Open access • Journal Article • DOI:10.1137/070685026

# Mixed $H_2/H_\infty$ Control via Nonsmooth Optimization — Source link

Pierre Apkarian, Dominikus Noll, A. Rondepierre

**Institutions:** Community emergency response team

**Published on:** 15 Mar 2008 - Siam Journal on Control and Optimization (Society for Industrial and Applied Mathematics)

# Mixed H2/H∞ Control via Nonsmooth Optimization

P. Apkarian, Dominikus Noll, Aude Rondepierre

# MIXED $H_2/H_\infty$ CONTROL VIA NONSMOOTH OPTIMIZATION

P. APKARIAN *, D. NOLL †, AND A. RONDEPIERRE † ‡

**Abstract.** We present a new approach to mixed $H_2/H_\infty$ output feedback control synthesis. Our method uses non-smooth mathematical programming techniques to compute locally optimal $H_2/H_\infty$-controllers, which may have a pre-defined structure. We prove global convergence of our method and present numerical tests to validate it numerically.

**Key words.** Mixed $H_2/H_\infty$ output feedback control, multi-objective control, robustness and performance, non-smooth optimization, trust region technique.

**AMS subject classifications.** 93B36, 93B50, 90C29, 49J52, 90C26, 90C34, 49J35

**1. Introduction.** Mixed $H_2/H_\infty$ output feedback control is a prominent example of a multi-objective design problem, where the feedback controller has to respond favorably to several performance specifications. Typically in $H_2/H_\infty$ synthesis, the $H_\infty$ channel is used to enhance the robustness of the design, whereas the $H_2$ channel guarantees good performance of the system. Due to its importance in practice, mixed $H_2/H_\infty$ control has been addressed in various ways over the years, and we briefly review the main trends.

The interest in $H_2/H_\infty$ synthesis was originally risen by three publications [22, 23, 27] in the late 1980s and early 1990s. The numerical methods proposed by these authors are based on coupled Riccati equations in tandem with homotopy methods, but the numerical success of these strategies remains to be established. With the rise of LMIs in the later 1990s, different strategies which convexify the problem became increasingly popular. The price to pay for convexifying is either a considerable conservatism, or that controllers have large state dimension [29, 25].

In [45, 47, 48] Scherer developed LMI formations for $H_2/H_\infty$ synthesis for full-order controllers [48], and reduced the problem to solving LMIs in tandem with nonlinear algebraic equalities [48, 45]. In this form, $H_2/H_\infty$ problems could in principle be solved via nonlinear semidefinite programming techniques like specSDP [24, 39, 49] or Pennon [31, 32, 36], if only these techniques were suited for medium or large size problems. Alas, one of the disappointing lessons learned in recent years from investigating BMI and LMI problems is that this is just not the case. Due to the presence of Lyapunov variables, whose number grows quadratically with the system size, [13, p. 20ff], BMI and LMI programs quickly lead to problem sizes where existing numerical methods fail.

Following [3, 4, 5, 6, 7], we address $H_2/H_\infty$-synthesis by a new strategy which avoids the use of Lyapunov variables. This leads to a non-smooth and semi-infinite optimization program, which we solve with a spectral bundle method, inspired by the non-convex spectral bundle method of [37, 38] and [3, 5]. Important forerunners [19, 40, 28] are based on convexity and optimize functions of the form $\lambda_1 \circ A$ with affine $A$. We have developed our method further to deal with typical control applications like multi-disk [7] and multi frequency band synthesis [6], design under integral quadratic constraints (IQCs) [4, 9, 8], and to loop-shaping techniques [2, 1].

*CERT-ONERA, 2, avenue Edouard Belin, 31055 Toulouse, France

†Université Paul Sabatier, Institut de Mathématiques, 118 route de Narbonne, 31062 Toulouse, France

‡Corresponding author

The structure of the paper is as follows. The problem setting is given in Section 2. Computing the $H_2$ and $H_\infty$ norm is briefly recalled in Sections 3 and 4. The algorithm and its rationale are presented in Section 5. Global convergence is established in Section 6. The implementation is discussed in Section 7, and numerical test examples are discussed in Section 8.

**2. Problem setting.** We consider a plant in state space form

$$P: \qquad \left[\begin{array}{c} \dot{x} \\ \hline z_\infty \\ z_2 \\ y \end{array}\right] = \left[\begin{array}{c|ccc} A & B_\infty & B_2 & B \\ \hline C_\infty & D_\infty & 0 & D_{\infty u} \\ C_2 & 0 & 0 & D_{2u} \\ C & D_{y\infty} & D_{y2} & 0 \end{array}\right] \left[\begin{array}{c} x \\ \hline w_\infty \\ w_2 \\ u \end{array}\right] \tag{2.1}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{n_u}$ the control, $y \in \mathbb{R}^{n_y}$ the output, and where $w_\infty \to z_\infty$ is the $H_\infty$, $w_2 \to z_2$ the $H_2$ performance channel. We seek an output feedback controller

$$K: \qquad \left[\begin{array}{c} \dot{x}_K \\ u \end{array}\right] = \left[\begin{array}{cc} A_K & B_K \\ C_K & D_K \end{array}\right] \left[\begin{array}{c} x_K \\ y \end{array}\right] \tag{2.2}$$

where $x_K \in \mathbb{R}^{n_K}$ is the state of the controller, such that the closed-loop system, obtained by substituting (2.2) into (2.1), satisfies the following properties:

1. **Internal stability.** $K$ stabilizes $P$ exponentially in closed-loop.
2. **Fixed $H_\infty$ performance**. The $H_\infty$ performance channel has a pre-specified performance level $\|T_{w_\infty \to z_\infty}(K)\|_\infty \leq \gamma_\infty$.
3. **Optimal $H_2$ performance**. The $H_2$ performance $\|T_{w_2 \to z_2}(K)\|_2$ is minimized among all $K$ satisfying 1. and 2.

We will solve the $H_2/H_\infty$ synthesis problem by way of the following mathematical program

$$\begin{array}{ll} \text{minimize} & f(K) := \|T_{w_2 \to z_2}(K)\|_2^2 \\ \text{subject to} & g(K) := \|T_{w_\infty \to z_\infty}(K)\|_\infty^2 \leq \gamma_\infty^2 \end{array} \tag{2.3}$$

where $T_{w_2 \to z_2}(K, s)$ denotes the transfer function of the $H_2$ closed-loop performance channel, while $T_{w_\infty \to z_\infty}(K, s)$ stands for the $H_\infty$ robustness channel. Notice that $f(K)$ is a smooth function, whereas $g(K)$ is not, being an infinite maximum of maximum eigenvalue functions. The unknown $K$ is in the space $\mathbb{R}^{(n_K+n_u)\times(n_K+n_y)}$, so the dimension $n = (n_K + n_y)(n_K + n_u)$ of (2.3) is usually small, which is particularly attractive when small or medium size controllers for large systems are sought. Notice that as a BMI or LMI problem, $H_2/H_\infty$ synthesis (2.3) would feature $n_x^2$ additional Lyapunov variables, which would arise through the use of the bounded real lemma. See e.g. [46, 13].

Remark. Naturally, the approach chosen in (2.3) to fix the $H_\infty$ performance and optimize $H_2$ performance is just one among many other strategies in multi-objective optimization. One could just as well optimize the $H_\infty$ norm subject to a $H_2$-norm constraint, or minimize a weighted sum or even the maximum of both criteria. Other ideas have been considered, and even game theoretic approaches exist [35]. □

**3. The $H_2$ norm.** In program (2.3) we minimize composite functions $f = \|\cdot\|_2^2 \circ T_{w_2 \to z_2}$, where $\|\cdot\|_2$ denotes the $H_2$-norm. Let us for brevity write $T_2 := T_{w_2 \to z_2}$ for the $H_2$ transfer channel in (2.1). The corresponding plant $P^2$ is obtained by deleting the $w_\infty$ column and the $z_\infty$ line in $P$. The objective function can be written as

$$f(K) = \|T_2(K, \cdot)\|_2^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathrm{Tr}(T_2(K, j\omega)^H T_2(K, j\omega)) d\omega.$$

Algorithmically it is convenient to compute function values using a state space realization of $P^2$:

$$P^2(s) = \left[\begin{array}{cc} 0 & D_{2u} \\ D_{y2} & 0 \end{array}\right] + \left[\begin{array}{c} C_2 \\ C \end{array}\right] (sI - A)^{-1} [\ B_2\ B\ ].$$

Introducing the closed-loop state space data:

$$\mathcal{A}(K) = \left[\begin{array}{cc} A + BD_KC & BC_K \\ B_KC & A_K \end{array}\right], \qquad \mathcal{B}_2(K) = \left[\begin{array}{c} B_2 + BD_KD_{y2} \\ B_KD_{y2} \end{array}\right],$$

$$\mathcal{C}_2(K) = [\ C_2 + D_{2u}D_KC\ D_{2u}C_K\ ], \quad \mathcal{D}_2(K) = D_{2u}D_KD_{y2} = 0,$$

we either assume $D_{2u} = 0$ or $D_{y2} = 0$, or that the controller $K$ is strictly proper, to ensure finiteness of the $H_2$ norm. Then a realization of the closed-loop transfer function $T_2$ is given as:

$$T_2(K, s) = \mathcal{C}_2(K)(sI - \mathcal{A}(K))^{-1}\mathcal{B}_2(K)$$

and (see e.g. [21]) the objective function $f$ may be re-written as

$$f(K) = \mathrm{Tr}(\mathcal{B}_2(K)^T\ X(K)\ \mathcal{B}_2(K)) = \mathrm{Tr}(\mathcal{C}_2(K)\ Y(K)\ \mathcal{C}_2(K)^T),$$

where $X(K)$ and $Y(K)$ are the solutions of two Lyapunov equations:

$$\begin{array}{l} \mathcal{A}(K)^TX(K) + X(K)\mathcal{A}(K) + \mathcal{C}_2(K)^T\mathcal{C}_2(K) = 0, \\ \mathcal{A}(K)Y(K) + Y(K)\mathcal{A}(K)^T + \mathcal{B}_2(K)\mathcal{B}_2(K)^T = 0. \end{array} \tag{3.1}$$

As observed in [42, Section 3], one proves differentiability of the objective $f$ over the set $D$ of closed-loop stabilizing controllers $K$. In order to write the derivative $f'(K)dK$ in a gradient form, we introduce the gradient $\nabla f(K)$ of $f$ at $K$ defined by:

$$f'(K)dK = \mathrm{Tr}[\nabla f(K)^T dK],$$

meaning that $\nabla f(K)$ is now an element of the same matrix space as $K$. These results lead to the following lemma which is an extension of [42, Theorem 3.2.]:

LEMMA 3.1. *The objective function $f$ is differentiable on the open set $D$ of closed-loop stabilizing gains. For $K \in D$, the gradient of $f$ at $K$ is:*

$$\nabla f(K) = 2\left[B^TX(K) + D_{2u}^T\mathcal{C}_2(K)\right]Y(K)C^T + 2B^TX(K)\mathcal{B}_2(K)D_{y2}^T,$$

*where $X(K)$ and $Y(K)$ solve (3.1).*

**4. The $H_\infty$-norm.** The next element required in (2.3) is the constraint function $g = \|\cdot\|_\infty^2 \circ T_{w_\infty \to z_\infty}$, a composite function of the $H_\infty$-norm. To compute it we will use a frequency domain representation of the $H_\infty$ norm. Let us for brevity write $T_\infty := T_{w_\infty \to z_\infty}$. The corresponding plant is $P^\infty$, obtained by deleting the $w_2$ column and the $z_2$ line in $P$. The constraint function $g$ may be written as

$$g(K) = \max_{\omega \in [0,\infty]} \overline{\sigma}\left(T_\infty(K, j\omega)\right)^2 = \max_{\omega \in [0,\infty]} \lambda_1\left(T_\infty(K, j\omega)^H T_\infty(K, j\omega)\right),$$

where $\overline{\sigma}$ is the maximum singular value of a matrix, $\lambda_1$ the maximum eigenvalue of a Hermitian matrix. We re-write this as

$$g(K) = \max_{\omega \in [0,\infty]} g(K, \omega), \qquad g(K, \omega) = \lambda_1\left(T_\infty(K, j\omega)^H T_\infty(K, j\omega)\right).$$

Then it is clear that $g(K)$ is nonsmooth with two possible sources of non-smoothness, the infinite maximum, and the maximum eigenvalue function, which is convex but nonsmooth. We present two basic results, which allow to exploit the structure of $g$ algorithmically. The following can be found in several places, e.g. [12, 11]:

LEMMA 4.1. *Let $K$ be closed-loop stabilizing. Then $g(K) = \|T_\infty(K)\|_\infty^2 < \infty$, and the set of active frequencies at $K$, defined as $\Omega(K) = \{\omega \in [0,\infty] : g(K) = g(K,\omega)\}$ is either finite, or $\Omega(K) = [0,\infty]$.*

The case $\Omega(K) = [0,\infty]$ is when the closed-loop system is all-pass. It may very well arise in practice, for instance, full order ($n_x = n_K$) optimal $H_\infty$ controllers are all-pass; see [26]. A similar result holds for full order $H_2/H_\infty$-control; see [20]. But we never observed it in cases where the order of the controller $n_K < n_x$ is way smaller than the order of the system.

The following result was already used in [5, 7]. It allows to compute Clarke subgradients of the $H_\infty$ norm and its composite function $g$. To represent it, we find it convenient to introduce the notation

$$\begin{bmatrix} T_\infty(K,s) & G_{12}^\infty(K,s) \\ G_{21}^\infty(K,s) & \star \end{bmatrix} = \begin{bmatrix} \mathcal{C}_\infty(K) \\ C \end{bmatrix} (sI - \mathcal{A}(K))^{-1} \begin{bmatrix} \mathcal{B}_\infty(K) & B \end{bmatrix} + \begin{bmatrix} \mathcal{D}_\infty(K) & D_{\infty u} \\ D_{y\infty} & \star \end{bmatrix}$$

where the closed-loop state-space data $(\mathcal{A}(K), \mathcal{B}_\infty(K), \mathcal{C}_\infty(K), \mathcal{D}_\infty(K))$ are given by:

$$\mathcal{A}(K) = \begin{bmatrix} A + BD_KC & BC_K \\ B_KC & A_K \end{bmatrix}, \qquad \mathcal{B}_\infty(K) = \begin{bmatrix} B_\infty + BD_KD_{y\infty} \\ B_KD_{y\infty} \end{bmatrix},$$

$$\mathcal{C}_\infty(K) = \begin{bmatrix} C_\infty + D_{\infty u}D_KC & D_{\infty u}C_K \end{bmatrix}, \quad \mathcal{D}_\infty(K) = D_\infty + D_{\infty u}D_KD_{y\infty}.$$

LEMMA 4.2. *(See [5, Section IV], [14, p. 304]). Suppose $K$ is closed-loop stabilizing and $\Omega(K)$ is finite. Then the Clarke subdifferential of $g$ at $K$ is the set*

$$\partial g(K) = \left\{ \Phi_Y : Y = (Y_\omega)_{\omega\in\Omega(K)}, Y_\omega \succeq 0, \sum_{\omega\in\Omega(K)} \mathrm{Tr}(Y_\omega) = 1, Y_\omega \in \mathbb{S}^{r_\omega} \right\},$$

*where $r_\omega$ is the multiplicity of $\lambda_1\left(T_\infty(K,j\omega)^H T_\infty(K,j\omega)\right)$, and where*

$$\Phi_Y = \sum_{\omega\in\Omega(K)} 2\mathrm{Re}\left(G_{21}^\infty(K,j\omega)T_\infty(K,j\omega)^H Q_\omega Y_\omega Q_\omega^H G_{12}^\infty(K,j\omega)\right)^T.$$

*Here the columns of the $m \times r_\omega$ matrix $Q_\omega$ form an orthonormal basis of the eigenspace of $T_\infty(K,j\omega)^H T_\infty(K,j\omega) \in \mathbb{S}^m$ associated with its maximum eigenvalue.*

REMARK. Notice that the result extends to the all-pass case by replacing convex combinations over a finite set $\Omega(K)$ by Radon probability measures on $[0,\infty]$. This may still be exploited algorithmically, should the case of an all-pass system ever arise in practice. Since this never occurred in our tests, this line is not investigated here. □

**5. Nonsmooth algorithm.** In this central Section we present our main result, a nonsmooth and nonconvex optimization method for program (2.3). In subsection 5.1 we will have a look at the necessary optimality conditions for program (2.3). The

algorithm is elaborated and presented in Subsections 5.2 - 5.4. The convergence proof will follow in Section 6.

