Journal of Convex Analysis
Volume 16 (2009), No. 3&4, 641–666

A Proximity Control Algorithm to Minimize Nonsmooth and Nonconvex Semi-Infinite Maximum Eigenvalue Functions

Pierre Apkarian
ONERA-CERT, av. Edouard Belin, 31055 Toulouse, France

Dominikus Noll
Université Paul Sabatier, Institut de Mathématiques, 31062 Toulouse, France

Olivier Prot
*Université de Limoges, Département de Mathématiques,
123, av. Albert Thomas, 87060 Limoges, France*

Dedicated to Stephen Simons on the occasion of his 70th birthday.

Received: March 4, 2008

Proximity control is a well-known mechanism in bundle method for nonsmooth optimization. Here we show that it can be used to optimize a large class of nonconvex and nonsmooth functions with additional structure. This includes for instance nonconvex maximum eigenvalue functions, and also infinite suprema of such functions.

Keywords: Nonsmooth calculus, nonsmooth optimization, Clarke subdifferential, spectral bundle method, maximum eigenvalue function, semi-infinite problem, H_∞-norm

1. Introduction

Proximity control for bundle methods has been known for a long time, but its use is too often restricted to convex optimization, where its full strength cannot be gauged. As we shall demonstrate, as soon as the management of the proximity control parameter follows the lines of a trust region strategy, many nonconvex and nonsmooth locally Lipschitz functions can be optimized. In contrast, in the convex case, the proximity control parameter can usually be frozen, which suggests that under convexity the full picture is not seen, and something of the essence is missing to understand this mechanism. The method we discuss here will be developed in the context of a specific application, because that is where the motivation of our work arises from, but we will indicate in which way the method can be generalized to much larger classes of functions.

The application we have in mind is optimizing the H_∞-norm, which is structurally of the form

$$f(x) = \sup_{\omega\in[0,\infty]} \lambda_1\left(F(x,\omega)\right), \tag{1}$$

where $F : \mathbb{R}^n \times [0,\infty] \to \mathbb{S}^m$ is an operator with values in the space $\mathbb{S}^m$ of $m \times m$ symmetric or Hermitian matrices, equipped with the scalar product $X \bullet Y = \mathrm{Tr}(XY)$,

ISSN 0944-6532 / $ 2.50 © Heldermann Verlag

and where λ_1 denotes the maximum eigenvalue function on $\mathbb{S}^m$. We assume that F is jointly continuous in the variable (x, ω) and of class C^2 in the variable x, so that $F''(x, \omega)$ is still jointly continuous. Here derivatives always refer to the variable x. Our exposition will show how these hypotheses can easily be relaxed. The program we wish to solve is

$$\min_{x \in \mathbb{R}^n} f(x), \tag{2}$$

where f has the form (1).

The approach presented here was originally developed in the context of eigenvalue optimization, and [9] gives an overview of the history. The bases for the present extension to the semi-infinite case were laid in [3, 5, 2, 47, 12, 6, 7]. Our method is inspired by Helmberg and Rendl's spectral bundle method [28], where large semidefinite programs arising as relaxations of quadratic integer programming problems are developed. Helmberg and Rendl optimize a convex eigenvalue function of the form $\lambda_1(A(x))$, where $A : \mathbb{R}^n \to \mathbb{S}^m$ is affine. This method has also antecedents in classical bundling, like Lemaréchal [37, 38, 39, 40] or Kiwiel [34, 35, 33]. Extensions of the convex case to include bound constraints are given in [26].

Optimization of the H_∞-norm is an important application in feedback control synthesis, which has been pioneered by E. Polak and co-workers. See for instance [41, 42, 45] and the references given there. Our own approach to optimizing the H_∞ norm is developed in [5, 2, 6]. A version for maximum eigenvalue functions is presented in [9].

The structure of the paper is as follows. After some preparation in Sections 2 and 3, the core of the algorithm is explained in Section 5. The algorithm is presented in Section 7. Convergence proofs for the inner and outer loop follow in Sections 8 and 9. Numerical experiments in H_∞-synthesis are presented in Section 10.

2. Preparation

Observe that our objective function has the form

$$f(x) = \max_{\omega \in [0,\infty]} f(x, \omega), \tag{3}$$

where each $f(x, \omega) = \lambda_1 (F(x, \omega))$ is a composite maximum eigenvalue function. Recall that the maximum eigenvalue function $\lambda_1 : \mathbb{S}^m \to \mathbb{R}$ is the support function of the compact convex set

$$\mathcal{C} = \{Z \in \mathbb{S}^m : Z \succeq 0,\ \operatorname{Tr}(Z) = 1\},$$

where $\succeq 0$ means positive semidefinite. In other words,

$$f(x) = \max_{\omega \in [0,\infty]} \max_{Z \in \mathcal{C}} Z \bullet F(x, \omega). \tag{4}$$

Due to compactness of $\mathcal{C}$ and $[0, \infty]$, the suprema in (4) are attained. This suggests introducing an approximation of f in a neighbourhood of x, which is

$$\begin{aligned} \phi(y, x) &= \max_{\omega \in [0,\infty]} \lambda_1 \left(F(x, \omega) + F'(x, \omega)(y - x)\right) \qquad (5) \\ &= \max_{\omega \in [0,\infty]} \max_{Z \in \mathcal{C}} Z \bullet \left(F(x, \omega) + F'(x, \omega)(y - x)\right) \end{aligned}$$

where the derivative $F'(x,\omega)$ refers to the variable x. As (5) uses a Taylor expansion of the operator F in a neighbourhood of x, we expect $\phi(y,x)$ to be a good model of f for y near x. This is confirmed by the following

Lemma 2.1. *Let $B \subset \mathbb{R}^n$ be a bounded set. Then there exists a constant $L > 0$ such that*

$$|f(y) - \phi(y,x)| \leq L\|y-x\|^2$$

for all $x, y \in B$.

Proof. By Weil's theorem we have

$$\lambda_m(E) \leq \lambda_1(A+E) - \lambda_1(A) \leq \lambda_1(E)$$

for all matrices $A, E \in \mathbb{S}^m$. We apply this with $A = F(y,\omega)$ and $A + E = F(x,\omega) + F'(x,\omega)(y-x)$. Now observe that by hypothesis on F there exists $L > 0$ such that

$$\sup_{z\in B} \sup_{\omega\in[0,\infty]} \|F''(z,\omega)\| \leq L.$$

This proves $E = \mathcal{O}(\|y-x\|^2)$, uniformly over $x, y \in B$ and uniformly over $\omega \in [0,\infty]$. □

The following is a specific property of the H_∞-norm, which can be exploited algorithmically. A proof can be found in [13] or [12].

Lemma 2.2. *The set $\Omega(x) = \{\omega \in [0,\infty] : f(x) = f(x,\omega)\}$ is either finite, or $\Omega(x) = [0,\infty]$.*

We call $\Omega(x)$ the set of active frequencies. A system where $\Omega(x) = [0,\infty]$ is called *all-pass*. This is rarely encountered in practice.

For later use let us mention a different way to represent the convex model $\phi(y,x)$. We introduce the notations

$$\alpha(\omega,Z) = Z \bullet F(x,\omega) \in \mathbb{R}, \qquad g(\omega,Z) = F'(x,\omega)^\star Z \in \mathbb{R}^n.$$

and we let

$$\mathcal{G} = \text{co}\left\{(\alpha(\omega,Z), g(\omega,Z)) : \omega \in [0,\infty], Z \in \mathcal{C}\right\},$$

where co(X) is the convex hull of X. Then we have the following equivalent representation of the model:

$$\phi(y,x) = \max\{\alpha + g^\top(y-x) : (\alpha,g) \in \mathcal{G}\}. \tag{6}$$

3. Tangent program

Suppose x is the current iterate of our algorithm to be designed. In order to generate trial steps away from x, we will recursively construct approximations $\phi_k(y,x)$ of $\phi(y,x)$ of increasing quality. Using the form (6) we will choose suitable subsets $\mathcal{G}_k$ of the set $\mathcal{G}$ and define

$$\phi_k(y,x) = \max\{\alpha + g^\top(y-x) : (\alpha,g) \in \mathcal{G}_k\}. \tag{7}$$

Clearly $\phi_k \leq \phi$, and a suitable strategy will assure that the ϕ_k get closer to the model ϕ as k increases. Once the model $\mathcal{G}_k$ is formed, a new trial step y^{k+1} is generated by solving the tangent program

$$\min_{y \in \mathbb{R}^n} \phi_k(y,x) + \tfrac{\delta_k}{2}\|y-x\|^2, \tag{8}$$

where $\delta_k > 0$ is the proximity control parameter, which will be adjusted anew at each step k. Here we make the implicit assumption that solving (8) is much easier than solving the original problem.

Suppose the solution of (8) is y^{k+1}. Following standard terminology in nonsmooth optimization, y^{k+1} will be called a serious step if it is accepted to become the new iterate x^+. On the other hand, if y^{k+1} is not satisfactory and has to be rejected, it is called a null step. In that case, a new model $\mathcal{G}_{k+1}$ is built, using information from the previous $\mathcal{G}_k$, and integrating information provided by y^{k+1}. The proximity parameter is updated, $\delta_k \to \delta_{k+1}$, and the tangent program is solved again. In other words, the construction of the $\mathcal{G}_k$ in (7) is recursive.

In order to guarantee convergence of our method, we have isolated three basic properties of the sets $\mathcal{G}_k$. The most basic one is that $\phi_k(x,x) = \phi(x,x) = f(x)$, and this is covered by the following:

Lemma 3.1. *Let $\omega_0 \in \Omega(x)$ be any of the active frequencies at x. Choose a normalized eigenvector e_0 associated with the maximum eigenvalue $f(x) = \lambda_1(F(x,\omega_0))$ of $F(x,\omega_0)$, and let $Z_0 := e_0 e_0^\top \in \mathcal{C}$. If $(\alpha(\omega_0,Z_0), g(\omega_0,Z_0)) \in \mathcal{G}_k$, then $\phi_k(x,x) = \phi(x,x) = f(x)$.* □

A second more sophisticated property of our model $\phi_k(\cdot,x)$ is that it is improved at each step by adding suitable affine support functions of $\phi(\cdot,x)$, referred to as cutting planes. Suppose a trial step y^{k+1} away from x is computed via (8), based on the current model $\phi_k(\cdot,x)$ with approximation $\mathcal{G}_k$ and proximity control parameter δ_k. If y^{k+1} fails because the progress in the function value is not satisfactory (null step), we add an affine support function of $\phi(\cdot,x)$ to the next model $\phi_{k+1}(\cdot,x)$. This will assure that the bad step y^{k+1} will be cut away at the next iteration $k+1$, hopefully paving the way for something better to come. What we have in mind is made precise by the following:

Lemma 3.2. *Let $\omega_{k+1} \in [0,\infty]$ and $Z_{k+1} \in \mathcal{C}$ be where the maximum (5) for the solution y^{k+1} of (8) is attained, that is,*

$$\phi(y^{k+1},x) = Z_{k+1} \bullet \left(F(x,\omega_{k+1}) + F'(x,\omega_{k+1})(y^{k+1}-x)\right).$$

If $(\alpha(\omega_{k+1},Z_{k+1}), g(\omega_{k+1},Z_{k+1})) \in \mathcal{G}_{k+1}$, then we have $\phi_{k+1}(y^{k+1},x) = \phi(y^{k+1},x)$.

