SIAM J. CONTROL OPTIM.
Vol. 44, No. 6, pp. 1923–1949

CONTROLLER DESIGN VIA NONSMOOTH MULTIDIRECTIONAL SEARCH*

PIERRE APKARIAN† AND DOMINIKUS NOLL‡

Abstract. We propose an algorithm which combines multidirectional search (MDS) with nonsmooth optimization techniques to solve difficult problems in automatic control. Applications include static and fixed-order output feedback controller design, simultaneous stabilization, H_2/H_∞-synthesis, and much else. We show how to combine direct search techniques with nonsmooth descent steps in order to obtain convergence certificates in the presence of nonsmoothness. Our technique is efficient when small and medium size controllers for plants with large state dimension are sought. Our numerical testing includes several benchmark examples. For instance, our algorithm needs 0.41 s to compute a static output feedback stabilizing controller for the Boeing 767 flutter benchmark problem [E. E. J. Davison, *IFAC Technical Committee Reports*, Pergamon Press, Oxford, 1990], a system with 55 states. The first static controller without performance specifications for this system was obtained in [J. Burke, A. Lewis, and M. Overton, *SIAM J. Optim.*, 15 (2003), pp. 751–779].

Key words. *NP*-hard design problems, static output feedback, fixed-order synthesis, simultaneous stabilization, mixed H_2/H_∞-synthesis, pattern search algorithm, moving polytope, nonsmooth analysis, spectral bundle method, ε-subgradients, bilinear matrix inequality (BMI)

AMS subject classifications. 93B36, 93B40, 93B50, 93B51, 90C22, 90C56, 90C34, 90C26, 49J52, 49J35

DOI. 10.1137/S0363012904441684

1. Introduction.

Pattern search or moving polytope methods belong to a large class of derivative-free optimization methods referred to as direct search (DS) techniques. In this paper, we present a nonsmooth modification of Virginia Torczon's multidirectional search (MDS) [66, 67] algorithm and apply it to a broad class of problems in automatic control. We aim at several nonconvex and even *NP*-hard problems, for which LMI techniques or algebraic Riccati equations are impractical. In particular, we propose algorithmic solutions for static and fixed-order output feedback control, simultaneous stabilization problems, and mixed H_2/H_∞-control.

1.1. Direct search methods.

The idea of DS methods can be traced back to the pioneering work of Box [11] and Hook and Jeeves [37], who first coined the term "direct search." The MDS algorithm is due to Torczon [66, 67] and is directly inspired by the work of Spendley, Hext, and Himsworth [63], and the popular method of Nelder and Mead [55]. MDS significantly revived the interest in DS methods, because it came with a sound convergence theory [66]. This is in contrast with the Nelder–Mead algorithm, which may fail to converge even for smooth convex objective functions; see [52]. Later, Torczon generalized her work to the entire class of DS techniques [67].

*Received by the editors March 4, 2004; accepted for publication (in revised form) April 22, 2005; published electronically January 6, 2006.
http://www.siam.org/journals/sicon/44-6/44168.html
†ONERA-CERT, Centre d'études et de recherche de Toulouse, Control System Department, 2 av. Edouard Belin, 31055 Toulouse, France, and Mathématiques pour l'Industrie et la Physique, Université Paul Sabatier, Toulouse, France (apkarian@cert.fr).
‡Université Paul Sabatier, Mathématiques pour l'Industrie et la Physique, 118, route de Narbonne, CNRS UMR 5640, 31062 Toulouse, France (noll@mip.ups-tlse.fr).

DS methods compute local minima of unconstrained optimization programs:

$$\text{minimize} \quad f(x),\ x \in \mathbb{R}^n, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a $\mathcal{C}^1$ function. DS techniques are *derivative-free* in the sense that they do not require gradient information in order to *compute* descent steps. This is a convenient feature if derivatives or their finite difference approximations are not available and/or too expensive to compute or when automatic differentiation is hindered by the presence of for loops in the function evaluation.

However, contrary to what the name suggests, the term derivative-free does not mean that derivatives do not altogether exist. On the contrary, DS methods are designed for $\mathcal{C}^1$ functions, and their convergence theory is heavily based on differentiability [67]. Problems encountered when search methods are used with genuinely nonsmooth criteria are discussed in [46].

DS techniques can also be used for constrained optimization programs. The ideas to attack those range from quadratic or exact penalty techniques over barrier functions to the augmented Lagrangian method.

1.2. Nonsmoothness. In the present paper, we apply the ideas of MDS to several constrained and unconstrained optimization problems in automatic control, where nonsmooth functions like the maximum eigenvalue function, the spectral abscissa, the distance to instability, and the H_∞-norm arise naturally. Due to the failure of convergence under nonsmoothness, DS methods may not be applied in their original form and additional tools from nonsmooth optimization are required. An algorithm combining both ideas is what will eventually emerge. Using nonsmooth techniques in control design is not altogether a new idea; see, e.g., [62, 61, 44, 53, 40]. What has not been tried before is combining nonsmooth techniques with DS strategies.

The lack of a convergence certificate under nonsmoothness has not prevented practitioners from applying DS methods in such cases. It is often argued that the contingency of a failure due to nonsmoothness is a remote one. The argument on which such reasoning is usually based is that even nonsmooth functions are, as a rule, almost everywhere differentiable, so that nonsmooth points are never encountered in practice. Our present work reveals this as an illusory argument. Nonsmoothness may and will cause failure of DS techniques, as we demonstrate by several striking examples.

In response, we show how MDS can be combined with nonsmooth descent steps in order to avoid the typical failure, where simplices shrink and iterates converge to a nonstationary point, which we also call a *dead point*. It is crucial to be able to distinguish dead points from local minima, and this is done by adding a nonsmooth stopping test to the usual hand tools of MDS. Such a test either indicates success or allows one to escape from a dead point, keeping the search algorithm moving.

However, this is not the end of the story. Calling for a nonsmooth stopping test whenever the simplex shrinks below a certain threshold may keep MDS moving, but it is not strong enough to ensure convergence. In order to get a convergence certificate in the presence of nonsmoothness, we need to supply MDS with *quantified descent steps* similar to those employed by nonsmooth optimization techniques to ensure convergence. We will refer to these two types of nonsmooth substrata to MDS as *crisis intervention* and *crisis prevention*. While crisis intervention is done only occasionally, being therefore less costly, crisis prevention is more complex, as it requires that the nonsmooth technique assists the search during the whole process.

We will indicate in which way crisis intervention and crisis prevention should be organized for application in automatic control, but our approach is in principle open to more general nonsmooth objectives.

We mention that a different approach to integrate nonsmoothness into MDS was recently proposed by Audet and Dennis [7, 1] for general locally Lipschitz functions. Their approach and ours are somewhat complementary. While we are more specific as far as the applications are concerned, our combined method can accommodate composite functions with the spectral abscissa, which are not even locally Lipschitz smooth. Also, our intervention technique is applicable to other derivative-free method, like for instance the wedge algorithm of Marrazzi and Nocedal [51].

The paper is organized as follows. We start with an introductory section 2, where three nonsmooth criteria are discussed. We proceed with the central sections 3 and 4, where we indicate why and in which form nonsmoothness arises in automatic control. In section 5 we briefly recall the mode of operation of MDS, including the possibility of the two types of intervention steps, by which the failure at dead points can be avoided. In section 6 we proceed to the implementation of *crisis intervention* and *crisis prevention* for nonsmooth objectives like the maximum eigenvalue function, the spectral abscissa, and the H_∞-norm. Crisis intervention is discussed in section 6, while the more sophisticated crisis prevention is discussed in section 7. Numerical experiments to validate the proposed tools and techniques are discussed in section 8 for a rich set of control applications.

1.3. Notation. Notation from convex and nonsmooth analysis are covered by [35] and [22]. We let $\mathbb{S}^m$ denote the set of $m \times m$ symmetric matrices, equipped with the scalar product $\langle X, Y\rangle = X \cdot Y = \operatorname{Tr}(XY)$. Let $\mathbb{M}_n$ be the space of real $n \times n$ matrices, $\mathbb{M}_{n,m}$ the space of $n \times m$ matrices, equipped with the corresponding scalar product $\langle X, Y\rangle = \operatorname{Tr}(X^{\mathrm{T}}Y)$, where X^{T} is the transpose of the matrix X, $\operatorname{Tr} X$ its trace. For complex matrices X^{H} stands for its transconjugate. For Hermitian or symmetric matrices, $X \succ Y$ means that $X - Y$ is positive definite, $X \succeq Y$ that $X - Y$ is positive semidefinite. We shall use superscripts for the iteration index, lower scripts to indicate vector components. Our notation from feedback control is standard and follows, e.g., [14].

2. Examples of nonsmooth functions in control. In this section we briefly discuss several nonsmooth functions arising in automatic control applications.

Our first example is the maximum eigenvalue function $\lambda_1 : \mathbb{S}^m \to \mathbb{R}$, defined on the space $\mathbb{S}^m$ of symmetric $m \times m$ matrices. We will use composite functions of the form $f(x) = \lambda_1(\mathcal{B}(x))$, where $\mathcal{B} : \mathbb{R}^n \to \mathbb{S}^n$ is usually a bilinear, quadratic, or class $\mathcal{C}^2$-operator. The interest in $f = \lambda_1 \circ \mathcal{B}$ stems from the fact that the matrix inequality $\mathcal{B}(x) \preceq 0$ is equivalent to the scalar constraint $f(x) \le 0$. Notice that λ_1 is convex, which gives f a lot of structure. For instance, the Clarke subdifferential of f (cf. [22]) is the set

$$\partial f(x) = \mathcal{B}'(x)^\star[\partial\lambda_1(\mathcal{B}(x))] = \{\mathcal{B}'(x)^\star Z : Z = QYQ^{\mathrm{T}},\, Y \succeq 0, \operatorname{Tr}(Y) = 1\}, \tag{2}$$

where the columns of the matrix Q form an orthonormal basis of the eigenspace of $\lambda_1(\mathcal{B}(x))$. Here and in what follows, $\mathcal{B}'(x)$ denotes the derivative of $\mathcal{B}$ at x, understood as a linear operator $\mathbb{R}^n \to \mathbb{S}^m$, while $\mathcal{B}'(x)^\star$ denotes its adjoint, mapping $\mathbb{S}^m \to \mathbb{R}^n$. A case of special interest is when $\mathcal{B}$ is quadratic:

$$\mathcal{B}(x) = A_0 + \sum_{i=1}^{n} x_i A_i + \sum_{i,j=1}^{n} x_i x_j B_{ij}.$$

Then $\mathcal{B}'(x)d = \sum_{i=1}^{n} d_i A_i + \sum_{i,j=1}^{n} (x_i d_j + x_j d_i) B_{ij}$, and the adjoint is obtained as

$$(\mathcal{B}'(x)^\star Z)_i \,=\, \left(A_i + \sum_{j=1}^{n} x_j B_{ij} + x_j B_{ji} \right) \cdot Z.$$

Our second example of a nonsmooth function is the pseudospectral abscissa. Following Trefethen [68], the pseudospectral abscissa of a matrix $A \in \mathbb{M}_m$ is defined as

$$\alpha_\varepsilon(A) \,=\, \max \left\{ \operatorname{Re} \lambda : \lambda \in \Lambda_\varepsilon(A) \right\},$$

where Λ_ε is the ε-pseudospectrum of A, that is, the set of all eigenvalues of matrices $A + E$ with euclidean norm $\|E\| \leq \varepsilon$. For $\varepsilon = 0$ we recover $\alpha = \alpha_0$, the spectral abscissa, $\Lambda = \Lambda_0$ the spectrum of A. Our second class of nonsmooth functions is now of the form $g(x) = \alpha\left(\mathcal{A}(x)\right)$ or $g(x) = \alpha_\varepsilon\left(\mathcal{A}(x)\right)$, where $\mathcal{A}$ is a smooth operator defined for $x \in \mathbb{R}^n$ with values in the matrix space $\mathbb{M}_m$. Use of this function for static feedback synthesis was first proposed by Burke, Lewis, and Overton in [17, 18]. We will discuss this particular application in sections 6 and 8.1. The interest in $g = \alpha \circ \mathcal{A}$ is obviously due to the fact that $\mathcal{A}(x) \in \mathbb{M}_m$ is Hurwitz if and only if $g(x) < 0$. Notice that $g = \alpha \circ \mathcal{A}$ is smooth at x when $\alpha(\mathcal{A}(x)) = \operatorname{Re} \lambda_i(\mathcal{A}(x))$ for a single eigenvalue, where complex conjugate pairs are counted once. On the other hand, g is nonsmooth in general for multiple eigenvalues. What is worse is that neither $g = \alpha \circ \mathcal{A}$ nor $g = \alpha_\varepsilon \circ \mathcal{A}$ is locally Lipschitz function in general [17], which makes the functions somewhat delicate to handle.

