IQC analysis and synthesis via nonsmooth optimization

Pierre Apkarian * Dominikus Noll †

Abstract

Integral quadratic constraints (IQCs) are used in system theory to model nonlinear phenomena within the framework of linear feedback control. IQC theory addresses parametric robustness, saturation effects, sector nonlinearities, passivity, and much else. In IQC analysis specially structured linear matrix inequalities (LMIs) arise and are currently addressed by structure exploiting LMI solvers. Controller synthesis under IQC constraints is non-convex and much harder and has been attempted sporadically by global optimization techniques such as branch-and-bound, cutting plane or *D*-*K* type coordinate descent ideas. Here we revisit IQC theory and propose a completely different algorithmic solution based on local and nonsmooth optimization methods. This is less ambitious than global methods, but is very promising in practice. Our approach, while aiming high at IQC synthesis, offers new answers even for IQC analysis, because we optimize without Lyapunov variables. For high order systems this leads to a significant reduction of the number of unknowns.

Keywords: Nonsmooth optimization, IQC theory, robust control, parametric uncertainty, robustness analysis, structured controllers, NP-hard problems.

Notation

Let $\mathbb{R}^{n\times m}$ be the space of $n\times m$ matrices, equipped with the corresponding scalar product $\langle X, Y\rangle = \mathrm{Tr}(X^TY)$, where X^T is the transpose of the matrix X, $\mathrm{Tr}\,(X)$ its trace. For an arbitrary square matrix M, $\mathrm{tril}(M)$ is used to denote a matrix with the same non strict lower triangle as M and zeros elsewhere. For complex matrices X^H stands for its transconjugate. For Hermitian or symmetric matrices, $X \succ Y$ means that $X - Y$ is positive definite, $X \succeq Y$ that $X - Y$ is positive semi-definite. We use the symbol λ_1 to denote the maximum eigenvalue of a symmetric or Hermitian matrix. We use concepts from nonsmooth analysis covered by [11]. For a locally Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$, $\partial f(x)$ denotes its Clarke subdifferential at x.

*Corresponding author, ONERA-CERT, Centre d'études et de recherches de Toulouse, Control System Department, 2 av. Edouard Belin, 31055 Toulouse, France - and - Institut de Mathématiques, Université Paul Sabatier, Toulouse, France - Email: apkarian@cert.fr - Tel: +33 5.62.25.27.84 - Fax: +33 5.62.25.27.64.

†Université Paul Sabatier, Institut de Mathématiques, 118, route de Narbonne, 31062 Toulouse, France - Email: noll@mip.ups-tlse.fr - Tel: +33 5.61.55.86.22 - Fax: +33 5.61.55.83.85.

1 Introduction

In many control applications stability and performance have to be guaranteed in the presence of uncertainties or distortions introduced by nonlinear system components. These are due to the fact that model equations are not perfectly known or do not fully reflect reality. The present paper discusses integral quadratic constraints (IQCs) as an appropriate response to these uncertainties and proposes an efficient algorithmic framework for analysis and synthesis under IQC constraints. To begin with, we consider the robust control problem of an uncertain plant in LFT (Linear Fractional Transformation) form:

$$\begin{array}{rcl} \begin{bmatrix} \dot{x} \\ z_\Delta \\ z \\ y \end{bmatrix} & = & \begin{bmatrix} A & B_\Delta & B_1 & B_2 \\ C_\Delta & D_{\Delta\Delta} & D_{\Delta 1} & D_{\Delta 2} \\ C_1 & D_{1\Delta} & D_{11} & D_{12} \\ C_2 & D_{2\Delta} & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ w_\Delta \\ w \\ u \end{bmatrix} \\ w_\Delta & = & \Delta(z_\Delta)\,, \end{array} \tag{1}$$

where Δ is an uncertain continuous nonlinear operator. Here, we are using the following notation: $x \in \mathbb{R}^n$ is the state of the system, $u \in \mathbb{R}^{m_2}$ the control, $w \in \mathbb{R}^{m_1}$ the exogenous input, $z \in \mathbb{R}^{p_1}$ the performance variable, $y \in \mathbb{R}^{p_2}$ the measurement and $(w_\Delta, z_\Delta) \in \mathbb{R}^{m_\Delta} \times \mathbb{R}^{p_\Delta}$ represents the uncertainty channel. We assume that all admissible Δ satisfy an IQC defined by $\Pi_\Delta = \Pi_\Delta^H$, that is,

$$\int_{-\infty}^{+\infty} \begin{bmatrix} z_\Delta(j\omega) \\ \Delta(z_\Delta)(j\omega) \end{bmatrix}^H \Pi_\Delta \begin{bmatrix} z_\Delta(j\omega) \\ \Delta(z_\Delta)(j\omega) \end{bmatrix} d\omega \geq 0\,, \tag{2}$$

for all square integrable signals z_Δ. We further assume that performance is expressed by the channel (w, z) through the IQC defined by $\Pi_p = \Pi_p^H$

$$\int_{-\infty}^{+\infty} \begin{bmatrix} z(j\omega) \\ w(j\omega) \end{bmatrix}^H \Pi_p \begin{bmatrix} z(j\omega) \\ w(j\omega) \end{bmatrix} d\omega \leq 0 \tag{3}$$

for all square integrable signals w. Note that Π_Δ and Π_p are called multipliers and are for the moment restricted to constant Hermitian matrices for a moment. Extensions to general dynamic multipliers $\Pi(j\omega)$ are considered in Section 4.4.

The robust control problem now requires finding a linear time-invariant output feedback controller

$$K(s) = C_K(sI - A_K)^{-1}B_K + D_K, \qquad A_K \in \mathbb{R}^{k\times k}, \tag{4}$$

for the uncertain plant (1), such that the following conditions are satisfied:

(i) The closed-loop system (1), (4) is internally stable for all admissible Δ in (2).

(ii) Performance condition (3) holds for all admissible Δ in (2).

Here k is the order of the controller, and the possibility $k = 0$ of a static controller $K(s) = D_K$ is included. Let us introduce the closed-loop transfer matrix

$$T(s,K) = \begin{bmatrix} T_{\Delta\Delta}(s,K) & T_{\Delta w}(s,K) \\ T_{z\Delta}(s,K) & T_{zw}(s,K) \end{bmatrix} := \left\{ \begin{array}{rcl} \dot{x_{c\ell}} & = & \mathcal{A}(K)x_{c\ell} + \mathcal{B}(K) \begin{bmatrix} w_\Delta \\ w \end{bmatrix} \\ \begin{bmatrix} z_\Delta \\ z \end{bmatrix} & = & \mathcal{C}(K)x_{c\ell} + \mathcal{D}(K) \begin{bmatrix} w_\Delta \\ w \end{bmatrix} \end{array} \right., \tag{5}$$

where state-space data $\mathcal{A}(K)$, $\mathcal{B}(K)$, $\mathcal{C}(K)$ and $\mathcal{D}(K)$ determine the closed-loop system (1) and (4) with the Δ-loop $w_\Delta = \Delta(z_\Delta)$ still open. Here closed-loop data are given as:

$$\begin{array}{c}\mathcal{A}(K) := \mathcal{A} + \mathcal{B}_2 K \mathcal{C}_2,\ \mathcal{B}(K) := \mathcal{B}_1 + \mathcal{B}_2 K \mathcal{D}_{21},\ \mathcal{C}(K) := \mathcal{C}_1 + \mathcal{D}_{12} K \mathcal{C}_2, \\ \mathcal{D}(K) := \mathcal{D}_{11} + \mathcal{D}_{12} K \mathcal{D}_{21}\,, \end{array} \tag{6}$$