We need yet another support function to improve the model, and this is usually called the aggregation element. The idea is as follows. As we keep updating our approximation and $\mathcal{G}_k$, we expect our model $\phi_k(\cdot,x)$ to get closer to f. The easiest way to assure this would seem to let the sequence increase: $\mathcal{G}_k \subset \mathcal{G}_{k+1}$, so that previous attempts (null steps) are perfectly memorized. However, this would quickly lead to overload. To avoid this, we drive ϕ_k toward ϕ in a more sophisticated way by a clever use of the information obtained from the null steps. As we have seen, adding a cutting plane avoids the last unsuccessful step y^{k+1}. This could be considered a reality check, where ϕ_k is matched with ϕ. What is further needed is relating ϕ_{k+1} to its past, ϕ_k, and this is what aggregation is about.

According to the definition of y^{k+1} as minimum of the tangent program (8) we have $0 \in \partial\phi_k(y^{k+1}, x) + \delta_k(y^{k+1} - x)$. The way ϕ_k is built in (7) shows that this may be written as

$$0 = \sum_{i=1}^{r} \tau_i^* g_i^* + \delta_k(y^{k+1} - x) \tag{9}$$

for certain $\tau_i^* \geq 0$ summing up to 1, and $(\alpha_i^*, g_i^*) \in \mathcal{G}_k$. We let

$$\alpha^* = \sum_{i=1}^{r} \tau_i^* \alpha_i^*, \qquad g^* = \sum_{i=1}^{r} \tau_i^* g_i^*, \tag{10}$$

and keep $(\alpha^*, g^*) \in \mathcal{G}_{k+1}$. Notice that this pair belongs indeed to $\mathcal{G}$ by convexity, and because $\mathcal{G}_k \subset \mathcal{G}$.

Altogether, we have now isolated three properties, which our approximations $\mathcal{G}_k$ have to satisfy:

($G1$) $\mathcal{G}_k$ contains at least one pair $(\alpha(\omega_0, Z_0), g(\omega_0, Z_0))$, where $\omega_0 \in \Omega(x)$ is an active frequency, $Z_0 = e_0 e_0^\top$ for a normalized eigenvector e_0 of $F(x, \omega_0)$ associated with $\lambda_1(F(x, \omega_0))$.

($G2$) For every null step y^{k+1}, $\mathcal{G}_{k+1}$ contains a pair $(\alpha(\omega_{k+1}, Z_{k+1}), g(\omega_{k+1}, Z_{k+1}))$, where ω_{k+1}, Z_{k+1} satisfy $\phi(y^{k+1}, x) = Z_{k+1} \bullet [F(x, \omega_{k+1}) + F'(x, \omega_{k+1})(y^{k+1} - x)]$.

($G3$) If $\delta_k(x - y^{k+1}) \in \partial\phi_k(y^{k+1}, x)$ for a null step y^{k+1}, then $\mathcal{G}_{k+1}$ contains the aggregate pair (α^*, g^*) satisfying (9) and (10).

As we shall see, these properties guarantee a weak form of convergence of our method. Practical considerations, however, require richer sets $\mathcal{G}_k$ which in general are no longer finitely generated. The way these are built is explained in the next section. To conclude, we state the consequences of the three axioms in the following

Lemma 3.3. *Axioms* ($G1$)–($G3$) *guarantee that* $\phi_k(x, x) = \phi(x, x) = f(x)$, *that* $\phi_{k+1}(y^{k+1}, x) = \phi(y^{k+1}, x)$, *that* $\phi_{k+1}(y^{k+1}, x) \geq \phi_k(y^{k+1}, x)$, *and that relation* (9) *is satisfied.*

4. Solving the tangent program

Our numerical experience shows that it is useful to generate approximations $\mathcal{G}_k$ larger than what is required by the minimal axioms ($G1$)–($G3$). More precisely, we will keep the procedures in ($G2$) and ($G3$), but improve on ($G1$).

Consider the case where the set $\Omega(x)$ of active frequencies is finite. We let Ω_k be a finite extension of $\Omega(x)$, enriched along the lines discussed in [5]. For every $\omega \in \Omega_k$, we allow all sets $Z_\omega \in \mathcal{C}$ of the form

$$Z_\omega = Q_\omega Y_\omega Q_\omega^\top, \quad Y_\omega \succeq 0, \ \mathrm{Tr}(Y_\omega) = 1, \tag{11}$$

where the columns of Q_ω are an orthonormal basis of some invariant subspace of $F(x, \omega)$, containing the eigenspace associated with the maximum eigenvalue. This assures axiom ($G1$), because $\omega_0 \in \Omega_k$ at all times, and because e_0 belongs to the span of the columns of Q_{ω_0}. Similarly, to force ($G2$), for every null step y^{k+1} we simply have to keep $\omega_{k+1} \in \Omega_{k+1}$

and let the normalized eigenvector e_{k+1} of $F(x, \omega_{k+1}) + F'(x, \omega_{k+1})(y^{k+1} - x)$ associated with λ_1 be in the span of the columns of $Q_{\omega_{k+1}}$. Then

$$\mathcal{G}_k = \{(\alpha(\omega, Z_\omega), g(\omega, Z_\omega)) : \omega \in \Omega_k, Y_\omega \succeq 0, \mathrm{Tr}(Y_\omega) = 1\} \cup \{(\alpha^*, g^*)\}, \tag{12}$$

where (α^*, g^*) is the aggregate from the previous sweep $k-1$. Notice that $\mathrm{co}(\mathcal{G}_k) \not\subset \mathrm{co}(\mathcal{G}_{k+1})$ in general, because the active frequencies change at each step.

Let us now pass to the more practical aspect on how setting up and solving the tangent program (8) at each step. Writing the tangent program in the form

$$\min_{y \in \mathbb{R}^n} \max_{(\alpha,g) \in \mathrm{co}(\mathcal{G}_k)} \alpha + g^\top (y - x) + \frac{\delta_k}{2} \|y - x\|^2$$

we can use Fenchel duality to swap the min and max operators. The then inner infimum over y is unconstrained and can be computed explicitely, which leads to $y = x - \delta_k^{-1} g$. Substituting this back gives the following form of the dual program

$$\max_{(\alpha,g) \in \mathrm{co}(\mathcal{G}_k)} \alpha - \tfrac{1}{2\delta_k} \|g\|^2.$$

This abstract program takes the following more concrete form if we use the sets $\mathcal{G}_k$ in (12):

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{\omega \in \Omega_k} Y_\omega \bullet Q_\omega^\top F(x,\omega) Q_\omega + \tau \alpha^* - \frac{1}{2\delta_k} \left\| \sum_{\omega \in \Omega_k} F'(x,\omega)^\star \left[Q_\omega Y_\omega Q_\omega^\top \right] + \tau g^* \right\|^2 \\ \text{subject to} & \tau \geq 0, Y_\omega \succeq 0 \\ & \tau + \displaystyle\sum_{\omega \in \Omega_k} \mathrm{Tr}(Y_\omega) = 1 \end{array}$$

The reader will recognize this as a semidefinite program. The return formula takes the explicit form

$$y^{k+1} = x - \frac{1}{\delta_k} \left(\sum_{\omega \in \Omega_k} F'(x,\omega)^\star \left[Q_\omega Y_\omega^* Q_\omega^\top) \right] + \tau^* g^* \right), \tag{13}$$

where (Y^*, τ^*) is the dual optimal solution.

Finally, if we assume that the multiplicity of each maximum eigenvalue is 1, we may further simplify the dual program. This is most often the case in practice. Indeed, in this case the matrices $Z_\omega = e_\omega y_\omega e_\omega^\top$ are of rank 1, so in particular $y_\omega = 1$ is scalar. In other words, we have a finite set of $\alpha_\omega = e_\omega^\top F(x,\omega) e_\omega$ and $g_\omega = F'(x,\omega)^\star e_\omega e_\omega^\top$, $\omega \in \Omega_k$, to which we add the aggregate element (α^*, g^*), and where ω_k required for the last cutting plane is included in Ω_k to assure $(G2)$. Arranging this finite set into a sequence $r = 1, \ldots, R_k$, we can write ϕ_k as

$$\phi_k(y, x) = \max_{r=1,\ldots,R_k} \alpha_r + g_r^\top (y - x),$$

where $R_k = |\Omega_k| + 1$.

Solving the tangent program at stage k can now be obtain by convex duality. We have the primal form of (8):

$$\min_{y\in\mathbb{R}^n} \max_{r=1,\dots,R_k} \alpha_r + g_r^\top(y-x) + \tfrac{\delta_k}{2}\|y-x\|^2.$$

Standard convex duality shows that the concave dual of this is

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{r=1}^{R_k} \tau_r \alpha_r - \frac{1}{2\delta_k}\left\|\sum_{r=1}^{R_k} \tau_r g_r\right\|^2 \\ \text{subject to} & \displaystyle\sum_{r=1}^{R_k} \tau_r = 1 \\ & 0 \leq \tau_r \leq 1, r = 1,\dots,R_k \end{array}$$

with unknown variable τ. This is the concave form of a convex quadratic program. The return formula to recover the solution of the primal from the solution of the dual is

$$y^{k+1} = x - \frac{1}{\delta_k}\sum_{r=1}^{R_k} \tau_r^* g_r,$$

where τ^* is the optimal solution of the dual.

5. Management of the proximity parameter

At the core of our method is the management of the proximity control parameter δ_k in (8). In order to decide whether the solution y^{k+1} of (8) can be accepted as the new iterate x^+, we compute the control parameter

$$\rho_k = \frac{f(x) - f(y^{k+1})}{f(x) - \phi_k(y^{k+1}, x)},$$

which relates our current model $\phi_k(\cdot, x)$ to the truth f. If $\phi_k(\cdot, x)$ is a good model of f, we expect $\rho_k \approx 1$. But we accept y^{k+1} already when $\rho_k \geq \gamma$, (serious step), where the reader might for instance imagine $\gamma = .25$. We say that the agreement between f and ϕ_k is good when $\rho_k \geq \Gamma$, where $\Gamma = .75$ makes sense, and we call it bad when $\rho_k < \gamma$. So we accept steps which are *not bad*. Notice that bad includes in particular those cases where $\rho_k < 0$. As the denominator in ρ_k is always > 0, $\rho_k < 0$ corresponds to those cases where y^{k+1} is not even a descent step for f.