Notice that function evaluation for α_ε may be based on the criss-cross method in [19], a generically globally quadratically convergent algorithm, which bears some resemblance with the Hamiltonian algorithm [12] to compute the H_∞-norm. For smooth points x, the criss-cross algorithm computes the gradient, while it still gives a subgradient of $\alpha_\varepsilon \circ \mathcal{A}$ at x if x is a nonsmooth point.

Our third example is the H_∞-norm. Notice that the stability requirement $\alpha_\varepsilon(A) < 0$ is equivalent to the estimate $\|(sI - A)^{-1}\|_\infty < \varepsilon^{-1}$. This means that α_ε could be avoided and replaced by composite functions of the H_∞-norm.

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

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

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

$$\begin{aligned}\phi(H) = \operatorname{Re}\left(u^{\mathrm{H}} H(j\omega) v\right) \,&=\, \|G\|_\infty^{-1} \operatorname{Re} \operatorname{Tr} v v^{\mathrm{H}} G(j\omega)^{\mathrm{H}} H(j\omega) \\ &= \|G\|_\infty^{-1} \operatorname{Re} \operatorname{Tr} G(j\omega)^{\mathrm{H}} u u^{\mathrm{H}} H(j\omega)\end{aligned}$$

is continuous on the space $\mathbf{H}_\infty$ of stable transfer functions and is a subgradient of $\|\cdot\|_\infty$ at G [13]. More generally, assume the columns of Q_u form an orthonormal basis of the eigenspace of $G(j\omega)G(j\omega)^{\mathrm{H}}$ associated with the largest eigenvalue $\lambda_1\left(G(j\omega)G(j\omega)^{\mathrm{H}}\right) = \overline{\sigma}(G(j\omega))^2$, and assume the columns of Q_v form an orthonormal basis of the eigenspace of $G(j\omega)^H G(j\omega)$, associated with the same eigenvalue; then for every $Y_v \succeq 0$, $Y_u \succeq 0$ with $\operatorname{Tr}(Y_v) = 1$ and $\operatorname{Tr}(Y_u) = 1$,

$$\phi(H) = \|G\|_\infty^{-1} \operatorname{Re} \operatorname{Tr} Q_v Y_v Q_v^{\mathrm{H}} G(j\omega)^{\mathrm{H}} H(j\omega) = \|G\|_\infty^{-1} \operatorname{Re} \operatorname{Tr} G(j\omega)^{\mathrm{H}} Q_u Y_u Q_u^{\mathrm{H}} H(j\omega) \tag{3}$$

are subgradients of $\|\cdot\|_\infty$ at G, where Y_v and Y_u are (complex) Hermitian matrices. Finally, assume that $G(s)$ is rational, and that there exist finitely many frequencies $\omega_1, \dots, \omega_p$ where the supremum $\|G\|_\infty = \overline{\sigma}(G(j\omega_\nu))$ is attained. Then the subgradients of $\|\cdot\|_\infty$ at G are precisely of the form

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

where the columns of Q_ν form an orthonormal basis of the eigenspace of $G(j\omega_\nu)G(j\omega_\nu)^{\mathrm{H}}$ associated with the leading eigenvalue $\|G\|_\infty^2$, and where $Y_\nu \succeq 0$, $\sum_{\nu=1}^{p} \operatorname{Tr}(Y_\nu) = 1$. See [22, Prop. 2.3.12 and Thm. 2.8.2] for this.

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

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

where $\partial\|\cdot\|_\infty$ is the subdifferential of the H_∞-norm above. In section 6 we will compute this adjoint $\mathcal{G}'(x)^\star$ in a more specific situation. Suitable chain rules for this case are covered by [22, sect. 2.3].

3. Nonsmoothness in control. In automatic control, difficulties with computing derivatives arise frequently. This happens, for instance, when design specifications include time-domain constraints (settling-time, overshoot) and function evaluations depend on simulations or experiments. But even genuine nonsmoothness arises when criteria like the maximum eigenvalue function, the spectral abscissa, or the H_∞-norm are optimized. For a large class of problems in robust control theory, these nonsmooth criteria can be avoided since a smooth reformulation is available. The price to pay is a significant increase of the number of variables. There are situations where this becomes the major impediment to currently available optimization codes.

The situation we have in mind occurs for problems where bilinear matrix inequalities (BMIs) arise:

$$\begin{array}{ll} \text{minimize} & a^{\mathrm{T}}x + b^{\mathrm{T}}y, \quad x \in \mathbb{R}^r, \quad y \in \mathbb{R}^s \\ \text{subject to} & A_0 + \sum_{i=1}^{r} x_i A_i + \sum_{j=1}^{s} y_j B_j + \sum_{\ell=1}^{r}\sum_{k=1}^{s} x_\ell y_k C_{\ell k} \preceq 0, \end{array} \tag{4}$$

with $a \in \mathbb{R}^r$, $b \in \mathbb{R}^s$ and $A_i, B_j, C_{\ell k} \in \mathbb{S}^m$ given. Typically in (4) the decision vector splits into $x \in \mathbb{R}^r$, which gathers all free components or gains in the controller to be designed, while $y \in \mathbb{R}^s$ regroups the Lyapunov variables. All our examples discussed in section 8 may be brought to this form. In order to understand the problem better, let us discuss an application of particular importance.

3.1. Static output-feedback synthesis. It is well known that static output H_2- or H_∞-synthesis are NP-hard problems (cf. [56]), which may be cast as BMI-optimization programs. Given the plant

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}$$

with $x \in \mathbb{R}^{n_1}$, $u \in \mathbb{R}^{m_2}$, $w \in \mathbb{R}^{m_1}$, $y \in \mathbb{R}^{p_2}$, $z \in \mathbb{R}^{p_1}$, we ask for a static feedback control law $u = Ky$ such that the closed-loop system is internally stable and, moreover, a suitable operator norm of the performance channel $w \to z$ is minimized. For the H_∞-norm, the existence of such a K with the norm estimate $\|T_{w\to z}(K)\|_\infty < \gamma$ is equivalent to the existence of a Lyapunov matrix $Y \in \mathbb{S}^{n_1}$ satisfying $Y \succ 0$ and

(5)
$$\begin{bmatrix} (A+B_2KC_2)^{\mathrm{T}}Y + Y(A+B_2KC_2) & Y(B_1+B_2KD_{21}) & (C_1+D_{12}KC_2)^{\mathrm{T}} \\ * & -\gamma I & (D_{11}+D_{12}KD_{21})^{\mathrm{T}} \\ * & * & -\gamma I \end{bmatrix} \prec 0.$$

If we optimize the gain γ, we obtain a BMI program (4) with unknown variables $\gamma \in \mathbb{R}$, $K \in \mathbb{R}^{m_2 \times p_2}$, and $Y \in \mathbb{S}^{n_1}$. We may identify $x \in \mathbb{R}^r$ with the true decision variables γ and K, so $r = 1 + m_2 p_2$, while $y \in \mathbb{R}^s$ gathers the Lyapunov variables Y, so $s = n_1(n_1+1)/2$. If the system size n_1 is large, the number of Lyapunov variables is dominant. A somewhat extreme example is the Boeing 767 under flutter condition (AC10), treated in section 8, where $n_1 = 55$, while $m_2 = p_2 = 2$. Here the BMI problem has 1490 variables, while there are only 4 true decision parameters (see [47, 24] for details).

The BMI problem (4) can be handled via smooth techniques by exploiting stationarity conditions [41] or via interior-point methods [36] and [49, 48]. An alternative is to use augmented Lagrangian techniques like Mosheyev and Zibulevsky [54]; see also [45] and [65]. Their approach extends naturally to nonlinear SDPs like (4). Unfortunately, all these approaches lead to large-size optimization problems even for control problems of moderate sizes due to the presence of Lyapunov variables y. One way to partly alleviate the difficulty in the nonlinear case is to use the projection lemma [27], whenever possible, to reduce at least the number of variables in x. The new cast is then a program with LMI constraints in tandem with nonlinear equality constraints:

$$\min \left\{ c^{\mathrm{T}} y : A_0 + \sum_{i=1}^{r} y_i A_i \preceq 0,\ h(y) = 0 \right\} \tag{6}$$

where $h : \mathbb{R}^n \to \mathbb{R}^p$ represents a finite number of nonlinear equality constraints. As suggested by our notation, the projection lemma reduces the x part in (4) to size $r = 1$ (to size $r = 0$ for pure stabilization), but gives only a slight reduction of the number s of Lyapunov variables y. The additional benefit of the projection lemma is that it avoids the redundancies of the controller state-space representations. For static output-feedback stabilization (northwest $(1,1)$ block in inequality (5)), a controller-free version is as follows: A stabilizing static controller K exists if there exist Lyapunov matrices Y_1, $Y_2 \in \mathbb{S}^{n_1}$ such that

$$\begin{aligned}\mathcal{N}_Q^{\mathrm{T}}\left(A^{\mathrm{T}}Y_1 + Y_1 A\right)\mathcal{N}_Q &\prec 0,\\ \mathcal{N}_P^{\mathrm{T}}\left(AY_2 + Y_2 A^{\mathrm{T}}\right)\mathcal{N}_P &\prec 0,\\ \begin{bmatrix} Y_1 & I \\ I & Y_2 \end{bmatrix} \succ 0, \qquad Y_1 Y_2 - I &= 0,\end{aligned}$$

where $\mathcal{N}_P$ and $\mathcal{N}_Q$ are bases of the nullspaces of C and B^{T}, respectively. A version including H_∞-norm performance has the same form and may be found, e.g., in [59].

Different techniques have been developed to solve problems (5), (6) or problems with more general matrix inequality and equality constraints. Leibfritz and Mustafa [49, 48] use interior-point techniques in tandem with ideas from sequential quadratic programming to separate Lyapunov and true decision variables in the tangent programs. A successive SDP approach is given in [25] and an augmented Lagrangian approach in [6]. These techniques are supported by local and global convergence theory [59], but have shown some limitations:

- Our experiments have revealed size limitations to about 1500 variables [5]. This allows solving problems with up to $n_1 = 40$ states.
- The transformation of (4) into (6) is not always possible. Only a restricted and well-identified class of problems is amenable to the projection lemma. A prominent case where this is *not* possible is simultaneous stabilization, considered in section 8.

In our testing, we have compared the nonsmooth MDS method to the BMI-based methods in [5, 65] (see the corresponding column in Table 2).

4. Nonsmoothness by avoiding Lyapunov variables. For large systems, the number $s = n_1(n_1+1)/2$ of Lyapunov variables y is a serious obstacle to the BMI-optimization approach (4) or (6). It seems natural to consider alternatives where Lyapunov variables y can be avoided, so that the optimization concentrates on the true decision variables $x = (\gamma, K)$. This is possible if one accepts nonsmooth optimization programs. Here we propose to replace (5) by the following constrained program:

$$\begin{array}{ll} \text{minimize} & \|T_{w\to z}(K,s)\|_\infty \\ \text{subject to} & \alpha_\varepsilon\left(A + B_2 K C_2\right) \le 0, \\ & K \in \mathbb{R}^{m_2 \times p_2}, \end{array} \tag{7}$$

for fixed $\varepsilon \ge 0$, where the performance channel $w \to z$ is specified by the transfer function

$$\begin{gathered} T_{w\to z}(K,s) = \mathcal{C}(K)\left(sI - \mathcal{A}(K)\right)^{-1}\mathcal{B}(K) + \mathcal{D}(K), \\ \mathcal{A}(K) := A + B_2 K C_2,\ \mathcal{B}(K) := B_1 + B_2 K D_{21},\ \mathcal{C}(K) := C_1 + D_{12} K C_2, \\ \mathcal{D}(K) := D_{11} + D_{12} K D_{21}. \end{gathered} \tag{8}$$

An alternative is the constrained program

$$\begin{array}{ll} \text{minimize} & \|T_{w\to z}(K,s)\|_\infty \\ \text{subject to} & \left\|(sI - \mathcal{A}(K))^{-1}\right\|_\infty \le \varepsilon^{-1}, \\ & K \in \mathbb{R}^{m_2 \times p_2}. \end{array} \tag{9}$$

Notice that in both programs, the controller K has to be stabilizing, or what is the same, iterates have to be feasible. This requires a feasible initial point K^0, which we compute by the unconstrained optimization program (with $\varepsilon \ge 0$ fixed):

$$\text{minimize} \quad \alpha_\varepsilon\left(A + B_2 K C_2\right), K \in \mathbb{R}^{m_2 \times p_2}. \tag{10}$$

Using (10) for static feedback control has first been proposed in [17, 19].

