Second-order nonsmooth optimization for H_∞ synthesis

Vincent Bompart * Dominikus Noll † Pierre Apkarian ‡

Abstract

The standard way to compute H_∞ feedback controllers uses algebraic Riccati equations and is therefore of limited applicability. Here we present a new approach to the H_∞ output feedback control design problem, which is based on nonlinear and nonsmooth mathematical programming techniques. Our approach avoids the use of Lyapunov variables, and is therefore flexible in many practical situations.

Keywords: Output feedback control – H_∞ control – semi-infinite programming – sequential quadratic programming (SQP) – nonsmooth optimization – eigenvalue optimization – trust-region.

1 Introduction

Well designed feedback control systems in high technology fields are expected to respond favorably to a list of concurring performance specifications such as stability, good regulation against disturbances, desirable responses to commands, robustness, control law specifications, system security, and much else. In addition, controllers should be hardware implementable at low cost, and should allow a flexible adaption to strategic changes during the model building process. These ever growing demands have shown the limitations of

*ONERA, Control System Department, 2 av. Edouard Belin, 31055 Toulouse, France - and - Université Paul Sabatier, Institut de Mathématiques, 118 route de Narbonne, 31062 Toulouse, France - Email: bompart@onera.fr - Tel: +33 5.62.25.25.25 +22.11 - Fax: +33 5.62.25.27.64.

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

‡ONERA, Control System Department - and - Université Paul Sabatier, Institut de Mathématiques, Toulouse, France - Email: apkarian@onera.fr - Tel: +33 5.62.25.27.84 - Fax: +33 5.62.25.27.64.

currently used mathematical tools for synthesis, which are based on solving algebraic Riccati equations (AREs). The rise of linear matrix inequalities (LMIs) since the early 1990s has certainly improved the situation, but the main limitations of these approaches persist.

In response, various new design methods have been proposed since the late 1990s, including methods from global optimization, [6, 7, 37, 34], matrix inequality constrained nonlinear programming, [14, 24, 25, 5, 36], augmented Lagrangian methods, [13, 30], eigenvalue optimization, [1, 2, 3, 19], and others. Presently we discuss a new strategy for H_∞ output feedback control synthesis, which has several advantages over existing approaches, because it avoids the use of Lyapunov variables. We shall show that this leads to much smaller and better manageable problems, at the price that controller synthesis becomes a nonsmooth optimization program. We develop local nonsmooth optimization strategies suited for this new context, and show that they improve the situation considerably.

2 Problem setting

We consider a linear time-invariant plant described in standard form by the state-space equations

$$P(s): \qquad \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0_{p_2 \times m_2} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix},$$

where s is the Laplace variable, $x \in \mathbb{R}^n$ is the state of the system, $u \in \mathbb{R}^{m_2}$ the control, $w \in \mathbb{R}^{m_1}$ an exogenous signal (noise, commands), $z \in \mathbb{R}^{p_1}$ the regulated variable (response of the system to the input w), and $y \in \mathbb{R}^{p_2}$ the measured output. The goal of feedback control synthesis is to design an output feedback controller

$$K(s): \qquad \begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \begin{bmatrix} x_K \\ y \end{bmatrix}$$

where $x_K \in \mathbb{R}^{n_K}$, $A_K \in \mathbb{R}^{n_K \times n_K}$, $B_K \in \mathbb{R}^{n_K \times p_2}$, $C_K \in \mathbb{R}^{m_2 \times n_K}$ and $D_K \in \mathbb{R}^{m_2 \times p_2}$, such that the following three criteria are met:

Internal stability: The closed-loop system obtained by substituting K into P is exponentially stable.

Performance: Among all internally stabilizing controllers, K minimizes the H_∞-norm of the performance channel $w \to z$.

Control law specifications: The controller K may be subject to additional structural constraints. For instance, one may look for decentralized controllers, reduced-order controllers, PID, feed-forward, lead lag structures, etc.

Let us explain the performance requirement for the case $u = Ky$ of a static output feedback controller, where $D_K = K \in \mathbb{R}^{m_2 \times p_2}$. Here the closed-loop transfer function for the performance channel $w \to z$ is

$$T_{w\to z}(K)(s) = \mathcal{D}(K) + \mathcal{C}(K)\big(sI_n - \mathcal{A}(K)\big)^{-1}\mathcal{B}(K) \tag{1}$$

where the closed loop data are

$$\begin{aligned} \mathcal{A}(K) &= A + B_2KC_2 & \mathcal{B}(K) &= B_1 + B_2KD_{21} \\ \mathcal{C}(K) &= C_1 + D_{12}KC_2 & \mathcal{D}(K) &= D_{11} + D_{12}KD_{21} \end{aligned}$$

For brevity, we shall subsequently write $T(K,\omega) := T_{w\to z}(K)(j\omega)$. Denoting by λ_1 the maximum eigenvalue function on the space of Hermitian matrices, and by $\bar{\sigma}$ the maximum singular value of $m_1 \times p_1$ matrices, we define

$$f(K,\omega) = \lambda_1(T(K,\omega)^H T(K,\omega)) = \left[\bar{\sigma}(T(K,\omega))\right]^2,$$

where Z^H stands for the conjugate transpose of the complex matrix Z.

The square of the H_∞ norm is then defined as

$$\|T_{w\to z}(K)\|_\infty^2 = \max_{\omega\in[0,\infty]} f(K,\omega). \tag{2}$$

Optimizing the performance of the controller therefore leads to minimizing the function $f(K) = \max_{\omega\in[0,\infty]} f(K,\omega)$ over the set $\mathcal{S}$ of all closed-loop exponentially stabilizing controllers $K \in \mathbb{R}^{m_2\times p_2}$.

Notice that f is nonsmooth with two possible sources of nonsmoothness: (a) the infinite max-operator, and (b) the nonsmoothness of λ_1, which may lead to nonsmoothness of $f(\cdot,\omega)$ for fixed ω.

Remark. A difficulty is that the set of exponentially stabilizing controllers $\mathcal{S}$ is open and not a constraint set in the usual sense of nonlinear programming. We have observed that it is usually possible to ignore the constraint $K \in \mathcal{S}$, and start the minimization (3) with an initial closed-loop stabilizing controller $K \in \mathcal{S}$. The fact that $\|T_{w\to z}(K)\|_\infty$ is minimized, and therefore stays finite, very often implies closed-loop stability of the limit point $\bar{K}$. That is why we shall consider program

$$\min_{K\in\mathbb{R}^{m_2\times p_2}} \quad f(K) = \|T_{w\to z}(K)\|_\infty^2. \tag{3}$$

under the implicit assumption that iterates K_k and the limit $\bar{K}$ are closed-loop stabilizing.

In those cases where $\|T_{w\to z}(K)\|_\infty < \infty$ does *not* imply closed-loop stability, a problem occurs. Here a minimizing sequence $K_k \in \mathcal{S}$ for (3) will converge to a controller $\bar{K}$ on the boundary of the domain $\mathcal{S}$. Such a $\bar{K}$ is not a valid solution of the control problem. Fortunately, the phenomenon is easily detected, and a practical way out is to consider an additional closed-loop transfer channel $T_{\text{stab}}(K,s) = (sI - \mathcal{A}(K))^{-1}$, called the stabilizing channel. For a small parameter $\epsilon > 0$, we may then minimize the objective

$$f(K) = \max\left(\|T_{w\to z}(K)\|_\infty^2, \epsilon\|T_{\text{stab}}(K)\|_\infty^2\right), \tag{4}$$

and we are now sure that $f(K) < \infty$ implies closed-loop stability of K. Notice that f is again of the form (3) if a transfer matrix with the two diagonal blocks $T_{w\to z}$ and ϵT_{stab} is introduced. □

Remark. Our approach to H_∞ controller synthesis based on program (3) differs substantially from standard ways to solve such problems, where the H_∞ norm is avoided and replaced by a matrix inequality using the bounded real lemma [10]. In solving (3) directly we dispense with Lyapunov variables, which leads to a spectacular reduction in the number of unknowns, especially when small controllers for large systems are sought. In exchange we have to deal with nonsmoothness and with semi-infinity, and those are addressed during the next sections. □

First-order nonsmooth optimization methods for (3) have for the first time been proposed in [1], and have been further developed in [2, 4, 3]. Here we propose a second-order approach, which when combined with the first-order technique speeds up the minimization process at the end, improves the final precision, and serves as an additional convergence certificate.

