Minimizing memory effects of a system

Minh Ngoc Dao, Dominikus Noll

Math. Control Signals Syst. manuscript No.
(will be inserted by the editor)

Minimizing memory effects of a system

Minh Ngoc Dao · Dominikus Noll

Received: 20 December 2013 / Accepted: 15 June 2014

Abstract Given a stable linear time-invariant system with tunable parameters, we present a method to tune these parameters in such a way that undesirable responses of the system to past excitations, known as system ringing, are avoided or reduced. This problem is addressed by minimizing the Hankel norm of the system, which quantifies the influence of past inputs on future outputs. We indicate by way of examples that minimizing the Hankel norm has a wide scope for possible applications. We show that the Hankel norm minimization program may be cast as an eigenvalue optimization problem, which we solve by a nonsmooth bundle algorithm with a local convergence certificate. Numerical experiments are used to demonstrate the efficiency of our approach.

Keywords System ringing · system memory · Hankel norm · system reduction · controller design · system with tunable parameters

1 Introduction

Ringing generally designates undesired responses of a system to past excitations. In electronic systems, ringing arises under various forms of noise, such as gate ringing in converters, undesired oscillations in digital controllers, or input ring back in clock signals. In mechanical systems, ringing effects, when combined with resonance, may accelerate breakdown. In audio systems, ringing may cause echoes to occur before transients.

M. N. Dao
Department of Mathematics and Informatics,
Hanoi National University of Education, Vietnam
and Institut de Mathématiques, Université de Toulouse, France
E-mail: minhdn@hnue.edu.vn

D. Noll
Institut de Mathématiques, Université de Toulouse, France
E-mail: noll@mip.ups-tlse.fr

In more abstract terms, ringing may be understood as a tendency of the system to store energy, which is retrieved later to produce undesired effects. One way to quantify this capacity uses the Hankel norm of a system, which measures the effects of past inputs on future outputs.

This paper focuses on the problem of minimizing system ringing by casting it as a Hankel norm minimization program. This leads to an eigenvalue optimization problem, for which we propose a nonsmooth bundle algorithm which assures convergence to a critical point from an arbitrary starting point. We demonstrate that a variety of problems such as Hankel synthesis, maximizing the memory of a system, and control of flow in a graph, can be interpreted as Hankel norm minimization programs and solved efficiently using the proposed algorithm.

There is a considerable body of literature dedicated to Hankel norm system reduction, the original contribution being [12]. Our present approach is complementary to this classical line, as we focus on Hankel norm optimization problems which cannot be solved by linear algebra techniques. This makes our method closer in spirit to H_2- or H_∞-controller or filter design [26].

The structure of the paper is as follows. After presenting the problem in abstract form in Sect. 2, we show in Sect. 3 how it can be cast as a nonconvex eigenvalue optimization program. Section 4 describes how Clarke subgradients of a Hankel norm objective can be computed. In Sect. 5 we extend the Hankel norm to systems with direct transmission in a physically meaningful way. Sections 6, 7 present typical applications for the purpose of motivation of the Hankel minimization problem. Section 8 discusses a proximal bundle algorithm used to solve the Hankel norm minimization program. We propose a smooth relaxation of the Hankel norm in Sect. 9. Experiments with typical applications are given in Sect. 10.

Notation

Terminology in nonsmooth optimization is covered by [8], system theory by [26]. Following the latter reference, given a transfer matrix function $G(s) = C(sI - A)^{-1}B + D$, we use the standard notations

$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \text{ or } G = (A, B, C, D)$$

to indicate that

$$G : \begin{cases} \dot{x} = Ax + Bw \\ z = Cx + Dw \end{cases}$$

is a state-space realization of $z(s) = G(s)w(s)$. Similar notations apply to discrete time systems.

We shall work in the set of rectangular matrices with the corresponding scalar product $\langle M, N\rangle = \mathrm{Tr}(M^\top N) = \mathrm{Tr}(N^\top M)$, where $M^\top$ and $\mathrm{Tr}(M)$ are transpose and trace of a matrix. For symmetric matrices, $M \succ 0$ means positive definite, $M \succeq 0$ positive semidefinite.

2 Hankel norm minimization

Consider a linear time-invariant system

$$G : \begin{cases} \dot{x} = Ax + Bw \\ z = Cx \end{cases}$$

with state $x \in \mathbb{R}^{n_x}$, input $w \in \mathbb{R}^m$, and output $z \in \mathbb{R}^p$. Suppose G is internally stable in the sense that all eigenvalues of A have negative real part. If we think of $w(t)$ as an excitation at the input which acts over the time period $0 \leqslant t \leqslant T$ with dynamics started at $x(0) = 0$, then the ring of the system after the excitation has stopped at time T is $z(t)$ for $t > T$. If signals are measured in the energy norm, this leads to the definition of the Hankel norm of an internally stable system $G = (A, B, C)$ with input w and output $z = Gw$ as

$$\|G\|_H = \sup_{T>0} \left\{ \left(\int_T^\infty z^\top z \, \mathrm{d}t \right)^{1/2} : \int_0^T w^\top w \, \mathrm{d}t \leqslant 1, w(t) = 0 \text{ for } t > T \right\}.$$

For the discrete time case, the Hankel norm of an internally stable system

$$G : \begin{cases} x(t+1) = Ax(t) + Bw(t) \\ z(t) \qquad = Cx(t) \end{cases}$$

is given by

$$\|G\|_H = \sup_{T>0} \left\{ \left(\sum_{t=T}^\infty z(t)^\top z(t) \right)^{1/2} : \sum_{t=0}^T w(t)^\top w(t) \leqslant 1, w(t) = 0 \text{ for } t > T \right\},$$

where now internally stable means that all eigenvalues of A have magnitude < 1, and where it is again understood that $z = Gw$. A formula which works in both cases is

$$\|G\|_H = \sup_{T>0} \left\{ \|z\|_{2,[T,\infty)} : \|w\|_{2,[0,T]} \leqslant 1, w \in L^2[0,T], w(t) = 0, t > T \right\}. \quad (1)$$

Note that the system G in the above definition has no direct transmission D. This accounts for the fact, proved in Lemma 2 in Sect. 5, that D causes no memory effects, and is therefore not seen by the Hankel norm (1). In consequence, on the space of systems $G = (A, B, C, D)$ with direct transmission, $\|\cdot\|_H$ is only a semi-norm and not a norm.

By definition, the Hankel norm can be interpreted as a measure of the effects of past inputs, that is, the memory of the system, on the states and future outputs. Here, we are interested in systems $G(\mathbf{x})$ with tunable parameters $\mathbf{x} \in \mathbb{R}^n$, where the matrices $A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x})$ depend smoothly on a design parameter $\mathbf{x}$ varying in $\mathbb{R}^n$ or in some constrained subset of $\mathbb{R}^n$. Our goal is to tune $\mathbf{x}$ such that system ringing is avoided or reduced while internal

stability of the system is guaranteed. This leads to the following Hankel norm minimization program

$$\begin{array}{ll} \text{minimize} & \|G(\mathbf{x})\|_H \\ \text{subject to} & G(\mathbf{x}) \text{ internally stable} \\ & \mathbf{x} \in \mathbb{R}^n. \end{array} \tag{2}$$

We will discuss various instances, where program (2) may be of interest. Then, we present a nonsmooth optimization method based on techniques from eigenvalue optimization to solve (2), and discuss a smooth relaxation motivated by a result of Nesterov in [15].

3 Representation of the Hankel norm

A representation of the Hankel norm $\|\cdot\|_H$ amenable to computations is obtained through the observability and controllability Gramians, defined in [26, Section 3.8]. Based on the results in [12, Section 2.3], see also [26, Theorem 8.1], we have the following

Lemma 1 *Let $G = (A, B, C)$ be an internally stable linear time-invariant system with input w and output z, and let $\Gamma_G : L^2(-\infty, 0] \longrightarrow L^2[0, \infty)$ be the Hankel operator associated with G, defined by*

$$(\Gamma_G w)(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bw(\tau)\mathrm{d}\tau,\ t \geqslant 0.$$

Then, the following definitions are equivalent:

(i) $\|G\|_H = \sup_{T>0} \left\{ \|z\|_{2,[T,\infty)} : \|w\|_{2,[0,T]} \leqslant 1, w \in L^2[0,T], w(t) = 0, t > T \right\}$.

