Nonsmooth techniques for stabilizing linear systems

Vincent Bompart[*], Pierre Apkarian[†], Dominikus Noll[‡]

September 19, 2006

Abstract

We discuss closed-loop stabilization of linear time-invariant dynamical systems, a problems which frequently arises in controller synthesis, either as a stand-alone task, or to initialize algorithms for H_∞ synthesis or related problems. Classical stabilization methods based on Lyapunov or Riccati equations appear to be inefficient for large systems. Recently, non-smooth optimization methods like gradient sampling [19] have been successfully used to minimize the spectral abscissa of the closed-loop state matrix (the largest real part of its eigenvalues) to solve the stabilization problem. These methods have to address the non-smooth and even non-Lipschitz character of the spectral abscissa function. In this work, we develop an alternative non-smooth technique for solving similar problems, with the option to incorporate second-order elements to speed-up convergence to local minima. Using several case studies, the proposed technique is compared to more conventional approaches including direct search methods and techniques where the spectral abscissa minimization problem is recast as a traditional smooth non-linear mathematical programming problem.

1 Introduction and notations

Internal stability is certainly the most fundamental design specification in linear control. Necessary and sufficient conditions for output feedback stabilizability are still not known [6]. From an algorithmic point of view, the problem is clearly in the class NP and conjectured to be NP-hard (see [7]). Output feedback stabilizability can be checked via more general decidability algorithms, like the Tarski and Seidenberg reduction methods [1], but the computational cost is exponential in the problem size and becomes rapidly prohibitive as the number of controller parameters increases.

Internal stability could also be expressed in terms of a bilinear matrix inequality (BMI), but this leads to similar problems. Solving BMIs globally is algorithmically difficult, and introducing Lyapunov variables to symmetrize the problem further increases the number of decision parameters and often leads to numerical ill-conditioning.

A less ambitious line is to address internal stability as a local optimization problem. Recent approaches using non-smooth optimization techniques are [19, 11] for stabilization, and [3, 4, 5, 8] for H_∞ synthesis. In [11] for instance the authors propose to optimize the spectral abscissa of the closed-loop matrix via specific non-smooth techniques until a strictly negative value is obtained. If the method converges to a local minimum with positive value, indicating failure to solve the control problem, the method has to be restarted from a different initial guess. Our present contribution is also a local optimization technique, but our method to generate descent steps is new. In particular, in contrast with [11], our approach is deterministic. We believe that while local optimization techniques do not provide the strong certificates of global techniques, they offer better chances in practice to solve the stability problem.

The paper is organized as follows. In section 2 we recall differentiability properties of the spectral abscissa and discuss strategies to minimize it. In section 3 we compute sub-gradients of the closed-loop spectral abscissa, and specify to cases of practical interest with structured controllers like PID or others. The computation of descents steps is explained in section 4. It leads to the presentation of the

[*]Université Paul Sabatier, Institut de Mathématiques, 118 route de Narbonne, 31062 Toulouse, France, and ONERA-CERT, 2 av. Edouard Belin, 31055 Toulouse, France, e-mail: *bompart@cert.fr*

[†]ONERA-CERT, 2 av. Edouard Belin, 31055 Toulouse, France, and Université Paul Sabatier, Institut de Mathématiques, 118 route de Narbonne, 31062 Toulouse, France, e-mail: *apkarian@cert.fr*

[‡]Université Paul Sabatier, Institut de Mathématiques, 118 route de Narbonne, 31062 Toulouse, France, e-mail: *noll@mip.ups-tlse.fr*

algorithm and its second order variant in section 5. Finally, section 6 presents numerical experiments on miscellaneous automatic control applications, with some examples involving structured controllers. For some of them, the final spectral properties of the state matrix are analyzed. Our method is compared to other design methods based on a) continuous smooth optimization, b) direct search algorithm, and c) probabilistic elements. In particular, two Matlab Optimization Toolbox functions are tested.

Matrix notations

In the following $(\mathbb{R}^{n\times n}, \langle\cdot,\cdot\rangle)$ is the Hilbert space of real $n \times n$ matrices with inner product $\langle M, N\rangle = \text{tr}\, M^T N$. The induced Hilbert space norm is the Frobenius norm and noted $\|\cdot\|$. $\mathbb{S}^n$ is the linear subspace of $n \times n$ real symmetric matrices.

For $M \in \mathbb{R}^{n\times n}$, $M \succeq 0$ means that $M \in \mathbb{S}^n$ is positive semi-definite. The n eigenvalues of $M \in \mathbb{R}^{n\times n}$ (repeated with multiplicity) are denoted $\lambda_1(M), \ldots, \lambda_n(M)$ and ordered lexicographically, that is by decreasing real part first, and next by decreasing imaginary part in case of equal real parts. When the eigenvalues of M are considered without their multiplicities, they are denoted $\mu_1(M), \ldots, \mu_q(M)$ and ordered in the same manner as $(\lambda_i(M))$, with respective multiplicities $n_1, \ldots, n_q$ in the characteristic polynomial $p_M(\lambda) = \det(M - \lambda I_n)$ (algebraic multiplicities, with $n_1 + \cdots + n_q = n$). We say that an eigenvalue $\mu_j(M)$ is semisimple if its algebraic multiplicity n_j coincides with the dimension of the associated eigenspace (geometric multiplicity, denoted by p_j; $1 \le p_j \le n_j$ in the general case). Otherwise, $\mu_j(M)$ is said defective.

In the sequel, $\alpha(M)$ denotes the spectral abscissa of M, defined as

$$\alpha(M) = \max_{1\le j\le q} \text{Re}\,(\mu_j\,(M))\,.$$

Any eigenvalue of M whose real part attains $\alpha(M)$ is said to be active. The set of all active eigenvalues of M is denoted $\mathcal{A}(M) = \{\mu_j\,(M) \mid \text{Re}\,(\mu_j\,(M)) = \alpha(M)\}$, the corresponding active indices sets are

$$\mathcal{I}(M) = \{i \in \mathbb{N} \mid 1 \le i \le n \text{ and } \lambda_i(M) \in \mathcal{A}(M)\}\,,$$

and without multiplicity

$$\mathcal{J}(M) = \{j \in \mathbb{N} \mid 1 \le j \le q \text{ and } \mu_j(M) \in \mathcal{A}(M)\}\,.$$

Plant and controller notations

The open-loop system we wish to stabilize is a continuous linear time-invariant plant, described by the state-space equations

$$P(s): \quad \begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$and $C \in \mathbb{R}^{p\times n}$. We assume without loss of generality that there is no direct transmission between u and y. We consider static or dynamic output feedback control laws of the form $u = K(s)y$ in order to stabilize (1) internally, that is, to place all the eigenvalues of the closed-loop state matrix in the open left half-plane. We suppose that the controller order $k \in \mathbb{N}$ is fixed.

In the case of a static feedback ($k = 0$), the controller is denoted by $K \in \mathbb{R}^{m\times p}$. For dynamic controllers defined by the state-space equations

$$K(s): \quad \begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \begin{bmatrix} x_K \\ y \end{bmatrix} \tag{2}$$

we use the following standard substitutions in order to reduce to the static feedback case:

$$\begin{array}{llll} K \to \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}, & A \to \begin{bmatrix} A & 0 \\ 0 & 0_k \end{bmatrix}, & B \to \begin{bmatrix} 0 & B \\ I_k & 0 \end{bmatrix}, & C \to \begin{bmatrix} 0 & I_k \\ C & 0 \end{bmatrix}, \\ k \to 0, & n \to n+k, & m \to m+k, & p \to m+p. \end{array} \tag{3}$$

The affine mapping $K \mapsto A + BKC$ is denoted as A_c.

2 Minimizing the spectral abscissa

We start by writing the stabilization problem as an unconstrained optimization program

$$\min_{K\in\mathcal{K}} \alpha(A+BKC) \tag{4}$$

where the search space $\mathcal{K}$ is either the whole controller space $\mathbb{R}^{m\times p}$, or a subset of $\mathbb{R}^{m\times p}$ in those cases where a stabilizing controller with a fixed structure is sought.

Closed-loop stability is satisfied as soon as $\alpha(A+BKC) < 0$, so that the minimization process can be stopped before convergence. In fact, minimization of the spectral abscissa below zero can be interpreted as maximizing the asymptotic decay rate of the closed-loop system [9]. Convergence to a local minimum is important only in those cases where the method fails to locate negative values $\alpha < 0$. If the process converges toward a local minimum K^* with positive value $\alpha \geq 0$, we know at least that the situation cannot be improved in a neighborhood of K^*, and that a restart away from the local minimum is inevitable.

Program (4) is difficult to solve for two reasons. Firstly, the minimax formulation calls for nonsmooth optimization techniques, but more severely, the spectral abscissa $M \mapsto \alpha(M)$ as a function $\mathbb{R}^{n\times n} \to \mathbb{R}$ is not even locally Lipschitz everywhere. The variational properties of α have been analyzed by Burke and Overton [14]. In [13] the authors show that if the active eigenvalues of M are all semisimple, α is directionally differentiable at M and admits a Clarke subdifferential $\partial\alpha(M)$. This property fails in the presence of a defective eigenvalue in the active set $\mathcal{A}(M)$ associated with a non-trivial Jordan block in the Jordan form of M.