The question is what we should do when y^{k+1} is bad (null step). Here we compute a second control quotient

$$\widetilde{\rho}_k = \frac{f(x) - \phi(y^{k+1}, x)}{f(x) - \phi_k(y^{k+1}, x)}$$

which compares the models ϕ and ϕ_k. Introduce a similar parameter $\widetilde{\gamma} \in (0, 1)$, where $\gamma < \widetilde{\gamma}$, but typically only slightly. We say that agreement between ϕ and ϕ_k is bad if $\widetilde{\rho}_k < \widetilde{\gamma}$, and not bad otherwise. Our decision is now as follows. If $\rho_k < \gamma$ and also $\widetilde{\rho}_k < \widetilde{\gamma}$, then we keep the proximity control parameter unchanged and rely on cutting planes and aggregation, being reluctant to increase δ_k prematurely. Instead we rely on driving ϕ_k closer to ϕ, hoping that this will also bring it closer to f. On the other hand, when $\rho_k < \gamma$, but $\widetilde{\rho}_k \geq \widetilde{\gamma}$, then we are in the more delicate situation where ϕ_k is already reasonably close to ϕ, yet our trial steps do not work because ϕ itself is too far from f. Here it will not suffice to drive ϕ_k even closer to ϕ. We also need to bring $\phi(\cdot, x)$ closer to f. This could only be achieved by tightening proximity control, that is, by increasing δ_k. This is what is done in step 7 of the algorithm. Notice however that even here we continue driving ϕ_k toward ϕ via cutting planes and aggregation, so this process is never stopped.

Finally, if a serious step is accepted with $\rho_k > \Gamma$, we can take confidence in our model, and this is where we relax proximity control by reducing δ_k for the next sweep. This is arranged in step 4 of the algorithm. It may therefore happen that by a succession of such successful steps δ_k approaches 0. This in indeed the ideal case, which in a trust region context corresponds to the case where the trust region constraint becomes inactive.

Even though this is well-known, it is useful to compare the proximity control model (8) to the trust region approach

$$\begin{array}{ll} \text{minimize} & \phi_k(y, x) \\ \text{subject to} & \|y - x\| \leq t_k \end{array} \tag{14}$$

where t_k is the trust region radius. Indeed, following [30, II, Prop. 2.2.3, p. 291] solutions of (8) and (14) are in one-to-one correspondence in the sense that if y^{k+1} solves (14) such that the constraint is active with Lagrange multiplier $\lambda_k > 0$, then y^{k+1} solves (8) with $\delta_k = \lambda_k$. Conversely, if y^{k+1} solves (8) with proximity parameter δ_k, then it solves (14) with $t_k = \|y^{k+1} - x\|$. It is now clear that increasing δ_k corresponds to decreasing t_k, and conversely.

6. Recycling subgradients

Apart from the management of the proximity control parameter there is yet another important difference between convex and nonconvex programs. Namely, in convex bundling the working model $\phi_k(\cdot, x)$ is not thrown away if a serious step $x \longrightarrow x^+$ is taken. Indeed, affine support functions or aggregates of f which have been found during the inner loop at x are still useful at x^+, because they remain affine minorants of f. This is no longer the case if f is non-convex. In order to recycle some of the information from iteration x to the next step x^+, we have to exploit the specific structure of our objective (2). Indeed, let $m(y) = \alpha + g^\top(y - x)$ be one of the planes which contribute to the working model $\phi_k(\cdot, x)$ at x. Then either $\alpha = \alpha(\omega, Z)$ and $g = g(\omega, Z)$, or $\alpha = \alpha^*$, $g = g^*$ in case of an aggregate. In the first case we have $g = F'(x, \omega)^* Z$ for some $Z \in \mathcal{C}$. If we put

$$g^+ = F'(x^+, \omega)^* Z, \qquad \alpha^+ = Z \bullet F(x^+, \omega),$$

then the plane $m^+(y) = \alpha^+ + g^{+\top}(y - x^+)$ is the recycled version of $m(\cdot)$ at the new point x^+. For aggregate planes this is more complicated, even though principally possible, as we need to de-aggregate what was aggregated previously.

7. The algorithm

In this section we present our algorithm.

Proximity control algorithm for $\min_{x\in\mathbb{R}^n}\max_{\omega\in[0,\infty]} f(x,\omega)$

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

0. **Initialize outer loop**. Choose initial x such that $f(x) < \infty$.
1. **Outer loop**. Stop at the current x if $0 \in \partial f(x)$. Otherwise compute $\Omega(x)$ and continue with inner loop.
2. **Initialize inner loop**. Choose initial approximation $\mathcal{G}_1$, which contains at least $(\alpha(\omega_0, Z_0), g(\omega_0, Z_0))$, where $\omega_0 \in \Omega(x)$ and e_0 is normalized eigenvector associated with $\lambda_1(F(x,\omega_0))$. Possibly enrich $\mathcal{G}_1$ as in (12) via finite extension $\Omega_1 \supset \Omega(x)$. Initialize $\delta_1 > 0$. If old memory element for δ is available, use it to initialize δ_1. Put inner loop counter $k = 1$.
3. **Trial step**. At inner loop counter k for given $\mathcal{G}_k$ and proximity parameter δ_k, solve tangent program

   $$\min_{y\in\mathbb{R}^n} \phi_k(y,x) + \frac{\delta_k}{2}\|y-x\|^2.$$

   The solution is y^{k+1}.
4. **Test of progress**. Check whether

   $$\rho_k = \frac{f(x) - f(y^{k+1})}{f(x) - \phi_k(y^{k+1},x)} \geq \gamma.$$

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

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

   and go back to step 1. If $\rho_k < \gamma$ continue with step 5 (null step).
5. **Cutting plane**. Select a frequency ω_{k+1} where $\phi(y^{k+1},x)$ is active and pick a normalized eigenvector e_{k+1} associated with the maximum eigenvalue of $F(x,\omega_{k+1}) + F'(x,\omega_{k+1})(y^{k+1}-x)$. Assure $\Omega_{k+1} \supset \Omega(x) \cup \{\omega_0, \omega_{k+1}\}$ and that e_{k+1} is among the columns of $Q_{\omega_{k+1}}$, e_0 among the columns of Q_{ω_0}. Possibly enrich $\mathcal{G}_{k+1}$ as in (12) by adding more frequencies to Ω_{k+1}.
6. **Aggregation**. Compute aggregate pair (α^*, g^*) via (9), (10) based on y^{k+1}, and keep $(\alpha^*, g^*) \in \mathcal{G}_{k+1}$.
7. **Proximity control**. Compute control parameter

   $$\widetilde{\rho}_k = \frac{f(x) - \phi(y^{k+1},x)}{f(x) - \phi_k(y^{k+1},x)}.$$

   Update proximity parameter δ_k as

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

   Increase inner loop counter k and go back to step 3.

8. Finiteness of inner loop

We have to show that the inner loop terminates after a finite number of updates k with a new iterate $y^{k+1} = x^+$. This will be proved in the next two Lemmas.

Lemma 8.1. *Suppose the inner loop creates an infinite sequence y^{k+1} of null steps with $\rho_k < \gamma$. Then there must be an instant k_0 such that the control parameter $\widetilde{\rho}_k$ satisfies $\widetilde{\rho}_k < \widetilde{\gamma}$ for all $k \geq k_0$.*

Proof. Indeed, by assumption none of the trial steps y^{k+1} passes the acceptance test in step 4, so $\rho_k < \gamma$ at all times k. Suppose now that $\widetilde{\rho}_k \geq \widetilde{\gamma}$ for an infinity of times k. Then according to step 7 the proximity parameter δ_k is increased infinitely often, meaning $\delta_k \to \infty$.

Using the fact that y^{k+1} is the optimal solution of the tangent program gives $0 \in \partial\phi_k(y^{k+1}, x) + \delta_k(y^{k+1} - x)$. By convexity of ϕ_k we have

$$-\delta_k(y^{k+1} - x)^\top(x - y^{k+1}) \leq \phi_k(x, x) - \phi_k(y^{k+1}, x).$$

Using $\phi_k(x, x) = f(x)$, assured by keeping $\omega_0 \in \Omega_k$ and $Z_0 \in \mathcal{C}_k$ at all times (Lemma 3.1), we deduce

$$\frac{\delta_k\|y^{k+1} - x\|^2}{f(x) - \phi_k(y^{k+1}, x)} \leq 1. \tag{15}$$

Now we expand

$$\begin{array}{rcll} \widetilde{\rho}_k & = & \rho_k + \dfrac{f(y^{k+1}) - \phi(y^{k+1}, x)}{f(x) - \phi_k(y^{k+1}, x)} & \\ & \leq & \rho_k + \dfrac{L\|y^{k+1} - x\|^2}{f(x) - \phi_k(y^{k+1}, x)} & \text{(using Lemma 2.1)} \\ & \leq & \rho_k + \dfrac{L}{\delta_k} & \text{(using (15)).} \end{array}$$

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

So far we know that if the inner loop turns forever, this implies that $\rho_k < \gamma$ and $\widetilde{\rho}_k < \widetilde{\gamma}$ from some counter k_0 onwards. We show that this cannot happen, by proving the following

Lemma 8.2. *Suppose $\rho_k < \gamma$ and $\widetilde{\rho}_k < \widetilde{\gamma}$ for all $k \geq k_0$. Then $0 \in \partial f(x)$.*

Proof. 1) Step 7 of the algorithm tells us that we are in the case where the proximity parameter is no longer increased, and remains therefore constant. Let us say $\delta := \delta_k$ for all $k \geq k_0$.