(ii) $\|G\|_H = \|\Gamma_G\| = \sup \left\{ \|\Gamma_G w\|_{2,[0,\infty)} : \|w\|_{2,(-\infty,0]} \leqslant 1, w \in L^2(-\infty, 0] \right\}$.

(iii) $\|G\|_H = \sqrt{\lambda_1(XY)}$, *where λ_1 denotes the maximum eigenvalue of a matrix, and X, Y are the controllability and observability Gramians of the system.*

Proof We assume $x(-\infty) = 0$ for the Hankel operator Γ_G and obtain

$$z(t) = \int_{-\infty}^{t} Ce^{A(t-\tau)}Bw(\tau)\mathrm{d}\tau.$$

If we now focus on input signals w_- that live for times $t \leqslant 0$ and vanish for $t > 0$, then the output restricted to $t \geqslant 0$ is

$$z_+(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bw_-(\tau)\mathrm{d}\tau = \Gamma_G w_-,\ t \geqslant 0.$$

Assuming $x(0) = 0$ in (i), it now follows from the time-invariance that

$$\sup_{\substack{T>0 \\ 0\neq w\in L^2[0,T] \\ w(t)=0,\, t>T}} \frac{\|z\|_{2,[T,\infty)}}{\|w\|_{2,[0,T]}} = \sup_{\substack{T>0 \\ 0\neq w\in L^2[-T,0] \\ w(t)=0,\, t>0}} \frac{\|z\|_{2,[0,\infty)}}{\|w\|_{2,[-T,0]}} = \sup_{\substack{0\neq w\in L^2(-\infty,0] \\ w(t)=0,\, t>0}} \frac{\|z\|_{2,[0,\infty)}}{\|w\|_{2,(-\infty,0]}}$$

$$= \sup_{0\neq w_-\in L^2(-\infty,0]} \frac{\|z_+\|_{2,[0,\infty)}}{\|w_-\|_{2,(-\infty,0]}} = \|\Gamma_G\|.$$

This gives the equivalence of (i) and (ii). Next, we have

$$\begin{aligned}\langle w, \Gamma_G^* z\rangle_{L^2(-\infty,0]} &= \langle \Gamma_G w, z\rangle_{L^2[0,\infty)} \\ &= \int_0^\infty \left(\int_{-\infty}^0 w(\tau)^\top B^\top e^{A^\top(t-\tau)} C^\top \mathrm{d}\tau\right) z(t)\mathrm{d}t \\ &= \int_{-\infty}^0 w(\tau)^\top \left(\int_0^\infty B^\top e^{A^\top(t-\tau)} C^\top z(t)\mathrm{d}t\right) \mathrm{d}\tau,\end{aligned}$$

which implies

$$(\Gamma_G^* z)(\tau) = \int_0^\infty B^\top e^{A^\top(t-\tau)} C^\top z(t)\mathrm{d}t,\ \tau \leqslant 0.$$

Note that the operator norm of Γ_G is equal to its maximum singular value. Therefore, to complete the proof, we show that $\sigma_i^2(\Gamma_G) = \lambda_i(XY)$, where $\sigma_i(\cdot)$ and $\lambda_i(\cdot)$ denote, respectively, the ith singular value and ith eigenvalue of an operator or matrix. Suppose σ is a nonzero singular value of Γ_G, and w is an eigenvector corresponding to the eigenvalue σ^2 of $\Gamma_G^*\Gamma_G$, i.e., $\Gamma_G^*\Gamma_G w = \sigma^2 w$. Setting $z(t) = (\Gamma_G w)(t) = Ce^{At}x_0$ with $x_0 = \int_{-\infty}^0 e^{-A\tau}Bw(\tau)\mathrm{d}\tau$, and noting by [26, Lemma 3.18] that

$$X = \int_0^\infty e^{At}BB^\top e^{A^\top t}\mathrm{d}t,\ Y = \int_0^\infty e^{A^\top t}C^\top Ce^{At}\mathrm{d}t,$$

we have

$$\begin{aligned}\sigma^2 w = \Gamma_G^* z &= B^\top e^{-A^\top\tau}\int_0^\infty e^{A^\top t}C^\top z(t)\mathrm{d}t \\ &= B^\top e^{-A^\top\tau}\int_0^\infty e^{A^\top t}C^\top Ce^{At}x_0\mathrm{d}t = B^\top e^{-A^\top\tau}Yx_0.\end{aligned}$$

It follows that

$$\sigma^2 x_0 = \int_{-\infty}^0 e^{-A\tau}B\sigma^2 w(\tau)\mathrm{d}\tau = \int_{-\infty}^0 e^{-A\tau}BB^\top e^{-A^\top\tau}Yx_0\mathrm{d}\tau = XYx_0.$$

Moreover, $x_0 \neq 0$ since otherwise $\sigma^2 w = 0$, which is impossible. Thus, σ^2 is an eigenvalue of XY. Conversely, if $\sigma^2 \neq 0$ is an eigenvalue and $x_0 \neq 0$ is a corresponding eigenvector of XY, i.e., $XYx_0 = \sigma^2 x_0$, then by setting $w = B^\top e^{-A\tau}Yx_0$ we obtain $w \neq 0$ and $\Gamma_G^*\Gamma_G w = \sigma^2 w$. Hence, $\sigma_i^2(\Gamma_G) = \lambda_i(XY)$, and so

$$\|\Gamma_G\| = \sigma_1(\Gamma_G) = \sqrt{\lambda_1(XY)}.$$

The lemma is proved. □

Lemma 1 shows that the Hankel norm can be considered as a measure of controllability and observability of the system, and that it does not depend on the state-space representation of the system. It is now clear that problem (2) may be cast as an eigenvalue optimization program. In the sequel, we examine how this problem can be solved algorithmically.

4 Subgradients of the Hankel norm

In this section, we compute Clarke subgradients [8, Section 2.1] of the nonconvex composite function $f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2$. This is a fundamental tool for our optimization method.

Let $G(\mathbf{x})$ be a linear time-invariant system with state-space realization $(A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x}))$ depending smoothly on a design parameter $\mathbf{x} \in \mathbb{R}^n$. Let $X(\mathbf{x}), Y(\mathbf{x})$ be the controllability and observability Gramians. Suppose the maximum eigenvalue $\lambda_1(Z(\mathbf{x}))$ of the matrix $Z(\mathbf{x}) = X(\mathbf{x})^{\frac{1}{2}} Y(\mathbf{x}) X(\mathbf{x})^{\frac{1}{2}}$ has multiplicity $r(\mathbf{x})$, and let $R = R(\mathbf{x})$ be a matrix whose columns form an orthonormal basis of the eigenspace associated with $\lambda_1(Z(\mathbf{x}))$. For any matrix function $M(\mathbf{x})$, put $M_k(\mathbf{x}) = \frac{\partial M(\mathbf{x})}{\partial \mathbf{x}_k}$ and write $M_k^{\frac{1}{2}}$ for $(M^{\frac{1}{2}})_k$, $k = 1, \ldots, n$. We have the following

Proposition 1 *The function $f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2$ is well defined and locally Lipschitz on the set $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : A(\mathbf{x}) \text{ stable}\}$. In addition, for every $\mathbf{x}$ in the set $\mathcal{S}_0 = \{\mathbf{x} \in \mathcal{S} : (A(\mathbf{x}), B(\mathbf{x})) \text{ controllable}\}$ the Clarke subgradients of f at $\mathbf{x}$ have the form*

$$g_U = \left[\operatorname{Tr}(UR^\top Z_1(\mathbf{x})R) \ \ldots \ \operatorname{Tr}(UR^\top Z_n(\mathbf{x})R)\right]^\top, \tag{3}$$

where U is symmetric of size $r \times r$, $U \succeq 0$, $\operatorname{Tr}(U) = 1$, and where the partial derivatives $Z_k(\mathbf{x}), k = 1, \ldots, n$ are given by

$$Z_k(\mathbf{x}) = X_k^{\frac{1}{2}}(\mathbf{x}) Y X^{\frac{1}{2}} + X^{\frac{1}{2}} Y_k(\mathbf{x}) X^{\frac{1}{2}} + X^{\frac{1}{2}} Y X_k^{\frac{1}{2}}(\mathbf{x}). \tag{4}$$