Remark. Notice an important difference between programs like (7), (9) and program (10), used to initialize the others. While all programs encountered are nonconvex and often exhibit multiple local minima, it is usually satisfactory to accept a local minimum of the H_∞-norm in (7), (9), because the controller K is always stabilizing. This is different in program (10), where a local minimum K is useless as long as it satisfies $\alpha\left(A + B_2KC_2\right) \geq 0$, because it does not provide a stabilizing controller. In such a case, we have to restart the algorithm. Notice, however, that this does not mean that we require the full machinery of a global optimization technique, because we are not interested in *the* global minimum of (10). A value $\alpha < 0$ is all what is wanted. □

Similar nonsmooth formulations can be obtained for various other robust control problems, such as static and fixed-order stabilization, H_2- and H_∞-synthesis problems, simultaneous (multimodel) synthesis problems, control design with fixed structure controllers, robust synthesis and synthesis problems involving scaling and multipliers, and linear parameter-varying syntheses, to cite just a few.

Some of these problems are investigated in section 8. Our experiments seem to indicate that as soon as Lyapunov variables y in (4) dominate, nonsmooth programs like (7), (9) in conjunction with nonsmooth techniques are very attractive. The MDS algorithm and more general DS or pattern search techniques, supplemented by nonsmooth techniques, are serious alternatives to BMI- or LMI-based methods. This is most promising when the number of controller variables $x = (\gamma, K)$ is small. In our experiments, small means not more than 30–35 controller variables x. This situation occurs when simple controllers for large systems are sought. For problems with high-order controllers, a pure nonsmooth approach is inevitable. This is investigated in [3].

Remark. We end this paragraph by pointing the reader to a very important feature of optimization programs (7), (9), (10), which seem to invite techniques like MDS. Namely, in MDS and other search algorithms exact function evaluations can often be avoided. All that is needed is that we be able to compare the value of the objective at the different nodes to the current best value. This is in perfect agreement with function evaluations for α_ε, λ_1 and the H_∞-norm, which are all based on iterative procedures. For instance, the bisection algorithm for the H_∞-norm [12] need not be run to completion, a premature stopping criterion can be exploited to enhance efficiency. This renders our present approach open to larger problem sizes. □

5. The MDS algorithm with nonsmooth steps. In this section we give a description of the MDS algorithm and indicate in which way a nonsmooth step may be added to cope with nonsmoothness. For an in-depth discussion of MDS in the smooth case the interested reader is referred to [66].

The MDS algorithm requires a "seed" or base point v_0 and an initial simplex S in $\mathbb{R}^n$ with vertices $v_0, v_1, \ldots, v_n$. The vertices are then relabeled so that v_0 becomes the best vertex, that is, $f(v_0) \leq f(v_i)$ for $i = 1, \ldots, n$. The initial S is chosen from one of three different shapes; see Figure 1. The scaled simplex is used when prior knowledge on the problem scaling is available, but right-angled and regular simplices are generally preferred in the absence of information. The algorithm updates the current simplex S into a new simplex S^+ by performing three types of operations, which drive the search for a better point: reflection, expansion, and contraction; see Figure 2. First vertices $v_1, \ldots, v_n$ are reflected through the current best vertex v_0 to give $r_1, \ldots, r_n$. If a reflected vertex r_i gives a better function value than v_0, the algorithm tries an expansion step. This is done by increasing the distance between v_0 and r_i for

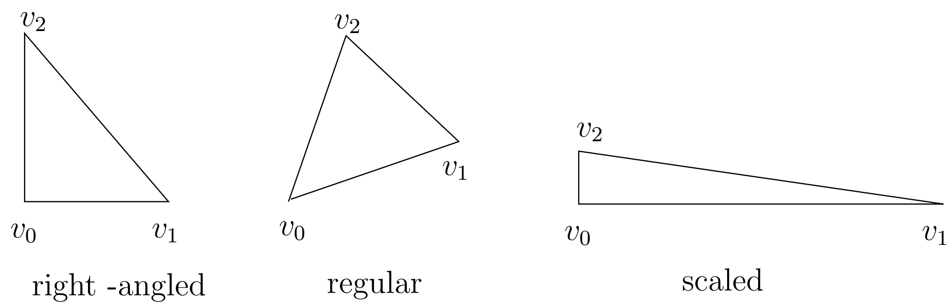

FIG. 1. *Selection of initial simplex.*

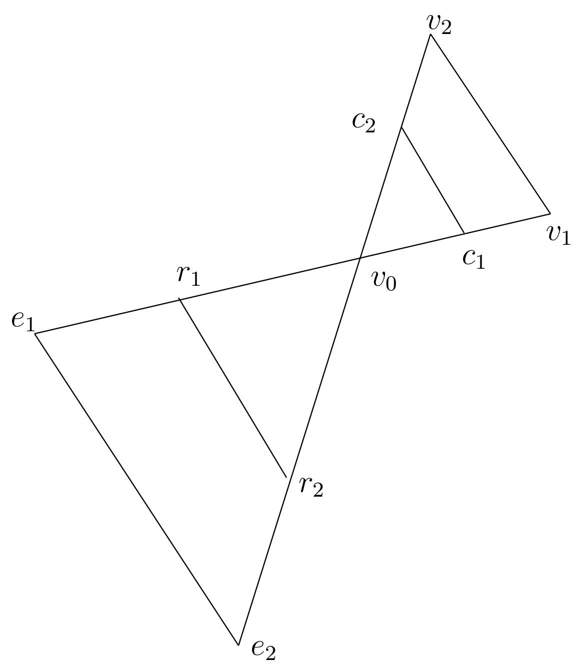

FIG. 2. *Reflection, expansion, and contraction of current simplex.*

$i = 1, \ldots, n$ and yields new expansion vertices e_i for $i = 1, \ldots, n$. The current simplex S is then replaced by either $S^+ = \{v_0, r_1, \ldots, r_n\}$ or $S^+ = \{v_0, e_1, \ldots, e_n\}$, depending on whether the best point was among the reflection or expansion vertices. If neither reflection nor expansion provide a point better than v_0, a contraction step is performed. This is done by decreasing the distances from v_0 to $v_1, \ldots, v_n$. If a point better than v_0 is found among the contraction vertices $c_1, \ldots, c_n$, the simplex S is replaced by $S^+ = \{v_0, c_1, \ldots, c_n\}$. To complete one iteration (or sweep) of the algorithm, v_0^+ is taken to be the best vertex of S^+.

In the presence of nonsmoothness, we endow the MDS algorithm with a fourth element. MDS may take a nonsmooth step w away from the current best node v_0 under consideration. In our applications, w will typically be the result of a nonsmooth descent step away from v_0, computed at the beginning of each sweep of MDS. If the sweep produces a new vertex v_0^+ better than w, MDS ignores w and keeps moving as

1. Select initial simplex $S = \{v_0, \dots, v_n\}$, where v_0 is the best vertex. Fix an expansion factor $\mu \in (1, \infty)$ and a contraction factor $\theta \in (0, 1)$, and an intervention tolerance $\omega > 0$.
2. Given the current simplex S with best vertex v_0, call for a nonsmooth step w if the size of S is below threshold ω. If $w = v_0$ stop at critical point v_0.
3. Perform a reflection step $r_i = v_0 - (v_i - v_0)$. Compute $f(r_i)$.
4. If improvement $f(r_i) < f(v_0)$
   - perform expansion step $e_i = (1 - \mu)v_0 + r_i$. Compute $f(e_i)$.
   - If improvement $f(e_i) < f(v_0)$
     - put $S^+ = \{v_0, e_1 \dots, e_n\}$. Goto step 5.
   - else
     - put $S^+ = \{v_0, r_1 \dots, r_n\}$. Goto step 5.

   else
   - perform contraction step $c_i = (1 + \theta)v_0 - \theta r_i$. Compute $f(c_i)$. Put $S^+ = \{c_0, \dots, c_n\}$.
5. Compare best vertex in S^+ to $f(w)$. If w is better, replace S^+ by new simplex containing w as a vertex. Otherwise accept S^+. Go back to step 2 to loop on.

FIG. 3. *MDS with nonsmooth steps.*

planned. On the other hand, if w is better than all the nodes tested by MDS during reflection, expansion, and contraction, we include w among the vertices of the new simplex S^+. In that event, we have to decide in which way the old vertices produced by MDS are recycled, or whether new nodes need to be created. This will obviously depend on geometrical properties. One possibility is to abandon the worst among the nodes of S^+ found by MDS and add the new node w as best point. If this produces angles below a certain threshold, one has to (partly) abandon S^+ and add new vertices to avoid bad geometry. In such a case, one can also build a completely new simplex with right-angled or regular geometry, using w as seed point. In our tests, we have observed that it is beneficial in such a situation to switch between the geometries (regular, right angled) in order to give MDS some additional help to move on. But all these considerations are clearly heuristic, depend on the context, and will need further testing.

In order to avoid serious slowdown of MDS, the nonsmooth step w is only solicited when the size of the simplex is below a certain threshold ω. Large S indicate that MDS is making good progress, so a costly nonsmooth step should be avoided. The situation we expect is that most of the time the point w is not better than the new best point v_0^+ of S^+ found by MDS. In that case, w plays a role similar to the Cauchy point in trust region methods. That is, it is hardly ever taken as the new iterate, but gives a convergence certificate. In our case, this will be made precise in Theorem 1. The different ways in which w may be computed will be explained subsequently. We sum up the above discussion in the pseudocode shown in Figure 3.

The following sections will show how the nonsmooth steps $v_0 \to w$ may be computed. From step 2 of the algorithm it is clear that the minimal requirement any w should satisfy is that $0 \notin \partial f(v_0)$ should give $f(w) < f(v_0)$, so when $w = v_0$, the algorithm stops with $0 \in \partial f(v_0)$.

The choice of the intervention tolerance ω should be compared to the usual stopping tests for smooth versions of MDS. Modern implementations use the relative size of the current simplex as a stopping test:

$$\frac{1}{\max(1,\, \|v_0\|)} \max_{1\le i\le n} \|v_i - v_0\| < \varepsilon, \tag{11}$$

where v_0 is the current best vertex of $S = \{v_0, \dots, v_n\}$ and $\varepsilon > 0$ is a prescribed tolerance. If a crisis intervention strategy is used, ω should be chosen slightly larger than the size (11). In the case of crisis prevention, an even larger ω is chosen.

The choice of the initial simplex S is a relatively unexplored topic. The convergence proof in [66] requires only that S be nondegenerate, which means that the $n+1$ points $\{v_0, v_1, \dots, v_n\}$ defining the simplex must span $\mathbb{R}^n$. Otherwise, MDS would only search over the subspace spanned by the degenerate simplex.

6. Nonsmooth stopping tests. Our first strategy is crisis intervention and uses a very small threshold ω. This means that the nonsmooth descent step $v_0 \to w$ is called for only when MDS gets stalled. What this essentially amounts to is a nonsmooth optimality test, which will either show that we are at a local minimum (or critical point) or give us a descent step $v_0 \to w$ to escape from the current point v_0, allowing MDS to move on. This strategy is preferable if nonsmooth descent steps are expensive. During the following we compute these steps for the criteria presented in section 2 and for the programs in section 4.

