Nonsmooth H_∞ Synthesis

Pierre Apkarian * *Dominikus Noll* ‡

Abstract

We develop nonsmooth optimization techniques to solve H_∞ synthesis problems under additional structural constraints on the controller. Our approach avoids the use of Lyapunov variables and therefore leads to moderate size optimization programs even for very large systems. The proposed framework is very versatile and can accommodate a number of challenging design problems including static, fixed-order, fixed-structure, decentralized control, design of PID controllers and simultaneous design and stabilization problems.

Our algorithmic strategy uses generalized gradients and bundling techniques suited for the H_∞-norm and other nonsmooth performance criteria. We compute descent directions by solving quadratic programs and generate steps via line search. Convergence to a critical point from an arbitrary starting point is proved and numerical tests are included to validate our methods.

Keywords: H_∞-synthesis, static output feedback, fixed-order synthesis, simultaneous stabilization, NP-hard problems, nonsmooth optimization, bundle methods, Clarke subdifferential, bilinear matrix inequality (BMI), linear matrix inequality (LMI).

1 Introduction

In this paper we consider H_∞-synthesis problems with additional structural constraints on the controller. This includes static and reduced-order H_∞-output feedback control, structured, sparse or decentralized synthesis, simultaneous stabilization problems, multiple performance channels, and much else. We propose to solve these problems with a nonsmooth optimization method exploiting the structure of the H_∞-norm.

In nominal H_∞-synthesis, feedback controllers are computed via semidefinite programming (SDP) [22, 2] or algebraic Riccati equations [18]. When structural constraints on the controller are added, the H_∞-synthesis problem is no longer convex. Some of the problems above have even been recognized as NP-hard [34] or as rationally undecidable [7]. These mathematical concepts indicate at least the inherent difficulty of H_∞-synthesis under constraints on the controller.

Even with structural constraints, the bounded real lemma may still be brought into play. The difference with customary H_∞ synthesis is that it no longer produces LMIs, but bilinear

*ONERA-CERT, Centre d'études et de recherche de Toulouse, Control System Department, 2 av. Edouard Belin, 31055 Toulouse, France - Email : `apkarian@cert.fr` - Tel : +33 5.62.25.27.84 - Fax : +33 5.62.25.27.64.

‡Université Paul Sabatier, Mathmatiques pour l'Industrie et la Physique, 118, route de Narbonne, 31062 Toulouse, France - Email : `noll@cict.fr` - Tel : +33 5.61.55.86.22 - Fax : +33 5.61.55.83.85.

matrix inequalities, BMIs, which are genuinely non-convex. Optimization code for BMI problems is currently developed by several groups, see e.g. [27, 4, 41, 33, 20], but it appears that the BMI approach runs into numerical difficulties even for problems of moderate size. This is mainly due to the presence of Lyapunov variables, whose number grows quadratically with the number of states.

Out present approach does *not* use the bounded real lemma and thereby avoids Lyapunov variables. This leads to moderate size optimization programs even for very large systems. In exchange, the cost functions are nonsmooth and require special optimization techniques. We evaluate the H_∞-norm via the Hamiltonian bisection algorithm [9, 8, 21] and exploit it further to compute subgradients, which are then used to compute descent steps. Notice, however, that our method is not a pure frequency domain method. In fact, it allows *both* frequency domain *and* state space domain parameterizations of the unknown controller. This makes it a very flexible tool in a number of situations of practical interest.

The structure of the paper is as follows. In section 2 we present the H_∞-synthesis problem and give several motivating examples. Section 3 computes subgradients of the H_∞-norm, which are then applied to closed-loop scenarios in section 4. In section 5 we start to develop our first-order descent method, which is completed in section 6. Section 6.7 discusses practical aspects of the method, and the final section 7 presents a number of experiments to validate our approach.

Notation

Let $\mathbb{M}_{n,m}$ the space of $n \times m$ matrices, equipped with the corresponding scalar product $\langle X, Y\rangle = \mathrm{Tr}(X^T Y)$, where X^T is the transpose of the matrix X, $\mathrm{Tr}\, X$ its trace. 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 shall use notion from nonsmooth analysis covered by [15].

2 H_∞ SYNTHESIS

The general setting of the H_∞ synthesis problem is as follows. We consider a linear time-invariant plant described in standard form by the state-space equations:

$$P(s): \qquad \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}, \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^{m_2}$ the vector of control inputs, $w \in \mathbb{R}^{m_1}$ a vector of exogenous inputs, $y \in \mathbb{R}^{p_2}$ the vector of measurements and $z \in \mathbb{R}^{p_1}$ the controlled or performance vector. Without loss of generality, it is assumed throughout that $D_{22} = 0$.

Let $u = K(s)y$ be a dynamic output feedback control law for the open loop plant (1), and let $T_{w\to z}(K)$ denote the closed-loop transfer function of the performance channel mapping w into z. Our aim is to compute $K(s)$ such that the following design requirements are met:

- *Internal stability:* For $w = 0$ the state vector of the closed-loop system (1) and (2) tends to zero as time goes to infinity.

- *Performance:* The H_∞ norm $\|T_{w\to z}(K)\|_\infty$ is minimized.

We assume that the controller K has the following frequency domain representation:

$$K(s) = C_K(sI - A_K)^{-1}B_K + D_K, \qquad A_K \in \mathbb{R}^{k\times k}, \tag{2}$$

where k is the order of the controller, and where the case $k = 0$ of a static controller $K(s) = D_K$ is included. Often practical considerations dictate additional challenging structural constraints. For instance it may be desired to design low-order controllers ($0 \le k \ll n$) or controllers with prescribed-pattern, sparse controllers, decentralized controllers, observed-based controllers, PID control structures, synthesis on a finite set of transfer functions, and much else. Formally, the synthesis problem may then be represented as

$$\begin{array}{ll} \text{minimize} & \|T_{w\to z}(K)\|_\infty \\ \text{subject to} & K \text{ stabilizes (1)} \\ & K \in \mathcal{K} \end{array} \tag{3}$$

where $K \in \mathcal{K}$ represents a structural constraint on the controller (2) like one of the above.

Without the restriction $K \in \mathcal{K}$, and under standard stabilizability and detectability conditions, it has become customary to synthesize $K(s)$ as follows. After substituting (2) into (1), the H_∞-synthesis problem is transformed into a matrix inequality condition using the bounded real lemma [1]. Then the projection lemma from [23] is used to eliminate the unknown controller data A_K, B_K, C_K, D_K from the cast, leaving an LMI problem, which may be solved by semidefinite programming (SDP). In a third step the controller state-space representation (2) is recovered.

This scenario changes dramatically as soon as constraints $K \in \mathcal{K}$ are added. Then the problem may no longer be transformed into an LMI or any other convex program, and alternative algorithmic strategies are required. The aim of this paper is to present and analyze one such alternative.

Example 1. Pure stabilization. Often the first important step in controller synthesis (3) is to find a stabilizing controller K. Already at this stage the H_∞-norm plays a prominent role, because of the well-known fact that under stabilizability and detectability, a linear-time invariant system is Lyapunov stable if and only if its H_∞-norm is finite [17]. More specifically, under stabilizability and detectability assumptions, the static control law $u = Ky$ stabilizes the plant

$$G(s) : \left\{ \begin{array}{rcl} \dot{x} & = & Ax + B_2 u \\ y & = & C_2 x\,, \end{array} \right. \tag{4}$$

if and only if the closed-loop transfer matrix $C_2(sI - (A + B_2KC_2))^{-1}B_2$ has finite H_∞-norm.

In order to construct a static stabilizing controller for an unstable open-loop system (4), the following procedure appears fairly natural. Suppose we are given an initial guess K_0, which leaves the closed-loop system unstable. Then we pick $a_0 > 0$ such that the a_0-shifted H_∞-norm of the closed-loop system is finite:

$$\|C_2(sI - (A + B_2K_0C_2))^{-1})B_2\|_{-a_0,\infty} < +\infty,$$

where the shifted H_∞-norm is given in [10]. The problem of finding a stabilizing controller K may now be addressed by an optimization program

$$\underset{K}{\text{minimize}}\, \|C_2(sI - (A + B_2KC_2 - aI))^{-1}B_2\|_\infty, \tag{5}$$

where the shift a is either fixed at the initial a_0 or is gradually decreased after each minimization step to accelerate the procedure. A stabilizing controller K is obviously obtained when $a \leq 0$ is reached, but usually $a > 0$ is satisfactory and leads to a stabilizing gain K. Numerical tests for this method will be presented in section 7.

While we stop the optimization (5) as soon as a stabilizing K is reached, it may happen that for a fixed shift a, the method converges to a local minimum K of (5), which fails to stabilize the closed-loop system. This is explained by the fact that (5) just like all the other methods in this paper are local optimization methods in the sense that they guarantee convergence to local minima (or critical points). If an unsatisfactory local minimum is reached, the only possibility is to do a restart with a new initial guess K_0, or switch to another method. Such a local convergence certificate may appear weak at first sight, but experience shows that local methods perform much better than global optimization techniques. Those may have stronger certificates, but run into numerical problems even for small problems. And indeed, our present approach is almost always successful even without restart. A similar comment applies to the pure stabilization method in [3]. □

Notice that pure stabilization is just a special case of the more general H_∞ synthesis problem in section 2 when we specialize the standard form to

$$P(s): \qquad \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_2 & B_2 \\ C_2 & 0 & 0 \\ C_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}. \tag{6}$$

Example 2. Spectral abscissa. In [11, 13] Burke *et al.* present an alternative approach to computing static stabilizing controllers K. The authors propose to solve the nonsmooth optimization program

$$\underset{K}{\text{minimize}} \quad \alpha(A + B_2 K C_2), \tag{7}$$

where α is the spectral abscissa of a matrix $M \in \mathbb{M}_{n,n}$ defined as

$$\alpha(M) = \max\{\operatorname{Re}\lambda : \lambda \text{ eigenvalue of } M\}.$$

Unfortunately, this function is not even locally Lipschitz, which renders application of existing nonsmooth algorithms impossible. The authors have therefore developed a probabilistic algorithm which allows to treat problems like (7); see [12, 13]. Since optimality tests and approximate subgradients for α are difficult to compute (see [3]), we prefer the use of (5) over (7). Numerical tests for (5) are presented in section 7 and show that this method is successful as a rule. Our own numerical tests with program (7) are published in [3]. □

The above scenario covers the case of a static stabilizing controller K, but it is clear that stabilization problems including structural constraints $K \in \mathcal{K}$ can be treated in exactly the same way. Several examples of classes $\mathcal{K}$ will be presented in the sequel.

Example 3. Simultaneous stabilization. Another instance of interest is the simultaneous stabilization problem, which can be cast as minimizing a finite family of closed-loop transfer functions. Formally, given the open-loop plants, G_i, $i = 1, \ldots, r$, we consider the problem

$$\underset{K}{\text{minimize}} \ \max_{i=1,\ldots,r} \gamma_i \|C_i(sI - (A_i + B_i K C_i) - a_i I)^{-1} B_i\|_\infty$$

where $\gamma_i > 0$ are appropriate weights, and the a_i are chosen so that the initial guess K renders the ith system stable after shifting by a_i. A lower bound for a_i is therefore the spectral abscissa (7) of the ith system. As before, the shifts are decreased after each H_∞-norm minimization step and a simultaneously stabilizing K is obtained e.g. if $a_i \leq 0$ for all $i = 1, \ldots, r$. But even when some $a_i > 0$, the solution may produce a simultaneously stabilizing K. □

Example 4. System reduction. A technique of considerable importance is system reduction. It is used by practitioners whenever an open-loop system $G(s)$ of large order N is difficult to control. Let $G(s)$ denote such a large size open-loop plant, and suppose a decomposition $G(s) = G_{\text{instab}}(s) + G_{\text{stab}}(s)$ into an unstable and a stable part is available. Then we may consider the problem

$$\underset{\widetilde{G}_{\text{stab}} \in \mathcal{K}}{\text{minimize}} \, \|G_{\text{stab}}(s) - \widetilde{G}_{\text{stab}}(s)\|, \tag{8}$$

where $\tilde{G}_{\text{stab}}(s)$ ranges over a prespecified class $\mathcal{K}$ of stable system of reduced order, $n \ll N$, and where some norm criterion is used to evaluate the mismatch between nominal and reduced systems. If $\|\cdot\|$ represents the Hankel norm, an explicit expression for $\widetilde{G}_{\text{stab}}$ is available [26]. But it may be preferable to use other criteria like the H_∞-norm, a problem which then falls within the class of problems considered in this work. Once a solution $\widetilde{G}_{\text{stab}}(s)$ to (8) is obtained, the new system $\widetilde{G}(s) = G_{\text{instab}}(s) + \widetilde{G}_{\text{stab}}(s)$, while easier to control, may be expected to have characteristics similar to those of the original system. □

3 Subdifferential of the H_∞-norm

In this section, we start by characterizing the convex subdifferential of the H_∞-norm, and derive expressions for the Clarke subdifferential of several nonconvex composite functions $f(x) = \|\mathcal{F}(x)\|_\infty$, where $\mathcal{F}$ is a smooth operator defined on some $\mathbb{R}^n$ with values in the space of stable matrix transfer functions $\mathbf{H}_\infty$.

