i=1}^n \lambda_i F_i, \quad R_0 + \sum_{i=1}^n \lambda_i R_i, \quad (3)$$

respectively. The feasible set of the IQC optimization problems is denoted by  $\Omega$ ; i.e.,

$$\Omega = \{\lambda \mid \mathbf{H}(\omega, \lambda) > 0, \forall \omega \in [0, \infty]\}. \quad (4)$$

We assume that  $\Omega$  is a bounded subset of  $\mathbf{R}^n$ .

In this paper, we propose new barrier functions for  $\Omega$ . Furthermore, standard interior path-following algorithms based on the proposed barrier function are implemented to solve IQC optimization problem (1).

### 3 The Interior Path-Following Algorithm

In this section, we briefly review the interior path-following method for solving general convex optimization problems. The idea of this method was first proposed in the 1960s by Fiacco and McCormick. Since then, a large literature on this subject has been produced. The contents in this section are more or less well-known, and can be found in, for example, [4, 2, 13].

Consider the optimization problem

$$\inf c' \lambda, \quad \text{subj. to } \lambda \in \Lambda \quad (5)$$

where  $\Lambda \subset \mathbf{R}^n$  is a bounded open convex set. 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  $\Lambda$ .

Such a function is often referred to as a barrier function of  $\Lambda$ . Now, let  $\varphi_t(\lambda)$  be defined as

$$\varphi_t(\lambda) = tc' \lambda + B(\lambda),$$

where  $t > 0$ . For any fixed  $t$ , consider the minimization problem

$$\inf \varphi_t(\lambda) \quad \text{subj. to } \lambda \in \Lambda. \quad (6)$$

Problem (6) is often referred to as *centering*, and the minimizer  $\lambda^*(t)$  is called the center of (6). Note that since  $\varphi_t(\lambda)$  is a convex function which blows up at the boundary of  $\Lambda$  and  $\Lambda$  is a convex set, problem (6) has a unique minimizer which is strictly inside  $\Lambda$ . Suppose that a point  $\lambda_0$  inside  $\Lambda$  is known and a gradient method is applied to solve (6) starting at  $\lambda_0$ . 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 the boundary of  $\Lambda$ . Therefore, the *infimum* in (6) can be replaced by *minimum*, and the problem (6) is solved as if there is no constraint  $\lambda \in \Lambda$ . The curve  $\lambda^*(t)$ ,  $t > 0$ , is referred to as the *central path*.

### 3.1 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 (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 (5).

The idea of the interior path-following algorithm is to follow the central path  $\lambda^*(t)$  to an optimal solution of (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 (5). 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 algorithm, it is assumed that a strictly feasible point is available to start with. A feasible point of optimization problem (5) can be found using the so-called Phase-I, or Big-M, method. Readers are referred to [4] for the details.

### 3.2 Convergency and Complexity

The following theorem gives a weak result regarding convergence of the interior path-following algorithm.

**Theorem 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 (5).*

*Proof.* The proof can be found in Chapter 4 of [2]. □

Theorem (1) 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. A major breakthrough in this subject is the result by Nesterov and Nemirovski [13]. They discovered two key properties of a barrier function which are responsible for the polynomiality of the associated interior path-following algorithm. Their results are briefly summarized in the followings.

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  $h \in \mathbf{R}^n$ :

$$|\nabla^3 F(x)[h, h, h]| \leq 2a^{-1/2}(\nabla^2 F(x)[h, h])^{\frac{3}{2}}. \quad (7)$$

$F(x)$  is called strongly  $a$ -self-concordant if furthermore  $F(x) \rightarrow \infty$  as  $x$  approaches the boundary of  $X$ .

Now, consider again the minimization problem (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  $|\nabla B(\lambda)[h]| \leq c_1(\nabla^2 B(\lambda)[h, h])^{\frac{1}{2}}$  holds for all  $\lambda \in \Lambda$ , and for all  $h \in \mathbf{R}^n$ .
- (P3). There exists a constant  $c_2$  such that inequality  $|\nabla^3 B(\lambda)[h, h, h]| \leq c_2(\nabla^2 B(\lambda)[h, h])^{\frac{3}{2}}$  holds for all  $\lambda \in \Lambda$ , and for all  $h \in \mathbf{R}^n$ .

By properties (P1) and (P3), 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 Nemirovski refer to a barrier function which satisfies all three properties with  $c_2 = 2$  (This implies that  $B(\lambda)$  is a strongly 1-self-concordant function.) and  $c_1 = \vartheta$  as a  $\vartheta$ -self-concordant barrier.

Nesterov and Nemirovski gave a complete complexity analysis of interior path-following algorithms with self-concordant barriers. They discovered that a path-following algorithm associated with a  $\vartheta$ -self-concordant barrier for a bounded convex domain minimizes a linear function over the domain to an accuracy  $\epsilon$  in  $\mathbf{O}(\sqrt{\vartheta} \ln(\vartheta/\epsilon))$  Newton steps. Their results about complexity of path-following methods are detailed in Chapters 3 of [13].

## 4 New Barrier Functions for IQC Optimization Problems

Now let us consider the IQC optimization problem (1). It is well-known [3] that the frequency dependent matrix inequality in (1) holds if and only if there exists a symmetric matrix  $P$  such that

$$E(P, \lambda) := \begin{bmatrix} PA + A'P & PB \\ B'P & 0 \end{bmatrix} + \Sigma_0 + \sum_{i=1}^n \lambda_i \Sigma_i > 0. \quad (8)$$

Hence, problem (1) is equivalent to a Semi-Definite Program (SDP)

$$\inf_{P=P', \lambda} c' \lambda, \quad E(P, \lambda) > 0. \quad (9)$$

The feasible set of problem (9) has a well-known self-concordant barrier

$$-\log \det(E(P, \lambda)), \quad (10)$$

where the notation  $\det(M)$  denotes the determinate of the matrix  $M$ . The barrier function (10), however, involves an additional matrix variable  $P$ . The number of 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 (10). 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 paper is to propose alternative barrier functions which can be used to construct *more* efficient interior point algorithms for solving IQC optimization problems.

#### 4.1 The First Barrier Function

Consider the following function from  $\Omega$  to  $\mathbf{R}^+$

$$\mathcal{G}(\lambda) := \frac{1}{\pi} \int_{-\infty}^{\infty} \mathbf{tr}(\mathbf{H}(\omega, \lambda)^{-1}) \frac{d\omega}{1 + \omega^2}, \quad \lambda \in \Omega. \quad (11)$$

It is obvious that  $\mathcal{G}(\lambda)$  is smooth. The  $i^{\text{th}}$  element of the gradient and the  $(i, j)$  entry of the Hessian of  $\mathcal{G}(\lambda)$  are given as follows:

$$\nabla_i \mathcal{G}(\lambda) = -\frac{1}{\pi} \int_{-\infty}^{\infty} \mathbf{tr}(\mathbf{H}(\omega, \lambda)^{-1} \mathbf{H}_i(\omega) \mathbf{H}(\omega, \lambda)^{-1}) \frac{d\omega}{1 + \omega^2}, \quad (12)$$

$$\nabla_{ij}^2 \mathcal{G}(\lambda) = \frac{2}{\pi} \int_{-\infty}^{\infty} \mathbf{tr}(\mathbf{H}(\omega, \lambda)^{-1} \mathbf{H}_i(\omega) \mathbf{H}(\omega, \lambda)^{-1} \mathbf{H}_j(\omega) \mathbf{H}(\omega, \lambda)^{-1}) \frac{d\omega}{1 + \omega^2}. \quad (13)$$

Furthermore, it is easy to verify that  $\mathcal{G}(\lambda)$  is a convex function.