6.1. Maximum eigenvalue function. This case is well known. From the formula (2) of the Clarke subdifferential of $f = \lambda_1 \circ \mathcal{B}$ we see that $0 \in \partial f(x^*)$ if and only if the value t of the following semidefinite program is zero:

$$\min\{t : Q^{\mathrm{T}}[\mathcal{B}'(x^*)d]Q \preceq tI, \|d\| \le 1\}.$$

On the other hand, when the value is negative, the optimal solution (t, d) of this SDP gives the steepest descent direction d for $f = \lambda_1 \circ \mathcal{B}$ at x^*. If x^* is the current best vertex v_0 in MDS, then the nonsmooth stopping test either shows $0 \in \partial f(x^*)$ or produces w with $f(w) < f(x^*)$ of the form $w = x^* + \tau d$, where $\tau > 0$ is found by a suitable line search.

6.2. Spectral abscissa. This is a more difficult case. Consider the minimization program

$$\min_{x\in\mathbb{R}^n} g(x) = \alpha\left(\mathcal{F}(x)\right),$$

where $\mathcal{F} : \mathbb{R}^n \to \mathbb{M}_m$ is smooth. Since α is not even locally Lipschitz in general, we need a more elaborate way to obtain a stopping test.

Suppose MDS gets stalled at x^* and we want to know whether x^* is a local minimum of g or a dead point. We use the following lemma.

LEMMA 1. *Let $F \in \mathbb{M}_m$. Then $\alpha(F) \le t$ if and only if there exists $Y \in \mathbb{S}^m$, $0 \prec Y \prec I$, such that $F^{\mathrm{T}}Y + YF - 2tY \preceq 0$.* □

For a bounded set of matrices F, the condition number of Y is bounded. The inequality $Y \succ 0$ can therefore be replaced by $Y \succeq \theta I$ for a fixed small enough $\theta > 0$, uniformly over all F in that bounded set. Assume now that we have chosen an initial iterate x_0 such that $L = \{x \in \mathbb{R}^n : g(x) \le g(x_0)\}$ is bounded. Since we use a method of descent type, all our iterates x lie in L, so that the condition number of the

Lyapunov matrices Y arising at the corresponding $F = \mathcal{F}(x)$ are uniformly bounded: $\theta I \preceq Y \preceq I$ for some $0 < \theta \ll 1$. This allows us to consider the optimization program

$$\text{(P)} \qquad \begin{array}{ll} \text{minimize} & t \\ \text{subject to} & Y \succeq \theta I,\ Y \preceq I, \\ & \mathcal{F}(x)^{\mathrm{T}} Y + Y\mathcal{F}(x) - 2tY \preceq 0, \end{array}$$

with decision vector $(x, t, Y) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{S}^m$. Let $x^* \in L$. Define $F^* = \mathcal{F}(x^*)$ and $t^* = \alpha(F^*)$. Correspondingly, compute Y^* with $\theta I \preceq Y^* \preceq I$ such that $F^{*T} Y^* + Y^* F^* - 2t^* Y^* \preceq 0$. As a consequence of Lemma 1 we have the following proposition.

Proposition 1. *$x^* \in L$ is a local minimum of $g = \alpha \circ \mathcal{F}$ if and only if (x^*, t^*, Y^*) is a local minimum of program* (P). □

In order to decide whether the latter is the case, we use a general result from [9]. Define $f(x, t, Y) = t$ and

$$\mathcal{G}(x,t,Y) = \begin{bmatrix} Y - I & 0 & 0 \\ 0 & \theta I - Y & 0 \\ 0 & 0 & \mathcal{F}(x)^{\mathrm{T}} Y + Y\mathcal{F}(x) - 2tY \end{bmatrix}. \tag{12}$$

Then (P) is equivalent to the abstract program

$$\min f(x,t,Y) \ \text{ subject to } \ \mathcal{G}(x,t,Y) \in \mathbb{S}^{3m}_{-}.$$

Assume that Robinson's constraint qualification [9] is satisfied for this program. Then if (x^*, t^*, Y^*) is a local minimum, the tangent program

$$\begin{array}{ll} \text{minimize} & f'(x^*,t^*,Y^*)^{\mathrm{T}}(\delta x, \delta t, \delta Y) \\ \text{subject to} & \mathcal{G}'(x^*,t^*,Y^*)(\delta x, \delta t, \delta Y) \in T(\mathbb{S}^{3m}_{-}, \mathcal{G}(x^*,t^*,Y^*)) \end{array} \tag{13}$$

has the unique solution $(\delta x, \delta t, \delta Y) = (0, 0, 0)$. Here $T(\mathbb{S}^{3m}_{-}, G)$ is the Clarke tangent cone, which according to [9] is $T(\mathbb{S}^{3m}_{-}, G) = \{Z \in \mathbb{S}^{3m} : Q^{\mathrm{T}} Z Q \preceq 0\}$ if $\lambda_1(G) = 0$, where the columns of the matrix Q are an orthonormal basis of the eigenspace of G associated with the maximum eigenvalue $\lambda_1(G) = 0$, while $T(\mathbb{S}^{3m}_{-}, G) = \mathbb{S}^{3m}$ if $\lambda_1(G) < 0$, $T(\mathbb{S}^{3m}_{-}, G) = \emptyset$ if $\lambda_1(G) > 0$.

It turns out that optimality of $(0, 0, 0)$ in (13) is a condition which may be checked by solving an SDP. Indeed, observe that

$$f'(x^*,t^*,Y^*)^{\mathrm{T}}(\delta x, \delta t, \delta Y) = \delta t$$

and

$$\mathcal{G}'(x^*,t^*,Y^*)(\delta x, \delta t, \delta Y) = \begin{bmatrix} \delta Y & 0 & 0 \\ 0 & -\delta Y & 0 \\ 0 & 0 & \delta Z \end{bmatrix},$$

where as before $\mathcal{G}'$ denotes the differential of the operator $\mathcal{G}$, and where we use the shorthand notation

$$\begin{aligned} Z^* &:= \mathcal{F}(x^*)^{\mathrm{T}} Y^* + Y^* \mathcal{F}(x^*) - 2t^* Y^*, \\ \delta Z &:= [\mathcal{F}'(x^*)\delta x]^{\mathrm{T}} Y^* + \mathcal{F}(x^*)^{\mathrm{T}} \delta Y + Y^* [\mathcal{F}'(x^*)\delta x] + \delta Y \mathcal{F}(x^*) - 2t^* \delta Y - 2\delta t Y^*. \end{aligned}$$

Clearly, the tangent cone in question is

$$T(\mathbb{S}^{3m}_{-}, \mathcal{G}(x^*,t^*,Y^*)) = T(\mathbb{S}^{m}_{-}, Y^* - I) \times T(\mathbb{S}^{m}_{-}, \theta I - Y^*) \times T(\mathbb{S}^{m}_{-}, Z^*),$$

so we have to compute these three tangent cones.

Let Q_1 be an orthonormal basis of the eigenspace of $Y^* - I$ associated with the eigenvalue 0, and let Q_θ be a basis of the eigenspace of $\theta I - Y^*$ associated with the eigenvalue 0. Finally, let P be a basis of the eigenspace of Z^* associated with the eigenvalue 0. Then the tangent program becomes

$$\begin{array}{ll} \text{minimize} & \delta t \\ \text{subject to} & Q_1^{\mathrm{T}} \delta Y Q_1 \preceq 0, \\ & Q_\theta^{\mathrm{T}} \delta Y Q_\theta \succeq 0, \\ & P^{\mathrm{T}} \delta Z P \preceq 0, \\ & \|\delta x\| \leq 1,\ |\delta t| \leq 1,\ \|\delta Y\| \leq 1. \end{array} \tag{14}$$

This is an SDP in the unknown variable $(\delta x, \delta t, \delta Y)$. The decision is now as follows. If our tangent program reveals (x^*, t^*, Y^*) as a critical point, we stop and thereby accept the solution proposed by MDS. Otherwise, δx will show us the way to escape from the current point x^*. In terms of the MDS algorithm, when $x^* = v_0$, the nonsmooth descent step will be $w = x^* + \tau \delta x$ for some $\tau > 0$ found by a line search.

6.3. Stopping test for the H_∞-norm. For constrained programs like those in section 4, the situation is principally the same as in the unconstrained case. When we get stalled at some iterate K^*, we would like to know whether we have a local minimum (a KKT point), or whether we could keep making progress by avoiding a dead point.

In this section, we consider a stopping test for the nonsmooth program (9), which is based on the frequency domain representation of the H_∞-norm.

Suppose we have reached an iterate K^* such that $\|T_{w\to z}(K^*)\|_\infty = \gamma^*$ and $\|(sI - \mathcal{A}(K^*))^{-1}\|_\infty = \varepsilon^{-1}$. We want to decide whether K^* is a critical point of the program

$$\min\{\|T_{w\to z}(K)\|_\infty : \|(sI - \mathcal{A}(K))^{-1}\|_\infty \leq \varepsilon^{-1}\}.$$

This may be based on a nonsmooth stationarity test, which checks whether or not $0 \in \partial n(K^*) + \mathbb{R}_+ \partial m(K^*)$, where $n(K) = \|T_{w\to z}(K)\|_\infty$, $m(K) = \max(0, \| (sI - \mathcal{A}(K))^{-1} \|_\infty - \varepsilon^{-1})$ (see [22, Thm. 6.1.1, Prop. 3.3.1]). We therefore need to compute the subdifferentials $\partial n(K^*)$ and $\partial m(K^*)$.

Let us start with $\partial n(K^*)$, which is more general. The subdifferential $\partial m(K^*)$ will then follow as a special case. Recall that $T_{w\to z}(K, s)$ is of the form

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

where $\mathcal{A}(K)$, $\mathcal{B}(K)$, $\mathcal{C}(K)$, and $\mathcal{D}(K)$ are given in (8). Defining $\mathcal{F}(K, s) = (sI - \mathcal{A}(K))^{-1}$, we obtain the derivative $T'_{w\to z}$ of $T_{w\to z}$ at K^* as

$$\begin{aligned} T'_{w\to z}(K^*)\, \delta K(s) = {} & D_{12}\, \delta K\, C_2 \mathcal{F}(K^*, s)\mathcal{B}(K^*) \\ & + \mathcal{C}(K^*)\mathcal{F}(K^*, s) B_2\, \delta K\, C_2 \mathcal{F}(K^*, s)\mathcal{B}(K^*) \\ & + \mathcal{C}(K^*)\mathcal{F}(K^*, s) B_2\, \delta K\, D_{21} + D_{12}\, \delta K\, D_{21}. \end{aligned} \tag{15}$$

Now let $\phi = \phi_Y$ be a subgradient of $\|\cdot\|_\infty$ at $T_{w\to z}(K^*)$ of the form (3), specified by $Y \succeq 0$, $\mathrm{Tr}(Y) = 1$ and with $\|T_{w\to z}(K^*)\|_\infty$ attained at frequency ω. We wish to compute $\Phi_Y := T'_{w\to z}(K^*)^\star \phi_Y \in \mathbb{M}_{m_2,p_2}$. The adjoint $T'_{w\to z}(K^*)^\star$ acts on ϕ_Y through

$$
\begin{aligned}
&\langle T'_{w\to z}(K^*)^\star \phi_Y, \delta K\rangle \\
&\quad = \langle T'_{w\to z}(K^*)\delta K, \phi_Y\rangle \\
&\quad = \|T_{w\to z}(K^*)\|_\infty^{-1} \mathrm{Re}\,\mathrm{Tr}\,\left(T_{w\to z}(K^*, j\omega)^{\mathrm{H}} Q Y Q^{\mathrm{H}} T'_{w\to z}(K^*)\delta K\,(j\omega)\right) \\
&\quad = \|T_{w\to z}(K^*)\|_\infty^{-1} \mathrm{Re}\,\mathrm{Tr}\,\left[C_2\mathcal{F}(K^*, j\omega)\mathcal{B}(K^*)T_{w\to z}(K^*, j\omega)^{\mathrm{H}} Q Y Q^{\mathrm{H}} D_{12}\right. \\
&\qquad + C_2\mathcal{F}(K^*, j\omega)\mathcal{B}(K^*)T_{w\to z}(K^*, j\omega)^{\mathrm{H}} Q Y Q^{\mathrm{H}} \mathcal{C}(K^*)\mathcal{F}(K^*, j\omega)B_2 \\
&\qquad + D_{21}T_{w\to z}(K^*, j\omega)^{\mathrm{H}} Q Y Q^{\mathrm{H}} \mathcal{C}(K^*)\mathcal{F}(K^*, j\omega)B_2 \\
&\qquad + \left. D_{21}T_{w\to z}(K^*, j\omega)^{\mathrm{H}} Q Y Q^{\mathrm{H}} D_{12}\,\right]\delta K.
\end{aligned}
$$