Consider the H_∞-norm of a nonzero transfer matrix function $G(s)$:

$$\|G\|_\infty = \sup_{\omega \in \mathbb{R}} \overline{\sigma}\left(G(j\omega)\right),$$

where G is stable and $\overline{\sigma}(X)$ is the maximum singular value of X. Suppose $\|G\|_\infty = \overline{\sigma}\left(G(j\omega)\right)$ is attained at some frequency ω, where the case $\omega = \infty$ is allowed. Let $G(j\omega) = U\Sigma V^H$ be a singular value decomposition. Pick u the first column of U, v the first column of V, that is, $u = G(j\omega)v/\|G\|_\infty$. Then the linear functional $\phi = \phi_{u,v,\omega}$ defined as

$$\phi(H) = \text{Re}\left(u^H H(j\omega)v\right) = \|G\|_\infty^{-1}\text{Re}\,\text{Tr}\, vv^H G(j\omega)^H H(j\omega) = \|G\|_\infty^{-1}\text{Re}\,\text{Tr}\, G(j\omega)^H uu^H H(j\omega)$$

is continuous on the space $\mathbf{H}_\infty$ of stable transfer functions and is a subgradient of $\|\cdot\|_\infty$ at G [10]. More generally, assume that the columns of Q_u form an orthonormal basis of the eigenspace of $G(j\omega)G(j\omega)^H$ associated with the largest eigenvalue $\lambda_1\left(G(j\omega)G(j\omega)^H\right) = \overline{\sigma}(G(j\omega))^2$, and that the columns of Q_v form an orthonormal basis of the eigenspace of $G(j\omega)^H G(j\omega)$ associated with the same eigenvalue. Then for all complex Hermitian matrices $Y_v \succeq 0$, $Y_u \succeq 0$ with $\text{Tr}\,(Y_v) = 1$ and $\text{Tr}\,(Y_u) = 1$,

$$\phi(H) = \|G\|_\infty^{-1}\,\text{Re}\ \text{Tr}\, Q_v Y_v Q_v^H G(j\omega)^H H(j\omega) = \|G\|_\infty^{-1}\text{Re}\,\text{Tr}\, G(j\omega)^H Q_u Y_u Q_u^H H(j\omega) \tag{9}$$

is a subgradient of $\|\cdot\|_\infty$ at G. Finally, with $G(s)$ rational and assuming that there exist finitely many frequencies $\omega_1, \ldots, \omega_p$ where the supremum $\|G\|_\infty = \overline{\sigma}(G(j\omega_\nu))$ is attained, all subgradients of $\|\cdot\|_\infty$ at G are precisely of the form

$$\phi(H) = \|G\|_\infty^{-1} \operatorname{Re} \sum_{\nu=1}^{p} \operatorname{Tr} G(j\omega_\nu)^H Q_\nu Y_\nu Q_\nu^H H(j\omega_\nu),$$

where the columns of Q_ν form an orthonormal basis of the eigenspace of $G(j\omega_\nu)G(j\omega_\nu)^H$ associated with the leading eigenvalue $\|G\|_\infty^2$, and where $Y_\nu \succeq 0$, $\sum_{\nu=1}^{p} \operatorname{Tr}(Y_\nu) = 1$. See [15, Prop. 2.3.12 and Thm. 2.8.2] and [3] for this.

Suppose now we have a smooth operator $\mathcal{G}$, mapping $\mathbb{R}^n$ onto the space $\mathbf{H}_\infty$ of stable transfer functions G. Then the composite function $f(x) = \|\mathcal{G}(x)\|_\infty$ is Clarke subdifferentiable at x with

$$\partial f(x) \;=\; \mathcal{G}'(x)^\star[\partial\|\cdot\|_\infty(\mathcal{G}(x))], \tag{10}$$

where $\partial\|\cdot\|_\infty$ is the subdifferential of the H_∞-norm obtained above, and where $\mathcal{G}'(x)^\star$ is the adjoint of $\mathcal{G}'(x)$, mapping the dual of $\mathbf{H}_\infty$ into $\mathbb{R}^n$. In the sequel, we will compute this adjoint $\mathcal{G}'(x)^\star$ for special classes of closed-loop transfer functions. Suitable chain rules covering this case are for instance given in [15, section 2.3].

4 Clarke subdifferentials in closed-loop

Given a stabilizing controller $K(s)$ and a plant with the usual partition

$$P(s) := \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix},$$

the closed-loop transfer function is obtained as

$$T_{w\to z}(K) := P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}\,,$$

where the state-space data of P_{11}, P_{12}, P_{21} and P_{22} are given in (1) and the dependence on s is omitted for brevity. Our aim is to compute the subdifferential $\partial f(K)$ of $f := \|\cdot\|_\infty \circ T_{w\to z}$ at K. We first notice that the derivative $T'_{w\to z}(K)$ of $T_{w\to z}$ at K is

$$T'_{w\to z}(K)\delta K := P_{12}(I - KP_{22})^{-1}\delta K(I - P_{22}K)^{-1}P_{21},$$

where δK is an element of the same matrix space as K.

Now let $\phi = \phi_Y$ be a subgradient of $\|\cdot\|_\infty$ at $T_{w\to z}(K)$ of the form (9), specified by $Y \succeq 0$, $\operatorname{Tr}(Y) = 1$ and with $\|T_{w\to z}(K)\|_\infty$ attained at frequency ω. According to the chain rule, the subgradients Φ_Y of f at K are of the form $\Phi_Y := T'_{w\to z}(K)^\star\phi_Y \in \mathbb{M}_{m_2,p_2}$, where the adjoint $T'_{w\to z}(K)^\star$ acts on ϕ_Y through

$$\begin{aligned}
\langle T'_{w\to z}(K)^\star\phi_Y, \delta K\rangle &= \langle T'_{w\to z}(K)\delta K, \phi_Y\rangle \\
&= \|T_{w\to z}(K)\|_\infty^{-1}\operatorname{Re}\operatorname{Tr}\left(T_{w\to z}(K,j\omega)^H QYQ^H T'_{w\to z}(K)\delta K\,(j\omega)\right) \\
&= \|T_{w\to z}(K)\|_\infty^{-1}\operatorname{Re}\operatorname{Tr}\left(T_{w\to z}(K,j\omega)^H QYQ^H P_{12}(j\omega)(I - K(j\omega)P_{22}(j\omega))^{-1}\right. \\
&\qquad \left.\delta K(j\omega)(I - P_{22}(j\omega)K(j\omega))^{-1}P_{21}(j\omega)\,\right) \\
&= \|T_{w\to z}(K)\|_\infty^{-1}\operatorname{Re}\operatorname{Tr}\left((I - P_{22}(j\omega)K(j\omega))^{-1}P_{21}(j\omega)T_{w\to z}(K,j\omega)^H\right. \\
&\qquad \left. QYQ^H P_{12}(j\omega)(I - K(j\omega)P_{22}(j\omega))^{-1}\delta K(j\omega)\,\right)\,.
\end{aligned} \tag{11}$$

In consequence, for a static K, the Clarke subdifferential of $f(K) := \|T_{w\to z}(K)\|_\infty$ at K consists of all subgradients Φ_Y of the form

$$\begin{aligned}\Phi_Y &= \|T_{w\to z}(K)\|_\infty^{-1}\mathrm{Re}\,\big((I-P_{22}(j\omega)K)^{-1}P_{21}(j\omega)T_{w\to z}(K,j\omega)^H \\ &\quad QYQ^H\,P_{12}(j\omega)(I-KP_{22}(j\omega))^{-1}\,\big)^T\,,\end{aligned} \tag{12}$$

where $Y \succeq 0$ and $\mathrm{Tr}\,(Y) = 1$. Recall that Φ_Y is now an element of the same matrix space as K and acts on test vectors δK through $\langle \Phi_Y, \delta K\rangle = \mathrm{Tr}(\Phi_Y^T \delta K)$.

This formula is easily adapted if the H_∞-norm is attained at a finite number of frequencies $\omega_1, \ldots, \omega_q$. In this more general situation, subgradients of f at K are of the form

$$\begin{aligned}\Phi_Y &= \|T_{w\to z}(K)\|_\infty^{-1}\textstyle\sum_{\nu=1}^q \mathrm{Re}\,\big((I-P_{22}(j\omega_\nu)K)^{-1}P_{21}(j\omega_\nu)T_{w\to z}(K,j\omega_\nu)^H \\ &\quad QY_\nu Q^H\,P_{12}(j\omega_\nu)(I-KP_{22}(j\omega_\nu))^{-1}\,\big)^T\,,\end{aligned} \tag{13}$$

where $Y = (Y_1, \ldots, Y_q)$, $Y_\nu \succeq 0$ and $\sum_{\nu=1}^q \mathrm{Tr}(Y_\nu) = 1$.

At this stage, it is important to stress that expressions (11), (12) and (13) are general and can accommodate any problem discussed in previous sections. Below we resume and expand this list by considering more examples of practical interest.

Example 5. Dynamic controllers. Assume now that the controller is dynamic as in (2). The subgradient set is again obtained via formula (13) by performing the substitutions:

$$\begin{aligned}&K \to \begin{bmatrix} A_K & B_K \\ C_K & D_K\end{bmatrix}, \quad A \to \begin{bmatrix} A & 0 \\ 0 & 0_k\end{bmatrix} \quad B_1 \to \begin{bmatrix} B_1 \\ 0\end{bmatrix}, \quad C_1 \to [\,C_1 \quad 0\,] \\ &B_2 \to \begin{bmatrix} 0 & B_2 \\ I_k & 0\end{bmatrix}, \quad C_2 \to \begin{bmatrix} 0 & I_k \\ C_2 & 0\end{bmatrix}, \quad D_{12} \to [\,0 \quad D_{12}\,], \quad D_{21} \to \begin{bmatrix} 0 \\ D_{21}\end{bmatrix}.\end{aligned} \tag{14}$$

This yields the following Clarke partial subgradients with respect to the controller variable A_K, B_K, C_K and D_K:

$$\begin{aligned}\Phi_{Y,A_K} &= \|T_{w\to z}(K)\|_\infty^{-1}\textstyle\sum_{\nu=1}^q \mathrm{Re}\,\big((j\omega_\nu - A_K)^{-1}B_K G_{21}(j\omega_\nu)T_{w\to z}(K,j\omega_\nu)^H QY_\nu Q^H \\ &\quad G_{12}(j\omega_\nu)C_K(j\omega_\nu - A_K)^{-1}\,\big)^T\,, \\ \Phi_{Y,B_K} &= \|T_{w\to z}(K)\|_\infty^{-1}\textstyle\sum_{\nu=1}^q \mathrm{Re}\,\big(G_{21}(j\omega_\nu)T_{w\to z}(K,j\omega_\nu)^H QY_\nu Q^H G_{12}(j\omega_\nu)C_K(j\omega_\nu - A_K)^{-1}\big)^T\,, \\ \Phi_{Y,C_K} &= \|T_{w\to z}(K)\|_\infty^{-1}\textstyle\sum_{\nu=1}^q \mathrm{Re}\,\big((j\omega_\nu - A_K)^{-1}B_K G_{21}(j\omega_\nu)T_{w\to z}(K,j\omega_\nu)^H QY_\nu Q^H G_{12}(j\omega_\nu)\big)^T\,, \\ \Phi_{Y,D_K} &= \|T_{w\to z}(K)\|_\infty^{-1}\textstyle\sum_{\nu=1}^q \mathrm{Re}\,\big(G_{21}(j\omega_\nu)T_{w\to z}(K,j\omega_\nu)^H QY_\nu Q^H G_{12}(j\omega_\nu)\big)^T\,,\end{aligned}$$

with the notations

$$G_{21}(s) := (I - P_{22}(s)K(s))^{-1}P_{21}(s), \quad G_{12}(s) := P_{12}(s)(I - K(s)P_{22}(s))^{-1}.$$