with

$$\begin{array}{lll} \mathcal{A} := \begin{bmatrix} A & 0 \\ 0 & 0_k \end{bmatrix} & \mathcal{B}_1 := \begin{bmatrix} B_\Delta & B_1 \\ 0 & 0 \end{bmatrix}, & \mathcal{B}_2 := \begin{bmatrix} 0 & B_2 \\ I_k & 0 \end{bmatrix} \\ \mathcal{C}_1 := \begin{bmatrix} C_\Delta & 0 \\ C_1 & 0 \end{bmatrix}, & \mathcal{D}_{11} := \begin{bmatrix} D_{\Delta\Delta} & D_{\Delta 1} \\ D_{1\Delta} & D_{11} \end{bmatrix} & \mathcal{D}_{12} := \begin{bmatrix} 0 & D_{\Delta 2} \\ 0 & D_{12} \end{bmatrix}, \\ \mathcal{C}_2 := \begin{bmatrix} 0 & I_k \\ C_2 & 0 \end{bmatrix}, & \mathcal{D}_{21} := \begin{bmatrix} 0 & 0 \\ D_{2\Delta} & D_{21} \end{bmatrix} & K := \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \end{array} \tag{7}$$

Then we have the following fundamental fact, see [17, 19] for details.

Theorem 1.1 *Suppose K is closed-loop stabilizing, i.e. $\mathcal{A}(K)$ is Hurwitz. Then the robust performance conditions (i) and (ii) hold for all Δ satisfying the IQC defined by Π_Δ provided the following frequency domain inequality (FDI) is satisfied:*

$$F(K, \Pi; j\omega) := \left[\frac{T(j\omega, K)}{I}\right]^H \Pi \left[\frac{T(j\omega, K)}{I}\right] \prec 0, \quad \forall \omega \in [0, \infty] \tag{8}$$

where

$$\Pi := \left[\begin{array}{c|c} \Pi_{11} & \Pi_{12} \\ \hline \Pi_{12}^H & \Pi_{22} \end{array}\right] := \left[\begin{array}{cc|cc} \Pi_{\Delta,11} & 0 & \Pi_{\Delta,12} & 0 \\ 0 & \Pi_{p,11}^H & 0 & \Pi_{p,12} \\ \hline \Pi_{\Delta,12}^H & 0 & \Pi_{\Delta,22} & 0 \\ 0 & \Pi_{p,12}^H & 0 & \Pi_{p,22} \end{array}\right].$$

Inequality (8) is known as the robust performance FDI, see [19]. It strongly suggests introducing the nonsmooth function

$$\begin{aligned} f(K, \Pi) &:= \lambda_{1,\infty}\left(\left[\frac{T(j\omega, K)}{I}\right]^H \Pi \left[\frac{T(j\omega, K)}{I}\right]\right) \\ &:= \max_{\omega \in [0, \infty]} \lambda_1 \left(\left[\frac{T(j\omega, K)}{I}\right]^H \Pi \left[\frac{T(j\omega, K)}{I}\right]\right), \end{aligned} \tag{9}$$

and considering the optimization program

$$\begin{array}{ll} \text{minimize} & f(K, \Pi) \\ \text{subject to} & \Pi \in \boldsymbol{\Pi} \\ & K \text{ is closed-loop stabilizing} \end{array} \tag{10}$$

which is then minimized until a value $f(K, \Pi) < 0$ is found. Notice that $\boldsymbol{\Pi}$ denotes a convex cone of appropriate multipliers Π describing the uncertainty Δ and will be specified later.

We will now address the following questions. How to compute the function value of f? How to compute Clarke subgradients of f? And finally, how to generate descent steps in order to decrease the value of f below 0? Notice that our approach to minimizing f until a negative value occurs is based on a local optimization paradigm. In consequence, we may occasionally end up with a local minimum of (10) whose value is ≥ 0, meaning failure to solve the control problem. In this case the method has to be restarted with a different initial seed.

A special case of program (10) is robustness analysis, where K is held fixed. The question is then whether the closed loop system achieves the robust performance Π_p, that is, whether (i) and (ii) holds uniformly for all admissible Δ. This problem is easier, being convex in the decision variable Π_Δ. Currently analysis problems are solved by tailored interior point methods for LMIs arising from the Kalman-Yakubovich-Popov lemma [23, 13]. Even in that case we propose to change strategy and proceed via (10), because this avoids Lyapunov variables. The reduction in the number of unknowns may be dramatic for systems with large state dimension n.

2 Computing $\lambda_{1,\infty}$

Introducing the semi-infinite objective function $f(K,\Pi)$ avoids the use of Lyapunov variables, but seems to pose a new major problem. Namely, program (10) is now semi-infinite, and such programs are often difficult since discretization has to be used, which leads back to a large number of unknowns. Fortunately, our situation is different, because there is an efficient way to compute the function value $f(x)$ for a given datum $x = (K,\Pi)$.

During the following we shall write $F(K,\Pi;j\omega)$ for the FDI in (8), so that $f(K,\Pi) = \max_{\omega\in[0,\infty]} \lambda_1\left(F(K,\Pi;j\omega)\right)$. We start by explaining how the function value is computed by an iterative procedure introduced by Boyd *et al.* [8, 7].

Note first that the estimate $f(K,\Pi) < \lambda$ is equivalent to the following frequency domain test:

$$\begin{bmatrix} (j\omega I - \mathcal{A}(K))^{-1}\mathcal{B}(K) \\ I \end{bmatrix}^H \Psi \begin{bmatrix} (j\omega I - \mathcal{A}(K))^{-1}\mathcal{B}(K) \\ I \end{bmatrix} \prec 0, \quad \forall \omega \in [0,\infty]\,,$$

where

$$\Psi := \begin{bmatrix} \mathcal{C}(K) & \mathcal{D}(K) \\ 0 & I \end{bmatrix}^H \begin{bmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{12}^H & \Pi_{22} - \lambda I \end{bmatrix} \begin{bmatrix} \mathcal{C}(K) & \mathcal{D}(K) \\ 0 & I \end{bmatrix}.$$

We can now infer from spectral factorization theory [25, p. 350] that $f(K,\Pi) < \lambda$ holds if and only if $\Psi_{22} \prec 0$ and the matrix

$$\begin{bmatrix} \mathcal{A}(K) - \mathcal{B}(K)\Psi_{22}^{-1}\Psi_{12}^H & -\mathcal{B}(K)\Psi_{22}^{-1}\mathcal{B}(K)^T \\ -(\Psi_{11} - \Psi_{12}\Psi_{22}^{-1}\Psi_{12}^H) & -(\mathcal{A}(K) - \mathcal{B}(K)\Psi_{22}^{-1}\Psi_{12}^H)^H \end{bmatrix}$$

has no eigenvalue on $j\mathbb{R}^+$. Similarly to the computation of the H_∞ norm, this important result can be exploited to construct a quadratically convergent algorithm to compute $f(K,\Pi)$ [10]. As our testing in [2, 3] shows, the function value $f(K,\Pi)$ can be computed very efficiently even for high order systems. The algorithm from [10] not only computes the function value, it also provides the set of active frequencies:

Definition. The set $\Omega(K,\Pi) = \{\omega \in [0,\infty] : \lambda_1\left(F(K,\Pi;j\omega)\right) = f(K,\Pi)\}$ is called the set of active frequencies at $x = (K,\Pi)$.

This information will be crucial during the next sections.