**Proposition 1.**  $\mathcal{G}(\lambda)$  is a convex function.

*Proof.* We show that the Hessian of  $\mathcal{G}(\lambda)$  is strictly positive definite on  $\Omega$ . By (13), we have<sup>1</sup>

$$\begin{aligned} \nabla^2 \mathcal{G}(\lambda)[h, h] &= \sum_{i,j} \nabla_{ij}^2 \mathcal{G}(\lambda) h_i h_j = \frac{2}{\pi} \int_{-\infty}^{\infty} \mathbf{tr}(\mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1}) \frac{d\omega}{1 + \omega^2} \\ &= \frac{2}{\pi} \int_{-\infty}^{\infty} \left\| \mathbf{H}^{-\frac{1}{2}} \mathcal{A} \mathbf{H}^{-1} \right\|_F^2 \frac{d\omega}{1 + \omega^2}, \end{aligned}$$

where  $\mathcal{A} = \sum_{i=1}^n h_i \mathbf{H}_i(\omega)$ . Therefore  $\nabla^2 \mathcal{G}(\lambda)[h, h] > 0$  for all  $\lambda \in \Omega$  and for all  $h \neq 0$ . This concludes the proof.  $\square$

Although the integration over an infinite horizon makes  $\mathcal{G}(\lambda)$  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  $\mathcal{G}(\lambda)$  is to solve one Riccati equation and one Lyapunov equation. We here

<sup>1</sup>The  $\omega, \lambda$  dependence of  $\mathbf{H}(\omega, \lambda)$  and  $\mathbf{H}_i(\omega)$  are suppressed for simplifying the notation.

present the computational procedure. Furthermore, we can also show that the value of  $\mathcal{G}(\lambda)$  approaches infinity as  $\lambda$  approaches the boundary of  $\Omega$ . This property, together with smoothness and convexity, makes  $\mathcal{G}(\lambda)$  a natural barrier for  $\Omega$ . Before we prove this property, we first discuss how to evaluate  $\mathcal{G}(\lambda)$ .

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 (3). The following lemma is essential for obtaining an efficient computational procedure to evaluate  $\mathcal{G}(\tilde{\lambda})$ .

**Lemma 1.** *Assume that  $A \in \mathbf{R}^{m \times m}$  is a Hurwitz matrix. The following three statements are equivalent*

- 1  $\mathbf{H}(\omega, \tilde{\lambda}) > 0 \forall \omega \in [0, \infty]$ .
- 2  $\mathbf{H}(\omega, \tilde{\lambda})^{-1}$  can be factorized as  $D_H + G_H(j\omega) + G_H(j\omega)^*$ , where

$$G_H(s) = C_H(sI - A_H)^{-1}B_H, \quad (14)$$

$$A_H = A + BC_H, \quad B_H = B\tilde{R}^{-1} + YC'_H, \quad (15)$$

$$C_H = -\tilde{R}^{-1}(PB + \tilde{F})', \quad D_H = \tilde{R}^{-1}, \quad (16)$$

and  $P, Y$  satisfy the following Riccati and Lyapunov equations, respectively

$$PA + A'P + \tilde{Q} - (PB + \tilde{F})\tilde{R}^{-1}(PB + \tilde{F})' = 0, \quad (17)$$

$$A_H Y + Y A'_H + B\tilde{R}^{-1}B' = 0. \quad (18)$$

Furthermore, matrix  $A_H$  is a Hurwitz matrix and has the same dimension of the matrix  $A$ .

- 3  $\mathbf{H}(\omega, \tilde{\lambda})^{-1}$  can be factorized as  $\Psi(j\omega)\Psi(j\omega)^*$ , where

$$\Psi(s) = \tilde{R}^{\frac{1}{2}} + C_H(sI - A_H)^{-1}B\tilde{R}^{\frac{1}{2}}. \quad (19)$$

*Proof.* See [20]. □

**Lemma 2.** *Let  $\tilde{\lambda} \in \Omega$ . Then  $\mathcal{G}(\tilde{\lambda}) = \text{tr}(D_H) + 2\text{tr}(C_H(I - A_H)^{-1}B_H)$ , where  $A_H, B_H, C_H, D_H$  are defined in (15) to (18).*

*Proof.* First, notice that the order of the trace operator and the integral operator can be reversed; therefore, by the second statement of Lemma 1, we have

$$\begin{aligned} \mathcal{G}(\tilde{\lambda}) &= \text{tr} \left( \frac{1}{\pi} \int_{-\infty}^{\infty} (D_H + G_H(j\omega) + G_H(j\omega)^*) \frac{d\omega}{1 + \omega^2} \right) \\ &= \text{tr}(D_H) + \text{tr} \left( \frac{1}{\pi} \int_{-\infty}^{\infty} (G_H(j\omega) + G_H(j\omega)^*) \frac{d\omega}{1 + \omega^2} \right). \end{aligned}$$

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  $\mathcal{G}(\tilde{\lambda})$  is equal to the formula stated in the lemma. □

Therefore, computation of  $\mathcal{G}(\tilde{\lambda})$  mainly involves solving Riccati equation (17) and Lyapunov equation (18). Solving Riccati equations and Lyapunov equations has been well studied, and efficient computational routines are widely available. Thus, evaluation of  $\mathcal{G}(\lambda)$  at a given  $\lambda$  can be performed very efficiently.

The following proposition shows that that  $\mathcal{G}(\lambda)$  approaches to infinity as  $\lambda$  approaches to the boundary of  $\Omega$ .

**Proposition 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  $\mathcal{G}(\lambda_n) \rightarrow \infty$ .*

*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  $\Gamma = \{\pm j\omega_i, i = 1, \dots, m\}$ . 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 2 that  $\mathcal{G}(\lambda)$  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 (19). Note that

$$\frac{1}{\pi} \int_{-\infty}^{\infty} \mathbf{tr}(\mathbf{H}(\omega, \tilde{\lambda})^{-1}) \frac{d\omega}{1 + \omega^2} = \frac{1}{\pi} \int_{-\infty}^{\infty} \mathbf{tr}(\Psi(j\omega)\Psi(j\omega)^*) \frac{d\omega}{1 + \omega^2} = 2\|\tilde{\Psi}(s)\|_{\mathcal{H}_2}, \quad (20)$$

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  $\mathcal{G}(\tilde{\lambda})$  is unbounded. This concludes the proof.  $\square$

## Gradients and Hessians of $\mathcal{G}(\lambda)$

Lemma 2 gives the following equivalent expression for  $\mathcal{G}(\lambda)$

$$\mathcal{G}(\lambda) = \mathbf{tr}(D_H(\lambda) + 2C_H(\lambda)(I - A_H(\lambda))^{-1}B_H(\lambda)), \quad (21)$$

where  $A_H(\lambda)$ ,  $B_H(\lambda)$ ,  $C_H(\lambda)$ , and  $D_H(\lambda)$  are defined as in (14) to (18) with  $\tilde{Q}$ ,  $\tilde{F}$  and  $\tilde{R}$  replaced by  $Q(\lambda)$ ,  $F(\lambda)$ , and  $R(\lambda)$ . By partially differentiating (21) with respect to  $\lambda_i$ , we obtain the following expressions for the  $i^{\text{th}}$  component of the gradient of  $\mathcal{G}(\lambda)$

$$\begin{aligned} \nabla_i \mathcal{G}(\lambda) &= \mathbf{tr}(\nabla_i D_H) + 2((\nabla_i C_H)(I - A_H)^{-1}B_H - C_H(I - A_H)^{-1}(\nabla_i A_H)(I - A_H)^{-1}B_H \\ &\quad + C_H(I - A_H)^{-1}(\nabla_i B_H)) \\ &= \mathbf{tr}(\nabla_i D_H + 2(I + C_H(I - A_H)^{-1}B)((\nabla_i C_H)(I - A_H)^{-1}B_H) \\ &\quad + 2C_H(I - A_H)^{-1}(\nabla_i B_H)), \end{aligned} \quad (22)$$