Here, $X_k(\mathbf{x})$, $Y_k(\mathbf{x})$ and $X_k^{\frac{1}{2}}(\mathbf{x})$ are the solutions of the following Lyapunov equations

$$AX_k(\mathbf{x}) + X_k(\mathbf{x})A^\top = -A_k(\mathbf{x})X - XA_k(\mathbf{x})^\top - B_k(\mathbf{x})B^\top - BB_k(\mathbf{x})^\top, \tag{5}$$

$$A^\top Y_k(\mathbf{x}) + Y_k(\mathbf{x})A = -A_k(\mathbf{x})^\top Y - YA_k(\mathbf{x}) - C_k(\mathbf{x})^\top C - C^\top C_k(\mathbf{x}), \tag{6}$$

$$X^{\frac{1}{2}} X_k^{\frac{1}{2}}(\mathbf{x}) + X_k^{\frac{1}{2}}(\mathbf{x}) X^{\frac{1}{2}} = X_k(\mathbf{x}). \tag{7}$$

Proof 1. By Lemma 1,

$$f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2 = \lambda_1(X(\mathbf{x})Y(\mathbf{x})),$$

where the Gramians $X(\mathbf{x})$ and $Y(\mathbf{x})$ depend on the tunable parameters $\mathbf{x}$ and are the solutions of the Lyapunov equations

$$A(\mathbf{x})X + XA(\mathbf{x})^\top + B(\mathbf{x})B(\mathbf{x})^\top = 0, \tag{8}$$

$$A(\mathbf{x})^\top Y + YA(\mathbf{x}) + C(\mathbf{x})^\top C(\mathbf{x}) = 0. \tag{9}$$

Note that despite the symmetry of X and Y the product XY is not necessarily symmetric, but stability of $A(\mathbf{x})$ guarantees $X \succeq 0$, $Y \succeq 0$ in (8), (9), so that we can write

$$\lambda_1(XY) = \lambda_1(X^{\frac{1}{2}} Y X^{\frac{1}{2}}) = \lambda_1(Y^{\frac{1}{2}} X Y^{\frac{1}{2}}),$$

which brings us back to the realm of eigenvalue theory of symmetric matrices. By positive semidefiniteness of $X(\mathbf{x})$ and $Y(\mathbf{x})$, the function f is now well defined on $\mathcal{S}$.

2. Let us next prove that f is locally Lipschitz on $\mathcal{S}$. Using the Kronecker product [3], Eq. (8) can be written as

$$(I \otimes A(\mathbf{x}) + A(\mathbf{x}) \otimes I)\mathrm{vec}(X(\mathbf{x})) = -\mathrm{vec}(B(\mathbf{x})B(\mathbf{x})^\top),$$

where I is a conformable identity matrix, and where $\mathrm{vec}(\cdot)$ vectorizes a matrix by stacking its columns in order. Since $A(\mathbf{x})$ is smooth in $\mathbf{x}$ and $M(\mathbf{x}) = (I \otimes A(\mathbf{x}) + A(\mathbf{x}) \otimes I)$ is invertible by the stability of $A(\mathbf{x})$, $M(\mathbf{x})^{-1}$ is also smooth in $\mathbf{x}$, and since $B(\mathbf{x})$ depends smoothly on $\mathbf{x}$, then so does $\mathrm{vec}(X(\mathbf{x})) = -M(\mathbf{x})^{-1}\mathrm{vec}\left(B(\mathbf{x})B(\mathbf{x})^\top\right)$. A similar argument shows smooth dependence of $Y(\mathbf{x})$ on $\mathbf{x}$. This can also be justified based on the explicit formulas

$$X(\mathbf{x}) = \int_0^\infty e^{A(\mathbf{x})t}B(\mathbf{x})B(\mathbf{x})^\top e^{A(\mathbf{x})^\top t}\mathrm{d}t, Y(\mathbf{x}) = \int_0^\infty e^{A(\mathbf{x})^\top t}C(\mathbf{x})^\top C(\mathbf{x})e^{A(\mathbf{x})t}\mathrm{d}t$$

(see e.g., [26, Lemmas 2.7 and 3.18]), where uniform convergence of these integrals on any bounded set of $\mathbf{x}$ gives differentiability in $\mathbf{x}$. We infer that the coefficients of the characteristic polynomial of $X(\mathbf{x})Y(\mathbf{x})$ also depend smoothly on $\mathbf{x}$. Since this characteristic polynomial is hyperbolic, that is, has only real roots, we may invoke the multi-parameter version of Bronstein's theorem [6], for which an elegant proof is given in [19, Theorem 2], to conclude that $f(\mathbf{x}) = \lambda_1(X(\mathbf{x})Y(\mathbf{x}))$ is locally Lipschitz on $\mathcal{S}$.

3. Let us finally establish formula (3) for the subdifferential $\partial f(\mathbf{x})$ at points $\mathbf{x} \in \mathcal{S}_0$. By the above argument, $f(\mathbf{x}) = \lambda_1(Z(\mathbf{x}))$. Observe that controllability of $(A(\mathbf{x}), B(\mathbf{x}))$ implies that $X(\mathbf{x})$ is positive definite [26, Theorem 3.1], and since the operator $X \to X^{\frac{1}{2}}$ is smooth on the set of matrices $X \succ 0$, the chain rule gives smoothness of $\mathbf{x} \to X^{\frac{1}{2}}(\mathbf{x})$, and so of $Z(\mathbf{x}) = X^{\frac{1}{2}}YX^{\frac{1}{2}}$, on $\mathcal{S}_0$.
Applying [18, Theorem 3], the Clarke subgradients of f at $\mathbf{x}$ are of the form $g_U = \begin{bmatrix} g_1 & \dots & g_n \end{bmatrix}^\top$, where

$$g_k = \left\langle U, R^\top Z_k(\mathbf{x})R \right\rangle = \mathrm{Tr}(UR^\top Z_k(\mathbf{x})R)$$

for U symmetric of size $r \times r$, $U \succeq 0$, $\mathrm{Tr}(U) = 1$. It now remains to calculate $Z_k(\mathbf{x}), k = 1, \dots, n$. We first have (4) by the definition of Z. Taking derivatives with respect to $\mathbf{x}$ on both sides of (8)–(9), we get (5)–(6), and then also $X_k(\mathbf{x})$, $Y_k(\mathbf{x})$. Finally, to compute $X_k^{\frac{1}{2}}(\mathbf{x})$, we use (7), which is obtained by differentiating $X^{\frac{1}{2}}X^{\frac{1}{2}} = X$. Altogether, we obtain Clarke subgradients of f at each $\mathbf{x}$ due to (3)–(9). □

Remark 1 Formula (3) also holds if controllability of $(A(\mathbf{x}), B(\mathbf{x}))$ is replaced by observability of $(A(\mathbf{x}), C(\mathbf{x}))$ (cf. [26, Definition 3.4]). Here, we work with $Z = Y^{\frac{1}{2}}XY^{\frac{1}{2}}$ instead.

Remark 2 In the discrete time case, the Gramians $X(\mathbf{x})$ and $Y(\mathbf{x})$ are the solutions of the discrete Lyapunov equations

$$A(\mathbf{x})XA(\mathbf{x})^\top - X + B(\mathbf{x})B(\mathbf{x})^\top = 0,$$
$$A(\mathbf{x})^\top YA(\mathbf{x}) - Y + C(\mathbf{x})^\top C(\mathbf{x}) = 0,$$

so that $X_k(\mathbf{x})$ and $Y_k(\mathbf{x})$ are solutions, respectively, of the following equations

$$AX_k(\mathbf{x})A^\top - X_k(\mathbf{x}) = -A_k(\mathbf{x})XA^\top - AXA_k(\mathbf{x})^\top - B_k(\mathbf{x})B^\top - BB_k(\mathbf{x})^\top,$$
$$A^\top Y_k(\mathbf{x})A - Y_k(\mathbf{x}) = -A_k(\mathbf{x})^\top YA - A^\top YA_k(\mathbf{x}) - C_k(\mathbf{x})^\top C - C^\top C_k(\mathbf{x}).$$

Remark 3 Subgradients of f at $\mathbf{x} \in \mathcal{S} \setminus \mathcal{S}_0$ are no longer represented by (3), since the solution of (7) need not exist, as only $X^{\frac{1}{2}} \succeq 0$ is guaranteed. Nonetheless, by Clarke subdifferentiability at points $\mathbf{x} \in \mathcal{S}\setminus\mathcal{S}_0$ proved above, we can be sure that for every sequence $\mathbf{x}_k \in \mathcal{S}_0$ converging to $\mathbf{x} \in \mathcal{S} \setminus \mathcal{S}_0$ and $g_k \in \partial f(\mathbf{x}_k)$ computed via (3), the g_k stay bounded and each of their accumulation points g is an element of $\partial f(\mathbf{x})$. This guarantees stability of our numerical procedure even when iterates get close to the set $\mathcal{S} \setminus \mathcal{S}_0$.

