# Non-smooth techniques for stabilizing linear systems

Vincent Bompart \* Pierre Apkarian <sup>†</sup> Dominikus Noll <sup>‡</sup>

#### Abstract

We discuss closed-loop stabilization of linear time-invariant dynamical systems, a problem which frequently arises in controller synthesis, either as a stand-alone task, or to initialize algorithms for  $H_{\infty}$ 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 [11] have been successfully used to minimize the spectral abscissa of the closed-loop state matrix (the largest real part of its eigenvalues). 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 minimizing the spectral abscissa 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. From an algorithmic point of view, the output feedback sta-

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

<sup>&</sup>lt;sup>†</sup>ONERA, Control System Department - and - Université Paul Sabatier, Institut de Mathématiques, Toulouse, France - Email: apkarian@onera.fr - Tel: +33 5.62.25.27.84 -Fax: +33 5.62.25.27.64.

<sup>&</sup>lt;sup>‡</sup>Université Paul Sabatier, Institut de Mathématiques, Toulouse, France -Email: noll@mip.ups-tlse.fr - Tel: +33 5.61.55.86.22 - Fax: +33 5.61.55.83.85.

bilization problem is clearly in the class NP and conjectured to be NP-hard [6]. Necessary and sufficient conditions leading to an efficient algorithmic solution are still not known [5].

A less ambitious line is to address internal stability as a local optimization problem. Recent approaches using non-smooth optimization techniques are [9, 11] for stabilization, and [2, 3, 4, 7] for  $H_{\infty}$  synthesis. In [11] for instance the authors propose to optimize the spectral abscissa of the closed-loop matrix via specific non-smooth techniques.

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. While local optimization techniques do not provide the strong certificates of global methods, we believe that they offer better chances in practice to solve the stability problem.

## Matrix notations

The *n* eigenvalues of  $M \in \mathbb{C}^{n \times n}$  (repeated with multiplicity) are denoted  $\lambda_1(M), \ldots, \lambda_n(M)$  in lexicographic order. The distinct eigenvalues are denoted  $\mu_1(M), \ldots, \mu_q(M)$ , with respective algebraic multiplicities  $n_1, \ldots, n_q$  and geometric multiplicities  $p_1, \ldots, p_q$ .

In the sequel,  $\alpha(M)$  denotes the spectral abscissa of M, defined as  $\alpha(M) = \max_{1 \le j \le q} \operatorname{Re}(\mu_j(M))$ . Any eigenvalue of M whose real part attains  $\alpha(M)$  is said to be active.

## Plant and controller notations

The open-loop system we wish to stabilize is a continuous linear time-invariant plant, described without loss of generality by the state-space equations

$$P(s): \begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$
(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 consider static or dynamic output feedback control laws of the form u = K(s)y in order to stabilize (1) internally. We suppose that the order of the controller  $k \in \mathbb{N}$  is fixed. In the case of static feedback (k = 0), the controller is denoted by  $K \in \mathbb{R}^{m \times p}$ . For dynamic controllers we use standard substitutions in order to reduce to the static feedback case. The affine mapping  $K \mapsto A + BKC$ is denoted as  $A_c$ . The set of all closed-loop active eigenvalues is denoted  $\mathcal{A}(K) = \{\mu_j (A_c(K)) : \operatorname{Re}(\mu_j (A_c(K))) = \alpha(A_c(K)))\}$ , the corresponding set of active indices is  $\mathcal{J}(K)$ .

## 2 Minimizing the spectral abscissa

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

$$\min_{K \in \mathcal{K}} \alpha(A + BKC) \tag{2}$$

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. 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  $\alpha \ge 0$ , we know at least that the situation cannot be improved in a neighborhood of  $K^*$ , and that a restart away from that local minimum is inevitable.

Program (2) is difficult to solve for two reasons. Firstly, the minimax formulation calls for non-smooth 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  $\alpha$  have been analyzed by Burke and Overton [14]. In [13] the authors show that if the active eigenvalues of M are all semisimple  $(n_j = p_j)$ ,  $\alpha$  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  $\mathcal{A}(K)$ .

Several strategies for addressing the non-smoothness in (2) have been put forward: Burke, Lewis and Overton have extended the idea of gradient bundle methods (see [16] for the convex case and [17] 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.