where the second equality is obtained by noting that  $\nabla_i A_H = B(\nabla_i C_H)$ . It can be easily verified that the partial derivatives of  $B_H$ ,  $C_H$ , and  $D_H$  have the following expressions

$$\nabla_i B_H = (\nabla_i Y)C_H' + Y(\nabla_i C_H)', \quad (23)$$

$$\nabla_i C_H = R^{-1}R_i R^{-1}(PB + F)' - R^{-1}((\nabla_i P)B + F_i)', \quad (24)$$

$$\nabla_i D_H = -R^{-1}R_i R^{-1}, \quad (25)$$

and the partial derivatives of  $P$  and  $Y$  satisfy the following equations

$$(\nabla_i P)A_H + A_H'(\nabla_i P) + (Q_i + F_i C_H + C_H' F_i' + C_H' R_i C_H) = 0, \quad (26)$$

$$A_H(\nabla_i Y) + (\nabla_i Y)A_H' + (B(\nabla_i C_H)Y + Y(\nabla_i C_H)'B' - BR^{-1}R_i R^{-1}B') = 0. \quad (27)$$

For a given point  $\tilde{\lambda} \in \Omega$ , computation of  $\nabla_i \mathcal{G}(\tilde{\lambda})$  can be performed as follows: first, notice that for a fixed  $\tilde{\lambda}$ , equation (26) is a Lyapunov equation with respect to  $\nabla_i P$ . Thus, the value of  $\nabla_i P$  can be obtained by solving the Lyapunov equation. Then the values of  $\nabla_i D_H$ ,  $\nabla_i C_H$  can be computed according to expressions (25) and (24), respectively. As soon as the value of  $\nabla_i C_H$  is available, one can solve another Lyapunov equation (27) to obtain the value of  $\nabla_i Y$  and then evaluate  $\nabla_i B_H$  using expression (23). Finally,  $\nabla_i \mathcal{G}(\tilde{\lambda})$  can be evaluated using equation (22). Hence, the main computation for obtaining the value of every entry of the gradient of  $\mathcal{G}(\lambda)$  is to solve **two Lyapunov equations**.

If we further differentiate (22) with respect to  $\lambda_j$ , we obtain an expression for  $\nabla_{ij} \mathcal{G}(\lambda)$

$$\nabla_{ij} \mathcal{G}(\lambda) = \text{tr}(\nabla_{ij}^2 D_H) + 2\text{tr}(C_H(I - A_H)^{-1}(\nabla_{ij}^2 B_H) + (I + C_H(I - A_H)^{-1}B)T_1), \quad (28)$$

where

$$T_1 = (\nabla_{ij}^2 C_H)(I - A_H)^{-1}B_H + (\nabla_i C_H)(I - A_H)^{-1}((\nabla_j B_H) + B(\nabla_j C_H)(I - A_H)^{-1}B_H) \\ + (\nabla_j C_H)(I - A_H)^{-1}((\nabla_i B_H) + B(\nabla_i C_H)(I - A_H)^{-1}B_H), \quad (29)$$

$$\nabla_{ij}^2 B_H = (\nabla_{ij}^2 Y)C_H' + Y(\nabla_{ij}^2 C_H)' + (\nabla_i Y)(\nabla_j C_H)' + (\nabla_j Y)(\nabla_i C_H)', \quad (30)$$

$$\nabla_{ij}^2 C_H = -R^{-1}(R_j(\nabla_i C_H) + R_i(\nabla_j C_H) + B'(\nabla_{ij}^2 P)'), \quad (31)$$

$$\nabla_{ij}^2 D_H = R^{-1}R_i R^{-1}R_j R^{-1} + R^{-1}R_j R^{-1}R_i R^{-1}. \quad (32)$$

$\nabla_{ij}^2 P$  and  $\nabla_{ij}^2 Y$  satisfy the following Lyapunov equations

$$(\nabla_{ij}^2 P)A_H + A_H'(\nabla_{ij}^2 P) + (T_2 + T_2') = 0, \quad (33)$$

$$(\nabla_{ij}^2 Y)A_H + A_H'(\nabla_{ij}^2 Y) + (T_3 + T_3') = 0, \quad (34)$$

where  $T_2$  and  $T_3$  denote the following expressions

$$T_2 = ((\nabla_i P)B + C_H R_i + F_i)(\nabla_j C_H), \quad (35)$$

$$T_3 = B((\nabla_i C_H)(\nabla_j Y) + (\nabla_j C_H)(\nabla_i Y) + (\nabla_{ij}^2 C_H)Y) + BR^{-1}R_i R^{-1}R_j R^{-1}B'. \quad (36)$$

For a given  $\tilde{\lambda} \in \Omega$ , the computational procedure for evaluating  $\nabla_{ij}^2 \mathcal{G}(\tilde{\lambda})$  is similar to the one for computing  $\nabla_i \mathcal{G}(\tilde{\lambda})$ . 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 (33) can be solved for the value of  $\nabla_{ij}^2 P$ . As soon as the value of  $\nabla_{ij}^2 P$  is available, one can evaluate  $\nabla_{ij}^2 B_H$ ,  $\nabla_{ij}^2 C_H$ , and  $\nabla_{ij}^2 D_H$  using expressions (30) to (32). Once the value of  $\nabla_{ij}^2 C_H$  is obtained, Lyapunov equation (34) can be solved for the value of  $\nabla_{ij}^2 Y$ . Finally,  $\nabla_{ij}^2 \mathcal{G}(\tilde{\lambda})$  can be computed according to expression (28). Thus, the main computation for obtaining the value of every entry of the Hessian of  $\mathcal{G}(\lambda)$  is again to solve **two Lyapunov equations**.

## Self-Concordant Properties

In order to apply Nesterov and Nemirovski's results to construct a polynomial time interior path-following algorithm, the barrier function  $\mathcal{G}(\lambda)$  has to be a self-concordant barrier; i.e., to satisfy properties (P1) to (P3). The following counter-example demonstrates that in general,  $\mathcal{G}(\lambda)$  does not satisfy property (P3).

**Example 1.** 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 2, that  $\mathcal{G}(\lambda) =$

$2/\sqrt{\lambda(\lambda-1)}$ . Therefore, the first and second derivatives of  $\mathcal{G}(\lambda)$  can be computed straightforwardly:

$$\begin{aligned}\dot{\mathcal{G}}(\lambda) &= -\frac{2\lambda-1}{(\lambda(\lambda-1))^{\frac{3}{2}}}, \\ \ddot{\mathcal{G}}(\lambda) &= \frac{1}{2} \frac{8\lambda^2-8\lambda+3}{(\lambda(\lambda-1))^{\frac{5}{2}}}.\end{aligned}$$

Let

$$r_1 := \frac{\dot{\mathcal{G}}(\lambda)^2}{\ddot{\mathcal{G}}(\lambda)} = \frac{2(2\lambda-1)^2}{(\lambda(\lambda-1))^{\frac{1}{2}}(8\lambda^2-8\lambda+3)}.$$

Obviously,  $r_1 \rightarrow \infty$  as  $\lambda \rightarrow 1$ . Thus, there exists no constant  $c$  such that  $|\dot{\mathcal{G}}(\lambda)| \leq c \cdot (\ddot{\mathcal{G}}(\lambda))^{\frac{1}{2}}$ ,  $\forall \lambda \in (1, \infty)$ .