Remark 4 Practical parametrizations $G(\mathbf{x})$ use elementary computable operations, which can be expressed in mathematical terms by assuming that $A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x})$ are smooth *definable* functions of $\mathbf{x}$ in the sense of [25, Chap. 1, Sect. 5.3]. In that case, one can say a little more about the behavior of f at points $\mathbf{x} \in \mathcal{S}$. Namely, it then follows from [21, Theorem 4.12] that for every smooth definable curve $\mathbf{x}(t) \in \mathcal{S}$ the eigenvalues $\lambda_i(t) = \lambda_i(X(\mathbf{x}(t))Y(\mathbf{x}(t)))$ are smooth functions of t, so that $f(\mathbf{x}(t))$ is a finite maximum of smooth functions of t. On $\mathcal{S}_0$ this property is a consequence of symmetric eigenvalue theory, which is true without the definability hypothesis. Note that this does not mean that f is a finite maximum of smooth functions of $\mathbf{x} \in \mathbb{R}^n$, but it nonetheless indicates a favorable structure.

5 An extension of the Hankel norm

Lemma 1 shows why the Hankel norm is only a semi-norm on the space of internally stable systems G. It does not see a direct transmission D from w to z, as the latter does not create memory transmitted from the past to the future. This rises the question how to assess a direct transmission block in the context of (1) or (2). Namely, in some applications, attributing no cost to a block $D(\mathbf{x})$ which is free to vary with the tunable parameters $\mathbf{x}$ bears the risk that optimization favors a solution with a high energy direct transmission.

It is well known that $\|G\|_H \leqslant \|G\|_\infty$ in the case $D = 0$ (See e.g., [5, Sect. 5.5]), and this may guide us to define an extension. Note first that

Lemma 2 $\|(A, B, C)\|_H \leqslant \|(A, B, C, D)\|_\infty$ *for every internally stable system* $G = (A, B, C, D)$.

Proof Let $G^0 = (A, B, C)$ be the system where the direct transmission is ignored. Consider an input w with $w(t) = 0$ for $t > T$, and let $z^0 = G^0 w$, $z = Gw$. Then, $z(t) = z^0(t)$ for $t > T$, because the direct transmission creates no memory, and since $w(t) = 0$ for $t > T$, its influence on the output ends at T. Combining this with $\|w\|_{2,[0,T]} = \|w\|_2$ and $\|z\|_{2,[T,\infty)} \leqslant \|z\|_2$, we obtain

$$\begin{aligned}\|(A,B,C)\|_H &= \sup_{\substack{T>0\\ 0\neq w\in L^2[0,T]\\ w(t)=0,\, t>T}} \frac{\|z\|_{2,[T,\infty)}}{\|w\|_{2,[0,T]}} \leqslant \sup_{\substack{T>0\\ 0\neq w\in L^2[0,T]\\ w(t)=0,\, t>T}} \frac{\|z\|_2}{\|w\|_2} \\ &\leqslant \sup_{w\neq 0} \frac{\|z\|_2}{\|w\|_2} = \|(A,B,C,D)\|_\infty.\end{aligned}$$

□

This suggests the following extension of Hankel norm $\|\cdot\|_H$ to systems $G = (A, B, C, D)$ with direct transmission D.

Definition 1 Let $G = (A, B, C, D)$ be an internally stable linear time-invariant system. Then,

$$\|G\|_H = \max\{\|(A,B,C)\|_H, \sigma_1(D)\} \tag{10}$$

is called the extended Hankel norm of the system. Here, σ_1 denotes the maximum singular value of a matrix. □

This definition agrees with the usual Hankel norm for a system without direct transmission, and also preserves the inequality $\|G\|_H \leqslant \|G\|_\infty$, since the term $\sigma_1(D)$ is part of the maximum $\|G\|_\infty = \max_\omega \sigma_1(G(j\omega))$ at $\omega = \infty$.

As the proof of Lemma 2 shows, a direct transmission does not change the value of $\|\cdot\|_H$ defined according to (1). In the sequel, we therefore adopt the convention that in the case $D \neq 0$, $\|(A, B, C)\|_H$ is the usual Hankel norm, where the direct transmission is ignored, while $\|(A, B, C, D)\|_H$ is the extended Hankel norm.

An advantage of (10) is that the new function is still a maximum eigenvalue function. Namely, stability of G implies positive semidefiniteness of the Gramians X and Y, and so

$$\|G\|_H^2 = \max\left\{\lambda_1(X^{\frac{1}{2}}YX^{\frac{1}{2}}), \lambda_1(D^\top D)\right\} = \lambda_1\begin{bmatrix} X^{\frac{1}{2}}YX^{\frac{1}{2}} & 0 \\ 0 & D^\top D\end{bmatrix}. \tag{11}$$

Proceeding as in the proof of Proposition 1, we get immediately the following

Corollary 1 *Let $G(\mathbf{x})$ be a linear time-invariant system depending smoothly on $\mathbf{x} \in \mathcal{S}$ with $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : A(\mathbf{x})$ stable$\}$. Suppose the maximum eigenvalue $\lambda_1(\mathcal{Z}(\mathbf{x}))$ of the matrix*

$$\mathcal{Z}(\mathbf{x}) = \begin{bmatrix} X(\mathbf{x})^{\frac{1}{2}}Y(\mathbf{x})X(\mathbf{x})^{\frac{1}{2}} & 0 \\ 0 & D(\mathbf{x})^\top D(\mathbf{x})\end{bmatrix}$$

has multiplicity $r = r(\mathbf{x})$*, and* $R = R(\mathbf{x})$ *is a matrix whose columns form an orthonormal basis of the eigenspace associated with* $\lambda_1(\mathcal{Z}(\mathbf{x}))$*. With the notations of Proposition* 1*, the function* $f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2$ *is locally Lipschitz on* $\mathcal{S}$ *and its Clarke subgradients on* $\mathcal{S}_0 = \{\mathbf{x} \in \mathcal{S} : (A(\mathbf{x}), B(\mathbf{x}))$ controllable$\}$ *have the form*

$$g_U = \left[\mathrm{Tr}(UR^\top \mathcal{Z}_1(\mathbf{x})R) \ \ldots \ \mathrm{Tr}(UR^\top \mathcal{Z}_n(\mathbf{x})R)\right]^\top,$$

for U *symmetric of size* $r \times r$*,* $U \succeq 0$*,* $\mathrm{Tr}(U) = 1$*, where the partial derivatives* $\mathcal{Z}_k(\mathbf{x}), k = 1, \ldots, n$ *are given by*

$$\mathcal{Z}_k(\mathbf{x}) = \begin{bmatrix} Z_k(\mathbf{x}) & 0 \\ 0 & D_k(\mathbf{x})^\top D(\mathbf{x}) + D(\mathbf{x})^\top D_k(\mathbf{x}) \end{bmatrix}$$

and the $Z_k(\mathbf{x})$ *are defined in Proposition* 1*.* □

To justify the use of (10) rigorously, we consider the extended Hankel norm minimization program (2) based on (10), and compare it to the following constraint program

$$\begin{array}{l} \text{minimize} \quad f(\mathbf{x}) = \|(A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x}))\|_H \\ \text{subject to } h(\mathbf{x}) = \sigma_1\left(D(\mathbf{x})\right) \leqslant \eta. \end{array} \tag{12}$$

For the following, recall from [13] that $\mathbf{x}^* \in \mathbb{R}^n$ is called a Fritz John critical point of the constraint program $\min\{f(\mathbf{x}) : h(\mathbf{x}) \leqslant \eta\}$ if there exist multipliers $\lambda_0^* \geqslant 0$, $\lambda_1^* \geqslant 0$, *not both zero*, such that

$$0 \in \lambda_0^* \partial f(\mathbf{x}^*) + \lambda_1^* \partial h(\mathbf{x}^*), \quad h(\mathbf{x}^*) \leqslant \eta, \quad \lambda_1^*\left(h(\mathbf{x}^*) - \eta\right) = 0.$$

If in addition $\lambda_0^* > 0$, then $\mathbf{x}^*$ is called a Karush–Kuhn–Tucker point. Remember that every local minimum $\mathbf{x}^*$ of the constraint program is automatically a Fritz John critical point, while it will in general only be a Karush–Kuhn–Tucker point if an additional constraint qualification is satisfied [13, Chapter 7]. For later on, we call $\mathbf{x}^*$ a critical point of constraint violation if $0 \in \partial h(\mathbf{x}^*)$ and $h(\mathbf{x}^*) > \eta$.