As the reader will notice, our method can be applied to a larger class of programs with a structure similar to (2.3). In consequence, during what follows we aim at a certain level of generality. In particular, to comply with the more standard notation in optimization, we denote the decision variable as $x \in \mathbb{R}^n$, where $n = (n_K + n_u)(n_K + n_y)$ in our previous terminology. This means vectorization of the matrix variable previously denoted $K$.

**5.1. Optimality conditions.** Following an idea in [41], we address program (2.3) by introducing a progress function:

$$F(y;x) = \max\left\{f(y) - f(x) - \mu[g(x) - \gamma_\infty^2]_+; [g(y) - \gamma_\infty^2] - [g(x) - \gamma_\infty^2]_+\right\}, \quad (5.1)$$

where $\mu > 0$ is a fixed parameter. All we need to know about $f$ is that it is of class $C^2$, while $g$ is assumed of the form

$$g(x) = \max_{\omega\in[0,\infty]} g(x,\omega) = \max_{\omega\in[0,\infty]} \lambda_1(G(x,\omega))$$

with $G : \mathbb{R}^n \times [0,\infty] \to \mathbb{S}^m$ of class $C^2$ in the variable $x \in \mathbb{R}^n$, and jointly continuous in $(x,\omega)$. This is in accordance with our previous terminology, where $G(x,\omega) = T_\infty(K,j\omega)^H T_\infty(K,j\omega)$ with $x = \mathrm{vec}(K)$, and where $m = n_{z_\infty}$ or $m = n_{w_\infty}$, and where $n = (n_K + n_y)(n_K + n_u)$. We have the following preparatory

LEMMA 5.1. *1) If $\bar{x} \in \mathbb{R}^n$ is a local minimum of (2.3), then $\bar{x}$ is also a local minimum of $F(\cdot;\bar{x})$. In particular, this implies $0 \in \partial_1 F(\bar{x};\bar{x})$.*

*2) If $\bar{x}$ satisfies the F. John necessary optimality conditions for program (2.3), then $0 \in \partial_1 F(\bar{x};\bar{x})$.*

*3) Conversely, suppose $0 \in \partial_1 F(\bar{x};\bar{x})$ for some $\bar{x} \in \mathbb{R}^n$. Then we have the following possibilities:*

- *(i) Either $g(\bar{x}) > \gamma_\infty^2$, then $\bar{x}$ is a critical point of $g$, called a critical point of constraint violation.*
- *(ii) Or $g(\bar{x}) \le \gamma_\infty^2$, then $\bar{x}$ satisfies the F. John necessary optimality conditions for program (2.3). In addition, there are two sub-cases*
  - *(iia) Either $\bar{x}$ is a Karush-Kuhn-Tucker (KKT) point of (2.3), or*
  - *(iib) $\bar{x}$ fails to be a KKT-point of (2.3). This could only happen when $g(\bar{x}) = \gamma_\infty^2$ and at the same time $0 \in \partial g(\bar{x})$.*

*Proof.* a) Let us prove statement 1). Notice that $F(\bar{x};\bar{x}) = 0$. We therefore have to show $F(x;\bar{x}) \ge 0$ for $x$ in a neighborhood of $\bar{x}$. If $x$ is feasible in (2.3), i.e., $g(x) \le \gamma_\infty^2$, then $F(x;\bar{x}) = \max\{f(x)-f(\bar{x}); g(x)-\gamma_\infty^2\} \ge f(x)-f(\bar{x}) \ge 0$ for $x$ in a neighborhood of $\bar{x}$. Here we use the fact that $\bar{x}$, being optimal, is feasible, so $[g(\bar{x}) - \gamma_\infty^2]_+ = 0$. On the other hand, when $x$ is infeasible, we find $F(x;\bar{x}) \ge g(x) - \gamma_\infty^2 > 0$. This settles statement 1).

b) To prepare the remaining statements, let us first notice that $0 \in \partial_1 F(\bar{x};\bar{x})$ is equivalent to the following condition: There exists $0 \le \bar{t} \le 1$ such that $0 = \bar{t}f'(\bar{x}) + (1-\bar{t})\phi$ for some $\phi \in \partial g(\bar{x})$, where both branches of $F(\bar{x};\bar{x})$ have to be active as soon as $0 < \bar{t} < 1$. The latter allows to distinguish the cases $g(\bar{x}) > \gamma_\infty^2$ and $g(\bar{x}) \le \gamma_\infty^2$.

c) First consider the case $g(\bar{x}) > \gamma_\infty^2$. Here the left hand branch of $F(\bar{x};\bar{x})$, being strictly negative, cannot be active, which means $\bar{t} = 0$. In consequence, $0 \in \partial g(\bar{x})$. This is the case of a critical point of constraint violation, so it proves $(i)$ in 3).

d) Next consider the case $g(\bar{x}) \leq \gamma_\infty^2$. In order to show that $\bar{x}$ satisfies the F. John necessary optimality conditions, it remains to check complementarity. If $g(\bar{x}) = \gamma_\infty^2$, there is nothing to prove, so assume $g(\bar{x}) < \gamma_\infty^2$. Then the right hand branch of $F(\bar{x};\bar{x})$ is negative, so it cannot be active, meaning that $(1-\bar{t}) = 0$. Since this is the Lagrange multiplier for the constraint, this proves the first part of statement 3 $(ii)$.

e) It remains to distinguish the two cases $(iia)$ and $(iib)$. Let us see in which cases a F. John critical point can fail to satisfy the KKT-conditions. That concerns the case where $\bar{t} = 0$, and at the same time $g(\bar{x}) \leq \gamma_\infty^2$. But $g(\bar{x}) < \gamma_\infty^2$ is impossible here, because the right hand branch of $F(\bar{x};\bar{x})$ has to be active. Then it turns out that $g(\bar{x}) = \gamma_\infty^2$ and $0 \in \partial g(\bar{x})$ is the only case where KKT fails. It may be considered as the limiting case of a critical point $\bar{x}$ of constraint violation. This settles all cases in statement 3).

f) Finally, to prove statement 2), let $\bar{x}$ satisfy the F. John necessary optimality conditions for (2.3). From b) we immediately see that it also satisfies $0 \in \partial_1 F(\bar{x};\bar{x})$. □

Remark. 1) Lemma 5.1 shows why we should search for points $\bar{x}$ satisfying $0 \in \partial_1 F(\bar{x};\bar{x})$. It also indicates that minimizing $F$ leads to so-called phase I/phase II methods (see [41, section 2.6]). Namely, as long as iterates stay infeasible, the right hand term in $F$ is dominant, so reducing $F$ reduces constraint violation. This corresponds to phase I. Once a feasible iterate has been found, phase I terminates successfully and iterates will henceforth stay feasible. This is where phase II begins and $f$ is optimized.

2) Condition $(i)$ above addresses the case where phase I fails because iterates get stuck at a limit point $\bar{x}$ with value $g(\bar{x}) > \gamma_\infty^2$, which is a local minimum (a critical point) of $g$ alone. A first-order method may get trapped at such points, and in classical mathematical programming second order techniques are used to avoid them. Here we are working with a nonsmooth program, where second order methods are difficult to come up with (see however [38], where such a method is discussed, and also [11]). Fortunately, in $H_2/H_\infty$ control, feasible iterates are usually available, so phase I can even be avoided. Notice also that case $(iib)$ may be considered the limiting case of $(i)$.

3) In [43] Sagastizábal and Solodov use a different progress function, referred to as an *improvement function*, which does not feature the penalty term $\mu[g(x) - \gamma_\infty^2]_+$. Since this term equals 0 in phase II, both criteria lead to the same steps in phase II, and differences could only occur in phase I. Now observe that with the *improvement function*, every step has to be a descent step for both the objective $f$ and the constraint $g$. In contrast, in our approach, when reducing constraint violation, a slight increase in $f$ not exceeding $\mu[g(x) - \gamma_\infty^2]_+$ is granted. This helps the algorithm in not being trapped at infeasible local minima of $f$ alone, and is therefore a possible advantage. Naturally, the difficulty of local minima of $g$ alone (local minima of constraint violation) remains with both criteria. We will come back to this issue in section 7.5, where numerical results are discussed. It turns out that a sound choice of $\mu$ is important and gives better numerical results. □

**5.2. First local model.** In this Section we introduce a local model for $F$ in a neighborhood of the current iterate $x$. Let us first introduce an approximation of $g$ in a neighborhood of $x$, by linearizing the operator $y \mapsto G(y,\omega)$ around $x$:

$$\begin{aligned} \widetilde{g}(y;x) &= \max_{\omega\in[0,\infty]} \lambda_1\left(G(x,\omega) + G'(x,\omega)(y-x)\right) \\ &= \max_{\omega\in[0,\infty]} \max_{Z\in\mathcal{C}} Z \bullet \left(G(x,\omega) + G'(x,\omega)(y-x)\right), \end{aligned} \tag{5.2}$$

where $\mathcal{C} = \{Z \in \mathbb{S}^m : Z \succeq 0, \mathrm{tr}(Z) = 1\}$, and where the derivative $G'(x,\omega)$ refers to the variable $x$. Notice that $\widetilde{g}(x;x) = g(x)$. By Taylor's theorem we expect $\widetilde{g}(y;x)$ to be a good approximation of $g(y)$ for $y$ in a neighborhood of $x$.

We now obtain an approximation of $F$ in a neighborhood of $x$ by introducing:

$$\widetilde{F}(y;x) = \max\left\{f'(x)(y-x) - \mu[g(x)-\gamma_\infty^2]_+; [\widetilde{g}(y;x)-\gamma_\infty^2] - [g(x)-\gamma_\infty^2]_+\right\}. \tag{5.3}$$

Notice that $\widetilde{F}(x;x) = F(x;x)$, and that $\widetilde{F}(y;x)$ is close to $F(y;x)$ for $y$ close to $x$. The following result renders these statements exact:

LEMMA 5.2. *Let $B \subset \mathbb{R}^n$ be a bounded set. Then there exists $L > 0$ such that for all $x,\ y \in B$:*

$$|g(y) - \widetilde{g}(y;x)| \le L\|y-x\|^2 \quad \textit{and} \quad |F(y;x) - \widetilde{F}(y;x)| \le L\|y-x\|^2.$$

*Proof.* By Weyl's theorem we have $\lambda_m(E) \le \lambda_1(A+E) - \lambda_1(A) \le \lambda_1(E)$, for all matrices $A, E \in \mathbb{S}^m$. We apply this to $A = G(y,\omega)$ and $A+E = G(x,\omega)+G'(x,\omega)(y-x)$. Then $E = \mathcal{O}(\|y-x\|^2)$, uniformly over $x, y \in B$ and uniformly over $\omega \in [0,\infty]$, which is a compact set. Here we use the fact that the operators $G(\cdot,\omega)$ are of class $C^2$ in $x$ and jointly continuous in $(x,\omega)$. More precisely

$$\sup_{\omega\in[0,\infty]} \sup_{z\in\mathrm{co}(B)} \|G''(z,\omega)\| < \infty.$$

This proves $|g(y) - \widetilde{g}(y;x)| \le L_1\|y-x\|^2$ for some $L_1 > 0$ and all $x, y \in B$.

Moreover, $f$ is of class $C^2$, so that by Taylor's formula there exists $L_2 > 0$ such that $|f(y) - f(x) - f'(x)(y-x)| \le L_2\|y-x\|^2$ uniformly over $x, y \in B$. With $L = \max\{L_1, L_2\}$ we obtain

$$\begin{array}{rcl} |F(y;x) - \widetilde{F}(y;x)| & \le & \max\{|f(y)-f(x)-f'(x)(y-x)|; |g(y)-\widetilde{g}(y;x)|\} \\ & \le & L\|y-x\|^2. \end{array}$$

□

It is convenient to represent the local model (5.2) differently. Let us introduce

$$\alpha(\omega,Z) = [Z \bullet G(x,\omega) - \gamma_\infty^2] - [g(x)-\gamma_\infty^2]_+ \in \mathbb{R}, \qquad \phi(\omega,Z) = G'(x,\omega)^\star Z \in \mathbb{R}^n,$$

where dependence on the point $x$ is suppressed for convenience. Then the right hand branch of $\widetilde{F}(y;x)$ may be written as the envelope of cutting planes

$$[\widetilde{g}(y;x)-\gamma_\infty^2] - [g(x)-\gamma_\infty^2]_+ = \sup_{\omega\in[0,\infty]} \sup_{Z\in\mathcal{C}} \alpha(\omega,Z) + \phi(\omega,Z)^T(y-x).$$

Adding the left hand branch of $\widetilde{F}(y;x)$ by introducing

$$\alpha_0 = -\mu[g(x)-\gamma_\infty^2]_+, \qquad \phi_0 = f'(x),$$

we can introduce

$$\mathcal{G} = \mathrm{co}\left(\{(\alpha(\omega,Z),\phi(\omega,Z)) : \omega \in [0,\infty], Z \in \mathcal{C}\} \cup \{(\alpha_0,\phi_0)\}\right).$$

Then the local model $\widetilde{F}(y;x)$ may be written as

$$\widetilde{F}(y;x) = \max\{\alpha + \phi^T(y-x) : (\alpha,\phi) \in \mathcal{G}\}. \tag{5.4}$$

The advantage of (5.4) over (5.3) is that elements $(\alpha,\phi)$ of $\mathcal{G}$ are easier to store than elements $(\omega,Z) \in [0,\infty] \times \mathcal{C}$. Also, as we shall see, it is more convenient to construct approximations $\mathcal{G}_k$ of $\mathcal{G}$. This is addressed in the next section.

**5.3. Second local model and tangent program.** Suppose $x$ is the current iterate of our algorithm to be designed. In order to generate trial steps away from $x$, we will recursively generate approximations $\widetilde{F}_k(y;x)$ of $\widetilde{F}(y;x)$, referred to as the working models. Using (5.4), these will be of the form

$$\widetilde{F}_k(y;x) = \max\{\alpha + \phi^\top(y-x) : (\alpha,\phi) \in \mathcal{G}_k\}, \tag{5.5}$$

where $\mathcal{G}_k \subset \mathcal{G}$. In particular, $\widetilde{F}_k(y;x) \leq \widetilde{F}(y;x)$, with exactness $\widetilde{F}_k(x;x) = \widetilde{F}(x;x) = F(x;x) = 0$ at $y = x$. Moreover, our construction presented below assures that $\partial_1\widetilde{F}_k(x;x) \subset \partial_1 F(x;x)$ for all $k$ and that the $\widetilde{F}_k$ get closer to $\widetilde{F}$ as $k$ increases. In tandem with the proximity control management described in Section 6, this will also assure that the $\widetilde{F}_k$ get closer to the true $F$. Once the set $\mathcal{G}_k$ is formed, a new trial step $y^{k+1}$ is computed via the tangent program:

$$\min_{y\in\mathbb{R}^n} \widetilde{F}_k(y;x) + \tfrac{\delta_k}{2}\|y-x\|^2. \tag{5.6}$$

Here $\delta_k > 0$ is the so-called proximity control parameter, which is specified anew at each step. How this should be organized will be explained in section 6.

Notice that by convexity $y^{k+1}$ is a solution of (5.6) as soon as

$$0 \in \partial_1\widetilde{F}_k(y^{k+1};x) + \delta_k(y^{k+1}-x). \tag{5.7}$$

The first question is what happens if the solution of the program (5.6) is $y^{k+1} = x$?

LEMMA 5.3. *Suppose $y^{k+1} = x$ is solution of the tangent program (5.6). Then $0 \in \partial_1 F(x;x)$.*

This is indeed clear in view of (5.7), because we get $0 \in \partial_1\widetilde{F}_k(x;x)$, which implies $0 \in \partial_1 F(x;x)$ by the property $\partial_1\widetilde{F}_k(x;x) \subset \partial_1 F(x;x)$ of a working model. The conclusion is that as soon as $0 \notin \partial_1 F(x;x)$, then $0 \notin \partial_1\widetilde{F}_k(x;x)$, and the trial step $y^{k+1}$ will always offer something new. In particular, if $0 \notin \partial_1 F(x,x)$, then we know for sure that $\widetilde{F}_k(y^{k+1};x) < \widetilde{F}(x;x) = 0$, so that there is always a progress predicted by $\widetilde{F}_k$.