3 Ingredients from nonsmooth analysis

In order to prepare our nonsmooth descent technique to minimize $f(K,\Pi)$, we need to show how to compute derivative information for f. To this aim, we shall repeatedly invoke the concept of active frequencies defined above. The following can be found in [7, 6]:

Lemma 3.1 *For fixed K,Π, the set $\Omega(K,\Pi)$ of active frequencies is either finite, or $\Omega(K,\Pi) = [0,\infty]$.*

The following result is useful for computing Clarke subgradients of the nonsmooth $f = \lambda_{1,\infty} \circ F$.

Lemma 3.2 *$f = \lambda_{1,\infty} \circ F$ is regular in the sense of Clarke [11].*

Proof. $f = \lambda_{1,\infty} \circ F$ is a composite function of $\lambda_{1,\infty}$, which is nonsmooth but convex, and the smooth nonlinear operator F, mapping the space $\mathbb{R}^N$ with $N = \dim(K) + \dim(\Pi)$ to the infinite dimensional space $C(j[0,\infty],\mathbb{H})$ of continuous functions $j[0,\infty] \to \mathbb{H}$, where $\mathbb{H}$ is the space of Hermitian matrices of appropriate dimension. f is therefore regular in the sense of Clarke [11]. $\square$

These Lemmas make it possible to give a full description of the subdifferential of f. We start by characterizing the subdifferential $\partial\lambda_{1,\infty}(\pi)$ at a given $\pi \in C(j[0,\infty],\mathbb{H})$.

Proposition 3.3 *Let $\pi \in C(j[0,\infty],\mathbb{H})$, and suppose the set $\Omega(\pi)$ of active frequencies at π:*

$$\Omega(\pi) := \{\omega \in [0,\infty] : \lambda_{1,\infty}(\pi) = \lambda_1(\pi(j\omega))\}$$

is finite. For every active frequency $\omega \in \Omega(\pi)$, let Q_ω denote a matrix whose columns form an orthogonal basis of the eigenspace of $\pi(j\omega)$ associated with the largest eigenvalue $\lambda_1(\pi(j\omega))$ of $\pi(j\omega)$. Then the subdifferential $\partial\lambda_{1,\infty}(\pi)$ of the mapping $\lambda_{1,\infty}$ at $\pi \in C(j[0,\infty],\mathbb{H})$ is the set of all linear functionals $\Phi_Y \in C(j[0,\infty],\mathbb{H})^*$ of the form

$$\Phi_Y(\mu) = \sum_{\omega\in\Omega(\pi)} \operatorname{Tr}\left(Q_\omega Y_\omega Q_\omega^H \mu(j\omega)\right), \qquad \mu \in C(j[0,\infty],\mathbb{H}), \tag{11}$$

indexed by the family $Y = (Y_\omega)_{\omega\in\Omega(\pi)}$, where $Y_\omega = Y_\omega^H \succeq 0$ and $\sum_{\omega\in\Omega(\pi)} \operatorname{Tr}(Y_\omega) = 1$.

Proof: This is established using subdifferential formulas for λ_1 and the convex hull rule for max functions. The reader is referred to [16, 11] for details. $\square$

Our next step is as follows. Given the subdifferential of $\lambda_{1,\infty}$ at $\pi = F(K,\Pi) \in C(j[0,\infty],\mathbb{H})$, we obtain the subdifferential of f at $x = (K,\Pi)$ using the chain rule

$$\partial f(K,\Pi) = F'(K,\Pi)^* \partial\lambda_{1,\infty}(\pi),$$

where $F'(K,\Pi)$ is the Frechet derivative of F at (K,Π), and $F'(K,\Pi)^*$ its adjoint, which we now need to compute. This may seem arduous at first, since the Banach space dual $C(j[0,\infty],\mathbb{H})^*$ of $C(j[0,\infty],\mathbb{H})$ does not have an easy to manage representation. Fortunately, we only need to know the action of the adjoint $F'(K,\Pi)^*$ on functionals of the special form Φ_Y in (11), and this is easily found. Indeed, the definition of an adjoint gives

$$\langle F'(K,\Pi)(\delta K,\delta\Pi), \Phi_Y\rangle = \langle(\delta K,\delta\Pi), F'(K,\Pi)^*(\Phi_Y)\rangle,$$

where the right hand side is the standard scalar product in a suitable matrix space. Put differently, writing $F'(K,\Pi)^*(\Phi_Y) = (\Lambda_Y, \Sigma_Y)$, where Λ_Y is a matrix compatible with K and Σ_Y a matrix compatible with Π, we have

$$\langle (\delta K, \delta \Pi), (\Lambda_Y, \Sigma_Y) \rangle = \mathrm{Tr}(\delta K^T \Lambda_Y) + \mathrm{Tr}(\delta \Pi^H \Sigma_Y).$$

In order to pursue, we need the Frechet derivative $F'(K,\Pi)$. For simplification purposes, we introduce the notations

$$\begin{bmatrix} T(K,s) & G_{12}(K,s) \\ G_{21}(K,s) & \star \end{bmatrix} := \begin{bmatrix} \mathcal{C}(K) \\ \mathcal{C}_2 \end{bmatrix} (sI - \mathcal{A}(K))^{-1} \begin{bmatrix} \mathcal{B}(K) & \mathcal{B}_2 \end{bmatrix} + \begin{bmatrix} \mathcal{D}(K) & \mathcal{D}_{12} \\ \mathcal{D}_{21} & \star \end{bmatrix}.$$

This leads to the following representation of F':

$$\begin{aligned} F'(K,\Pi)(\delta K, \delta \Pi) &= (G_{12}\delta K G_{21})^H \Pi_{11} T(K) + T(K)^H \Pi_{11} G_{12} \delta K G_{21} + \Pi_{12}^H G_{12} \delta K G_{21} \\ &+ (G_{12}\delta K G_{21})^H \Pi_{12} + \begin{bmatrix} T(K) \\ I \end{bmatrix}^H \delta \Pi \begin{bmatrix} T(K) \\ I \end{bmatrix} \end{aligned} \qquad (12)$$

where the dependence on $j\omega$ has been omitted for simplicity.

With $\pi = F(K,\Pi)$ and $\mu = F'(K,\Pi)(\delta K, \delta \Pi)$, we use formula (11) to match coefficients in $\Phi_Y(\mu) = \mathrm{Tr}(\delta K^T \Lambda_Y) + \mathrm{Tr}(\delta \Pi^H \Sigma_Y)$. This gives

$$\Lambda_Y = 2 \sum_{\omega \in \Omega(\pi)} \mathrm{Re}\, \left(G_{21}(j\omega) Q_\omega Y_\omega Q_\omega^H \left(T(j\omega, K)^H \Pi_{11} + \Pi_{12}^H \right) G_{12}(j\omega) \right)^T \qquad (13)$$

and

$$\Sigma_Y = \sum_{\omega \in \Omega(\pi)} \begin{bmatrix} T(j\omega, K) \\ I \end{bmatrix} Q_\omega Y_\omega Q_\omega^H \begin{bmatrix} T(j\omega, K) \\ I \end{bmatrix}^H . \qquad (14)$$

We sum up our findings in the following

Theorem 3.4 *Consider a stabilizing controller K, i.e $\mathcal{A}(K)$ is Hurwitz, and a multiplier Π. Assume the set of active frequencies $\Omega(K,\Pi)$ for the FDI in (8) is finite. Then the Clarke subdifferential $\partial f(K,\Pi)$ of f at (K,Π) is the set of subgradients $\Big\{(\Lambda_Y, \Sigma_Y) : Y = (Y_\omega)_{\omega \in \Omega(K,\Pi)}$, $Y_\omega = Y_\omega^H \succeq 0, \sum_{\omega \in \Omega(K,\Pi)} \mathrm{Tr}\,(Y_\omega) = 1 \Big\}$, where Λ_Y is given by (13), Σ_Y by (14).*

4 Applications

Note that the results in the previous section naturally extend to a wide variety of IQCs, including those with dynamic multipliers, as long as f has a composite structure $\lambda_{1,\infty} \circ F(K,\Pi)$ where Π gathers the multiplier or scaling, the latter represented in a suitable finite basis. It is possible to extend the proposed framework to several synthesis FDIs, because the maximum of a finite family of FDIs $\lambda_{1,\infty} \circ F_i(K,\Pi) < 0$, $i = 1,\ldots,q$ can be written as a single FDI $\lambda_{1,\infty} \circ \mathrm{diag}\,(F_1(K,\Pi),\ldots,F_q(K,\Pi)) < 0$.