3 Approach via semi-infinite programming

In the previous section we have introduced program (3) for the case of static controllers, and we shall continue to do so because the general case of dynamic K may be reduced to the static case by prior dynamic augmentation of the system. More precisely, the following substitutions are performed:

$$\begin{array}{llll} K \to \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}, & A \to \begin{bmatrix} A & 0 \\ 0 & 0_k \end{bmatrix} & B_1 \to \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, & C_1 \to \begin{bmatrix} C_1 & 0 \end{bmatrix} \\ B_2 \to \begin{bmatrix} 0 & B_2 \\ I_k & 0 \end{bmatrix}, & C_2 \to \begin{bmatrix} 0 & I_k \\ C_2 & 0 \end{bmatrix}, & D_{12} \to \begin{bmatrix} 0 & D_{12} \end{bmatrix}, & D_{21} \to \begin{bmatrix} 0 \\ D_{21} \end{bmatrix}. \end{array} \tag{5}$$

Now we have the following

Lemma 1. *For a closed-loop stabilizing controller K, the set of active frequencies $\Omega(K) := \{\omega \in [0,\infty] : f(K) = f(K,\omega)\}$ is either finite, or $\Omega(K) = [0,\infty]$, that is, $f(K,\omega) = f(K)$ for all ω.*

Proof. Notice that as K is fixed, we are concerned with a one parameter family $\omega \mapsto T(K,\omega)^H T(K,\omega)$ of $m_1 \times m_1$ Hermitian matrices. Here the bundle of the m_1 eigenvalues $\lambda_i\left(T(K,\omega)^H T(K,\omega)\right)$ as functions of ω has a very specific structure: it consists of m_1 real analytic functions $\phi_i(\omega)$, $\omega \in \mathbb{R}$; see e.g. [21, p. 82, p. 138], or [26]. More precisely, there are m_1 real analytic functions $\phi_1, \ldots, \phi_{m_1}$ such that

$$\{\sigma_1(T(K,\omega)), \ldots, \sigma_{m_1}(T(K,\omega))\} = \{|\phi_1(\omega)|, \ldots, |\phi_{m_1}(\omega)|\}, \tag{6}$$

where $\sigma_i(T(K,\omega))$ are the singular values of $T(K,\omega)$. In consequence, the maximum eigenvalue function (2) is the maximum of m_1 real analytic functions $\phi_i^2(\omega)$. These ϕ_i^2 are also analytic at ∞, because we are dealing with a stable (proper) transfer matrix $T(K,\omega)$, which is rational as a function of ω. (The statement above may be made more precise: There exist functions ϕ_i which are analytic on a strip $\mathcal{B}$ on the Riemann sphere $\mathbb{S}^2 = \mathbb{C} \cup \{\infty\}$, such that $\mathcal{B}$ contains the meridian $\mathbb{S}^1 = \mathbb{R} \cup \{\infty\}$ passing through the north pole ∞, with ϕ_i taking real values on $\mathbb{S}^1$, such that (6) is satisfied for all $\omega \in \mathbb{S}^1$.)

Suppose now $\Omega(K)$ is infinite. Then one of these m_1 real analytic functions ϕ_i^2 has an infinity of maxima on $\mathbb{S}^1$ with the same value $f(K)$. Since $\mathbb{S}^1$ is compact, these maxima have an accumulation point $\bar{\omega}$. In terms of the analytic extensions on $\mathcal{B}$, that means the Taylor expansion of the ϕ_i^2 in question at $\bar{\omega}$ is the same as the Taylor expansion of the function with constant value $f(K)$. This implies indeed $\phi_i^2(\omega) = f(K)$ for all ω. □

Remark. 1) The stable transfer function $\phi : s \mapsto \frac{s-1}{s+1}$ satisfies $\phi(j\omega)\phi(-j\omega) = 1$, so it is theoretically possible that $\Omega(K) = [0,\infty]$. This may even arrive in practice: It is known that the so-called central (full order) optimal H_∞-controller K_∞ renders the frequency plot $\omega \mapsto f(K_\infty,\omega)$ flat or *all-pass*.

2) Notice also that we cannot omit the frequency $\omega = \infty$ in the discussion, as witnessed by the stable transfer function $\phi(s) : s \mapsto \frac{s+1}{s+2}$. Here $\max_{\omega\in[0,\infty]} \phi(j\omega)\phi(-j\omega) = 1$ is attained at $\omega = \infty$.

3) The frequencies $\omega \in \Omega(K)$ will be called the peaks of the frequency curve at K. Local maxima of the frequency curve $\omega \mapsto f(K,\omega)$ which are not peaks will be called secondary peaks. Notice that the above argument also shows that the number of secondary peaks is finite. □

For the technical formulas we shall concentrate on those K, where the set of active frequencies or peaks $\Omega(K) = \{\omega \in [0,\infty] : f(K) = f(K,\omega)\}$ is finite. In practice we have never observed flat frequency curves when the order of the controller is smaller than the order of the plant.

During the following we will analyze the case where the multiplicity of $\lambda_1(T(K,\omega)^H T(K,\omega))$ is one at all active frequencies ω. This is motivated by practical considerations, because nonsmoothness (b) never occurred in our tests. The necessary changes required for the general case will be discussed in Section 4.3.

It is useful to cast program (3) as an equivalent constrained semi-infinite program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & f(K,\omega) - t \leq 0,\ \omega \in [0,\infty] \end{array} \tag{7}$$

with decision variable (K,t).

In [18] three approaches to semi-infinite programming are discussed: exchange of constraints, discretization, and local reduction. Here we use a local reduction method. The main ideas are recalled below; see also [20] for this approach.

Let $(\bar{K},\bar{t})$, with $\bar{t} = f(\bar{K})$, be a local solution of (7). Indexing the active frequencies $\Omega(\bar{K}) = \{\bar{\omega}_1,\ldots,\bar{\omega}_p\}$ at $\bar{K}$, we suppose that the following conditions are satisfied

(i) $f'_\omega(\bar{K},\bar{\omega}_i) = 0, \quad i = 1,\ldots,p.$

(ii) $f''_{\omega\omega}(\bar{K},\bar{\omega}_i) < 0, \quad i = 1,\ldots,p.$

(iii) $f(\bar{K},\omega) < f(\bar{K}), \quad$ for every $\omega \notin \Omega(\bar{K}) = \{\bar{\omega}_1,\ldots,\bar{\omega}_p\}.$

These assumptions define the setting denoted as the "standard case" in semi-infinite programming [18]. The three conditions express the fact that the frequencies $\bar{\omega}_i \in \Omega(\bar{K})$ are the strict global maximizers of $f(\bar{K},\cdot)$. Notice that condition (iii) is the finiteness hypothesis already mentioned, justified by Lemma 1, while condition (ii) is slightly conservative, because the necessary optimality condition only tells us that $f''_{\omega\omega}(\bar{K},\bar{\omega}_i) \leq 0$.

Remark. Notice that $f(\bar{K},\cdot)$ is twice differentiable at each $\bar{\omega}_i$ even without the hypothesis that the maximum singular values of the $T(K,\bar{\omega}_i)$ have multiplicity 1. This is because the maximum singular value function $\omega \to \bar{\sigma}(G(j\omega))$ of a stable transfer matrix G is always twice continuously differentiable at its local maxima [9]. □

Conditions (i) and (ii) allow the implicit function theorem, according to which we can find a neighborhood U of $\bar{K}$, and neighborhoods V_i of $\bar{\omega}_i$

$(i = 1, \ldots, p)$, together with C^1-functions $\omega_i : U \to V_i$, such that the following conditions are satisfied

(iv) $\omega_i(\bar{K}) = \bar{\omega}_i$, $i = 1, \ldots, p$.

(v) $f'_\omega(K, \omega_i(K)) = 0$, $i = 1, \ldots, p$.

(vi) Whenever $K \in U$ and $\omega \in V_i$ satisfy $f'_\omega(K, \omega) = 0$, then $\omega = \omega_i(K)$.