## 3 Subgradients of the spectral abscissa

### 3.1 Subgradients 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 (s if counted with their multiplicity). The Jordan form J(K) of

 $A_c(K)$  is then partly diagonal, more precisely :

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

- $D(K) = \text{diag} [\lambda_1(A_c(K)), \dots, \lambda_s(A_c(K))]$  is the diagonal part of active eigenvalues.
- $J_j(K)$ , for  $r < j \le q$  are  $n_j \times n_j$  block-diagonal matrices of Jordan blocks,
- $V(K) = [v_1(A_c(K)), \dots, v_n(A_c(K))],$  where the first s columns are right eigenvectors of  $A_c(K)$  associated with the active eigenvalues,
- $V(K)^{-1} = \begin{bmatrix} u_1 (A_c(K))^H \\ \vdots \\ u_n (A_c(K))^H \end{bmatrix}$ , where the first *s* rows are left eigenvectors of

 $A_c(K)$  associated with the active eigenvalues.

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

The function  $\alpha \circ A_c$  is Clarke regular at K, as a composition of the affine mapping  $A_c$  with  $\alpha$ , which is locally Lipschitz continuous at K. Let  $\mu_i \in \mathcal{A}(K)$  be an active eigenvalue of  $A_c(K)$ , then the real matrix

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

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

$$\phi_Y(K) = \sum_{j \in \mathcal{J}(K)} \left( \operatorname{Re} \left( CV_j Y_j U_j^H B \right) \right)^T$$
(3)

where  $Y_j \succeq 0$  and  $\sum_{j \in \mathcal{J}(K)} \operatorname{Tr}(Y_j) = 1$ . Notice that the complex conjugate paired active eigenvalues  $\mu_j$  and  $\mu_k =$  $\bar{\mu}_j \ (k \neq j)$  share the same closed-loop spectral abscissa subgradient  $\phi_j = \phi_k$ .

**Remark 1** 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  $\mu_l(A) = \mu_i(A_c(K))$ . Moreover, if this eigenvalue is semisimple and active for  $\alpha \circ A_c$ , the associated subgradients are null, because  $U_i^H B =$ 

0. The case of unobservable modes leads to the same conclusion, because  $CV_j = 0$ . In this way, whenever an uncontrollable or unobservable openloop 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  $Re \ \mu_l(A)$  is a lower bound for  $\alpha \circ A_c$ .

## 3.2 Subgradients and dynamic controllers

The problem of stabilizing the plant P by dynamic output feedback reduces formally to the static case. 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  $B_K = 0$  and  $C_K = 0$  (neither observable nor controllable) and with  $\alpha(A_K) < \alpha(A + BD_K C)$ , it is straightforward to show that the resulting subgradients of the closed-loop spectral abscissa are convex linear combinations of matrices of the form

$$\phi_j(K) = \begin{bmatrix} 0 & 0 \\ 0 & \operatorname{Re}\left(CV_jY_jU_j^HB\right)^T \end{bmatrix}$$

where  $V_j$  (resp.  $U_j^H$ ) are blocks of right (resp. left) eigenvectors associated with the active eigenvalues of  $A + BD_KC$ , and  $Y_j \succeq 0$ ,  $\text{Tr}(Y_j) = 1$ . As the successive search directions have the same structure, see (6), 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 [20].

## 3.3 Subgradients with structured controllers

Formulation (2) is general enough to handle state-space structured controllers, such as decentralized or PID controllers. Let  $K : \mathbb{R}^{\tilde{k}} \longrightarrow \mathbb{R}^{m \times p}$ be a smooth parametrization of an open subset  $\mathcal{K} \subset \mathbb{R}^{m \times p}$ , containing statespace realizations of a family of controllers of a given structure. Then the stabilization problem can be written as  $\min_{\kappa \in \mathbb{R}^{\tilde{k}}} \alpha (A_c \circ K(\kappa))$ . The Clarke subgradients  $\psi \in \mathbb{R}^{\tilde{k}}$  of the composite function  $\alpha \circ A_c \circ K$  are derived from (3) with the chain rule (see [15, section 2.3])  $\psi(\kappa) = J_{\text{vec}(K)}(\kappa)^T \text{vec}(\phi(K(\kappa)))$  where  $J_{\operatorname{vec}(K)}(\kappa) \in \mathbb{R}^{mp \times \tilde{k}}$  is the Jacobian matrix of  $\operatorname{vec}(K)$  :  $\kappa \in \mathbb{R}^{\tilde{k}} \mapsto \operatorname{vec}(K(\kappa)) \in \mathbb{R}^{mp}$ .