REMARK. In the light of Lemma 5.3 it may seem natural to confine the test $0 \in \partial_1\widetilde{F}(x;x)$ (step 2 of the algorithm) to the first instance of the tangent program $k = 1$. Indeed, if $0 \in \partial_1\widetilde{F}_1(x;x)$, then the first tangent program will detect this and return $y^2 = x$, in which case we quit. However, notice that this does not work the other way round. If $0 \in \partial_1 F(x;x)$, then the tangent program based on $\widetilde{F}_k$ may still find $y^{k+1} \neq x$, in which case we would not necessarily stop the inner loop. Only when $\partial_1\widetilde{F}_k(x;x) = \partial_1 F(x;x)$ are we certain that $y^{k+1} = x$. In other words, if we wish to confine the test in step 2 of the algorithm to the first instance of the tangent program in step 4, we have to use the full subdifferential $\partial_1\widetilde{F}_1(x;x) = \partial_1 F(x;x)$. As soon as $y^2 \neq x$, then the inner loop is entered, and this condition is no longer required for the following $\widetilde{F}_k$. In any case, letting $\partial_1\widetilde{F}_k(x;x) = \partial_1 F(x;x)$ does not pose a numerical problem if $\partial_1 F(x;x)$ is not exceedingly large. □

From now on we assume $0 \notin \partial_1 F(x,x)$. The solution $y^{k+1}$ of (5.6) is then predicting a decrease of the value of the progress function (5.1) at $y^{k+1}$. This gives $y^{k+1}$ the option to improve over the current iterate $x$ and become the new iterate $x^+$. For this to happen, we have to make sure that $\widetilde{F}_k$ is a good model of $F$ in the neighborhood of $x$.

According to standard terminology, when $y^{k+1}$ is accepted as the new iterate $x^+$, it is called a serious step, while trial points $y^{k+1}$ which are rejected are called null

steps. If $y^{k+1}$ is a null step and has to be rejected, we use it to improve the model $\mathcal{G}_{k+1}$ at the next sweep.

Let us now show in detail how to construct the sets $\mathcal{G}_k$. We choose them of the form

$$\mathcal{G}_k = \mathrm{co}\,(\mathcal{G}_0 \cup \mathcal{G}_k^c \cup \mathcal{G}_k^*)\,, \quad k = 1, 2, \ldots, \tag{5.8}$$

where we refer to $\mathcal{G}_0$ as the subgradient elements, to $\mathcal{G}_k^c$ as the cutting planes, and to $\mathcal{G}_k^*$ as the aggregate element. The first property concerns $\mathcal{G}_0$, which is held fixed during the iteration $k$.

LEMMA 5.4. *Let $\omega_0 \in \Omega(x)$ be any of the active frequencies at $x$. Choose a normalized eigenvector $e_0$ associated with the maximum eigenvalue $g(x) = \lambda_1(G(x,\omega_0))$ of $G(x,\omega_0)$, and let $Z_0 := e_0 e_0^T \in \mathcal{C}$. If we let $(\alpha_0, \phi_0) \in \mathcal{G}_0$ and $(\alpha(\omega_0, Z_0), \phi(\omega_0, Z_0)) \in \mathcal{G}_0$, and if $\mathcal{G}_0 \subset \mathcal{G}_k$, then we have $\widetilde{F}_k(x;x) = F(x;x) = 0$ at all times $k$.*

In practice it is useful to enrich the set $\mathcal{G}_0$ so that it contains the subdifferential $\partial_1 F(x;x)$ at $x$. This can be arranged in those cases where $\Omega(x)$, the set of active frequencies, is finite. For every $\omega \in \Omega(x)$ let $r_\omega \geq 1$ be the eigenvalue multiplicity of $\lambda_1\,(G(x,\omega))$. Let the $r_\omega$ columns of $Q_\omega$ be an orthonormal basis of the maximum eigenspace of $G(x,\omega)$. Then put

$$\begin{aligned}\mathcal{G}_0 = \mathrm{co}\Big( \big\{(\alpha(\omega, Z_\omega), \phi(\omega, Z_\omega)) : \omega \in \Omega(x), Z_\omega = Q_\omega^T Y_\omega Q_\omega,\\ Y_\omega \in \mathbb{S}^{r_\omega}, Y_\omega \succeq 0, \mathrm{Tr}(Y_\omega) = 1\big\} \cup \{(\alpha_0, \phi_0)\} \Big).\end{aligned} \tag{5.9}$$

We observe that this set is not finitely generated, but can be handled as a semidefinite programming constraint via the matrices $Y_\omega$. However, for our convergence proof it would be sufficient to keep just the one element required by Lemma 5.4 in $\mathcal{G}_0$.

Let us now look at the cutting plane sets $\mathcal{G}_k^c$. Here we use a recursive construction. Suppose the solution $y^{k+1}$ of tangent program (5.6) based on the latest model $\mathcal{G}_k$ is a null step. Then we need to improve the next model $\mathcal{G}_{k+1}$, and this is done by including a cutting plane in the new set $\mathcal{G}_{k+1}^c$, which cuts away the unsuccessful trial step $y^{k+1}$.

LEMMA 5.5. *Let $y^{k+1}$ be the solution of tangent program (5.6) at stage $k$ and suppose $y^{k+1}$ is a null step. Suppose the right hand branch of (5.3) is active at $y^{k+1}$, and let $\omega_{k+1} \in [0,\infty]$ and $Z_{k+1} \in \mathcal{C}$ be one of the pairs where the maximum (5.2) is attained, that is, $\widetilde{g}(y^{k+1};x) - \gamma_\infty^2 - [g(x) - \gamma_\infty^2]_+ = \alpha(\omega_{k+1}, Z_{k+1}) + \phi(\omega_{k+1}, Z_{k+1})^T(y^{k+1} - x)$. If we keep $(\alpha(\omega_{k+1}, Z_{k+1}), \phi(\omega_{k+1}, Z_{k+1})) \in \mathcal{G}_{k+1}^c$ then*

$$\widetilde{F}_{k+1}(y^{k+1};x) = \widetilde{F}(y^{k+1};x).$$

REMARK. 1) Following standard terminology, we refer to this procedure as the cutting plane element. In fact, adding $\omega_{k+1}$ and $Z_{k+1}$ to the approximations at the next step $k+1$ will cut away the unsuccessful null step $y^{k+1}$, paving the way for a better $y^{k+2}$ at the next sweep.
2) If the right hand branch in (5.3) is not active, it suffices to have the pair $(\alpha_0, \phi_0) \in \mathcal{G}_{k+1}$. As we keep this in $\mathcal{G}_0$ anyway, no action on cutting planes is required in this event, i.e. we may have $\mathcal{G}_{k+1}^c = \emptyset$. □

In practice it will be useful to enrich the set $\mathcal{G}_{k+1}^c$ by what we call anticipating cutting planes. Let us again consider the case of a finite set $\Omega(x)$. We select a finite

extension $\Omega_e(x)$ of $\Omega(x)$ along the lines described in [5]. We let

$$\mathcal{G}^c_{k+1} = \mathrm{co}\Big(\{(\alpha(\omega_{k+1}, Z_{k+1}), \phi(\omega_{k+1}, Z_{k+1}))\} \cup \tag{5.10}$$

$$\big\{(\alpha(\omega, Z_\omega), \phi(\omega, Z_\omega)) : \omega \in \Omega_e(x) \setminus \Omega(x), Z_\omega = Q_\omega^T Y_\omega Q_\omega, Y_\omega \succeq 0, \mathrm{Tr}(Y_\omega) = 1\big\}\Big),$$

where the columns of $Q_\omega$ are an orthonormal basis of some invariant subspace of $\lambda_1\left(G(x,\omega)\right)$. Notice that for $\omega \in \Omega_e(x) \setminus \Omega(x)$, the support planes belonging to $(\alpha(\omega, Z_\omega), \phi(\omega, Z_\omega))$ are indeed different in nature from those retained in $\mathcal{G}_0$, because they will not be exact at $y = x$. We may have $\alpha(\omega, Z_\omega) < 0$, so these planes resemble cutting planes, which are exact at the null steps $y^{k+1}$.

Notice that convergence theory requires only $(\alpha(\omega_{k+1}, Z_{k+1}), \phi(\omega_{k+1}, Z_{k+1})) \in \mathcal{G}^c_{k+1}$ for the element of Lemma 5.5.

Remark. Notice that the planes in $\mathcal{G}_0$ are exact at $x$, while genuine cutting planes are exact at the null steps $y^{k+1}$. Anticipated cutting planes need not be exact anywhere, but we have observed that they often behave similar to true cutting planes and can help to avoid a large number of unsuccessful null steps. □

We need yet another process to improve the model $\mathcal{G}_{k+1}$, which in the nonsmooth terminology is referred to as aggregation, and which is needed in order to avoid storing an increasing number of cutting planes. Suppose that the solution $y^{k+1}$ of the old tangent program (5.6) based on $\mathcal{G}_k$ is a null step. By the optimality condition we have $0 \in \partial_1 \widetilde{F}_k(y^{k+1}; x) + \delta_k(y^{k+1} - x)$. Using the representation (5.4) and the form (5.8), we find $(\alpha_0, \phi_0) \in \mathcal{G}_0$, $(\alpha_{k+1}, \phi_{k+1}) \in \mathcal{G}^c_k$ and $(\alpha^*_k, \phi^*_k) \in \mathcal{G}^*_k$ together with convex coefficients $\tau_0 \geq 0, \tau_{k+1} \geq 0, \tau^*_k \geq 0$, $\tau_0 + \tau_{k+1} + \tau^*_k = 1$, such that

$$0 = \tau_0\phi_0 + \tau_{k+1}\phi_{k+1} + \tau^*_k\phi^*_k + \delta_k(y^{k+1} - x).$$

We put $\alpha^*_{k+1} = \tau_0\alpha_0 + \tau_{k+1}\alpha_{k+1} + \tau^*_k\alpha^*_k \in \mathbb{R}$, $\phi^*_{k+1} = \tau_0\phi_0 + \tau_{k+1}\phi_{k+1} + \tau^*_k\phi^*_k \in \mathbb{R}^n$ and keep $(\alpha^*_{k+1}, \phi^*_{k+1}) \in \mathcal{G}^*_{k+1}$, calling it the aggregate element. Notice that we have $(\alpha^*_{k+1}, \phi^*_{k+1}) \in \mathcal{G}$ by convexity. Altogether, this shows

$$0 = \phi^*_{k+1} + \delta_k(y^{k+1} - x). \tag{5.11}$$

Lemma 5.6. *Keeping the aggregate element $(\alpha^*_{k+1}, \phi^*_{k+1})$ in the new $\mathcal{G}^*_{k+1}$ assures $\widetilde{F}_{k+1}(y^{k+1}; x) \geq \widetilde{F}_k(y^{k+1}; x)$, and that (5.11) is satisfied.*

To conclude this section, let us outline how the tangent program based on the form (5.4) and (5.8) is solved. Notice first that elements of $\mathcal{G}_0 \cup \mathcal{G}^c_k$ have the same form $(\alpha(\omega, Z_\omega), \phi(\omega, Z_\omega))$, where $\omega \in \Omega_e(x)$ for some finite extension of $\Omega(x)$, and $Z_\omega = Q_\omega^T Y_\omega Q_\omega$ for some $Y_\omega \succeq 0$, $\mathrm{Tr}(Y_\omega) = 1$. To this we add the aggregate element $(\alpha^*_k, \phi^*_k)$, and the element $(\alpha_0, \phi_0)$ coming from the left hand branch of $\widetilde{F}$. This means, after relabelling the finite set $\Omega_e(x)$ as $\{\omega_1, \ldots, \omega_p\}$, we can write (5.6) in the form

$$\min_{y\in\mathbb{R}^n} \max\Big\{\alpha_0 + \phi_0^T(y - x); \max_{r=1,\ldots,p} \max_{Y_r\succeq 0, \mathrm{Tr}(Y_r)=1} \alpha_r(Y_r) + \phi_r(Y_r)^T(y - x);$$
$$\alpha_{p+1} + \phi_{p+1}^T(y - x)\Big\} + \frac{\delta_k}{2}\|y - x\|^2,$$

where $\alpha_r(Y_r) = \alpha(\omega_r, Z_{\omega_r})$, etc., and where the aggregate element $(\alpha^*_k, \phi^*_k)$ is relabelled $(\alpha_{p+1}, \phi_{p+1})$. Replacing the maximum over the three branches by a maximum

over the convex hull of the three does not change the value of this program. Using Fenchel duality, we may then swap the min and max operators. The then inner minimum can be computed explicitly, which leads to the expression

$$y^{k+1} = x - \frac{1}{\delta_k}\left(\tau_0\phi_0 + \sum_{r=1}^{p}\tau_r\phi_r(Y_r) + \tau_{p+1}\phi_{p+1}\right),$$

where $(\tau, Y)$ is the dual variable. Substituting this back into the dual program, using linearity of $\phi(Y)$ in $Y$, and re-writing $\tau_r Y_r$ as a new matrix $Y_r$ with $\mathrm{Tr}(Y_r) = \tau_r$, leads to the dual program

$$\begin{array}{ll}\text{maximize} & \tau_0\alpha_0 + \displaystyle\sum_{r=1}^{p}\alpha_r(Y_r) + \tau_{p+1}\alpha_{p+1} - \frac{1}{2\delta_k}\left\|\tau_0\phi_0 + \sum_{r=1}^{p}\phi_r(Y_r) + \tau_{p+1}\phi_{p+1}\right\|^2 \\ \text{subject to} & \tau_0 \geq 0, \tau_{p+1} \geq 0, Y_r \succeq 0 \text{ and } \tau_0 + \displaystyle\sum_{r=0}^{p+1}\mathrm{Tr}(Y_r) + \tau_{p+1} = 1\end{array}$$

which we recognize as the concave form of a semidefinite program (SDP), as soon as we write $\phi_r(Y_r)$ in its original form $G'(x,\omega_r)^\star Q_{\omega_r}^T Y_r Q_{\omega_r}$. The return formula becomes

$$y^{k+1} = x - \frac{1}{\delta_k}\left(\tau_0^*\phi_0 + \sum_{r=1}^{p}\phi_r(Y_r^*) + \tau_{p+1}^*\phi_{p+1}\right), \tag{5.12}$$

where the dual optimal solution is $(\tau_0^*, Y_1^*, \ldots, Y_p^*, \tau_{p+1}^*)$. Notice that this SDP is usually of small size, so that solving a succession of these programs seems a satisfactory strategy.

To conclude, we consider the case of particular interest, where the eigenvalue multiplicity of all matrices involved is 1, or where we decide to keep only one eigenvector for each leading eigenvalue. If $\lambda_1(G(x,\omega))$ has eigenvalue multiplicity $r_\omega = 1$, the matrices $Q_\omega$ are just column vectors $e_\omega$, where $e_\omega$ is the normalized eigenvector associated with $\lambda_1(G(x,\omega))$ and $Y_\omega = 1$. Similarly, for the latest cutting plane we then have $Q_\omega = e_\omega$ for the normalized eigenvector of $\lambda_1\left(G(x,\omega) + G'(x,\omega)(y^{k+1} - x)\right)$. In this case the sets $\mathcal{G}_0$, $\mathcal{G}_k^c$ are finite, and so $\mathcal{G}_k$ itself is a polyhedron $\mathrm{co}\{(\alpha_0,\phi_0),\ldots,(\alpha_{p+1},\phi_{p+1})\}$ where $\mathrm{card}(\Omega_e(x)) = p$. In this case the dual program is a convex quadratic program which can be solved very efficiently:

$$\begin{array}{ll}\text{maximize} & \displaystyle\sum_{r=0}^{p+1}\tau_r\alpha_r - \frac{1}{2\delta_k}\left\|\sum_{r=0}^{p+1}\tau_r\phi_r\right\|^2 \\ \text{subject to} & \tau_r \geq 0, r = 0,\ldots,p+1 \text{ and } \displaystyle\sum_{r=0}^{p+1}\tau_r = 1\end{array}$$

with dual optimal solution $\tau^*$, and the return formula is $y^{k+1} = x - \dfrac{1}{\delta_k}\displaystyle\sum_{r=0}^{p+1}\tau_r^*\phi_r$.