Condition (ii) says $f''_{\omega\omega}(\bar{K}, \bar{\omega}_i) < 0$, so by shrinking U if required, we may arrange that $f''_{\omega\omega}(K, \omega_i(K)) < 0$ for $K \in U$. Then $\omega_i(K)$ are local maxima of $f(K, \cdot)$. Moreover, by (vi), $\omega_i(K)$ is the only critical point of $f(K, \cdot)$ in V_i. We then have the following

Lemma 2. *Under conditions (i) – (iii) the neighborhood U of $\bar{K}$ may be chosen such that $\max_{\omega \in [0,\infty]} f(K, \omega) = \max_{i=1,\ldots,p} f(K, \omega_i(K))$ for every $K \in U$. In particular, $\Omega(K) \subset \{\omega_1(K), \ldots, \omega_p(K)\}$ for every $K \in U$.*

Proof. Choose U and V_i such that conditions (iv) - (vi) are satisfied. Suppose that contrary to the statement there exists a sequence $K_r \to \bar{K}$ such that

$$m_r := \max_{i=1,\ldots,p} f(K_r, \omega_i(K_r)) < \max_{\omega \in [0,\infty]} f(K_r, \omega) =: M_r.$$

Pick ω^r where the maximum M_r is attained. Passing to a subsequence, we may assume $\omega^r \to \omega^*$ for some $\omega^* \in [0, \infty]$. Since $m_r \to f(\bar{K})$ and $M_r \to f(\bar{K})$, we have $f(\bar{K}) = f(\bar{K}, \omega^*)$. By axiom (iii), that means $\omega^* = \bar{\omega}_i$ for some $i = 1, \ldots, p$. Then $(K_r, \omega^r) \in U \times V_i$ from some index r_0 onwards.

But $f'_\omega(K_r, \omega^r) = 0$, because ω^r is a maximum at K_r, so condition (vi) implies $\omega^r = \omega_i(K_r)$. That shows $m_r = M_r$ for $r \geq r_0$, a contradiction. □

Altogether, we have shown that program (3) is locally equivalent to the standard constrained nonlinear program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & f(K, \omega_i(K)) - t \leq 0,\ i = 1, \ldots, p \end{array} \tag{8}$$

which we may solve via a SQP method. In the next section we will discuss how this should be organized, and that the jet information may be computed efficiently. Local convergence of this approach will be assured under the following hypothesis

(vii) $f'_K(\bar{K}, \bar{\omega}_1), \ldots, f'_K(\bar{K}, \bar{\omega}_p)$ are linearly independent

because this guarantees that program (8) satisfies the linear independence constraint qualification hypothesis (LICQ).

4 Solving with SQP

4.1 Quadratic tangent subproblem

In this section we assume that for K in a neighborhood of $\bar{K}$, the set $\{\omega_1(K), \ldots, \omega_p(K)\}$ is available. The computation of this set will be discussed in Section 5.1.

In order to derive the tangent quadratic program for (8), let us write $G_i(K,t) = f\left(K, \omega_i(K)\right) - t$, $F(K,t) = t$. The Lagrangian of (8) is then

$$L(K,t;\tau) = F(K,t) + \sum_{i=1}^{p} \tau^i G_i(K,t),$$

so that

$$\begin{aligned} L'_K(K,t;\tau) &= \sum_{i=1}^{p} \tau^i \left(f'_K\left(K, \omega_i(K)\right) + f'_\omega\left(K, \omega_i(K)\right) \omega'_i(K) \right) \\ &= \sum_{i=1}^{p} \tau^i f'_K\left(K, \omega_i(K)\right) \end{aligned}$$

using condition (v) above. Similarly,

$$L'_t(K,t;\tau) = 1 - \sum_{i=1}^{p} \tau^i.$$

The second-order elements are $L''_{Kt}\left(K,t;\tau\right) = 0$, $L''_{tt}\left(K,t;\tau\right) = 0$, and

$$\begin{aligned} L''_{KK}(K,t;\tau) &= \sum_{i=1}^{p} \tau^i [f''_{KK}\left(K, \omega_i(K)\right) \\ &\quad + f'_\omega\left(K, \omega_i(K)\right) \omega''_i(K) \\ &\quad\quad + f''_{\omega K}\left(K, \omega_i(K)\right) \omega'_i(K)^\top] \\ &= \sum_{i=1}^{p} \tau^i \left[f''_{KK}\left(K, \omega_i(K)\right) + f''_{\omega K}\left(K, \omega_i(K)\right) \omega'_i(K)^\top \right], \end{aligned}$$

again using condition (v). Differentiating (v) gives

$$0 = f''_{\omega K}\left(K, \omega_i(K)\right) + f''_{\omega\omega}\left(K, \omega_i(K)\right) \omega'_i(K),$$

which allows us to express $\omega'_i(K)$ through derivatives of f. Altogether,

$$L''(K,t;\tau) = \begin{bmatrix} L''_{KK}(K,t;\tau) & 0 \\ 0 & 0 \end{bmatrix}$$

with

$$L''_{KK}(K,t;\tau) = \sum_{i=1}^{p} \tau^i [f''_{KK}\left(K,\omega_i(K)\right) \\ - f''_{\omega K}\left(K,\omega_i(K)\right) f''_{\omega\omega}\left(K,\omega_i(K)\right)^{-1} f''_{\omega K}\left(K,\omega_i(K)\right)^{\top}].$$

The tangent quadratic program is now

$$\begin{array}{ll} \text{minimize} & \delta t + \frac{1}{2}\delta K^{\top} L''_{KK}(K,t;\tau)\delta K \\ \\ \text{subject to} & f\left(K,\omega_i(K)\right) + f'_K\left(K,\omega_i(K)\right)^{\top}\delta K - t - \delta t \leq 0, \\ & i = 1,\ldots,p \\ & \delta t \in \mathbb{R}, \delta K \in \mathbb{R}^{m_2 \times p_2} \end{array} \tag{9}$$

4.2 First and second derivative formulas

We first recall some useful results from matrix perturbation theory, concerning the first derivatives of a non-degenerate eigenvalue and an associated eigenvector. General formulae may be found in [17]. We specialize to the case of a Hermitian matrix depending on real parameters: given $A(u)$, a $n \times n$ complex Hermitian matrix depending smoothly on $u \in \mathbb{R}^p$, we denote by $\lambda_1(u) \geq \cdots \geq \lambda_n(u)$ the eigenvalues of $A(u)$, and by $\big(q_1(u),\ldots,q_n(u)\big)$ an orthonormal basis of eigenvectors, such that

$$\begin{cases} A(u) = Q(u)\Lambda(u)Q(u)^H, \\ \Lambda(u) = \operatorname{diag}\big(\lambda_1(u),\ldots,\lambda_n(u)\big) \text{ and} \\ Q(u) = \big[q_1(u),\ldots,q_n(u)\big]. \end{cases}$$

For a fixed k and a given $u_0 \in \mathbb{R}^p$, under the assumption that $\lambda_k(u_0)$ is a simple (non-degenerate) eigenvalue of $A(u_0)$, λ_k and q_k are differentiable at u_0, and for all δu in $\mathbb{R}^p$, we have

$$\begin{aligned} \lambda'_k(u_0).\delta u &= q_k(u_0)^H \big(A'(u_0).\delta u\big) q_k(u_0) \\ q'_k(u_0).\delta u &= \sum_{\substack{i=1 \\ i\neq k}}^{n} \frac{q_i(u_0)^H \big(A'(u_0).\delta u\big) q_k(u_0)}{\lambda_k(u_0) - \lambda_i(u_0)}\ q_i(u_0) \end{aligned}$$

Notice from the second equality that $q'_k(u_0).\delta u$ is orthogonal to the eigenvector $q_k(u_0)$.

Furthermore we derive from these expressions that λ_k is twice differentiable at u_0, and for all δu_1, δu_2 in $\mathbb{R}^p$, dropping the dependency on u_0 for the right hand side terms, we get

$$\lambda_k''(u_0).(\delta u_1, \delta u_2) = q_k^H \left(A''.(\delta u_1, \delta u_2)\right) q_k + 2 \text{ Re } \left(\sum_{\substack{i=1 \\ i \neq k}}^{n} \frac{q_k^H \left(A'.\delta u_1\right) q_i q_i^H \left(A'.\delta u_2\right) q_k}{\lambda_k - \lambda_i} \right).$$

We specialize to the case of $f(K, \omega)$, the squared maximum singular value of the transfer matrix $T(K, \omega)$. Then $u = (K, \omega)$ and $A(u) = A(K, \omega) = T(K, \omega)^H T(K, \omega)$, and $f = \lambda_1 \circ A$.