The entire Clarke subdifferential is then described by the set of subgradients in $\mathbb{M}_{k+m_2,k+p_2}$

$$\Phi_Y := \begin{bmatrix} \Phi_{Y,A_K} & \Phi_{Y,B_K} \\ \Phi_{Y,C_K} & \Phi_{Y,D_K}\end{bmatrix},$$

where as before $Y = (Y_1, \ldots, Y_q)$, $Y_\nu \succeq 0$ and $\sum_{\nu=1}^q \mathrm{Tr}(Y_\nu) = 1$. □

Example 6. Structured controllers. In practice, it is sometimes required that some entries in the controller gain be put to zero, while the others may be freely assigned. This is the case in decentralized control, where the controller must enjoy a block-diagonal structure. Consider a

pattern matrix W with entries $W_{ij} \in \{0,1\}$, where $W_{ij} = 0$ means that the controller gain $K_{ij} = 0$ must be zero, whereas $W_{\ell k} = 1$ means that $K_{\ell k}$ can be freely assigned. The Clarke subdifferential of $f = \|\cdot\|_\infty \circ \mathcal{F}$ at K is then of the form

$$W \odot \Phi_Y \,,$$

where $\Phi_Y \in \partial\|\cdot\|_\infty(\mathcal{F}(K))$ is as in (13) and where $\odot$ denotes the entry-wise Hadamard or Schur product [29]. □

Example 7. PID controllers. PID control is one of the most classical approaches in control system design. The controller is generally written as

$$K(s) = K_P + \frac{1}{s}K_I + \frac{s}{1+\tau s}K_D \,,$$

where K_P, K_I and K_D are matrix static gains to be computed, and τ is a small positive scalar. Using the general formula (11), the subgradients with respect to $[\,K_P; K_I; K_D\,]$ at $K(s)$ are obtained through

$$\begin{aligned}\Phi_{Y,[K_P;K_I;K_D]} \;=\; & \|T_{w\to z}(K)\|_\infty^{-1} \textstyle\sum_{\nu=1}^q \mathrm{Re}\,\big((I - P_{22}(j\omega_\nu)K(j\omega_\nu))^{-1}P_{21}(j\omega_\nu)T_{w\to z}(K,j\omega_\nu)^H \\ & QY_\nu Q^H\, P_{12}(j\omega)(I - K(j\omega_\nu)P_{22}(j\omega_\nu))^{-1}\,[\,I \quad \tfrac{1}{j\omega_\nu}I \quad \tfrac{j\omega_\nu}{1+\tau j\omega_\nu}I\,]\,\big)^T \,.\end{aligned}$$

where as before $Y = (Y_1, \ldots, Y_q)$, $Y_\nu \succeq 0$ and $\sum_{\nu=1}^q \mathrm{Tr}(Y_\nu) = 1$.

Note that this formula readily extends to arbitrary basis functions $\{Q_j(s)\}_{j=1,\ldots,r}$

$$K(s) := \sum_{j=1}^r K_j Q_j(s) \,,$$

where the K_j's are the design variables.

Example 8. Matrix fraction representations. An alternative representation of controllers is via matrix fraction descriptions. For instance, the left matrix fraction representation is given as

$$K(s) = N(s)D(s)^{-1} = (\mathcal{N}J_N(s))(\mathcal{D}J_D(s))^{-1} \,,$$

with

$$J_N(s) := [\,I \quad sI \quad \ldots \quad s^n I\,]^T \,, J_D(s) := [\,I \quad sI \quad \ldots \quad s^d I\,]^T$$

and

$$\mathcal{N} := [\,N_0 \quad \ldots \quad N_n\,] \,, \mathcal{D} := [\,D_0 \quad \ldots \quad D_n\,]$$

Now, $\mathcal{N}$ and $\mathcal{D}$ are the design variables. As before, it is immediate to show that partial subgradients with respect to $\mathcal{N}$ and $\mathcal{D}$ are given as

$$\begin{aligned}\Phi_{Y,\mathcal{N}} \;:=\; & \|T_{w\to z}(K)\|_\infty^{-1} \textstyle\sum_{\nu=1}^q \mathrm{Re}\,(J_N(j\omega_\nu)(D(j\omega_\nu)J_D(j\omega_\nu))^{-1}(I - P_{22}(j\omega_\nu)K(j\omega_\nu))^{-1} \\ & P_{21}(j\omega_\nu)T_{w\to z}(K,j\omega_\nu)^H\, QY_\nu Q^H\, P_{12}(j\omega_\nu)(I - K(j\omega_\nu)P_{22}(j\omega_\nu))^{-1}\,)^T \\ \Phi_{Y,\mathcal{D}} \;:=\; & -\|T_{w\to z}(K)\|_\infty^{-1} \textstyle\sum_{\nu=1}^q \mathrm{Re}\,(J_D(j\omega_\nu)(D(j\omega_\nu)J_D(j\omega_\nu))^{-1}(I - P_{22}(j\omega_\nu)K(j\omega_\nu))^{-1} \\ & P_{21}(j\omega_\nu)T_{w\to z}(K,j\omega_\nu)^H\, QY_\nu Q^H\, P_{12}(j\omega_\nu)(I - K(j\omega_\nu)P_{22}(j\omega_\nu))^{-1}K(j\omega_\nu)\,)^T \,,\end{aligned}$$

respectively. □

Example 9. Multiple performance channels. Practical specifications often impose that several closed-loop channels $i = 1, \ldots, r$ be minimized simultaneously. One way to address multi-objective optimization of this type is to solve a program of the form

$$\underset{K}{\text{minimize}} \max \left\{ \gamma_i \|T^i_{w\to z}(K)\|_\infty : \ i = 1, \ldots, r \right\} ,$$

where $T^i_{w\to z}$ is the ith performance specification to be optimized. Since the maximum of a finite number of maximum eigenvalue functions is itself a maximum eigenvalue function of a block diagonal operator $T = \text{diag}(T^1_{w\to z}, \ldots, T^r_{w\to z})$, the Clarke subgradients could be obtained directly from (13). When the usual max formula is used, the result is the same, i.e., subgradients are of the form

$$\phi_{(Y,\tau)} = \sum_{i\in I(K)} \tau_i \gamma_i \phi_{Y_i},$$

where $I(K)$ are the $i = 1, \ldots, r$ which are active at K, $\tau_i \geq 0$, $\sum_{i\in I(K)} \tau_i = 1$ and $\phi_{Y_i} \in (T^i_{w\to z})^\star \partial \|\cdot\|_\infty(K)$ as specified in (13). □

Before going further, it is worth mentioning that our methodology carries over to a wide range of controller structures of practical interest. This is in particular the case when the structural constraint is of the form $\mathcal{K} = \{K : K = \mathcal{S}(\ell), \ell \in \mathcal{L}\}$, where $(\mathcal{S}, \mathcal{L})$ is a suitable differentiable parametrization of the class $\mathcal{K}$. This includes for instance observed-based controllers, feed-forward compensators, controllers defined through Youla parameterizations and much else.

5 Steepest descent method

Nonsmooth techniques have been used before in algorithms for controller synthesis. For instance, E. Polak and co-workers have proposed a variety of techniques suited for eigenvalue or singular-value optimization and for extensions to the semi-infinite case, covering in particular the H_∞-norm (see [39], [40] and the citations given there). Another reference is [10], where the authors exploit the Youla parameterization via convex nondifferentiable analysis to derive the cutting plane and ellipsoid algorithms.

Let us consider the problem of minimizing $f(x) = \|\mathcal{G}(x)\|_\infty$, where x regroups the controller data, referred to as K in the previous section, and where $\mathcal{G}$ maps $\mathbb{R}^n$ smoothly into a space $\mathbf{H}_\infty$ of stable transfer functions. We write $\mathcal{G}(x, s)$ or $\mathcal{G}(x, j\omega)$ when the complex argument of $\mathcal{G}(x) \in \mathbf{H}_\infty$ needs to be specified.

A necessary condition for optimality is $0 \in \partial f(x) = \mathcal{G}'(x)^\star \partial \|\cdot\|_\infty (\mathcal{G}(x))$. It is therefore reasonable to consider the program

$$d = -\frac{g}{\|g\|}, \qquad g = \text{argmin}\{\|\phi_Y\| : Y = (Y_1, \ldots, Y_q), Y_\nu \succeq 0, \sum_{\nu=1}^{q} \text{Tr}(Y_\nu) = 1\} \tag{15}$$

which either shows $0 \in \partial f(x)$, or produces the direction d of steepest descent at x if $0 \notin \partial f(x)$, and where the ϕ_Y are as in (13). If we vectorize $y = \text{vec}(Y)$, $Y = (Y_1, \ldots, Y_q)$, then we may represent ϕ_Y by a matrix vector product, $\phi_Y = \Phi y$, with a suitable matrix Φ. Program (15) is

then equivalent to the following SDP:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} t & y^T\Phi^T \\ \Phi y & tI \end{bmatrix} \succeq 0 \\ & Y_i \succeq 0, i = 1, \ldots, q \\ & e^T y = 1 \end{array} \tag{16}$$

where $e^T y = 1$ encodes the constraint $\sum_i \mathrm{Tr}(Y_i) = 1$. The direction d of steepest descent at x is then obtained as $d = -\Phi\, y/\|\Phi\, y\|$, where (t, y) is solution of (16) with $y \neq 0$. This suggests the following:

<u>Steepest descent method for the H_∞-norm</u>

1. If $0 \in \partial f(x)$ stop. Otherwise:
2. Solve (16) and compute the direction d of steepest descent at x.
3. Perform a line search and find a descent step $x^+ = x + t\, d$.
4. Replace x by x^+ and go back to step 1.

The drawback of this approach is that it may fail to converge due to the nonsmoothness of f. We believe that a descent method should at least give the weak convergence certificate that accumulation points of the sequence of iterates are critical. This is not guaranteed by the above scheme. The reason is that the steepest descent direction at x does not depend continuously on x. This is why modifications of the steepest descent scheme are discussed in the next section.

Remark. Spectral abscissa versus H_∞-norm. Notice that the stopping test (16) is convenient since it leads to relatively small size SDPs. Indeed, the matrices Y_ν, $\nu = 1, \ldots, q$, are of the size of the multiplicity of $\lambda_1(\mathcal{F}(x, j\omega_\nu))$, and our experiments indicate that $\dim(y)$ in (16) rarely exceeds 30. The situation is very different for the spectral abscissa (7). In [3] we have derived a stopping test for program (7). The difficulty is that Lyapunov variables re-enter the scene. Indeed, x^* is a local minimum of the composite function $\alpha \circ \mathcal{F}$ like in (7) with value $\alpha(\mathcal{F}(x^*)) = t^*$ if and only if (x^*, t^*, X^*) is a local minimum of the optimization program:

$$(P) \quad \begin{array}{ll} \text{minimize} & t \\ \text{subject to} & X \succeq \sigma I,\ X \preceq I \\ & \mathcal{F}(x)^T X + X\mathcal{F}(x) - 2tX \preceq 0 \end{array}$$

($0 < \sigma < 1$ fixed). An optimality test for $\alpha \circ \mathcal{F}$ is therefore derived from an optimality test for program (P), which follows standard lines (see [3]). This leads to an SDP with unknown variable of $\dim(x, X)$, which may be prohibitively large. This is one of the reasons why our present approach privileges the use of the H_∞-norm (5) over (7). □

6 First-order descent method

In this section we devise a first-order algorithm for composite functions of the H_∞ norm. Along with $\mathcal{G} : \mathbb{R}^n \to \mathbf{H}_\infty$ we consider the symmetrized operators $\mathcal{F}(x, s) = \mathcal{G}(x, s)\mathcal{G}(x, s)^H$ respectively

$\mathcal{F}(x,s) = \mathcal{G}(x,s)^H \mathcal{G}(x,s)$. We represent $f(x)$ as

$$f(x) = \sup_{\omega \in \mathbb{R}} f(x,\omega), \qquad f(x,\omega) = \lambda_1\left(\mathcal{F}(x,j\omega)\right), \tag{17}$$

and solve the optimization program

$$\min_{x \in \mathbb{R}^n} f(x) = \|\mathcal{G}(x)\|_\infty^2 = \sup_{\omega \in \mathbb{R}} \lambda_1\left(\mathcal{F}(x,j\omega)\right)$$

Notice that for fixed $\omega \in \mathbb{R}$, $x \mapsto \mathcal{F}(x, j\omega)$ is a smooth operator into the space of symmetric $m \times m$ matrices, $\mathbb{S}^m$, while $\lambda_1 : \mathbb{S}^m \to \mathbb{R}$ is the maximum eigenvalue function. Similar techniques could be applied to broader classes with a structure like (17). Deriving the method will require three steps, which we regroup into subsections. We start with the important special case $f = \lambda_1 \circ \mathcal{F}$, where $\mathcal{F}$ maps $\mathbb{R}^n$ smoothly into $\mathbb{S}^m$.