**5.4. The algorithm.** In this section we present the nonsmooth spectral bundle algorithm for program (2.3).

**Algorithm 1.** Proximity control algorithm for the $H_2/H_\infty$ program (2.3)

**Parameters:** $0 < \gamma < \widetilde{\gamma} < \Gamma < 1$.

1: **Initialize outer loop**. Find initial $x^1$ such that $f(x^1) < \infty$ and $g(x^1) < \infty$. Put outer loop counter $j = 1$.

2: **Outer loop**. At outer loop counter $j$, stop at the current iterate $x^j$ if $0 \in \partial_1 F(x^j; x^j)$. Otherwise compute $\Omega(x^j)$ and continue with inner loop.

3: **Initialize inner loop**. Choose approximation $\mathcal{G}_1$ of $\mathcal{G}$ as in (5.8), where $\mathcal{G}_0$ contains $(\alpha(\omega_0, Z_0), \phi(\omega_0, Z_0))$ for some fixed $\omega_0 \in \Omega(x^j)$ and $Z_0 = e_0 e_0^T$, where $e_0$ is a normalized eigenvector associated with $\lambda_1(G(x^j, \omega_0))$. Possibly enrich $\mathcal{G}_0$ as in (5.9). Initialize $\mathcal{G}_1^c = \emptyset$, $\mathcal{G}_1^* = \emptyset$, but possibly enrich using anticipated cutting planes (5.10). Initialize proximity parameter $\delta_1 > 0$. If memory element for $\delta$ is available, use it to initialize $\delta_1$. Put inner loop counter $k = 1$.

4: **Trial step**. At inner loop counter $k$ for given approximation $\mathcal{G}_k$ and proximity control parameter $\delta_k > 0$, solve tangent program:

$$\min_{y \in \mathbb{R}^n} \widetilde{F}_k(y; x^j) + \frac{\delta_k}{2}\|y - x^j\|^2,$$

whose solution is $y^{k+1}$.

5: **Test of progress**. Check whether

$$\rho_k = \frac{F(y^{k+1}; x^j)}{\widetilde{F}_k(y^{k+1}; x^j)} \geq \gamma.$$

If this is the case, accept trial step $y^{k+1}$ as the new iterate $x^{j+1}$ (serious step). Compute new memory element $\delta^+$ as:

$$\delta^+ = \begin{cases} \dfrac{\delta_k}{2} & \text{if } \rho_k > \Gamma \\ \\ \delta_k & \text{otherwise} \end{cases}$$

Increase outer loop counter $j \to j + 1$, and go back to step 2. If $\rho_k < \gamma$ continue inner loop with step 6 (null step).

6: **Cutting plane**. Select a frequency $\omega_{k+1}$ where $\widetilde{g}(y^{k+1}, x^j)$ is active and pick a normalized eigenvector $e_{k+1}$ associated with the maximum eigenvalue of $G(x^j, \omega_{k+1}) + G'(x^j, \omega_{k+1})(y^{k+1} - x^j)$. Put $Z_{k+1} = e_{k+1} e_{k+1}^T$ and assure $(\alpha(\omega_{k+1}, Z_{k+1}), \phi(\omega_{k+1}, Z_{k+1})) \in \mathcal{G}_{k+1}^c$. Possibly enrich $\mathcal{G}_{k+1}^c$ by anticipating cutting planes as in (5.10).

7: **Aggregation**. Keep aggregate pair $(\alpha_{k+1}^*, \phi_{k+1}^*)$ as in (5.11) in $\mathcal{G}_{k+1}^*$.

8: **Proximity control**. Compute control parameter

$$\widetilde{\rho}_k = \frac{\widetilde{F}(y^{k+1}; x^j)}{\widetilde{F}_k(y^{k+1}; x^j)}.$$

Update proximity parameter $\delta_k$ as

$$\delta_{k+1} = \begin{cases} \delta_k, & \text{if } \rho_k < \gamma \text{ and } \widetilde{\rho}_k < \widetilde{\gamma} \\ \\ 2\delta_k & \text{if } \rho_k < \gamma \text{ and } \widetilde{\rho}_k \geq \widetilde{\gamma} \end{cases}$$

Increase inner loop counter $k$ and go back to step 4.

**6. Management of the proximity parameter.** In this Section the convergence proof of algorithm 1 will be given.

To begin with, let us explain the management of the proximity control parameter in steps 5 and 8. Notice that there are two control mechanisms, governed by the control parameters $\rho_k$ and $\widetilde{\rho}_k$. In step 5, test parameter $\rho_k$ compares the current model $\widetilde{F}_k$ to the truth $F$. The ideal case would be $\rho_k \approx 1$, but we accept $y^{k+1} = x^{j+1}$

much earlier, namely if $\rho_k \geq \gamma$, where the reader might for instance imagine $\gamma = \frac{1}{4}$. Let us call $y^{k+1}$ bad if $\rho_k < \gamma$. So null steps are bad, while serious steps are not bad. Imagine further that $\Gamma = \frac{3}{4}$, then steps $y^{k+1}$ with $\rho_k > \Gamma$ are good steps. In the good case the model $\widetilde{F}_k$ seems very reliable, so we can relax proximity control a bit at the next outer step. This is arranged by memorizing $\delta^+ = \delta_k/2$ in step 5 of the algorithm.

It is more intriguing to decide what we should do when $\rho_k < \gamma$, meaning that $y^{k+1}$ is bad (a null step). Here we need the second control parameter $\widetilde{\rho}_k$ in step 8 to support our decision. Adopting the same terminology, we say that the agreement between $\widetilde{F}$ and $\widetilde{F}_k$ is bad if $\widetilde{\rho}_k < \widetilde{\gamma}$. If this is the case, we keep $\delta_{k+1} = \delta_k$ unchanged, being reluctant to increase the $\delta$-parameter prematurely, and continue to rely on cutting planes and aggregation, hoping that this will drive $\widetilde{F}_k$ closer to $\widetilde{F}$ (and also to $F$) and bring home the bacon in the end. On the other hand, if $\widetilde{\rho}_k \geq \widetilde{\gamma}$, then we have to accept that driving $\widetilde{F}_k$ closer to $\widetilde{F}$ alone will not do the job, simply because $\widetilde{F}$ itself is too far from the true $F$. Here we need to tighten proximity control, by increasing $\delta_{k+1} = 2\delta_k$ at the next sweep. This is done in step 8 and brings $\widetilde{F}$ closer to $F$.

REMARK. Notice that the control parameters $\rho_k$ and $\widetilde{\rho}_k$ in steps 5 and 8 are well defined because we only enter the inner loop when $0 \notin \partial_1 F(x;x)$, in which case we have $\widetilde{F}_k(y^{k+1};x) < \widetilde{F}_k(x;x) = 0$. □

**6.1. Finiteness of inner loop.** Let $x$ be the current iterate of the outer loop. We start our convergence analysis by showing that the inner loop terminates after a finite number of updates $k$ with a serious step $y^{k+1} = x^+$. This will be proved in the next three Lemmas.

Recall that $y^{k+1}$ is solution of the tangent program (5.6) and may be obtained from the dual optimal solution by the return formula (5.12), which is of the form

$$y^{k+1} = x - \frac{1}{\delta_k}\left[\tau_0 f'(x) + \sum_{\omega\in\Omega_e(x)} \tau_\omega G'(x,\omega)^\star Z_\omega\right]$$

for a finite extension $\Omega_e(x)$ of $\Omega(x)$ and for certain $Z_\omega \in \mathcal{C}$. Since the sequence $\delta_k$ in the inner loop is nondecreasing, we have the following

LEMMA 6.1. *The solutions $y^{k+1}$ of (5.6) satisfy*

$$\|y^{k+1}\| \leq \|x\| + \delta_1^{-1}\left(\|f'(x)\| + \max_{\omega\in[0,\infty]} \|G'(x,\omega)^\star\|\right) < \infty. \tag{6.1}$$

We are now ready to prove finite termination of the inner loop. Our first step is the following

LEMMA 6.2. *Suppose the inner loop turns forever and creates an infinite sequence $y^{k+1}$ of null steps with $\rho_k < \gamma$. Then there must be an instant $k_0$ such that the control parameter $\widetilde{\rho}_k$ satisfies $\widetilde{\rho}_k < \widetilde{\gamma}$ for all $k \geq k_0$.*

*Proof.* Indeed, by assumption none of the trial steps $y^{k+1}$ passes the acceptance test in step 5, so $\rho_k < \gamma$ at all times $k$. Suppose now that $\widetilde{\rho}_k \geq \widetilde{\gamma}$ an infinity of times $k$. Then according to step 8 the proximity control parameter $\delta_k$ is increased infinitely often, meaning $\delta_k \to \infty$.

Using the fact that $y^{k+1}$ is the optimal solution of the tangent program (5.6) gives $0 \in \partial_1 \widetilde{F}_k(y^{k+1};x) + \delta_k(y^{k+1} - x)$. Using convexity of $\widetilde{F}_k(\cdot;x)$, we deduce that

$$-\delta_k(y^{k+1} - x)^T(x - y^{k+1}) \leq \widetilde{F}_k(x;x) - \widetilde{F}_k(y^{k+1};x)$$

Using $\widetilde{F}_k(x;x) = F(x;x) = 0$, assured by keeping $(\alpha(\omega_0, Z_0), \phi(\omega_0, Z_0)) \in \mathcal{G}_0 \subset \mathcal{G}_k$ at all times (Lemma 5.4), we obtain

$$\frac{\delta_k \|y^{k+1} - x\|^2}{-\widetilde{F}_k(y^{k+1}; x)} \leq 1. \tag{6.2}$$

Next, applying Lemma 5.2 to the bounded set $B = \{y^{k+1} : k \in \mathbb{N}\} \cup \{x\}$ gives

$$\left| F(y^{k+1}; x) - \widetilde{F}(y^{k+1}; x) \right| \leq L \|y^{k+1} - x\|^2 \tag{6.3}$$

for some $L > 0$ and every $k \in \mathbb{N}$. Now we expand the control parameters $\rho_k$ and $\widetilde{\rho}_k$ as follows:

$$\begin{aligned} \widetilde{\rho}_k &= \rho_k + \frac{F(y^{k+1}; x) - \widetilde{F}(y^{k+1}; x)}{-\widetilde{F}_k(y^{k+1}; x)} \\ &\leq \rho_k + \frac{L\|y^{k+1} - x\|^2}{-\widetilde{F}_k(y^{k+1}; x)} \leq \rho_k + \frac{L}{\delta_k} \qquad \text{(using (6.3) and then (6.2))} \end{aligned}$$

Since $L/\delta_k \to 0$, we deduce $\limsup \widetilde{\rho}_k \leq \limsup \rho_k \leq \gamma < \widetilde{\gamma}$, which contradicts $\widetilde{\rho}_k > \widetilde{\gamma}$ for infinitely many $k$. □

So far we know that if the inner loop turns forever, this implies $\rho_k < \gamma$ and $\widetilde{\rho}_k < \widetilde{\gamma}$ from some counter $k_0$ onwards. Our next Lemma shows that this cannot happen. We refer the interested reader to [18, Proposition 4.3], where essentially the same result is proved. For the sake of completeness and the coherence of notation we give our own proof below.

LEMMA 6.3. *Suppose the inner loop turns forever and produces iterates $y^{k+1}$ with $\rho_k < \gamma$ and $\widetilde{\rho}_k < \widetilde{\gamma}$ for all $k \geq k_0$. Then $0 \in \partial_1 F(x;x)$.*

*Proof.* 1) Step 8 of the algorithm tells us that from counter $k_0$ onwards we are in the case where the proximity parameter is no longer increased. We may therefore assume that it remains unchanged for $k \geq k_0$, that is, $\delta := \delta_k$ for all $k \geq k_0$.

2) For later use, let us introduce the function

$$\psi_k(y;x) = \widetilde{F}_k(y;x) + \tfrac{\delta}{2}\|y - x\|^2.$$

As we have seen already, the necessary optimality condition for the tangent program implies

$$\delta\|y^{k+1} - x\|^2 \leq F(x;x) - \widetilde{F}_k(y^{k+1};x) = -\widetilde{F}_k(y^{k+1};x).$$

Now remember that in step 7 of the algorithm we keep the aggregate $(\alpha^*_{k+1}, \phi^*_{k+1}) \in \mathcal{G}_{k+1}$. Let us define the function

$$\psi^*_k(y;x) = \alpha^*_{k+1} + \phi^{*T}_{k+1}(y - x) + \tfrac{\delta}{2}\|y - x\|^2.$$

We claim that

$$\psi^*_k(y^{k+1};x) = \psi_k(y^{k+1};x) \text{ and } \psi^*_k(y;x) \leq \psi_{k+1}(y;x). \tag{6.4}$$

Indeed, the inequality on the right is clear because $(\alpha^*_{k+1}, \phi^*_{k+1})$ is retained in $\mathcal{G}_{k+1}$ and therefore contributes to the supremum building $\psi_{k+1}$. As for the equality on the left, observe that the aggregate subgradient $\phi^*_k$ is the one which realizes the necessary

optimality condition for tangent program (5.6) at stage $k$. Now $\psi_k(\cdot;x)$ is just the objective of this program, so the function $\psi_k^*(\cdot;x)$ must be exact at $y^{k+1}$.

We now prove the relationship

$$\psi_k^*(y;x) = \psi_k^*(y^{k+1};x) + \tfrac{\delta}{2}\|y-y^{k+1}\|^2. \tag{6.5}$$

Indeed, notice that $\psi_k^*$ is a quadratic function, so expanding it gives

$$\begin{aligned}\psi_k^*(y;x) = \psi_k^*(y^{k+1};x) + \nabla\psi_k^*(y^{k+1};x)^T(y-y^{k+1})\\ +\tfrac{1}{2}(y-y^{k+1})^T\nabla^2\psi_k^*(y^{k+1};x)(y-y^{k+1}).\end{aligned}$$

But $\nabla^2\psi_k^*(y^{k+1};x) = \delta I$, so in order to establish (6.5), we have but to show that $\nabla\psi_k^*(y^{k+1};x) = 0$. To prove this observe that

$$\begin{aligned}\nabla\psi_k^*(y^{k+1};x) &= \phi_{k+1}^* + \delta(y^{k+1}-x)\\ &= -\delta(y^{k+1}-x) + \delta(y^{k+1}-x) = 0 \qquad \text{(using (5.11))}\end{aligned}$$

so (6.5) is proved. Using this and the previous relations gives

$$\begin{aligned}\psi_k(y^{k+1};x) &\le \psi_k^*(y^{k+1};x) + \tfrac{\delta}{2}\|y^{k+2}-y^{k+1}\|^2 && \text{(using (6.4) left)}\\ &= \psi_k^*(y^{k+2};x) && \text{(using (6.5))}\\ &\le \psi_{k+1}(y^{k+2};x) && \text{(using (6.4) right)}\\ &\le \psi_{k+1}(x;x) && (y^{k+2}\text{ is minimizer of }\psi_{k+1})\\ &= \widetilde{F}_k(x;x) = F(x;x) = 0.\end{aligned}$$

This proves that the sequence $\psi_k(y^{k+1};x)$ is monotonically increasing and bounded above, so it converges to some limit $\psi^* \le F(x;x) = 0$. Since the term $\frac{\delta}{2}\|y^{k+2}-y^{k+1}\|^2$ is squeezed in between two terms with the same limit $\psi^*$, we deduce:

$$\tfrac{\delta}{2}\|y^{k+2}-y^{k+1}\|^2 \to 0.$$

Since the sequence $y^{k+1}$ is bounded by Lemma 6.1, we deduce using a geometric argument that:

$$\|y^{k+2}-x\|^2 - \|y^{k+1}-x\|^2 \to 0. \tag{6.6}$$

Recalling the relation $\widetilde{F}_k(y;x) = \psi_k(y;x) - \frac{\delta}{2}\|y-x\|^2$, we finally obtain

$$\begin{aligned}&\widetilde{F}_{k+1}(y^{k+2};x) - \widetilde{F}_k(y^{k+1};x) \qquad (6.7)\\ &\quad = \psi_{k+1}(y^{k+2};x) - \psi_k(y^{k+1};x) - \tfrac{\delta}{2}\|y^{k+2}-x\|^2 + \tfrac{\delta}{2}\|y^{k+1}-x\|^2\end{aligned}$$

which converges to 0 due to $\psi_k(y^{k+1};x) \to \psi^*$ proved above and property (6.6).