In consequence, the Clarke subgradients of $n = \|\cdot\|_\infty \circ T_{w\to z}$ at K^* are of the form

$$
\begin{aligned}
\Phi_Y = \|T_{w\to z}(K^*)\|_\infty^{-1} \mathrm{Re}\,&\left[C_2\mathcal{F}(K^*, j\omega)\mathcal{B}(K^*)T_{w\to z}(K^*, j\omega)^{\mathrm{H}} Q Y Q^{\mathrm{H}} D_{12}\right. \\
&+ C_2\mathcal{F}(K^*, j\omega)\mathcal{B}(K^*)T_{w\to z}(K^*, j\omega)^{\mathrm{H}} Q Y Q^{\mathrm{H}} \mathcal{C}(K^*)\mathcal{F}(K^*, j\omega)B_2 \\
&+ D_{21}T_{w\to z}(K^*, j\omega)^{\mathrm{H}} Q Y Q^{\mathrm{H}} \mathcal{C}(K^*)\mathcal{F}(K^*, j\omega)B_2 \\
&+ \left. D_{21}T_{w\to z}(K^*, j\omega)^{\mathrm{H}} Q Y Q^{\mathrm{H}} D_{12}\right]^{\mathrm{T}},
\end{aligned}
$$

or more simply,

$$
\Phi_Y = \|T_{w\to z}(K^*)\|_\infty^{-1} \mathrm{Re}\,\left\{G_{21}(K^*, j\omega)\,T_{w\to z}(K^*, j\omega)^{\mathrm{H}} Q Y Q^{\mathrm{H}} G_{12}(K^*, j\omega)\right\}^{\mathrm{T}},
$$

where

$$
\begin{aligned}
G_{21}(K^*, j\omega) &:= C_2\mathcal{F}(K^*, j\omega)\mathcal{B}(K^*) + D_{21}, \\
G_{12}(K^*, j\omega) &:= \mathcal{C}(K^*)\mathcal{F}(K^*, j\omega)B_2 + D_{12}.
\end{aligned}
$$

The subdifferential of the function $m(.)$ is obtained through similar calculations. We first note that up to a constant term, the second component of $m(.)$ is $\|\mathcal{F}(K)\|_\infty$, a simplification of $T_{w\to z}(K)$ with $\mathcal{C}(K) = I$, $\mathcal{B}(K) = I$, and $\mathcal{D}(K) = 0$. Assuming this time that the supremum is attained at frequency ω', the Clarke subgradients of $\|\mathcal{F}(K)\|_\infty$ at K^* are of the form

$$
\Psi_{\widehat{Y}} := \|\mathcal{F}(K^*)\|_\infty^{-1} \mathrm{Re}\,\left\{C_2\mathcal{F}(K^*, j\omega')\mathcal{F}(K^*, j\omega')^{\mathrm{H}} \widehat{Q}\widehat{Y}\widehat{Q}^{\mathrm{H}} \mathcal{F}(K^*, j\omega')B_2\right\}^{\mathrm{T}},
$$

with $\widehat{Y} \succeq 0$, $\mathrm{Tr}\,(\widehat{Y}) = 1$. Since both components of the max function $m(\cdot)$ are active at K^*, the subdifferential of m at K^* is the convex hull of the origin with the subdifferential of $\|\mathcal{F}(K)\|_\infty$ at K^* [22]. Those subgradients are therefore of the form $\Psi_{\widehat{Y}}$, $\widehat{Y} \succeq 0$, and $\mathrm{Tr}\,(\widehat{Y}) \le 1$. These formulae are easily adapted if the first H_∞-norm is attained at frequencies $\omega_1, \dots, \omega_p$, and the second at $\omega'_1, \dots, \omega'_q$.

Suppose $\|T_{w\to z}(K^*)\|_\infty$ is attained at a single ω, and $\|\mathcal{F}(K^*)\|_\infty$ at a single ω'. Then the optimality test leads to solving the optimization program

$$
\min\{\|\Phi_Y + \Psi_{\widehat{Y}}\|_2 : Y \succeq 0, \mathrm{Tr}\,(Y) = 1, \widehat{Y} \succeq 0\}
$$

which is a low-dimensional SDP. If the value of this program is 0, then K^* is a critical point.

7. Crisis prevention. The nonsmooth stopping tests developed in the previous section could be adapted to many other programs. We should be aware, however, that the steps $v_0 \to w$ they generate are steepest descent steps, which cannot guarantee convergence under nonsmoothness (see [50] for a discussion). Put differently, even though the stopping test may allow us to move on, we have no guarantee that an accumulation point of the sequence so generated would not be another dead point. In order to exclude this categorically, a more sophisticated strategy, crisis prevention, is required. Here we get a convergence certificate, which is built on the possibility to *quantify* descent.

A well-known tool of convex nonsmooth analysis which allows us to quantify descent is ε-subgradients (see [35, Thm. 1.1.5]). Since our present criteria are nonconvex, those may not be used directly and some modifications are required (see [57, 58]). But the idea is essentially the same.

7.1. Quantitative descent for $\boldsymbol{f = \lambda_1 \circ \mathcal{B}}$. To begin with, let us examine a strategy suited for eigenvalue optimization, used in the simultaneous stabilization problem section 8.3. We consider a nonconvex maximum eigenvalue function of the form

$$f(x) = \lambda_1\left(\mathcal{B}(x)\right) \tag{16}$$

with a bilinear (or more generally $\mathcal{C}^2$) operator $\mathcal{B}$. We solve the unconstrained optimization problem:

$$\text{minimize} \quad f(x) = \lambda_1\left(\mathcal{B}(x)\right),\ x \in \mathbb{R}^n.$$

We follow [57, 58], which extends previous work by Cullum, Donath, and Wolfe [23] and Oustry [60], where affine operators were used, to more general functions $f = \lambda_1 \circ \mathcal{B}$. We use an approximation $\delta_\varepsilon f(x)$ of the ε-subdifferential $\partial_\varepsilon f(x)$ of f at the current x, called the *ε-enlarged subdifferential.* We compute the approximate subgradient $g \in \delta_\varepsilon f(x)$, which gives rise to the so-called *steepest ε-enlarged descent direction.* Let us define

$$\delta_\varepsilon f(x) \;=\; \left\{\mathcal{B}'(x)^\star Z : Z = Q_\varepsilon Y Q_\varepsilon^{\mathrm{T}},\ Y \succeq 0,\, \mathrm{tr}(Y) = 1,\, Y \in \mathbb{S}^{r(\varepsilon)}\right\},$$

where the first $r(\varepsilon)$ eigenvalues of $\mathcal{B}(x) \in \mathbb{S}^m$ are those which satisfy $\lambda_i > \lambda_1 - \varepsilon$, and where the columns of the $r(\varepsilon) \times m$-matrix Q_ε form an orthonormal basis of the invariant subspace associated with these eigenvalues. Then

$$\partial f(x) \subset \delta_\varepsilon f(x) \subset \partial_\varepsilon f(x),$$

and $\delta_\varepsilon f(x)$ is an inner approximation of $\partial_\varepsilon f(x)$, which has the advantage of being computable. Namely, the direction of steepest ε-enlarged descent d is obtained as

$$d = -\frac{g}{\|g\|}, \quad g = \operatorname{argmin}\left\{\|g\| : g \in \delta_\varepsilon f(x)\right\}. \tag{17}$$

The solution g of (17) is the projection of the origin onto the compact convex set $\delta_\varepsilon f(x)$. This is in complete analogy with the direction of steepest descent, which is obtained by projecting the origin onto the subdifferential $\partial f(x) = \delta_0 f(x)$. What would be the most useful is the direction of steepest ε-descent, obtained by projecting 0 onto $\partial_\varepsilon f(x)$, but this quantity is difficult to compute (see, however, [35] for some ideas how this may be tried).

1. Given iterate x, stop if $0 \in \partial f(x) = \delta_0 f(x)$, because x is a critical point. Otherwise choose $\varepsilon > 0$.
2. Given $\varepsilon > 0$, compute the direction d of steepest ε-enlarged descent by solving (19). Let (t, d) be the solution.
3. If $d = 0$ (and hence $t = 0$), then $0 \in \delta_\varepsilon f(x)$. Decrease ε and go back to step 2.
4. If $d \neq 0$, then $0 \notin \delta_\varepsilon f(x)$ and we obtain $x^+ = x + \tau d$ with $f(x^+) < f(x)$ using a line search like in [57]. Let $w = x^+$ be the intervention step for MDS and quit.

FIG. 4. *Quantified descent $v_0 \to w$ for $f = \lambda_1 \circ \mathcal{B}$.*

Contrary to $\partial_\varepsilon f(x)$, the support function of the compact convex set $\delta_\varepsilon f(x)$ is known explicitly. We have (cf. [23, 60, 57])

$$\tilde{f}'_\varepsilon(x; d) := \max\{g^{\mathrm{T}} d : g \in \delta_\varepsilon f(x)\} = \lambda_1\left(Q_\varepsilon^{\mathrm{T}}\,[\mathcal{B}'(x)d]\,Q_\varepsilon\right),$$

where $\tilde{f}'_\varepsilon(x; d)$ is the directional derivative considered in [23, 60]. Therefore, the direction of steepest ε-enlarged descent is found by solving the program

$$\min_{\|d\| \le 1} \lambda_1\left(Q_\varepsilon^{\mathrm{T}}\,[\mathcal{B}'(x)d]\,Q_\varepsilon\right), \tag{18}$$

and the solution $\mathbf{d} = -g/\|g\|$ satisfies

$$-\|g\| = -\mathrm{dist}\,(0, \delta_\varepsilon f(x)) = \tilde{f}'_\varepsilon(x; d) < 0.$$

Notice that (18) is equivalent to the SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & Q_\varepsilon^{\mathrm{T}}\,[\mathcal{B}'(x)d]\,Q_\varepsilon \preceq tI, \\ & \|d\| \le 1. \end{array} \tag{19}$$

A descent direction d for $f = \lambda_1 \circ \mathcal{B}$ at x is therefore found as soon as the value of (19) is negative, and the corresponding d gives even a quantifiable descent in the sense of Theorem 1 below. The appealing feature of this method is that the size of the LMI in (18) and (19) is $r(\varepsilon)$, which is usually small. An important consequence is that it can be solved very cheaply if a dual SDP formulation is used. Altogether we have the crisis prevention method shown in Figure 4.

The possible decrease $f(x^+) < f(x)$ is quantified by the following result, whose proof is given in [57] for a spectral bundle algorithm which generates descent steps as above. Since the convergence properties of the nonsmooth MDS method hinge on the properties of the sequence of Cauchy points w, the result carries over to our present situation.

THEOREM 1. *Consider the minimization of $f = \lambda_1 \circ \mathcal{B}$. Suppose x^0 is such that $\{x \in \mathbb{R}^n : f(x) \le f(x^0)\}$ is compact. Let the sequence x^k with starting point x^0 be generated by the MDS method with nonsmooth descent step. Suppose at stage k the parameter ε_k is chosen according to the ε-management of* [57, 58]. *Then there exists a constant $C > 0$ such that the nonsmooth MDS method achieves a decrease of at least $f(x^{k+1}) - f(x^k) \le -C\,\Delta_{\varepsilon_k}\,|\tilde{f}'_{\varepsilon_k}(x^k; d^k)|^2$, where d^k is the direction of steepest ε_k-enlarged descent at x^k and $\Delta_{\varepsilon_k} = \lambda_{r(\varepsilon_k)} - \lambda_{r(\varepsilon_k)+1}$. Moreover, some subsequence of x^k converges to a critical point of f.* ∎

7.2. Quantifiable descent for $g = \alpha \circ \mathcal{F}$. In this section we discuss the difficult case of the spectral abscissa. Due to its highly nonsmooth character, quantified decrease for $g = \alpha \circ \mathcal{F}$ is more difficult to guarantee than for $f = \lambda_1 \circ \mathcal{B}$.