Therefore, we can not apply Nesterov and Nemirovski's results to construct a polynomial time interior path-following algorithm using barrier function  $\mathcal{G}(\lambda)$ . Nevertheless,  $\mathcal{G}(\lambda)$  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 1 guarantees that the interior path-following algorithm based on  $\mathcal{G}(\lambda)$  converges globally.

## 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 from  $\Omega$  to  $\mathbf{R}$

$$\mathcal{B}(\lambda) = \log \mathcal{G}(\lambda). \quad (37)$$

Obviously,  $\mathcal{B}(\lambda)$  is a smooth function, and  $\mathcal{B}(\lambda)$  approaches infinity as  $\lambda$  approaches the boundary of  $\Omega$  since  $\mathcal{G}(\lambda)$  does. The following proposition shows that  $\mathcal{B}(\lambda)$  is also a convex function. Hence,  $\mathcal{B}(\lambda)$  is a well-defined barrier for  $\Omega$ .

**Proposition 3.**  $\mathcal{B}(\lambda)$  is a convex function.

*Proof.* Again, we show that the Hessian of  $\mathcal{B}(\lambda)$  is strictly positive definite on  $\Omega$ . It can be easily verified that

$$\nabla^2 \mathcal{B}(\lambda) = \mathcal{G}^{-1}(\lambda) \nabla^2 \mathcal{G}(\lambda) - \nabla \mathcal{G}(\lambda) \mathcal{G}^{-2}(\lambda) \nabla \mathcal{G}(\lambda)'$$

Since  $\mathcal{G}(\lambda) > 0$  for any  $\lambda \in \Omega$ , therefore, given any  $\lambda \in \Omega$ ,  $\nabla^2 \mathcal{B}(\lambda) > 0$  if and only if  $\nabla^2 \mathcal{G}(\lambda) - \nabla \mathcal{G}(\lambda) \mathcal{G}^{-1}(\lambda) \nabla \mathcal{G}(\lambda)' > 0$ . To prove  $\nabla^2 \mathcal{G}(\lambda) - \nabla \mathcal{G}(\lambda) \mathcal{G}^{-1}(\lambda) \nabla \mathcal{G}(\lambda)' > 0$ , it is sufficient to show that

$$\begin{bmatrix} \nabla^2 \mathcal{G}(\lambda) & \nabla \mathcal{G}(\lambda) \\ \nabla \mathcal{G}(\lambda)' & \mathcal{G}(\lambda) \end{bmatrix} > 0. \quad (38)$$

Matrix inequality (38) holds if  $\nabla^2 \mathcal{G}(\lambda)[h, h] + 2\nabla \mathcal{G}(\lambda)[h] + \mathcal{G}(\lambda) > 0$ , for all non-zero  $h$  in  $\mathbf{R}^n$ . Let  $\mathcal{A} = \sum_i^n h_i \mathbf{H}_i(\omega)$ . By (12) and (13), we have

$$\begin{aligned}\nabla^2 \mathcal{G}(\lambda)[h, h] + 2\nabla \mathcal{G}(\lambda)[h] + \mathcal{G}(\lambda) &= \sum_{i,j} h_i h_j \partial_{ij}^2 \mathcal{G}(\lambda) + 2 \sum_i h_i \partial_i \mathcal{G}(\lambda) + \mathcal{G}(\lambda) \\ &= \frac{1}{\pi} \int_{-\infty}^{\infty} \text{tr} \left( 2\mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} - 2\mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} + \mathbf{H}^{-1} \right) \frac{d\omega}{1+\omega^2} \\ &= \frac{1}{\pi} \int_{-\infty}^{\infty} 2 \|\mathbf{H}^{-\frac{1}{2}} \mathcal{A} \mathbf{H}^{-1} - 0.5 \mathbf{H}^{-\frac{1}{2}}\|_F^2 + 0.5 \|\mathbf{H}^{-\frac{1}{2}}\|_F^2 \frac{d\omega}{1+\omega^2}.\end{aligned}$$

Hence, for any  $\lambda \in \Omega$ ,  $\nabla^2 \mathcal{G}(\lambda)[h, h] + 2\nabla \mathcal{G}(\lambda)[h] + \mathcal{G}(\lambda) > 0$  for all  $h \neq 0$ . This in turn implies that  $\nabla^2 \mathcal{B}(\lambda) > 0$  for all  $\lambda \in \Omega$ , and thus  $\mathcal{B}(\lambda)$  is a convex function.  $\square$

Since  $\mathcal{B}(\lambda) = \log(\mathcal{G}(\lambda))$ , evaluation of barrier function  $\mathcal{B}(\lambda)$  is performed by a computational procedure similar to the one for evaluating  $\mathcal{G}(\lambda)$ . The only difference is that the logarithm of the value of  $\mathcal{G}(\lambda)$  is taken in the end. Therefore, the complexity of evaluating  $\mathcal{B}(\lambda)$  is almost the same as the complexity of evaluating  $\mathcal{G}(\lambda)$ . As to the gradient and the Hessian of  $\mathcal{B}(\lambda)$ , it can be readily verified that

$$\begin{aligned}\nabla \mathcal{B}(\lambda) &= \mathcal{G}^{-1}(\lambda) \nabla \mathcal{G}(\lambda), \\ \nabla^2 \mathcal{B}(\lambda) &= \mathcal{G}^{-1}(\lambda) \nabla^2 \mathcal{G}(\lambda) - \mathcal{G}^{-2}(\lambda) \nabla \mathcal{G}(\lambda) \nabla \mathcal{G}(\lambda)'.\end{aligned}$$

Therefore, as soon as the value, the gradient, and the Hessian of  $\mathcal{G}(\lambda)$  are available, the gradient and the Hessian of  $\mathcal{B}(\lambda)$  can be easily computed with little extra computational effort. Hence, the computational complexities of computing the gradient and the Hessian of  $\mathcal{B}(\lambda)$  are only slightly higher than those of  $\mathcal{G}(\lambda)$ . What we gain from the extra computational effort is that the barrier function  $\mathcal{B}(\lambda)$  is a self-concordant barrier.

### Self-Concordant Properties

The next proposition shows that barrier function  $\mathcal{B}(\lambda)$  satisfies property (P2) with constant  $c_1 = 1$ .

**Proposition 4.** *The barrier function  $\mathcal{B}(\lambda)$  satisfies*

$$|\nabla \mathcal{B}(\lambda)[h]| \leq \nu (\nabla^2 \mathcal{B}(\lambda)[h, h])^{\frac{1}{2}}, \quad \forall \lambda \in \Omega, \quad \forall h \in \mathbf{R}^n, \quad (39)$$

where  $\nu$  is greater than or equal to 1.