3) Let $(\alpha_{k+1},\phi_{k+1})$ be the cutting plane element obtained from the null step $y^{k+1}$ which we retain in $\mathcal{G}_{k+1}$. By construction this defines an affine support plane of $\widetilde{F}(\cdot;x)$ at $y^{k+1}$. But on the other hand the pair $(\alpha_{k+1},\phi_{k+1})$ also contributes to building of the new model $\widetilde{F}_{k+1}(\cdot;x)$, so the new model must be exact at $y^{k+1}$, because always $\widetilde{F}_{k+1} \le \widetilde{F}$, so the value of $\widetilde{F}$ is the best $\widetilde{F}_{k+1}$ could possible attain. In other words, $\phi_{k+1}$ is also a subgradient of $\widetilde{F}_{k+1}(\cdot;x)$ at $y^{k+1}$. That means

$$\phi_{k+1}^T(y-y^{k+1}) \le \widetilde{F}_{k+1}(y;x) - \widetilde{F}_{k+1}(y^{k+1};x).$$

Using $\widetilde{F}_{k+1}(y^{k+1};x) = \widetilde{F}(y^{k+1};x)$ we therefore have

$$\widetilde{F}(y^{k+1};x) + \phi_{k+1}^T(y - y^{k+1}) \le \widetilde{F}_{k+1}(y;x). \tag{6.8}$$

Now observe that

$$\begin{aligned}
0 &\le \widetilde{F}(y^{k+1};x) - \widetilde{F}_k(y^{k+1};x)\\
&= \widetilde{F}(y^{k+1};x) + \phi_{k+1}^T(y^{k+2} - y^{k+1}) - \widetilde{F}_k(y^{k+1};x) - \phi_{k+1}^T(y^{k+2} - y^{k+1})\\
&\le \widetilde{F}_{k+1}(y^{k+2};x) - \widetilde{F}_k(y^{k+1};x) + \|\phi_{k+1}\|\|y^{k+2} - y^{k+1}\| \qquad \text{(using (6.8))}
\end{aligned}$$

and this term tends to 0 because of (6.7), boundedness of $\phi_{k+1}$, and because $y^{k+1} - y^{k+2} \to 0$. We conclude that

$$\widetilde{F}(y^{k+1};x) - \widetilde{F}_k(y^{k+1};x) \to 0. \tag{6.9}$$

4) We now show that $\widetilde{F}_k(y^{k+1};x) \to F(x;x) = 0$, and therefore by (6.9) also $\widetilde{F}(y^{k+1};x) \to F(x;x) = 0$. Suppose contrary to the claim that $\eta := F(x;x) - \limsup \widetilde{F}_k(y^{k+1};x) > 0$. Choose $0 < \theta < (1-\widetilde{\gamma})\eta$. It follows from (6.9) that there exists $k_1 \ge k_0$ such that

$$\widetilde{F}(y^{k+1};x) - \theta \le \widetilde{F}_k(y^{k+1};x)$$

for all $k \ge k_1$. Using $\widetilde{\rho}_k < \widetilde{\gamma}$ for all $k \ge k_1$ gives

$$\begin{aligned}
\widetilde{\gamma}(\widetilde{F}_k(y^{k+1};x) - F(x;x)) &\le \widetilde{F}(y^{k+1};x) - F(x;x)\\
&\le \widetilde{F}_k(y^{k+1};x) + \theta - F(x;x).
\end{aligned}$$

Passing to the limit implies $\widetilde{\gamma}\eta \ge \eta - \theta$, contradicting the choice of $\theta$. This proves $\eta = 0$ as claimed.

5) Having shown $\widetilde{F}_k(y^{k+1};x) \to F(x;x) = 0$, we now argue that we must have $y^{k+1} \to x$. This follows from the definition of $y^{k+1}$, because

$$\psi_k(y^{k+1};x) = \widetilde{F}_k(y^{k+1};x) + \tfrac{\delta}{2}\|y^{k+1} - x\|^2 \le \psi_k(x;x) = F(x;x) = 0.$$

Since $\widetilde{F}_k(y^{k+1};x) \to 0$ by part 4), we have indeed $y^{k+1} \to x$. To finish the proof, observe that $0 \in \partial_1\psi_k(y^{k+1};x)$ implies

$$\begin{aligned}
\delta(x - y^{k+1})^T(y - y^{k+1}) &\le \widetilde{F}_k(y;x) \ - \widetilde{F}_k(y^{k+1};x)\\
&\le \widetilde{F}(y;x) - \widetilde{F}_k(y^{k+1};x)
\end{aligned} \tag{6.10}$$

for every $y$. Passing to the limit gives

$$0 \le \widetilde{F}(y;x) - \widetilde{F}(x;x),$$

because the left hand side in (6.10) converges to 0 in view of $y^{k+1} \to x$, and since $\widetilde{F}_k(y^{k+1};x) \to F(x;x)$ by 3) above. Since $\partial_1\widetilde{F}(x;x) \subset \partial_1 F(x;x)$, we are done. □

**6.2. Convergence of outer loop.** Let us consider the sequence $(x^j)_{j\in\mathbb{N}}$ of serious steps generated by algorithm 1. We want to show that $0 \in \partial_1 F(\bar{x};\bar{x})$ for every accumulation point $\bar{x}$ of $(x^j)_{j\in\mathbb{N}}$. We start by proving that under reasonable hypotheses, the sequence of serious iterates of our algorithm is bounded.

LEMMA 6.4. *Suppose the following two hypotheses are satisfied:*

($H_1$) *$g$ is weakly coercive in the sense that if a sequence $x^j$ satisfies $\|x^j\| \to \infty$ and $g(x^j) > \gamma_\infty^2$, then $g(x^j)$ is not strictly monotonically decreasing.*

($H_2$) *$f$ is weakly coercive on the level set $\{x \in \mathbb{R}^n : g(x) \le \gamma_\infty^2\}$ in the following sense: if $x^j$ is a sequence of feasible iterates with $\|x^j\| \to \infty$, then $f(x^j)$ is not strictly monotonically decreasing.*

*Then the sequence $x^j$ of serious iterates with starting point $x^1$ generated by our algorithm is bounded.*

*Proof.* There are two cases to be discussed.

a) Suppose the iterates are all infeasible $g(x^j) > \gamma_\infty^2$. In that case we use axiom ($H_1$). Notice that in phase I we have $g(x^{j+1}) - g(x^j) \le F(x^{j+1}, x^j) < 0$, so the sequence $g(x^j)$ is strictly decreasing. Then $x^j$ is bounded by axiom ($H_1$).

b) Suppose next that the iterates are feasible for $j \ge j_0$. In phase II we have $F(x^{j+1}, x^j) = \max\{f(x^{j+1}) - f(x^j), g(x^{j+1}) - \gamma_\infty^2\} \le 0$, hence $f(x^{j+1}) < f(x^j)$ for $j \ge j_0$. Then by axiom ($H_2$) the sequence $x^j$ could not be unbounded. □

REMARK. Notice that axiom ($H_2$) is certainly satisfied if $f$ is coercive in the usual sense on the feasible set, that is, if $f(x^j) \to \infty$ for feasible iterates with $\|x^j\| \to \infty$. Similarly, ($H_1$) could be replaced by the hypothesis that the set $\{x \in \mathbb{R}^n : \gamma_\infty^2 < g(x) \le g(x^1)\}$ is bounded. □

We are now ready to prove convergence of the outer loop of algorithm 1:

THEOREM 6.5. *Let axioms ($H_1$) and ($H_2$) be satisfied. Then every accumulation point $\bar{x}$ of the sequence of serious steps $x^j$ generated by the algorithm satisfies $0 \in \partial_1 F(\bar{x}; \bar{x})$. In particular, $\bar{x}$ is either a critical point of constraint violation, or a F. John critical point of the mixed $H_2/H_\infty$ program (2.3).*

*Proof.* The second part of the statement follows from Lemma 5.1. Let us prove $0 \in \partial_1 F(\bar{x}; \bar{x})$.

1) We first prove convergence $F(x^{j+1}; x^j) \to 0$, $(j \to \infty)$. By construction, we know that $F(x^{j+1}; x^j) \le 0$ for every $j \in \mathbb{N}$. We now distinguish two cases:

*1st case: there exists $j_0 \in \mathbb{N}$ such that $g(x^{j_0}) \le \gamma_\infty^2$.* From that index onwards we have

$$F(x^{j+1}; x^j) = \max\left\{f(x^{j+1}) - f(x^j); g(x^{j+1}) - \gamma_\infty^2\right\} \le 0,$$

hence $f(x^{j+1}) \le f(x^j)$ and $g(x^j) \le \gamma_\infty^2$. That means the sequence $(f(x^i))_{i\in\mathbb{N}}$ is monotone decreasing from $j_0$ onwards. For any accumulation point $\bar{x}$ of $(x^j)_{j\in\mathbb{N}}$, continuity of $f$ shows $f(\bar{x})$ is an accumulation point of $(f(x^i))_{i\in\mathbb{N}}$, and by the monotone sequences theorem, this implies $f(x^j) \longrightarrow f(\bar{x})$. Now for $j \ge j_0$ we have

$$F(x^{j+1}; x^j) = \max\left\{f(x^{j+1}) - f(x^j); g(x^{j+1}) - \gamma_\infty^2\right\},$$

hence: $\liminf_{j\to\infty} F(x^{j+1}; x^j) \ge \lim_{j\to\infty} f(x^{j+1}) - f(x^j) = 0$. In tandem with $F(x^{j+1}; x^j) \le 0$ this clearly implies $F(x^{j+1}; x^j) \to 0$, $(j \to \infty)$.

*2nd case: $g(x^j) > \gamma_\infty^2$ for all $j \in \mathbb{N}$.* Here

$$F(x^{j+1}; x^j) = \max\left\{f(x^{j+1}) - f(x^j) - \mu[g(x^j) - \gamma_\infty^2]; g(x^{j+1}) - g(x^j)\right\} \le 0.$$

Hence $(g(x^j))_{j\in\mathbb{N}}$ is monotonically decreasing. As in the first case, we prove that by continuity of $g$, $g(\bar{x})$ is an accumulation point and so a limit point of $(g(x^j))_{j\in\mathbb{N}}$. We deduce in the same way that $F(x^{j+1}; x^j) \to 0$.

2) Suppose that at the $j$th stage of the outer loop the inner loop accepts a serious step at $k = k_j$. Then $x^{j+1} = y^{k_j+1}$. By the definition of $y^{k_j+1}$ as minimizer of the

tangent program (5.6), this means

$$\delta_{k_j}\left(x^j - x^{j+1}\right) \in \partial_1 \widetilde{F}_{k_j}(x^{j+1}; x^j).$$

By the subgradient inequality this gives

$$\delta_{k_j}\left(x^j - x^{j+1}\right)^T\left(x^j - x^{j+1}\right) \leq \widetilde{F}_{k_j}(x^j; x^j) - \widetilde{F}_{k_j}(x^{j+1}; x^j) = -\widetilde{F}_{k_j}(x^{j+1}; x^j),$$

where $\widetilde{F}_{k_j}(x^j; x^j) = F(x^j; x^j) = 0$ by Lemma 5.4. Since $x^{j+1} = y^{k_j+1}$ was accepted in step 4 of the algorithm, we have $\rho_{k_j} \geq \gamma$, i.e.: $-\widetilde{F}_{k_j}(x^{j+1}; x^j) \leq -\gamma^{-1}F(x^{j+1}; x^j)$. Altogether

$$0 \leq \delta_{k_j}\|x^j - x^{j+1}\|^2 \leq -\gamma^{-1}F(x^{j+1}; x^j).$$

Since $F(x^{j+1}; x^j)$ converges to 0 by part 1), we deduce $\delta_{k_j}\|x^j - x^{j+1}\|^2 \to 0$. We claim that this implies $\phi_j = \delta_{k_j}\left(x^j - x^{j+1}\right) \to 0$, $(j \to \infty)$.

3) Suppose on the contrary that there exists an infinite subsequence $j \in \mathcal{N}$ of $\mathbb{N}$ such that $\|\phi_j\| = \delta_{k_j}\|x^j - x^{j+1}\| \geq \eta > 0$ for some $\eta > 0$ and every $j \in \mathcal{N}$. Therefore

$$\delta_{k_j}\|x^j - x^{j+1}\|^2 \geq \eta\|x^j - x^{j+1}\| \geq 0$$

for $j \in \mathcal{N}$, which implies $(x^j - x^{j+1})_{j\in\mathcal{N}} \to 0$. That is only possible when $(\delta_{k_j})_{j\in\mathcal{N}} \to \infty$. We now argue that there exists yet another infinite subsequence $\mathcal{N}'$ of $\mathbb{N}$ with the property that $\delta_{k_j} \to \infty$, $(j \in \mathcal{N}')$, and such that in addition for each $j \in \mathcal{N}'$, the doubling rule to increase $\delta_k$ in step 7 of the algorithm was applied at least once before $x^{j+1} = y^{k_j+1}$ was accepted by the inner loop. To construct $\mathcal{N}'$, we associate with every $j \in \mathcal{N}$ the last outer-loop instant $j' \leq j$ where the $\delta$-parameter was increased at least once while the inner loop was turning, and we let $\mathcal{N}'$ consist of all these $j'$, $j \in \mathcal{N}$. It could happen that $j' = j$, but in general we only know that

$$2\delta_{k_{j'-1}} \leq \delta_{k_{j'}} \quad \text{and} \quad \delta_{k_{j'}} \geq \delta_{k_{j'+1}} \geq \cdots \geq \delta_{k_j}.$$

The latter ensures $\delta_{k_{j'}} \to \infty$, $j' \in \mathcal{N}'$.