## 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 [21, 22]. It was described and applied in the semi-infinite context of the  $H_{\infty}$  synthesis in [2]. 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.

We first make the strong assumption that all the eigenvalues of the closedloop state matrix  $A_c(K)$  are semisimple. Then, with  $\delta > 0$  fixed, we define

$$\theta(K) = \min_{H \in \mathbb{R}^{m \times p}} \max_{1 \le j \le q} \max_{\substack{Y_j \succeq 0 \\ \operatorname{Tr}(Y_j) = 1}} \left[ \operatorname{Re}\left(\mu_j\left(A_c\left(K\right)\right)\right) - \alpha\left(A_c\left(K\right)\right) + \langle \phi_j(K), H \rangle + \frac{1}{2}\delta \|H\|^2 \right]$$
(4)

Using Fenchel duality for permuting the min and double max operators, we obtain the dual form of (4), where the inner minimization over H becomes unconstrained and can be computed explicitly, leading to:

$$\theta(K) = \max_{\substack{\tau_j \ge 0 \\ \sum_j \tau_j = 1 \\ }} \max_{\substack{Y_j \succeq 0 \\ \sum_j \tau_j = 1 \\ }} \left| -\alpha \left( A_c \left( K \right) \right) \right| \\ + \sum_{j=1}^q \tau_j \operatorname{Re} \left( \mu_j \left( A_c \left( K \right) \right) \right) - \frac{1}{2\delta} \left\| \sum_{j=1}^q \tau_j \phi_j(K) \right\|^2 \right]$$
(5)

and we get the minimizer H(K) of the primal formulation (4) from the solution  $\left(\left(\tau_{j}^{\star}(K)\right)_{1\leq j\leq q}, \left(Y_{j}^{\star}(K)\right)_{1\leq j\leq q}\right)$  of the dual expression (5) in the explicit form

$$H(K) = -\frac{1}{\delta} \sum_{j=1}^{q} \tau_j^{\star}(K) \operatorname{Re} \left( C V_j Y_j^{\star}(K) U_j^H B \right)^T.$$
(6)

We recall from [21] the basic properties of  $\theta$  and H:

1.  $\theta(K) \leq 0$  for all  $K \in \mathbb{R}^{m \times p}$ , and  $\theta(K) = 0$  if and only if  $0 \in \partial(\alpha \circ A_c)(K)$ . 2. If  $0 \notin \partial(\alpha \circ A_c)(K)$ , then H(K) is a descent direction for the closed-loop spectral abscissa at K. More precisely for all K:

$$d(\alpha \circ A_c)(K; H(K)) \le \theta(K) - \frac{1}{2}\delta \|H(K)\|^2 \le \theta(K).$$

- 3. The function  $\theta$  is continuous.
- 4. The operator  $K \mapsto H(K)$  is continuous.

Therefore direction H(K) will be chosen as a search direction in a descenttype algorithm and combined with a line search. The continuity of  $H(\cdot)$ ensures that every accumulation point  $\overline{K}$  in the sequence of iterates satisfies the necessary optimality condition  $0 \in \partial(\alpha \circ A_c)(\overline{K})$  (see [2]). Notice that even for semisimple eigenvalues, continuity fails for the steepest descent direction. This is why steepest descent steps for non-smooth functions may fail to converge. In our case this justifies the recourse to the quadratic, firstorder model  $\theta$  as a descent function. Moreover, properties 1) and 3) suggest a stopping test based on the value of  $\theta(K)$ , because as soon as  $\theta(K) \geq -\varepsilon_{\theta}$  (for a small given  $\varepsilon_{\theta} > 0$ ), the controller K is in a neighborhood of a stationary point.