2) For later use, let us introduce the function

$$\psi_k(y, x) = \phi_k(y, x) + \tfrac{\delta}{2}\|y - x\|^2.$$

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

$$\delta\|y^{k+1}-x\|^2 \leq f(x)-\phi_k(y^{k+1},x).$$

Now remember that in step 6 of the algorithm, and according to axiom $(G3)$, we have kept the aggregate pair $(\alpha^*,g^*)\in\mathcal{G}_{k+1}$. By its definition (9), (10) we have

$$\phi_k(y^{k+1},x)=\alpha^*+g^{*\top}(y^{k+1}-x).$$

Defining a new function

$$\psi_k^*(y,x):=\alpha^*+g^{*\top}(y-x)+\tfrac{\delta}{2}\|y-x\|^2$$

we therefore have

$$\psi_k^*(y^{k+1},x)=\psi_k(y^{k+1},x)\quad\text{and}\quad \psi_k^*(y,x)\leq\psi_{k+1}(y,x), \tag{16}$$

the latter because $(\alpha^*,g^*)\in\mathcal{G}_{k+1}$, so that this pair contributes to the new models ϕ_{k+1}, ψ_{k+1}. Notice that ψ_k^* is a quadratic function. Expanding it at y^{k+1} gives

$$\psi_k^*(y,x)=\psi_k^*(y^{k+1},x)+\nabla\psi_k^*(y^{k+1},x)(y-y^{k+1})+\tfrac{\delta}{2}(y-y^{k+1})^\top(y-y^{k+1}),$$

where $\nabla\psi_k^*(y,x)=g^*+\delta(y-x)$ and $\nabla^2\psi_k^*(y,x)=\delta I$. We now prove the formula

$$\psi_k^*(y,x)=\psi_k^*(y^{k+1},x)+\tfrac{\delta}{2}\|y-y^{k+1}\|^2. \tag{17}$$

Indeed, we have but to show that the first-order term in the above Taylor expansion vanishes at y^{k+1}. But this term is

$$\begin{aligned}
&\nabla\psi_k^*(y^{k+1},x)^\top(y-y^{k+1})\\
&\quad= \left[g^*+\delta(y^{k+1}-x)\right]^\top(y-y^{k+1})\\
&\quad= g^{*\top}(y-y^{k+1})+\delta(y^{k+1}-x)^\top(y-y^{k+1})\\
&\quad= \delta(x-y^{k+1})^\top(y-y^{k+1})+\delta(y^{k+1}-x)^\top(y-y^{k+1}) \qquad \text{(using (9), (10))}\\
&\quad= 0,
\end{aligned}$$

and so formula (17) is established. Therefore

$$\begin{aligned}
\psi_k(y^{k+1},x) &\leq \psi_k^*(y^{k+1},x)+\tfrac{\delta}{2}\|y^{k+2}-y^{k+1}\|^2 && \text{(using (16) left)}\\
&= \psi_k^*(y^{k+2},x) && \text{(using (17))}\\
&\leq \psi_{k+1}(y^{k+2},x) && \text{(using (16) right)}\\
&\leq \psi_{k+1}(x,x) && (y^{k+2}\text{ is minimizer of }\psi_{k+1})\\
&= f(x).
\end{aligned}$$

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

$$\|y^{k+1}\|\leq\|x\|+\delta_1^{-1}\max_{\omega\in[0,\infty]}\|F'(x,\omega)^\star\|$$

by formula (13), we deduce using a geometric argument that

$$\|y^{k+2} - x\|^2 - \|y^{k+1} - x\|^2 \to 0. \tag{18}$$

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

$$\begin{aligned} &\phi_{k+1}(y^{k+2}, x) - \phi_k(y^{k+1}, x) \\ &= \psi_{k+1}(y^{k+2}, x) - \psi_k(y^{k+1}, x) - \tfrac{\delta}{2}\|y^{k+2} - x\|^2 + \tfrac{\delta}{2}\|y^{k+1} - x\|^2 \to 0, \end{aligned} \tag{19}$$

which converges to 0 due to convergence of $\psi_k(y^{k+1}, x)$ proved above, and property (18).

3) Let e_{k+1} be the normalized eigenvector associated with the maximum eigenvalue of $F(x, \omega_{k+1}) + F'(x, \omega_{k+1})(y^{k+1} - x)$, which we pick in step 5 of the algorithm. Then $g_k = F'(x, \omega_{k+1})^* e_{k+1} e_{k+1}^\top$ is a subgradient of $\phi_{k+1}(\cdot, x)$ at y^{k+1}. That means

$$g_k^\top (y - y^{k+1}) \leq \phi_{k+1}(y, x) - \phi_{k+1}(y^{k+1}, x).$$

Using $\phi_{k+1}(y^{k+1}, x) = \phi(y^{k+1}, x)$ from Lemma 3.3 therefore implies

$$\phi(y^{k+1}, x) + g_k^\top (y - y^{k+1}) \leq \phi_{k+1}(y, x). \tag{20}$$

Now observe that

$$\begin{aligned} 0 &\leq \phi(y^{k+1}, x) - \phi_k(y^{k+1}, x) \\ &= \phi(y^{k+1}, x) + g_k^\top (y^{k+2} - y^{k+1}) - \phi_k(y^{k+1}, x) - g_k^\top (y^{k+2} - y^{k+1}) \\ &\leq \phi_{k+1}(y^{k+2}, x) - \phi_k(y^{k+1}, x) + \|g_k\|\|y^{k+2} - y^{k+1}\| \qquad \text{(using (20))} \end{aligned}$$

and this term tends to 0 because of (19), boundedness of g_k, and because $y^{k+1} - y^{k+2} \to 0$. We conclude that

$$\phi(y^{k+1}, x) - \phi_k(y^{k+1}, x) \to 0. \tag{21}$$

4) We now show that $\phi_k(y^{k+1}, x) \to f(x)$, and therefore also $\phi(y^{k+1}, x) \to f(x)$. Suppose on the contrary that $\eta := f(x) - \limsup \phi_k(y^{k+1}, x) > 0$. Choose $0 < \theta < (1 - \widetilde{\gamma})\eta$. It follows from (21) that there exists $k_1 \geq k_0$ such that

$$\phi(y^{k+1}, x) - \theta \leq \phi_k(y^{k+1}, x)$$

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

$$\begin{aligned} \widetilde{\gamma}(\phi_k(y^{k+1}, x) - f(x)) &\leq \phi(y^{k+1}, x) - f(x) \\ &\leq \phi_k(y^{k+1}, x) + \theta - f(x). \end{aligned}$$

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

5) Having shown $\phi(y^{k+1}, x) \to f(x)$, we now argue that we must have $y^{k+1} \to x$. This follows from the definition of y^{k+1}, because

$$\psi_k(y^{k+1}, x) = \phi_k(y^{k+1}, x) + \tfrac{\delta}{2}\|y^{k+1} - x\|^2 \leq \psi_k(x, x) = f(x).$$

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

$$\begin{aligned} \delta(x-y^{k+1})^\top(y-y^{k+1}) &\leq \phi_k(y,x) \; - \phi_k(y^{k+1},x) \\ &\leq \phi(y,x) - \phi_k(y^{k+1},x) \end{aligned}$$

for every y. Passing to the limit implies

$$0 \leq \phi(y,x) - \phi(x,x),$$

because the left hand side converges to 0 in vue of $y^{k+1} \to x$. By convexity, $0 \in \partial\phi(x,x)$. Since $\partial\phi(x,x) = \partial f(x)$, we are done. □

9. Convergence of outer loop

All that remains to do now is piece things together and prove global convergence of our method. We have the following

Theorem 9.1. *Suppose $x^1 \in \mathbb{R}^n$ is such that $\{x \in \mathbb{R}^n : f(x) \leq f(x^1)\}$ is compact. Then every accumulation point of the sequence x^j of serious iterates generated by our algorithm is a critical point of f.*

Proof. Let x^j be the sequence of serious steps. We have to show that $0 \in \partial f(\bar{x})$ for every accumulation point $\bar{x}$ of x^j. Suppose at the jth stage of the outer loop the inner loop accepts a serious step at $k = k_j$. Then $x^{j+1} = y^{k_j+1}$. By the definition of y^{k+1} as minimizer of the tangent program (8) this means

$$\delta_{k_j}\left(x^j - x^{j+1}\right) \in \partial\phi_{k_j}(x^{j+1},x^j).$$

By convexity this can be re-written as

$$\delta_{k_j}\left(x^j - x^{j+1}\right)^\top\left(x^j - x^{j+1}\right) \leq \phi_{k_j}(x^j,x^j) - \phi_{k_j}(x^{j+1},x^j) = f(x^j) - \phi_{k_j}(x^{j+1},x^j),$$

the equality $\phi_{k_j}(x^j,x^j) = f(x^j)$ being true by Lemma 3.1. Since $x^{j+1} = y^{k_j+1}$ was accepted in step 4 of the algorithm, we have

$$f(x^j) - \phi_{k_j}(x^{j+1},x^j) \leq \gamma^{-1}\left(f(x^j) - f(x^{j+1})\right).$$

Altogether

$$\delta_{k_j}\|x^j - x^{j+1}\|^2 \leq \gamma^{-1}\left(f(x^j) - f(x^{j+1})\right).$$

Summing over $j = 1,\dots,J-1$ gives

$$\sum_{j=1}^{J-1}\delta_{k_j}\|x^j - x^{j+1}\|^2 \leq \gamma^{-1}\sum_{j=1}^{J-1} f(x^j) - f(x^{j+1}) = \gamma^{-1}\left(f(x^1) - f(x^J)\right).$$

By hypothesis, f is bounded below on the set of iterates, because the algorithm is of descent type on the serious steps. Since the $f(x^J)$ are bounded by hypothesis, this implies convergence of the series

$$\sum_{j=1}^{\infty}\delta_{k_j}\|x^j - x^{j+1}\|^2 < \infty.$$

In particular $\delta_{k_j}\|x^j - x^{j+1}\|^2 \to 0$. We now claim that $g_j = \delta_{k_j}(x^j - x^{j+1}) \to 0$.

Suppose on the contrary that there exists an infinite subsequence $j \in \mathcal{N}$ of $\mathbb{N}$ where $g_j = \delta_{k_j}\|x^j - x^{j+1}\| \geq \eta > 0$. Due to summability of $\delta_{k_j}\|x^j - x^{j+1}\|^2$ we must have $x^j - x^{j+1} \to 0$ in that case. That in turn is only possible when $\delta_{k_j} \to \infty$. We now construct another infinite subsequence $\mathcal{N}'$ of $\mathbb{N}$ such that $\delta_{k_j} \to \infty$, $j \in \mathcal{N}'$, and such that the doubling rule to increase δ_k in step 7 of the inner loop of the algorithm was applied at least once before $x^{j+1} = y^{k_j+1}$ was accepted. To construct $\mathcal{N}'$, we associate with every $j \in \mathcal{N}$ the last $j' \leq j$ where the δ-parameter was doubled while the inner loop was turning, and we let $\mathcal{N}'$ consists of all these j', $j \in \mathcal{N}$. It is possible that $j' = j$, but in general we can only assure that

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

so that $\mathcal{N}'$ is not necessarily a subset of $\mathcal{N}$. What counts is that $\mathcal{N}'$ is infinite, that $\delta_{k_j} \to \infty$, $(j \in \mathcal{N}')$, and that the doubling rule was applied for each $j \in \mathcal{N}'$.