Let us say that for $j \in \mathcal{N}'$, the doubling rule was applied for the last time at $\delta_{k_j-\nu_j}$ for some $\nu_j \geq 1$. That is, we have $\delta_{k_j-\nu_j+1} = 2\delta_{k_j-\nu_j}$, while the $\delta$ parameter was frozen during the remaining steps before acceptance, i.e.:

$$\delta_{k_j} = \delta_{k_j-1} = \cdots = \delta_{k_j-\nu_j+1} = 2\delta_{k_j-\nu_j}. \tag{6.11}$$

Recall from step 7 of the algorithm that we have $\rho_k < \gamma$ and $\tilde{\rho}_k \geq \tilde{\gamma}$ for those $k$, where the step was not accepted and the doubling rule was applied. That is,

$$\rho_{k_j-\nu_j} = \frac{F(x^j; x^j) - F(y^{k_j-\nu_j+1}; x^j)}{F(x^j; x^j) - \widetilde{F}_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j)} = \frac{F(y^{k_j-\nu_j+1}; x^j)}{\widetilde{F}_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j)} < \gamma$$

and

$$\tilde{\rho}_{k_j-\nu_j} = \frac{F(x^j; x^j) - \widetilde{F}(y^{k_j-\nu_j+1}; x^j)}{F(x^j; x^j) - \widetilde{F}_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j)} = \frac{\widetilde{F}(y^{k_j-\nu_j+1}; x^j)}{\widetilde{F}_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j)} \geq \tilde{\gamma}.$$

By definition of $y^{k_j-\nu_j+1}$ and according to (6.11), we now have

$$\tfrac{1}{2}\delta_{k_j}\left(x^j - y^{k_j-\nu_j+1}\right) \in \partial_1 \widetilde{F}_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j).$$

Using $\widetilde{F}_{k_j-\nu_j}(x^j;x^j) = F(x^j;x^j) = 0$ and the subgradient inequality for $\widetilde{F}_{k_j-\nu_j}(\cdot;x^j)$ at $y^{k_j-\nu_j+1}$ gives

$$\begin{aligned}\tfrac{1}{2}\delta_{k_j}\left(x^j - y^{k_j-\nu_j+1}\right)^T\left(x^j - y^{k_j-\nu_j+1}\right) &\le \widetilde{F}_{k_j-\nu_j}(x^j;x^j) - \widetilde{F}_{k_j-\nu_j}(y^{k_j-\nu_j+1};x^j)\\ &\le -\widetilde{F}_{k_j-\nu_j}(y^{k_j-\nu_j+1};x^j).\end{aligned}$$

This could also be written as

$$\frac{\delta_{k_j}\|x^j - y^{k_j-\nu_j+1}\|^2}{-\widetilde{F}_{k_j-\nu_j}(y^{k_j-\nu_j+1};x^j)} \le 2. \tag{6.12}$$

Now we know from Lemma 6.4 that the set of serious iterates $x^j$ is bounded. In tandem with Lemma 6.1, which relates the norm of the null steps $y^{k+1}$ to the norm of $x^j$, we deduce that the set $B = \{x^j : j \in \mathbb{N}\} \cup \{y^{k+1} : k = 1,\dots,k_j, j \in \mathbb{N}\}$ is bounded. Then Lemma 5.2 provides $L > 0$ such that

$$|F(y^{k_j-\nu_j+1};x^j) - \widetilde{F}(y^{k_j-\nu_j+1};x^j)| \le L\|y^{k_j-\nu_j+1} - x^j\|^2 \tag{6.13}$$

for all $j \in \mathcal{N}'$. Now expanding the expression $\tilde{\rho}_{k_j-\nu_j}$ gives

$$\begin{aligned}\tilde{\rho}_{k_j-\nu_j} &= \rho_{k_j-\nu_j} + \frac{F(y^{k_j-\nu_j+1};x^j) - \widetilde{F}(y^{k_j-\nu_j+1};x^j)}{-\widetilde{F}_{k_j-\nu_j}(y^{k_j-\nu_j+1};x^j)}\\ &\le \rho_{k_j-\nu_j} + \frac{L\|x^j - y^{k_j-\nu_j+1}\|^2}{-\widetilde{F}_{k_j-\nu_j}(y^{k_j-\nu_j+1};x^j)} && \text{(using (6.13))}\\ &\le \rho_{k_j-\nu_j} + \frac{2L}{\delta_{k_j}} && \text{(using (6.12))}\end{aligned}$$

Since $\rho_j < \gamma$ and $L/2\delta_{k_j} \to 0$ for the infinite subsequence $j \in \mathcal{N}'$, we deduce $\limsup_{j\in\mathcal{N}'} \tilde{\rho}_{k_j-\nu_j} \le \limsup_{j\in\mathcal{N}'} \rho_{k_j-\nu_j} \le \gamma < \tilde{\gamma}$, contradicting $\tilde{\rho}_j \ge \tilde{\gamma} > \gamma$ for the infinitely many $j \in \mathcal{N}'$. This proves that an infinite sequence $j \in \mathcal{N}$ with $\|\phi_j\| \ge \eta > 0$ could not exist. The conclusion is that $(\phi_j)_{j\in\mathbb{N}} = (\delta_{k_j}(x^j - x^{j+1}))_{j\in\mathbb{N}}$ converges to 0.

4) Let $\bar{x}$ be an accumulation point of the sequence of serious steps $x^j$ and pick a convergent subsequence $x^j \to \bar{x}$, $j \in \mathcal{N}$. We have to prove $0 \in \partial_1 F(\bar{x};\bar{x})$.

Since $\phi_j = \delta_{k_j}(x^j - x^{j+1})$ is a subgradient of $\widetilde{F}_{k_j}(\cdot,x^j)$ at $y^{k_j+1} = x^{j+1}$ we have

$$\begin{aligned}\phi_j^T h &\le \widetilde{F}_{k_j}(x^{j+1}+h;x^j) - \widetilde{F}_{k_j}(x^{j+1};x^j)\\ &\le \widetilde{F}(x^{j+1}+h;x^j) - \widetilde{F}_{k_j}(x^{j+1};x^j) && \text{(using } \widetilde{F}_{k_j} \le \widetilde{F})\end{aligned}$$

for every test vector $h \in \mathbb{R}^n$. Now we use the fact that $y^{k_j+1} = x^{j+1}$ was accepted in step 4 of the algorithm. That means

$$-\widetilde{F}_{k_j}(x^{j+1};x^j) \le -\gamma^{-1}F(x^{j+1};x^j).$$

Combining these two estimates gives

$$\phi_j^T h \le \widetilde{F}(x^{j+1}+h;x^j) - \gamma^{-1}F(x^{j+1};x^j) \tag{6.14}$$

for every test vector $h$. Now fix $h' \in \mathbb{R}^n$ and choose the test vector $h^j = x^j - x^{j+1} + h'$ for $j \in \mathcal{N}'$. Substituting this in (6.14) we obtain

$$\delta_{k_j}\|x^j - x^{j+1}\|^2 + \phi_j^T h' \le \widetilde{F}(x^j + h';x^j) - \gamma^{-1}F(x^{j+1};x^j). \tag{6.15}$$

Now observe that $\delta_{k_j}\|x^j - x^{j+1}\|^2 \to 0$ by part 2), and $\phi_j = \delta_{k_j}(x^j - x^{j+1}) \to 0$ by part 3). This means that the left hand side of (6.15) converges to 0. As for the terms on the right, recall that $F(x^{j+1}; x^j) \to 0$ by part 1) of the proof. Finally, by joint continuity of $\widetilde{F}(\cdot;\cdot)$, the term $\widetilde{F}(x^j + h'; x^j)$ converges to $\widetilde{F}(\bar{x} + h'; \bar{x})$. We conclude, passing to the limit $j \in \mathcal{N}'$ in (6.15), and using $\widetilde{F}(\bar{x}; \bar{x}) = 0$, that

$$0 \leq \widetilde{F}(\bar{x} + h'; \bar{x}) = \widetilde{F}(\bar{x} + h'; \bar{x}) - \widetilde{F}(\bar{x}; \bar{x}).$$

As this works for every $h' \in \mathbb{R}^n$, we have shown $0 \in \partial_1 \widetilde{F}(\bar{x}; \bar{x})$, hence also $0 \in \partial_1 F(\bar{x}; \bar{x})$. □

**7. Implementation.** Algorithm 1 has been implemented for both structured and unstructured mixed synthesis, and we use the enriched versions of $\mathcal{G}_0$ and $\mathcal{G}_k^c$ to speed up convergence. Notice that in some of the examples in section 8, the controller has to be strictly proper to ensure well-posedness of the $H_2$ norm. (Namely $D_K = 0$ in (2.2) when $D_{2u}$ and $D_{y2}$ are non zero in the plant (2.1)). In those cases the data in (2.2) are no longer freely assigned, the parameterizations being $K = \mathcal{K}(A_K, B_K, C_K)$ with a linear operator $\mathcal{K}$. More general types of parameterizations would equally well fit into our approach, and are referred to as structural constraints on the controller.

**7.1. Stopping criteria.** Notice that algorithm 1 is a first order method, which may be slow in the neighborhood of a local solution of (2.3). As in [5], we have therefore implemented termination criteria, which avoid pointless computational efforts during the final phase, where iterates make minor progress. Our first stopping test checks criticality $0 \in \partial_1 F(x; x)$ by computing

$$\inf\{||h|| : h \in \partial_1 F(x; x)\} < \varepsilon_1.$$

Notice that this program is similar (but easier) than the SDP discussed in Section 5.3, because the linear terms in that cast are not needed.

A second test compares the progress of the local model around the current iterate:

$$|F(x^+; x)| \leq \varepsilon_2. \tag{7.1}$$

Our third test compares the relative step length to the controller gains:

$$||x^+ - x|| \leq \varepsilon_3(1 + ||x||). \tag{7.2}$$

For stopping, we require that either the first, or the second and third be satisfied.

**7.2. Choice of the performance level $\gamma_\infty$.** In all test examples we first compute (locally) optimal $H_2$ and $H_\infty$ controllers $K_2$ and $K_\infty$. It is now trivial (see e.g. [10]) that the performance level $\gamma_\infty$ in program (2.3) has to satisfy

$$\|T_\infty(K_\infty)\|_\infty \leq \gamma_\infty < \|T_\infty(K_2)\|_\infty. \tag{7.3}$$

Indeed, the mixed $H_2/H_\infty$ problem (2.3) is infeasible for $\gamma_\infty < \|T_\infty(K_\infty)\|_\infty$, while for $\gamma_\infty \geq \|T_\infty(K_2)\|_\infty$ the optimal $H_2$ controller $K_2$ is also optimal for (2.3). Disregarding complications due to (multiple) local minima, it would make sense, in a specific case study, to consider the entire one parameter family $K(\gamma_\infty)$ of solutions of (2.3) as a function of the gain value $\gamma_\infty$ over the range (7.3), as this would transform $K_\infty$ continuously into $K_2$ (see e.g. figure 7.1). In our tests we only compute $K(\gamma_\infty)$ for those values $\gamma_\infty$ which allow comparison to previous results in the literature.

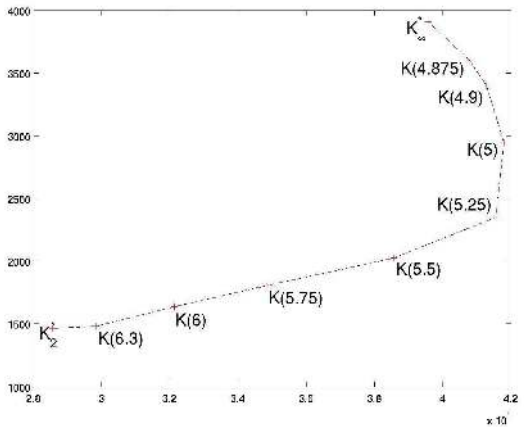


FIG. 7.1. *$H_2/H_\infty$ optimal static controllers $K(\gamma_\infty) = (K_1(\gamma_\infty), K_2(\gamma_\infty)) \in \mathbb{R}^2$ for the vehicular suspension control problem. $[\|T_\infty(K_\infty)\|_\infty, \|T_\infty(K_2)\|_\infty] \ni \gamma_\infty \mapsto K(\gamma_\infty)$ continuously transforms the $H_\infty$ optimal gain $K_\infty$ into the $H_2$ optimal gain $K_2$.*

Table 7.1 reports the problem dimensions $n_x, n_y, n_u$ and the synthesized controller orders $n_K$. Columns 5 and 6 report $\|T_\infty(K_\infty)\|_\infty$ and $\|T_\infty(K_2)\|_\infty$, which are the bounds in (7.3), needed to choose $\gamma_\infty$ correctly. In column 4 we report $\|T_2(K_2)\|_2$, because it gives a lower bound on the optimal value $\|T_2(K(\gamma_\infty))\|_2$ of (2.3).

Notice that in columns $\|T_2(K_2)\|_2$ and $\|T_\infty(K_\infty)\|_\infty$ we would expect decreasing values for a fixed example as $n_K$ increases. However, in 'CM4' the orders 0 and 50 give successively $9.2645e-01$ and $9.3844e-01$, which is not as it should be, because the order 50 controller is worse than the static controller. This phenomenon is due to the fact that in all cases $n_K < n_x$, we only compute local minima of the $H_\infty$ program, and similarly, of the $H_2$ program. As $n_K$ increases, more local minima appear, and it may be very difficult to improve the situation. This is obviously very unsatisfactory, and appropriate procedures to initialize at a given order $n_K$ are currently investigated.

**7.3. Initialization by a stabilizing controller.** In all our test examples, we use the techniques in [11] to compute a closed-loop stabilizing initial $K^0$, which is not necessarily feasible for (2.3). This allows to test phase I of our method. $K_\infty$ may always be chosen as a feasible initial iterate, so that phase I could in principle be avoided, but we prefer to use various ways to initialize algorithm 1. In the full order case $n_K = n_x$, $K_2$ and $K_\infty$ are computed by AREs as routinely available in the MATLAB control toolbox. In the reduced order case $n_K < n_x$, things are more complicated, and minima are in general only local. The locally optimal $H_\infty$ controller $K_\infty$ is computed by the method of [5], which uses the initial closed loop stabilizing $K^0$ to initialize the procedure. Methods to compute $K_2$ in the reduced order case $n_K < n_x$ are discussed in [42]. Since the objective function $f(K)$ is not defined everywhere, standard software for unconstrained programming may face difficulties, and we have implemented a Polak-Rivière conjugate gradient method (with a special safeguard to stay in the set $D$ of exponentially stabilizing controllers) to compute $K_2$. An alternative is of course to use algorithm 1 with $\gamma_\infty$ so large that $\gamma_\infty > \|T_\infty(K_2)\|_2$ can be assured. But this is often slow, because algorithm 1 is a first order method. This confirms the observation of the authors of [42], who report slow convergence

| Problem | $(n_x, n_y, n_u)$ | $n_K$ | $\|T_2(K_2)\|_2$ | $\|T_\infty(K_2)\|_\infty$ | $\|T_\infty(K_\infty)\|_\infty$ |
|---|---|---|---|---|---|
| Academic ex. [10] | $(2,1,1)$ | 0 | $6^{\frac{1}{4}}$ | $\frac{3}{\sqrt{5}}$ | 1 |
| Academic ex. [44] | $(3,1,1)$ | 3 | 7.748 | 23.586 | 9.5196 |
| Vehicular | $(4,2,1)$ | 0 | 32.416 | 6.3287 | 4.8602 |
| suspension [50] | | 2 | 32.299 | 6.1828 | 4.8573 |
| | | 4 | 32.267 | 6.3260 | 4.6797 |
| Four Disks [27] | $(8,1,1)$ | 2 | 0.5319 | 3.1658 | 0.31411 |
| | | 4 | 0.4767 | 2.6194 | 0.31393 |
| | | 8 | 0.3782 | 1.39 | 0.27537 |
| From $COMPl_eib$: | | | | | |
| 'AC14' | $(40,4,3)$ | 1 | 21.369 | 230.8318 | 104.15 |
| | | 10 | 8.1039 | 100.4121 | 100.11 |
| | | 20 | 7.5628 | 100.3566 | 100 |
| 'BDT2' | $(82,4,4)$ | 0 | 7.9389e-01 | 1.3167 | 0.67421 |
| | | 10 | 7.8877e-01 | 1.1386 | 0.72423 |
| | | 41 | 7.7867e-01 | 1.1302 | 0.77405 |
| 'HF1' | $(130,1,2)$ | 0 | 5.8193e-02 | 0.4611 | 0.44721 |
| | | 10 | 5.8151e-02 | 0.4617 | 0.44721 |
| | | 25 | 5.8149e-02 | 0.4613 | 0.44721 |
| 'CM4' | $(240,1,2)$ | 0 | 9.2645e-1 | 1.6546 | 0.81650 |
| | | 50 | 9.3844e-1 | 4.2541 | 0.81746 |

TABLE 7.1

*Problem dimensions and bounds obtained from locally optimal $H_2$ and $H_\infty$ synthesis for the test examples in section 8.*

for $H_2$-synthesis based on first order (gradient type) methods and recommend using second order methods instead.

**7.4. Stability constraint.** Notice that closed-loop stability of $K$ is not a constraint in the usual sense of mathematical programming, because the set $D$ of closed-loop exponentially stabilizing $K$ is an open domain. In the cast (2.3), closed-loop stability $K \in D$ is a hidden constraint, which may cause problems because the functions $f$ and $g$ are not defined outside $D$. The strategy which we adopt here is to compute an initial closed-loop stabilizing controller $K^0 \in D$, and ignore the hidden constraint during the optimization process. Since $f(K^0) < \infty$ and $g(K^0) < \infty$, our algorithm produces iterates $K^j$ with $f(K^j) < \infty$ and $g(K^j) < \infty$ at all times $j$, and most of the time this assures that $K^j$ remains closed-loop stabilizing, i.e., $K^j \in D$.

**7.5. Choice of $\mu$.** In [41, section 2.6] a similar progress function is discussed for objectives which are maxima of finite or infinite families of smooth functions, but a line search method is obtained. In both cases convergence theory works for arbitrary values of the parameter $\mu$, so that no immediate insight into the choice of $\mu$ is obtained. Yet in practice the choice of $\mu$ may influence the actual performance of the algorithm.

Figure 7.2 and Table 7.2 present the numerical results of our nonsmooth algorithm for the four disks problem presented in section 8. After computing an initial stabilizing controller $K_0$, the nonsmooth algorithm is run with four different values of the penalty parameter $\mu$, including the case $\mu = 0$ to compare with the improvement function of [43].