## 5 Non-smooth 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 (5) of  $\theta(K_l)$ . Without any additional hypothesis, it is a semidefinite program (SDP). Assuming that all the eigenvalues of  $A_c(K)$  are simple, the SDP (5) reduces to a concave quadratic maximization program

To go one step further, we reduce the dimension of the search space. For a given ratio  $\rho \in [0, 1]$ , we define the following extended set of active eigenvalues

$$\mathcal{A}_{\rho}(K) = \left\{ \mu_{j}\left(A_{c}(K)\right) : \alpha\left(A_{c}(K)\right) - \operatorname{Re}\left(\mu_{j}\left(A_{c}(K)\right)\right) \\ \leq \rho\left[\alpha\left(A_{c}(K)\right) - \min_{1 \leq i \leq n} \operatorname{Re}\left(\mu_{i}\left(A_{c}(K)\right)\right)\right] \right\}$$
(7)

 $\mathcal{J}_{\rho}(K)$  is the corresponding enriched active index set. It is clear that  $\rho \mapsto \mathcal{A}_{\rho}(K)$  is non-decreasing on [0, 1], and that  $\mathcal{A}(K) = \mathcal{A}_0(K) \subset \mathcal{A}_{\rho}(K) \subset \mathcal{A}_1(K) = \text{spec } (\mathcal{A}_c(K))$  for all  $\rho \in [0, 1]$ . Hence, we have locally

$$\alpha\left(A_{c}\left(K\right)\right) = \max_{j \in \mathcal{J}_{\rho}(K)} \operatorname{Re}\left(\mu_{j}\left(A_{c}(K)\right)\right)$$
(8)

By applying the descent function  $\theta$  to this local formulation, we finally get the quadratic program

$$\theta(K) = \max_{\substack{\tau_j \ge 0 \\ \sum_j \tau_j = 1}} \left| -\alpha \left( A_c \left( K \right) \right) + \sum_{j=1}^{|\mathcal{J}_{\rho}(K)|} \tau_j \operatorname{Re} \left( \mu_j \left( A_c \left( K \right) \right) \right) - \frac{1}{2\delta} \| \sum_{j=1}^{|\mathcal{J}_{\rho}(K)|} \tau_j \phi_j(K) \|^2 \right].$$
(9)

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

$$H(K) = -\frac{1}{\delta} \sum_{j=1}^{|\mathcal{J}_{\rho}(K)|} \tau_{j}^{\star}(K) \operatorname{Re} \left( C v_{j} u_{j}^{H} B \right)^{T}$$
(10)

Notice that for  $\rho = 0$  the QP in (9) reduces to the steepest descent finding problem while  $\rho = 1$  reproduces (5). The parameter  $\rho$  offers some additional numerical flexibility, and allows the weaker assumption that only eigenvalues in  $\mathcal{A}_{\rho}(K)$  are simple.

## 5.2 Variant II (second-order type)

In the optimality function (4) the parameter  $\delta$  acts as an estimate of the average of the curvatures of Re  $\mu_j \circ A_c$ . If second order information is available, it may therefore be attractive to replace the scalar  $\delta$  in (4) by Hessian matrices. Polak [22] 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 approach here which is based on a heuristic argument. The quadratic term of  $\hat{\theta}$  is weighted by a matrix  $Q_l$ , 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  $l \geq 1$ :

$$\hat{\theta}(K_l) = \min_{H \in \mathbb{R}^{m \times p}} \max_{\substack{j \in \mathcal{J}_{\rho}(K_l) \\ \mathrm{Tr}(Y_j) = 1}} \left[ \operatorname{Re}\left(\mu_j\left(A_c\left(K_l\right)\right)\right) - \alpha\left(A_c\left(K_l\right)\right) + \langle\phi_j(K), H\rangle + \frac{1}{2}\operatorname{vec}(H)^T Q_l \operatorname{vec}(H) \right]$$
(11)

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

Set  $\rho \in [0, 1]$ ,  $\delta > 0$ ,  $K_0 \in \mathbb{R}^{m \times p}$ ,  $\varepsilon_{\theta}$ ,  $\varepsilon_{\kappa} > 0$ ,  $\beta \in ]0, 1[$ . Set the counter  $l \leftarrow 0$ .

- 1. Compute  $\alpha(A_c(K_0))$ , the enriched active index set  $\mathcal{J}_{\rho}(K_0)$  and the corresponding subgradients  $\phi_i(K_0)$ .
- 2. Solve (9) for  $K = K_l$  and get the search direction  $H(K_l)$  from (10). If  $\theta(K_l) \ge -\varepsilon_{\theta}$  then stop.
- 3. Find a step length  $t_l > 0$  satisfying the Armijo line search condition

$$\alpha \left( A_c \left( K_l + t_l H(K_l) \right) \right) \leq \alpha \left( A_c \left( K_l \right) \right) + \beta t_l \theta(K_l)$$