We note eigenvalues of $T(K, \omega)^H T(K, \omega)$ as $(\lambda_i)_{1 \leq i \leq m_1}$ and an orthonormal basis of associated eigenvectors as $(q_i)_{1 \leq i \leq m_1}$. We drop the dependency on K and ω for ease of notation. We assume that $\lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_{m_1}$. Applying the above formulae gives:

$$f_K'(K, \omega).\delta K = 2 \text{ Re } \left(q_1^H T^H (T_K'.\delta K) q_1\right)$$
$$f_\omega'(K, \omega) = 2 \text{ Re } \left(q_1^H T^H T_\omega' q_1\right)$$

$$f_{KK}''(K, \omega).(\delta K_1, \delta K_2) = 2 \text{ Re } \left[q_1^H \left((T_K'.\delta K_2)^H T_K'.\delta K_1 + T^H T_{KK}''.(\delta K_1, \delta K_2) \right) q_1 + \sum_{i=2}^{m_1} \frac{q_1^H \left(T^H T_K'.\delta K_1 + (T_K'.\delta K_1)^H T\right) q_i q_i^H \left(T^H T_K'.\delta K_2 + (T_K'.\delta K_2)^H T\right) q_1}{\lambda_1 - \lambda_i} \right]$$

$$f_{\omega\omega}''(K, \omega) = 2 \text{ Re } \left[q_1^H \left({T_\omega'}^H T_\omega' + T^H T_{\omega\omega}'' \right) q_1 \right] + 2 \sum_{i=2}^{m_1} \frac{\left| q_1^H \left(T^H T_\omega' + {T_\omega'}^H T\right) q_i \right|^2}{\lambda_1 - \lambda_i}$$

$$f_{\omega K}''(K, \omega).\delta K = 2 \text{ Re } \left[q_1^H \left((T_K'.\delta K)^H T_\omega' + T^H T_{\omega K}''.\delta K \right) q_1 + \sum_{i=2}^{m_1} \frac{q_1^H \left(T^H T_\omega' + {T_\omega'}^H T\right) q_i q_i^H \left(T^H T_K'.\delta K + (T_K'.\delta K)^H T\right) q_1}{\lambda_1 - \lambda_i} \right]$$

The first and second partial derivatives of the transfer function T can be expanded with the closed-loop state-space matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. Starting out with formula (1), we derive

$$\begin{aligned} T'_K.\delta K &= T'_K(K,\omega).\delta K \\ &= \Big(D_{12} + \mathcal{C}\big(j\omega I_n - \mathcal{A}\big)^{-1}B_2\Big)\delta K\Big(C_2\big(j\omega I_n - \mathcal{A}\big)^{-1}\mathcal{B} + D_{21}\Big) \end{aligned}$$

$$\begin{aligned} T''_{KK}.(\delta K_1, \delta K_2) &= T''_{KK}(K,\omega).(\delta K_1, \delta K_2) \\ &= \Big(D_{12} + \mathcal{C}\big(j\omega I_n - \mathcal{A}\big)^{-1}B_2\Big) \\ &\quad \Big(\delta K_2 C_2\big(j\omega I_n - \mathcal{A}\big)^{-1}B_2\delta K_1 + \delta K_1 C_2\big(j\omega I_n - \mathcal{A}\big)^{-1}B_2\delta K_2\Big) \\ &\quad \Big(D_{21} + C_2\big(j\omega I_n - \mathcal{A}\big)^{-1}\mathcal{B}\Big) \end{aligned}$$

$$\begin{aligned} T'_\omega &= T'_\omega(K,\omega) & T''_{\omega\omega} &= T''_{\omega\omega}(K,\omega) \\ &= -j\ \mathcal{C}\big(j\omega I_n - \mathcal{A}\big)^{-2}\mathcal{B} & &= -2\ \mathcal{C}\big(j\omega I_n - \mathcal{A}\big)^{-3}\mathcal{B} \end{aligned}$$

$$\begin{aligned} T''_{\omega K}.\delta K &= T''_{\omega K}(K,\omega).\delta K \\ &= -j\ \Big(\Big(D_{12} + \mathcal{C}\big(j\omega I_n - \mathcal{A}\big)^{-1}B_2\Big)\delta K\Big(C_2\big(j\omega I_n - \mathcal{A}\big)^{-2}\mathcal{B}\Big) \\ &\quad + \Big(\mathcal{C}\big(j\omega I_n - \mathcal{A}\big)^{-2}B_2\Big)\delta K\Big(D_{21} + C_2\big(j\omega I_n - \mathcal{A}\big)^{-1}\mathcal{B}\Big)\Big) \end{aligned}$$

Introducing the following transfer matrices

$$\begin{aligned} F(K,\omega) &= \big(j\omega I_n - \mathcal{A}(K)\big)^{-1} & G_{12}(K,\omega) &= D_{12} + \mathcal{C}(K)F(K,\omega)B_2 \\ G_{21}(K,\omega) &= D_{21} + C_2F(K,\omega)\mathcal{B}(K) & G_{22}(K,\omega) &= C_2F(K,\omega)B_2. \end{aligned}$$

we finally get

$$\begin{aligned} T'_K.\delta K &= G_{12}\delta K G_{21} \\ T''_{KK}.(\delta K_1, \delta K_2) &= G_{12}\Big(\delta K_2 G_{22}\delta K_1 + \delta K_1 G_{22}\delta K_2\Big)G_{21} \\ T'_\omega &= -j\mathcal{C}F^2\mathcal{B} \\ T''_{\omega\omega} &= -2\mathcal{C}F^3\mathcal{B} \\ T''_{\omega K}.\delta K &= -j\Big(G_{12}\delta K C_2F^2\mathcal{B} + \mathcal{C}F^2B_2\delta K G_{21}\Big). \end{aligned}$$

Remark. Notice that the transfer matrices F, T, G_{12}, G_{21} and G_{22} can be reliably and efficiently evaluated at the active frequencies $\omega \in \Omega(K)$ with a Hessenberg method, as explained in [22]. □

Altogether, we have proved the following

Proposition 1 . *Suppose* $\lambda_1\left(T(K,\omega_i)^H T(K,\omega_i)\right)$ *has multiplicity 1 for every* $\omega_i \in \{\omega_1(K),\ldots,\omega_p(K)\}$. *Then the frequency response* $(w_1,w_2) \to (z_1,z_2)$ *of the plant*

$$\begin{bmatrix}\dot{x}\\ z_1\\ z_2\end{bmatrix} = \begin{bmatrix}\mathcal{A}(K) & \mathcal{B}(K) & B_2\\ \mathcal{C}(K) & \mathcal{D}(K) & D_{12}\\ C_2 & D_{21} & 0_{p_2\times m_2}\end{bmatrix}\begin{bmatrix}x\\ w_1\\ w_2\end{bmatrix},$$

can be used to compute all the jet elements of tangent program (9). Indeed, it suffices to partition in order to get T, G_{12}, G_{21} *and* G_{22}*:*

$$\begin{aligned} T(K,\omega) &= T_{w_1\to z_1}(K,j\omega) & \qquad G_{12}(K,\omega) &= T_{w_2\to z_1}(K,j\omega)\\ G_{21}(K,\omega) &= T_{w_1\to z_2}(K,j\omega) & \qquad G_{22}(K,\omega) &= T_{w_2\to z_2}(K,j\omega)\end{aligned}$$

□

Similar formulae for the derivatives of the transfer function T are obtained for dynamic controllers, using the substitutions (5). Proposition 1 has the practical consequence that the jet information for (9) may be computed using the MATLAB control tool environment. Our experience shows that this works efficiently even for systems with several hundreds of states.

4.3 Multiple eigenvalues

The working hypothesis of the previous section was that leading eigenvalues $\lambda_1\left(T(K,\omega_i(K))^H T(K,\omega_i(K))\right)$ had multiplicity 1 for all frequencies in the set $\{\omega_1(K),\ldots,\omega_p(K)\}$ and for all K in a neighborhood of $\bar{K}$. This hypothesis is motivated by our numerical experience, where we have never encountered multiple eigenvalues. This is clearly in contrast with experience in pure eigenvalue optimization problems, where multiplicity of limiting elements is > 1 *as a rule*, causing failure of those numerical methods which assume smoothness of the maximum eigenvalue function.