With these preparations, we have the following

Proposition 2 *Let* $\mathbf{x}^*$ *be a critical point of the extended Hankel norm minimization program* (2) *with* (10)*. Then,* $\mathbf{x}^*$ *is a Fritz John critical point of program* (12) *for a suitable choice of* η*. More precisely,* $\mathbf{x}^*$ *is either a Karush–Kuhn–Tucker point of* (12)*, or a critical point of* $h(\mathbf{x}) = \sigma_1(D(\mathbf{x}))$ *alone.*

Proof Note that $\|G(\mathbf{x})\|_H = \max\{f(\mathbf{x}), h(\mathbf{x})\}$. Now, if $\mathbf{x}^*$ is a critical point of $\|G(\mathbf{x})\|_H$, then we have three possibilities, $f(\mathbf{x}^*) > h(\mathbf{x}^*)$, $f(\mathbf{x}^*) = h(\mathbf{x}^*)$, or $f(\mathbf{x}^*) < h(\mathbf{x}^*)$. In the first case, $\mathbf{x}^*$ is a critical point of f alone, hence also a Karush–Kuhn–Tucker point of (12). The third case corresponds to a critical point of h alone. In the case of equality, the situation is more complex. There exist multipliers $\lambda_0^* \geqslant 0$, $\lambda_1^* \geqslant 0$, not both zero, such that $0 \in \lambda_0^* \partial f(\mathbf{x}^*) + \lambda_1^* \partial h(\mathbf{x}^*)$. If $\lambda_0^* = 0$ then $\lambda_1^* \neq 0$ and $0 \in \partial h(\mathbf{x}^*)$, so $\mathbf{x}^*$ is a critical point of h. In case $\lambda_0^* \neq 0$, we have $0 \in \partial f(\mathbf{x}^*) + (\lambda_1^*/\lambda_0^*)\partial h(\mathbf{x}^*)$. This is the first part of the Karush–Kuhn–Tucker conditions. If we put $\eta = f(\mathbf{x}^*)$, then we also get the second half. That completes the argument. □

Remark 5 Suppose we solve program $\min\{f(\mathbf{x}) : h(\mathbf{x}) \leqslant \eta\}$ starting at an infeasible point $h(\mathbf{x}^1) > \eta$, then we will usually try to minimize h alone to find a feasible iterate. Suppose a descent method used to minimize h runs into a local minimum $\mathbf{x}^*$ of h satisfying $h(\mathbf{x}^*) > \eta$. Such a *local minimum of constraint violation* indicates a failure, since nothing better will be found in a neighborhood of $\mathbf{x}^*$ due to local optimality, so that the search for a feasible point has to be stared anew elsewhere; cf. [20, Section 2.2] for this theme complex.

By Proposition 2 we can now interpret minimization of the extended Hankel norm (2) with (10) as a trade-off between minimizing the memory effects of $(A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x}))$, subject to a constraint $\sigma_1(D(\mathbf{x})) \leqslant \eta$, or dually, as of minimizing $\sigma_1(D(\mathbf{x}))$ subject to a constraint on the memory effects of $G(\mathbf{x})$. Since $f(\mathbf{x})$ is a valid measure of the memory or ringing effects of $G(\mathbf{x})$, such an interpretation is physically meaningful.

We conclude this section by showing that the Hankel norm is amenable to optimization techniques, as this will be needed later. According to Spingarn [24] a function $f : U \to \mathbb{R}$, where U is an open set in $\mathbb{R}^n$, is *lower-C^1* on U, if for each $\mathbf{x}_0 \in U$, there are a compact space K, a neighborhood V of $\mathbf{x}_0$, and a jointly continuous function $F : V \times K \to \mathbb{R}$ whose partial derivative $D_{\mathbf{x}}F$ with respect to $\mathbf{x}$ exists and is jointly continuous, such that $f(\mathbf{x}) = \max_{\mathbf{z}\in K} F(\mathbf{x}, \mathbf{z})$ for all $\mathbf{x} \in V$.

Proposition 3 *Let $G(\mathbf{x}) = (A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x}), D(\mathbf{x}))$ be a linear time-invariant system depending smoothly on the set $\mathcal{S}_0$ of all $\mathbf{x} \in \mathbb{R}^n$ such that $A(\mathbf{x})$ is stable and $(A(\mathbf{x}), B(\mathbf{x}))$ is controllable or $(A(\mathbf{x}), C(\mathbf{x}))$ is observable. Then, $f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2$ is lower-C^1 on $\mathcal{S}_0$.*

Proof For each $\mathbf{x} \in \mathcal{S}_0$, according to (11) and using the Rayleigh quotient,

$$f(\mathbf{x}) = \lambda_1(\mathcal{Z}(\mathbf{x})) = \max_{\|\mathbf{z}\|=1} \mathbf{z}^\top \mathcal{Z}(\mathbf{x})\mathbf{z},$$

where $\mathcal{Z}$ is symmetric and depends smoothly on $\mathbf{x}$. Set $K = \{\mathbf{z} \in \mathbb{R}^m : \|\mathbf{z}\| = 1\}$ and $F(\mathbf{x}, \mathbf{z}) = \mathbf{z}^\top \mathcal{Z}(\mathbf{x})\mathbf{z}$, then K is compact, $f(\mathbf{x}) = \max_{\mathbf{z}\in K} F(\mathbf{x}, \mathbf{z})$, and both F and its partial derivatives $F_{\mathbf{x}}$ are jointly continuous on $\mathcal{S}_0 \times K$ and smooth in $\mathbf{x}$. Therefore, f is lower-C^1 on $\mathcal{S}_0$. □

6 Hankel synthesis

The first application of program (2) we consider is output feedback controller synthesis, where performance is assessed by the Hankel norm. Consider a linear time-invariant plant in standard form

$$P(s) : \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}, \tag{13}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{m_2}$ the control, $w \in \mathbb{R}^{m_1}$ the vector of exogenous inputs, $y \in \mathbb{R}^{p_2}$ the measurements, and $z \in \mathbb{R}^{p_1}$ the controlled or performance vector,

$$P(s) := \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (sI - A)^{-1} \begin{bmatrix} B_1 & B_2 \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}.$$

Without loss of generality, it is assumed that $D_{22} = 0$. Let $u(s) = K(s)y(s)$ be an output feedback controller for the open-loop plant (13), with

$$K : \begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \begin{bmatrix} x_K \\ y \end{bmatrix},$$

where $x_K \in \mathbb{R}^k$ is the state of K. The closed-loop transfer function of the performance channel $w \to z$ is obtained as

$$T_{w\to z}(K, s) = P_{11}(s) + P_{12}(s)K(s)(I - P_{22}(s)K(s))^{-1}P_{21}(s).$$