In IQC analysis the composite function $f = \lambda_{1,\infty} \circ F$ is convex as a function of Π alone and its subdifferential $\partial f(\Pi)$ is the usual subdifferential of convex analysis [16]. Subgradient information is again covered by (14), and could in principle be used to find global linear lower bounds for FDI constraints via cutting planes [10], but we do not follow this route in the present work.

In synthesis, multipliers and controller variables are updated simultaneously until satisfaction of the FDI in (8). In accordance with [15], we advocate not to use D-K type methods, where K and Π are updated alternatingly.

Finally, as already observed in our nonsmooth approach to H_∞ synthesis [2, 3], specific structural constraints on the controller can be easily incorporated in (10) by applying chain rules to the subgradients. The reader is referred to [1] for details.

In the following, we investigate practically interesting options for analysis and synthesis with IQCs.

4.1 IQC analysis

Nonsmooth results from Section 3 can be used to compute the L_2 gain or H_∞ norm of a system or perform a passivity test. In this situation, the uncertainty channel (w_Δ, z_Δ) is removed, $T = T_{zw}$, and the multiplier $\Pi = \Pi_p$ is selected as

$$\Pi_p = \begin{bmatrix} \gamma^{-1} I & 0 \\ 0 & -\gamma I \end{bmatrix}, \text{ respectively } \Pi_p = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}. \tag{15}$$

Assume now for stability analysis that the system in (1) is subject to time-invariant parametric uncertainties $T = T_{\Delta\Delta}$, $w_\Delta = \Delta z_\Delta$, where Δ is a multiplication operator with the block-diagonal structure

$$\Delta = \operatorname{diag}(\ldots, \delta_i I, \ldots, \Delta_j, \ldots) \in \mathbb{R}^{m_\Delta \times m_\Delta} \tag{16}$$

with normalization $\Delta^T \Delta \preceq I$. Then robust stability can be tested using μ-upper bound multipliers of the form

$$\Pi = \Pi_\Delta = \begin{bmatrix} S & (jG)^H \\ jG & -S \end{bmatrix}$$

with $S = S^H \succ 0$ and $G = G^H$, where both S and G commute with Δ. For simplicity of the exposition, we shall first consider constant multipliers. The treatment of more complex multipliers is deferred to Section 4.4. The subgradient formulas in Theorem 3.4 must then be modified to cope with the particular structure of Π and with the extra constraint $S \succ 0$. We define $S = \Sigma\Sigma^H$ where Σ is a lower-triangular Cholesky factor of S. This leads to

$$\delta\Pi = \begin{bmatrix} \Sigma(\delta\Sigma)^H + (\delta\Sigma)\Sigma^H & (j\delta G)^H \\ j\delta G & -\Sigma(\delta\Sigma)^H - (\delta\Sigma)\Sigma^H \end{bmatrix}.$$

Substitution of this expression into the general formula (12), using Theorem 3.4 and the identities

$$\begin{aligned} \operatorname{Tr}(MN^H + M^H N) &= 2 \operatorname{Tr}(\operatorname{Re} M \operatorname{Re} N^T + \operatorname{Im} M \operatorname{Im} N^T) \\ \operatorname{Tr}(ML^T) &= \operatorname{Tr}(\operatorname{tril}(M) L^T) \end{aligned}$$

which hold for arbitrary complex M, N and lower-triangular L yields the sought formulas. Subgradients with respect to $\operatorname{Re}(\Sigma)$ and $\operatorname{Im}(\Sigma)$ are obtained respectively as $\operatorname{tril}(\operatorname{Re} U)$ and $\operatorname{tril}(\operatorname{Im} U)$ where

$$U := 2 \sum_{\omega \in \Omega(\Sigma, G)} T(j\omega, K) Q_\omega Y_\omega Q_\omega^H T(j\omega, K)^H \Sigma - Q_\omega Y_\omega Q_\omega^H \Sigma. \tag{17}$$

Subgradients with respect to $\operatorname{Re} G$ and $\operatorname{Im} G$ are obtained respectively as $-(\operatorname{Im} V + \operatorname{Im}(V)^T)$ and $\operatorname{Re} V - \operatorname{Re}(V)^T$ with the definition

$$V := \sum_{\omega \in \Omega(\Sigma, G)} T(j\omega, K) Q_\omega Y_\omega Q_\omega^H .$$

As before, $\Omega(\Sigma, G)$ is the set of active frequencies for given Σ and G, the family Y is as in Theorem 3.4. For time-varying uncertainties with arbitrarily fast variations $w_\Delta = \Delta(t) z_\Delta$, the associate multiplier is real

$$\Pi := \begin{bmatrix} S & \Gamma \\ \Gamma^T & -S \end{bmatrix}$$

where as before $S = \Sigma\Sigma^T \succ 0$ and $\Gamma^T = -\Gamma$. Subgradients are readily inferred from the complex case. For Σ, the subgradients are $\operatorname{tril}(\operatorname{Re} U)$ with U defined in (17). For Γ, we get the subgradients $\operatorname{Re} V - \operatorname{Re}(V)^T$ where V is as described above.

4.2 H_∞ and positive real syntheses

H_∞ and positive real syntheses are special instances where the uncertainty channel is removed, $T = T_{zw}$, and where $\Pi = \Pi_p$ is chosen as in (15). For H_∞ synthesis we obtain the subgradients with respect to K

$$\Phi_Y = 2/\gamma \sum_{\omega \in \Omega(K)} \operatorname{Re} \left(G_{21}(j\omega) Q_\omega Y_\omega Q_\omega^H \, T(j\omega, K)^H \, G_{12}(j\omega) \right)^T ,$$

which is consistent with the results already derived in [3].

4.3 Robust synthesis

The above reasoning is easily generalized to robust L_2-gain synthesis with structured parametric uncertainties. Again with constant multipliers we have

$$\Pi := \left[\begin{array}{c|c} \Pi_{11} & \Pi_{12} \\ \hline \Pi_{12}^H & \Pi_{22} \end{array} \right] := \left[\begin{array}{cc|cc} S & 0 & (jG)^H & 0 \\ 0 & \gamma^{-1} I & 0 & 0 \\ \hline jG & 0 & -S & 0 \\ 0 & 0 & 0 & -\gamma I \end{array} \right] . \tag{18}$$

Clarke subgradients with respect to K are obtained from (13) and the definition in (18). Partial subgradients with respect to the multiplier are easily inferred from (17) and the partitioning in (5). They are $\operatorname{tril}(\operatorname{Re} U)$ and $\operatorname{tril}(\operatorname{Im} U)$ for the subgradients with respect to $\operatorname{Re}(\Sigma)$ and $\operatorname{Im}(\Sigma)$, where $S = \Sigma\Sigma^H$ and

$$U := 2 \sum_{\omega \in \Omega(\Sigma, G, K)} [\, T_{\Delta\Delta} \quad T_{\Delta w} \,] \, Q_\omega Y_\omega Q_\omega^H \, [\, T_{\Delta\Delta} \quad T_{\Delta w} \,]^H \, \Sigma - [\, I \quad 0 \,] \, Q_\omega Y_\omega Q_\omega^H \, [\, I \quad 0 \,]^H \, \Sigma$$

Subgradients with respect to $\operatorname{Re} G$ and $\operatorname{Im} G$ are obtained respectively as $-(\operatorname{Im} V + \operatorname{Im}(V)^T)$ and $\operatorname{Re} V - \operatorname{Re}(V)^T$ with the definition

$$V := \sum_{\omega \in \Omega(\Sigma, G, K)} [\, T_{\Delta\Delta} \quad T_{\Delta w} \,] \, Q_\omega Y_\omega Q_\omega^H \, [\, I \quad 0 \,]^H$$

Note that $\Omega(\Sigma, G, K)$ is the set of active frequencies for a given triple (Σ, G, K).