*Proof.* Substituting the expressions of the gradient and the Hessian of  $\mathcal{B}(\lambda)$  into the inequality in (39) and noticing that  $\mathcal{G}(\lambda) > 0 \forall \lambda \in \Omega$ , we see (39) is equivalent to

$$(\nu^2 + 1) \mathcal{G}(\lambda)^{-1} (\nabla \mathcal{G}(\lambda)[h])^2 \leq \nu^2 \nabla^2 \mathcal{G}(\lambda)[h, h], \quad \forall \lambda \in \Omega, \quad \forall h \in \mathbf{R}^n. \quad (40)$$

Condition (40) is satisfied if the matrix

$$\begin{bmatrix} \nabla^2 \mathcal{G}(\lambda) & \nabla \mathcal{G}(\lambda) \\ \nabla \mathcal{G}(\lambda)' & \frac{\nu^2}{\nu^2 + 1} \mathcal{G}(\lambda) \end{bmatrix}$$

is positive definite for all  $\lambda \in \Omega$ . Therefore, what we have to show is that  $\nabla^2 \mathcal{G}(\lambda)[h, h] + 2\nabla \mathcal{G}(\lambda)[h] + \frac{\nu^2}{\nu^2 + 1} \mathcal{G}(\lambda) \geq 0$  for all  $\lambda \in \Omega$  and for all  $h \in \mathbf{R}^n$ . We have

$$\begin{aligned}& \nabla^2 \mathcal{G}(\lambda)[h, h] + 2\nabla \mathcal{G}(\lambda)[h] + \frac{\nu^2}{\nu^2 + 1} \mathcal{G}(\lambda) \\ &= \frac{1}{\pi} \int_{-\infty}^{\infty} \text{tr} \left( 2\mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} - 2\mathbf{H}^{-1} \mathcal{A} \mathbf{H}^{-1} + \frac{\nu^2}{\nu^2 + 1} \mathbf{H}^{-1} \right) \frac{d\omega}{1 + \omega^2} \\ &= \frac{1}{\pi} \int_{-\infty}^{\infty} 2 \|\mathbf{H}^{-\frac{1}{2}} \mathcal{A} \mathbf{H}^{-1} - 0.5 \mathbf{H}^{-\frac{1}{2}}\|_F^2 + \left( \frac{\nu^2}{\nu^2 + 1} - \frac{1}{2} \right) \|\mathbf{H}^{-\frac{1}{2}}\|_F^2 \frac{d\omega}{1 + \omega^2},\end{aligned}$$

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  $\nabla^2 \mathcal{G}(\lambda)[h, h] + 2\nabla \mathcal{G}(\lambda)[h] + \frac{\nu^2}{\nu^2 + 1} \mathcal{G}(\lambda) \geq 0$  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 (P3) also holds for  $\mathcal{B}(\lambda)$ .

**Theorem 2.** *There exists a constant  $c$  such that*

$$|\nabla^3 \mathcal{B}(\lambda)[h, h, h]| \leq c(\nabla^2 \mathcal{B}(\lambda)[h, h])^{\frac{3}{2}} \quad \forall h \in \mathbf{R}^n \quad (41)$$

holds for all  $\lambda \in \Omega$ .

*Proof.* See Appendix. □

Thus,  $\mathcal{B}(\lambda)$  satisfies all three properties (P1) to (P3) and is a strongly self-concordant barrier. However, although we are able to prove that (41) holds, the value of the constant  $c$  is not provided from our proof and yet to be determined. Hence, we are not able to apply Nesterov and Nemirovski's results to construct a polynomial time interior path-following algorithm based on  $\mathcal{B}(\lambda)$ . Nevertheless, the path-following algorithm based on  $\mathcal{B}(\lambda)$  converges globally according to Theorem 1. Furthermore, the following proposition provides a mean to measure global convergence quantitatively.

**Proposition 5.** *Consider optimization problem (1). Let  $\lambda^*(t)$  be the central path generated by barrier function  $\mathcal{B}(\lambda)$  and  $p^*$  be the optimal objective of (1). Then  $c' \lambda^*(t) - p^* \leq 1/t$ .*

*Proof.* This is a corollary of one of Nesterov and Nemirovski's results. See Section 3.2 of [13]. □

## 5 Path-following Algorithms for IQC Optimization Problems

Barrier functions  $\mathcal{G}(\lambda)$  and  $\mathcal{B}(\lambda)$  are used to construct an interior path-following algorithms for solving IQC optimization problems. These algorithms follow the basic principles of the interior path-following method described in Section 3. The algorithm based on  $\mathcal{G}(\lambda)$  is 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

$$\min_{\lambda} \varphi_t(\lambda) := tc' \lambda + \mathcal{G}(\lambda) \quad (42)$$

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 [4]. The algorithm for Phase-I is the same as the algorithm described above.

Since  $\mathcal{G}(\lambda)$  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 Nemirovski does not apply here. In the algorithm, we adopt a heuristic criterion

$$\text{Until } \left( \frac{1}{t} \mathcal{G}(\lambda_U) < \epsilon \right). \quad (43)$$

The idea of this criterion is straightforward: if  $\frac{1}{t} \mathcal{G}(\lambda_U)$  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  $\mathcal{B}(\lambda)$  is the same as the algorithm described above, except the stopping criterion. Instead of criterion (43), we use

$$\text{Until } \left( \frac{1}{t} < \epsilon \right). \quad (44)$$

We have already seen in Proposition 5 that the minimizer of (42) 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 (41)). In our case, since we do not know the value of the self-concordant parameter associated with  $\mathcal{B}(\lambda)$ , 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.

## 6 Comparison with the Conventional Method

In this section, we compare the proposed interior path-following algorithms to the conventional method of solving IQC problems from the point of view of computational complexity.

The conventional approach to solve the IQC optimization problem (1) is to transform it into the equivalent SDP

$$\begin{aligned} & \inf_{P=P', \lambda} c' \lambda, \text{ subj. to} \\ & \begin{bmatrix} PA + A'P & PB \\ B'P & 0 \end{bmatrix} + \begin{bmatrix} Q(\lambda) & F(\lambda) \\ F(\lambda)' & R(\lambda) \end{bmatrix} > 0. \end{aligned} \quad (45)$$

Problem (45) 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 (45) has a well-known strongly  $(n_x + n_r)$ -self-concordant barrier, and the interior point algorithm for solving (45) can be proven to converge in  $\mathcal{O}(\sqrt{n_x + n_r} \log \frac{n_x + n_r}{\epsilon})$  Newton steps [16, 13]. Furthermore, the computational complexity of each Newton step is counted as  $\mathcal{O}(n_x^2 (n_x^2 + n)^2)$ . Therefore, the total complexity is  $\mathcal{O}(\sqrt{n_x + n_r} \log \frac{n_x + n_r}{\epsilon}) \cdot \mathcal{O}(n_x^2 (n_x^2 + n)^2)$ .

In the case of the interior path-following algorithms proposed in this paper, the number of Newton steps required for the algorithms to converge is yet to be determined. However, the complexity of

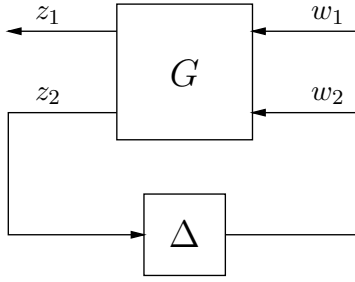


Figure 1: Standard block diagram for robustness analysis.

each Newton step of the algorithms can be estimated as follows: in each Newton step, the algorithms compute a descent direction and perform 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. The major computational burden of evaluating the barrier functions is to solve one Riccati equation in the form of (17) and one Lyapunov equation in the form of (18). The computation complexities of solving (17) and (18) are both  $\mathbf{O}(n_x^3)$ . To compute each entry of the gradient and the Hessian of the barriers, two Lyapunov equations in the form of (18) 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 (45) requires  $\mathbf{O}(n_x^6)$  arithmetic operations, while each Newton step of interior path-following algorithm proposed in Section 5 requires only  $\mathbf{O}(n^2n_x^3)$ . This is why the algorithms proposed in this paper are more efficient. Furthermore, we expect that when the ratio  $n_x/n$  is large enough, the algorithm proposed in Section 5 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 paper 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 algorithm proposed in Section 5. As we will see, these results agree with the expectation from the complexity analysis.

## 7 Examples

**Example 2.** Consider the standard block diagram for robustness analysis in Figure 1. The nominal system  $G$  is linear time-invariant and has a state space representation

$$\begin{aligned} \dot{x} &= Ax + B_1w_1 + B_2w_2, \\ z_1 &= C_1x + D_{11}w_1 + D_{22}w_2, \\ z_2 &= C_2x + D_{21}w_1 + D_{22}w_2, \end{aligned}$$

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 [1].