Our aim is to find an optimal controller K which stabilizes the system in closed-loop such that $\|T_{w\to z}(K)\|_H$ is minimized among all stabilizing K. By substituting $u = Ky$ into (13), the state-space representation of the closed-loop performance channel $w \to z$ is

$$T_{w\to z}(K) : \begin{bmatrix} \dot{\xi} \\ z \end{bmatrix} = \begin{bmatrix} \mathcal{A}(K) & \mathcal{B}(K) \\ \mathcal{C}(K) & \mathcal{D}(K) \end{bmatrix} \begin{bmatrix} \xi \\ w \end{bmatrix},$$

where $\xi = (x, x_K)$ and

$$\mathcal{A}(K) = \begin{bmatrix} A + B_2D_KC_2 & B_2C_K \\ B_KC_2 & A_K \end{bmatrix}, \quad \mathcal{B}(K) = \begin{bmatrix} B_1 + B_2D_KD_{21} \\ B_KD_{21} \end{bmatrix},$$
$$\mathcal{C}(K) = \begin{bmatrix} C_1 + D_{12}D_KC_2 & D_{12}C_K \end{bmatrix}, \mathcal{D}(K) = D_{11} + D_{12}D_KD_{21}.$$

This problem is now a specific instance of (2), where in agreement with our general theme we try to minimize the memory of a specific channel $w \to z$ within the plant P. If we allow structured control laws $K(\mathbf{x})$ in the sense of [1], then we obtain the following optimization program

$$\begin{array}{ll} \text{minimize} & \|T_{w\to z}(K)\|_H \\ \text{subject to} & K \text{ stabilizes (13) internally} \\ & K = K(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n. \end{array} \tag{14}$$

Example 1 Typical examples of structured controllers are, for instance, PIDs or observer-based controllers, which in state-space have the form

$$K_{\text{pid}}(\mathbf{x}) = \left[\begin{array}{cc|c} 0 & 0 & r_i \\ 0 & -\tau & r_d \\ \hline 1 & 1 & d_K \end{array}\right], \quad K_{\text{obs}}(\mathbf{x}) = \left[\begin{array}{c|c} A + B_2K_c + K_fC_2 & -K_f \\ \hline K_c & 0 \end{array}\right].$$

For a PID, the tunable parameters are $\mathbf{x} = (r_i, r_d, d_K, \tau)$, while for observer-based controllers $K_{\text{obs}}(\mathbf{x})$ the vector $\mathbf{x}$ gathers the elements of K_c, K_f. Other examples are decentralized, fixed reduced order controllers, and more generally, control architectures combining basic building blocks such as PIDs with filters, feed-forward blocks, and much else (see [1]).

Remark 6 The norm in program (14) is the usual Hankel norm (1) if $\mathcal{D}(K) = 0$, which is the case e.g., under standard assumption as in H_2-synthesis, where $D_{11} = 0$ and either $D_{21} = 0$ or $D_{12} = 0$ or K strictly proper. In contrast, if $\mathcal{D}(K) \neq 0$, then we should use the extended Hankel norm (10), or likewise, the constraint program (12), to control the direct transmission. It is also possible to neglect the direct transmission term $\mathcal{D}(K)$ and optimize the semi-norm $\|(\mathcal{A}(K), \mathcal{B}(K), \mathcal{C}(K))\|_H$. We then exercise caution by monitoring the term $\sigma_1(\mathcal{D}(K))$ during optimization to check whether a large direct transmission gain $\sigma_1(\mathcal{D}(K))$ is favored. If that is the case, switching to the extended Hankel norm becomes mandatory.

In the sequel of this section, we discuss two particular cases of the Hankel synthesis problem (14).

6.1 System reduction

System reduction is the most widely known application of the Hankel norm minimization problem. Given a stable system

$$G : \begin{cases} \dot{x} = Ax + Bw \\ z = Cx + Dw \end{cases}$$

of order n_x, we wish to find a stable system

$$G_k : \begin{cases} \dot{x} = A_k x + B_k w \\ z = C_k x + Dw \end{cases}$$

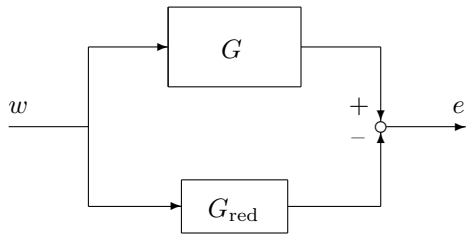
of reduced order $k < n_x$ with input–output behavior as close as possible to the original system G. If the model matching error $e = (G - G_k)w$ is measured in the Hankel norm, then the program

$$\begin{array}{ll} \text{minimize} & \|G - G_k(\mathbf{x})\|_H \\ \text{subject to} & G - G_k(\mathbf{x}) \text{ internally stable} \\ & \mathbf{x} = (A_k, B_k, C_k) \end{array} \tag{15}$$

is a particular case of (14), where we define plant and controller as

$$P : \left[\begin{array}{c|cc} A & B & 0 \\ \hline C & D & -I \\ 0 & I & 0 \end{array}\right], \qquad K : \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & D \end{array}\right], \tag{16}$$

the tunable parameters $\mathbf{x}$ being the elements of A_k, B_k and C_k.



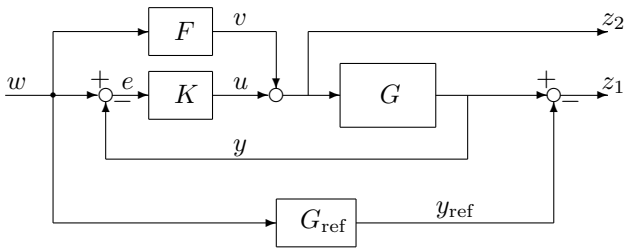
(17)

Due to the seminal work of Glover [12], program (15) has an explicit solution based on linear algebra, at least when no additional structural constraints on the matrices A_k, B_k, C_k are imposed. This allows us to implement a blind testing of Algorithm 1 in Sect. 8, which is applied to (15), considered as a particular case of (14) using (16). The value obtained by Algorithm 1 is then compared to the theoretical value obtained by an explicit Hankel system reduction.

6.2 Maximizing the memory of a system

Within the present framework, it is also possible to maximize the memory effects of a system G via feedback if a reference system G_{ref} with desirable memory properties is used. In other words, while minimizing $\|G(\mathbf{x})\|_H$ leads to a system which is the least biased, we now bias $G(\mathbf{x})$ as much as possible by bringing it as close as possible to G_{ref}, and we achieve this by making $G(\mathbf{x}) - G_{\text{ref}}$ as less biased as possible.

Example 2 As a motivating example, we consider a 2-DOF synthesis scheme of the following form



(18)

where the decentralized controller structure was chosen to challenge our method in a typical situation in practice.

Assuming that G_{ref} has desirable memory features which do not lead to ringing, the idea is to tune the parameters in feed-forward filter F and controller K in such a way that G in closed-loop follows G_{ref}, independently of the input w. That is, the undesirable part of the memory of G, which contributes to the mismatch $z_1 = y - y_{\text{ref}}$, is reduced by minimizing $\|T_{w\to z_1}(F,K)\|_H$. It may be beneficial to arrange this by adding a constraint $\|z_2\|_2 \leqslant \eta_2$ or $\|z_2\|_\infty \leqslant \eta_\infty$, where $z_2 = u + v$, to avoid exceedingly large controller actions. This problem can be cast as a particular case of program (14) if the following plant and decentralized controller structures are used

$$P: \left[\begin{array}{cc|ccc} A & 0 & 0 & B & B \\ 0 & A_{\text{ref}} & B_{\text{ref}} & 0 & 0 \\ \hline C & -C_{\text{ref}} & -D_{\text{ref}} & D & D \\ 0 & 0 & 0 & I & I \\ -C & 0 & I & -D & -D \\ 0 & 0 & I & 0 & 0 \end{array}\right], \qquad K: \left[\begin{array}{cc|cc} A_F & 0 & B_F & 0 \\ 0 & A_K & 0 & B_K \\ \hline C_F & 0 & D_F & 0 \\ 0 & C_K & 0 & D_K \end{array}\right].$$

Notice that

$$F : \begin{cases} \dot{x}_F = A_F x_F + B_F w \\ v \quad = C_F x_F + D_F w \end{cases}, \qquad K : \begin{cases} \dot{x}_K = A_K x_K + B_K e \\ u \quad = C_K x_K + D_K e \end{cases}$$

can be further structured if we wish. In our experiment, we will use this example with F a reduced-order filter, and K a PID.