Clearly, the above analysis is applicable to more general hybrid block-diagonal operators Δ where each sub-block Δ_i satisfies an IQC defined by Π_i. The reader is referred to the IQCs listed in [19] to enrich the discussion along this line.

4.4 Dynamic multipliers

To reduce conservatism in IQC analysis or synthesis it is possible to use dynamic multipliers. In this context $\Pi(s)$ is an unknown function, and the problem is infinite-dimensional, which is not directly tractable. What hinders for infinite dimensional $\Pi(s)$ are not the subgradient formulas, but the computation of the function $\lambda_{1,\infty} \circ F$. The way out is to use finitely generated multipliers, possibly of the form

$$\Pi(s) = R(s)^H \Phi R(s), \quad \Phi = \Phi^H , \tag{19}$$

where $R(s)$ is assumed to have a state-space realization $R(s) = C_R(sI - A_R)^{-1}B_R + D_R$ and where Φ, the quadruple (A_R, B_R, C_R, D_R) and the controller K are unknown. The framework developed in Sections 2 and 3 applies and differential information can be obtained.

We present an interesting alternative to (19), which is covered by our technique, but in contrast is hardly accessible by state-space methods which call for LMIs or BMIs. Fix a finite partition $[0, \infty] = I_i \cup \ldots \cup I_\ell$ into frequency bands I_i and choose different constant multipliers Π_i on each I_i. In other words, write a dynamic piecewise constant multiplier $\Pi(s) = \sum_{i=1}^{\ell} \Pi_i \chi_{I_i}(s)$, where χ_{I_i} is the indicatrix function of the ith band. Then robust performance can be expressed as

$$F_i(K, \Pi_i; j\omega) := \left[\frac{T(j\omega, K)}{I}\right]^H \Pi_i \left[\frac{T(j\omega, K)}{I}\right] \prec 0, \quad \forall \omega \in I_i, \forall i = 1, \ldots, \ell .$$

This is equivalent to

$$\max_{i=1,\ldots,\ell} \max_{\omega \in I_i} \lambda_1(F_i(K, \Pi_i; j\omega)) < 0 .$$

We get a max function for which the Clarke gradient is computed using the convex hull rule [11] and is similar in structure to the Clarke gradient in Theorem 3.4. In this approach the finite basis is more natural than in (19). The procedure in Section 2 to compute function values and active frequencies works on prescribed frequency intervals I_i just as well as on the whole $[0, \infty]$. See [4] for a proposal on how this can be organized for multiband H_∞ synthesis.

5 Nonsmooth descent method

In this section we explain the basic mechanism of our descent method. Due to the structure of the objective function $f = \lambda_{1,\infty} \circ F$, it will be helpful to look at pure eigenvalue optimization problems $\min_{x \in \mathbb{R}^n} \lambda_1(F(x))$. This class has been studied by various authors, see e.g. [12, 18, 21, 22, 20]. We will extend a method developed by Helmberg *et al.* [14] for semidefinite programming (SDP) to address objective functions of the form $f = \lambda_{1,\infty} \circ F$.

Definition. A mapping $S : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is called a descent step generator with memory for the function $f : \mathbb{R}^n \to \mathbb{R}$ if the following are satisfied.

(i) If $0 \notin \partial f(x)$, then $f(S(x, g)) < f(x)$ for every $g \in \mathbb{R}^n$.

(*ii*) Suppose $0 \notin \partial f(x)$ and $g \in \mathbb{R}^n$. Then there exists $\varepsilon > 0$ and $\delta > 0$ such that for every $x' \in B(x, \varepsilon)$ and $g' \in B(g, \varepsilon)$ we have $f(S(x', g')) \leq f(x) - \delta$.

We think of x as the current iterate, g the memory element, reflecting for instance information from previous iterates, or a subgradient at x aggregated over the past as in [14], and we call $S(x, g)$ the descent step generated at x, based on the information in x and g.

Clearly condition (*i*) says that a descent step at x is always possible (regardless of the information in g) as long as x is not a critical point. For a first-order method, this is what we expect. Condition (*ii*) is stronger than (*i*), but would follow automatically from (*i*) if $S(\cdot, \cdot)$ was continuous. Since the latter will not always be the case in practice, (*ii*) should be understood as a weak form of continuity of $S(\cdot, \cdot)$.

Consider the following example of a memoryless descent step generator for f of class $C^{1,1}$ [9]. Choosing the direction $d = -f'(x)$ of steepest descent of f at x, we start a backtracking line search in direction d with initial $t = 1$ and accept $S(x) = x + t(x)d$, where $t(x)$ is the first step length satisfying the Armijo condition. Then $t(x)$ is not necessarily a continuous function of x, but condition (*ii*) is satisfied. This method fails for nonsmooth f, and an alternative is given below.

Theorem 5.1 *Let the sequence of iterates x_k be generated by the descent step generator $S(\cdot, \cdot)$, that is, $x_{k+1} = S(x_k, g_k)$, where the memory elements g_k are in a bounded set. Then every accumulation point of the sequence x_k is a critical point.*

Proof. Let x_k, $k \in \mathcal{K}$ be a subsequence of the sequence of iterates converging to $\bar{x}$. Assume without loss that $g_k \to \bar{g}$, for some $\bar{g}$, passing if necessary to a sub-subsequence. Notice that $S(\cdot, \cdot)$ is a descent method, that is, $f(x_k) \geq f(x_{k+1}) \geq f(x_{k'})$, where k' is the successor of k in $\mathcal{K}$. Therefore $f(x_{k+1}) \to f(\bar{x})$, and hence $f(x_k) - f(x_{k+1}) \to 0$.

Now suppose $\bar{x}$ is not critical. Choose ε, δ as in axiom (*ii*) with respect to $\bar{x}$ and $\bar{g}$. Then $f(x_k) - f(x_{k+1}) = f(x_k) - f(S(x_k, g_k)) \geq \delta$ for $k \geq k_0$, because $x_k \in B(\bar{x}, \varepsilon)$ and $g_k \in B(\bar{g}, \varepsilon)$ for $k \geq k_0$. This contradicts $f(x_k) - f(x_{k+1}) \to 0$. □

Note that convergence toward a critical point may appear a weak certificate, but experience in numerical optimization shows that in practice solutions are almost always local minima [9]. The question is now how to construct descent step generators for functions of the form $\lambda_1 \circ F$ and $\lambda_{1,\infty} \circ F$. In [3] we have discussed two memoryless descent step generators. Here we discuss an extension of the method in [14], first to non-convex $\lambda_1 \circ F$, then to the semi-infinite case. Just as [14], the method in [12, 21] and its extension to the non-convex case [20] use non-polyhedral inner approximations of the ε-subdifferential of λ_1, but [14] includes information from past iterates by maintaining an aggregate subgradient from previous iterates.