6.1 Preparation

The function (17) is subject to two sources of nonsmoothness. The nonsmooth character of the maximum eigenvalue function, and the nonsmoothness introduced by the operator sup, which in the case of $\|\cdot\|_\infty$ is even infinite. Each individual function $f(\cdot,\omega)$ will be analytic at x if the multiplicity of $\lambda_1\left(\mathcal{F}(x,j\omega)\right)$ is one, but nonsmoothness needs to be taken into account as soon as eigenvalues coalesce.

From a practical point of view it is reasonable to make the following additional assumptions.

(H1) The functions $\omega \mapsto f(x,\omega)$ decay rapidly at infinity. Therefore $f(x) = \max_{|\omega| \leq \overline{\omega}} f(x,\omega)$ for a cutoff frequency $\overline{\omega} > 0$. This cutoff frequency may be chosen uniformly for all x in a bounded set.

(H2) The maximum $f(x)$ is always attained on a finite set of frequencies. The set of these frequencies is denoted by $\Omega(x)$.

Assumption (H2) is for instance satisfied when the multiplicity of λ_1 is 1, as $\omega \mapsto f(x,\omega)$ is then analytic in typical control applications.

Let us introduce some more notation. For $\Omega \subset \mathbb{R}$ we define $f_\Omega \leq f$ as

$$f_\Omega(x) = \max_{\omega \in \Omega} f(x,\omega).$$

Notice that $f_\Omega(x) = f(x)$ as soon as $\Omega(x) \subset \Omega$. Next recall the definition of the ε-subdifferential of the maximum eigenvalue function

$$\partial_\varepsilon \lambda_1(X) = \{Z \in \mathbb{S}^m : Z \bullet X \geq \lambda_1(X) - \varepsilon\}$$

which is an important analytical tool in nonsmooth analysis. Since $\partial_\varepsilon \lambda_1(X)$ is difficult to compute, we follow Cullum et al. [16] and Oustry [37] and introduce a modification $\delta_\varepsilon \lambda_1(X)$ of $\partial_\varepsilon \lambda_1(X)$, called the *$\varepsilon$-enlarged subdifferential* for the maximum eigenvalue function. For $\varepsilon > 0$ and $X \in \mathbb{S}^m$ let $r(\varepsilon, X)$ the index such that

$$\lambda_1(X) \geq \lambda_2(X) \geq \ldots \geq \lambda_{r(\varepsilon,X)}(X) \geq \lambda_1(X) - \varepsilon > \lambda_{r(\varepsilon,X)+1}(X) \geq \ldots \geq \lambda_m(X)$$

$r(\varepsilon, X)$ is also called the ε-multiplicity of $\lambda_1(X)$. Let Q_ε be a $r(\varepsilon, X) \times m$-matrix whose columns form an orthonormal basis of the invariant subspace of X associated with the first $r(\varepsilon, X)$ eigenvalues. Then we define

$$\delta_\varepsilon \lambda_1(X) = \{Q_\varepsilon Y Q_\varepsilon^H : Y \succeq 0, \mathrm{Tr}(Y) = 1, Y \in \mathbb{S}^{r(\varepsilon,X)}\}.$$

By construction $\partial\lambda_1(X) \subset \delta_\varepsilon\lambda_1(X) \subset \partial_\varepsilon\lambda_1(X)$, so $\delta_\varepsilon\lambda_1(X)$ is an enlargement of $\partial\lambda_1(X)$ and an inner approximation of the ε-subdifferential. The gap associated with the choice ε is $\Delta(\varepsilon, X) = \lambda_{r(\varepsilon,X)}(X) - \lambda_{r(\varepsilon,X)+1}(X) > 0$. If r is the multiplicity of $\lambda_1(X)$, then choosing ε small enough gives $r(\varepsilon, X) = r$. In this case we have $\delta_\varepsilon\lambda_1(X) = \partial\lambda_1(X)$.

The following is an important step towards the analysis of (17). Consider a differentiable mapping $\mathcal{F} : \mathbb{R}^n \to \mathbb{S}^m$. We extend $\partial_\varepsilon\lambda_1(X)$ and the enlarged subdifferential δ_ε to the composite function $f = \lambda_1 \circ \mathcal{F}$ by setting

$$\partial_\varepsilon f(x) = \mathcal{F}'(x)^\star \left[\partial_\varepsilon \lambda_1(X)\right], \qquad X = \mathcal{F}(x),$$

and similarly

$$\delta_\varepsilon f(x) = \mathcal{F}'(x)^\star \left[\delta_\varepsilon \lambda_1(X)\right], \qquad X = \mathcal{F}(x).$$

This allows us finally to consider $\partial_\varepsilon f(x, \omega)$ and $\delta_\varepsilon f(x, \omega)$, which are applied to the variable x.

6.2 Descent Step Generator

In this section we discuss a very simple mechanism which generates descent steps in such a way that the following weak form of convergence can be guaranteed: every accumulation point of the sequence of iterates is a stationary point.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function, and let $\partial f(x)$ denote its Clarke subdifferential [15]. Suppose we can exhibit a mechanism $s : \mathbb{R}^n \to \mathbb{R}^n$, the *descent step generator*, such that the following rules are satisfied:

(i) Whenever $0 \notin \partial f(x)$, then $f(s(x)) < f(x)$.

(ii) When $0 \notin \partial f(x)$, then there exists a neighbourhood $B(x, \varepsilon)$ of x and some $\delta > 0$ such that $f(s(x')) \leq f(x') - \delta$ for every $x' \in B(x, \varepsilon)$.

While (i) simply means that $s(x)$ is a descent step away from x, we can interpret (ii) as some sort of continuity of (i), because when the mapping $s(\cdot)$ describing descent is continuous, then (i) implies (ii) without further work. Clearly, (ii) implies (i).

Example. In order to understand the idea behind axiom (ii), consider a C^1-function f and let $s(x)$ denote the steepest descent step at x, obtained by an Armijo line search. If we define formally $s(x) = x - t(x)f'(x)$, where $t(x)$ is the smallest step satisfying an Armijo rule, then $s(x)$ turns out continuous, but it is clear that in practice we would accept *any* step $t(x)$ satisfying the Armijo condition, without insisting on a continuous dependence of $t(x)$ on x. In that case $s(x)$ will not be continuous, but property (ii) will still hold true. □

Clearly the situation we have in mind is when f is nonsmooth, so that a steepest descent step in tandem with Armijo search would typically fail even when $t(x)$ was continuous. This is explained by the fact that under nonsmoothness, defining $s(\cdot)$ along the lines above would miss axiom (ii), because the steepest descent direction behaves very discontinuously. Indeed, examples where this happens are easily produced.

Proposition 1. *Suppose the descent step generator $s(\cdot)$ for f satisfies axioms (i) and (ii). Let x_k be a sequence of iterates defined as $x_{k+1} = s(x_k)$. Then every accumulation point $\bar{x}$ of x_k is a critical point, that is, satisfies $0 \in \partial f(\bar{x})$.*

Proof. Let $\mathcal{N} \subset \mathbb{N}$ be an infinite sequence such that $x_k \to \bar{x}$, $k \in \mathcal{N}$. Then by monotonicity, $f(x_{k+1}) \to f(\bar{x})$. That means

$$f(x_{k+1}) - f(x_k) \to 0 \qquad (k \in \mathcal{N}).$$

Now use axiom (ii) at the limit point $\bar{x}$. There exist $\varepsilon, \delta > 0$ such that $f(s(x)) - f(x) \leq -\delta < 0$ for all $x \in B(\bar{x}, \varepsilon)$. Since $x_k \in B(\bar{x}, \varepsilon)$ for $k \in \mathcal{N}$ large enough, and since $x_{k+1} = s(x_k)$, we should have $f(x_{k+1}) - f(x_k) \leq -\delta$ for $k \in \mathcal{N}$ large enough, a contradiction. $\square$

6.3 Eigenvalue Optimization

How can we define a descent step generator $s(\cdot)$ with properties (i) and (ii) for a maximum eigenvalue function $f(x) = \lambda_1(\mathcal{F}(x))$? Suppose we are at a point x where $0 \notin \partial f(x)$. Let the eigenvalues of $X = \mathcal{F}(x) \in \mathbb{S}^m$ be arranged into groups:

$$\lambda_1(X) = \ldots = \lambda_{k_2-1}(X) > \lambda_{k_2}(X) = \ldots = \lambda_{k_3-1}(X) > \lambda_{k_3}(X) = \ldots$$

where $k_1 = 1$ and where k_i are the group leaders. Consequently, eigenvalue gaps occur between $k_i - 1$ and k_i. Let Q_1 be an orthonormal bases of the eigenspace belonging to the first block $\lambda_1(X), \ldots, \lambda_{k_2-1}(X)$, Q_2 an orthonormal basis containing Q_1 associated with the first two blocks $\lambda_1(X), \ldots \lambda_{k_3-1}(X)$, and so on. At $X = \mathcal{F}(x) \in \mathbb{S}^m$ we compute the quantities

$$\Delta_i(X) = \lambda_{k_{i+1}-1}(X) - \lambda_{k_{i+1}}(X) > 0,$$

$$d_i(x) = \min\{\|\mathcal{F}'(x)^\star Q_i Y_i Q_i^H\| : Y_i \in \mathbb{S}^{k_{i+1}-1}, Y_i \succeq 0, \mathrm{Tr}(Y_i) = 1\}$$

and keep those $i = 1, \ldots, r$ where $d_i(x) > 0$. Notice that $d_i(x) = \mathrm{dist}(0, \delta_{\varepsilon_i(x)} f(x))$, where $\varepsilon_i(x) > 0$ cuts into the ith gap, that is $\lambda_1(X) - \varepsilon_i(x) \in [\lambda_{k_{i+1}-1}(X), \lambda_{k_{i+1}}(X))$. Put differently, the $\varepsilon_i(x)$-multiplicity of $\lambda_1(X)$ is $k_{i+1} - 1$. Moreover, $d_1(x) \geq d_2(x) \geq \ldots \geq d_r(x)$ and $d_{r+1}(x) = d_{r+2}(x) = \ldots = 0$ eventually. We compute the quantity

$$M(x) = \max_{i=1\ldots r} \Delta_i(X) d_i(x)^2 \tag{18}$$

Now we use the following

Lemma 1. *Let $0 \notin \partial f(x)$ and $R > 0$. Let k_i be the leader of a group of eigenvalues of $X = \mathcal{F}(x) \in \mathbb{S}^m$ such that $d_i(x) = \mathrm{dist}(0, \delta_{\varepsilon_i(x)} f(x)) > 0$. Let $h_i(x)$ be the direction of steepest $\varepsilon_i(x)$ enlarged descent, that is*

$$h_i(x) = -\frac{g_i}{\|g_i\|}, \quad g_i = \mathrm{argmin}\{\|g\| : g = \mathcal{F}'(x)^\star Q_i Y Q_i^H, \mathrm{Tr}(Y) = 1, Y \succeq 0, Y \in \mathbb{S}^{k_{i+1}-1}\},$$

where the columns of Q_i are an orthonormal basis of the invariant subspace of X belonging to the eigenvalues up to $k_{i+1} - 1$. Then there exists a descent step $s_i(x)$ away from x in direction $h_i(x)$, which decreases the value of f by at least

$$f(s_i(x)) - f(x) \leq -\kappa(x) \Delta_i(X) d_i(x)^2 < 0.$$

Here $\kappa(x) > 0$ depends on $\sup\{\|\mathcal{F}'(x')\| : \|x - x'\| \leq R\}$ and in particular continuously on x. The line search required to compute this step is finite.

Proof. In the case of an affine operator $\mathcal{A} : \mathbb{R}^n \to \mathbb{S}^m$, Oustry [37, Theorem 5] shows that $f = \lambda_1 \circ \mathcal{A}$ may be decreased by the quantity

$$f(x + t(x)h_i(x)) - f(x) \leq -\frac{1}{4\|\mathcal{A}_*\|}\Delta_{\varepsilon_i(x)}(X)d_i(x)^2$$

where $\mathcal{A}_*$ is the linear part of $\mathcal{A}$. Here $\kappa = 1/4\|\mathcal{A}_*\|$ is even independent of x. Moreover, $t(x) > 0$ is computed by a line search which terminates after a finite number of steps, see [37, sect. 3.2].