Despite the lack of Clarke sub-differentiability of the function α, we consider program (4) as practically useful, if the following facts are taken into account:

- Starting our nonsmooth optimization from a closed-loop matrix $A+BK_0C$ with semisimple active eigenvalues, we expect that iterates $A+BK_lC$ will remain semisimple. Non-derogatory Jordan blocks will only occur at the limit point $A+BK^*C$. If a negative value $\alpha(A+BK_lC) < 0$ is obtained, we can stop the procedure at this iterate, and the question of optimality is irrelevant.

- On the other hand, if iterates are not stabilizing, $\alpha(A+BK_lC) \geq 0$ for all l, then we need to check whether the limit point $A+BK^*C$ is optimal or not. This requires a non-smooth stopping test. If the limit point is locally optimal, we know that a restart of the method away from K^* is inevitable. On the other hand, if K^* is a so-called *dead point*, that is, a limit point which is not locally optimal, then we should not do a restart, but use non-smooth decent techniques and keep optimizing until a local minimum is found.

Several strategies for addressing the nonsmoothness in (5) have been put forward: Burke, Lewis and Overton have extended the idea of gradient bundle methods (see [18] for the convex case and [21] for the Lipschitz continuous case) to certain non-Lipschitz functions, for which the gradient is defined, continuous and computable almost everywhere. The resulting algorithm, called gradient sampling algorithm, is presented in [11] (in the stabilization context) and analyzed in [10, 12] with convergence results. The outcome of this research is a package HIFOO, which will be included in our tests, see Section 6. The search direction in HIFOO depends on randomly sampled points which gives the algorithm a non-deterministic aspect. In contrast, Apkarian and Noll propose a composite approach in [2], combining a direct search method and a nonsmooth strategy for descent and stopping test. By construction, direct search algorithms require a lot of functions evaluations, which is demanding if the number of controller decision variable is large. In response, the present paper proposes an algorithm which uses exclusively non-smooth descent steps, allowing to avoid the drawbacks of both approaches.

3 Sub-gradients of the spectral abscissa

3.1 Sub-gradients in state-space

In this section, we suppose that all active eigenvalues of the closed-loop state matrix $A_c(K)$ are semisimple, with $r = |\mathcal{J}(K)| < q$ distinct active eigenvalues (or $s = |\mathcal{I}(\mathcal{K})| < n$ if counted with their multiplicity). We have $\mathcal{A}(K) = \{\mu_1(A_c(K)), \ldots, \mu_r(A_c(K))\} = \{\lambda_1(A_c(K)), \ldots, \lambda_s(A_c(K))\}$. The Jordan form $J(K)$ of $A_c(K)$ is then partly diagonal, more precisely :

$$J(K) = V(K)^{-1} A_c(K) V(K) = \begin{bmatrix} D(K) & & & \\ & J_{r+1}(K) & & \\ & & \ddots & \\ & & & J_q(K) \end{bmatrix}$$

- $D(K) = \text{diag}\ \left[\lambda_1\left(A_c(K)\right), \ldots, \lambda_s\left(A_c(K)\right)\right]$ is the diagonal part of active eigenvalues,
- $J_j(K)$, for $r < j \leq q$ are $n_j \times n_j$ block-diagonal blocks, such as

$$J_j(K) = \begin{bmatrix} J_{j,1}(K) & & \\ & \ddots & \\ & & J_{j,p_j}(K) \end{bmatrix}, \quad \text{with } J_{j,k} = \begin{bmatrix} \mu_j & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \mu_j \end{bmatrix},$$

- $V(K) = \left[v_1\left(A_c(K)\right), \ldots, v_n\left(A_c(K)\right)\right]$, where the first s columns are right eigenvectors of $A_c(K)$ associated with the active eigenvalues (the other columns are generalized right eigenvectors),
- $V(K)^{-1} = \begin{bmatrix} u_1\left(A_c(K)\right)^H \\ \vdots \\ u_n\left(A_c(K)\right)^H \end{bmatrix}$,where the first s rows are left eigenvectors of $A_c(K)$ associated with the active eigenvalues (the other rows are generalized left eigenvectors).

We define $U(K) = V(K)^{-H}$, and for $1 \leq j \leq r$, $V_j(K)$ (resp. $U_j(K)$) the $n \times n_j$ block from $V(K)$ (resp. from $U(K)$) composed of the right eigenvectors (resp. of the transpose conjugate of the left eigenvectors) associated with μ_j.

The function $\alpha \circ A_c$ is Clarke regular at K, as a composition of the affine mapping A_c with α, which is locally Lipschitz continuous at K(ref ?). Let $\mu_j \in \mathcal{A}\left(A_c\left(K\right)\right)$ be an active eigenvalue of $A_c(K)$, such that $\mu_j = \lambda_i\left(A_c\left(K\right)\right) = \cdots = \lambda_{i+n_j}\left(A_c\left(K\right)\right)$. Then the real matrix

$$\phi_j(K) = \text{Re}\ \left(B^T U_j Y_j V_j^H C^T\right) = \text{Re}\ \left(C V_j Y_j U_j^H B\right)^H = \left(\text{Re}\ \left(C V_j Y_j U_j^H B\right)\right)^T$$

is a Clarke subgradient of the composite function $\alpha \circ A_c$ at K, where $Y_j \succeq 0$ and $\text{Tr}(Y_j) = 1$. Moreover, the whole subdifferential $\partial(\alpha \circ A_c)(K)$ is described by matrices of the form

$$\phi(K) = \sum_{j \in \mathcal{J}(A_c(K))} \sum_{\substack{Y_j \succeq 0 \\ \text{Tr}(Y_j) = 1}} \left(\text{Re}\ \left(C V_j Y_j U_j^H B\right)\right)^T \tag{5}$$

Notice that every pair of complex conjugate active eigenvalues μ_j and $\mu_k = \bar{\mu}_j$ $(k \neq j)$ share the same closed-loop spectral abscissa subgradient $\phi_j = \phi_k$. The subdifferential is then kept unchanged if the active set only contains the active eigenvalues, whose imaginary part is nonnegative: $\mathcal{A}(M) = \{\mu_j\left(M\right) \mid \text{Re}\left(\mu_j\left(M\right)\right) = \alpha(M) \text{ and } \text{Im}\left(\mu_j\left(M\right)\right) \geq 0\}$.

Remark: If the open-loop plant is not controllable, then every uncontrollable mode $\mu_l(A)$ persists in the closed-loop (for all controllers K): there exists j such that $\mu_l(A) = \mu_j(A_c(K))$. Moreover, if this eigenvalue is semisimple and active for $\alpha \circ A_c$, the associated subgradients are null, because $U_j^H B = 0$. The case of unobservable modes leads to the same conclusion, because $C V_j = 0$. In this way, whenever an uncontrollable or unobservable open-loop mode $\mu_l(A)$ becomes active for the closed-loop spectral abscissa, we get $0 \in \partial(\alpha \circ A_c)(K)$ and then we have local optimality of K. Moreover, the optimality is global because $\text{Re}\ \mu_l(A)$ is a lower bound for $\alpha \circ A_c$.

3.2 Subgradients and dynamic controllers

The problem of stabilizing the plant by a dynamic output feedback reduces formally to the static case, with the substitutions (3). Nevertheless, the dynamic case is slightly more tricky, because the matrices A_K, B_K, C_K and D_K have to define a minimal controller realization, both at the initialization stage and at every subsequent iteration of the algorithm.

As an illustration, if the k-th order (non-minimal) realization of the initial controller is chosen with the following structure (neither observable nor controllable)

$$K(s): \quad \left[\begin{array}{cc} A_K & 0 \\ 0 & D_K \end{array}\right],$$

with $\alpha(A_K) < \alpha(A+BD_KC)$, it is straightforward to show that the resulting closed-loop spectral abscissa subgradients are convex linear combinations of matrices of the form

$$\phi_j(K) = \left[\begin{array}{cc} 0 & 0 \\ 0 & \mathrm{Re}\left(CV_jY_jU_j^HB\right)^T \end{array}\right]$$

where V_j (resp. U_j^H) are blocks of right (resp. left) eigenvectors associated with the active eigenvalues of $A + BD_KC$, and $Y_j \succeq 0$, $\mathrm{Tr}(Y_j) = 1$. As the successive search directions have the same structure (see (12) in the following), this results in unchanged A_K, B_K, C_K blocks among the new iterates. Put differently, they all represent static controllers.

In order to initialize the descent algorithm with a minimal k-th order controller, and to maintain this minimality for all subsequent iterates, we use an explicit parametrization of minimal, stable and balanced systems [26, 15]. With this canonical structure, the number of free parameters for a k-th order controller with p inputs and m outputs is $k(m+p)+mp$.

3.3 Sub-gradients with structured controllers

Formulation (4) is general enough to handle state-space structured controllers, such as decentralized or PID controllers, minimal realizations (see 3.2), and others. Let $K : \mathbb{R}^d \longrightarrow \mathbb{R}^{m\times p}$ be a smooth parametrization of an open subset $\mathcal{K} \subset \mathbb{R}^{m\times p}$, containing state-space realizations of a family of controllers of a given structure. Then the stabilization problem can be written as

$$\min_{X\in\mathbb{R}^d} \alpha\left(A_c \circ K\left(X\right)\right)$$

with d free decision variables. The Clarke subgradients $\psi \in \mathbb{R}^d$ of the composite function $\alpha \circ A_c \circ K$ are derived from (5) with the chain rule (ref Clarke ?)

$$\psi(X) = J_{\mathrm{vec}(K)}(X)^T \mathrm{vec}\left(\phi\left(K\left(X\right)\right)\right)$$

where $J_{\mathrm{vec}(K)}(X) \in \mathbb{R}^{mp\times d}$ is the Jacobian matrix of $\mathrm{vec}(K) : X \in \mathbb{R}^d \mapsto \mathrm{vec}(K(X)) \in \mathbb{R}^{mp}$.