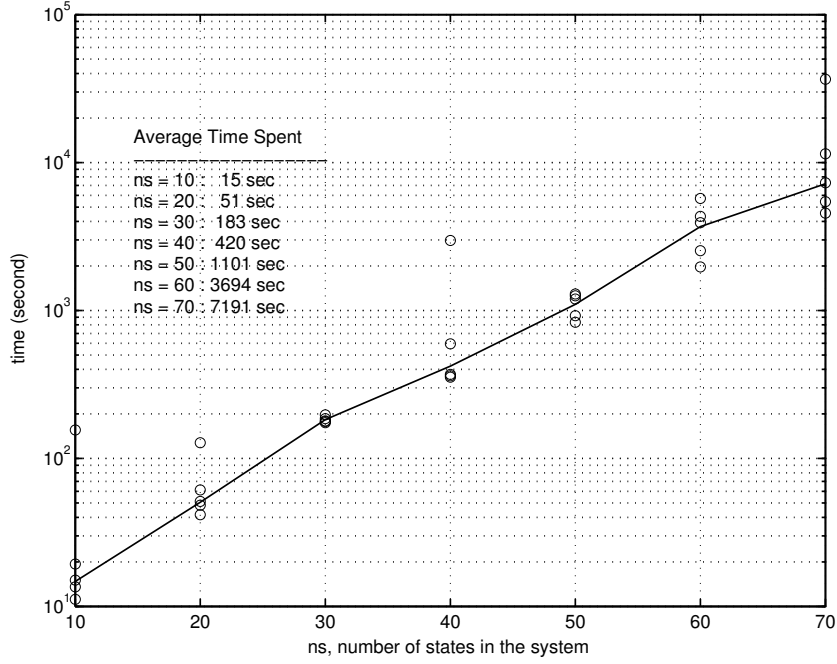


Figure 2: The amount of time that the MATLAB LMI Control Toolbox took to solve SDP (47).

In this example, we would like to compute an upper bound of the  $L_2$ -gain of the system in Figure 1. By the standard IQC analysis, an upper bound of the  $L_2$ -gain can be found by solving

$$\begin{aligned}
 & \inf_{\lambda} \quad \lambda_{n+1}, \\
 & \text{subj to} \quad H(\omega, \lambda) > 0, \quad \forall \omega \in [0, \infty], \\
 & \quad \quad \quad \lambda_i > 0, \quad i = 1, \dots, n+1,
 \end{aligned} \tag{46}$$

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 (46) by transforming the problem into its equivalent SDP formulation and then solving the SDP using the interior point method. The equivalent SDP of problem (46) can be expressed as

$$\begin{aligned}
 & \inf_{P, \lambda} \quad \lambda_{n+1}, \\
 & \text{subj to} \quad S(P, \lambda) > 0, \\
 & \quad \quad \quad P = P', \quad \lambda_i > 0, \quad i = 1, \dots, n+1,
 \end{aligned} \tag{47}$$

where  $P$  is a matrix variable, and

$$S(P, \lambda) := \begin{bmatrix} PA + A'P & PB \\ B'P & 0 \end{bmatrix} + \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}' \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}.$$

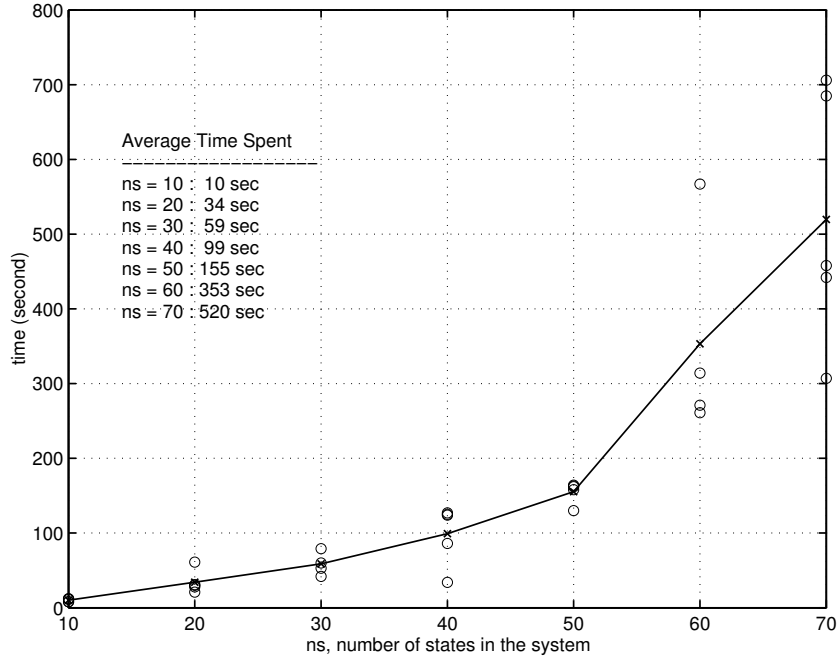


Figure 3: The amount of time that the interior path-following algorithm based on barrier function  $\mathcal{G}(\lambda)$  took to solve problem (46) – all problems were solved in less than 15 minutes. Compared with the Figure 2, we see that the proposed algorithm is significantly faster than the conventional 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.

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 (47) 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 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 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.

Figures 3 and 4 show the amount of time the proposed algorithms spent on solving the randomly generated problems. Compared to Figure 2, the performance of the proposed algorithms 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 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 proposed algorithms spent is about 1/3 of the time which the MATLAB LMI Control Toolbox took. For a problem of 50

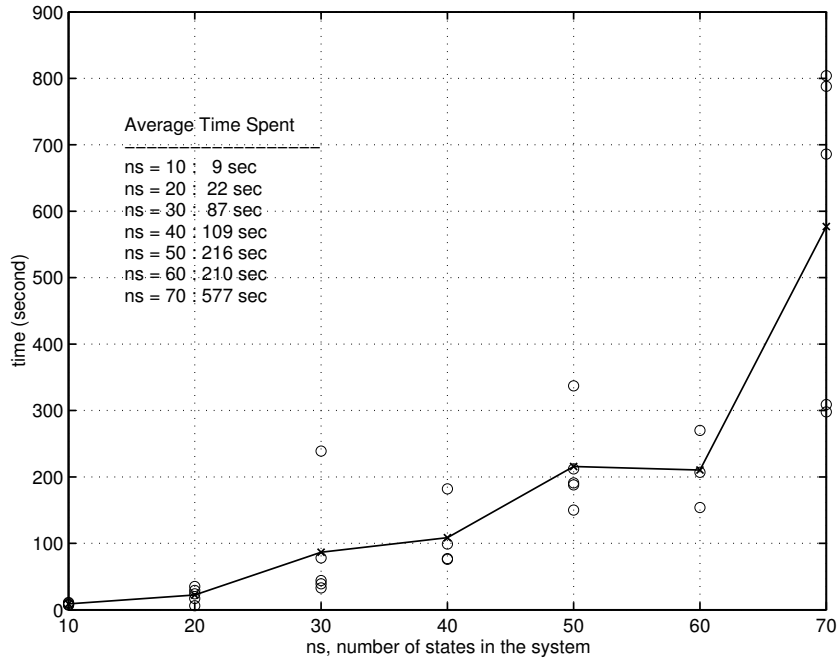


Figure 4: The amount of time that the interior path-following algorithm based on barrier function  $\mathcal{B}(\lambda)$  took to solve problem (46) – all problems were solved in less than 15 minutes. The performance of this algorithm is similar to the performance of that based on barrier function  $\mathcal{G}(\lambda)$ .