Nonetheless, our approach is still functional if the hypothesis of single eigenvalues at peak frequencies is dropped. What is needed is the weaker assumption that the eigenvalue multiplicities $\bar{r}_i$ at the limit point $\bar{K}$ are known for all active frequencies $\bar{\omega}_i$, $i=1,\ldots,p$. More precisely, $\lambda_1\left(T(\bar{K},\bar{\omega}_i)^H T(\bar{K},\bar{\omega}_i\right)$ has multiplicity $\bar{r}_i$, $i=1,\ldots,p$, and for K sufficiently close to $\bar{K}$, we have a mechanism to reliably guess $\bar{r}_i$, based solely on the information at the current iterate K.

This situation has been discussed by several authors, see e.g. [15, 31, 32, 29]. With $\bar{r}_i$ known, SQP can still be applied as follows. Consider $X \in \mathbb{S}^{m_1}$ where $\lambda_1(X)$ has multiplicity r. We replace the maximum eigenvalue function

λ_1 by the average of the first r eigenvalues

$$\hat{\lambda}_r(X) = \frac{1}{r}\sum_{\nu=1}^{r} \lambda_\nu(X)\,.$$

This function is smooth in a neighborhood of the smooth manifold

$$\mathcal{M}_r = \{X \in \mathbb{S}^{m_1} : \lambda_1(X) = \cdots = \lambda_r(X) > \lambda_{r+1}(X)\}$$

of Hermitian matrices $X \in \mathbb{S}^{m_1}$ with eigenvalue multiplicity r, and $\lambda_1 = \hat{\lambda}_r$ on $\mathcal{M}_r$. The manifold has co-dimension $d := \frac{r(r+1)}{2} - 1$ in $\mathbb{S}^{m_1}$, and in a neighborhood of $\bar{X}$ may be described by d equations $h_1(X) = 0, \ldots, h_d(X) = 0$. The tangent space of $\mathcal{M}_r$ can be computed, see [35, 32], and an SQP approach may be derived, if hypotheses (i) and (ii) are suitably modified. For pure eigenvalue optimization, this has been discussed in [29]. The extension to the semi-infinite case is clear under the finiteness assumption (iii). Introducing natural analogues of conditions (i), (ii), see for instance [18], we may then approach minimization of the H_∞-norm along the same lines and obtain the finite program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \hat{\lambda}_{\bar{r}_i}\left(T(K,\omega_i(K))^H T(K,\omega_i(K))\right) \le t \\ & T(K,\omega_i(K))^H T(K,\omega_i(K)) \in \mathcal{M}_{\bar{r}_i} \\ & i = 1,\ldots,p \end{array}$$

with decision variables (K, t). Its tangent quadratic program uses the explicit form of the tangent space $T(\mathcal{M}_r, X)$ given in [35].

The case of multiple eigenvalues therefore requires two separate estimation steps. As before we have to select p primary and secondary peaks. Then we have to estimate the eigenvalue multiplicity for each $T(K,\omega_i)^H T(K,\omega_i)$. There exist indeed practically useful ways to estimate the limiting multiplicity $\bar{r} = \bar{r}_i$ of λ_1 for each frequency ω_i. For instance, with a small threshold $\tau > 0$, a guess r of $\bar{r}$ could be obtained by letting

$$\lambda_1 - \lambda_r < \tau \max\{1, |\lambda_1|\}, \quad \lambda_1 - \lambda_{r+1} \ge \tau \max\{1, |\lambda_1|\}.$$

We leave the details of this extension to the care of the reader.

4.4 Globalisation via trust-region

Independently of whether or not the eigenvalue multiplicity at the active and nearly active frequency peaks is 1, the local program (8) remains dependent

on the characteristics of the local solution $(\bar{K}, \bar{t})$ of (7), because the number p of primary and secondary peaks may vary. Using Helly type theorems, an upper bound for p may be derived, see e.g. [18, Thm. 4.2]. For a controller K of order k we have $p \leq (m_1 + k)(p_1 + k)$ peaks, but this is pessimistic as a rule. In order to stabilize our method, we will therefore use a trust region strategy.

In order to control the step quality computed at a given primal-dual pair $(K, t; \tau)$, we use the following ℓ_1-merit functions ϕ_1 and ψ_1, respectively associated with the local program (8) and the tangent program (9);

$$\begin{aligned}
\phi_1(K, t; \mu) &= t + \frac{1}{\mu} \sum_{i=1}^{p} \left[f\left(K, \omega_i(K)\right) - t \right]^+ \\
\psi_1(\delta K, \delta t; \mu) &= \delta t + \frac{1}{2} \delta K^\top L''_{KK}(K, t; \tau) \delta K + \\
&\quad \frac{1}{\mu} \sum_{i=1}^{p} [\delta K^\top \nabla f\left(K, \omega_i(K)\right) - \delta t \\
&\quad + f\left(K, \omega_i(K)\right) - t]^+.
\end{aligned}$$

The agreement between the actual reduction and the predicted reduction is measured by the ratio

$$\rho = \frac{\phi_1(K, t; \mu) - \phi_1(K + \delta K, t + \delta t; \mu)}{\psi_1(0; \mu) - \psi_1(\delta K, \delta t; \mu)}. \tag{10}$$

Then the trust-region radius is managed according to the following algorithm, based on a basic trust-region algorithm from [12].

Fix $0 < \eta_1 \leq \eta_2 < 1$, $0 < \theta < 1$.

1. **Initialize**. Find an initial closed-loop stabilizing controller K_0. Fix $t_0 = f(K_0)$, p_0 and $\tau_0 \in \mathbb{R}_+^{p_0}$, and set counter $k = 0$.
2. **Peak estimation.** Given current K_k and $\tau_k \in \mathbb{R}_+^{p_{k-1}}$, estimate number p_k and positions of primary and secondary peaks $\{\omega_1(K_k), \ldots, \omega_{p_k}(K_k)\}$.
3. **Model inconsistency.** If p_k differs from p_{k-1}, or if last step taken was Cauchy step, then modify old $\tau_k \in \mathbb{R}_+^{p_{k-1}}$ or create a new consistent multiplier vector $\tau_k \in \mathbb{R}_+^{p_k}$.
4. **Initialize Newton method**. Put $K_{k,0} = K_k$, $\tau_{k,0} = \tau_k$.
   Set $\Delta_0 = 1$ if $k = 0$ or if $k \geq 1$ and $K_k = K^C$
   Otherwise, set $\Delta_0 = \Delta_{j+1}$.
   Set counter $j = 0$.
5. **Newton step.** Use current iterate $K_{k,j}$, multiplier estimate $\tau_{k,j}$ and trust region radius Δ_j to solve (9) and generate Newton trial step $K_{k,j+1}$, with associated multiplier estimate $\tau_{k,j+1} \in \mathbb{R}_+^{p_k}$.
6. **Decision.** Compare predicted progress in the local quadratic model ψ_1 to progress in ϕ_1 using progress ratio ρ_{j+1} in (10). There are two cases:
   Step accepted: $\rho_{j+1} \geq \eta_1$. If even $\rho_{j+1} \geq \eta_2$ and $\|K_{k,j} - K_{k,j+1}\| = \Delta_j$, then double radius $\Delta_{j+1} = 2\Delta_j$. Otherwise keep $\Delta_{j+1} = \Delta_j$. Put $K^N = K_{k,j+1}$ and $p^N := f(K_k) - f(K^N)$. Goto step 7.
   Step refused: $\rho_{j+1} < \eta_1$. Put $\Delta_{j+1} = \Delta_j/2$, increase counter j, and go back to step 5.
7. **Cauchy step.** Given the current iterate K_k, compute a Cauchy step K^C away from K_k using the first-order method described in [2]. Let $p^C := f(K_k) - f(K^C) \geq 0$ be first-order progress. If $p^N \geq \theta p^C$ let $K_{k+1} = K^N$, and $\tau_{k+1} = \tau_{k,j+1}$, otherwise put $K_{k+1} = K^C$.
8. **Stopping test.** If accepted step $K_{k+1} \in \{K^N, K^C\}$ offers no progress over K_k, stop. Otherwise increase counter k and go back to step 2.