Let us again take recourse to the SDP formulation of α. Suppose $g(x^*) = \alpha(\mathcal{F}(x^*)) = t^*$. We wish to decrease the value of g in a neighborhood U of x^*. Following Lemma 1, for fixed $0 < \theta \ll 1$, there exists $Y^* \in [\theta I, I]$ such that $\lambda_1(\mathcal{B}(x^*, Y^*, t^*)) = 0$, where we define $\mathcal{B}(x, Y, t) := \mathcal{F}(x)^{\mathrm{T}} Y + Y\mathcal{F}(x) - 2tY \preceq 0$. Finding Y^* amounts to solving an SDP. Now let us introduce

$$\widetilde{\mathcal{B}}(x, Y, t) = \begin{bmatrix} Y - I & 0 & 0 \\ 0 & \theta I - Y & 0 \\ 0 & 0 & \mathcal{B}(x, Y, t) \end{bmatrix}.$$

Then decreasing the value $g(x) = t$ below t^* is equivalent to decreasing the value t of the program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \widetilde{\mathcal{B}}(x, Y, t) \preceq 0 \end{array}$$

below t^*. We obtain such a decrease $t < t^*$ using Kiwiel's progress function [43], which in our situation may be written as

$$\kappa(x, Y, t; t^*) = \lambda_1 \begin{bmatrix} t - t^* & 0 \\ 0 & \widetilde{\mathcal{B}}(x, Y, t) \end{bmatrix} =: \lambda_1\left(\widehat{\mathcal{B}}(x, Y, t; t^*)\right).$$

We have the following.

LEMMA 2. *Suppose $g(x^*) = \alpha(\mathcal{F}(x^*)) = t^*$. Then decrease $t = g(x) < g(x^*) = t^*$ is achieved for some x in a neighborhood U of x^* if and only if $\kappa(x, Y, t; t^*) < 0$ for suitable Y.* □

What we are interested in is quantified decrease in the same sense as used before, so we use the ε-enlarged subdifferential $\delta_\varepsilon \kappa$ of the maximum eigenvalue function $\kappa = \lambda_1 \circ \widehat{\mathcal{B}}$. The procedure, whose convergence theory is covered by [57], is shown in Figure 5.

Notice that the costly part here is computing Y^*. The second SDP in step 3 is of small size, since the corresponding LMI is in the space of $r(\varepsilon) \times r(\varepsilon)$ matrices. Repeating this step to identify a suitable ε is therefore not expensive. This has the interesting feature that as long as ε-steepest descent steps are taken, the large SDP need not be solved at all. This makes a pure nonsmooth descent method seem attractive. Such an approach is developed in [32] for large SDPs arising as relaxations of integer programs. Similar to that reference, solving the SDP dual of (7.2) is more efficient. Finally, we stress that extending the quantified descent step for the spectral abscissa to a broader class of problems like those in (4) is straightforward and left to the reader.

Remark. As soon as search directions based on the ε-enlarged subdifferential are used, a good choice of ε is required. Based on extensive numerical testing, we have used a very small $\varepsilon = 1e{-}9$ for stopping tests, while good progress in a descent step seems to ask for moderate values $\varepsilon \in [0.01\,; 0.1]$. This is what has been used in section 8.

8. Numerical experiments. In this section, we test the MDS algorithm with nonsmooth descent steps on a wide range of synthesis problems from the literature.

1. Given $g(x^*) = \alpha(\mathcal{F}(x^*)) = t^*$, quit if the stopping test (14) indicates a critical point. Otherwise:
2. Solve an SDP to compute Y^* such that $\lambda_1(\mathcal{B}(x^*, Y^*, t^*)) = 0$. Choose $\varepsilon > 0$.
3. Given $\varepsilon > 0$, compute $d = (\delta x, \delta Y, \delta t)$, the direction of steepest ε-enlarged descent of $\kappa(\cdot, \cdot, \cdot; t^*)$ at the point (x^*, Y^*, t^*) by solving the SDP:
$$\begin{array}{ll} \text{minimize} & \rho \\ \text{subject to} & \widehat{Q}_\varepsilon^{\mathrm{T}} \left[\widehat{\mathcal{B}}'(x^*, Y^*, t^*; t^*)\, d\right] \widehat{Q}_\varepsilon \preceq \rho I, \\ & \|\delta x\| \le 1,\ \|\delta Y\| \le 1,\ |\delta t| \le 1. \end{array}$$
Here the $r(\varepsilon)$ columns of $\widehat{Q}_\varepsilon$ are an orthonormal basis of the invariant subspace of $\widehat{\mathcal{B}}(x^*, Y^*, t^*; t^*)$ associated with its ε-largest eigenvalues. Let $d = (\delta x, \delta Y, \delta t)$ be the solution.
4. If $d = 0$, then $0 \in \delta_\varepsilon \kappa(x^*, Y^*, t^*; t^*)$. Decrease ε and go back to step 3.
6. Having found $d \neq 0$, decrease the value of κ using a line search as in [57]. The corresponding step $x^+ = x^* + \tau \delta x$ decreases g accordingly. Let $w = x^+$ be the intervention step for MDS, and quit.

FIG. 5. *Quantified descent step $v_0 \to w$ for $g = \alpha \circ \mathcal{F}$.*

Computations were performed on a (low-level) SUN-Blade Sparc with 256 RAM and a 650 MHz sparcv9 processor. LMI-related computations needed for nonsmooth descent steps were performed using either the LMI Control Toolbox [28] or our homemade SDP code [5]. The contraction and expansion parameters were set to $\theta = 0.5$ and $\mu = 2.0$ throughout.

8.1. Static output-feedback stabilization. We start with static output-feedback stabilization without any performance specification. Solving (10) is a pure feasibility problem and somewhat simpler than the problems examined in what follows. It is used to initialize the constrained problem (9).

In our implementation, the MDS code was stopped as soon as a strictly negative spectral abscissa was obtained. Restarts were used as soon as the nonsmooth optimality test indicated a local minimum $\bar{x}$ of $g = \alpha \circ \mathcal{F}$ with positive value $g(\bar{x}) > 0$. We also encountered dead points, where the nonsmooth stopping test indicated a way to move on. What helps in this case is to restart MDS with the new seed proposed by the spectral bundling step, and change the geometry of the simplex. In all tests, the initial seed point was chosen to be the origin of the variable space. The vertices of the initial S are then relabeled so that v_0 is the best vertex. Contrary to what might seem plausible, MDS frequently encounters dead points and fails when run in default mode without nonsmooth steps. We discuss some of these at the end of this section.

As emphasized in the introductory section, the nonsmooth MDS is fairly insensitive to the number of states, since Lyapunov variables are not involved. A striking example is the Boeing 767 flutter problem (AC10), which our algorithm solved in 0.41-s cpu, starting from the initial point $K = 0$. This indicates that this problem is not as difficult as the size would suggest. (In fact, some of our smaller problems turned out more difficult.) The nonsmooth MDS technique appears surprisingly efficient compared to the gradient sampling algorithm proposed in [18], which for this problem required hours of cpu time and several hundreds of restarts. This example is

Table 1
Static output-feedback stabilization right-angled simplex.

Problem	(n, m, p)	Iteration	cpu (s)	Reference
Transport airplane	$(9, 1, 5)$	3	0.05	[29]
Horisberger's example	$(9, 1, 4)$	13	0.12	[38]
VTOL helicopter	$(4, 2, 1)$	1	0.01	[42]
Chemical reactor	$(4, 2, 2)$	2	0.02	[39]
Piezoelectric actuator	$(5, 1, 3)$	2	0.17	[21]
AC10	$(55, 2, 2)$	3	0.41	[47]
HF1	$(130, 1, 2)$	Stable	–	[47]

also included in the library [47] and has been solved by the technique of [49, 48].

Example. Let us illustrate a typical difficulty related to nonsmoothness of the spectral abscissa, when MDS stops at an iterate K^* where several eigenvalues of the closed-loop spectrum are active. This happens, e.g., in Horisberger's example with seed point at the origin and with the regular simplex geometry. When nonsmooth descent is switched off, MDS eventually hits such a nonsmooth iterate and starts contracting the simplex. This yields the static (nonstabilizing) gain

$$K = [\,-5.9176e{-}01 \quad 7.1864e{+}00 \quad -3.1396e{+}01 \quad 3.5870e{+}01\,],$$

with closed-loop spectrum

$$\Lambda(A + B_2KC_2) = \begin{cases} -6.6646e{-}01 \pm 6.2303e{+}01j \\ -3.9851e{+}00 \pm 1.8336e{+}01j \\ -7.8086e{+}00 \pm 4.0906e{+}00j \\ \;\;5.4005e{-}01 \pm 8.3040e{-}01j \\ \qquad\quad 5.4005e{-}01 \end{cases}.$$

The question is now whether we are at a dead point or at a local minimum. If the technique discussed in section 6 is switched on, a nonsmooth descent step $v_0 \to w$ is performed, which reduces the spectral abscissa from $5.4005e{-}01$ to $5.183e{-}01$. This is followed by a number of reflection/expansion/contraction steps of MDS, yielding the iterate

$$K = [\,-2.0595e - 01 \quad 6.4949e + 00 \quad -3.1503e + 01 \quad 3.6173e{+}01\,]$$

with closed-loop spectrum

$$\Lambda(A + B_2KC_2) = \begin{cases} -6.7523e{-}01 \pm 6.2320e{+}01j \\ -4.1595e{+}00 \pm 1.8393e{+}01j \\ -7.5240e{+}00 \pm 4.9624e{+}00j \\ \;\;4.7250e{-}01 \pm 3.8239e{-}01j \\ \qquad\quad 4.7250e{-}01 \end{cases}.$$

The nonsmooth stopping test now clearly identifies this as a local minimum (a critical point), since no descent direction exists. At this stage a restart of MDS is inevitable, because $\alpha(A + B_2KC_2) = 4.7250e{-}01 > 0$.

Our testing has shown that the following simple trick is successful when a restart is due. We keep the current best point but switch geometries, for instance from regular to right-angled or vice versa. In the example, we switched from regular to right-angled simplices, which generated different search directions. MDS was now successful and reached a stabilizing gain:

$$K = [\,3.1794e{+}01 \quad 6.4949e{+}00 \quad 4.3250e{+}02 \quad 5.1173e{+}01\,],$$

with corresponding spectral abscissa $\alpha(A + B_2KC_2) = -4.1442e{-}01 < 0$. □

8.2. Static and fixed-order output-feedback H_∞-synthesis. This section reports experiments with static and fixed-order output-feedback H_∞-synthesis. The out-set is from section 3, the extension to fixed-order problems is standard [6]. We solve program (9), using the corresponding controllers K^0 computed via (10) as initial value.

Results achieved with nonsmooth MDS are based on the infinite barrier

$$B(K) = \begin{cases} \|T_{w\to z}(s,K)\|_\infty & \text{if } \alpha(A + B_2KC_2) \le -\tau, \\ +\infty & \text{otherwise,} \end{cases} \tag{20}$$

where $\tau > 0$ is some small fixed threshold. The infinite barrier function works surprisingly well with the MDS technique, as also witnessed by [10] in different contexts. Function evaluation for the H_∞-norm is based on the efficient bisection algorithm in [12]. See also the MATLAB implementation described in [26]. A catalog of results is displayed in Table 2. The H_∞ performance achieved with the MDS method "H_∞ MDS" as well as with the spectral quadratic SDP method "H_∞ AL" in [5] are described. For completeness, in column "H_∞ full" the performance of the full-order H_∞ controller (computed by LMIs or algebraic Riccati equations) is shown and gives a lower bound for the H_∞-gain.

Table 2

Static and fixed-order H_∞-synthesis with MDS algorithm best results with right-angled and regular simplices stopping tolerance $\varepsilon = 1e{-}9$.