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 proposed algorithms are more than 10 times faster than the MATLAB LMI Control Toolbox. This fits our expectation from the complexity analysis.

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 1: Numbers of iterations that the MATLAB LMI Control Toolbox (denoted by “LMI Tool”), the interior path-following algorithm based on  $\mathcal{G}(\lambda)$  (denoted by “IPA-1”), and the interior path-following algorithm based on  $\mathcal{B}(\lambda)$  (denoted by “IPA-2”) took to solve problem (46).

Table 1 shows the average numbers of iterations which the path-following algorithms proposed in this paper 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 or not the proposed algorithms are polynomial-time convergent, the numerical experiments indicate that at least practically the proposed algorithm can work as well as the interior point methods for SDPs.

**Example 3 (Seismic Isolation Control).** This example is adapted from [11]. 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, and the dynamics of the building is governed 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 5, 6, 7, 8, 9, 10, and 11 stories, under the assumption that acceleration sensors are available at each floor of the buildings. The following values of  $m_r$ ,  $c_r$ , and  $k_r$  are used

$$m_r = 44.2 \quad (r = 1, \dots, 10), \quad m_{11} = 54.2,$$

$$k_r (r = 1, \dots, 11) = [91.6 \ 91.6 \ 88.3 \ 89.2 \ 79.1 \ 73.1 \ 66.1 \ 58.0 \ 48.8 \ 38.1 \ 25.5], \quad (48)$$

$$c_r (r = 1, \dots, 11) = [18.3 \ 18.3 \ 17.7 \ 17.8 \ 15.8 \ 14.6 \ 13.2 \ 11.6 \ 9.8 \ 7.6 \ 5.1]. \quad (49)$$

The controller design is based on  $\mathcal{H}_2$  optimization and carried out using MATLAB  $\mu$ -Analysis and Synthesis Toolbox.

In this example, 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 (48) and (49); 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  $\mathbf{L}_2$ -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 1, where input  $w_1 := v$ , output  $z_1 := \ddot{x}_n$ , and output  $z_2$  is a  $6 \times 1$  vector signal defined as

$$z_2' = \left[ \frac{k_1 x_1}{20} \quad \frac{k_2 (x_2 - x_1)}{20} \quad \frac{k_3 (x_3 - x_2)}{20} \quad \frac{c_1 \dot{x}_1}{20} \quad \frac{c_2 (\dot{x}_2 - \dot{x}_1)}{20} \quad \frac{c_3 (\dot{x}_3 - \dot{x}_2)}{20} \right].$$

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  $\mathbf{L}_2$ -gain from  $v$  to  $\ddot{x}$ , we apply standard IQC analysis. The IQC defined by the following quadratic form

$$\int_{-\infty}^{\infty} \begin{bmatrix} \hat{z}_{2i}(j\omega) \\ \hat{w}_{2i}(j\omega) \end{bmatrix}^* \Pi_i(j\omega) \begin{bmatrix} \hat{z}_{2i}(j\omega) \\ \hat{w}_{2i}(j\omega) \end{bmatrix} d\omega \quad (50)$$

is used to characterize the relationship  $w_{2i} = \delta_i z_{2i}$ , where the multiplier  $\Pi_i(j\omega)$  is of the form

$$\Pi_i(j\omega) = \begin{bmatrix} \mathbf{Re}(\lambda_{1i} + \lambda_{2i} \frac{1}{j\omega+1}) & \lambda_{3i} (\frac{1}{j\omega+1} - \frac{1}{-j\omega+1}) \\ \lambda_{3i} (\frac{1}{-j\omega+1} - \frac{1}{j\omega+1}) & -\mathbf{Re}(\lambda_{1i} + \lambda_{2i} \frac{1}{j\omega+1}) \end{bmatrix}, \quad (51)$$

	$n = 5$			$n = 6$			$n = 7$		
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 2: Results of solving the  $L_2$ -gain estimation problem in Example 3 using MATLAB LMI Control Toolbox and the proposed path-following algorithms. “LMI Tool“ denotes MATLAB LMI Control Toolbox. “IPA-1“ denotes the interior path-following algorithm based on  $\mathcal{G}(\lambda)$ . “IPA-2“ denotes the interior path-following algorithm based on  $\mathcal{B}(\lambda)$ . The numbers in the column “var“ indicate the number of decision variables in a problem.

and parameters  $\lambda_{1i}, \lambda_{2i}$  must satisfy  $\mathbf{Re}(\lambda_{1i} + \lambda_{2i} \frac{1}{j\omega+1}) > 0 \forall \omega$ . 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. The problems were solved using the MATLAB LMI Control Toolbox and the interior path-following algorithm proposed in this paper. The results are listed in Table 2.

Table 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.

## 8 Summary

In this paper, we proposed new barrier functions 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, the interior path-following algorithms proposed in this paper can solve standard IQC optimization problems in a more efficient fashion. The proposed algorithms were tested on a number of numerical examples. The results indicate that the proposed algorithms work well and significantly outperform the conventional method for solving IQC optimization problems, especially when the number of states in an IQC problem is very large.

## Acknowledgment

The content of this paper is a part of author's thesis work, which was under supervision of Prof. Alexander Megretski of the Department of Electrical Engineering at M.I.T.. The author would also like to thank Dr. Ulf Jönsson at KTH, Stockholm, for his valuable comments on this work.

## Appendix: Proof of Theorem 2

We prove Theorem 2 in this section. The following two lemmas will be used in the proof.

**Lemma 3.** *Let  $\omega_0$  be a real number and  $\beta \geq 2$  be a positive even integer. Let*

$$G(t, \omega) = \frac{r_0(t) + r_1(t)(\omega - \omega_0)^\beta + \cdots + r_{n-1}(t)(\omega - \omega_0)^{\beta(n-1)}}{(p(t) + (\omega - \omega_0)^\beta q(t))^n}, \quad (52)$$

where  $p(t), q(t), r_i(t), i = 0, \dots, n-1$  are polynomial functions in  $t$ . Now consider the integration

$$F(t) = \int_{\omega_0 - \epsilon}^{\omega_0 + \epsilon} G(t, \omega) d\omega. \quad (53)$$

Then, for any  $t^* > 0$ , there exists a sufficiently small  $\epsilon$  such that

$$F(t) \approx c(t) \cdot r_0(t) p(t)^{\frac{1}{\beta} - n} q(t)^{-\frac{1}{\beta}} \quad (54)$$

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 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

$$F(t) \approx t^{-n} s_0(t), \quad \dot{F}(t) \approx t^{-n-1} s_1(t), \quad (55)$$

$$\ddot{F}(t) \approx t^{-n-2} s_2(t), \quad \dddot{F}(t) \approx t^{-n-3} s_3(t), \quad (56)$$

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 (52). 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 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 (55) 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 3,

$$\dot{F}(t) \approx c_1(t)d_{10}(t)t^{-2\alpha+\frac{\alpha}{\beta}}p(t)^{\frac{1}{\beta}-2}q(t)^{-\frac{1}{\beta}}. \quad (57)$$

Substituting the expression of  $d_{10}(t)$  into (57), 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 (55).  $\ddot{F}(t) \approx t^{-n-2}s_2(t)$  and  $\dddot{F}(t) \approx t^{-n-3}s_3(t)$  are obtained in a similar fashion.  $\square$

## Proof of Theorem 2

We are now ready to prove Theorem 2. Recall that

$$\mathcal{B}(\lambda) = \log(\mathcal{G}(\lambda)), \quad \text{and} \quad \mathcal{G}(\lambda) = \left( \frac{1}{\pi} \int_{-\infty}^{\infty} \mathbf{tr}(\mathbf{H}(\omega, \lambda)^{-1}) \frac{d\omega}{1 + \omega^2} \right).$$

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  $F(t) : T \rightarrow \mathbf{R} := \mathcal{G}(\lambda + th)$  and  $E(t) : T \rightarrow \mathbf{R} := \mathcal{B}(\lambda + th) = \log(F(t))$ . Let  $\gamma(t) := \frac{\ddot{E}(t)^2}{E(t)^3}$ . To show  $\mathcal{B}(\lambda)$  satisfies (41), we have to prove that

$$\sup_{t \in T} \gamma(t) < \infty, \quad (58)$$

i.e.,  $\gamma(t)$  is bounded above for all  $t \in T$ . Since  $\gamma(t)$  is a continuous function, therefore (58) is true if  $\gamma(t)$  is finite as  $t$  approaches any boundary point of  $T$ .