In the inner loop j, the penalty parameter μ_j for the expressions ϕ_1, ψ_1 is initialized at $j = 0$ with $\mu_0 = (\|\tau_1\|_\infty + \alpha)^{-1}$ and updated according to the following rule for $j \geq 1$, with a chosen constant $\alpha > 0$

$$\mu_j = \begin{cases} \mu_{j-1} & \text{if } \mu_{j-1}^{-1} \geq \|\tau_{j+1}\|_\infty + \alpha, \\ (\|\tau_{j+1}\|_\infty + 2\alpha)^{-1} & \text{otherwise,} \end{cases}$$

In this way, $\mu_j < \|\tau_{j+1}\|_\infty^{-1}$ and the ℓ_1 merit function ψ_1 is exact (see [28]).

Notice that the trust region procedure in the inner loop j between steps 4 and 6 follows standard lines, but is based on the guess p and $\{\omega_1(K), \ldots, \omega_p(K)\}$ of model (8), so a few comments are in order here. Namely, since the model may be incorrect, the following phenomenon may be observed. The Newton step may be successful with regard to (8), i.e., with regard to the inner

loop, but may nevertheless fail when matched with reality in step 7. This is when the first order step K^C takes over. In the worst case, our method therefore converges with the speed of the underlying first-order technique [2]. Alternative first-order methods which could be used instead are for instance [33, 27, 4, 3].

A second phenomenon, which also arises due to the necessity to guess p, is addressed in step 3. It may happen that the new $K_{k+1} \in \{K^N, K^C\}$ is no longer consistent with the old model used in the previous step, because the number p had to undergo a change, or because a first-order step K^C had to be taken. The multiplier estimate $\tau_{k+1} = \tau_{k,j+1}$ from the last instance of step 6 is then of little use. We then restart multipliers afresh, or we recycle the old ones.

Finally, when p and $\omega_i(K)$ have been estimated correctly, the quadratic model will ultimately produce steps with quadratic progress. This means the test in step 8 will ultimately accept the Newton step, showing that our method has a fair chance to give local quadratic convergence.

Notice that two kinds of stopping tests are needed. We have to apply a standard second order test in order to halt the inner loop j if too small a trust region radius (indicating failure) arises. But even when the inner loop j is stopped with a successful step K^N, the first-order step K^C may override this decision in cases where the model was incorrect. It may even happen that the Newton model alerts a local minimum ($K^N = K_k$), but K^C allows to decrease the function value further. These cases are covered by the test in step 8.

5 Technical aspects

5.1 Identifying peak frequencies

It is important to observe that our approach differs substantially from standard semi-infinite programming methods in so far as we have a highly efficient procedure to evaluate the H_∞ norm of the transfer channel at a fixed closed-loop stabilizing controller $K \in \mathbb{R}^{(m_2+k)\times(p_2+k)}$.

Computing peak frequencies can be based on a classical algorithm for estimating the L_∞ norm of a transfer matrix $G(s) = D + C(sI - A)^{-1}B$ explained in detail in [9]; see also the variations in [9, 11, 16]. This algorithm detects in the first place the peak frequencies $\Omega(K)$, but may also be used to estimate secondary peaks. Its basic version is the following:

Computing $\|G\|_\infty$

1. **Initialization.** Find initial γ such that $\overline{\sigma}(D) \leq \gamma \leq \|G\|_\infty$. Fix tolerance $\epsilon > 0$.
2. **Step.** For current γ, find the frequency intervals $I_1, \ldots, I_\ell$ where $\overline{\sigma}(G(j\omega)) > \gamma$. If none is found, return the current γ.
3. **Update.** Otherwise let ω_k be the midpoint of I_k. Update $\gamma^+ = (1+\epsilon)\max_{k=1,\ldots,\ell} G(j\omega_k)$. Go back to step 2.

The returned value γ satisfies $\gamma \leq \|G\|_\infty < (1+\epsilon)\gamma$ on exit, which guarantees a relative tolerance ϵ for $\|G\|_\infty$ as well as for the positions of the peak frequencies. Secondary peaks may be estimated in several ways.

A first idea is to detected them on the way. Each interval I_k occurring in step 2 above contains one or several primary or secondary peaks. If an interval I_k at level γ does not have a successor at level γ^+, we conclude that a secondary peak $\omega_i \in I_i$ with $f(K, \omega_i) \in [\gamma, \gamma^+)$ has been left behind. Fixing a threshold $\alpha \in (0,1)$, we keep those secondary peaks ω_i with value $f(K, \omega_i) > \alpha\|G\|_\infty$. Since the values and slopes of the curve $\omega \mapsto f(K, \omega)$ at the endpoints of the I_k are known, and since the peak value is in the range $[\gamma, \gamma^+)$, a polynomial estimation of the ω_i is usually quite accurate.

A second more direct way to estimate secondary peaks is the following: As soon as the above algorithm is stopped at the final γ, the imaginary eigenvalues of $H(\gamma)$ indicate primary peaks. After fixing the threshold $\alpha\gamma$, we keep those complex eigenvalues λ of $H(\gamma)$ which lie in the strip $\alpha\gamma \leq \mathrm{Re}\lambda < \gamma$, and choose $\omega = \mathrm{Re}\lambda$ as our set of secondary peaks. This method is elegant as it also allows to estimate p very reliably.

It is clear that the above methods can be combined, and more sophisticated ways to estimate secondary peaks can be imagined. However, it should be maintained that knowing $\omega_i(K)$ to a high precision if $f(K, \omega_i(K)) < f(K)$ is not necessary, as the constraint is currently inactive. The closer the secondary peaks come to being active, the higher the precision to which they are computed.

Detecting intervals I_k where $\overline{\sigma}(G(j\omega)) > \gamma$ is based on the Hamiltonian matrix

$$H(\gamma) = \begin{pmatrix} A & 0 \\ 0 & -A^\top \end{pmatrix} - \begin{pmatrix} 0 & B \\ C^\top & 0 \end{pmatrix} \begin{pmatrix} \gamma I & D \\ D^\top & \gamma I \end{pmatrix}^{-1} \begin{pmatrix} C & 0 \\ 0 & -B^\top \end{pmatrix}.$$

We use the fact that

Lemma 3 . *γ is a singular value of $G(j\omega)$ if and only if $j\omega$ is an eigenvalue of $H(\gamma)$.*

Remark. Our approach is robust with respect to the estimation of the cardinality p of the unknown set $\Omega(\bar{K})$. If we overestimate p, we create a program (8), where some constraints remain inactive near $\bar{K}$, a situation which is automatically dealt with by the SQP solver. What is needed to justify this situation theoretically is an implicit function for any of these additional secondary peaks. That may be assured by extending conditions (i) and (ii) to secondary peaks. □

5.2 Stopping criteria

We implemented two stopping tests in order to check convergence. The first of them checks criticality of the iterate (K_k, t_k) with multipliers τ_k, through the absolute test

$$\|L'_{(K,t)}(K_k, t_k; \tau_k)\| < \varepsilon_1,$$

where $\|\cdot\|$ is a norm on $\mathbb{R}^{m_2 \times p_2 + 1}$.

The second stopping condition checks the relative step length on the decision variables

$$\|(K_k, t_k) - (K_{k-1}, t_{k-1})\| = \|(\delta K, \delta t)\| < \varepsilon_2(1 + \|(K_{k-1}, t_{k-1})\|)$$

As some SQP iterates may become infeasible for problem (8), we also check if $f(K, \omega_i(K)) - t < \varepsilon$ for all $i = 1, \ldots, p$. The algorithm stops as soon as the current iterate is feasible and one of the stopping tests is satisfied.

6 Numerical results

The results presented here were obtained with the algorithm described in section 4.4. The plants are taken from [23], and the acronyms refer to that reference and allow to retrieve the data from the literature.

At every iteration, Newton and Cauchy steps are generated, and the one giving best progress is performed. In order to initialize the method, a closed-loop stabilizing controller is computed in the initial phase (step 1) of the algorithm. This first stage uses first-order nonsmooth techniques for minimizing the closed-loop spectral abscissa, as described in [8]. Alternatively, derivative-free methods have been successfully used for stabilization (see [1]) if the controller size is not too large. If no prior information on K is available, we start this preliminary stabilization phase at $K = 0$. Then the initial multipliers for SQP are set to $\tau_0^i = \frac{1}{p}$ for $i = 1, \ldots, p$.