BLO donnent une regle en chaine pour la fonction alpha qui n'est pas basee sur la theorie de Clarke, sachant que celle ne marche pas toujours pour alpha. C'est dans le but de demontrer une condition d'optimalite. Il faudra chercher la ref.

4 Descent step and optimality function

In order to derive a descent step from the subdifferential $\partial(\alpha \circ A_c)(K)$, we follow a first-order step generation mechanism for minimax problems introduced by Polak in [27, 28]. It was described and applied in the semi-infinite context of the H_∞ synthesis in [3]. This descent scheme is based on the minimization of a local and strictly convex first-order model $\theta(K)$, which serves both as a descent step generator and as an optimality function.

In order to define $\theta(K)$, we first make the strong assumption that all the eigenvalues of the closed-loop state matrix $A_c(K)$ are semisimple. Then, with the notations from the introduction and in the previous section, we define

$$\theta(K) = \min_{H\in\mathcal{K}} \max_{1\le j\le q} \max_{\substack{Y_j \succeq 0 \\ \mathrm{Tr}(Y_j)=1}} \mathrm{Re}\left(\mu_j\left(A_c\left(K\right)\right)\right) - \alpha\left(A_c\left(K\right)\right) + \langle \phi_j(K), H\rangle + \frac{1}{2}\delta\left\|H\right\|^2 \qquad (6)$$

where $\delta > 0$ is fixed, and $\langle \phi_j(K), H\rangle = \mathrm{Tr}\left(\mathrm{Re}\left(CV_jY_jU_j^HB\right)H\right)$.

Using Fenchel duality for permuting the min and double max operators, we obtain the dual form of (6), where the step H in the controller space vanishes

$$\theta(K) = \max_{\substack{\tau_j \ge 0 \\ \sum_j \tau_j = 1}} \max_{\substack{Y_j \succeq 0 \\ \mathrm{Tr}(Y_j)=1}} \sum_{j=1}^q \tau_j \left[\mathrm{Re}\left(\mu_j\left(A_c\left(K\right)\right)\right) - \alpha\left(A_c\left(K\right)\right)\right] - \frac{1}{2\delta}\left\|\sum_{j=1}^q \tau_j\phi_j(K)\right\|^2 \qquad (7)$$

and we get the minimizer $H(K)$ of the primal formulation (6) from the solution

$$\left(\left(\tau_j^\star(K)\right)_{1\le j\le q}, \left(Y_j^\star(K)\right)_{1\le j\le q}\right)$$

of the dual expression (7) in the explicit form

$$H(K) = -\frac{1}{\delta}\sum_{j=1}^{q} \tau_j^\star(K)\ \mathrm{Re}\left(CV_jY_j^\star(K)U_j^HB\right)^T. \tag{8}$$

We recall the following basic properties of θ and H from [27]:

1. $\theta(K) \le 0$ for all $K \in \mathcal{K}$, and $\theta(K) = 0$ if and only if $0 \in \partial(\alpha \circ A_c)(K)$.
2. When $0 \notin \partial(\alpha \circ A_c)(K)$, then $H(K)$ is a descent direction for the closed-loop spectral abscissa at K. More precisely
$$d\left(\alpha \circ A_c\right)(K; H(K)) \le \theta(K) - \frac{1}{2}\delta \left\|H(K)\right\|^2 \le \theta(K)$$
for all K.
3. The function θ is continuous,
4. The operator $K \mapsto H(K)$ is continuous.

Therefore direction $H(K)$ will be chosen as a search direction in a descent-type algorithm by combining it with a line search. The continuity of $H(\cdot)$ ensures that every accumulation point $\bar{K}$ in the sequence of iterates satisfies the necessary optimality condition $0 \in \partial(\alpha \circ A_c)(\bar{K})$ (see [3]). It is not clear whether continuity in this sense is still satisfied when the hypothesis of semi-simplicity of the eigenvalues is dropped. Notice that even for semisimple eigenvalues, continuity fails for the steepest descent direction, defined as the solution of the program

$$\min_{\|H\|\le 1} d(\alpha \circ A_c)(K; H) = \min_{\|H\|\le 1} \max_{j\in\mathcal{J}(A_c(K))} \max_{\substack{Y_j \succeq 0 \\ \mathrm{Tr}(Y_j)=1}} \langle \phi_j(K), H\rangle.$$

This is why steepest descent steps for non-smooth functions may fail to converge. In our case this justifies the recourse to the quadratic, first-order model θ as a descent function. Moreover, the properties 1. and 3. suggest a stopping test based on the value of $\theta(K)$, because as soon as $\theta(K) \ge -\varepsilon_\theta$ (for a small given $\varepsilon_\theta > 0$), the controller K is in a neighborhood of a stationary point.

5 Descent algorithms

5.1 Variant I (first-order type)

We discuss details of a descent-type algorithm for minimizing the closed-loop spectral abscissa, based on the theoretical results from the previous section.

For a given iterate K_l, we have to address first the practical computation of the maximizer of the dual form (7) of $\theta(K_l)$. Without any additional hypothesis, it is a concave semidefinite program in τ_j and Y_j. The matrix variables come from the possible multiplicity of some eigenvalues of $A_c(K_l)$. Fortunately, this situation is unlikely to happen in a numerical framework, because the eigenvalues and eigenvectors are computed within a given tolerance. Except in some academic examples (see for instance 6.1), or for particular values of the controller, coalescence of eigenvalues will not be observed in practice. The SDP then reduces to a concave quadratic maximization program:

$$\theta(K) = \max_{\substack{\tau_j \ge 0 \\ \sum_j \tau_j = 1}} -\alpha\left(A_c\left(K\right)\right) + \sum_{j=1}^{n} \tau_j \mathrm{Re}\left(\mu_j\left(A_c\left(K\right)\right)\right) - \frac{1}{2\delta}\left\|\sum_{j=1}^{n}\tau_j\phi_j(K)\right\|^2 \tag{9}$$

where $\phi_j(K) = \mathrm{Re}\left(Cv_ju_j^HB\right)^T$, with v_j a right eigenvector associated with $\mu_j\left(A_c\left(K\right)\right)$, and u_j^H the corresponding left eigenvector (such that $u_j^Hv_j = 1$).

To go one step further, we reduce the dimension of the search space. There are n scalar variables τ_j in (9), n the order of the open-loop plant, augmented by k, the order of the controller, in the case of a dynamic controller. For a given ratio $\rho \in [0,1]$, we define the following enriched active eigenvalues set

$$\mathcal{A}_\rho(K) = \left\{ \mu_j\left(A_c(K)\right) \mid \alpha\left(A_c\left(K\right)\right) - \mathrm{Re}\left(\mu_j\left(A_c(K)\right)\right) \leq \rho \left[\alpha\left(A_c\left(K\right)\right) - \min_{1\leq i\leq n} \mathrm{Re}\left(\mu_i\left(A_c(K)\right)\right) \right] \right\}$$

where $\mathcal{J}_\rho(K) = \{j \in \mathbb{N} \mid 1 \leq j \leq n \text{ and } \mu_j\left(A_c(K)\right) \in \mathcal{A}_\rho(K)\}$ is the corresponding enriched active index set. It is clear that $\rho \mapsto \mathcal{A}_\rho(K)$ is nondecreasing on $[0,1]$, and that $\mathcal{A}(K) = \mathcal{A}_0(K) \subset \mathcal{A}_\rho(K) \subset \mathcal{A}_1(K) = \text{spec}\left(A_c(K)\right)$ for all $\rho \in [0,1]$. Hence, we have locally

$$\alpha\left(A_c\left(K\right)\right) = \max_{j\in\mathcal{J}_\rho(K)} \mathrm{Re}\left(\mu_j\left(A_c(K)\right)\right) \tag{10}$$

and by applying the descent function θ to this local formulation, we get finally the quadratic program

$$\theta(K) = \max_{\substack{\tau_j \geq 0 \\ \sum_j \tau_j = 1}} -\alpha\left(A_c\left(K\right)\right) + \sum_{j=1}^{|\mathcal{J}_\rho(K)|} \tau_j \mathrm{Re}\left(\mu_j\left(A_c\left(K\right)\right)\right) - \frac{1}{2\delta} \left\| \sum_{j=1}^{|\mathcal{J}_\rho(K)|} \tau_j \phi_j(K) \right\|^2 . \tag{11}$$

The descent direction $H(K)$ is obtained from the maximizer $\left(\tau_j^\star(K)\right)_{1\leq j\leq|\mathcal{J}_\rho(K)|}$ as

$$H(K) = -\frac{1}{\delta} \sum_{j=1}^{|\mathcal{J}_\rho(K)|} \tau_j^\star(K)\ \mathrm{Re}\left(C v_j u_j^H B\right)^T \tag{12}$$

Notice that for $\rho = 0$ the QP in (11) reduces to the steepest descent finding problem

$$\min_{\substack{\tau_j \geq 0 \\ \sum_j \tau_j = 1}} \left\| \sum_{j=1}^{|\mathcal{J}(K)|} \tau_j \phi_j(K) \right\| = \min_{\phi\in\partial(\alpha\circ A_c)(K)} \|\phi\|$$

while $\rho = 1$ reproduces (9). The parameter ρ offers some additional numerical flexibility.