At the given iterate, x, we consider two local models of $f = \lambda_1 \circ F$ at x:

$$\hat{f}(y) = \lambda_1 \left(F(x) + F'(x)(y - x) \right), \quad \hat{f}_{\mathcal{G}(x)}(y) = \max_{G \in \mathcal{G}(x)} G \bullet \left(F(x) + F'(x)(x - y) \right),$$

where $\mathcal{G}(x) = \{\alpha G^\sharp + QYQ^T : \alpha + \mathrm{Tr}(Y) = 1, \alpha \geq 0, Y \succeq 0\}$. Here $G^\sharp$ is the aggregate subgradient of λ_1, to be specified later, while the columns of Q form a truncated basis of k eigenvectors of $F(x)$, that is, $Q^T F(x) Q = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$ with $Q^T Q = I_k$. Notice that every QYQ^T is a subgradient of λ_1 at $F(x)$, which means $F'(x)^\star \mathcal{G}(x) \subset \partial f(x)$. In consequence, $\hat{f}_{\mathcal{G}(x)} \leq \hat{f}$, because $\hat{f}(y) = \sup_{G \in \partial \lambda_1(F(x))} G \bullet (F(x) + F'(x)(y - x))$. Notice that by adding new elements

$G \in \partial\lambda_1(F(x))$ to $\mathcal{G}(x)$, we improve the approximation $\hat{f}_{\mathcal{G}(x)}$ of $\hat{f}$. Notice that in the convex case, where F is an affine operator, we have $F(y) = F(x) + F'(x)(y - x)$, so here $\hat{f} = f$. In the general non convex case, $\hat{f}$ is only an approximation of f. In order to improve the approximation of f by $\hat{f}$, we have to restrict y to a sufficiently small ball $B(x, r)$, where $r > 0$ is the trust region radius.

Descent step generator $S(x, g)$ for $\lambda_1 \circ F$

Input: current x and aggregate subgradient $G^\sharp$. **Output:** $x^+ := S(x, g)$, $(G^\sharp)^+$.
Fix $0 < \alpha < \frac{1}{2}$, $0 < \beta < 1$.
1. Initialize $r = 1$ and choose k and orthonormal basis Q of the first k eigenvalues of $F(x)$. 2. For given $r > 0$, solve $\min_{\|x-y\| \leq r} \hat{f}_{\mathcal{G}(x)}(y)$, solution is $\hat{x}$. Find $\hat{G} = \hat{\alpha}G^\sharp + Q\hat{Y}Q^T$ where the supremum $\hat{f}_{\mathcal{G}(x)}(\hat{x})$ is attained. 3. If $\hat{f}(\hat{x}) - f(x) \leq \alpha \left(\hat{f}_{\mathcal{G}(x)}(\hat{x}) - f(x)\right)$ then put $\hat{x}$ on store and go to step 5. Otherwise 4. Update $\mathcal{G}(x)$ by updating aggregate subgradient $G^\sharp$ as $(G^\sharp)^+ = \hat{\alpha}G^\sharp + Q\hat{Y}Q^T$. Go back to step 2. 5. Check whether $f(\hat{x}) - f(x) < \beta \left(\hat{f}(\hat{x}) - f(x)\right)$. If this is the case accept $x^+ = \hat{x}$. Otherwise replace r by $r/2$ and go back to step 2.

Reference [14] studies the convex case $f = \hat{f}$ and therefore uses only the test in step 3 and the updating mechanism in step 4, which improves the approximation $\hat{f}_{\mathcal{G}(x)}$ by modifying $\mathcal{G}(x)$. The authors of [14] use an even more sophisticated update of $G^\sharp$ in cases where $\hat{Y}$ is large, but this is not mandatory in typical control applications. The test in step 5 is not used in [14], where the authors keep r fixed. It becomes necessary because f is non convex. Appropriate ways of choosing Q and k in step 1 have been discussed in [21, 20, 3], so we do not go into details here. Notice that steps 1 - 4 create output which varies continuously with respect to a change of the data. What makes our descent step generator discontinuous is step 5, where r is updated and finally chosen within the set $\{2^{-k} : k \geq 1\}$. This discrete element of the procedure destroys continuity, but fortunately property (ii) is still satisfied. Typically what may happen is that for a given datum (x, g), the solution test in step 4 is passed for the first time at $r = 2^{-k}$, but for a nearby (x', g'), one needs one additional reduction, so that the $r' = 2^{-k-1}$ may be obtained. The situation is similar to what happens in the case of a line search, when a step size $t \in \{2^{-k} : k \geq 1\}$ is picked. See [3] and in particular [24, p.223-224].

Finally, we need to extend the descent step generator to the semi-infinite case $f = \lambda_{1,\infty} \circ F$. We use the following observation. Computing $f(x)$ provides the finite set of active frequencies $\Omega(x)$. For a finite extension Ω of $\Omega(x)$, that is $\Omega(x) \subset \Omega$, let $f_\Omega(y) := \max_{\omega\in\Omega} \lambda_1 (F(y, \omega))$, then $f_\Omega \leq f$ and $f_\Omega(x) = f_{\Omega(x)}(x) = f(x)$. Each f_Ω is a maximum eigenvalue function $\lambda_1 \circ F_\Omega$, where F_Ω is block diagonal with the finitely many $F(\cdot, \omega)$, $\omega \in \Omega$ as blocks. So f_Ω admits a descent step generator $S_\Omega(\cdot, \cdot)$ by the above construction. We finally have the following

Descent step generator for $f = \lambda_{1,\infty} \circ F$

Fix $0 < \theta < 1$.	
1.	Compute $f(x)$ and detect active frequencies $\Omega(x)$.
2.	Select an extended set of frequencies $\Omega_e(x)$, for instance by setting a threshold $\theta f(x) < f(x)$, and taking a griding $\Omega_e(x)$ of the zone of those ω where $\theta f(x) < f(x,\omega) \leq f(x)$.
3.	Generate descent step $S_{\Omega_e(x)}(x,g)$ and return it.

This semi-infinite descent step generator has already been analyzed in [1], where a memoryless descent step generator $S(x)$ is used. The analysis of the present case follows similar lines and justifies theoretically what according to our previous experience works well in practice. In order to assure property (ii), it suffices to select the griding $\Omega_e(x)$ in such a way that it varies continuously with a change of the datum x, which is possible in a neighborhood of a local minimum x^*. For details we refer to [1]. The outlined selection of extended frequencies $\Omega_e(x)$ has been tested and shown to work well in H_∞-, multidisk, and multiband synthesis, and we recommend it for IQC synthesis based on that experience.

6 Illustration

In this section, a brief illustration of the proposed techniques is presented. Let us consider the mass-spring system shown in Figure 1. This is a small size example whose dynamics comprise two flexible modes. A difficulty lies in the presence of parametric uncertainties in the mass m_2 and stiffness k. Nominal synthesis techniques typically fail on this type of problem as uncertainties are not taken into account and controllers tend to inverse the badly-damped dynamics of the plant. The lack of collocation between the control input u and the measurement $y = x_2$ is a second source of difficulty. Note that similar benchmarks have previously been studied in the literature [5].