Test example AC7 (transport airplane from [23]) has been given special attention, because it is possible to visualize iterates in controller space

(Figure 2). This example highlights the benefits of the second order algorithm (denoted by **NS2**) in terms of convergence speed, compared to the nonsmooth first-order method from [2] for H_∞ synthesis (denoted by **NS1**). As required in step 1, a closed-loop stabilizing static controller $K_0 = [4.5931 \quad 1.2164]$ is first computed. The initial closed-loop H_∞ performance is $\|T_{w\to z}(K_0)\|_\infty = 1.4614$, which after optimization is reduced to the optimal value $\|T_{w\to z}(\bar{K})\|_\infty = 6.5091 \cdot 10^{-2}$.
Table 1 shows that the gain of the second-order algorithm **NS2** over the existing method **NS1** is significant in terms of the number of iterations and CPU.

Alg.	iterations		CPU	H_∞	optimality	final
	Cauchy	Newton	time (s)	norm	criteria	$\|\Omega\|$
NS1	434	0	95.42	$6.5098 \cdot 10^{-2}$	$-9.7008 \cdot 10^{-6}$	2
NS2	2	11	12.23	$6.5091 \cdot 10^{-2}$	$9.4168 \cdot 10^{-6}$	2

Table 1: H_∞ synthesis: transport airplane (AC7)

Notice that the optimality criteria for the first and second order methods are different: a nonpositive optimality function for **NS1** (see [2] for details), and the norm of the Lagrangian gradient for **NS2** (as defined in section 5.2).

The resulting controllers are respectively $K_{\mathbf{NS1}} = [2.0331 \quad 1.8446 \cdot 10^{-3}]$ and $K_{\mathbf{NS2}} = [2.0330 \quad 1.9655 \cdot 10^{-3}]$. Both $K_{\mathbf{NS1}}$ and $K_{\mathbf{NS2}}$ have two active frequencies. In the case of $K_{\mathbf{NS2}}$ these are $\omega_1 = 1.9067$ and $\omega_2 = 1.3057 \cdot 10^{-1}$, in rad.s^{-1}, with almost linearly dependent gradients

$$f'_K(K_{\mathbf{NS2}}, \omega_1) = \begin{bmatrix} 3.2855 \cdot 10^{-3} \\ -4.6984 \cdot 10^{-3} \end{bmatrix},$$
$$f'_K(K_{\mathbf{NS2}}, \omega_2) = \begin{bmatrix} -3.9785 \cdot 10^{-3} \\ 5.6738 \cdot 10^{-3} \end{bmatrix}.$$

Figure 1 quantifies the convergence rates. Our second-order algorithm (**NS2**) starts with two Cauchy steps, then SQP takes over and converges superlinearly within only 11 Newton iterations. In contrast, with Cauchy steps only (**NS1**), a slow first-order convergence occurs. The iterates and contour lines of the H_∞ norm are drawn in Figure 2.

Table 2 shows various other tests from [23], illustrating that the second-order method can speed up convergence by decreasing the number of iterations. The extra work needed for a second-order iteration, that is, defining the tangent quadratic program (9), locating peak frequencies, and computing first and second derivatives to solve it, is usually compensated by the reduction of the number of steps. For plants where the first-order method gives little progress and iterates many times (like AC7 or HE2), the gain is

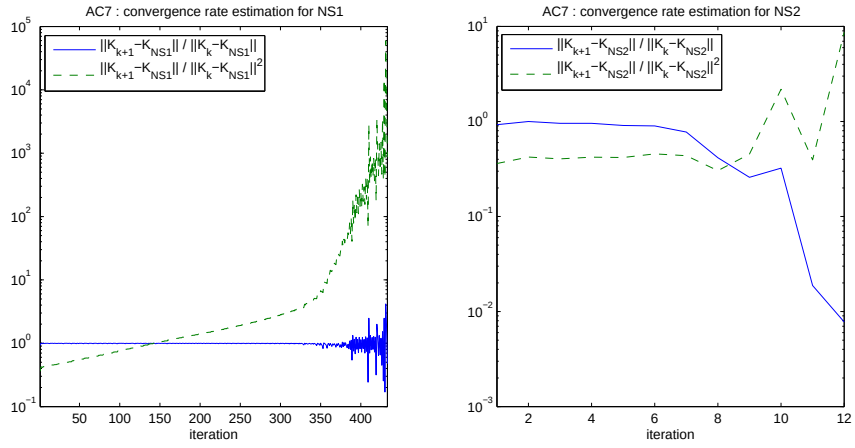

Figure 1: Convergence rates of **NS1** and **NS2** for AC7. **NS1** shows linear convergence with rate close to 1 (left). **NS2** gives quadratic convergence (right).

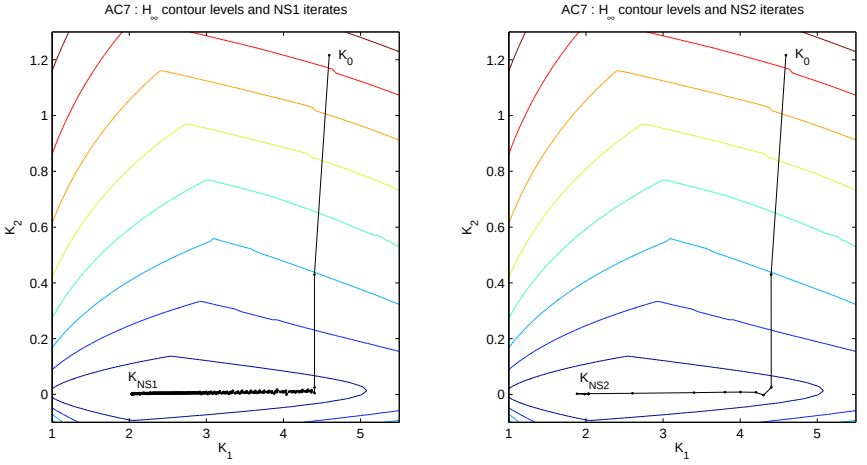

Figure 2: Iterates in controller space for **NS1** (left) and **NS2** (right). **NS2** is significantly faster despite the change of peaks caused by the non-smoothness.

dramatic. Other examples like REA3 show less significant gain or no gain at all, (AC6, AC8) in cases where no superlinear convergence occurs. Nonetheless, attempting second order steps is generally beneficial, because the final precision is improved (AC10).

An interesting case highlighting the importance of the final precision is the benchmark study Boeing 767 under flutter condition, which we have tested with two different scenarios, labeled AC10 and AC10*. In a first test,

plant	n	m_2	p_2	alg.	iter	CPU (s)	H_∞	$\lvert\Omega(K)\rvert$
AC6	7	2	4	**NS1**	79	30.28	4.1140	2
				NS2	25	41.45	4.1140	2
AC7	7	1	2	**NS1**	434	95.42	$6.5098 \cdot 10^{-2}$	2
				NS2	13	12.23	$6.5091 \cdot 10^{-2}$	2
AC8	9	1	5	**NS1**	14	23.50	2.0050	4
				NS2	9	37.20	2.0050	4
AC10	55	2	2	**NS1**	74	206.16	$1.3244 \cdot 10^{1}$	2
				NS2	35	187.47	$1.3236 \cdot 10^{1}$	5
AC10*	55	2	2	**NS1**	82	762.65	$1.3251 \cdot 10^{1}$	1
				NS2	71	877.18	$1.3237 \cdot 10^{1}$	5
HE2	4	2	2	**NS1**	746	196.20	4.2495	2
				NS2	12	14.67	4.2492	2
REA3	12	1	3	**NS1**	23	14.18	$7.4251 \cdot 10^{1}$	1
				NS2	9	11.06	$7.4251 \cdot 10^{1}$	1

Table 2: H_∞ synthesis: comparison of **NS1** and **NS2** on various plants. AC stands for aircraft models, HE for a helicopter model, and REA for a chemical reactor model.