Note that

$$\frac{1}{\pi} \frac{1}{(1 + \omega^2)} \mathbf{tr}(\mathbf{H}(\omega, \lambda + th)^{-1}) = \frac{r(\omega, t)}{s(\omega, t)},$$

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

$$\frac{r(\omega, t)}{s(\omega, t)} \rightarrow G_k(t) := \frac{r_k(t)}{t^{\alpha_k} p_k(t) + (\omega - \omega_k)^{\beta_k} q_k(t)} \quad (59)$$

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

$$\begin{aligned} F(t) &= \int_{-\infty}^{\infty} \frac{1}{\pi} \mathbf{tr}(\mathbf{H}(\omega, \lambda + th)^{-1}) \frac{d\omega}{(1 + \omega^2)} \\ &= \sum_{k=1}^n \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} \frac{r(\omega, t)}{s(\omega, t)} d\omega + \int_{[-\infty, \infty] \setminus \Sigma} \frac{r(\omega, t)}{s(\omega, t)} d\omega. \end{aligned}$$

Since  $s(\omega, t)$  is bounded away from 0 for all  $t \in T$  and for all  $\omega \in [-\infty, \infty] \setminus \Sigma$ , therefore

$$\sup_{t \in T} \int_{[-\infty, \infty] \setminus \Sigma} \frac{1}{\pi} \mathbf{tr}(\mathbf{H}(\omega, \lambda + th)^{-1}) \frac{d\omega}{(1 + \omega^2)} \leq M, \quad (60)$$

where  $M$  is a constant depending on  $\epsilon$ . Thus, by (59) and (60), we see that for every  $t \in T$ ,

$$F(t) = \sum_{k=1}^n \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} G_k(\omega, t) d\omega + \mathbf{O}(1), \quad (61)$$

where  $\mathbf{O}(1)$  depends on  $\epsilon$  and  $t$ . Let us consider the  $\kappa^{\text{th}}$  derivative of  $F(t)$ . We have

$$\begin{aligned} \frac{d^\kappa}{dt^\kappa} F(t) &= \int_{-\infty}^{\infty} \frac{1}{\pi} \frac{d^\kappa}{dt^\kappa} \mathbf{tr}(\mathbf{H}(\omega, \lambda + th)^{-1}) \frac{d\omega}{(1 + \omega^2)} \\ &= \sum_{k=1}^n \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} \frac{d^\kappa}{dt^\kappa} \left( \frac{r(\omega, t)}{s(\omega, t)} \right) d\omega + \int_{[-\infty, \infty] \setminus \Sigma} \frac{d^\kappa}{dt^\kappa} \left( \frac{r(\omega, t)}{s(\omega, t)} \right) d\omega \\ &= \sum_{k=1}^n \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} \frac{r_\kappa(\omega, t)}{s(\omega, t)^\kappa} d\omega + \int_{[-\infty, \infty] \setminus \Sigma} \frac{r_\kappa(\omega, t)}{s(\omega, t)^\kappa} d\omega, \end{aligned}$$

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

$$\frac{r_\kappa(\omega, t)}{s(\omega, t)^\kappa} \rightarrow \frac{d^\kappa}{dt^\kappa} G_k(\omega, t) \quad (62)$$

as  $\omega \rightarrow \omega_k$ , thus, we conclude that

$$\frac{d^n}{dt^n} F(t) = \sum_{k=1}^n \int_{\omega_k - \epsilon}^{\omega_k + \epsilon} \frac{d^\kappa}{dt^\kappa} G_k(\omega, t) d\omega + \mathbf{O}(1). \quad (63)$$

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 4, we conclude that

$$F_k(t) \approx t^{-m_k} s_{k0}(t), \quad (64)$$

$$\dot{F}_k(t) \approx t^{-m_k-1} s_{k1}(t), \quad (65)$$

$$\ddot{F}_k(t) \approx t^{-m_k-2} s_{k2}(t), \quad (66)$$

$$\dddot{F}_k(t) \approx t^{-m_k-3} s_{k3}(t), \quad (67)$$

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 (61), and (63) to (67) imply that

$$F(t) \approx t^{-m_1} s_0(t), \quad (68)$$

$$\dot{F}(t) \approx t^{-m_1-1} s_1(t), \quad (69)$$

$$\ddot{F}(t) \approx t^{-m_1-2} s_2(t), \quad (70)$$

$$\dddot{F}(t) \approx t^{-m_1-3} s_3(t). \quad (71)$$

Again, in (68) to (71),  $s_i(t)$ ,  $i = 0, \dots, 3$ , 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

$$\gamma(t) := \frac{\ddot{E}(t)^2}{\dot{E}(t)^3} = \frac{(F(t)^2 \ddot{F}(t) - 3F(t) \dot{F}(t) \ddot{F}(t) + 2\dot{F}(t)^3)^2}{(F(t) \ddot{F}(t) - \dot{F}(t)^2)^3}. \quad (72)$$

Substituting (68) to (71) into (72), 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

$$\sup_{t \in T} \frac{\ddot{E}(t)^2}{\dot{E}(t)^3}$$

is bounded. This in turn implies  $\mathcal{B}(\lambda)$  satisfies (41).

## References

- [1] G.J. Balas, J.C. Doyle, K. Glover, A. Packard, and R. Smith.  *$\mu$ -Analysis and Synthesis Toolbox*. The Math Works Inc, 1993.
- [2] D. P. Bertsekas. *Nonlinear Programming*. Athen Scientific, Massachusetts, 1995.
- [3] 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.
- [4] S. Boyd and L. Vandenberghe. *Lecture Notes on Convex Optimization*.
- [5] 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.
- [6] 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.
- [7] A. Hansson and L. Vandenberghe. A primal-dual potential reduction method for integral quadratic constraints. In *Proceedings of the 2001 American Control Conference*, pages 3013–3017, Arlington, VA, June 2001.
- [8] 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.
- [9] Chung-Yao Kao. *Efficient Computational Algorithms for Robustness Analysis*. PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA, 2002.
- [10] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.
- [11] A. Megretski. *Lecture Notes on Multivariable Control Systems*. MIT, Cambridge, MA, 2000.
- [12] A. Megretski and A. Rantzer. System analysis via Integral Quadratic Constraints. *IEEE Transactions on Automatic Control*, 42(6):819–830, June 1997.
- [13] Y. Nesterov and A. Nemirovski. *Interior Point Polynomial Methods in Convex Programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, 1994.
- [14] 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.
- [15] 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.
- [16] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, March 1996.
- [17] 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.

- [18] J. C. Willems. Dissipative dynamical system, Part I: General theory. *Archive for Rational Mechanics and Analysis*, 45:321–353, 1972.
- [19] V. A. Yakubovich. Frequency conditions for the absolute stability of control systems with several nonlinear or linear nonstationary blocks. *Avtomatika i Telemekhanika*, 6:5–30, June 1967.
- [20] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, New Jersey, 1996.