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

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

The dual form of (11) is then a convex QP and the vectorized descent direction derived from the optimal  $(\tau_j^{\star}(K_l))$  convex coefficients is:

$$\operatorname{vec}\left(\hat{H}(K_l)\right) = -Q_l^{-1} \sum_{j=1}^{|\mathcal{J}_{\rho}(K_l)|} \tau_j^{\star}(K_l) \operatorname{vec}\left(\phi_j(K_l)\right)$$
(12)

## 6 Numerical examples

In this section we test our non-smooth algorithm on a variety of output feedback stabilization problems from the literature. We use variants I and II of the descent algorithm in the following applications, with the default parameters values (unless other values are specified):  $\rho = 0.8$ ,  $\delta = 0.1$ ,  $\varepsilon_{\theta} = 10^{-5}$ ,  $\varepsilon_{\alpha} = 10^{-6}$ ,  $\varepsilon_{K} = 10^{-6}$  and  $\beta = 0.9$ .

We compare the performance of our method with that of other minimization algorithms, namely multi-directional search (MDS), two algorithms implemented in the Matlab Optimization Toolbox, and the Matlab package HIFOO [8].

Multidirectional search (MDS) belongs to the family of direct search algorithms [23]. This derivative-free method explores the controller space via successive geometric transformations of a simplex. Its convergence to a local minimum is established for  $C^1$ -functions, but non-smoothness can make it converge to a non-differentiable and non-optimal point [24], called a *dead point*. In [1] we have shown how to combine MDS with non-smooth descent steps in order to avoid this difficulty and guarantee convergence. Experiments were performed with two simplex shapes (MDS 1: right-angled, MDS 2: regular).

Secondly, two Matlab Optimization Toolbox 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, line search and an exact merit function, see [19]. 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). As discussed in [8], the underlying algorithm consists in a succession of (at most) three optimization phases: BFGS, local bundle (LB) and gradient sampling (GS). By virtue of its probabilistic nature, HIFOO does not return the same final controller even when started from the same initial guess. This probabilistic feature of HIFOO is inherent to the multiple starting point 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.

Examples 6.1, 6.2 and 6.3 are initialized with  $K_0 = 0$ . We discuss the status of every termination case in terms of active eigenvalues multiplicity, and of associated eigenspaces dimension.

### 6.1 Transport airplane

This linearized plant of 9th-order describes the longitudinal motion of a transport airplane at given flight conditions (system AC8 from [18]). 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 differentiable at  $K_0 = 0$ .

#### 6.1.1 Non-smooth optimization algorithm

In table 1, we show the influence of the ratio  $\rho$  (see equation (7)) on the non-smooth algorithm (variant I here). Notice that the first case ( $\rho = 0$ ) is steepest descent. In each of the first two cases, the final value of  $\theta$  is not

1					
case $\#$	1	2	3		
ratio $\rho$	0 %	0.1~%	2 %		
first $\alpha < 0$	$-7.07 \cdot 10^{-2}$	$-1.07 \cdot 10^{-2}$	$-1.05 \cdot 10^{-2}$		
(iter.)	(1)	(1)	(1)		
final $\alpha$	$-1.15 \cdot 10^{-1}$	$-1.43 \cdot 10^{-1}$	$-4.45 \cdot 10^{-1}$		
(iter.)	(20)	(27)	(9)		
fun. eval.	96	121	43		
final $\theta$	$-1.54 \cdot 10^{2}$	$-1.30 \cdot 10^{1}$	$-5.60 \cdot 10^{-17}$		

 Table 1: Transport airplane stabilization

reliable for optimality, because  $\alpha \circ A_c$  looses Clarke regularity.

The third case is more favorable. The enlargement of  $\mathcal{A}_{\rho}(K)$  generates better descent directions for  $\alpha \circ A_c$  and allows longer descent steps and fewer iterations. The final value of  $\theta$  is close to zero, indicating local optimality. There are three active eigenvalues at 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  $\phi_3 \approx 0$ , leading to a singleton subdifferential  $\partial(\alpha \circ A_c)(K_9) = \{0\}$ . The final point is then smooth, in spite of multiple active eigenvalues, and the uncontrollability of the active modes gives a global optimality certificate (see Remark 1, section 3.1).

#### 6.1.2 Other algorithms

MDS is very greedy in function evaluations, and 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: numerous BFGS iterations (99) are needed for each of the four initial controllers ( $K_0 = 0$  and three perturbed  $K_0$ ). The final optimality measure is  $5.28 \cdot 10^{-4}$ .