Problem	Order	Iteration	cpu (s)	H_∞ MDS	H_∞ AL	H_∞ full
Transport airplane	Static	37	20	2.34	2.22	1.60
VTOL helicopter	Static	10	2.69	0.190	0.157	0.096
Chemical reactor	Static	38	16.96	1.183	1.202	1.141
Piezoelectric actuator	Static	112	8.62	$1.76e{-}4$	$3.05e{-}3$	$9.63e{-}5$
AC10	Static	72	612	14.22	Intractable	0.052
HF1	Static	11	1100	0.447	Intractable	0.449

The choice of the simplex geometry, right-angled or regular, may influence the computed solution. Contrary to what might be guessed, the regular geometry is not always better than the right-angled geometry. We have therefore decided to test both and report the best result. This is reasonably affordable with regard to cpu time, as seen in Table 2 even for high-order systems. The initial seed point was the origin in all examples. As already discussed in [5], the augmented Lagrangian (AL) technique is no longer operational for systems with roughly more than 40 states. Again, we would like to stress the good results obtained with the MDS method for the Boeing 767 problem (AC10). Actually, the projective SDP code of MATLAB ran into difficulties to solve the LMI problem corresponding to the (convex) full-order problem and diagnosed the problem as infeasible after more than 4 hours of execution time in default mode.

The computed static controller obtained by the MDS method for the Boeing 767 flutter problems (AC10) is

$$K_{\text{static}} = \begin{bmatrix} -0.0966 & 0.0000 \\ 3.1681 & 0.0000 \end{bmatrix}.$$

The large-size HF1 problem is taken from the library [47]. It does not require prior stabilization as the plant is open-loop stable. Hence, $K = 0$ may serve as a starting point for the H_∞-optimization in Table 2. Here the static gain $K = [\,1.9943 \quad -3.4943\,]$ is found.

8.3. Simultaneous stabilization problems. Simultaneous stabilization is a longstanding problem in the automatic control literature. It consists in the search of a single controller which stabilizes a finite set of plants. This is of great practical interest in different situations. A system may have several modes of operation, but the controller is required to stabilize all modes. A more challenging situation is when the system may be subject to different failures such as actuator/detector breakdown, which often result in drastic deviations of the plant from its nominal description. The controller is then required to stabilize normal and abnormal operating modes. Unfortunately, the simultaneous stabilization problem has no analytical solution for more than two plants and is classified as NP-hard [8]. Existing techniques usually try to verify sufficient conditions. If successful, this leads to high-order controllers. Our experiments indicate that local optimization techniques and in particular DS methods may be of interest for designing simpler controllers, which is crucial for applications.

For single-input single-output systems $\{G_i(s),\ i = 1, \ldots, q\}$, the simultaneous stabilization problem can be formulated as follows:

- find a controller with transfer function

$$K(s,x) = \frac{N_K(s,x)}{D_K(s,x)} = \frac{x_1 s^m + \cdots + x_m s + x_{m+1}}{s^n + x_{m+2}s^{n-1} + \cdots + x_{m+n}s + x_{m+n+1}}, \tag{21}$$

  where as before $x := [x_1 \cdots x_{m+n+1}]^{\mathrm{T}}$ gathers the decision variables,
- such that the closed-loop characteristic polynomials

$$p_i(s,x) := N_{G_i}(s)N_K(s,x) + D_{G_i}(s)D_K(s,x)$$

  have only stable roots for $i = 1, \ldots, q$.

This may be addressed by the optimization program

$$\underset{x\in\mathbb{R}^{m+n+1}}{\text{minimize}} \ \max_{i=1,\ldots,q} \ \mathrm{Re}\,(\text{roots of } p_i(s,x)) \tag{22}$$

and a simultaneous stabilizing controller is found as soon as the value of this program is < 0. Program (22) resembles the static stabilization formulation discussed in section 8.1 and we follow a similar line of attack.

A challenging variant of this problem is the strong stabilization problem, where the controller itself is required to be stable. This is incorporated into the cast (22) by just adding $D_K(s,x)$ to the family of plant polynomials.

TABLE 3
*Simultaneous stabilization with MDS right-angled simplex * strong stabilization problem.*

Problem	Order	Iteration	cpu (s)	Restart	Reference
F4e aircraft	Static	2	0.71	None	[2]
cao	Static	13	0.28	None	[20]
cao	1	1	0.25	3	[20]
henrion	1	2	0.51	None	[34]
bredemann1*	1	6	2.05	3	[16, p. 68]
bredemann2*	1	3	0.82	3	[15]

In this testing, the nonsmooth MDS was again successful on a list of applications from the literature. Restarts have been used with a different initial seed point when an unsatisfactory local minimum was encountered. Often we obtained simpler controllers than those previously published and derived from constructive sufficient conditions. For example, the method in [16] yields a fifth-order controller for example bredemann1, whereas the MDS technique was able to show that first-order strong simultaneous stabilization is possible. A similar comment applies to example bredemann2.

An alternative cast for simultaneous stabilization is via Hermite–Fujiwara matrices. In this setting, the nonsmooth program (22) reduces to a finite set of quadratic matrix inequality constraints [33]:

$$\mathcal{H}(x) := \sum_{i=1}^{m+n+1} \sum_{j=i}^{m+n+1} x_i x_j H_{ij} \prec 0,$$

where the decision vector x comprises controller parameters in (21). Here MDS is applied to the eigenvalue optimization program

$$\min_x \lambda_1 \left(\mathcal{H}(x)\right). \tag{23}$$

We apply MDS to a problem from [33], which consists in the simultaneous stabilization of four plants. Hence, x is required to be strictly feasible for four quadratic matrix inequalities of the form (23). This problem is of special interest because numerous dead points and unsatisfactory local minima were found if different seed points were used.

TABLE 4

Simultaneous stabilization using Hermite–Fujiwara BMI characterization final spectrum of quadratic SDP results with two starting points and regular simplices.

Seed	$1, 1, -1, -1$	$-1, -1, 1, 1$
Final iterate	$3.5068, 4.2139, 0.0925, 0.0925$	$-4.4420, 0.4275, 0.5059, 0.1618$
Final spectrum of quadratic SDP (23)	$-1.3846e{+}03$	$-2.2926e{+}03$
	$-1.0473e{+}03$	$-3.7468e{+}02$
	$-8.0116e{+}02$	$-3.1919e{+}02$
	$-3.9982e{+}02$	$-2.1174e{+}01$
	$-3.8359e{+}02$	$-6.7302e{+}00$
	$-2.0603e{+}02$	$-6.1145e{+}00$
	$-1.3928e{+}02$	$-2.3900e{+}00$
	$-8.4586e{+}01$	$-2.3453e{+}00$
	$-2.6890e{+}00$	$-5.0609e{-}02$
	$-1.4544e{+}00$	$1.8472e{-}01$
	$-7.6821e{-}01$	$1.8528e{-}01$
	$-6.3137e{-}01$	$1.8528e{-}01$
Controller	$\dfrac{3.5068s + 4.2139}{9.2540e{-}02s + 9.2540e{-}02}$	none

Example. For the purpose of testing, MDS was first run without nonsmooth steps. Table 4 shows two scenarios with default MDS. In column 2 the nonsmooth stopping test from section 7.1 was switched on as soon as MDS got stalled. It reveals that we are at a dead point and not at a local minimum. While nonsmooth steps $v_0 \to w$ allowed MDS to move on, crisis intervention ultimately did not lead to a stabilizing controller in this case. The procedure gets again stalled and this time achieves convergence to an infeasible local minimum. □

Example (continued). In a second testing, we examined this case more closely. As it is too late to shut the stable door after the horse has gone, we opted to used the ε-descent nonsmooth technique of section 7.1 in order to avoid failure. We call for a nonsmooth step as soon as the MDS simplex shrinks below $\omega = 0.1$ in relative size. Starting with the same initial point, the simultaneous stabilization problem is now satisfactorily solved in a few iterations: four MDS iterations and a single call for the nonsmooth intervention technique of section 7.1. The evolution of the maximum eigenvalue of the quadratic SDP in (23) as a function of the iteration index is the five-element sequence

$$\{6.5072e{+}01,\ 1.5172e{+}01,\ 5.1720e{+}00,\ 2.868e{+}00,\ -1.4778e{+}00\},$$

where the nonsmooth descent step $v_0 \to w$ corresponds to the decrease from $5.1720e{+}00$ to $2.868e{+}00$. Note that since sole stabilization is of interest, the algorithm has been stopped as soon as the maximum eigenvalue was found negative. The associated first-order controller solution is described by the transfer function

$$K(s) = \frac{4.9843s - 4.2577}{4.2783e{-}01s + 6.2861e{-}01}. \qquad \square$$

Example (continued). In our third experiment, we assess the performance of the nonsmooth descent technique alone. We no longer sample the space using MDS. Instead we follow descent steps $v_0 \to w$ proposed by the nonsmooth technique in section 7.1. This option corresponds to a pure spectral bundle method [57]. With the same starting point causing failure of the default MDS, the problem is now solved in seven calls according to the sequence

$$\{65.0718,\ 43.0862,\ 39.8725,\ 20.1852,\ 19.9853,\ 3.5719,\ 2.8877,\ -0.0228\}.$$

The resulting stabilizing controller is

$$K(s) = \frac{0.6029s + 1.115}{0.03361s + 0.1064}.$$

All controllers computed in this application have significantly different pole/zero patterns, but all stabilize the four-plant family. □

8.4. Mixed H_2/H_∞ state-feedback synthesis. Mixed H_2/H_∞-synthesis with state- or output-feedback is one of those archetype problems which cannot be simplified using the projection lemma and resist to linearizing changes of variable like [30]. What remains are special BMI techniques or algorithmic approaches like the one we propose here. The mixed H_2/H_∞ state-feedback synthesis problem is as follows. Given a synthesis state-space representation

$$\begin{cases} \dot{x} = Ax + B_{1,2}w_2 + B_{1,\infty}w_\infty + B_2u, \\ z_2 = C_{1,2}x + D_{12,2}u, \\ z_\infty = C_{1,\infty}x + D_{11,\infty}w_\infty + D_{12,\infty}u, \end{cases}$$

the goal is to compute a state-feedback control law $u = Kx$ such that

- the closed-loop system is asymptotically stable, i.e., $\alpha(A + B_2K) < 0$,
- the H_2-norm of the channel $\|T_{w_2 \to z_2}(K, s)\|_2$ is minimized subject to an H_∞-norm constraint on the channel $\|T_{w_\infty \to z_\infty}(K, s)\|_\infty \le \gamma$.

An example of this type is given in [31], and we reexamine it here using our nonsmooth MDS. We proceed as follows. First a state-feedback gain satisfying both stability and the H_∞ constraint is computed as in section 8.2. In a second phase, the H_2-norm is added and minimized, using an infinite barrier

$$B(K) = \begin{cases} \|T_{w_2\to z_2}(K,s)\|_2 & \text{if } \alpha(A+B_2K) \le -\tau \text{ and } \|T_{w_\infty\to z_\infty}(K,s)\|_\infty \le \gamma, \\ +\infty & \text{otherwise,} \end{cases} \tag{24}$$

now maintaining the constraints of phase 1.

With data imported from [31] and $\gamma = 2$, MDS computed a gain K in 25 MDS iterations within 11.1 s of cpu time. The solution found is

$$K = [\,1.8236,\ 2.5648e{-}01,\ -2.0453e{-}01\,]$$

with $\|T_{w_2\to z_2}(K,s)\|_2 = 7.502e{-}01$. The H_∞-norm constraint was of course active at this point. Note *en passant* that this result outperforms those achieved via the spectral augmented Lagrangian method in [65], which gave $\|T_{w_2\to z_2}(K,s)\|_2 = 0.8384$, and the successive linearization approach in [31], which found $\|T_{w_2\to z_2}(K,s)\|_2 = 0.8930$. Since this problem has multiple local minima, this fact does not imply that any one of those methods is better than any other, except perhaps for cases where a solution without optimality certificate is presented. The solution in [65] is a local minimum, and we checked optimality of our present K by adapting the nonsmooth frequency domain test for program (9) from section 6. This requires not much extra work, as the H_2-norm is smooth (see [13]). We observed that the H_∞-norm is attained at a single frequency, which seems to be rather the rule than the exception.