As we can see in Table 7.2, for $\mu = 0$ the algorithm fails to reach a feasible point. This is indeed a case where we could identify the final $K$ of phase I where $g(K) > \gamma_\infty^2$ as a local minimum of $f$ alone. Recall that when $\mu = 0$, every descent step of the

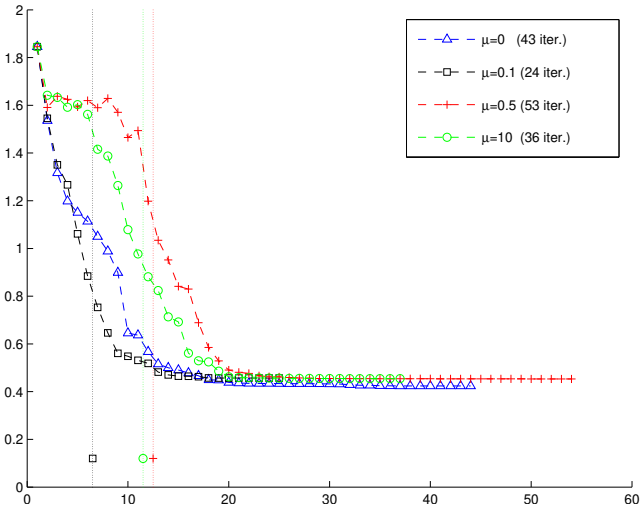


FIG. 7.2. *Full order mixed $H_2/H_\infty$ synthesis for the four disks problem: the values of the $H_2$ norm in relation to the number of serious steps are shown for four different values of the penalty parameter $\mu$. Here we choose: $\mu \in \{0, 0.1, 0.5, 10\}$. Vertical lines point out the instant at which the iterates become feasible.*

| Problem | Four Disks [27] | | | |
|---|---|---|---|---|
| $(n_x, n_y, n_u)$ | $(8, 1, 1)$ | | | |
| $\gamma_\infty$ | 0.6 | | | |
| $\mu$ | 0 | 0.1 | 0.5 | 10 |
| Serious steps | 43 | 24 | 53 | 36 |
| $\|T_2(K(\gamma_\infty))\|_2^2$ | 0.1795 | 0.2087 | 0.2054 | 0.2068 |
| $\|T_\infty(K(\gamma_\infty))\|_\infty$ | 0.7411 | 0.6000 | 0.6000 | 0.6000 |
| Stop test | Tests (7.1) and (7.2) | Criticality | | |

TABLE 7.2
*Data and numerical results of $H_2/H_\infty$ synthesis for the four disks for four different values of $\mu$.*

improvement function is a descent step of both $f$ and $g$, and the algorithm gets then trapped as soon as it reaches a local minimum of either $f$ or $g$. Choosing $\mu > 0$ allows a possible increase of the objective $f$ during phase I, so that being trapped at an infeasible local minimum of $f$ alone can be avoided.

Among the choices $\mu > 0$ we have noticed that when $\mu$ is not too small, the number of iterations to reach a feasible point decreases as $\mu$ increases. However choosing too large a $\mu$ as shown by the two last columns in Table 7.2, does not give the best results either, so this trend seems to be true only on a certain range. Nothing decisive can be proposed to date, but $\mu$ of the same order of magnitude as the progress function without the penalty term gave so far the best results in practice.

**7.6. Choice of $\Gamma$.** The last issue we address is the choice of $\Gamma$, which is crucial, because step 5 is the only place in the algorithm where the proximity parameter $\delta_k$ can be reduced. Too large a $\Gamma$ gives few reductions of $\delta_k$, and since the latter is often increased during the inner loop, this bears the risk of exceedingly large $\delta_k$, causing the algorithm to stop.

To illustrate this observation, we have run the four disks example in Section 8 for three different values $\Gamma \in \{0.4, 0.6, 0.8\}$. The results are illustrated in Figure 7.3.

We observe that the number of iterations increases with the values of $\Gamma$. The best numerical results were obtained for $\Gamma = 0.6$, and this is the value we retained for all the numerical tests. At least over a certain range one can say that the larger $\Gamma$, the smaller the steps accepted as serious steps $x \to x^+$, and the more outer iterations are needed to reach the same $H_2$ performance.

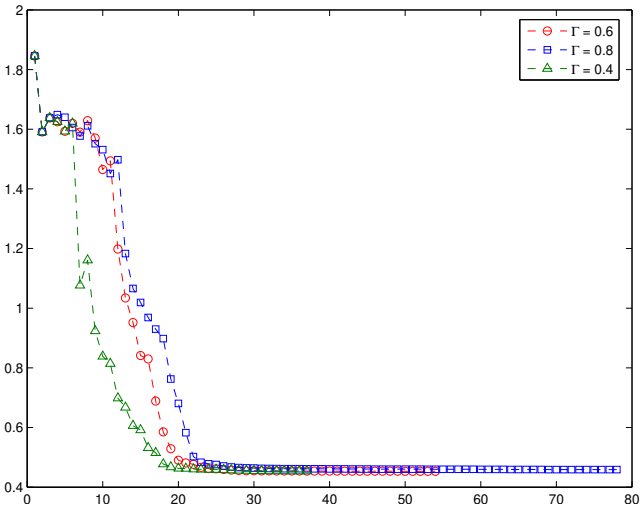


FIG. 7.3. *Full order mixed $H_2/H_\infty$ synthesis for the four disks problem: the values of the $H_2$ norm versus the number of serious steps for three different values of the parameter* $\Gamma \in \{0.4, 0.6, 0.8\}$.

| Problem | Four Disks [27] | | |
|---|---|---|---|
| $(n_x, n_y, n_u)$ | $(8, 1, 1)$ | | |
| $\gamma_\infty$ | 0.6 | | |
| $\mu$ | 0.5 | | |
| $\Gamma$ | 0.4 | 0.6 | 0.8 |
| serious steps | 36 | 53 | 77 |
| $\|T_2(K(\gamma_\infty))\|_2^2$ | 0.2062 | 0.2054 | 0.2106 |
| $\|T_\infty(K(\gamma_\infty))\|_\infty$ | 0.6000 | 0.6000 | 0.6000 |

**8. Numerical experiments.** In this section we test our nonsmooth algorithm on a variety of $H_2/H_\infty$ synthesis problems from the literature.

**8.1. Two academic examples.** We first present two academic examples whose models are described in [10] and [44, example 1]. Notice that the first one is simple enough to allow explicit computation of static output feedback controllers $u = Ky$ for $H_2$, $H_\infty$ and $H_2/H_\infty$ synthesis. The problem data are given in Table 7.1.

Table 8.1 confirms that our proximity control algorithm successfully performs the $H_2/H_\infty$ synthesis on the two considered examples. We not only improve the results computed by LMI approaches in [10] and [44], we even obtain the theoretical values of the $H_2$ and $H_\infty$ norms.

**8.2. Vehicular suspension controller design.** The model of the vehicular suspension is described in [15] and [50]. We first focus on static $H_2/H_\infty$-synthesis. The $H_\infty$ performance level in (2.3) is chosen as $\gamma_\infty = 5.225$ and the optimal solution we obtain is

$$K(\gamma_\infty) = [\ 41600 \quad 2393\ ].$$

| Problem | Academic ex. [10] | | | Academic ex. [44] | |
|---|---|---|---|---|---|
| $(n_x, n_y, n_u)$ | $(2,1,1)$ | | | $(3,1,1)$ | |
| $\mu$ | 10 | | | 1 | |
| $n_K$ | 0 | 0 | 1 | 3 | |
| $\gamma_\infty$ | 2 | 1.2 | 1.2 | 23.6 | 12 |
| Serious steps | 6 | 8 | 14 | 83 | 56 |
| $\|T_2(K(\gamma_\infty))\|_2$ | 1.5651 | 1.5735 | 1.5394 | 7.7484 | 10.4538 |
| $\|T_\infty(K(\gamma_\infty))\|_\infty$ | 1.3416 | 1.2 | 1.2 | 23.591 | 12.0000 |
| $K(\gamma_\infty)$ | [ −0.8165 ] | [ −0.9458 ] | $K_{f_1}$ | $K_{f_2}$ | $K_{f_3}$ |
| Stop test | Criticality | | | | |
| (LMI) $H_2$ norm | - | 1.5778 | - | 8.07 | - |
| Explicit $H_2$ norm | - | 1.5735 | - | 7.748 | - |

TABLE 8.1

*Results of $H_2/H_\infty$ synthesis for two academic examples*

$$K_{f_1} = \begin{bmatrix} -1.437 & -0.8101 \\ 0.8141 & -0.4998 \end{bmatrix}, \; K_{f_2} = \begin{bmatrix} -2.5810 & 1.0823 & -0.0623 & -0.5097 \\ -0.5748 & -1.5170 & 2.1121 & 1.6238 \\ -0.1396 & -2.8266 & -2.1852 & 1.7986 \\ 0.2724 & -0.4702 & -2.6967 & 0 \end{bmatrix}$$

$$K_{f_3} = \begin{bmatrix} -1.9113 & -0.7161 & -1.8332 & -0.0065 \\ 0.6940 & -4.4787 & 1.7584 & -2.1896 \\ 0.5231 & -3.2821 & -3.0458 & 3.4518 \\ -3.2830 & 1.1238 & -2.6107 & 0 \end{bmatrix}$$

The $H_2$ norm computed by our algorithm is $||T_2(K(\gamma_\infty))||_2 = 34.446$, compared to 35.8065 obtained in [50], which gives an improvement of 3.8%. Moreover, the $H_\infty$ performance is $||T_\infty(K(\gamma_\infty))||_\infty = 5.2250$, compared to 5.0506 obtained in [50]. This shows that the $H_\infty$ constraint is not active in the heuristic [50], highlighting the inevitable conservatism of the LMI approaches. In contrast, our method always attains the constraint within the numerical precision.

These results are shown in Table 8.2, which also gives the results of $H_2/H_\infty$ synthesis for dynamic controllers of order $n_K = 2, 4$. Notice that in the first column

| Problem | Vehicular suspension controller design [50] | | | |
|---|---|---|---|---|
| $(n_x, n_y, n_u)$ | $(4,2,1)$ | | | |
| $n_K$ | 0 | 0 | 2 | 4 |
| $\mu$ | 1 | 1 | $10^2$ | $10^2$ |
| $\gamma_\infty$ | 10 | 5.225 | 5.225 | 5.225 |
| Serious steps | 502 | 155 | 496 | 157 |
| $\|T_2(K(\gamma_\infty))\|_2$ | 32.474 | 34.446 | 33.312 | 33.311 |
| $\|T_\infty(K(\gamma_\infty))\|_\infty$ | 6.2641 | 5.2250 | 5.2250 | 5.2236 |
| $K(\gamma_\infty)$ | [ 37016 1473 ] | [ 41600 2393 ] | $K_{f_1}$ | $K_{f_2}$ |
| Stop test | Criticality | | | |

TABLE 8.2

*Mixed $H_2/H_\infty$ synthesis for the vehicular suspension problem*

$$K_{f_1} = \begin{bmatrix} 0.0895 & 0.3310 & 0.7272 & 0.0644 \\ -0.0670 & -0.1540 & 0.4871 & -0.0103 \\ 0.7986 & 0.2764 & 0.1618 & 1.7366 \end{bmatrix} e+03$$

$$K_{f_2} = \begin{bmatrix} -0.2156 & -0.5614 & -0.0012 & 0.0006 & -0.0927 & 0.0817 \\ 0.0728 & 0.1511 & -0.0042 & 0.0020 & -0.5677 & -0.0378 \\ 0.0010 & -0.0003 & 0.0011 & 0.0044 & 0.0004 & 0.0005 \\ 0.0044 & -0.0018 & -0.0006 & -0.0018 & 0.0007 & 0.0005 \\ 0.4743 & -0.2764 & 0.0004 & 0.0004 & 0.0922 & 1.7370 \end{bmatrix} e+03$$

of Table 8.2 by choosing the $H_\infty$ performance level $\gamma_\infty > \gamma_2 = \|T_\infty(K_2)\|_\infty$ where $\|T_\infty(K_2)\|_\infty$ is given in Table 7.1, the $H_2/H_\infty$ solution is close to the solution of the $H_2$ synthesis.

**8.3. Four disks.** The four disks model is originally described in [16] and has previously been studied to evaluate reduced-order design methods. The open loop plant is of order $n_x = 8$ and has two stable poles.

We first focus on mixed $H_2/H_\infty$ synthesis of full order controllers in order to compare our nonsmooth algorithm to the original Riccati equation approach in [27]. The results are presented in Table 8.3. We also give results of $H_2/H_\infty$ synthesis of reduced-order controllers in Table 8.4.

| Problem | Four Disks [27] | | | | |
|---|---|---|---|---|---|
| $(n_x, n_y, n_u)$ | $(8,1,1)$ | | | | |
| $\mu$ | 0.1 | 0.1 | 0.1 | 0.1 | 0.5 |
| $\gamma_\infty$ | 1 | 0.9 | 0.8 | 0.7 | 0.52 |
| Serious steps | 35 | 17 | 29 | 49 | 39 |
| $\|T_2(K(\gamma_\infty))\|_2^2$ | 0.1558 | 0.1612 | 0.1707 | 0.1829 | 0.2299 |
| $\|T_\infty(K(\gamma_\infty))\|_\infty$ | 1.000 | 0.9000 | 0.8000 | 0.7000 | 0.5200 |
| Stop test | Criticality | | | | |
| Square $H_2$ norm in [27] | 0.168 | 0.176 | 0.187 | 0.203 | 0.262 |
| $H_\infty$ norm in [27] | .855 | 0.797 | 0.732 | 0.661 | 0.511 |
| Improvement | 7.26% | 8.41 % | 8.72% | 9.90% | 12.25% |

TABLE 8.3

*Full order mixed $H_2/H_\infty$ synthesis for the four disks problem ($n_K = 8$): the square $H_2$ norm is computed in order to compare our results to those in [27]*

| Problem | Four Disks [27] | | | |
|---|---|---|---|---|
| $(n_x, n_y, n_u)$ | $(8,1,1)$ | | | |
| $\mu$ | 1 | | | |
| $\gamma_\infty$ | 0.52 | | | |
| $n_K$ | 2 | 4 | 6 | 7 |
| Serious steps | 23 | 18 | 30 | 47 |
| $\|T_2(K(\gamma_\infty))\|_2$ | 0.2321 | 0.2308 | 0.23041 | 0.2304 |
| $\|T_\infty(K(\gamma_\infty))\|_\infty$ | 0.52 | 0.52 | 0.52 | .52 |
| Stop test | Criticality | | | |

TABLE 8.4

*Reduced order mixed $H_2/H_\infty$ synthesis for the four disks problem.*

As can be seen in Table 8.3, our method gives significant improvement over the older results in [27] based on coupled Riccati equations. This highlights the reduction of conservatism of our approach compared to Riccati and LMI methods.

**8.4.** $COMPl_eib$ **examples.** The models in this section are from the $COMPl_eib$ collection [34]: aircraft model 'AC14', distillation tower 'BDT2', heat flow in a thin rod 'HF1' and cable mass model 'CM4'. They are originally designed for $H_\infty$ synthesis, so an $H_2$ channel was added as suggested by F. Leibfritz [33, 34]. The same channel is used for both $H_2$ and $H_\infty$ performance in example 'AC14', while we choose $B_2 = B_\infty$ and $D_{y2} = 0$ for the three others models. This way the $H_2$ norm is well-posed.

In each example, we first choose the $H_\infty$ performance level $\gamma_\infty$ larger than $\|T_\infty(K_2)\|_\infty$. In doing this we have to obtain an estimate of the optimal $H_2$ performance $\|T_2(K_2)\|_2$ given in Table 7.1. Numerical results are in Tables 8.5 and 8.6.

As an illustration, Figs. 8.1 and 8.2 show the evolution of the $H_2$ and $H_\infty$ norms for example 'BDT2' during the first iterations.