Algorithm 1 First-order descent type algorithm for the closed-loop spectral abscissa

Set the parameters $\rho \in [0,1]$, $\delta > 0$, $K_0 \in \mathcal{K}$, ε_θ, ε_α, $\varepsilon_K > 0$, $\beta \in]0,1[$.
Set the counter $l \leftarrow 0$.

1. Compute $\alpha\left(A_c\left(K_0\right)\right)$, the enriched active index set $\mathcal{J}_\rho(K_0)$ and the corresponding subgradients $\phi_j(K_0)$.
2. Solve (11) for $K = K_l$ and get the search direction $H(K_l)$ from (12).
   If $\theta(K_l) \geq -\varepsilon_\theta$then stop.
3. Find a step length $t_l > 0$ satisfying the Armijo line search condition

$$\alpha\left(A_c\left(K_l + t_l H(K_l)\right)\right) \leq \alpha\left(A_c\left(K_l\right)\right) + \beta t_l d(\alpha\circ A_c)\left(K_l; H(K_l)\right)$$

4. Set $K_{l+1} \leftarrow K_l + t_l H(K_l)$.
   Compute $\alpha\left(A_c\left(K_{l+1}\right)\right)$, the enriched active index set $\mathcal{J}_\rho(K_{l+1})$ and the corresponding subgradients $\phi_j(K_{l+1})$.
5. If $\alpha\left(A_c\left(K_l\right)\right) - \alpha\left(A_c\left(K_{l+1}\right)\right) \leq \varepsilon_\alpha(1 + \alpha\left(A_c\left(K_l\right)\right))$ and $\|K_l - K_{l+1}\| \leq \varepsilon_K\left(1 + \|K_l\|\right)$
   then stop.
   Otherwise set $l \leftarrow l + 1$ and go back to 2.

In step 3, the directional derivative is easily derived from the sub-gradients of the closed-loop spectral abscissa, by observing that, for any H in the controller-space,

$$\begin{aligned} d(\alpha\circ A_c)\left(K_l; H\right) &= \max_{\phi\in\partial(\alpha\circ A_c)(K_l)} \langle \phi, H\rangle \\ &= \max_{j\in\mathcal{J}(K_l)} \langle \phi_j(K_l), H\rangle \end{aligned}$$

The second equality holds since $\partial(\alpha \circ A_c)(K_l) = \text{co}\,\{\phi_j(K_l)\,|\,j \in \mathcal{J}(K_l)\}$, and since the maximum of a linear function over a polytope is necessarily attained on a vertex.

The additional stopping tests in step 5 allows the algorithm to stop when neither the controller nor the spectral abscissa updates are satisfactory, whereas the stationarity criterion on the value of θ is not met in step 2. The situation can occur if some of the active eigenvalues coalesce:

- In case of a multiple but semisimple eigenvalue, the Clarke subdifferential still exists (see 3.1), but is underestimated in the algorithm because we made the hypothesis that active eigenvalues were simple in order to reduce the SDP to a QP. More precisely in that case

$$\begin{aligned}\partial(\alpha \circ A_c)(K_l) &= \text{co}\left\{\text{Re}\ \left(CV_jY_jU_j^HB\right)^T\ |\, j \in \mathcal{J}(K_l),\, Y_j \succeq 0 \,\text{and}\, \text{Tr}(Y_j) = 1\right\} \\ &\supsetneq \text{co}\left\{\text{Re}\ \left(Cv_ju_j^HB\right)^T\ |\, j \in \mathcal{J}(K_l)\right\}\end{aligned}$$

- In case of a defective eigenvalue, the Clarke subdifferential no longer exists.

5.2 Variant II (second-order type)

The optimality function (6) does not specify the choice of the parameter δ. If second order information is available, it may therefore be attractive to replace the matrix δI in (6) by the inverse of the corresponding Hessian. Polak [28] extends the Newton method to min-max problems, but the corresponding dual expression for $\theta(K_l)$ does no longer reduce to a quadratic program like (9). We propose a different line here which is based on a heuristic argument. The quadratic term of θ is weighted by a scalar δ, which is updated at each step using a second-order model of $\alpha \circ A_c$. We suggest a quasi-Newton method based on the new optimality function $\hat{\theta}$ at iteration $l \geq 1$:

$$\hat{\theta}(K_l) = \min_{H \in \mathcal{K}} \max_{j \in \mathcal{J}_\rho(K_l)} \max_{\substack{Y_j \succeq 0 \\ \text{Tr}(Y_j) = 1}} \text{Re}\left(\mu_j\left(A_c\left(K_l\right)\right)\right) - \alpha\left(A_c\left(K_l\right)\right) + \langle \phi_j(K), H\rangle + \frac{1}{2}\text{vec}(H)^T Q_l \text{vec}(H) \tag{13}$$

The matrix Q_l is a positive-definite, symmetric $mp \times mp$ matrix, updated with the symmetric rank-two BFGS update

$$Q_{l+1} = Q_l + \frac{y_l y_l^T}{y_l^T s_l} - \frac{Q_l s_l s_l^T Q_l}{s_l^T Q_l s_l} \tag{14}$$

where $s_l = \text{vec}(K_{l+1} - K_l)$, and $y_l = \text{vec}(g_{l+1} - g_l)$, with g_l the subgradient of minimal norm in $\partial(\alpha \circ A_c)(K_l)$, by analogy with the gradient for smooth functions.

The dual form of (13) is then the convex QP

$$\hat{\theta}(K_l) = \max_{\substack{\tau_j \geq 0 \\ \sum_j \tau_j = 1}} -\alpha\left(A_c\left(K\right)\right) + \sum_{j=1}^{|\mathcal{J}_\rho(K_l)|} \tau_j \text{Re}\left(\mu_j\left(A_c\left(K_l\right)\right)\right) - \frac{1}{2}\text{vec}\left(\sum_{j=1}^{|\mathcal{J}_\rho(K_l)|} \tau_j \phi_j(K_l)\right)^T Q_l^{-1} \text{vec}\left(\sum_{j=1}^{|\mathcal{J}_\rho(K_l)|} \tau_j \phi_j(K_l)\right) \tag{15}$$

and the vectorized descent direction derived from the optimal $\left(\tau_j^\star(K_l)\right)$ convex coefficients is:

$$\text{vec}\left(\hat{H}(K_l)\right) = -Q_l^{-1} \sum_{j=1}^{|\mathcal{J}_\rho(K_l)|} \tau_j^\star(K_l)\, \text{vec}\left(\phi_j(K_l)\right) \tag{16}$$

Notice that the quadratic subproblem (15) invokes the inverse of the BFGS update Q_l. Inversion can be avoided by directly updating Q_l^{-1} at step 5, with the update formula

$$\begin{aligned}Q_{l+1}^{-1} &= Q_l^{-1} + \frac{(s_l - Q_l^{-1}y_l)s_l^T + s_l(s_l - Q_l^{-1}y_l)^T}{s_l^T y_s} - \frac{y_l^T(s_l - Q_l^{-1}y_l)s_l s_l^T}{(s_l^T y_l)^2} \\ &= \left(I - \frac{s_l y_l^T}{s_l^T y_l}\right) Q_l^{-1} \left(I - \frac{s_l y_l^T}{s_l^T y_l}\right)^T + \frac{s_l s_l^T}{s_l^T y_l}\end{aligned}$$

Algorithm 2 Second-order descent type algorithm for the closed-loop spectral abscissa

Set the parameters $\rho \in [0,1]$, $Q_0 \succ 0$, $K_0 \in \mathcal{K}$, ε_θ, ε_α, $\varepsilon_K > 0$, $\beta \in]0,1[$.
Set the counter $l \leftarrow 0$.

1. Compute $\alpha\left(A_c\left(K_0\right)\right)$, the enriched active index set $\mathcal{J}_\rho(K_0)$, the corresponding subgradients $\phi_j(K_0)$, and the subgradient of minimal norm g_0 in $\partial(\alpha \circ A_c)(K_0)$.

2. Solve (15) for $K = K_l$ and get the search direction $\hat{H}(K_l)$ from (16).
   If $\hat{\theta}(K_l) \geq -\varepsilon_\theta$then stop.

3. Find a step length $t_l > 0$ satisfying the Armijo line search condition

$$\alpha\left(A_c\left(K_l + t_l \hat{H}(K_l)\right)\right) \leq \alpha\left(A_c\left(K_l\right)\right) + \beta t_l d(\alpha \circ A_c)\left(K_l; \hat{H}(K_l)\right)$$

   (try $t_l = 1$ first)

4. Set $K_{l+1} \leftarrow K_l + t_l \hat{H}(K_l)$.
   Compute $\alpha\left(A_c\left(K_{l+1}\right)\right)$, the enriched active index set $\mathcal{J}_\rho(K_{l+1})$, the corresponding subgradients $\phi_j(K_{l+1})$, and the subgradient of minimal norm g_{l+1} in $\partial(\alpha \circ A_c)(K_{l+1})$.

5. Compute Q_{l+1} with the BFGS update formula (14).

6. If $\alpha\left(A_c\left(K_l\right)\right) - \alpha\left(A_c\left(K_{l+1}\right)\right) \leq \varepsilon_\alpha(1 + \alpha\left(A_c\left(K_l\right)\right))$ and $\|K_l - K_{l+1}\| \leq \varepsilon_K\left(1 + \|K_l\|\right)$
   then stop.
   Otherwise set $l \leftarrow l + 1$ and go back to 2.