## 6.2 VTOL helicopter

This model (HE1 from [18]) with four states, one measurement and two control variables, describes the longitudinal motion of a VTOL (Vertical

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

Table 2: Transport airplane stabilization

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.

#### 6.2.1Non-smooth optimization algorithm (variants I and II)

Table 3: VIOL helicopter stabilization					
Alg. Variant	Ι	II (with BFGS)			
first $\alpha < 0$ (iter.)	$-6.16 \cdot 10^{-2} (1)$	$-6.16 \cdot 10^{-2} (1)$			
final $\alpha$ (iter.)	$-2.39 \cdot 10^{-1} (216)$	$-2.47 \cdot 10^{-1} (26)$			
fun. eval.	796	90			
final $\theta$	$-9.77 \cdot 10^{-6}$	$-1.70 \cdot 10^{-6}$			

Table 2. VTOI baliant

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

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

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

and the 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 non-smoothness comes from several simple active eigenvalues for the closed-loop spectral abscissa: the Clarke subdifferential is then well defined and the value of  $\theta(K)$ is reliable as an optimality criterion.

#### 6.2.2Other algorithms

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.

Table 1. VIOL hencepter stabilization					
	first	(iter.)	final	(iter.)	fun.
algorithm	$\alpha < 0$		$\alpha$		eval.
MDS 1	$-1.23 \cdot 10^{-1}$	(1)	$-1.35 \cdot 10^{-1}$	(15)	63
MDS 2	$-9.95 \cdot 10^{-2}$	(1)	$-2.47 \cdot 10^{-1}$	(58)	235
fmincon	$-2.47 \cdot 10^{-1}$	(36)	$-2.47 \cdot 10^{-1}$	(36)	73
fminimax	$-5.09 \cdot 10^{-2}$	(1)	$-2.47 \cdot 10^{-1}$	(36)	73
HIFOO	$-6.16 \cdot 10^{-2}$	$(1{+}0{+}0)$	$-2.46 \cdot 10^{-1}$ (	(116+0+0)	216

Table 4: VTOL helicopter stabilization

## 6.3 B-767 airplane

Our last example taken from [18] (system AC10) is of higher order (55 states), with two controlled inputs and two measured outputs and describes a modified Boeing B-767 at flutter condition. 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.

#### 6.3.1 Non-smooth optimization algorithm (variants I and II)

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

Table 5. D-707 an plane stabilization					
Alg. Variant	Ι	II (with BFGS)			
first $\alpha < 0$ (iter.)	$-2.37 \cdot 10^{-2} (1)$	$-2.36 \cdot 10^{-2} (1)$			
final $\alpha$ (iter.)	$-7.99 \cdot 10^{-2} (99)$	$-3.50 \cdot 10^{-2} (29)$			
fun. eval.	387	111			
final $\theta$	$-8.00 \cdot 10^{-4}$	$-8.70 \cdot 10^{-6}$			

Table 5: B-767 airplane stabilization

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  $(\lambda_1 = -7.992 \cdot 10^{-2} + 4.912 \cdot 10^{-1}i \text{ and } \lambda_2 = \bar{\lambda}_1)$ ; another pair is very close  $(\lambda_3 = -7.996 \cdot 10^{-2} + 4.892 \cdot 10^{-1}i \text{ and } \lambda_4 = \bar{\lambda}_3)$ . The associated subgradients are nearly opposite matrices which explains the small final value of the optimality function  $\theta$ .

#### 6.3.2 Other algorithms

See table 6.

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

Table 6: B-767 airplane stabilization

## 6.4 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 3 input, 3 output stabilizing PID controller for an open-loop marginally unstable aircraft model (AC2 from [18]), given as:

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

where  $K_P$ ,  $K_I$ ,  $K_D \in \mathbb{R}^{2 \times 2}$  and  $\varepsilon > 0$ . The algorithm is initialized with  $K_P = 0$ ,  $K_I = K_D = I_3$  and  $\varepsilon = 10^{-3}$ . The resulting closed-loop is unstable  $(\alpha(A_c(K)) = 8.06)$ . The algorithm (variant I) finds a stabilizing PID controller after 2 iterations. It stops after 42 iterations, with:

$$K_P = \begin{bmatrix} 72.23 & 4.62 & -28.69\\ 17.89 & -3.93 & 33.70\\ 33.65 & 13.17 & 114.77 \end{bmatrix},$$
  

$$K_I = \begin{bmatrix} -41.38 & 3.80 & -102.45\\ 23.93 & -3.82 & 42.75\\ 11.79 & 18.42 & -20.37 \end{bmatrix},$$
  

$$K_D = \begin{bmatrix} 0.91 & 0 & 0\\ 0 & 0.91 & 0\\ 0 & 0 & 0.91 \end{bmatrix}, \varepsilon = 9.55 \cdot 10^{-4}$$

The final closed-loop spectral abscissa is  $\alpha = -6.03 \cdot 10^{-1}$ . Notice that  $K_D$  and  $\varepsilon$  are nearly unchanged.