9. Conclusion. We have proposed a new algorithmic strategy for difficult and even NP-hard synthesis problems in automatic control, which are inaccessible via convexity methods. Our algorithm combines DS methods like Torczon's MDS with spectral bundle techniques, imported from nonsmooth optimization, in order to cope with typical nonsmooth criteria in control like the spectral abscissa, the maximum eigenvalue function, or the H_∞-norm. Our approach is a serious alternative to nonlinear programming algorithms based on bilinear matrix inequalities, as long as the number of *controller* decision variables is not too large. Since our approach avoids Lyapunov variables, it may be used to design small- or medium-size controllers even for very large systems, as witnessed by the Boeing 767 and Heat Flow (HF1) benchmark examples, systems with 55 and 130 states, respectively. As soon as the number of controller gain parameters gets sizable, the search method is often too slow, and pure nonsmooth approaches like spectral bundling perform better. How those should be organized for control applications is discussed in [3]. In a similar vein, a pure nonsmooth and frequency-domain approach for solving multidisk problems is proposed in [4]. A nonsmooth spectral bundle method for solving state-space BMI programs is developed in [64].

Our approach combines MDS with suitable nonsmooth descent steps. This gives a convergence certificate toward critical points, an important feature lacking in all the heuristic approaches proposed to date. We have observed that MDS is fairly insensitive to noise corrupting the function evaluation. This makes it particularly useful in control applications, where objective functions typically result from iterative procedures to compute the H_∞- or H_2-norm. We have noticed that in the neighborhood of nonsmooth surfaces, gradient directions behave irregularly and are often distorted and unreliable, while progress is still achievable with MDS.

In conclusion, we believe the proposed framework is very versatile and can accommodate a vast array of design problems, expanding on those discussed in this paper. Structured feedback design is near at hand, while robust control is currently under investigation.

Acknowledgment. Fruitful discussions with Adrian Lewis and Michael Overton are gratefully acknowledged.

REFERENCES

[1] M. A. Abramson, C. Audet, and J. E. Dennis Jr., *Generalized pattern searches with derivative information*, Math. Program. Ser. B, 100 (2004), pp. 3–25.

[2] J. Ackermann, *Robust Control. Systems with Uncertain Physical Parameters*, Springer-Verlag, Berlin, Heidelberg, New York, 1993.

[3] P. Apkarian and D. Noll, *Nonsmooth H_∞ Synthesis*, submitted.

[4] P. Apkarian and D. Noll, *Nonsmooth Optimization for Multidisk H_∞ Synthesis*, submitted.

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

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

[7] C. Audet and J. E. Dennis Jr., *Mesh adaptive direct search algorithms for constrained optimization*, SIAM J. Optim., to appear.

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

[9] J. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer Series in Operations Research, Springer-Verlag, New York, 2000.

[10] A. Booker, J. E. Dennis Jr., P. Frank, D. Serafini, V. Torczon, and M. Trosset, *A Rigorous Framework for Optimization of Expertise Functions by Surrogates*, Technical report, 1998, pp. 1–24.

[11] G. E. P. Box, *Evolutionary operation: A method for increasing industrial productivity*, Appl. Stat., 6 (1957), pp. 81–101.

[12] S. Boyd, V. Balakrishnan and P. Kabamba, *A bisection method for computing the H_∞ norm of a transfer matrix and related problems*, Math. Control Signals Syst., 2 (1989), pp. 207–219.

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

[14] S. Boyd, L. ElGhaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*, SIAM Stud. Appl. Math. 15, SIAM, Philadelphia, 1994.

[15] M. Bredemann, C. T. Abdallah, and P. Dorato, *Polynomial solutions for simultaneous stabilization*, in Proceedings of the 2nd IFAC Symposium on Robust Control Design, Budapest, Hungary, 1997, pp. 193–198.

[16] M. V. Bredemann, *Feedback Controller Design for Simultaneous Stabilization*, Ph.D. thesis, University of New Mexico, 1995.

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

[18] J. Burke, A. Lewis, and M. Overton, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM J. Optim., 15 (2003), pp. 751–779.

[19] J. Burke, A. Lewis, and M. Overton, *Robust stability and a criss-cross algorithm for pseudospectra*, IMA J. Numer. Anal., 23 (2003), pp. 1–17.

[20] Y. Y. Cao and Y. X. Sun, *Static output feedback simultaneous stabilization: ILMI approach*, Int. J. Control, 70 (1998), pp. 803–814.

[21] B. M. Chen, *H_∞ Control and Its Applications*, Lecture Notes in Control and Inform. Sci. 235, Springer-Verlag, New York, Heidelberg, Berlin, 1998.

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

[23] J. Cullum, W. Donath, and P. Wolfe, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Math. Program. Stud., 3 (1975), pp. 35–55.

[24] E. E. J. Davison, *Benchmark Problems for Control System Design*, Technical report, IFAC Technical Committee Reports, Pergamon Press, Oxford, 1990.

[25] B. Fares, D. Noll, and P. Apkarian, *Robust control via sequential semidefinite programming*, SIAM J. Control Optim., 40 (2002), pp. 1791–1820.
[26] P. Gahinet and P. Apkarian, *Numerical computation of the L_∞ norm revisited*, in Proceedings of the IEEE Conference on Decision and Control, 1992.
[27] P. Gahinet and P. Apkarian, *A linear matrix inequality approach to H_∞ control*, Int. J. Robust Nonlinear Control, 4 (1994), pp. 421–448.
[28] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox*, The MathWorks Inc., Natick, MA, 1995.
[29] D. Gangsaas, K. Bruce, J. Blight, and U.-L. Ly, *Application of modern synthesis to aircraft control: Three case studies*, IEEE Trans. Automat. Control, AC-31 (1986), pp. 995–1014.
[30] J. C. Geromel, P. L. D. Peres, and J. Bernussou, *On a convex parameter space method for linear control design of uncertain systems*, SIAM J. Control Optim., 29 (1991), pp. 381–402.
[31] A. Hassibi, J. How, and S. Boyd, *A pathfollowing method for solving BMI problems in control*, in Proceedings of the American Control Conference, 1999, pp. 1385–1389.
[32] C. Helmberg and F. Rendl, *Spectral bundle method for semidefinite programming*, SIAM J. Optim., 10 (2000), pp. 673–696.
[33] D. Henrion and M. Sebek, *LMIs and polynomial methods in control: Illustrative examples*, in IFAC/IEEE Symposium on Advances in Control Education, Australia, 2000.
[34] D. Henrion, S. Tarbouriech, and M. Sebek, *Rank-one LMI approach to simultaneous stabilization of linear systems*, Syst. Control Lett., 11 (1998), pp. 167–172.
[35] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms* II: *Advanced Theory and Bundle Methods*, Grundlehren Math. Wiss. 306, Springer-Verlag, New York, 1993.
[36] C. Hol, C. Scherer, E. van der Meché, and O. Bosgra, *A nonlinear SDP approach to fixed-order controller synthesis and comparison with two other methods applied to an active suspension system*, Eur. J. Control, 9 (2003), pp. 13–28.
[37] R. Hook and T. A. Jeeves, *"Direct search" solution of numerical and statistical problems*, J. Assoc. Comput. Mach., 8 (1961), pp. 212–229.
[38] H. P. Horisberger and P. R. Belanger, *Solution of the optimal constant output feedback problem by conjugate gradients*, IEEE Trans. Automat. Control, 19 (1974), pp. 434–435.
[39] Y. S. Hung and A. G. J. MacFarlane, *Multivariable Feedback: A Classical Approach*, Lecture Notes in Control and Inform. Sci. 40, Springer-Verlag, New York, Heidelberg, Berlin, 1982.
[40] J. Imae and T. Furudate, *A design method for fixed-order H_∞ controllers via bilinear matrix inequalities*, in Proceedings of the American Control Conference, San Diego, CA, 1999, pp. 1876–1880.
[41] F. Jarre, *A QQP-Minimization Algorithm Method for Semidefinite and Smooth Nonconvex Programs*, Technical report.
[42] L. H. Keel, S. P. Bhattacharyya, and J. W. Howze, *Robust control with structured perturbations*, IEEE Trans. Automat. Control, 36 (1988), pp. 68–77.
[43] K. C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Math. 1133, A. Dold and B. Eckmann, eds., Springer-Verlag, Berlin, Heidelberg, New York, 1985.
[44] K. C. Kiwiel, *A linearization algorithm for optimizing control systems subject to singular value inequalities*, IEEE Trans. Automat. Control, AC-31 (1986), pp. 595–602.
[45] M. Kocvara and M. Stingl, *A code for convex nonlinear and semidefinite programming*, Optim Methods Software, 18 (2003), pp. 317–333.
[46] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.
[47] F. Leibfritz, *COMPL$_e$IB, Constraint Matrix-Optimization Problem Library: A Collection of Test Examples for Nonlinear Semidefinite Programs, Control System Design and Related Problems*, Technical report, Universität Trier, 2003.
[48] F. Leibfritz and E. M. E. Mostafa, *Trust region methods for solving the optimal output feedback design problem*, Int. J. Control, 76 (2000), pp. 501–519.
[49] F. Leibfritz and E. M. E. Mostafa, *An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems*, SIAM J. Optim., 12 (2002), pp. 1048–1074.
[50] C. Lemarechal, *Nondifferentiable optimization*, in Optimization, G. L. Nemhauser and A. H. G. Rinnooy Kan, eds., Elsevier North-Holland, New York, 1989, pp. 529–572.
[51] M. Marazzi and J. Nocedal, *Wedge trust region methods for derivative free optimization*, Math. Program. Ser. A, 91 (2002), pp. 289–305.
[52] K. I. M. McKinnon, *Convergence of the Nelder–Mead simplex method to a nonstationary*

point, SIAM J. Optim., 9 (1998), pp. 148–158.
[53] K. Miettinen and M. M. Mäkelä, *An interactive method for nonsmooth multiobjective with an application to optimal control*, Optim. Methods Software, 2 (1993), pp. 31–44.
[54] L. Mosheyev and M. Zibulevsky, *Penalty/barrier multiplier algorithm for semidefinite programming*, Optim. Methods Software, 13 (2000), pp. 235–261.
[55] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J., (1965), pp. 441–461.
[56] A. Nemirovskii, *Several NP-hard problems arising in robust stability analysis*, Math. Control Signals Syst., 6 (1993), pp. 99–105.
[57] D. Noll and P. Apkarian, *Spectral bundle methods for non-convex maximum eigenvalue functions: First-order methods*, Math. Program. Ser. B, to appear.
[58] D. Noll and P. Apkarian, *Spectral bundle methods for non-convex maximum eigenvalue functions: Second-order methods*, Math. Program. Ser. B, to appear.
[59] D. Noll, M. Torki, and P. Apkarian, *Partially augmented Lagrangian method for matrix inequality constraints*, SIAM J. Optim., 15 (2004), pp. 161–184.
[60] F. Oustry, *A second-order bundle method to minimize the maximum eigenvalue function*, Math. Program. Ser. A, 89 (2000), pp. 1–33.
[61] E. Polak and S. Salcudean, *On the design of linear multivariable feedback systems via constrained nondifferentiable optimization in H_∞ spaces*, IEEE Trans. Automat. Control, AC-34 (1989), pp. 268–276.
[62] E. Polak and Y. Wardi, *A nondifferentiable optimization algorithm for the design of control systems subject to singular value inequalities over a frequency range*, Automatica, 18 (1982), pp. 267–283.
[63] W. Spendley, G. R. Hext, and F. R. Himsworth, *Sequential application of simplex designs in optimisation and evolutionary operation*, Technometrics, 4 (1962), pp. 441–461.
[64] J. Thevenet, P. Apkarian, and D. Noll, *A bundle-type method for "large scale" output feedback control design*, SIAM Conference on Optimization, Stockholm, Sweden, 2005.
[65] J. Thevenet, D. Noll, and P. Apkarian, *Nonlinear spectral SDP method for BMI-constrained problems: Applications to control design*, in ICINCO Proceedings, Setúbal, Portugal, 2004, pp. 237–248.
[66] V. Torczon, *On the convergence of the multidirectional search algorithm*, SIAM J. Optim., 1 (1991), pp. 123–145.
[67] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim., 7 (1997), pp. 1–25.
[68] L. Trefethen, *Pseudospectra of linear operators*, SIAM Rev., 39 (1997), pp. 383–406.