6 Numerical examples

In this section we test our nonsmooth algorithm on a variety of stabilization problems in automatic control. In every application, we look for a static output feedback (SOF) controller first, starting our algorithm at $D_K = 0$. Dynamic controllers with fixed order are obtained using the dynamic augmentations (3). Finally, we will also show how to compute internally stabilizing controllers with fixed structure.

We use variant I of the descent algorithm in the following applications, with the default parameters values (unless other specific values are pointed out):

$$\rho = 0.8, \quad \delta = 0.1, \quad \varepsilon_\theta = 10^{-5}, \quad \varepsilon_\alpha = 10^{-6}, \quad \varepsilon_K = 10^{-6} \text{ and } \quad \beta = 0.9$$

We compare performance of our method with other minimization algorithms, namely multi-directional search (MDS), two algorithms implemented in the Matlab Optimization Toolbox, and the gradient sampling method of Overton *et al.* [11].

Multidirectional search (MDS) belongs to a larger family of direct search algorithms [29]. This derivative-free method explores the controller space via successive geometric transformations of a simplex (contraction, expansion and reflection). Its convergence to a local minimum is established for C^1-functions, but nonsmoothness can make it converge to a non-differentiable and non-optimal point [30], called a *dead point*. In [2] we have shown how to combine MDS with non-smooth descent steps in order to guarantee convergence. Here we use MDS with two stopping criteria: the first is based on the relative size of the simplex: $\frac{\max_i \|v_i - v_0\|}{\max(1, \|v_0\|)} < \varepsilon_1$ where v_i are the vertices of the simplex, and v_0 is the vertex of smallest function value of the objective f. The second stopping test quantifies the function values variations over the v_i: $\frac{\max_i f(v_i) - f(v_0)}{\max(1, |f(v_0)|)} < \varepsilon_2$. Experiments were performed with $\varepsilon_1 = \varepsilon_2 = 10^{-6}$.

Secondly, two Matlab Toolbox Optimization functions have been tested, one designed for general constrained optimization (`fmincon`), the second suited for min-max problems (`fminimax`). Both functions are essentially based on SQP algorithm with BFGS and line search and exact merit function (see [25]). The finite constrained cast of (4) passed to `fmincon` is

$$\begin{array}{l} \min \quad t \quad \text{subject to } \operatorname{Re}\left(\lambda_i(A + BKC)\right) \leq t \quad \text{for all } 1 \leq i \leq n \\ t \in \mathbb{R} \\ K \in \mathcal{K} \end{array}$$

Equivalently, the min-max formulation used for `fminimax` is

$$\min_{K \in \mathcal{K}} \max_{1 \leq i \leq n} \operatorname{Re}\left(\lambda_i(A + BKC)\right)$$

Clearly here we make the implicit assumption that all the eigenvalues are simple in order to work with smooth constraints or maximum of smooth functions, which is required by SQP. Our testing will show whether the toolbox functions run into difficulties in those cases where this hypothesis is violated.

Finally, we use the Matlab package HIFOO (version 1.0), provided at `http://www.cs.nyu.edu/overton/software/hifoo`. As discussed in [20], the underlying algorithm consists in a succession of (at most) three optimization phases: BFGS, local bundle (LB) and gradient sampling (GS). HIFOO is probabilistic and does not return the same final controller even when started from the same initial guess. We noticed that for the same reason the number of evaluations of the spectral abscissa could greatly change from one run to the other, even when all the settings are unchanged. This probabilistic feature of HIFOO is inherent to the multiple starting points strategy (by default, 3 random controllers, in addition to the user input), and to the gradient sampling algorithm itself. The first stabilizing controller is obtained with the parameter `'+'`, whereas the final one is with `'s'`. The iteration number of each stage is given as BFGS+LB+GS.

As we have indicated before, final controllers are not necessarily local minimizers of $\alpha \circ A_c$. The optimality is here of lower importance than in design problems with performance criteria. However, we discuss the status of every termination case encountered in the following examples, depending on the multiplicity of the active eigenvalues at the end, and on the dimension of the associated eigenspaces.

6.1 Academic test

We begin with an academic 2nd order SISO plant from the output feedback literature [24], cited in [23], and representing a damped linear oscillator. The open-loop is marginally unstable, with two pure imaginary modes ($\lambda_1 = i$ and $\lambda_2 = -i$). The closed-loop state matrix is

$$A_c(K) = A + BKC = \begin{bmatrix} 0 & 1 \\ -1 & K \end{bmatrix}$$

For $K \in \mathbb{R} \setminus [-2, 2]$ (resp. $K \in]-2, 2[$), $A_c(K)$ is diagonable with two real eigenvalues $\lambda = \frac{K}{2} \pm \sqrt{K^2 - 4}$ (resp. two complex conjugate eigenvalues $\lambda = \frac{K}{2} \pm i\sqrt{4 - K^2}$), and then $\alpha \circ A_c$ is Clarke regular. This is not the case at $K = -2$ (resp. $K = 2$), where the eigenvalues of $A_c(K)$ coalesce into the double root $\lambda = -1$ (resp. $\lambda = 1$). For these two controller values, $\alpha \circ A_c$ looses Lipschitz continuity, and the Clarke subdifferential is not defined.

Finally, we get $\alpha \circ A_c(K) = \frac{K}{2} + \sqrt{[K^2 - 4]_+}$. The global minimum is attained at $K = -2$, for which the θ based optimality certificate is unfortunately useless. Still for $K \notin \{-2, 2\}$, the stabilization algorithm provides a descent direction and allows to compute a descent step. We have tried various initial controller values K_0, taken on different pieces of the spectral abscissa curve.

For every given K_0, the algorithm converges to the global minimum. As soon as the relative step length and the relative spectral abscissa decrease are small enough, the algorithm stops.

case #	1	2	3	4
K_0	-5	0	2	5
first $\alpha < 0$ (iter.)	$-2.09 \cdot 10^{-1}$ (0)	$-2.50 \cdot 10^{-1}$ (1)	$-2.50 \cdot 10^{-1}$ (1)	$-5.73 \cdot 10^{-1}$ (3)
final α (iter.)	-1.00 (30)	-1.00 (17)	-1.00 (17)	-1.00 (17)
fun. evaluations	117	71	72	70
final θ	$-1.25 \cdot 10^{-1}$	$-1.25 \cdot 10^{-1}$	$-1.25 \cdot 10^{-1}$	$-1.25 \cdot 10^{-1}$
final bundle size	1	1	1	1
final active set size	1	1	1	1

Table 1: Damped linear oscillator stabilization

Surprisingly, even from $K_0 = 2$, our non-smooth stabilization algorithm converges to the global minimum, even though Clarke subgradients are not well defined at the first iteration. Figure 1 clearly shows the nonsmooth and even non-Lipschitz behavior of the closed-loop eigenvalues around at $K = \pm 2$.

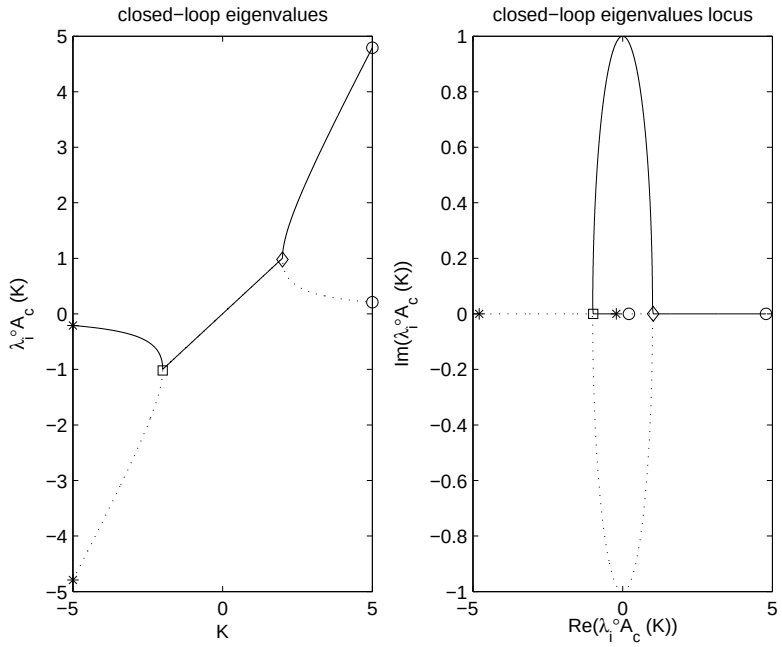

Figure 1: closed-loop eigenvalues variation around $K = 0$

Remark: For $K \in \{-2, 2\}$, the double eigenvalue of $A_c(K)$ is necessarily nonderogatory (for dimension reasons). The spectral abscissa is then subdifferentially regular at $A_c(-2)$ and $A_c(2)$, and in this academic example with exact and integer data, the corresponding subdifferential of $\alpha \circ A_c$ can be obtained explicitly. The Jordan decompositions are respectively

$$\begin{aligned} A_c(-2) &= \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix} \\ A_c(2) &= \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix} \end{aligned}$$

leading to the following regular subdifferentials

$$\begin{aligned} \hat{\partial}\left(\alpha \circ A_c\right)(-2) &= \left\{ \left(\mathrm{Re}\, C \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \varphi \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix} B \right)^T \mid \varphi \in \mathbb{C},\, \mathrm{Re}\, \varphi \geq 0 \right\} \\ &= \left\{ \frac{1}{2} - \varphi \mid \varphi \in \mathbb{R}_+ \right\} \\ &= \left[-\infty, \frac{1}{2} \right] \\ \hat{\partial}\left(\alpha \circ A_c\right)(2) &= \left\{ \left(\mathrm{Re}\, C \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \varphi \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix} B \right)^T \mid \varphi \in \mathbb{C},\, \mathrm{Re}\, \varphi \geq 0 \right\} \\ &= \left\{ \frac{1}{2} + \varphi \mid \varphi \in \mathbb{R}_+ \right\} \\ &= \left[\frac{1}{2}, +\infty \right] \end{aligned}$$

Even though the regular subdifferential could be used for characterizing the sharp local minimizers of $\alpha \circ A_c$ (see [19]), the detection of a Jordan block in A_c is a numerical challenge, because of the unstability of the Jordan form.