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

$$\delta_{k_j} = \delta_{k_j-1} = \cdots = \delta_{k_j-\nu_j+1} = 2\delta_{k_j-\nu_j}. \tag{22}$$

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

$$\rho_{k_j-\nu_j} = \frac{f(x^j) - f(y^{k_j-\nu_j+1})}{f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)} < \gamma$$

and

$$\tilde{\rho}_{k_j-\nu_j} = \frac{f(x^j) - \phi(y^{k_j-\nu_j+1}, x^j)}{f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)} \geq \tilde{\gamma}.$$

By (22) we now have

$$\tfrac{1}{2}\delta_{k_j}\left(x^j - y^{k_j-\nu_j+1}\right) \in \partial\phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j).$$

Using $\phi_{k_j-\nu_j}(x^j, x^j) = f(x^j)$ and the subgradient inequality for $\phi_{k_j-\nu_j}(\cdot, x^j)$ at $y^{k_j-\nu_j+1}$ gives

$$\begin{array}{rcl} \tfrac{1}{2}\delta_{k_j}\left(x^j - y^{k_j-\nu_j+1}\right)^\top\left(x^j - y^{k_j-\nu_j+1}\right) & \leq & \phi_{k_j-\nu_j}(x^j, x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j) \\ & = & f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j). \end{array}$$

This could also be written as

$$\frac{\delta_{k_j}\|x^j - y^{k_j-\nu_j+1}\|^2}{f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)} \leq 2. \tag{23}$$

Substituting (23) into the expression $\tilde{\rho}_{k_j-\nu_j}$ gives

$$\begin{array}{rcll} \tilde{\rho}_{k_j-\nu_j} & = & \rho_{k_j-\nu_j} + \dfrac{f(y^{k_j-\nu_j+1}) - \phi(y^{k_j-\nu_j+1}, x^j)}{f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)} & \\ & \leq & \rho_{k_j-\nu_j} + \dfrac{L\|x^j - y^{k_j-\nu_j+1}\|^2}{f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)} & \text{(using Lemma 2.1)} \\ & \leq & \rho_{k_j-\nu_j} + \dfrac{2L}{\delta_{k_j}} & \text{(using (23)).} \end{array}$$

Since $\rho_{k_j-\nu_j} < \gamma$ and $L/2\delta_{k_j} \to 0$, $(j \in \mathcal{N}')$, we have $\limsup_{j\in\mathcal{N}'} \tilde{\rho}_{k_j-\nu_j} \leq \limsup_{j\in\mathcal{N}'} \rho_{k_j-\nu_j} \leq \gamma$, contradicting $\tilde{\rho}_{k_j-\nu_j} \geq \tilde{\gamma} > \gamma$ for all $j \in \mathcal{N}'$. This proves our claim $g_j \to 0$ as $j \to \infty$.

Let $\bar{x}$ be an accumulation point of the sequence x^j of serious iterates. We have to prove $0 \in \partial f(\bar{x})$. Pick a convergent subsequence $x^j \to \bar{x}$, $j \in \mathcal{N}$. Observe that the sequence x^{j+1} is also bounded, so passing to a subsequence of $\mathcal{N}$ if necessary, we may assume $x^{j+1} \to \tilde{x}$, $j \in \mathcal{N}$. In general it could happen that $\tilde{x} \neq \bar{x}$. Only when δ_{k_j}, $j \in \mathcal{N}$, are bounded away from 0 can we conclude that $x^{j+1} - x^j \to 0$.

Now as $g_j = \delta_{k_j}(x^j - x^{j+1})$ is a subgradient of $\phi_{k_j}(\cdot, x^j)$ at $y^{k_j+1} = x^{j+1}$ we have

$$\begin{aligned} g_j^\top h &\leq \phi_{k_j}(x^{j+1}+h, x^j) - \phi_{k_j}(x^{j+1}, x^j) \\ &\leq \phi(x^{j+1}+h, x^j) - \phi_{k_j}(x^{j+1}, x^j) \qquad \text{(using } \phi_{k_j} \leq \phi\text{)} \end{aligned}$$

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

$$\gamma^{-1}\left(f(x^j) - f(x^{j+1})\right) \geq f(x^j) - \phi_{k_j}(x^{j+1}, x^j).$$

Combining these two estimates gives

$$\begin{aligned} g_j^\top h &\leq \phi(x^{j+1}+h, x^j) - f(x^j) + f(x^j) - \phi_{k_j}(x^{j+1}, x^j) \\ &\leq \phi(x^{j+1}+h, x^j) - f(x^j) + \gamma^{-1}\left(f(x^j) - f(x^{j+1})\right). \end{aligned}$$

Passing to the limit (using $g_j \to 0$, $x^{j+1} \to \tilde{x}$, $x^j \to \bar{x}$, $f(\bar{x}) = \phi(\bar{x}, \bar{x})$, and $f(x^j) - f(x^{j+1}) \to 0$ in the order named) shows

$$0 \leq \phi(\tilde{x}+h, \bar{x}) - \phi(\bar{x}, \bar{x})$$

for every h. This being true for every h, we can fix h' and choose $h = \bar{x} - \tilde{x} + h'$, which then gives

$$0 \leq \phi(\bar{x}+h', \bar{x}) - \phi(\bar{x}, \bar{x}).$$

As this is true for every h', we have $0 \in \partial\phi(\cdot, \bar{x})(\bar{x})$, and hence also $0 \in \partial f(\bar{x})$. □

10. Numerical Experiments

H_∞ feedback controller synthesis was one of the motivating application for the developpement of the proximity control bundle algorithm presented in Section 7. We consider a linear time invariant dynamical system in the standard LFT form

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \cdot \begin{bmatrix} x \\ w \\ u \end{bmatrix},$$

where $x \in \mathbb{R}^{n_x}$ is the state, $y \in \mathbb{R}^{n_y}$ the output, $u \in \mathbb{R}^{n_u}$ the command input and $w \in \mathbb{R}^{n_w}, z \in \mathbb{R}^{n_z}$ the performance channel. To cancel direct transmission from input u to output y, the assumption $D_{22} = 0$ is made. This is no loss of generality (see [49], chapter 17).

Let K be a static feedback controller, then the closed loop state space data and transfer function $T(K,\cdot)$ read

$$\begin{cases} \dot{x} = A(K)x + B(K)w \\ z = C(K)x + D(K)w, \end{cases} \tag{24}$$

$$T(K,j\omega) = C(K)(j\omega I - A(K))^{-1}B(K) + D(K), \tag{25}$$

where

$$\begin{array}{ll} A(K) = A + B_2KC_2, & B(K) = B_1 + B_2KD_{21}, \\ C(K) = C_1 + D_{12}KC_2, & D(K) = D_{11} + D_{12}KD_{21}. \end{array} \tag{26}$$

Dynamic controllers can be addressed in the same way by prior augmention of the plant (26), see e.g. [5].

In H_∞ synthesis we compute K to minimize the H_∞ norm of the transfer function $T(K,\cdot)$, that is,

$$\|T(K,\cdot)\|_\infty := \sup_{\omega\in[0,\infty]} \sigma_1(T(K,j\omega)),$$

see e.g. [49]. The standard approach to H_∞ synthesis in the literature uses the Kalman-Yakubovitch-Popov Lemma and leads to a bilinear matrix inequality (BMI) [15]. Here we use a different and much more direct approach based on our proximity control algorithm. The advantage of this is that Lyapunov variables can be avoided, which is beneficial because they are a source of numerical trouble. Not only does their number grow quadratically with the system order, they may also cause strong disparity between the optimization variables [9]. The price to be paid for avoiding them is that a difficult semi-infinite and non-smooth program has to be solved. To synthesize a dynamic controller K of order $n_k \in \mathbb{N}$, $n_k \leq n_x$, the objective $f : \mathbb{R}^{(n_k+n_u)\times(n_k+n_y)} \to \mathbb{R}^+$ is defined as

$$f(K) := \max_{\omega\in[0,\infty]} \lambda_1(T(K,j\omega)^{\mathsf{H}}T(K,j\omega)) = \|T(K,\cdot)\|_\infty^2, \tag{27}$$

which is nonsmooth and non-convex with two sources of non-smoothness, the infinite max operator, and the maximum eigenvalue function.

10.1. Computing the objective

Computation of the function value and the subgradients in (27) presents the main difficulty. Fortunately this can be done in an efficient way using the bisection algorithm [14, 44, 49] based on Hamiltonian calculus. The objective f has the following nice property: either $f(K,\omega)$ has the same constant value for all $\omega \in [0,\infty]$, or the number of frequencies ω where the maximum is attained is finite, see [12, 13] and Lemma 2.2. In the sequel we call

$$\Omega(K) := \{\omega \in [0,\infty] : \sigma_1(T(K,j\omega)) = \|T(K,\cdot)\|_\infty\}$$

the set of active frequencies. Now consider the matrix transfer function G defined by

$$G(j\omega) := \begin{bmatrix} (j\omega I - A(K))^{-1}B(K) \\ I \end{bmatrix}^{\mathsf{H}} M \begin{bmatrix} (j\omega I - A(K))^{-1}B(K) \\ I \end{bmatrix}, \tag{28}$$

with $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$, and the associated Hamiltonian

$$\mathscr{H} := \begin{bmatrix} A(K) - B(K)M_{22}^{-1}M_{21} & -B(K)M_{22}^{-1}B^\top(K) \\ -M_{11} + M_{12}M_{22}^{-1}M_{21} & -A^\top(K) + M_{12}M_{22}^{-1}B^\top(K) \end{bmatrix}.$$

Theorem 10.1. *Assume that $A(K)$ has no imaginary eigenvalues and $M_{22} \prec 0$, then $G(j\omega)$ is singular iff $j\omega$ is an eigenvalue of $\mathscr{H}$.*

This result is the key element to compute the H_∞ norm by computing eigenvalues of $\mathscr{H}$ [44]. Let $\gamma \geq 0$ satisfy the inequality $\sigma_1(D) < \gamma \leq \|T(K,\cdot)\|_\infty$, and consider the matrix

$$M = \begin{bmatrix} C(K)^\top C(K) & C(K)^\top D(K) \\ D(K)^\top C(K) & D(K)^\top D(K) - \gamma^2 I \end{bmatrix}.$$

Then $G(j\omega) = T(K,j\omega)^{\mathsf{H}}T(K,j\omega) - \gamma^2 I$. Using Theorem 10.1, the frequencies $\omega \in [-\infty,\infty]$ satisfying

$$\sigma_1(T(K,j\omega)) = \gamma$$

can now be computed by finding the purely imaginary eigenvalues $j\omega$ of the Hamiltonian $\mathscr{H}$. The bisection algorithm of [14] to compute the H_∞ norm is based on this property. The set of active frequencies $\Omega(K)$, needed for the computation of the subgradients of f is also determined by this algorithm.

10.2. Convex model

A convex local model ϕ of f at the stability center $K \in \mathbb{R}^{(n_k+n_u)\times(n_k+n_y)}$ is defined as

$$\phi(K^+,K) = \max_{\omega\in[0,\infty]} \lambda_1(T_2(K,K^+,j\omega)),$$

where

$$\begin{aligned} T_2(K,Y,j\omega) \;:=\; & T(K,j\omega)^{\mathsf{H}}T(K,j\omega) + (T'(K,j\omega)(Y-K))^{\mathsf{H}}T(K,j\omega) \\ & +T(K,j\omega)^{\mathsf{H}}T'(K,j\omega)(Y-K), \end{aligned}$$

$T'(K,\cdot)$ being the derivative of transfer function (25) with respect to controller K. See [5] for a complete description of $T'(K,\cdot)$.