## 6.5 Conclusion

Formulated as an optimization program, static or fixed-structure output feedback stabilization has been solved for several case studies from the literature. The proposed non-smooth algorithm addresses the non-smoothness 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 very few non-smooth steps will generally yield a stabilizing static controller. Our framework is generic enough to handle realistic stabilization problems with structured compensators. The two proposed variants are deterministic and numerically efficient, with significantly fewer evaluations of the spectral abscissa than MDS or HIFOO. Finally, although BFGS is designed for smooth optimization, we noticed that it performs quite well for static output feedback stabilization (in variant II, in the initial phase of HIFOO and in the Matlab optimization tools), agreeing with [8].

## References

- [1] P. Apkarian and D. Noll. Controller design via nonsmooth multidirectional search. *SIAM Journal on Control and Optimization*, 44(6):1923–1949, 2006.
- [2] P. Apkarian and D. Noll. Nonsmooth  $H_{\infty}$  synthesis. *IEEE Transactions* on Automatic Control, 51(1):71–86, 2006.
- [3] P. Apkarian and D. Noll. Nonsmooth optimization for multiband frequency domain control design. *Automatica*, In Press, Corrected Proof, 2006.
- [4] P. Apkarian and D. Noll. Nonsmooth optimization for multidisk  $H_{\infty}$  synthesis. European Journal of Control, 12(3):229–244, 2006.
- [5] 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.
- [6] V. Blondel and J. Tsitsiklis. NP-hardness of some linear control design problems. SIAM Journal on Control and Optimization, 35(6):2118–2127, 1997.
- [7] V. Bompart, D. Noll, and P. Apkarian. Second-order nonsmooth optimization for  $H_{\infty}$  synthesis. In 5th IFAC Symposium on Robust Control Design, Toulouse, France, July 2006.
- [8] J. V. Burke, D. Henrion, A. S. Lewis, and M. L. Overton. HIFOO A MATLAB package for fixed-order controller design and  $H_{\infty}$  optimization. In 5th IFAC Symposium on Robust Control Design, Toulouse, France, July 2006.

- J. V. Burke, A. S. Lewis, and M. L. Overton. Optimizing matrix stability. *Proceedings of the American Mathematical Society*, 129:1635–1642, 2001.
- [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] F. H. Clarke. Optimization and Nonsmooth Analysis. Canadian Math. Soc. Series. John Wiley & Sons, New York, 1983.
- [16] J. B. Hiriart-Urruty and C. Lemaréchal. Convex Analysis and Minimization Algorithms II. Springer-Verlag, 1993.
- [17] K. C. Kiwiel. Methods of Descent for Nondifferentiable Optimization, volume 1133 of Lecture Notes in Mathematics 1133. Springer-Verlag, 1985.
- [18] F. Leibfritz and W. Lipinski. Description of the benchmark examples in  $\text{COMPl}_e$ ib 1.0. Technical report, University of Trier, 2003.
- [19] The MathWorks. Optimization Toolbox User's Guide, march 2006. Version 3.
- [20] R. J. Ober. Balanced parametrizations of classes of linear systems. SIAM Journal on Control and Optimization, 29:1251–1287, 1991.
- [21] E. Polak. On the mathematical foundations of nondifferentiable optimization in engineering design. SIAM Review, 29(1):21–89, 1987.

- [22] E. Polak. Optimization: Algorithms and Consistent Approximations. Springer-Verlag, 1997.
- [23] V. Torczon. Multi-Directional Search: A Direct Search Algorithm for Parallel Machines. PhD thesis, Department of Mathematical Sciences, Rice University, Houston, Texas, May 1989.
- [24] V. Torczon. On the convergence of the multidirectional search algorithm. SIAM Journal on Optimization, 1(1):123–145, 1991.