6.1.1 Matlab optimization toolbox and direct search

The first series is run with the initial controller $K_0 = 0$.

algorithm	MDS right-angled	`fmincon`	`fminimax`	`hifoo`
first $\alpha < 0$ (iter.)	-1.00 (1)	- (-)	-0.25 (1)	-0.25 (1+0+0)
final α (iter.)	-1.00 (21)	0 (198)	$-9.95 \cdot 10^{-1}$ (5)	-1.00 (8+2+5)
fun. evaluations	44	201	38	308

Table 2: Damped linear oscillator stabilization

`fmincon` could not converge after 200 function evaluations, and returned $K = -1.84 \cdot 10^{14}$. In this second benchmark, we tried to initialize the algorithms with the problematic controller value $K_0 = 2$.

`hifoo` finds the global minimum at the BFGS stage, tries in vain to improve it with local bundle and gradient sampling iterations, but doesn't give any local optimality certificate at the end (local optimality measure is $1.5 \cdot 10^{-1}$).

algorithm	MDS right-angled	`fmincon`	`fminimax`	`hifoo`
first $\alpha < 0$ (iter.)	-1.00 (2)	- (-)	- (-)	$-1.49 \cdot 10^{-8}$ (1)
final α (iter.)	-1.00 (21)	1.00 (100)	1.00 (100)	-1.00 (76+2+5)
fun. evaluations	44	201	201	1212

Table 3: Damped linear oscillator stabilization

`fmincon` and `fminimax` got stuck at $K = 2$, even after 200 function evaluations. MDS converged in both cases to the minimizer $K = -2$, with fewer function evaluations than the nonsmooth algorithm: the very small dimension of the search space is favorable to this direct search technique. In contrast, `hifoo` needs many more spectral abscissa calls in order to complete the optimization, without any optimality certificate, as above.

Remark: In the one-dimension case here, right-angled and regular simplices are intervals, the shape of the simplex has no influence on the MDS algorithm.

6.2 Transport airplane

The linearized plant of 9th-order describes the longitudinal motion of a transport airplane at given flight conditions [17, 22]. The open loop is unstable, with spectral abscissa $\alpha = 1.22 \cdot 10^{-2}$, attained by a simple, real mode: the composite function $\alpha \circ A_c$ is then perfectly differentiable at $K_0 = D_{K_0} = 0$ (all the other modes are stable, with real parts between $-5.00 \cdot 10^1$ and $-3.18 \cdot 10^{-2}$).

6.2.1 Nonsmooth optimization algorithm

Our nonsmooth algorithm solves the SOF stabilization problem after one iteration only, whether the initial subdifferential, reduced to a singleton, was enlarged or not. But if we let the algorithm run until reaching one of the stopping test described in algorithm 1, the final closed-loop depends on the bundle enrichment. The first three columns in table 4 show some termination information for no enlargement or for a very small enlargements of the subdifferential (indicated as 0 %, 0.1 % and 1 % enlargement). The low value of the optimality function θ indicates that no minimum is reached and that the algorithm got stuck near a dead point for $\alpha \circ A_c$. This is suggested by the graphical representation of $\alpha \circ A_c$ versus small variations δK of the final gain around the final controller, along a chosen direction (see figure 2 for the 0.1 % enlargement case). This is further confirmed by inspection of closed-loop modes: two complex conjugate eigenvalues with small imaginary parts are active ($\lambda_1 = -1.43 \cdot 10^{-1} + 1.15 \cdot 10^{-4} i$ and $\lambda_2 = \bar{\lambda}_1$) and they are very near from coalescence into a defective (but nonderogatory) real eigenvalue.

case #	1	2	3	4
subdiff. enlargement	0 %	0.1 %	1 %	2 %
first $\alpha < 0$ (iter.)	$-7.07 \cdot 10^{-2}$ (1)	$-1.07 \cdot 10^{-2}$ (1)	$-1.18 \cdot 10^{-2}$ (1)	$-1.05 \cdot 10^{-2}$ (1)
final α (iter.)	$-1.15 \cdot 10^{-1}$ (20)	$-1.43 \cdot 10^{-1}$ (27)	$-1.30 \cdot 10^{-1}$ (16)	$-4.45 \cdot 10^{-1}$ (9)
fun. evaluations	96	121	63	43
final θ	$-1.54 \cdot 10^{2}$	$-1.30 \cdot 10^{1}$	$-3.10 \cdot 10^{-1}$	$-5.60 \cdot 10^{-17}$
final bundle size	1	2	4	4
final active set size	1	1	1	2

Table 4: Transport airplane stabilization

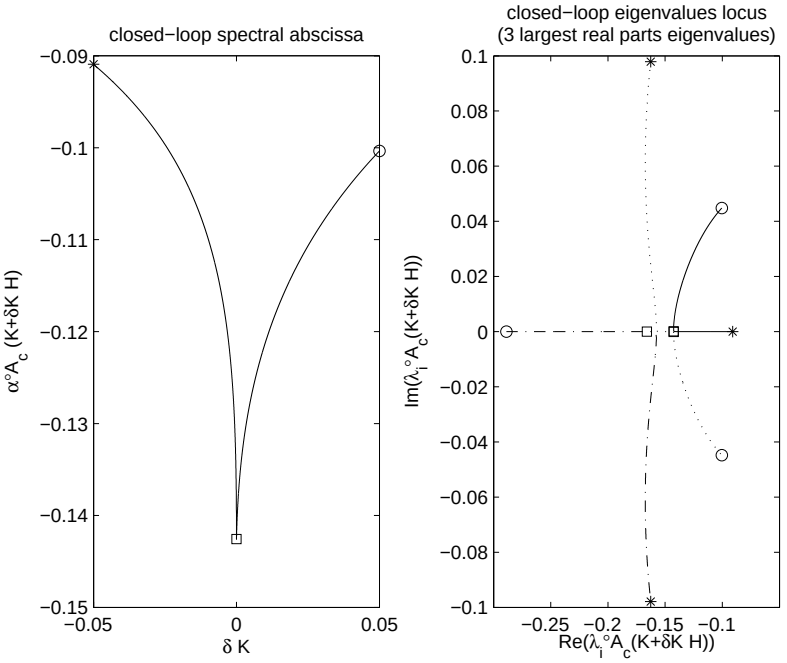

Figure 2: closed-loop eigenvalues variation around final K along $H = [1, 0, 0, 0, 0]$ (0.1 % enlargement)

The fourth case is more favorable. The inclusion of more subgradients into the bundle generates some better descent directions for $\alpha \circ A_c$ and allows longer descent steps. The final value of θ is close to zero, indicating local optimality.

There are three active eigenvalues on the last iteration: two of them are complex conjugate ($\lambda_1 = -4.45 \cdot 10^{-1} + 4.40 \cdot 10^{-3} i$ and $\lambda_2 = \bar{\lambda}_1$), the other one is real ($\lambda_3 = -4.45 \cdot 10^{-1}$). We notice that these three modes come directly from the plant, and are not controllable. This is confirmed by the associated closed-loop subgradients, $\phi_1 = \phi_2 \approx 0$ and $\phi_3 \approx 0$, driving to a singleton subdifferential $\partial(\alpha \circ A_c)(K_9) = \{0\}$. The final point is then smooth, in spite of multiple eigenvalue activity, and the uncontrollability of the active modes gives a global optimality certificate. As an illustration, see figure 3.