We need to explain how to compute $\phi(\cdot,K)$ at a given K^+. Notice that ϕ can not be written directly as an H_∞ norm of a transfer function, because $\lambda_1(T_2(K,K^+,j\omega))$ can be negative. In order to use the H_∞ norm computation algorithm, some additional work is needed.

Lemma 10.2. *Let $k \in \mathbb{N}$ and denote*

$$M_k = \begin{bmatrix} I_k & I_k \\ I_k & 0 \end{bmatrix} \in \mathbb{S}^{2k}.$$

Then $M_k = P_k^\top \Delta_k P_k$, where

$$\Delta_k = \begin{bmatrix} r_1 I_k & 0 \\ 0 & r_2 I_k \end{bmatrix}, \qquad P_k = \begin{bmatrix} \kappa_1 I_k & \kappa_2 I_k \\ -\kappa_2 I_k & \kappa_1 I_k \end{bmatrix},$$

with $r_1 = \frac{1-\sqrt{5}}{2}, r_2 = \frac{1+\sqrt{5}}{2}, \kappa_1 = \cos\alpha, \kappa_2 = \sin\alpha, \alpha = \arctan\left(\frac{1+\sqrt{5}}{2}\right)$.

Proof. Starting with matrix M_{k+1}, we apply the following sequence of row/column transpositions: row $k/2 \leftrightarrow k-1$, column $k/2 \leftrightarrow 1k-1$, row $k/2 \leftrightarrow k/2+1$, and finally column $k/2 \leftrightarrow k/2+1$. The matrix obtained is

$$\begin{bmatrix} M_k & 0 \\ 0 & M_1 \end{bmatrix}.$$

Repeating this process with the submatrices M_k, M_{k-1}, and so on, we finally obtain a block diagonal matrix, where each of the $k+1$ blocks equals M_1. Hence M_k has the two eigenvalues of M_1, each with multiplicity $k+1$.

Eigenvectors of M_k can now be obtained from those of M_1.

$$\begin{bmatrix} \kappa_1 I_k & \kappa_2 I_k \\ -\kappa_2 I_k & \kappa_1 I_k \end{bmatrix} \begin{bmatrix} I_k & I_k \\ I_k & 0 \end{bmatrix} \begin{bmatrix} \kappa_1 I_k & \kappa_2 I_k \\ -\kappa_2 I_k & \kappa_1 I_k \end{bmatrix}^\top = \begin{bmatrix} \kappa_1^2 + 2\kappa_1\kappa_2 I_k & \kappa_1^2 - \kappa_2^2 - 2\kappa_1\kappa_2 I_k \\ \kappa_1^2 - \kappa_2^2 - 2\kappa_1\kappa_2 I_k & \kappa_2^2 - 2\kappa_1\kappa_2 I_k \end{bmatrix}.$$

Using $\kappa_1^2 - \kappa_2^2 - 2\kappa_1\kappa_2 = 0$, and $\kappa_1 = \cos\alpha, \kappa_2 = \sin\alpha$, we deduce

$$1 - \tan^2\alpha - \tan\alpha = 0, \text{ hence } \tan\alpha \in \left\{ \frac{1-\sqrt{5}}{2}, \frac{1+\sqrt{5}}{2} \right\}.$$

Choosing $\alpha = \arctan\left(\frac{1+\sqrt{5}}{2}\right)$ gives the desired result, while the other case corresponds to a diagonal matrix with eigenvalues in decreasing order. □

Writing the transfer function $T_2(K, Y, \cdot)$ in the factorized form

$$T_2(K, Y, j\omega) = \begin{bmatrix} T(K, j\omega) \\ T'(K, j\omega)(Y - K) \end{bmatrix}^{\mathsf{H}} \begin{bmatrix} I & I \\ I & 0 \end{bmatrix} \begin{bmatrix} T(K, j\omega) \\ T'(K, j\omega)(Y - K) \end{bmatrix},$$

Lemma 10.2 leads to

$$T_2(K, Y, j\omega) = T_3(K, Y, j\omega)^{\mathsf{H}} \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} T_3(K, Y, j\omega), \tag{29}$$

where $T_3(K, Y, \cdot)$ is the transfer function defined by

$$T_3(K, Y, j\omega) = \begin{bmatrix} \sqrt{-r_1} I & 0 \\ 0 & \sqrt{r_2} I \end{bmatrix} P_k \begin{bmatrix} T(K, j\omega) \\ T'(K, j\omega)(Y - K) \end{bmatrix}.$$

We denote A_3, B_3, C_3, D_3 the state space data of transfer function $T_3(K, Y, \cdot)$ and let $\gamma \in \mathbb{R}$ such that

$$\lambda_1(D_3^\top \Sigma D_3) < \gamma \le \max_{\omega \in [0,\infty]} \lambda_1(T_2(K, Y, j\omega)).$$

Define

$$G_3(j\omega) = \begin{bmatrix} (j\omega I - A_3)^{-1} B_3 \\ I \end{bmatrix}^{\mathsf{H}} M_3 \begin{bmatrix} (j\omega I - A_3)^{-1} B_3 \\ I \end{bmatrix},$$

with

$$M_3 = \begin{bmatrix} C_3^\top \Sigma C_3 & C_3^\top \Sigma D_3 \\ D_3^\top \Sigma C_3 & D_3^\top \Sigma D_3 - \gamma I, \end{bmatrix}, \quad \text{and } \Sigma := \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix}.$$

Then with Theorem 10.1, the frequencies ω where $\lambda_1(T_3(K, Y, j\omega)) = \gamma$ can be computed from the eigenvalues of the associated Hamiltonian matrix. In consequence, the bisection algorithm of [14] can be generalized to compute the values of ϕ and also its subgradients.

Remark 10.3. The fact that ϕ cannot be directly expressed as an H_∞ norm appears more clearly in factorisation (29), due to multiplication with the non-positive matrix Σ.

10.3. Alternative convex model

The above construction shows that function values and subgradients of ϕ can be computed using a bisection method similar to the one used to compute the H_∞ norm. However, the transfer function $T_3(K, Y, \cdot)$ is formed by the parallel connection between $T(K, \cdot)$ and $T'(K, \cdot)$, so its number of states equals the sum of the number of states of $T(K, \cdot)$ and $T'(K, \cdot)$. Since $T(K, \cdot)$ has n_x states, $T'(K, \cdot)$ has $2 \times n_x$ states, being the serial connection of two transfer functions with n_x states [5]. Hence, $T_3(K, Y, \cdot)$ has $3 \times n_x$ states, the associated Hamiltonian matrix is then 3 times larger than that of the transfer function $T(K, \cdot)$.

For a $n \times n$ matrix, the number of floating point operation to compute eigenvalues is $O(n^3)$, as described in the LAPACK benchmark [1]. To compute ϕ, the computational cost for eigenvalue identification of the Hamiltonian matrix is then 27 times the cost of the computation for the transfer function $T(K, \cdot)$. In other words, $\phi(Y, K)$ is 27 times more expensive than $f(Y)$, because the cost for the factorization above is approximately the same as the cost of the factorization needed to compute f. The use of ϕ is therefore convenient only for small to medium order systems. When the system order is large, another convex model with lower computational cost has to be used.

A natural idea is to use a simplified version of ϕ by performing the frequency maximization over an adequatly chosen subset of frequencies. We explain in which way this can be arranged, so that the arguments in the proof of Theorem 9.1 remain valid.

Consider the model

$$\widetilde{\phi}(Y, K) := \max_{\omega \in \Omega(K) \cup \Omega(Y)} \lambda_1(T_2(K, Y, j\omega)),$$

where $\Omega(K)$ is the set of active frequencies of f at K, $\Omega(Y)$ active frequencies of f at Y. At least in the case where $\Omega(K) \cup \Omega(Y)$ is finite, $\widetilde{\phi}$ can be computed efficiently. Unfortunately, $\widetilde{\phi}$ is not suited as a model for f, because it lacks continuity used at the very end of the proof of Theorem 9.1. Namely, when $K_i \to K$ and $Y_i \to Y$, it may happen that $\limsup_{i\to\infty} \Omega(K_i) \cup \Omega(Y_i) \not\supset \Omega(K) \cup \Omega(Y)$, because it is well-known that the number of active peaks and also the eigenvalue multiplicity at each peak may increase brusquely as we pass to the limit.

Assuming in the following that all $\Omega(K)$, $\Omega(Y)$ encountered by our algorithm are finite, we can arrange a different but still practical way to define a model, which has the desired semi-continuity property. Let us use the following notation. For any set Ω with $\Omega(K) \cup \Omega(Y) \subset \Omega \subset [0, \infty]$ define

$$\phi_\Omega(Y, K) = \sup_{\omega \in \Omega} \lambda_1 \left(T_2(K, Y, j\omega) \right).$$

Then $\phi = \phi_{[0,\infty]}$ and $\widetilde{\phi} = \phi_{\Omega(K) \cup \Omega(Y)}$. The first Ω is too large (CPU), the second too small (lack of continuity). We need something intermediate, which we call $\Omega(K, Y)$, and which will have a weak form of continuity. We will then put

$$\bar{\phi}(Y, K) := \phi_{\Omega(Y,K)}(Y, K) = \sup_{\omega \in \Omega(Y,K)} \lambda_1 \left(T_2(K, Y, j\omega) \right).$$

Then $\bar{\phi}(\cdot,K) = \phi_{\Omega(Y,K)}(\cdot,K) \geq \phi_{\Omega(K)\cup\Omega(Y)}(\cdot,K) = \widetilde{\phi}(\cdot,K)$, and therefore $\bar{\phi}(K,K) = \widetilde{\phi}(K,K) = f(K)$. What we need to ascertain when defining $\Omega(Y,K)$ is that $\Omega(K) \cup \Omega(Y) \subset \limsup_{i\to\infty} \Omega(Y_i,K_i)$ for $K_i \to K$, $Y_i \to Y$. Then the argument at the end of the proof of Theorem 9.1 remains valid.