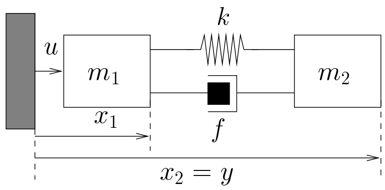

Figure 1: mass-spring system

Nominal parameter values are :

$$m_1 = m_2 = 0.5\,Kg;\ k = 1\,N/m;\ f = 0.0025\,Ns/m. \tag{20}$$

Measurement $y = x_2$ must track a reference signal y_{ref}, typically a step input. Performance specification is a settling time of about 4 seconds for the step response. Performance should be robust with respect to relative parameter variations δ_k and δ_{m_2} of $\pm 30\,\%$ on k and m_2, respectively. The plant, augmented by the uncertainty channel, is described in LFT (Linear Fractional

Transformation) format and is given in state-space as :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_1 \\ \ddot{x}_2 \\ \hline z_{\Delta_k} \\ z_{\Delta_{m_2}} \\ \hline y \end{bmatrix} = \left[\begin{array}{cccc|cc|c} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -k/m_1 & k/m_1 & -f/m_1 & f/m_1 & -1/m_1 & 0 & 1/m_1 \\ k/m_2 & -k/m_2 & f/m_2 & -f/m_2 & 1/m_2 & -1 & 0 \\ \hline k & -k & 0 & 0 & 0 & 0 & 0 \\ k/m_2 & -k/m_2 & f/m_2 & -f/m_2 & 1/m_2 & -1 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ \dot{x}_1 \\ \dot{x}_2 \\ \hline w_{\Delta_k} \\ w_{\Delta_{m_2}} \\ \hline u \end{bmatrix} \tag{21}$$

with uncertainty channel:

$$\begin{bmatrix} w_{\Delta_k} \\ w_{\Delta_{m_2}} \end{bmatrix} = \begin{bmatrix} \delta_k & 0 \\ 0 & \delta_{m_2} \end{bmatrix} \begin{bmatrix} z_{\Delta_k} \\ z_{\Delta_{m_2}} \end{bmatrix} . \tag{22}$$

The problem is solved using a classical mixed-sensitivity formulation [25, p. 130-141]

$$\min_{K(s)} \left\| \begin{array}{c} \Sigma \\ G\Sigma \end{array} \right\|_\infty \quad \text{where} \quad \Sigma := (I - KG)^{-1} . \tag{23}$$

In order to illustrate the lack of robustness of nominal synthesis, a standard H_∞ synthesis with the same performance channel is computed. The resulting compensator is analyzed through four conventional graphical tools:

- a root locus (upper-left plot) to diagnose pole-zero cancellations.
- a Nichols plot (upper-right plot) to evaluate classical stability margins.
- a stability domain (lower-left plot) which displays the stability region in the parameter plane (dotted area). The square corresponds to the prescribed parameter variations of ±30% and should be contained in the dotted area when robust stability is achieved.
- closed-loop step responses (lower-right plot) for all models in the uncertainty set $(\delta_k, \delta_{m_2}) \in \{-30, 0, 30\} \times \{-30, 0, 30\}$.

As seen in Figure 2, the standard H_∞ controller is not satisfactory in this example, which motivates computing a robust controller using the method of section 4.3. The uncertainty channel is now incorporated into the design, and correspondingly, suitable multipliers are defined, see (18). In this preliminary study we have used simple constant multipliers as discussed in 4.3. The synthesis problem is cast as in (9), where both controller K and multiplier Π are the design variables. Note that the previously computed H_∞ controller will serve as an initial stabilizing solution for the algorithm. A simple safeguard is used to maintain closed-loop stability in the course of the algorithm of section 5 by setting $f(x) = \lambda_{1,\infty} \circ F(x) = +\infty$ whenever K is not stabilizing. The multiplier Π in (18) is initialized with $S = I$, $G = 0$ and $\gamma = 3$. The extended set of frequencies introduced is section 5 is obtained by griding frequency intervals where function values exceed 80% of the peak value. The initial FDI function $\lambda_1 \circ F(K, \Pi, j\omega)$ is shown in Figure 3, which also indicates the frequencies which were selected to construct the extended set $\Omega_e(x)$ for the descent step generator of $f = \lambda_{1,\infty} \circ F$. The algorithm reaches a negative peak for f after 31 iterations and converges to a local minimum after 88 iterations leading to the FDI curve with strictly negative peak value in Figure 4. Note that running the algorithm to completion is neither necessary nor advisable in this application since only a strictly feasible solution to the FDI is required. This can certainly be exploited to reduce the computational overhead. Frequency- and time-domain analysis of the computed robust controller are displayed in Figure 5.