In Fig. 8.1 we observe phase I and phase II of the algorithm. As long as iterates remain infeasible, descent steps to reduce constraint violation are generated, sometimes causing the objective to increase. As soon as the feasible domain $g(x) \leq \gamma_\infty^2$ is

| Pb | $(n_x, n_y, n_u)$ | $n_K$ | $\gamma_\infty$ | Serious steps | $\|T_2(K(\gamma_\infty))\|_2$ | $\|T_\infty(K(\gamma_\infty))\|_\infty$ |
|---|---|---|---|---|---|---|
| 'AC14' | $(40, 4, 3)$ | 1 | 1000 | 300(max.) | 21.370 | 231.31 |
| | | 10 | 1000 | 300(max.) | 8.7813 | 101.26 |
| | | 1 | 200 | 263$^\star$ | 21.476 | 200 |
| | | 20 | 200 | 300(max.) | 7.9879 | 100 |
| 'BDT2' | $(82, 4, 4)$ | 0 | 10 | 148$^\star$ | 8.0402e-01 | 1.0585 |
| | | 10 | 10 | 543$^\star$ | 7.6480e-01 | 1.1438 |
| | | 0 | 0.8 | 324$^\star$ | 7.9092e-01 | 7.9999e-01 |
| | | 10 | 0.8 | 404$^\star$ | 7.7146e-01 | 0.8000 |
| | | 41 | 0.8 | 115$^\star$ | 7.8882e-01 | 0.8000 |
| 'HF1' | $(130, 1, 2)$ | 0 | 10 | 7 | 5.8193e-02 | 4.6087e-01 |
| | | 0 | 0.45 | 7$^\star$ | 5.8795e-02 | 4.4999e-01 |
| | | 10 | 0.45 | 7$^\star$ | 5.8706e-02 | 4.5000e-01 |
| | | 25 | 0.45 | 33$^\star$ | 5.8700e-02 | 4.4993e-01 |
| 'CM4' | $(240, 1, 2)$ | 0 | 10 | 5$^\star$ | 9.2645e-01 | 1.6555 |
| | | 0 | 1 | 20$^\star$ | 9.8438e-01 | 1 |
| | | 25 | 1 | 15$^\star$ | 9.5330e-01 | 1.000 |
| | | 50 | 1 | 41$^\star$ | 9.4038e-01 | 1.000 |

TABLE 8.5

*Results of mixed $H_2/H_\infty$ synthesis for test examples from COMPl$_e$ib - Criticality is pointed out by a $\star$ on the number of serious steps*

| Problem | $\gamma_\infty$ | $K(\gamma_\infty)$ |
|---|---|---|
| 'BDT2' | 10 | $\begin{bmatrix} -0.6186 & -0.1426 & -0.5414 & 4.929 \\ 0.6357 & -0.5457 & -3.851 & 16.85 \\ -0.07527 & 0.2962 & -1.287 & 6.601 \\ 0.9223 & 0.4668 & -4.091 & 22.34 \end{bmatrix}$ |
| | .8 | $\begin{bmatrix} -0.9207 & 0.9647 & -5.4243 & 9.8225 \\ 0.7452 & -1.3280 & -4.4241 & -0.8141 \\ -0.7119 & 2.1754 & -10.226 & 14.3827 \\ 0.0887 & 1.7433 & -13.4102 & 12.1358 \end{bmatrix}$ |
| 'HF1' | 10 | $[\ -0.1002 \quad -1.1230\ ]$ |
| | .45 | $[\ -0.2521 \quad -1.116\ ]$ |
| 'CM4' | 10 | $[\ -0.5448 \quad -1.3322\ ]$ |
| | 1 | $[\ -0.5146 \quad -0.8073\ ]$ |

TABLE 8.6

*Static $H_2/H_\infty$ output feedback controllers for examples from COMPl$_e$ib*

reached, descent of the objective $f$ begins, and iterates stay feasible.

Fig. 8.2 shows the frequency plot $\omega \mapsto \lambda_1\left(T_\infty(K_i, j\omega)^H T_\infty(K_i, j\omega)\right)$ of the $H_\infty$ constraint during the first 6 iterations (serious steps) $K_i$, $i = 1, \ldots, 6$, along with the second eigenvalue $\lambda_2$ (in blue). As can be seen, the maximum $\|T_\infty(K_i)\|_\infty^2$ is sometimes attained at a single marked peak $\omega$, while other cases feature rather a flat plateau in the low frequency band. Multiple peaks appear usually at the end of the process, but cannot be ruled out at any moment, as shown by the lower right plot, which has a plateau where $\lambda_1$ and $\lambda_2$ are close. Stars indicate frequencies kept in the extended set $\Omega_e(K_i)$.

**9. Conclusion.** We have studied and tested a nonlinear mathematical programming approach to the mixed $H_2/H_\infty$ controller synthesis problem. The importance of this problem was recognized in the late 1980s, but approaches based on AREs could not be brought to work satisfactory. It is possible to characterize the optimal $H_2/H_\infty$-controller by way of the Q-parameterization, but as soon as the controller has to satisfy additional structural constraints, like for instance reduced order $n_K < n_x$, an analytic solution does not exist. In that situation convexity methods based on

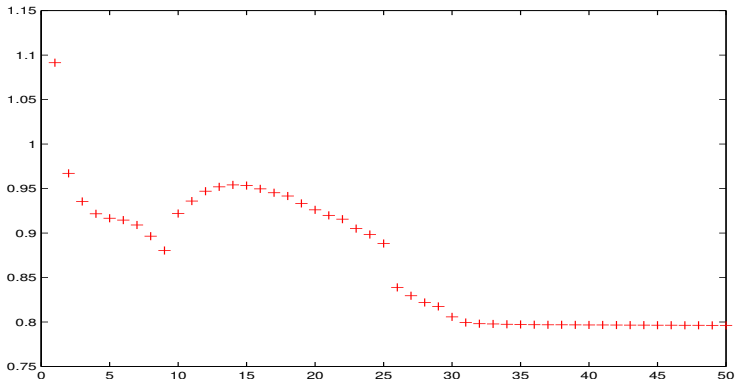


FIG. 8.1. *Example 'BDT2' - $H_2$ norm during the first 50 iterations*

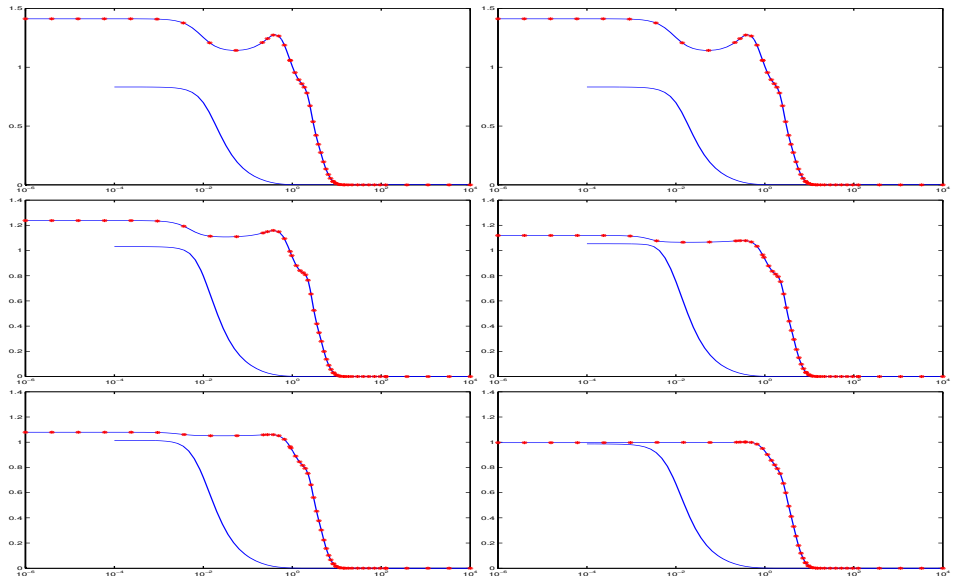

FIG. 8.2. *Example 'BDT2' - Largest and second largest eigenvalues versus frequency in logarithmic scale - first 6 iterations. Observe that the second eigenvalue $\lambda_2$ is strictly below the first one on the range $\omega \leq 10^4$, except for the bottom right plot, where coalescence on a low frequency band seems to occur.*

LMIs and AREs are then no longer suitable, and finding the globally optimal solution is known to be $NP$-hard. In consequence, we propose a strategy based on local optimization, which comes with a weaker certificate, but has the benefit to work in practice. The problem being nonconvex, nonsmooth and semi-infinite, we have developed a nonsmooth constrained programming technique suited for the $H_2/H_\infty$ problem and other programs of a similar structure. The new method has been tested on several benchmark studies and shown to perform better than existing methods.

**Acknowledgement.** The present work was supported by research grants from *Fondation d'entreprise EADS* under contract *Solving challenging problems in feedback control*, from the *Agence Nationale de Recherche* (ANR) under contract *Guidage*, and from ANR under contract *Controvert*.

REFERENCES


[1] P. Apkarian, V. Bompart, and D. Noll. Control design in the time and frequency domain using nonsmooth techniques. *Systems and Control Letters*, accepted 2007. Published electronically october 25, 2007.

[2] P. Apkarian, V. Bompart, and D. Noll. Nonsmooth structured control design with application to PID loop-shaping of a process. *International Journal of Robust and Nonlinear Control*, 17(14):1320–1342, 2007.

[3] P. Apkarian and D. Noll. Controller design via nonsmooth multidirectional search. *SIAM J. Control Optim.*, 44(6):1923–1949 (electronic), 2006.

[4] P. Apkarian and D. Noll. IQC analysis and synthesis via nonsmooth optimization. *Systems and Control Letters*, 55(12):971–981, 2006.

[5] P. Apkarian and D. Noll. Nonsmooth $H_\infty$ synthesis. *IEEE Trans. Automat. Control*, 51(1):71–86, 2006.

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

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

[8] P. Apkarian, D. Noll, and O. Prot. A trust region spectral bundle method for nonconvex maximum eigenvalue functions. *SIAM J. on Optimization*, accepted 2007.

[9] P. Apkarian, D. Noll, and O. Prot. Nonsmooth methods for analysis and synthesis with integral quadratic constraints. In *Proceedings of IEEE Conference on Decision and Control*, New Orleans, December 2007.

[10] D. Arzelier and D. Peaucelle. An iterative method for mixed $H_2/H_\infty$ synthesis via static output-feedback. In *Proceedings of IEEE Conference on Decision and Control*, pages 3464–3469, December 2002.

[11] V. Bompart, D. Noll, and P. Apkarian. Second-order nonsmooth optimization for feedback control. *Numerische Mathematik*, 107(3):433–454, 2007.

[12] 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. *Systems and Control Letters*, 15:1–7, 1990.

[13] S. Boyd, L. Elghaoui, E. Feron and V. Balakrishnan. Linear matrix inequalities in system and control theory. SIAM Studies in Applied Mathematics. SIAM, Philadelphia, 1994.

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

[15] J. Camino, D. Zampieri, and P. Peres. Design of a vehicular suspension controller by static output feedback. *Proceedings of American Control Conference*, pages 3168–3172, 1999.

[16] R.H. Cannon and D.E. Rosenthal. Experiments in control of flexible structures with noncolocated sensors and actuators. *Journal of Guidance, Control and Dynamics*, 7(5):546–553, 1984.

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

[18] R. Correa, and C. Lemaréchal. Convergence of some algorithms for convex minimization. *Math. Programming*, 62(2):261–275, 1993.

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

[20] S.M. Djouadi, C.D. Charalambous, and D.W. Repperger. On multiobjective $H_2/H_\infty$ optimal control. *Proc. Amer. Control Conf.*, 2001, Arlington, VA, USA.

[21] J.C Doyle, K. Glover, P. Khargonekar and B. A. Francis. State-space solutions to standard $H_2$ and $H_\infty$ control problems. *IEEE Trans. Automat. Control*, 34(8):831–847, 1989.

[22] J.C. Doyle, K. Zhou, and B. Bodenheimer. Optimal control with mixed $H_2$ and $H_\infty$ performance objectives. In *Proceedings of the American Control Conference*, volume 3, pages 2065–2070, 1989.

[23] J.C. Doyle, K. Zhou, K. Glover, and B. Bodenheimer. Mixed $H_2$ and $H_\infty$ performance objectives. II. Optimal control. *IEEE Trans. Automat. Control*, 39(8):1575–1587, 1994.

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

[25] J.C. Geromel, P.L.D. Peres, and S.R. Ouza. A convex approach to the mixed $H_2/H_\infty$ control problem for discrete-time uncertain systems. *SIAM J. Control Optim.*, 33(6):1816–1833, 1995.

[26] K. Glover, All optimal Hankel-norm approximations of linear multivariable systems and their $L_\infty$-error bounds. *International Journal of Control*, vol. 39:1115–1193, 1984.

[27] W. M. Haddad, D. S. Bernstein. LQG control with a $H_\infty$ performance bound: a Riccati equation approach. *IEEE Trans. Aut. Control*, AC-34(3):293–305, march 1989.

[28] C. Helmberg and F. Rendl. Spectral bundle method for semidefinite programming. *SIAM J. Optimization*, 10(3):673 – 696, 2000.

[29] H.A. Hindi, B. Hassibi, and S. Boyd. Multiobjective $H_2/H_\infty$-optimal control via finite dimensional Q-parametrization and linear matrix inequalities. *Proceedings of the American Control Conference*, pages 3244–3248, 1998.

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

[31] M. Kočvara and M. Stingl. PENNON: a code for convex nonlinear and semidefinite programming. *Optimization Methods and Software*, 18(3):317–333, 2003.

[32] M. Kočvara and M. Stingl. Solving nonconvex SDP problems of structural optimization with stability control. *Optimization Methods and Software*, 19(5):595–609, 2004.

[33] F. Leibfritz. A LMI-based algorithm for designing suboptimal static $H_2/H_\infty$ output feedback controllers. *SIAM J. Control Optim.*, 39(6):1711–1735 (electronic), 2001.

[34] F. Leibfritz. COMP$L_e$IB, COnstraint Matrix-optimization Problem LIbrary - a collection of test examples for nonlinear semidefinite programs, control system design and related problems. Technical report, Universität Trier, 2003.

[35] D.J.N. Limbeer, B.D.O. Andersen, and B. Hendel. A Nash game approach to mixed $H_2/H_\infty$ control. *IEEE Trans. Aut. Control*, 39:69–82, 1994.

[36] D. Noll. Local convergence of an augmented Lagrangian method for matrix inequality constrained programming. *Optimization, Methods and Software*, 22(5):777–802, 2007.

[37] D. Noll and P. Apkarian. Spectral bundle methods for nonconvex maximum eigenvalue functions: first-order methods. *Mathematical Programming Series B*, 104(2-3):701–727, 2005.

[38] D. Noll and P. Apkarian. Spectral bundle methods for nonconvex maximum eigenvalue functions: second-order methods. *Mathematical Programming Series B*, 104(2-3):729–747, 2005.

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

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

[41] E. Polak. *Optimization: Algorithms and Consistent Approximations*, volume 124 of *Applied Mathematical Sciences*. Springer Verlag, New York, 1997.

[42] T. Rautert and E.W. Sachs. Computational design of optimal output feedback controllers. *SIAM J. on Optimization*, 7(3):837–852, 1997.

[43] C. Sagastizábal, M. Solodov. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or filter. *SIAM J. on Optimization*, 16(1):146–169, 2005.

[44] C. Scherer, P. Gahinet, and M. Chilali. Multi-objective output-feedback control via LMI optimization. *IEEE Trans. Aut. Control*, 42:896–911, 1997.

[45] C.W. Scherer. Multiobjective $H_2/H_\infty$ control. *IEEE Trans. Aut. Control*, 40(6):1054–1062, 1995.

[46] C.W. Scherer. Mixed $H_2/H_\infty$ control. A. Isidori, Ed., Trends in Control, A European Perspective, Springer Verlag, Berlin (1995) 173-216.

[47] C.W. Scherer. Lower bounds in multi-objective $H_2/H_\infty$ problems. *Proceedings of the 38th IEEE Conference on Decision and Control*, 1999.

[48] C.W. Scherer. An efficient solution to multi-objective control problems with LMI objectives. *Systems and Control Letters*, 40(1):43–57, 2000.

[49] J.-B. Thevenet, D. Noll, and P. Apkarian. Non linear spectral SDP method for BMI constrained problems: Applications to control design. *ICINCO (1)*:237–248, 2004.

[50] J. Yu. A new static output feedback approach to the suboptimal mixed $H_2/H_\infty$ problem. *Internat. J. Robust Nonlinear Control*, 14(12):1023–1034, 2004.

[51] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, NJ, USA, 1996.