This can be arranged in the following way. Let $f(K) > 0$ and, fixing $0 < \theta < 1$, choose a tolerance level $\theta f(K) < f(K)$. Now let $\Omega_e(K)$ be an extended set of frequencies which contains the peaks $\omega \in \Omega(K)$, but also an additional sample of frequencies ω in the range $\theta f(K) \leq \lambda_1(K,K,j\omega)) \leq f(K)$. $\Omega_e(K)$ could be a gridding of the set $\{\omega : \theta f(K) \leq \lambda_1(K,K,j\omega)) \leq f(K)\}$, which is a finite union of open intervals. The gridding should be arranged to depend continuously on K, and such that it contains all local maxima of the curve $\omega \mapsto \lambda_1(T_2(K,Y,j\omega))$ in that frequency range. The idea is that as $K_i \to K$, some of these *secondary peaks* $\Omega_e(K_i)$ will become peaks at K, so that $\Omega(K) \subset \limsup_{i\to\infty} \Omega_e(K_i)$. Put differently, knowledge of the secondary peaks in the band $\theta f(K_i) < f(K_i)$ for i sufficiently large allows to anticipate the peaks at K. Now use a similar construction for Y letting $\Omega_e(Y)$ be a gridding of the set $\{\omega : \theta \sup_{\omega\in\Omega(Y)} \lambda_1(T_2(K,Y,j\omega)) \leq \lambda_1(T_2(K,Y,j\omega)) \leq \sup_{\omega\in\Omega(Y)} \lambda_1(T_2(K,Y,j\omega))\}$. Finally put $\Omega(Y,K) = \Omega_e(Y) \cup \Omega_e(K)$. Notice that ways to estimate secondary peaks have been discussed in [12]. Numerical experience shows that these secondary peaks need not be computed with a very high accuracy. It suffices that the accuracy increases as these local maxima get closer to the global maximum of the frequency plot, and this is usually easy to arrange.

In order to guarantee convergence of our algorithmic scheme when $\bar{\phi}$ is used instead of ϕ, we need to ascertain that the estimate of Lemma 2.1 remains valid. But this is guaranteed at all trial points Y visited during the iteration simply by having $\Omega(Y) \subset \Omega(K,Y)$. The uniformity of the constant L follows from Weyl's theorem used in the proof of Lemma 2.1.

10.4. Implementation and initialisation

We have implemented the proximity control bundle algorithm (PC) for H_∞ output feedback controller synthesis in Matlab. Both the ideal model ϕ and its approximation $\bar{\phi} = \phi_{\Omega(Y,K)}$ have been used to compare performance. For comparison we have included two software tools for H_∞ synthesis. The first is the linesearch method (LS) described in [5], where descent direction are derived from enhanced subgradient information. The second one is `HIFOO` from [16], based on the gradient sampling method of [17].

The same stopping criteria have been used for LS and PC. The algorithm is stopped if descent of objective and steplength are too small, i.e.

$$f(K) - f(K^+) < \varepsilon(|f(K)| + 1) \quad \text{and} \quad \left\|K - K^+\right\| < \varepsilon(\|K\| + 1),$$

where $\varepsilon > 0$ is a tolerance parameter, fixed to $1e-5$ in all numerical tests. Stopping in `HIFOO` is rather different. We have therefore fixed the numerical tolerance for its stopping criterion to the same value ε.

LS allows to compute a criticality measure θ, deduced from its tangent program [2]. This criticality measure has been used *a posteriori* to measure criticality of synthesized controllers in the static synthesis case.

There is also another stopping criterion for PC used to avoid entering the inner loop at near optimal points K, where it may perform a large number of trial steps, only to end with a serious step with negligeable progress. This stopping criterion is based on the tangent program. The algorithm is halted if

$$f(K) - \bar{\phi}(K^+, K) < \varepsilon_2,$$

where number ε_2 is chosen small to stop only when the algorithm is stuck in null steps. In our test we have set $\varepsilon_2 = 0.01 \times \varepsilon$.

plant	n_x	n_z	n_w	n_y	n_u	$\bar{\gamma}_\infty$
AC2	5	5	3	3	3	0.111495
HE4	8	12	8	6	4	22.838570
AC18	10	5	3	2	2	5.394531
HE6	20	16	6	6	4	2.388637
HE7	20	16	9	6	4	2.611759
JE1	30	8	30	5	3	3.882812
AC14	40	11	4	4	3	100
BDT2	82	4	2	4	4	0.234014

Table 10.1: State dimensions of the models used in numerical experiments. Performance $\bar{\gamma}_\infty$ of the full-order H_∞ controller is shown on the right and gives a lower bound for the tests in Tables 10.2 and 10.3.

For H_∞ synthesis models from the COMPL$_e$iB library [36] have been used: four aircraft models (AC2, AC10, AC14, and AC18), three helicopter models (HE4, HE6, and HE7), one jet engine mode (JE1), and one distillation column model (BDT2). State dimensions of these plants are given in Table 10.1 and range from small to large. The optimal full-order H_∞ performance $\bar{\gamma}_\infty$ has been computed for each plant using the MATLAB `hinfric` solver to give a strict lower bound for the locally optimal gains computed by the algorithms.

For static H_∞ synthesis the four methods have been compared. To allow a fair comparison, the same initial stabilizing controller has been used to start each algorithms. Notice however that `HIFOO` uses a random multistart strategy, so in each run three new initial points are generated from random perturbations of K_0. Then the best result of these four runs is chosen. In contrast, LS and PC perform only one run not using any random perturbations. For dynamic synthesis we only compare the proximal bundle method with model $\bar{\phi}$ to `HIFOO`.

Finding an initial stabilizing controller is sometimes intricate. Here we use software based on spectral abscissa optimization developped in [10].

10.5. Results

Results for static H_∞ synthesis are displayed in Table 10.2. The final performance γ_∞, CPU time T, and criticality measure $|\theta|$ are given for each of the four methods. For LS and PC the number of outer steps `it` and mean time by iteration `CPU` are given. `CPU_in` and `it_in` are respectively the mean inner iteration, CPU time for PC algorithm, and total number of inner iterations. As can be seen on this Table, the use of model ϕ is

plant, method	γ_∞	it	it_in	CPU	CPU_in	T	$\lvert\theta\rvert$
HE4, LS	34.25801	46	-	1.53e-2	-	0.70	8.8e-3
HE4, PC-$\bar{\phi}$	23.58456	154	1078	1.02e-2	2.20e-2	25.28	5.7e-3
HE4, PC-ϕ	23.02933	128	1297	1.43e-2	4.94e-2	65.93	1.9e-3
HE4, HIFOO	22.83907	-	-	-	-	36.67	3.7e-3
HE6, LS	462.52976	289	-	1.91e-2	-	5.52	4.3e-2
HE6, PC-$\bar{\phi}$	192.35793	548	692	1.14e-2	2.09e-2	20.02	1.2
HE6, PC-ϕ	192.35718	599	1165	1.48e-2	8.39e-2	106.71	2.7e-4
HE6, HIFOO	192.35881	-	-	-	-	128.78	1.1e-3
AC2, LS	0.11149	57	-	1.24e-2	-	0.71	7.2e-7
AC2, PC-$\bar{\phi}$	0.11149	147	43	6.20e-3	1.16e-2	1.41	1.3e-4
AC2, PC-ϕ	0.11149	143	43	6.53e-3	4.48e-2	2.86	1.4e-4
AC2, HIFOO	0.11149	-	-	-	-	0.74	1.7e-6
AC14, LS	104.93211	51	-	5.10e-2	-	2.60	6.3e-4
AC14, PC-$\bar{\phi}$	102.64676	73	325	2.84e-2	4.12e-2	15.45	1.9e-3
AC14, PC-ϕ	102.55853	72	227	2.68e-2	2.73e-1	64.00	5.7e-4
AC14, HIFOO	106.36521	-	-	-	-	291.92	2.6
AC18, LS	10.71487	30	-	1.90e-2	-	0.57	6.2-02
AC18, PC-$\bar{\phi}$	10.70115	68	296	1.00e-2	1.84e-2	6.11	3.8e-2
AC18, PC-ϕ	10.71141	45	224	1.07e-2	5.41e-2	12.58	3.7e-2
AC18, HIFOO	27.23054	-	-	-	-	13.32	3.8e-4
BDT2, LS	0.82903	70	-	2.72e-1	-	19.04	8.8e-5
BDT2, PC-$\bar{\phi}$	0.67307	811	1393	8.93e-2	1.01e-1	214.18	7.5e-5
BDT2, PC-ϕ	0.67301	769	1361	8.98e-2	1.73	2430.00	8.9e-5
BDT2, HIFOO	0.82050	-	-	-	-	1154.26	1.1e-3

Table 10.2: Static H_∞ synthesis. Four methods have been compared on 6 plants. The columns show final objective value γ_∞, number of iterations it, number of inner iterations it_in, mean iteration time CPU, mean inner iteration time CPU_in, total synthesis time T, and criticality measure $|\theta|$. All cpu's are in seconds.

much more costly to compute than $\bar{\phi} = \phi_{\Omega(Y,K)}$. For small sized system this difference is slight, but becomes important for larger plants like BDT2.

Results of PC bundle method with model ϕ and $\bar{\phi}$ are very close except for the HE4, where the use of the true model ϕ leads to a better result. It seems that PC needs fewer outer and inner iterations often with better quality if the model ϕ is used. This is not the case for HE6 and can be explained by the fact that here the algorithm gets stuck at some ill-conditioned point, where subgradient information is not trustworthy, leading to many inner iterations with only a small progress in f. In general, the global computation time for PC with model ϕ is much larger than with the approximate model $\bar{\phi}$, while it is still faster than HIFOO for half of the experiments.

Controller synthesis with LS is very fast and gives good results on AC2 and AC18. However on most examples LS stops earlier than for the other methods, leading to controllers with slightly worse H_∞ performance. This behavior can be explained by the fact that LS is not suited to handle situation where the first singular values of transfer function coalesce. This is explained in [5], where for numerical simplifications the authors

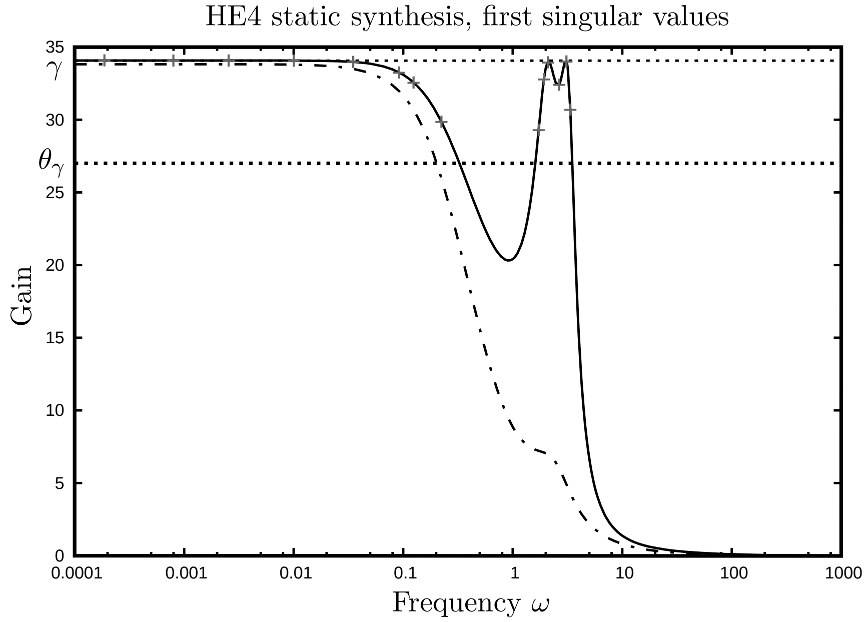

Figure 10.1: HE4 singular value plot of LS method synthesized controller. Selected frequencies $\omega \in \Omega_e(K)$ above the threshold $\theta_\gamma < \gamma$ are shown.

made the assumption that the maximal singular value is simple. As can be seen in Figure 10.1, this assumption is not true for the controller synthesized with LS. Coalescence of singular values also occurs for HE6, AC14 and AC18, indicating that it may not be a rare phenomenon.