In [35], this result is generalized to nonconvex maximum eigenvalue functions $f = \lambda_1 \circ \mathcal{F}$, with a constant $\kappa(x)$ now depending on the Lipschitz constant of $\mathcal{F}$ on a bounded region around x, like for instance $B(x, R)$ with some fixed $R > 0$. In the nonconvex case, the line search procedure locating $t(x)$ is more complicated than for $f = \lambda_1 \circ \mathcal{A}$, but finite termination is still guaranteed (cf. [35, sect. 3.7]). The constant $\kappa(x)$ may be computed via the formulae (15), (19) and (20) of that reference. □

Following the lead of [35], it is now clear how to obtain our step $s(x)$. Choose i so that the maximum (18) is attained and take $s(x) = s_i(x)$. Then by Lemma 1, $s(x)$ gives a guaranteed decrease of

$$f(s(x)) - f(x) \leq -\kappa(x)\Delta_{\varepsilon_i(x)}(X)d_i(x)^2 = -\kappa(x)M(x),$$

where the constant $\kappa(x) > 0$ depends continuously on x as argued above. What remains to be checked is

Lemma 2. *This choice of $s(x)$ and $M(x)$ guarantees property (ii).*

Proof. Consider a sequence $x_k \to x$. We need to compare $M(x_k)$ to $M(x)$. Now observe that due to the continuity of the eigenvalue functions $\lambda_j(X)$, every eigenvalue gap at X is also an eigenvalue gap at X_k as soon as X_k is sufficiently close to X. On the other hand, X_k may (and will) have many more gaps than X. But notice that the maximum (18) is over all eigenvalue gaps. So each gap in $M(x)$ will occur in the computation of $M(x_k)$. More precisely, it will be approximated by some of the gaps considered in $M(x_k)$. Put differently, for the ith eigenvalue gap of X we have $\Delta_{i_k}(X_k)d_{i_k}(x_k)^2 \to \Delta_i(X)d_i(x)^2$, where i_k is the index of the eigenvalue gap of X_k corresponding to the ith gap of X. Here $\Delta_{i_k}(X_k) \to \Delta_i(X)$ and $d_{i_k}(x_k) \to d_i(x)$ rely on the fact that gaps at X_k remain gaps at X. Since $d_{i_k}(x_k)$ calls for all the $Q_1, Q_2, \ldots$ up to the i_kth gap at X_k, and since these converge to the corresponding basis Q regrouping the gap i, the result follows.

That means $\limsup M(x_k) \geq M(x)$, so we can assume $M(x_k) \geq \frac{1}{2}M(x)$ from some index k on. This in turn implies

$$f(s(x_k)) - f(x_k) \leq -\kappa(x_k)M(x_k) \leq -\frac{1}{4}\kappa(x)M(x)$$

as soon as $\kappa(x_k)$ is close enough to $\kappa(x)$, proving property (*ii*). □

The method outlined in this section gives a convergence certificate because *all* eigenvalue gaps are included in the computation of (18). This may seem inconvenient for very large size matrices $\mathcal{F}(x)$. If we decide to truncate and consider only some of the eigenvalue gaps among the largest eigenvalues, the theoretical convergence properties of the method are weaker, even though convergence may still be guaranteed e.g. when f is convex (see [35]). Notice however that even for large m the quantity $M(x)$ in (18) may be computed fairly reliably by considering

$\Delta_{\varepsilon_i(x)}(X)d_i(x)^2$ for some of the first i only. As $d_i(x) \to 0$ rather quickly, the higher $\Delta_{\varepsilon_i(x)}(X)d_i(x)^2$ as a rule do not contribute to the computation of $M(x)$. Also notice that since the sequence $d_i(x)$ is monotone, computing $d_i(x)$ may often be avoided, for instance when $\Delta_{\varepsilon_{i-1}(x)}(X) \geq \Delta_{\varepsilon_i(x)}(X)$ or $\Delta_{\varepsilon_i(x)}(X) \leq \Delta_{\varepsilon_{i-\nu}(x)}(X)$ when the current best value is located at index $i - \nu$.

6.4 Descent by a local model

In this section we present an alternative way to obtain a descent step generator $s(\cdot)$ for the maximum eigenvalue function $f(x) = \lambda_1(\mathcal{F}(x))$. We start by constructing an intermediate function $\theta(x)$, which serves as an optimality test, and secondly, will allow us to quantify decrease. In this section our method follows the line of [40, Thm. 2.1.6].

Let $X = \mathcal{F}(x) \in \mathbb{S}^m$. Let $\mu_1(X) > \mu_2(X) > \ldots > \mu_r(X)$ be the eigenvalues of X without repetitions. That means $\mu_i = \lambda_{k_i}$ in our old terminology, where we agree that there are $r \leq m$ distinct eigenvalues. We define

$$\theta(x) = \inf_{h \in \mathbb{R}^n} \sup_{i=1,\ldots,r} \sup_{\mathrm{Tr}(Y_i)=1,\, Y_i \succeq 0} -f(x) + \mu_i(X) + Y_i \bullet Q_i^H[\mathcal{F}'(x)h]Q_i + \tfrac{1}{2}\delta\|h\|^2 \tag{19}$$

Here Q_i has $k_{i+1} - 1$ columns which form an orthonormal basis of the invariant subspace of X associated with the first $k_{i+1} - 1$ eigenvalues. It is immediately clear that $\theta(x) \leq 0$, because putting $h = 0$ gives the upper bound $\theta(x) \leq \sup_{i=1,\ldots,r} -f(x) + \mu_i(X) = -f(x) + \mu_1(X) = 0$.

Lemma 3. *We have $\theta(x) = 0$ if and only if $0 \in \partial f(x)$.*

Proof. The easiest way to see this is to swap max and min in (19). This requires that we first replace the inner double supremum in (19) by a double supremum over the convex hull of $Y_i \succeq 0$, $\mathrm{Tr}(Y_i) = 1$, $i = 1, \ldots, r$, a manoeuvre which does not change the value $\theta(x)$. Then we use Fenchel duality to interchange the inner (double) supremum and the outer infimum, which goes again without changing the value. The now inner infimum is unconstrained and may be computed explicitly. For fixed Y_i and convex coefficients τ it is attained at $h(x) = \delta^{-1}\sum_{i=1}^r \tau_i \mathcal{F}'(x)^\star Q_i Y_i Q_i^H$. Substituting this back into (19) leaves the dual expression

$$\theta(x) = \sup_{\tau_i \geq 0, \sum_i \tau_i = 1} \sup_{Y_i \succeq 0, \mathrm{Tr}(Y_i)=1} -f(x) + \sum_{i=1}^r \tau_i \mu_i(X) - \frac{1}{2\delta}\left\| \sum_{i=1}^r \tau_i \mathcal{F}'(x)^\star Q_i Y_i Q_i^H \right\|^2 \tag{20}$$

which the reader recognizes as a semidefinite program. Since $\mu_i(X) < f(x)$ for $i \geq 2$, equality $\theta(x) = 0$ is only possible when $\tau_2 = \ldots = \tau_r = 0$ and hence

$$0 = \theta(x) = \sup_{Y_1 \succeq 0, \mathrm{Tr}(Y_1)=1} -\frac{1}{2\delta}\left\|\mathcal{F}'(x)^\star Q_1 Y_1 Q_1^H\right\|^2$$

But the quantity on the right hand side is only zero when $\mathcal{F}'(x)^\star Q_1 Y_1 Q_1^H = 0$, i.e., when $0 \in \partial f(x)$. The latter follows readily from the representation

$$\partial f(x) = \{\mathcal{F}'(x)^\star Q_1 Y_1 Q_1^H : Y_1 \succeq 0, \mathrm{Tr}(Y_1) = 1\}$$

of the subdifferential $\partial f(x)$. □

As a byproduct of the proof via duality we have the following

Corollary. *The infimum in (19) is attained at*

$$h(x) = -\frac{1}{\delta}\sum_{i=1}^{r} \tau_i \mathcal{F}'(x)^\star Q_i Y_i Q_i^H, \tag{21}$$

where (τ, Y) is the solution of the dual program (20). As soon as $\theta(x) < 0$, $h(x)$ is a direction of descent of f at x. □

In order to establish the utility of θ as a measure of descent, we need to solve two more questions. Firstly, we need to show that decrease at x may be quantified by θ. Secondly, we have to establish continuity of θ.

Lemma 4. *The function θ is continuous.*

Proof. Notice that by the dual formula (20), $\theta(x)$ is the supremum of an infinite family of functions of the form $-f(x) + \sum_{i=1}^{r} \tau_i \mu_i(X) + c = -f(x) + \sum_{j=1}^{m} \sigma_j \lambda_j(X) + c$ for $\sigma_j = \tau_i/(k_{i+1} - k_i)$, where $k_i \le j < k_{i+1}$, and where $c = -1/2\delta \| \sum_i \tau_i \mathcal{F}'(x) Q_i Y_i Q_i^H \|^2$ does not depend on x. Notice that $\mu_i(X) = (k_{i+1} - k_i)^{-1} \sum_{j=k_i}^{k_{i+1}-1} \lambda_j(X)$. This shows that $\theta(x)$ the supremum of continuous functions, indexed by (τ, Y). θ is therefore lower semi-continuous. It remains to prove that θ is also upper semi-continuous.

Let $x_j \to x$ such that $\theta(x_j) \to \theta$. We have to show $\theta \le \theta(x)$. We use the representation (20). Let $\varepsilon_j > 0$, $\varepsilon_j \to 0$ and choose τ_i^j and Y_i^j such that

$$\theta(x_j) \le -f(x_j) + \sum_{i=1}^{r} \tau_i^j \mu_i(X_j) - \frac{1}{2\delta}\left\| \sum_{i=1}^{r} \tau_i^j g_i^j \right\|^2 + \varepsilon_j,$$

where $g_i^j = \mathcal{F}'(x_j)^\star Q_i^j Y_i^j Q_i^{jH}$. Passing to a subsequence if necessary, we may assume that each X_j has exactly r eigenvalue gaps, which remain in the same places k_i, $i = 1, \ldots, r$. That is $\mu_i(X_j) = \lambda_{k_i}(X_j)$. Passing to yet another subsequence, we may assume $\tau^j \to \tau$, $Y_i^j \to Y_i$ and $Q_i^j \to Q_i$, where the limiting elements are all of the same types and dimensions as the elements at stage j.

However, $\lambda_{k_i}(X)$ are no longer the distinct eigenvalues of X, because some of the distinct $\mu_i(X_j) = \lambda_{k_i}(X_j)$ may coalesce in the limit $j \to \infty$. Suppose for instance that $\lim_{j\to\infty} \mu_i(X_j) = \ldots = \lim_{j\to\infty} \mu_{i+t}(X_j)$, so that the blocks $i, i+1, \ldots, i+t$ coalesce in the limit, forming a new larger eigenvalue block of X:

$$\lambda_{k_i - 1}(X) > \lambda_{k_i}(X) = \ldots = \lambda_{k_{i+t+1}-1}(X) > \lambda_{k_{i+t}}(X),$$

which is represented by a certain $\mu_\nu(X)$. Suppose there are N such new block leaders. For each of these $\mu_\nu(X)$ we define

$$\sigma_\nu = \lim_{j\to\infty} \sum_{s=i}^{i+t} \tau_s^j,$$

then

$$\lim_{j\to\infty} \sum_{i=1}^{r} \tau_i^j \mu_i(X_j) = \sum_{\nu=1}^{N} \sigma_\nu \mu_\nu(X),$$

where $\sum_{\nu=1}^{N} \sigma_\nu = 1$. This represents the limiting linear term in (20) in a new form suited for the dual representation of $\theta(x)$.