AC10, model (3) is used. Here the first-order method **NS1** converges after 74 iterates. The second order method **NS2** needs 35 steps and converges

Alg.	iterations		CPU	H_∞	optimality	final
	Cauchy	Newton	time (s)	norm	criteria	dist
NS1	74	0	206.16	$1.3244 \cdot 10^{1}$	$-5.4287 \cdot 10^{-2}$	$3.4000 \cdot 10^{-5}$
NS2	28	7	187.47	$1.3236 \cdot 10^{1}$	$1.1926 \cdot 10^{-6}$	$3.3987 \cdot 10^{-5}$

Table 3: Comparison of **NS1** and **NS2** for large scale example Boeing 747 under flutter condition (AC10). Column 'dist' shows final distance to instability.

quadratically. Despite the higher cost for setting up and computing second order steps, this leads to a better CPU. What is more important in this example, however, is that the second order stopping test is much more stringent (second column from the right in Table 3) than for the first order method. This is particularly significant for the Boeing 767 example, because iterates of both **NS1** and **NS2** get extremely close to the hidden boundary of the stability region. The rightmost column of Table 3 gives the distance to instability of the two solutions, and iterates on the way have approximately the same distance. For **NS1** it is therefore impossible to decide whether the optimal solution is strictly inside or on the boundary of the stability region, because the final precision (the size of the steps before stopping) is comparable in magnitude to the distance to instability. In contrast, for **NS2**, the higher precision of the stopping test allows to conclude that the local minimum of **NS2** is indeed *inside* the stability region. In this example the **NS1** solution has only two active peaks, while **NS2** finds 5 peaks, which

gives further evidence that **NS1** indeed fails to converge, (even though the solution is still reasonable from a control point of view).

In this study the stability region turns out extremely narrow, which explains numerical difficulties with this example in the past. We have therefore performed a second test, labeled AC10*, where a stabilizing channel as in (4) is added to the objective (3). Not surprisingly, as the transfer channel is now much larger, this almost doubles CPU times, even though the number of steps is practically the same. The controllers obtained are fairly similar to those obtained for AC10, but with AC10* closed-loop stability of the limit $\bar{K}$ can now be *certified*. In the numerical testing the parameter ϵ in (4) was chosen such that for the initial iterate K_0, $\|T_{w\to z}(K_0)\|_\infty^2 = 4\epsilon\|(sI - \mathcal{A}(K_0))^{-1}\|_\infty^2$, which gave $\epsilon = 2.3150 \cdot 10^{-4}$.

We mention that adding the stabilizing channel does *not* resolve the intrinsic difficulty of this example, caused by the fact that the objective function in (3) or (4) is only defined on a narrow region.

7 Conclusion

We have developed and tested a second-order method to compute static and reduced-order H_∞-controllers and more general synthesis problems with structural constraints on the controller. Our approach avoids the use of Lyapunov variables and therefore leads to small or medium size optimization programs even for large systems (AC10 with 55 states). We use a methodology from semi-infinite programming to obtain a local nonlinear programming model and apply a trust region SQP method. A first-order nonsmooth spectral bundle method is used to initialize the second-order algorithm in the neighbourhood of a local solution. Our numerical testing indicates that speed of convergence and numerical reliability may be improved by using the second-order method.

References

[1] P. Apkarian and D. Noll. Controller design via nonsmooth multidirectional search. *SIAM Journal on Control and Optimization*, 44(6):1923–1949, 2006.

[2] P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control*, 51(1):71–86, 2006.

[3] P. Apkarian and D. Noll. Nonsmooth optimization for multidisk H_∞ synthesis. *European Journal of Control*, 12(3):229–244, 2006.

[4] P. Apkarian and D. Noll. Nonsmooth optimization for multiband frequency domain control design. *Automatica*, 43(4):724–731, 2007.

[5] P. Apkarian, D. Noll, J.-B. Thevenet, and H. D. Tuan. A spectral quadratic SDP method with applications to fixed-order H_2 and H_∞ synthesis. *European Journal of Control*, 10(6):527–538, 2004.

[6] P. Apkarian and H. D. Tuan. Robust control via concave minimization — local and global algorithms. *IEEE Transactions on Automatic Control*, 45(2):299–305, 2000.

[7] V. Balakrishnan, S. Boyd, and S. Balemi. Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. *International Journal of Robust and Nonlinear Control*, 1(4):295–317, 1991.

[8] V. Bompart, P. Apkarian, and D. Noll. Nonsmooth techniques for stabilizing linear systems. In *Proc. American Control Conf.*, New York, July 2007.

[9] S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm. *System and Control Letters*, 15(1):1–7, 1990.

[10] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15 of *SIAM Studies in Applied Mathematics*. SIAM, Philadelphia, 1994.

[11] N. A. Bruinsma and M. Steinbuch. A fast algorithm to compute the H_∞-norm of a transfer function matrix. *System Control Letters*, 14(5):287–293, 1990.

[12] A. Conn, N. Gould, and Ph. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization, Philadelphia, 2000.

[13] B. Fares, P. Apkarian, and D. Noll. An augmented Lagrangian method for a class of LMI-constrained problems in robust control. *International Journal on Control*, 74(4):348–360, 2001.

[14] B. Fares, D. Noll, and P. Apkarian. Robust control via sequential semidefinite programming. *SIAM Journal on Control and Optimization*, 40(6):1791–1820, 2002.

[15] R. Fletcher. Semi-definite matrix constraints in optimization. *SIAM Journal on Control and Optimization*, 23(4):493–513, 1985.

[16] P. Gahinet and P. Apkarian. Numerical computation of the L_∞-norm revisited. In *Proc. IEEE Conf. on Decision and Control*, pages 2257–2258, Tucson, 1992.

[17] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore and London, second edition, 1989.

[18] R. Hettich and K. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

[19] F. Jarre. An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices. *SIAM Journal on Control and Optimization*, 31(5):1360–1377, 1993.

[20] H. Th. Jongen, K. Meer, and E. Triesch. *Optimization theory*. Kluwer Academic Publisher, 2004.

[21] T. Kato. *Perturbation theory for linear operators*. Springer-Verlag, New York, second edition, 1980.

[22] A. Laub. Efficient multivariable frequency response computations. *IEEE Transactions on Automatic Control*, 26(2), 1981.

[23] F. Leibfritz. COMPl$_e$ib: COnstrained Matrix-optimization Problem *lib*rary — a collection of test examples for nonlinear semidefinite programs, control system design and related problems. Technical report, Universität Trier, 2003.

[24] F. Leibfritz and E. M. E. Mostafa. An interior-point constrained trust region method for a special class of nonlinear semi-definite programming problems. *SIAM Journal on Optimization*, 12(4):1048–1074, 2002.

[25] F. Leibfritz and E. M. E. Mostafa. Trust region methods for solving the optimal output feedback design problem. *International Journal of Control*, 76(5):501–519, 2003.

[26] A. G. J. MacFarlane and Y. S. Hung. Analytic properties of the singular value of a rational matrix. *International Journal of Control*, 37(2):221–234, 1983.

[27] D. Q. Mayne and E. Polak. Algorithms for the design of control systems subject to singular value inequalities. *IEEE Transactions on Automatic Control*, AC-20(4):546–548, 1975.

[28] J. Nocedal and S. Wright. *Numerical optimization*. Springer-Verlag, New York, 1999.

[29] D. Noll and P. Apkarian. Spectral bundle methods for nonconvex maximum eigenvalue functions: second-order methods. *Mathematical Programming Series B*, 104(2):729–747, 2005.

[30] D. Noll, M. Torki, and P. Apkarian. Partially augmented Lagrangian method for matrix inequality constraints. *SIAM Journal on Optimization*, 15(1):161–184, 2004.

[31] M. L. Overton. A quadratically convergent method for minimizing a sum of Euclidean norms. *Mathematical Programming*, 27:34–63, 1983.

[32] M. L. Overton. On minimizing the maximum eigenvalue of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications*, 9(2):256–268, 1988.

[33] E. Polak and Y. Wardi. A nondifferentiable optimization algorithm for the design of control systems subject to singular value inequalities over a frequency range. *Automatica*, 18(3):267–283, 1982.

[34] M. G. Safonov, K. C. Goh, and J. H. Ly. Control system synthesis via bilinear matrix inequalities. In *Proc. American Control Conf.*, pages 45–49, 1994.

[35] A. Shapiro. First and second order analysis of nonlinear semidefinite programs. *Mathematical Programming*, 77(2):301–320, 1997.

[36] J.-B. Thevenet, D. Noll, and P. Apkarian. Nonlinear spectral SDP method for BMI-constrained problems: applications to control design. In J. Braz, H. Araújo, A. Vieira, and B. Encarnação, editors, *Informatics in Control, Automation and Robotics I*, pages 61–72. Springer Verlag, 2006.

[37] H. D. Tuan, P. Apkarian, and Y. Nakashima. A new Lagrangian dual global optimization algorithm for solving bilinear matrix inequalities. *International Journal of Robust and Nonlinear Control*, 10(7):561–578, 2000.