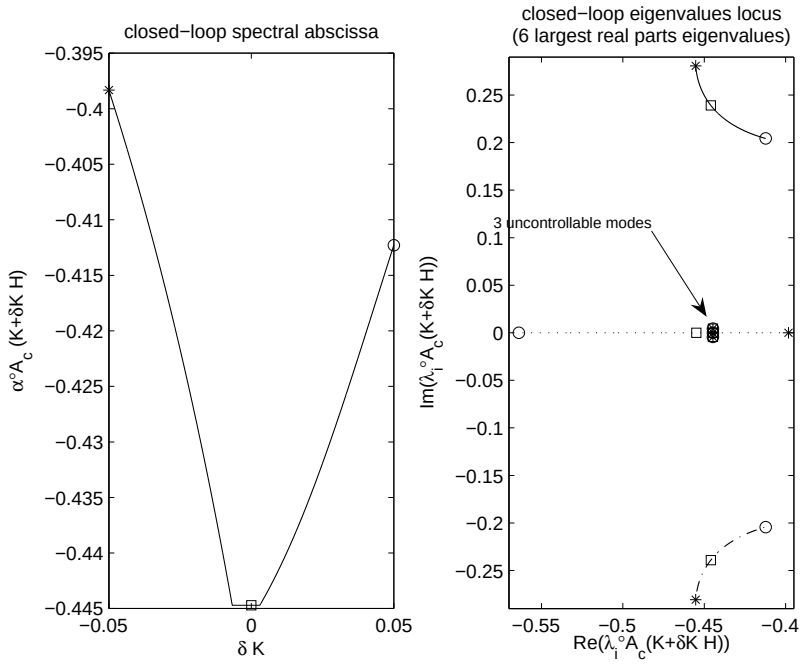

Figure 3: closed-loop eigenvalues variation around final K along $H = [1, 0, 0, 0, 0]$ (2 % enlargement)

6.2.2 Matlab optimization toolbox and direct search

The multidirectional search (MDS) algorithm was initialized with $K_0 = 0$ and applied with two different shapes of simplices in the controller space. Two of the Matlab optimization toolbox functions were tried: `fmincon` and `fminimax`.

algorithm	MDS right-angled	MDS regular	`fmincon`	`fminimax`	`hifoo`
first $\alpha < 0$ (iter.)	$-1.09 \cdot 10^{-1}$ (3)	$-1.04 \cdot 10^{-2}$ (7)	$-4.45 \cdot 10^{-1}$ (13)	$-1.13 \cdot 10^{-2}$ (1)	$-2.62 \cdot 10^{-2}$ (1+0+0)
final α (iter.)	$-4.21 \cdot 10^{-1}$ (36)	$-1.57 \cdot 10^{-1}$ (37)	$-4.45 \cdot 10^{-1}$ (13)	$-4.45 \cdot 10^{-1}$ (25)	$-2.31 \cdot 10^{-1}$ (396+3+2)
fun. evaluations	366	376	32	131	1140

Table 5: Transport airplane stabilization

MDS becomes very greedy in function evaluations, as the controller size increases in this example. Moreover, the global minimum is not found, either because of an unsuccessful local minimum, or a dead point.

Both Matlab functions return the global minimum, after very few iterations for `fmincon`.

`hifoo` terminates far from the global minimum, because slow convergence occurs: a large amount of BFGS iterations (99) is reached for each of the four starting controllers ($K_0 = 0$ and three perturbed K_0). The final optimality measure is $5.28 \cdot 10^{-4}$.

6.3 VTOL helicopter

This model with four states, one measurement and two control variables, describes the longitudinal motion of a VTOL (Vertical Take-Off and Landing) helicopter, at given flight conditions. The open-loop spectral abscissa is $\alpha = 2.76 \cdot 10^{-1}$, attained by two complex conjugate eigenvalues. All the open-loop eigenvalues are simple.

6.3.1 Nonsmooth optimization algorithms (variants I and II)

Alg. Variant	I	II (with BFGS)
first $\alpha < 0$ (iter.)	$-6.16 \cdot 10^{-2}$ (1)	$-6.16 \cdot 10^{-2}$ (1)
final α (iter.)	$-2.39 \cdot 10^{-1}$ (216)	$-2.47 \cdot 10^{-1}$ (26)
fun. evaluations	796	90
final θ	$-9.77 \cdot 10^{-6}$	$-1.70 \cdot 10^{-6}$
final bundle size	2	2
final active set size	1	2

Table 6: VTOL helicopter stabilization

Using our method, the closed-loop becomes stable after the first iteration already, and the spectral abscissa decreases slowly until satisfaction of the local optimality stopping test. This slow convergence strongly calls for variant II of our nonsmooth descent algorithm, which finds a lower closed-loop spectral abscissa within much less iterations. For both cases, the value close to 0 of θ indicates local optimality.

The final closed-loop spectrum at convergence obtained by algorithm variant I is

$$\Lambda \quad = \quad \left\{-2.39 \cdot 10^{-1} \pm 5.76 \cdot 10^{-1} i, -2.39 \cdot 10^{-1}, -7.91 \cdot 10^{1}\right\},$$

and the bundle subgradients associated with $\mu_1 = -2.39 \cdot 10^{-1} + 5.76 \cdot 10^{-1} i$ and with $\mu_3 = -2.39 \cdot 10^{-1}$ are, respectively,

$$\phi_1 = \phi_2 = \begin{bmatrix} -1.26 \cdot 10^{-1} \\ +2.92 \cdot 10^{-2} \end{bmatrix}, \quad \phi_3 = \begin{bmatrix} +6.26 \cdot 10^{-2} \\ -1.55 \cdot 10^{-2} \end{bmatrix}.$$

Convergence analysis is favorable for our algorithm, because the nonsmoothness comes from multiple active eigenvalues for the closed-loop spectral abscissa, each of them being simple: the Clarke subdifferential is then well defined and the value of $\theta(K)$ is reliable as an optimality criterion.

A plot in the neighborhood of the convergence point of variant I (see figure 4) clearly shows the nonsmoothness at optimality.

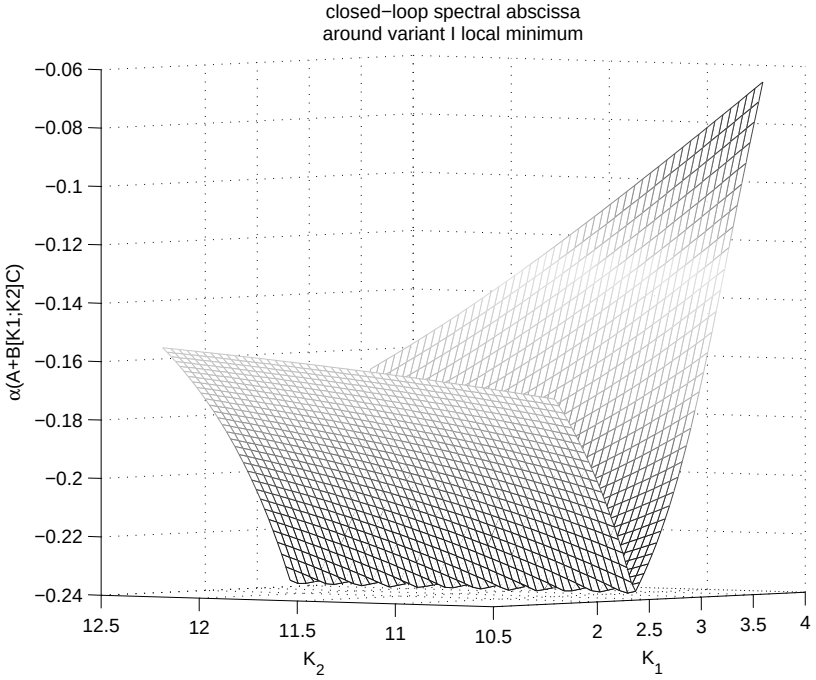

Figure 4: closed-loop spectral abscissa around optimal K of variant I algorithm

6.3.2 Matlab optimization toolbox and direct search

algorithm	MDS right-angled	MDS regular	`fmincon`	`fminimax`	`hifoo`
first $\alpha < 0$ (iter.)	$-1.23 \cdot 10^{-1}$ (1)	$-9.95 \cdot 10^{-2}$ (1)	$-2.47 \cdot 10^{-1}$ (36)	$-5.09 \cdot 10^{-2}$ (1)	$-6.16 \cdot 10^{-2}$ (1+0+0)
final α (iter.)	$-1.35 \cdot 10^{-1}$ (15)	$-2.47 \cdot 10^{-1}$ (58)	$-2.47 \cdot 10^{-1}$ (36)	$-2.47 \cdot 10^{-1}$ (36)	$-2.46 \cdot 10^{-1}$ (116+0+0)
fun. evaluations	63	235	73	73	216

Table 7: VTOL helicopter stabilization

The same closed-loop spectral abscissa is found by MDS (with regular simplex shape), the Matlab routines, which are very efficient in this example, and by `hifoo` (with final local optimality measure $9.87 \cdot 10^{-4}$). Notice that iterates of `fmincon` become feasible only at the last iteration, a classical feature of SQP algorithms.

6.4 High order model

Our last example taken from [23] is of rather high order (55 states), with two controlled inputs and two measured outputs. The state-space data describe a modified Boeing B-767 at flutter condition (see [16]). The open-loop is unstable, but the only active eigenvalues of A for the spectral abscissa are $\mu_1 = 1.015 \cdot 10^{-1}$ and $\bar{\mu}_1$, with multiplicity one.

6.4.1 Nonsmooth optimization algorithm

We use algorithm variants I and II, with all gains of the initial controller set to 0.

Alg. Variant	I	II (with BFGS)
first $\alpha < 0$ (iter.)	$-2.37 \cdot 10^{-2}$ (1)	$-2.36 \cdot 10^{-2}$ (1)
final α (iter.)	$-7.99 \cdot 10^{-2}$ (99)	$-3.50 \cdot 10^{-2}$ (29)
fun. evaluations	387	111
final θ	$-8.00 \cdot 10^{-4}$	$-8.70 \cdot 10^{-6}$
final bundle size	29	29
final active set size	1	1

Table 8: B-767 airplane stabilization

The two versions of our algorithm stabilize the plant after only one iteration. If the optimization is continued, variant II gives fast convergence to a local minimum (certified by the small value of θ). Variant I is slower here.