We next have to treat the norm square term arising in (20) in much the same way. Notice first that the Q_i^j, $i = 1, \ldots, r$ together form a nested sequence of basis vectors adding up to an orthonormal basis of eigenvectors of X_j. Passing to the limit $j \to \infty$ gives an orthonormal basis of eigenvectors of X. We regroup it according to the eigenvalue gaps of X and rename the corresponding parts $P_1 \subset P_2 \subset \ldots P_N$. All that remains to do is to re-write the limit $\lim_{j\to\infty} \sum_{i=1}^{r} \tau_i^j Q_i^j Y_i^j Q_i^{jH}$ in the form $\sum_{\nu=1}^{N} \sigma_\nu P_\nu Z_\nu P_\nu^H$ for certain $Z_\nu \succeq 0$ with $\mathrm{Tr}(Z_\nu) = 1$. This is done by writing

$$Q_i Y_i Q_i^H = Q_j \begin{bmatrix} Y_i & 0 \\ 0 & 0 \end{bmatrix} Q_j^H = P_\nu \begin{bmatrix} Y_i & 0 \\ 0 & 0 \end{bmatrix} P_\nu^H$$

whenever $i < j$ have to be regrouped in the same P_ν, and where j is the last among the old indices subsumed into the new index ν, so that $Q_j = P_\nu$. Then

$$Z_\nu = \sum_{s=i}^{i+t} \frac{\tau_i}{\sigma_\nu} \begin{bmatrix} Y_i & 0 \\ 0 & 0 \end{bmatrix}$$

is as required, because $\sum_{s=i}^{i+t} \tau_s = \sigma_\nu$, hence $\mathrm{Tr}(Z_\nu) = 1$, while $Z_\nu \succeq 0$ is clear.

The argument shows that $\theta = -f(x) + \sum_{\nu=1}^{N} \sigma_\nu \mu_\nu(X) - \frac{1}{2\delta} \left\| \sum_{\nu=1}^{N} \sigma_\nu \mathcal{F}'(x)^\star P_\nu Z_\nu P_\nu^H \right\|^2$, which is to say that θ is now of the form required to be admitted to the supremum (20) defining $\theta(x)$. In other words, $\theta \leq \theta(x)$, and this is what we had to prove. □

Notice that [40, Thm. 2.1.6 (e)] is obtained as a special case of Lemma 4 if the operator $\mathcal{F}(x)$ is specialized to a diagonal matrix. The extension to multiple eigenvalues is possible because *all* the eigenvalues are taken into account simultaneously. For large matrices, this may again seem inconvenient since it will lead to large SDPs in (20).

Lemma 5. *The mapping $h(x)$ defined by (21) is continuous.*

The proof follows along the lines of the previous Lemma and is therefore omitted. Let us now construct the mapping $\vartheta(x)$ which allows to quantify descent of $f = \lambda_1 \circ \mathcal{F}$ at x. Fix $0 < \tau < 1$. Then

$$\begin{aligned} f'(x; h(x)) &= \sup_{Y \succeq 0, \mathrm{Tr}(Y)=1} h(x)^T \left(\mathcal{F}'(x)^\star Q_1 Y Q_1^H \right) \\ &= \sup_{Y \succeq 0, \mathrm{Tr}(Y)=1} Y \bullet \left(Q_1^H [\mathcal{F}'(x) h(x)] Q_1 \right) \\ &\leq \theta(x) - \frac{1}{2\delta} \|h(x)\|^2 < \theta(x) < 0 \end{aligned}$$

which follows readily from the dual formula (20) for θ if we use the fact that $-f(x) + \mu_1(X) = 0$. Now we have the following

Lemma 6. *Let $0 < \tau < 1$. There exist $\varepsilon > 0$ such that $f(x' + th(x')) - f(x') \leq t\tau\theta(x')$ for every $x' \in B(x, \varepsilon)$ and for $t > 0$ small enough.*

Proof. By the definition of the directional derivative and the fact that $0 < \tau < 1$ we have

$$\begin{aligned} f(x+th(x)) - f(x) &\leq t\tau f'(x;h(x)) \\ &\leq t\tau \left(\theta(x) - \frac{1}{2\delta}\|h(x)\|^2\right) \end{aligned}$$

for some $t_0(x) > 0$ and all $0 < t < t_0$. Since $h(\cdot)$ and $\theta(\cdot)$ are continuous, we can find a neighbourhood $B(x,\varepsilon)$ of x such that for $x' \in B(x,\varepsilon)$ and some $t_1(x) > 0$, $f(x'+th(x')) - f(x') \leq t(\tau/2)\theta(x)$ for all $0 < t < t_1(x)$. We therefore have but to choose $\delta = -t_1(x)(\tau/2)\theta(x)$ in order to obtain a descent step with property (ii). To define $s(x)$ properly, it suffices to specify any procedure which picks a step $t(x')$ close to $t_1(x)$. For instance, [40, 3c, p. 223] and also [38] is such a procedure, but other choices are possible. □

6.5 The semi-infinite case

Our last step is now to address the semi-infinite case. We are in the situation (17). Suppose that for finite $\Omega \subset \mathbb{R}$ we already dispose of a descent step generator s_Ω for f_Ω, satisfying axioms (i) and (ii). This is naturally the case when the $f(x,\omega)$ are maximum eigenvalue functions, because a finite maximum of maximum eigenvalue functions is itself a maximum eigenvalue function. So here we obtain s_Ω as in sections 6.3 or 6.4. Suppose now that we can specify a sequence of finite sets $\Omega_1 \subset \Omega_2 \subset \ldots$ such that the following conditions are satisfied:

(iii) If $0 \notin \partial f(x)$, then $\limsup_{k\to\infty} f(s_{\Omega_k}(x)) - f(x) < 0$.

(iv) For every $x \in \mathbb{R}^n$ and $\varepsilon > 0$, $\lim_{k\to\infty} \max_{x' \in B(x,\varepsilon)} \inf_{\omega' \in \Omega_k} |f(x',\omega) - f(x',\omega')| = 0$.

Notice that both axioms guarantee that the approximation Ω_k improves with growing k. Axiom (iii) tells that as soon as $0 \notin \partial f(x)$, descent steps may be eventually generated by using approximations f_{Ω_k} of f. Axiom (iv) is simply saying that approximations get better as k increases, and that this happens uniformly on small neighbourhoods of each x.

Using these axioms, let us construct a descent step generator $s(\cdot)$ for f. We proceed as follows:

<u>Semi-infinite descent step generator</u>

1. If $0 \in \partial f(x)$ then $s(x) = x$ and return. Otherwise put counter $k = 1$ and continue.
2. At counter k, if $0 \in \partial f_{\Omega_k}(x)$, then increase k until $0 \notin \partial f_{\Omega_k}(x)$.
3. At counter k with $0 \notin \partial f_{\Omega_k}(x)$, compute the descent step $s_{\Omega_k}(x)$ for f_{Ω_k} at x. Let $\varepsilon_k, \delta_k > 0$ be such that
$$f_{\Omega_k}(s_k(x')) - f_{\Omega_k}(x') \leq -\delta_k < 0$$
for every $x' \in B(x,\varepsilon_k)$, as guaranteed by axiom (ii) for s_{Ω_k}.
4. Compute
$$\eta_k := \inf\{|f(x',\omega) - f(x',\omega')| : \omega' \in \Omega_k, x' \in B(x,\varepsilon_k) \cup s\left(B(x,\varepsilon_k)\right)\}$$
5. If $3\eta_k < \delta_k$, then let $s(x) = s_{\Omega_k}(x)$ and stop. Otherwise increase k by one and go back to step 3.

We have to make sure that this scheme is well-defined and introduces a step generator $s(\cdot)$ for the infinite maximum function f.

Lemma 7. *The descent step generator $s(\cdot)$ for $f(x) = \sup_\omega f(x,\omega)$ is well-defined and satisfies axioms (i) and (ii). Moreover, each of the above loops ends after a finite number of iterations.*

Proof. Notice first that if $0 \notin \partial f(x)$, then descent around x is possible. Since $f_{\Omega_k}(x) \to f(x)$ as $k \to \infty$, we can also decrease f_{Ω_k} around x for k sufficiently large. So step 2 ends with a descent step of f_{Ω_k} at x after a finite number of trials. Moreover, this remains so for the following counters k, because $\Omega_k \subset \Omega_{k+1}$.

Next observe that $\eta_k \to 0$ by axiom (iv), while $\limsup -\delta_k < 0$ by axiom (iii). That means $3\eta_k < \delta_k$ for k sufficiently large, i.e., the procedure ends in step 4 after a finite number of updates $k \to k+1$. It remains to check that s is as required.

Suppose $f_{\Omega_k}(s_{\Omega_k}(x')) - f_{\Omega_k}(x') \le -\delta_k$ for every $x' \in B(x, \varepsilon_k)$. Then $|f(s_{\Omega_k}(x')) - f_{\Omega_k}(s_{\Omega_k}(x'))| \le \eta_k$, $|f(x') - f_{\Omega_k}(x')| \le \eta_k$, hence $f(s_{\Omega_k}(x')) - f(x') \le -\delta_k + 2\eta_k \le -\frac{1}{3}\delta_k$. This proves axiom (ii). $\square$

The procedure is sufficiently flexible to accommodate a problem oriented step generation. We may adapt the choice of Ω_k to the structure of f, and what is more important, to the local behavior of f around the current x.

6.6 First-order algorithm for the H_∞-norm

We are now ready to present our algorithm, which follows the lines of the previous sections. We start with a version based on sections 6.3 and 6.5.

First-order algorithm for the H_∞-norm: variant I

Fix $0 < \tau < 1$.
1. Given x_k choose a finite set Ω_k containing $\Omega(x_k)$. 2. Compute $M(x_k)$ for f_{Ω_k} according to (18). If $M(x_k) = 0$ then stop, because $0 \in \partial f(x)$. Otherwise 3. Use a line search to find a step t_k such that the predicted decrease satisfies $\pi_k = f_{\Omega_k}(x_k + t_k h_k) - f(x_k) \le -\kappa(x_k) M(x_k) < 0$. Compare with the actual decrease $\alpha_k = f(x_k + t_k h_k) - f(x_k)$. If $\alpha_k \le \tau \pi_k$, accept t_k, put $x_{k+1} = x_k + t_k h_k$ and goto step 5. 4. If $\alpha_k > \tau \pi_k$ then reject t_k and add nodes to Ω_k to obtain the finer mesh Ω_{k+1}. Increase counter k by one and go back to step 2. 5. Increase counter k by one and go back to step 1.

In steps 3 and 4 of the algorithm we recognize the mechanism of the previous section, which obtains a descent step generator for the semi-infinite function by using those of the finite models f_{Ω_k}. We accept the step for f if it exceeds a small fraction of the descent predicted by the finite model f_{Ω_k}. This is essentially the same procedure as in 6.5.

Next comes the version based on section 6.4 in tandem with section 6.5. The semi-infinite case is handled in exactly the same fashion, but the descent step generators s_{Ω_k} are different.

First-order algorithm for the H_∞-norm: variant II

Fix $0 < \tau_1, \tau_2 < 1$ and $\delta > 0$.
1. Given x_k choose a finite set Ω_k containing $\Omega(x_k)$. 2. Compute the value θ_k and the solution (τ^k, Y^k) of the SDP (19). If $\theta_k = 0$, then stop because $0 \in \partial f(x_k)$. Otherwise compute the descent direction h_k for f_{Ω_k} at x_k according to (21). 3. Using a line search, find a step t_k such that the predicted decrease satisfies $\pi_k = f_{\Omega_k}(x_k + t_k h_k) - f(x_k) < \tau_1 t_k \theta_k < 0$. Compare with the actual decrease $\alpha_k = f(x_k + t_k h_k) - f(x_k)$. If $\alpha_k \leq \tau_2 \pi_k$, accept t_k, put $x_{k+1} = x_k + t_k h_k$ and goto step 5. 4. If $\alpha_k > \tau_2 \pi_k$ then reject t_k and add nodes to Ω_k to obtain the finer mesh Ω_{k+1}. Increase counter k by one and go back to step 2. 5. Increase counter k by one and go back to step 1.

6.7 Practical aspects

In this section we comment on the salient features of the nonsmooth first-order descent algorithm and address some of the practical aspects.

In our testing we have observed that often the leading eigenvalues $\lambda_1(\mathcal{F}(x_k, j\omega_\nu))$ have multiplicity 1 at all frequencies $\omega_\nu \in \Omega(x_k)$. In this situation step 2 of the descent algorithm variant II could be simplified. Suppose at the current iterate x_k we have selected a finite set of frequencies $\Omega_k = \{\omega_1, \ldots, \omega_q\}$ containing $\Omega(x_k)$. Suppose all $\lambda_1(\mathcal{F}(x_k, j\omega_\nu))$ have multiplicity 1. Let $g_\nu = \mathcal{F}'(x_k, j\omega_\nu)^\star e_\nu e_\nu^T = f'(x_k; \omega_\nu)$, where e_ν is the normalized eigenvector associated with $\lambda_1(\mathcal{F}(x_k, j\omega_\nu))$. Then the semidefinite program (16) respectively (20) simplifies to a convex quadratic program

$$\theta(x_k) = \sup_{\tau_\nu \geq 0, \sum_\nu \tau_\nu = 1} -f_\Omega(x_k) + \sum_{\nu=1}^{q} \tau_\nu f(x_k, \omega_\nu) - \frac{1}{2\delta} \left\| \sum_{\nu=1}^{q} \tau_\nu g_\nu \right\|^2$$

and the associated direction of descent $h_k = h(x_k)$ is

$$h(x_k) = -\frac{1}{\delta} \sum_{\nu=1}^{q} \tau_\nu g_\nu,$$

where τ is the optimal solution of the quadratic program. Observe that for $\Omega_k = \Omega(x_k)$, $h(x_k)$ coincides with the steepest descent direction for a finite max function.