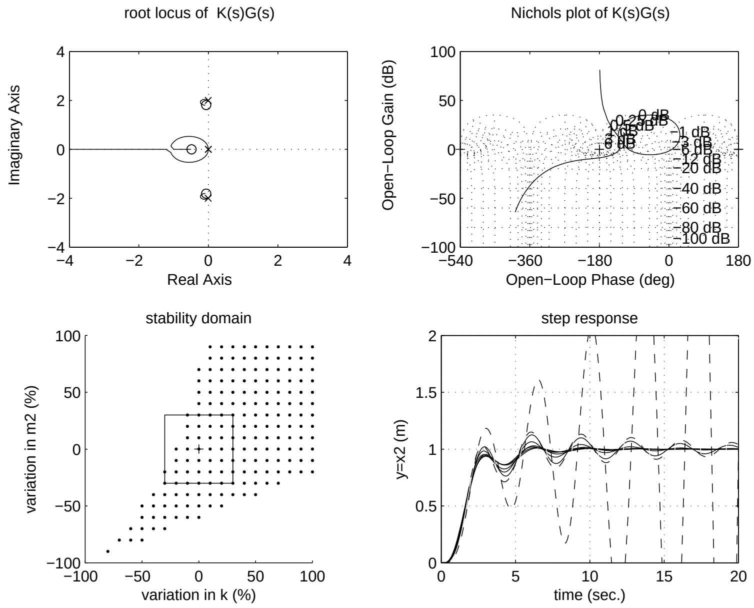

Figure 2: analysis of standard H_∞ controller

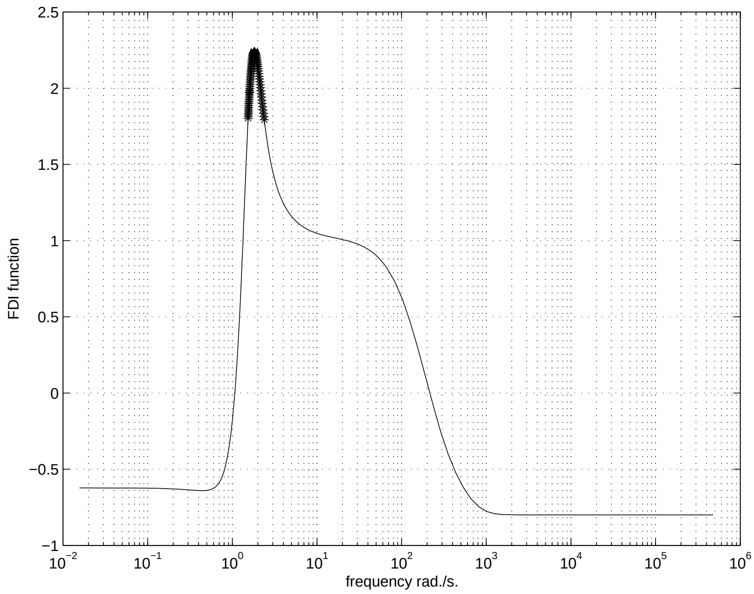

Figure 3: initial FDI curve
'*' selected frequencies in extended set

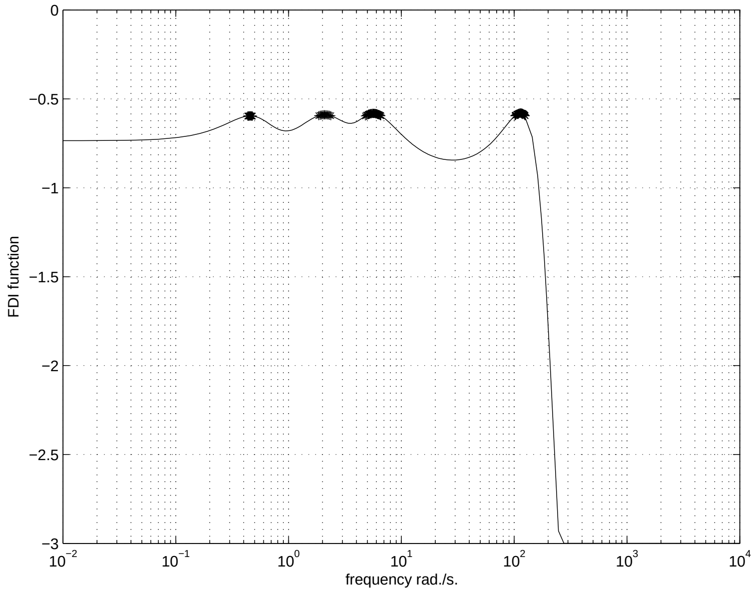

Figure 4: final FDI curve
'*' selected frequencies in extended set

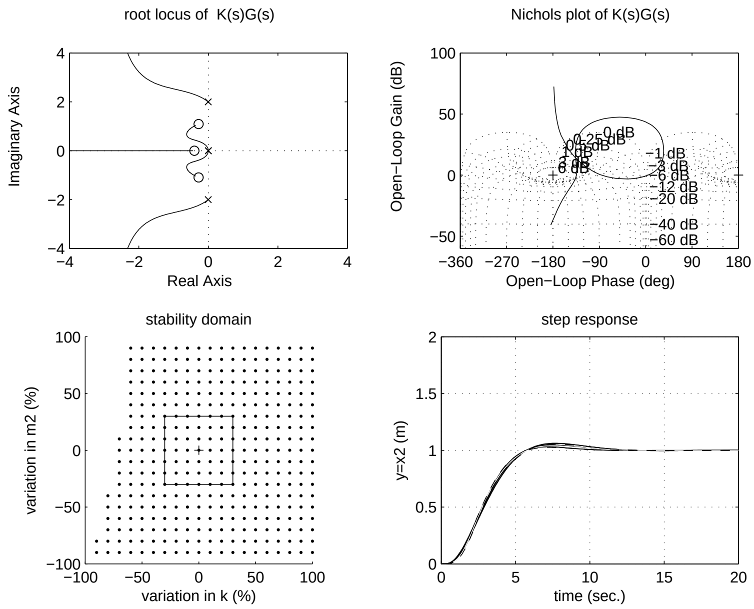

Figure 5: analysis of final robust controller

7 Conclusion

We have used nonsmooth analysis to establish useful differential properties of FDIs arising in IQC theory. The proposed framework is general and can handle a broad class of IQCs including those with rational or even more general dynamic multipliers. Moreover, both control system analysis and synthesis are covered by our theory. A new nonsmooth bundle-type algorithm with memory has been developed and its convergence has been established.

As illustrated on a simple benchmark problem, and in accordance with our preliminary testing in H_∞ synthesis [3] and some of its variants [1, 4], the proposed approach appears very promising, particularly so for problems where the state dimension is an issue.

References

[1] P. Apkarian and D. Noll, *Nonsmooth optimization for multidisk H_∞ synthesis*, to appear in European Journal of Control, (2005).

[2] ——, *Controller design via nonsmooth multi-directional search*, SIAM J. on Control and Optimization, 44 (2006), pp. 1923–1949.

[3] ——, *Nonsmooth H_∞ synthesis*, IEEE Trans. Aut. Control, 51 (2006), pp. 71–86.

[4] ——, *Nonsmooth optimization for multiband frequency domain control design.*, submitted, (2006).

[5] D. S. B. B. Wie, *A benchmark problem for robust controller design*, in Proc. American Control Conf., San Diego, CA, May 1990, pp. 961–962.

[6] V. Bompart, D. Noll, and P. Apkarian, *Second-order nonsmooth optimization for H_∞ and H_2 syntheses*, submitted, (2005).

[7] S. Boyd and V. Balakrishnan, *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $\mathbf{L}_\infty$-norm*, Syst. Control Letters, 15 (1990), pp. 1–7.

[8] S. Boyd, V. Balakrishnan, and P. Kabamba, *A bisection method for computing the H_∞ norm of a transfer matrix and related problems*, Mathematics of Control, Signals, and Systems, 2 (1989), pp. 207–219.

[9] R. Burachik, L. Drummond, and B. Svaiter, *Full convergence of the steepest descent method with inexact line searches*, Journal of Optimization, 32 (1995), pp. 137–146.

[10] a. J. C.-Y. Kao, A. Megretski, *Specialized fast algorithms for IQC feasibility and optimization problems*, Automatica, 40 (2004), pp. 239–252.

[11] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Canadian Math. Soc. Series, John Wiley & Sons, New York, 1983.

[12] J. Cullum, W. Donath, and P. Wolfe, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Math. Programming Stud., 3 (1975), pp. 35 – 55.

[13] A. Hansson and L. Vandenberghe, *Efficient solution of linear matrix inequalities for integral quadratic constraints*, in Proc. IEEE Conf. on Decision and Control, vol. 5, Sydney, NSW, Australia, 2000, pp. 5033–5034.

[14] C. Helmberg and F. Rendl, *Spectral bundle method for semidefinite programming*, SIAM J. Optimization, 10 (2000), pp. 673 – 696.

[15] J. W. Helton and O. Merino, *Coordinate optimization for bi-convex matrix inequalitites*, in Proc. IEEE Conf. on Decision and Control, San Diego, CA, 1997, pp. 3609–3613.

[16] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*, vol. 306 of Grundlehren der mathematischen Wissenschaften, Springer-Verlag, New York, 1993.

[17] U. Jönsson, *Robustness Analysis of Uncertain and Nonlinear Systems*, PhD thesis, Lund Institute of Technology, Lund ,Sweden, 1996.

[18] C. Lemaréchal and F. Oustry, *Nonsmooth algorithms to solve semidefinite programs, SIAM Advances in Linear Matrix Inequality Methods in Control series, ed. L. El Ghaoui & S.-I. Niculescu*, (2000).

[19] A. Megretski and A. Rantzer, *System analysis via integral quadratic constraints*, IEEE Trans. Aut. Control, 42 (1997), pp. 819–830.

[20] D. Noll and P. Apkarian, *Spectral bundle methods for nonconvex maximum eigenvalue functions: first-order methods*, Mathematical Programming Series B, 104 (2005), pp. 701–727.

[21] F. Oustry, *A second-order bundle method to minimize the maximum eigenvalue function*, Math. Programming Series A, 89 (2000), pp. 1 – 33.

[22] M. Overton, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. on Matrix Analysis and Applications, 9 (1988), pp. 256–268.

[23] P. Parrilo, *On the numerical solution of LMIs derived from the KYP lemma*, in Proc. IEEE Conf. on Decision and Control, Phoenix, AZ, 1999, pp. 2334–2338.

[24] E. Polak, *Optimization : Algorithms and Consistent Approximations*, Applied Mathematical Sciences, 1997.

[25] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Printice Hall, 1996.