An inspection of the largest real part eigenvalues of $A_c(K)$ from the variant I final controller shows that one pair of complex conjugate eigenvalues is active ($\lambda_1 = -7.992 \cdot 10^{-2} + 4.912 \cdot 10^{-1} i$ and $\lambda_2 = \bar{\lambda_1}$); another pair is very close ($\lambda_3 = -7.996 \cdot 10^{-2} + 4.892 \cdot 10^{-1} i$ and $\lambda_4 = \bar{\lambda}_3$). The associated subgradients are nearly opposite matrices

$$\phi_1 = \phi_2 = \begin{bmatrix} -2.83 \cdot 10^2 & -5.56 \cdot 10^1 \\ 1.18 \cdot 10^6 & 2.11 \cdot 10^5 \end{bmatrix}, \quad \phi_3 = \phi_4 = \begin{bmatrix} 2.82 \cdot 10^2 & 5.54 \cdot 10^1 \\ -1.19 \cdot 10^6 & -2.12 \cdot 10^5 \end{bmatrix}$$

which explains the small final value of the optimality function θ.

Deeper analysis shows that the eigenvectors $v_1$and v_3 (resp. v_2 and v_4) are close from collinearity, indicating the neighboring coalescence of λ_1 and λ_3 (resp. $\lambda_2$and λ_4) into a defective eigenvalue.

algorithm	MDS right-angled	MDS regular	`fmincon`	`fminimax`	`hifoo`
first $\alpha < 0$ (iter.)	$-3.59 \cdot 10^{-3}$ (2)	$-2.35 \cdot 10^{-2}$ (13)	$-2.47 \cdot 10^{-2}$ (3)	$-2.36 \cdot 10^{-2}$ (1)	$-2.34 \cdot 10^{-2}$ (1+0+0)
final α (iter.)	$-3.23 \cdot 10^{-2}$ (60)	$-3.54 \cdot 10^{-2}$ (100)	$-3.44 \cdot 10^{-2}$ (9)	$-5.24 \cdot 10^{-2}$ (15)	$-3.62 \cdot 10^{-2}$ (132+3+15)
fun. evaluations	485	805	21	31	1858

Table 9: B-767 airplane stabilization

6.4.2 Matlab optimization toolbox and direct search

6.4.3 PID controllers

As our algorithm can handle controller structure (see 3.3), it offers an interesting framework for PID controller design, particularly attractive for MIMO plants where very few generic tuning techniques are available. In this example, we seek a 2 input, 2 output stabilizing PID controller with the following transfer matrix:

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

where K_P, K_I, $K_D \in \mathbb{R}^{2\times 2}$ and $\varepsilon > 0$. The algorithm is initialized with $K_P = K_I = K_D = 0$ and $\varepsilon = 10^{-2}$. The resulting closed-loop is unstable ($\alpha(A_c(K)) = 1.015 \cdot 10^{-1}$). The algorithm (variant I) stops after 22 iterations, and returns the following values:

$$K_P = \begin{bmatrix} 1.91 \cdot 10^{-2} & 5.03 \cdot 10^{-6} \\ 1.52 \cdot 10^{-3} & 3.50 \cdot 10^{-5} \end{bmatrix}, K_I = \begin{bmatrix} 1.61 \cdot 10^{-1} & 1.56 \cdot 10^{-4} \\ 7.88 \cdot 10^{-2} & -1.02 \cdot 10^{-3} \end{bmatrix},$$
$$K_D = \begin{bmatrix} -1.97 \cdot 10^{-8} & -6.92 \cdot 10^{-7} \\ -1.88 \cdot 10^{-9} & -4.45 \cdot 10^{-7} \end{bmatrix}, \varepsilon = 10^{-2}$$

The final closed-loop spectral abscissa is $-2.97 \cdot 10^{-3}$. Notice that the derivative coefficients are nearly 0, while ε is unchanged.

6.4.4 Dynamic controllers

k (contr. order)	1	2	3	4	5
first $\alpha < 0$ (iter.)	$-2.15 \cdot 10^{-2}$ (2)	$-2.46 \cdot 10^{-2}$ (1)	$-3.49 \cdot 10^{-3}$ (11)	$-9.40 \cdot 10^{-5}$ (11)	$-1.44 \cdot 10^{-4}$ (1)
final α (iter.)	$-5.91 \cdot 10^{-2}$ (53)	$-1.76 \cdot 10^{-1}$ (98)	$-3.88 \cdot 10^{-3}$ (25)	$-2.98 \cdot 10^{-3}$ (25)	$-5.70 \cdot 10^{-4}$ (10)
fun. evaluations	264	536	147	143	67

Table 10: B-767 airplane stabilization by dynamic output feedback

Stabilizing dynamic controllers were obtained with the minimal, stable and balanced parametrization in section 3.2, combined with algorithm variant I. By testing several initial guesses K, we noticed that the algorithm converged to local minimizers that were not stabilizing. In these cases a restart with a different initial K_0 became inevitable.

6.5 Conclusion

Formulated as an optimization program, fixed-order output feedback stabilization has been solved for several case studies from the literature. The proposed nonsmooth algorithm addresses the nonsmoothness of the spectral abscissa and generates successive descent steps. Even if the theoretical assumption of semisimple active eigenvalues may seem restrictive, the experimental results show that our framework is generic enough to handle realistic stabilization problems. The two proposed variants are deterministic and numerically efficient, often with much less spectral abscissa evaluations than MDS or HIFOO. Surprisingly, we noticed that the smooth optimization tools from the Matlab optimization toolbox offer good alternatives, except in the academic test example.

References

[1] B.D.O. Anderson, N.K. Bose, and E.I. Jury. Output feedback stabilization and related problems–solution via decision methods. *IEEE Transactions on Automatic Control*, 20(1), 1975.

[2] P. Apkarian and D. Noll. Controller design via nonsmooth multi-directional search. *SIAM Journal on Control and Optimization*, 44(6):1923–1949, 2006. also to appear in SICON.

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

[4] P. Apkarian and D. Noll. Nonsmooth optimization for multiband frequency domain control design. *Automatica*, 2006.

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

[6] D.S. Bernstein. Some open problems in matrix theory arising in linear systems and control. *Linear Algebra and its Applications*, 162–164:409–432, 1992.

[7] V. Blondel and J. Tsitsiklis. NP-hardness of some linear control design problems. *SIAM Journal on Control and Optimization*, 35(6):2118–2127, 1997.

[8] V. Bompart, D. Noll, and P. Apkarian. Second-order nonsmooth optimization for h_∞ synthesis. In *5th IFAC Symposium on Robust Control Design*, Toulouse, France, July 2006.

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

[10] J.V. Burke, A.S. Lewis, and M.L. Overton. Approximating subdifferentials by random sampling of gradients. *Mathematics of Operations Research*, 27:567–584, 2002.

[11] J.V. Burke, A.S. Lewis, and M.L. Overton. Two numerical methods for optimizing matrix stability. *Linear Algebra and its Applications*, 351-352:117–145, 2002.

[12] J.V. Burke, A.S. Lewis, and M.L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15:751–779, 2005.

[13] J.V. Burke and M.L. Overton. Differential properties of the spectral abscissa and the spectral radius for analytic matrix-valued mappings. *Nonlinear Analysis: Theory, Methods & Applications*, 23:467–488, 1994.

[14] J.V. Burke and M.L. Overton. Variational analysis of non-Lipschitz spectral functions. *Mathematical Programming*, 90(2):317–351, 2001.

[15] C.T. Chou and J.M. Maciejowski. System identification using balanced parametrizations. *IEEE Transactions on Automatic Control*, 42(7):956–974, 1997.

[16] E.J. Davison. Benchmark problems for control system design. Technical report, International Federation of Automatic Control, 1990.

[17] D. Gangsaas, K.R. Bruce, J.D. Blight, and U.-L. Ly. Application of modern synthesis to aircraft control: Three case studies. *IEEE Transactions on Automatic Control*, 31(11):995–1014, 1986.

[18] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Springer-Verlag, 1993.

[19] A.S. Lewis J.V. Burke and M.L. Overton. Optimizing matrix stability. *Proceedings of the American Mathematical Society*, 129:1635–1642, 2001.

[20] A.S. Lewis J.V. Burke, D. Henrion and M.L. Overton. HIFOO - A MATLAB package for fixed-order controller design and H_∞ optimization. In *5th IFAC Symposium on Robust Control Design*, Toulouse, France, July 2006.

[21] K.C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*, volume 1133 of *Lecture Notes in Mathematics 1133*. Springer-Verlag.

[22] F. Leibfritz. Computational design of stabilizing static output feedback controllers. Technical Report 99-01, University of Trier, 1999.

[23] F. Leibfritz and W. Lipinski. Description of the benchmark examples in COMPl$_e$ib 1.0. Technical report, University of Trier, 2003.

[24] W.S. Levine and M. Athans. On the determination of the optimal constant output-feedback gains for linear multivariable systems. *IEEE Transactions on Automatic Control*, 15(1):44–48, 1970.

[25] The MathWorks. *Optimization Toolbox User's Guide*, march 2006. Version 3.

[26] R.J. Ober. Balanced parametrizations of classes of linear systems. *SIAM Journal on Control and Optimization*, 29:1251–1287, 1991.

[27] Elijah Polak. On the mathematical foundations of nondifferentiable optimization in engineering design. *SIAM Review*, 29(1):21–89, 1987.

[28] Elijah Polak. *Optimization: Algorithms and Consistent Approximations*. Springer, 1997.

[29] V. Torczon. *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*. PhD thesis, Department of Mathematical Sciences, Rice University, Houston, Texas, May 1989.

[30] V. Torczon. On the convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, 1(1):123–145, 1991.