7 Control of flow in a graph

We consider the flow in a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ with interior nodes, sources and sinks, $\mathscr{V} = \mathscr{V}_{\text{stay}} \cup \mathscr{V}_{\text{in}} \cup \mathscr{V}_{\text{out}}$, and not excluding self-arcs. For nodes $i, j \in \mathscr{V}$ connected by an arc $(i, j) \in \mathscr{A}$ the transition probability $i \to j$ quantifies the tendency of flow going from node i towards node j. As an example consider for instance a large fairground with separated entrances and exits, with itineraries represented by the graph. By acting on the transition probabilities between nodes connected by arcs, we expect to guide the crowd in such a way that a steady flow is assured, and a safe evacuation is possible.

Assume that an individual at interior node $j \in \mathscr{V}_{\text{stay}}$ decides with probability $a_{jj'} \geqslant 0$ to proceed to a neighboring node $j' \in \mathscr{V}_{\text{stay}}$, where neighboring means $(j, j') \in \mathscr{A}$, or with probability $a_{jk} \geqslant 0$ to move to a neighboring exit node $k \in \mathscr{V}_{\text{out}}$, where $(j, k) \in \mathscr{A}$. The case $(j, j) \in \mathscr{A}$ of deciding to stay at stand $j \in \mathscr{V}_{\text{stay}}$ is not excluded. Similarly, an individual entering at $i \in \mathscr{V}_{\text{in}}$ proceeds to a neighboring interior node $j \in \mathscr{V}_{\text{stay}}$ with probability $b_{ij} \geqslant 0$, where $(i, j) \in \mathscr{A}$. We suppose for simplicity that there is no direct transmission from entrances to exits. Then,

$$\sum_{j' \in \mathscr{V}_{\text{stay}} : (j,j') \in \mathscr{A}} a_{jj'} + \sum_{k \in \mathscr{V}_{\text{out}} : (j,k) \in \mathscr{A}} a_{jk} = 1, \tag{19}$$

for every $j \in \mathscr{V}_{\text{stay}}$, and

$$\sum_{j \in \mathscr{V}_{\text{stay}} : (i,j) \in \mathscr{A}} b_{ij} = 1 \tag{20}$$

for every $i \in \mathscr{V}_{\text{in}}$. Let $x_j(t)$ denote the number of people present at interior node $j \in \mathscr{V}_{\text{stay}}$ and time t, and $w_i(t)$ the number of people entering the fairground through entry $i \in \mathscr{V}_{\text{in}}$ at time t. Then, the number of people present at interior node $j \in \mathscr{V}_{\text{stay}}$ and time $t + 1$ is

$$x_j(t+1) = \sum_{j' \in \mathscr{V}_{\text{stay}} : (j',j) \in \mathscr{A}} a_{j'j} x_{j'}(t) + \sum_{i \in \mathscr{V}_{\text{in}} : (i,j) \in \mathscr{A}} b_{ij} w_i(t),$$

while the number of people leaving the fairground at time t through exit $k \in \mathscr{V}_{\text{out}}$ is $\sum_{j \in \mathscr{V}_{\text{stay}} : (j,k) \in \mathscr{A}} a_{jk} x_j(t)$. To assess the evacuation pattern, we quantify the total number of people still inside the fairground at time t via the weighted sum

$$z(t) = \sum_{j \in \mathscr{V}_{\text{stay}}} c_j x_j(t),$$

where $c_j > 0$ are fixed weights, and where $c_j = 1$ would correspond to simply counting the number of people inside the fairground. We let $\mathbf{x}$ regroup the parameters $a_{jj'}, a_{jk}, b_{ij}$, so that the discrete linear time-invariant system has the form $G(\mathbf{x}) = (A(\mathbf{x}), B(\mathbf{x}), C)$, where C is the row vector of c_j's.

Let us now consider an evacuation scenario, where at time T the inflow $w(t)$ through the entrance gates is stopped by closing the gates, and the time until the fairground is evacuated is assessed by measuring the evacuation pattern $z(t)$, $t > T$. This corresponds to computing the Hankel norm $\|G(\mathbf{x})\|_H$, which identifies the worst case evacuation scenario. Minimizing $\|z\|_{2,[T,\infty)}/\|w\|_{2,(0,T]}$ may then be understood as enhancing overall safety of the network by orienting the crowd in such a way that the worst case evacuation time is minimized. This leads to the optimization program

$$\begin{array}{ll} \text{minimize} & \|G(\mathbf{x})\|_H \\ \text{subject to} & G(\mathbf{x}) \text{ internally stable} \\ & a_{jj'} \geqslant 0, a_{jk} \geqslant 0, b_{ij} \geqslant 0, (19), (20) \end{array} \tag{21}$$

which is a discrete version of (2) including linear constraints. Notice that these linear constraints are readily added in our algorithmic approach. In an extended model, one might consider measuring the number of people y at some selected nodes $i \in \mathscr{V}_{\text{stay}} \cup \mathscr{V}_{\text{out}}$, and use this to react via feedback $u = Ky$ at the entry gates. This leads to a problem where controller and parts of the plant are optimized simultaneously. Other variants include cases, where some of the probabilities $a_{jj'}$, b_{ij} are imposed and cannot be modified by the designer.

8 Proximal bundle algorithm

In this section, we present our main algorithm to solve programs (2) and (12). Let us consider an abstract constrained optimization program of the form

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & h(\mathbf{x}) \leqslant 0 \end{array} \tag{22}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the decision variable, and f and h are locally Lipschitz but potentially nonsmooth and nonconvex functions, representing objective and constraints. To find solutions of the constraint program (22), using an idea inspired by Polak [20, Section 2.2.2], we introduce the progress function

$$F(\mathbf{y}, \mathbf{x}) = \max\{f(\mathbf{y}) - f(\mathbf{x}) - \nu h(\mathbf{x})_+, h(\mathbf{y}) - h(\mathbf{x})_+\},$$

where $h(\mathbf{x})_+ = \max\{h(\mathbf{x}), 0\}$, and $\nu > 0$ is some fixed parameter (with $\nu = 1$ a typical value). One can think of $\mathbf{x}$ as the current iterate, and $\mathbf{y}$ as the next iterate or as a candidate to become the next iterate. We need to collect a few facts about F. Note first that $F(\mathbf{x}, \mathbf{x}) = 0$. For the subdifferential, we have the useful

Lemma 3 *Suppose f and h are lower-C^1 functions. Then, the Clarke subdifferential of the progress function F with respect to the first variable is obtained as*

$$\partial_1 F(\mathbf{x},\mathbf{x}) = \begin{cases} \partial f(\mathbf{x}) & \text{if } h(\mathbf{x}) < 0, \\ \operatorname{conv}\{\partial f(\mathbf{x}) \cup \partial h(\mathbf{x})\} & \text{if } h(\mathbf{x}) = 0, \\ \partial h(\mathbf{x}) & \text{if } h(\mathbf{x}) > 0. \end{cases}$$

Proof Applying the formula for the Clarke subdifferential of a maximum [8, Proposition 2.3.12] we readily get $\partial_1 F(\mathbf{x},\mathbf{x}) = \partial f(\mathbf{x})$ if $h(\mathbf{x}) < 0$, $\partial_1 F(\mathbf{x},\mathbf{x}) \subset \operatorname{conv}\{\partial f(\mathbf{x}) \cup \partial h(\mathbf{x})\}$ if $h(\mathbf{x}) = 0$, and $\partial_1 F(\mathbf{x},\mathbf{x}) = \partial h(\mathbf{x})$ if $h(\mathbf{x}) > 0$. But since f and g are lower-C^1, according to [24, Proposition 2.4, Theorem 3.9], they are Clarke regular, so we have equality in the second case $h(\mathbf{x}) = 0$. □

Lemma 4 *Suppose $\mathbf{x}^*$ is a local minimum of program* (22)*, then it is also a local minimum of $F(\cdot,\mathbf{x}^*)$, and $0 \in \partial_1 F(\mathbf{x}^*,\mathbf{x}^*)$. Conversely, if $0 \in \partial_1 F(\mathbf{x}^*,\mathbf{x}^*)$ then $\mathbf{x}^*$ is either a Karush–Kuhn–Tucker point of* (22)*, or a critical point of constraint violation.*