HIFOO performed similar to PC on HE4, HE6 and AC2, while in all other examples the best results were obtained with PC. Moreover, HIFOO was much slower than PC-$\bar{\phi}$ for all examples, except for AC2.

Table 10.3 shows results for dynamic controller synthesis of order $0 < n_k \leq n_x$. Performance and CPU were compared between PC-$\bar{\phi}$ and HIFOO for each model. Results are encouraging, PC-$\bar{\phi}$ is fast and computes controllers which outperform HIFOO for most examples. Only BDT2 caused trouble, as the PC algorithm got stuck in the inner loop.

plant	n_k	γ_∞ PC-$\bar{\phi}$	γ_∞ HIFOO	T PC-$\bar{\phi}$	T HIFOO
HE6	1	187.42768	187.42745	22.25	102.50
HE6	2	16.44215	19.93352	46.50	168.80
HE6	3	10.03032	10.07088	106.61	175.25
HE7	6	2.87754	10.21527	106.90	403.21
AC10	2	7.63791	10.49584	1165.19	215.59
JE1	5	5.67469	33.67678	252.06	591.40
BDT2	6	5.08779	4.97796	3732.23	3150.41

Table 10.3: Dynamic H_∞ synthesis. For each of the 7 models, PC-$\bar{\phi}$ and HIFOO are compared. γ_∞ is the objective value reached by the method, T the total synthesis CPU in seconds.

Conclusion

We have presented a proximity control bundle algorithm to optimize the H_∞-norm or other nonsmooth criteria which are infinite maxima of maximum eigenvalue functions [4, 8]. Global convergence of the algorithm was proved. The method was tested on examples in feedback control design and shown to have good performance compared to the linesearch method of [5] and `HIFOO` [16].

Acknowledgements. The authors acknowledge financial support from Agence Nationale de Recherche (ANR) under contract `SSIA_NV_6 Controvert`, from Agence Nationale de Recherche (ANR) under contract `NT05 − 1_43040 Guidage`, and from *Fondation de Recherches pour l'Aéronautique et l'Espace* under contract `Survol`.

References

[1] E. Anderson et al.: LAPACK User's Guide, SIAM, Philadelphia (1992).

[2] P. Apkarian, D. Noll: Nonsmooth optimization for multidisk H_∞ synthesis, Eur. J. Control 12(3) (2006) 229–244.

[3] P. Apkarian, D. Noll: Controller design via nonsmooth multi-directional search, SIAM J. Control Optimization 44 (2006) 1923–1949.

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

[5] P. Apkarian, D. Noll: Nonsmooth H_∞ synthesis, IEEE Trans. Autom. Control 51 (2006) 71–86.

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

[7] P. Apkarian, V. Bompart, D. Noll: Nonsmooth structured control design with applications to PID loopshaping of a process, Int. J. Robust Nonlinear Control 17(14) (2007) 1320–1342.

[8] P. Apkarian, D. Noll, O. Prot: Nonsmooth methods for analysis and synthesis with integral quadratic constraints, in: Conference Proceedings, ACC (New Orleans, 2007).

[9] P. Apkarian, D. Noll, O. Prot: Trust region spectral bundle method for nonconvex maximum eigenvalue functions, SIAM J. Optim. 19(1) (2008) 281–306.

[10] V. Bompart: Optimisation non Lisse pour la Commande des Systèmes de l'Aéronautique, Thèse de l'Université Paul Sabatier, Toulouse (November 2007).

[11] V. Bompart, P. Apkarian, D. Noll: Nonsmooth techniques for stabilizing linear systems, in: Proceedings of the American Control Conference (New York, 2007).

[12] V. Bompart, D. Noll, P. Apkarian: Second-order nonsmooth optimization of the H_∞ norm, Numer. Math. 107(3) (2007) 433–454.

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

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

[15] S. Boyd, L. Elghaoui, E. Feron, V. Balakrishnan: Linear Matrix Inequalities in System and Control Theory, SIAM Studies in Applied Mathematics 15, SIAM, Philadelphia (1994).

[16] J. V. Burke, D. Henrion, A. S. Lewis, M. L. Overton: HIFOO - A MATLAB package for fixed-order controller design and H-infinity optimization, in: Proceedings of ROCOND 2006 (Toulouse, July 2006).

[17] J. V. Burke, A. S. Lewis, M. L. Overton: A robust gradient sampling algorithm for nonsmooth, nonconvex optimization, SIAM J. Optim. 15 (2005) 751–779.

[18] B. M. Chen: H_∞ Control and its Applications, Lecture Notes in Control and Information Sciences 235, Springer, New York (1998).

[19] F. Clarke: Optimization and Nonsmooth Analysis, Canadian Math. Society Series, John Wiley & Sons, New York (1983).

[20] J. Cullum, W. Donath, P. Wolfe: The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices, in: Nondiffer. Optim., Math. Program. Study 3, M. L. Balinski, P. Wolfe (eds.), North Holland, Amsterdam (1975) 35–55.

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

[22] R. Fletcher: Semidefinite matrix constraints in optimization, SIAM J. Control Optimization 23 (1985) 493–513.

[23] A. Fuduli, M. Gaudioso, G. Giallombardo: A DC piecewise affine model and a bundling technique in nonconvex nonsmooth optimization, Optim. Methods Softw. 19 (2004) 89–102.

[24] A. Fuduli, M. Gaudioso, G. Giallombardo: Minimizing nonconvex nonsmooth functions via cutting planes and proximity control, SIAM J. Optim. 14 (2004) 743–756.

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

[26] C. Helmberg, K. C. Kiwiel: A spectral bundle method with bounds, Math. Program., Ser. A 93 (2002) 173–194.

[27] C. Helmberg, F. Oustry: Bundle methods to minimize the maximum eigenvalue function, in: Handbook of Semidefinite Programming. Theory, Algorithms and Applications, H. Wolkowicz et al. (ed.), Int. Ser. Oper. Res. Manag. Sci. 27, Kluwer, Dordrecht (2000) 307–337.

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

[29] J. W. Helton, O. Merino: Coordinate optimization for bi-convex matrix inequalities, in: Proc. IEEE Conf. Decision and Control (San Diego, 1997) 3609–3613.

[30] J.-B. Hiriart-Urruty, C. Lemaréchal: Convex Analysis and Minimization Algorithms. Part 1: Fundamentals; Part II: Advanced Theory and Bundle Methods, Grundlehren der mathematischen Wissenschaften 305, 306, Springer, Berlin (1993).

[31] Y. S. Hung, A. G. J. MacFarlane: Multivariable Feedback: A Classical Approach, Lect. Notes in Control and Information Sciences 40, Springer, Berlin (1982).

[32] L. H. Keel, S. P. Bhattacharyya, J. W. Howe: Robust control with structured perturbations, IEEE Trans Autom. Control 36 (1988) 68–77.

[33] K. C. Kiwiel: Proximity control in bundle methods for convex nondifferentiable minimization, Math. Programming, Ser. A 46 (1990) 105–122.

[34] K. C. Kiwiel: Methods of Descent for Nondifferentiable Optimization, Lect. Notes in Math. 1133, Springer, Berlin (1985).

[35] K. C. Kiwiel: A linearization algorithm for computing control systems subject to singular value inequalities, IEEE Trans. Autom. Control 31 (1986) 595–602.

[36] F. Leibfritz: $COMPL_eIB$, COnstrained Matrix-optimization Problem LIbrary - a collection of test examples for nonlinear semidefinite programs, control system design and related problems, Tech. Report, Universität Trier, Trier (2003).

[37] C. Lemaréchal: An extension of Davidson method to nondifferentiable problems, in: Nondiffer. Optim., Math. Program. Study 3, M. L. Balinski, P. Wolfe (eds.), North Holland, Amsterdam (1975) 95–109.

[38] C. Lemaréchal: Bundle methods in nonsmooth optimization, in: Nonsmooth Optimization, Proc. IIASA Workshop 1977, C. Lemaréchal, R. Mifflin (eds.), IIASA Proc. Ser. 3, Pergamon Press, Oxford (1978) 79–102.

[39] C. Lemaréchal: Nondifferentiable optimization, in: Handbooks in Operations Research and Management Sciences 1, G. L. Nemhauser et al. (ed.), North-Holland, Amsterdam (1989) 529–572.

[40] C. Lemaréchal, A. Nemirovskii, Y. Nesterov: New variants of bundle methods, Math. Program., Ser. B 69 (1995) 111–147.

[41] D. Mayne, E. Polak: Algorithms for the design of control systems subject to singular value inequalities, Math. Program. Study 18 (1982) 112–134.

[42] D. Mayne, E. Polak, A. Sangiovanni: Computer aided design via optimization, Automatica 18(2) (1982) 147–154.

[43] D. Noll, P. Apkarian: Spectral bundle method for nonconvex maximum eigenvalue functions: first-order methods, Math. Program., Ser. B 104 (2005) 701–727.

[44] P. A. Parrilo: On the numerical solution of LMIs derived from the KYP lemma. in: Proc. IEEE Conf. Decision and Control 3 (Phoenix, Arizona, 1999) 2334–2338.

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

[46] E. Polak: Optimization: Algorithms and Consistent Approximations, Applied Mathematical Sciences 124, Springer, New York (1997).

[47] J.-B. Thevenet, D. Noll, P. Apkarian: Nonlinear spectral SDP method for BMI-constrained problems: applications to control design, in: Informatics in Control, Automation and Robotics I (Setúbal, 2004), J. Braz et al. (ed.), Springer, Berlin (2006) 61–72.

[48] P. Wolfe: A method of conjugate subgradients for minimizing nondifferentiable functions, in: Nondiffer. Optim., Math. Program. Study, M. L. Balinski, P. Wolfe (eds.), North Holland, Amsterdam (1975) 145–173.

[49] K. Zhou, J. C. Doyle, K. Glover: Robust and Optimal Control, Prentice Hall, Upper Saddle River (1996).