Let us now specify in which way we select the frequency set Ω_k at each step. The finite set of frequencies $\Omega(x_k)$ where the H_∞-norm is attained is computed via the Hamiltonian technique [9]. We then form an enriched set Ω_k of frequencies by adding to the peak frequencies a collection of logarithmically spaced frequencies ω_ν such that

$$\|T_{w\to z}(K)\|_\infty - \overline{\sigma}(T_{w\to z}(K, j\omega_\nu)) \leq \varepsilon_\omega \, \|T_{w\to z}(K)\|_\infty \,,$$

where ε_ω is a user-specified tolerance. We usually limit the set to the first 50 frequencies with largest singular values, as this appears to work well on a broad range of numerical tests. Typical

values for ε_ω range from 0.05 to 0.5. The algorithm requires that this set be iteratively refined when descent steps cannot be computed, but in practice our choice is usually satisfactory, and numerical problems due to exceedingly large Ω_k can be avoided.

7 Numerical experiments

In this section we test our nonsmooth algorithms on a variety of synthesis problems from the $COMPl_eib$ collection by F. Leibfritz [32]. Computations were performed on a (low-level) SUN-Blade Sparc with 256 RAM and a 650 MHz sparcv9 processor. LMI-related computations for search directions used the LMI Control Toolbox [24] or our home made SDP code [5].

Our algorithm is a first-order method. Not surprisingly, it may be slow in the neighborhood of a local solution. We have implemented various stopping criteria to ensure that an adequate approximation of a solution has been found and to avoid unwarranted computational efforts as is often the case with a first-order algorithm. The first of these termination criteria is an absolute stopping test, which provides a criticality assessment

$$\inf\{\|g\| : g \in \partial f(x)\} < \varepsilon_1, \tag{22}$$

This is reasonable, as $0 \in \partial f(x)$ indicates a critical point. It is also mandatory to use relative stopping criteria to reduce the dependence on the problem scaling. The test

$$\|T_{w\to z}(K)\|_\infty - \|T_{w\to z}(K^+)\|_\infty < \varepsilon_2(1 + \|T_{w\to z}(K)\|_\infty)\,, \tag{23}$$

compares the progress achieved relatively to the current H_∞ performance, while

$$\|K^+ - K\| < \varepsilon_3(1 + \|K\|) \tag{24}$$

compares the step-length to the controller gains. The tolerances

$$\varepsilon_1 = 1e\text{–}5,\ \varepsilon_2 = 1e\text{–}3,\ \varepsilon_3 = 1e\text{–}3$$

have been used in our numerical testing. For stopping we required that either the first two tests or the third one are satisfied.

7.1 Stabilization

Pure stabilization may be regarded as H_∞ synthesis under the special form (6). The optimization program is (5), but we stop the algorithm as soon as a stabilizing controller is obtained. Iterating until a local optimum of f is reached does not seem to improve any of the usual performance specifications of the stabilizing feedback controller. Such questions have to be addressed in a second procedure, where the stabilizing controller will serve as a starting point. The spectral abscissa (7) is used to check stability of K, but is not used as a cost function. Previous experiments where (7) was used to find stabilizing controllers are reported in [3].

Table 1 displays results obtained for static stabilization problems borrowed from the literature. The triple (n, m, p) gives the number of states, inputs and outputs. Column 'iter' corresponds to the number of iterations required to meet the combined stopping tests. Column α displays the final closed-loop spectral abscissa (7). Negative $\alpha < 0$ indicates a stable closed-loop system. Column

'cpu' gives the cputime in seconds. Note that the reported cputimes are only indicative as the α-version of our code includes a number of extra tests and even graphics that have been exploited for fine tuning of the various algorithm parameters. Column 'Ref.' points to the associated reference.

The initial shift for the H_∞ norm minimization was chosen according to the following rule

$$a_0 := \max\left(\alpha(A) * (1 + 10\%),\ 1e - 2\right) , \tag{25}$$

with a zero initial static controller $K_0 = 0$.

From the table, we observe that in all cases but one, the original choice a_0 of the shift was sufficient. As indicated, there is no need to run the algorithm until convergence. A stabilizing solution is therefore obtained very early. A single SDP or convex QP with line search suffices in most cases.

In the second example, we had to reduce the shift three times before stability was reached. This was done according to the same rule in (25) applied to the closed-loop dynamics. In this example the H_∞ criterion is flat about a (global) optimum, which allows longer steps and renders the step-length-based termination criterion more stringent. Note that in this case the solution is globally optimal because a zero H_∞ norm is reached.

problem	(n, m, p)	iter	α	cpu (sec.)	Ref.
Transport airplane	$(9, 1, 5)$	1	$-2.22e{-}2$	2.26	[25]
Horisberger's example	$(9, 1, 4)$	4	-0.01	12.0	[28]
VTOL helicopter	$(4, 2, 1)$	1	$-6.00e{-}2$	1.90	[31]
Chemical reactor	$(4, 2, 2)$	1	-1.73	2.26	[30]
Piezoelectric actuator	$(5, 1, 3)$	1	$-9.95e{-}1$	2.88	[14]
Boeing 767	$(55, 2, 2)$	1	$-2.33e{-}2$	6.47	[19]

TABLE 1: Static output-feedback stabilization
steepest descent method - $\varepsilon_\omega = 0.05$

For test reasons, the pure stabilization problems have been solved with the steepest descent method in section 5. As expected, this technique is less stable than algorithmic variants I and II. Specifically, the choice of ε_ω is rather critical and finding a general selection rule appears difficult. Together with our analysis in section 6, this encouraged us to use algorithmic variant II in the syntheses problems below.

7.2 H_∞ SYNTHESIS PROBLEMS

This section is devoted to H_∞ synthesis problems. The synthesis procedure is based on the scheme (3) and must be initialized with a stabilizing controller. This initial phase I is described in the previous section, but alternative techniques to find an initial stabilizing controller K may prove useful [3]. All examples are extracted from the $COMPl_eib$ collection [32]. Column 'problem' now indicates the $COMPl_eib$ acronym attached to each example.

Notice that for a static K, our program could formally be solved as

$$\begin{array}{ll} \underset{K}{\text{minimize}} & \|T_{w\to z}(K)\|_\infty \\ \text{subject to} & \|C_2\left(sI - A(K)\right)^{-1} B_2\|_\infty \le M \end{array} \tag{26}$$

where $M > 0$ is a constant e.g. such that the initial stabilizing controller K_0 satisfies

$$\|C_2 (sI - A(K_0))^{-1} B_2\|_\infty \ll M < +\infty$$

and where $A(K)$ refers to the closed-loop A matrix. Note that the extra constraint in (26) enforces asymptotic stability of the closed-loop system.

It often happens that stability of the performance channel $T_{w\to z}(K)$ alone implies stability of the closed-loop system, so that the (internal) stability constraint is redundant. This observation is significant, as the problem becomes unconstrained. When this does not hold true, we introduce the composite problem

$$\underset{K}{\text{minimize}} \left\| \begin{bmatrix} T_{w\to z}(K) & 0 \\ 0 & \varepsilon_s C_2 (sI - A(K))^{-1} B_2 \end{bmatrix} \right\|_\infty$$

where the lower-right term enforces internal stability, with $\varepsilon_s > 0$ a small enough parameter. The new problem can now be handled by the techniques discussed so far without change. Similar modifications readily extend to any design problem with additional structural constraints on the controller.

We compare the results of our nonsmooth algorithm variant II in columns 'nonsmooth H_∞' to older results obtained with the specialized augmented Lagrangian (AL) algorithm described in [6], displayed in columns 'H_∞ AL'. In column 'H_∞ full' we also display the gain obtained with a full-order feedback controller, synthesized by LMI-methods or via the algebraic Riccati equation solver. This is a lower bound for the gain in column 'nonsmooth H_∞'. The results obtained with our present technique are close to those obtained in [6], except for problems with large state dimension as 'AC10' (55 states), 'BDT2' (82 states) and 'HF1' (130 states) where the augmented Lagrangian method fails, while the present nonsmooth method is still functional. In the same vein, we have observed that even customary Riccati or LMI solvers encounter serious difficulties or even break down when solving the full-order problem for 'AC10', 'BDT2' and 'HF1'.

The nonsmooth technique achieves the same H_∞ performance as the AL algorithm for which convergence to a local minimum is established in [36].

As an illustration, figure 1 shows for example 'AC8' the evolution of the maximum singular value of $T_{w\to z}(K)$ for the first 5 iterations and indicates the frequencies ω_i that were regrouped in the set Ω_k to construct a bundle of Clarke subgradients at iterate x_k. Also, figure 2 depicts the evolution of the absolute value of the criticality measure of algorithm variant II (19) and (20) vs. iterations. As theoretically expected, it gradually tends to zero until a local minimum is reached. Finally, controllers for large problems 'AC10', 'BDT2' and 'HF1' are given in table 3.

problem	(n, m, p)	order	iter	cpu (sec.)	nonsmooth H_∞	H_∞ AL	H_∞ full
AC8	$(9, 1, 5)$	0	20	45	2.005	2.02	1.62
HE1	$(4, 2, 1)$	0	4	7	0.154	0.157	0.073
REA2	$(4, 2, 2)$	0	31	51	1.192	1.155	1.141
AC10	$(55, 2, 2)$	0	15	294	13.11	intractable	3.23
AC10	$(55, 2, 2)$	1	46	408	10.21	intractable	3.23
BDT2	$(82, 4, 4)$	0	44	1501	0.8364	intractable	0.2340
HF1	$(130, 1, 2)$	0	11	1112	0.447	intractable	0.447

TABLE 2: H_∞ synthesis with nonsmooth algorithm
algorithmic variant II - $\varepsilon_\omega = 0.05$

problem	order	K(s)
AC10	0	$K = \begin{bmatrix} -9.0747e{-1} & 2.1249e{-5} \\ 4.3042 & 2.2467e{-5} \end{bmatrix}$
AC10	1	$\begin{bmatrix} -6.110e{-2} \\ -2.361e{-2} \end{bmatrix} (s+2.633)^{-1} \begin{bmatrix} -6.476e{-2} & 1.245e{-2} \end{bmatrix} + \begin{bmatrix} 2.698e{-1} & 1.389e{-5} \\ -3.285e{-01} & 5.226e{-5} \end{bmatrix}$
BDT2	0	$\begin{bmatrix} -1.0402 & 1.2997 & 1.4684 & 3.0555 \\ -3.7306 & -3.5572 & -1.9894 & 2.2705 \\ -2.8162 & 2.1944 & 1.8146 & 9.6202 \\ -9.8493 & -5.5376 & -2.4885 & 5.1699 \end{bmatrix}$
HF1	0	$\begin{bmatrix} -0.1907 & -1.3093 \end{bmatrix}$

TABLE 3: H_∞-controllers for large problems
computed via algorithm variant II

8 CONCLUSION

We have proposed several new algorithms to minimize the H_∞-norm subject to structural constraints on the controller dynamics. The proposed method uses nonsmooth techniques suited for H_∞ synthesis and for semi-infinite eigenvalue or singular value optimization programs. Variant I and variant II of our algorithm are supported by global convergence theory, a crucial parameter for the reliability of an algorithm in practice. Variant II has been shown to perform satisfactorily on a number of difficult examples. In particular, three examples with large state dimension ($n = 55$ $n = 82$ and $n = 130$) have been solved.

More importantly, our present techniques and tools pave the way for investigating an even larger scope of synthesis problems, characterized through frequency domain inequalities of the form $\lambda_1(H(x, \omega)) \leq 0$, $\omega \geq 0$, where $H(x, \omega)$ is Hermitian-valued and x stands for controller parameters and possibly multiplier variables, as is the case when IQC formulations are used. This is a strong incentive for further developments.

References

[1] B. Anderson and S. Vongpanitlerd, *Network analysis and synthesis: a modern systems theory approach*, Prentice-Hall, 1973.

[2] P. Apkarian and P. Gahinet, *A Convex Characterization of Gain-Scheduled H_∞ Controllers*, IEEE Trans. Aut. Control, 40 (1995), pp. 853–864. See also pp. 1681.

[3] P. Apkarian and D. Noll, *Controller design via nonsmooth multi-directional search*, submitted, (2004). Rapport Interne , MIP, CNRS UMR 5640, Dép. de Math. - Université Paul Sabatier.

[4] P. Apkarian, D. Noll, J. B. Thevenet, and H. D. Tuan, *A Spectral Quadratic-SDP Method with Applications to Fixed-Order $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Synthesis*, in Asian Control Conference, Melbourne, AU, 2004.

[5] ——, *A Spectral Quadratic-SDP Method with Applications to Fixed-Order $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Synthesis*, in Asian Control Conference, Melbourne, AU, 2004.

[6] P. Apkarian, D. Noll, and H. D. Tuan, *Fixed-order $\mathcal{H}_\infty$ control design via an augmented Lagrangian method* , Int. J. Robust and Nonlinear Control, 13 (2003), pp. 1137–1148.

[7] V. Blondel and M. Gevers, *Simultaneous stabilizability question of three linear systems is rationally undecidable*, Mathematics of Control, Signals, and Systems, 6 (1994), pp. 135–145.

[8] 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.

[9] S. Boyd, V. Balakrishnan, and P. Kabamba, *A bisection method for computing the $\mathbf{H}_\infty$ norm of a transfer matrix and related problems*, Mathematics of Control, Signals, and Systems, 2 (1989), pp. 207–219.

[10] S. Boyd and C. Barratt, *Linear Controller Design: Limits of Performance*, Prentice-Hall, 1991.

[11] J. Burke, A. Lewis, and M. Overton, *Two numerical methods for optimizing matrix stability*, Linear Algebra and its Applications 351-352, (2002), pp. 147–184.

[12] ——, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, submitted to SIAM J. Optimization , (2003).

[13] ——, *Robust stability and a criss-cross algorithm for pseudospectra*, IMA Journal of Numerical Analysis, 23 (2003), pp. 1–17.

[14] B. M. Chen, *H_∞ Control and Its Applications*, vol. 235 of Lectures Notes in Control and Information Sciences, Springer Verlag, New York, Heidelberg, Berlin, 1998.

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

[16] 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.

[17] C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties*, Academic Press, New York, 1975.

[18] J. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, *State-space solutions to standard* $\mathbf{H}_2$ *and* $\mathbf{H}_\infty$ *control problems*, in Proc. American Control Conf., 1988, pp. 1691–1696.

[19] E. E. J. Davison, *Benchmark problems for control system design*, tech. rep., Oxford, Pergamon Press, IFAC Technical Committee Reports, 1990.

[20] F. Leibfritz and E. M. E. Mostafa, *Trust region methods for solving the optimal output feedback design problem*, International Journal of Control, 76 (2000), pp. 501 – 519.

[21] P. Gahinet and P. Apkarian, *Numerical Computation of the* L_∞ *Norm Revisited*, in Proc. IEEE Conf. on Decision and Control, 1992.

[22] ——, *A LMI-based parametrization of all* $\mathbf{H}_\infty$ *controllers with applications*, in Proc. IEEE Conf. on Decision and Control, San Antonio, Texas, Dec. 1993, pp. 656–661.

[23] ——, *A Linear Matrix Inequality Approach to* H_∞ *Control*, Int. J. Robust and Nonlinear Control, 4 (1994), pp. 421–448.

[24] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox* , The MathWorks Inc., 1995.

[25] D. Gangsaas, K. Bruce, J. Blight, and U.-L. Ly, *Application of modern synthesis to aircraft control: Three case studies*, IEEE Trans. Aut. Control, AC-31 (1986), pp. 995–1014.

[26] K. Glover, *All Optimal Hankel-Norm Approximations of Linear Multivariable Systems and their* L_∞*-Error Bounds* , Int. J. Control, 39 (1984), pp. 1115–1193.

[27] D. Henrion, M.Kocvara, and M. Stingl, *Solving simultaneous stabilization BMI problems with PENNON*, in IFIP Conference on System Modeling and Optimization, vol. 7, Sophia Antipolis, France, July 2003.

[28] H. P. Horisberger and P. R. Belanger, *Solution of the Optimal Constant Output Feedback Problem by Conjugate Gradients*, IEEE Trans. Aut. Control, 19 (1974), pp. 434–435.

[29] R. A. Horn and C. A. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[30] Y. S. Hung and A. G. J. MacFarlane, *Multivariable feedback: A classical approach*, Lectures Notes in Control and Information Sciences, Springer Verlag, New York, Heidelberg, Berlin, 1982.

[31] L. H. Keel, S. P. Bhattacharyya, and J. W. Howze, *Robust control with structured perturbations*, IEEE Trans. Aut. Control, 36 (1988), pp. 68–77.

[32] F. Leibfritz, *COMPL$_e$IB, COnstraint Matrix-optimization Problem LIbrary - a collection of test examples for nonlinear semidefinite programs, control system design and related problems*, tech. rep., Universität Trier, 2003.

[33] 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 J. on Optimization, 12 (2002), pp. 1048–1074.

[34] A. Nemirovskii, *Several NP-Hard problems arising in robust stability analysis*, Mathematics of Control, Signals, and Systems, 6 (1994), pp. 99–105.

[35] D. Noll and P. Apkarian, *First and second order spectral bundle methods for nonconvex maximum eigenvalue functions*, to appear in Maths. Programming Series B, (2004).

[36] D. Noll, M. Torki, and P. Apkarian, *Partially Augmented Lagrangian Method for Matrix Inequality Constraints*, To appear in SIAM Journal on Optimization, (2004).

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

[38] O. Pironneau and E. Polak, *On the rate of convergence of a class of methods of centers*, Mathematical Programming, 2 (1972), pp. 230–258.

[39] E. Polak, *On the mathematical foundations of nondifferentiable optimization in engineering design*, SIAM Rev., 29 (1987), pp. 21–89.

[40] ——, *Optimization : Algorithms and Consistent Approximations*, Applied Mathematical Sciences, 1997.

[41] J. Thevenet, D. Noll, and P. Apkarian, *Nonlinear spectral SDP method for BMI-constrained problems: Applications to control design*, in ICINCO proceedings, Portugal, 2004.

Acknowledgements

Thanks to Friedemann Leibfritz, Trier Universität, for providing the $COMPl_eib$ collection and for fruitful discussions.

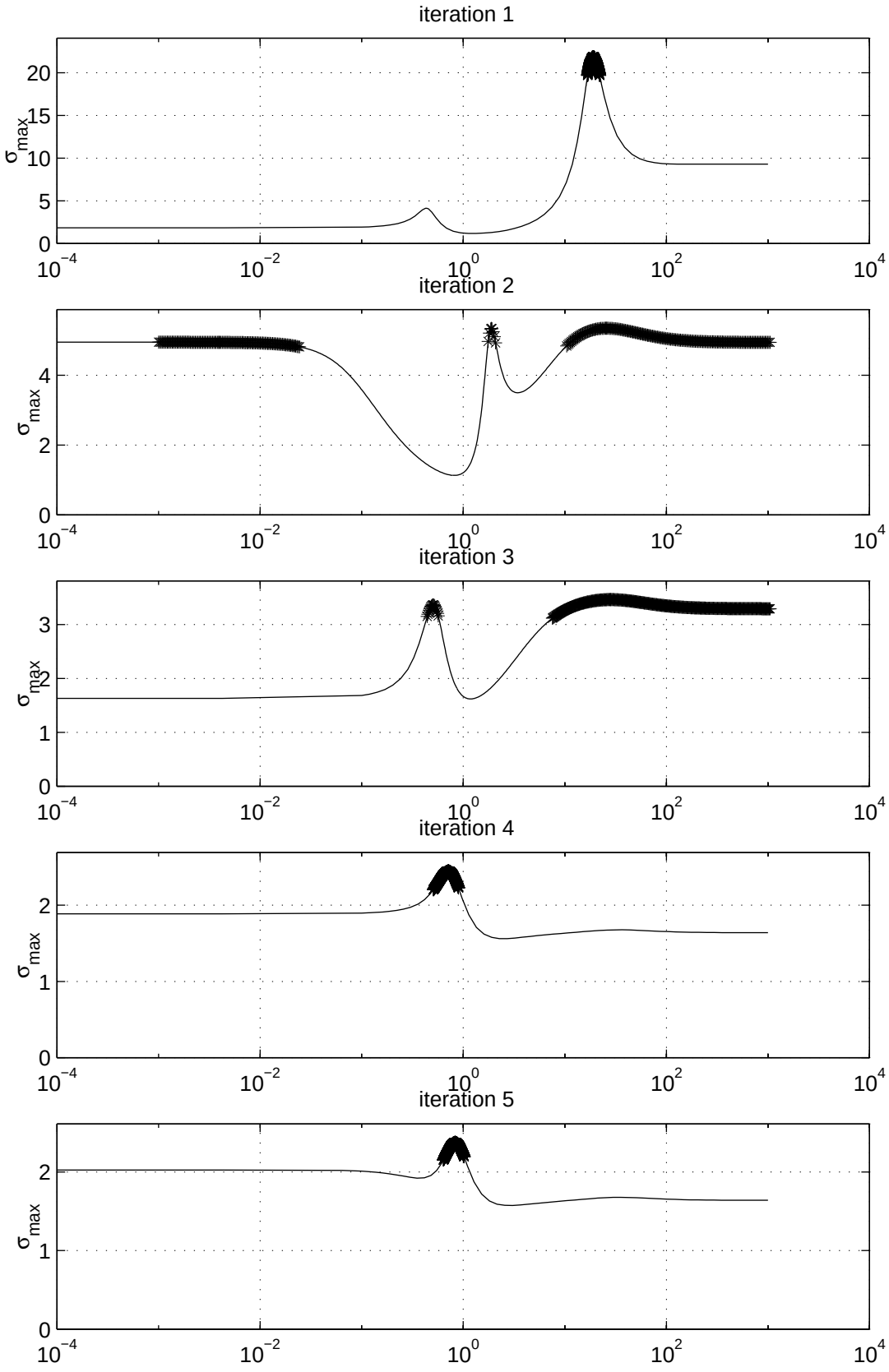

FIGURE 1: max singular values (transport airplane 'AC8')
first 5 iterations - '*' selected frequencies

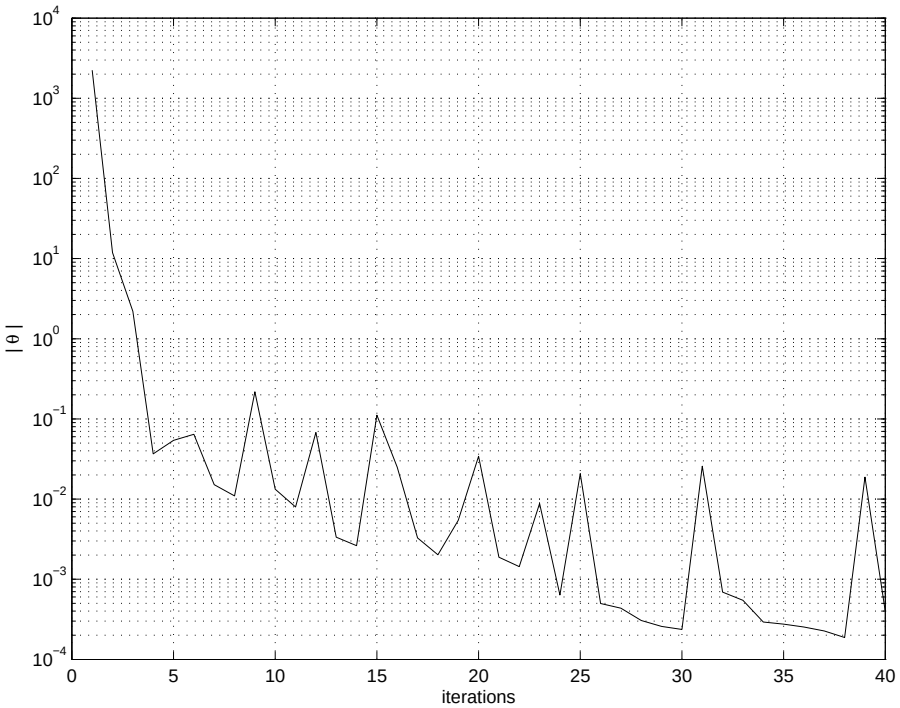

FIGURE 2: criticality measure $\theta(x)$ vs. iterations for 'BDT2'