Proof Since $\mathbf{x}^*$ is a local minimum of (22), we have feasibility $h(\mathbf{x}^*) \leqslant 0$, and so $h(\mathbf{x}^*)_+ = 0$, which implies $F(\mathbf{y},\mathbf{x}^*) = \max\{f(\mathbf{y}) - f(\mathbf{x}^*), h(\mathbf{y})\}$. Now, there exists a neighborhood U of $\mathbf{x}^*$ such that $f(\mathbf{y}) \geqslant f(\mathbf{x}^*)$ for every $\mathbf{y} \in U$ with $h(\mathbf{y}) \leqslant 0$. We argue that $F(\mathbf{y},\mathbf{x}^*) \geqslant F(\mathbf{x}^*,\mathbf{x}^*)$ for every $\mathbf{y} \in U$. Namely, if $h(\mathbf{y}) > 0$, then $F(\mathbf{y},\mathbf{x}^*) \geqslant h(\mathbf{y}) > 0 = F(\mathbf{x}^*,\mathbf{x}^*)$. On the other hand, if $h(\mathbf{y}) \leqslant 0$, then $\mathbf{y}$ is feasible, and we have $f(\mathbf{y}) \geqslant f(\mathbf{x}^*)$ by what was said before. But then $F(\mathbf{y},\mathbf{x}^*) \geqslant f(\mathbf{y}) - f(\mathbf{x}^*) \geqslant 0 = F(\mathbf{x}^*,\mathbf{x}^*)$. This proves $\mathbf{x}^*$ is a local minimum of $F(\cdot,\mathbf{x}^*)$, and so $0 \in \partial_1 F(\mathbf{x}^*,\mathbf{x}^*)$.

Next, suppose $0 \in \partial_1 F(\mathbf{x}^*,\mathbf{x}^*)$, then by Lemma 3, there exist non-negative constants λ_0^*, λ_1^* summing up to 1 such that $0 \in \lambda_0^* \partial f(\mathbf{x}^*) + \lambda_1^* \partial h(\mathbf{x}^*)$. If $h(\mathbf{x}^*) > 0$, we have $\partial_1 F(\mathbf{x}^*,\mathbf{x}^*) = \partial h(\mathbf{x}^*)$, and then $0 \in \partial h(\mathbf{x}^*)$, meaning that $\mathbf{x}^*$ is a critical point of h. If $h(\mathbf{x}^*) < 0$ then $\partial_1 F(\mathbf{x}^*,\mathbf{x}^*) = \partial f(\mathbf{x}^*)$, so $\lambda_1^* = 0$ and $\mathbf{x}^*$ is a Karush–Kuhn–Tucker point of (22). Assume that $h(\mathbf{x}^*) = 0$ but $\mathbf{x}^*$ fails to meet the Karush–Kuhn–Tucker conditions, we then obtain $\lambda_0^* = 0$ and $0 \in \partial h(\mathbf{x}^*)$. This completes the proof of the lemma. □

The consequence of this argument is that we should seek points $\mathbf{x}^*$ with $0 \in \partial_1 F(\mathbf{x}^*,\mathbf{x}^*)$. We now present our method for computing solutions of program (22), which is based on this rationale. It generates a sequence $\mathbf{x}^j$ of estimates which converges to a solution $\mathbf{x}^*$ in the sense of subsequences. At the current iterate $\mathbf{x}$, the inner loop of the algorithm constructs first-order working models $\phi_k(\cdot,\mathbf{x})$ and the corresponding second-order working models

$$\Phi_k(\mathbf{y},\mathbf{x}) = \phi_k(\mathbf{y},\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathbf{x})^\top Q(\mathbf{x})(\mathbf{y}-\mathbf{x}),$$

updated with counter k. The $\Phi_k(\cdot,\mathbf{x})$ are approximations of $F(\cdot,\mathbf{x})$ around $\mathbf{x}$, where $Q(\mathbf{x})$ is symmetric, depends only on the current iterate $\mathbf{x}$, and may reflect second-order information of F around $\mathbf{x}$. The first-order working model $\phi_k(\cdot,\mathbf{x})$ has to satisfy $\phi_k(\mathbf{x},\mathbf{x}) = F(\mathbf{x},\mathbf{x}) = 0$ and $\partial_1 \phi_k(\mathbf{x},\mathbf{x}) \subset \partial_1 F(\mathbf{x},\mathbf{x})$ at

all instants k. This is guaranteed when $m_e(\cdot, \mathbf{x}) = g(\mathbf{x})^\top(\cdot - \mathbf{x})$ with $g(\mathbf{x}) \in \partial_1 F(\mathbf{x}, \mathbf{x})$ is an affine minorant of $\phi_k(\cdot, \mathbf{x})$ at all times k. We refer to $m_e(\cdot, \mathbf{x})$ as the exactness plane at $\mathbf{x}$.

For a given working model, we solve the tangent program

$$\min_{\mathbf{y}\in\mathbb{R}^n} \Phi_k(\mathbf{y}, \mathbf{x}) + \frac{\tau_k}{2}\|\mathbf{y} - \mathbf{x}\|^2,$$

with the so-called proximity control parameter $\tau_k > 0$. We require $Q(\mathbf{x})+\tau_k I \succ 0$, which assures that the tangent program is strictly convex and has a unique solution $\mathbf{y}^k$, called the trial step. According to standard terminology, $\mathbf{y}^k$ is called a serious step if it is accepted as the new iterate $\mathbf{y}^k = \mathbf{x}^+$, and a null step otherwise. Suppose $\mathbf{y}^k$ is a null step, then we will have to make sure that the next working model $\phi_{k+1}(\cdot, \mathbf{x})$ improves over $\phi_k(\cdot, \mathbf{x})$. This is achieved by adding cutting and aggregate planes. Let us first look at aggregation. The optimality condition for the tangent program implies

$$g_k^* := (Q(\mathbf{x}) + \tau_k I)(\mathbf{x} - \mathbf{y}^k) \in \partial_1 \phi_k(\mathbf{y}^k, \mathbf{x}).$$

We call $m_k^*(\cdot, \mathbf{x}) = \phi_k(\mathbf{y}^k, \mathbf{x}) + g_k^{*\top}(\cdot - \mathbf{y}^k) = a_k^* + g_k^{*\top}(\cdot - \mathbf{x})$ with $a_k^* = \phi_k(\mathbf{y}^k, \mathbf{x}) + g_k^{*\top}(\mathbf{x} - \mathbf{y}^k)$ the aggregate plane. By assuring that $m_k^*(\cdot, \mathbf{x})$ is an affine minorant of $\phi_{k+1}(\cdot, \mathbf{x})$, we have $\phi_{k+1}(\mathbf{y}^k, \mathbf{x}) \geqslant m_k^*(\mathbf{y}^k, \mathbf{x}) = \phi_k(\mathbf{y}^k, \mathbf{x})$.

A central element in bundle methods is the cutting plane whose role is to cut away the unsuccessful trial step $\mathbf{y}^k$. For each subgradient $g_k \in \partial_1 F(\mathbf{y}^k, \mathbf{x})$, the affine function $t_k(\cdot) = F(\mathbf{y}^k, \mathbf{x}) + g_k^\top(\cdot - \mathbf{y}^k)$ is a tangent to $F(\cdot, \mathbf{x})$ at $\mathbf{y}^k$. Without convexity, we cannot use $t_k(\cdot)$ directly as a cutting plane. Instead, we use a technique first analyzed in [14], which shifts the tangent down. Fixing a parameter $c > 0$, we define the cutting plane as

$$m_k(\cdot, \mathbf{x}) = t_k(\cdot) - s = a_k + g_k^\top(\cdot - \mathbf{x}), \tag{23}$$

where $a_k = \min\{t_k(\mathbf{x}), -c\|\mathbf{y}^k - \mathbf{x}\|^2\}$, and where $s = [t_k(\mathbf{x}) + c\|\mathbf{y}^k - \mathbf{x}\|^2]_+$ is the downshift. The detailed statement is described as Algorithm 1, while a flowchart of the algorithm is shown in Fig. 1. For more details we refer to [17, Section 3], [16, Section 4] for unconstrained optimization case, and [2, Section 5], [11, Section 3] for the constrained case.

Next, we establish the following result on the convergence of Algorithm 1.

Theorem 1 *Suppose that f and h in* (22) *are lower-C^1 functions, and let $\{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \leqslant f(\mathbf{x}^1)\}$ be bounded. Then, every accumulation point $\mathbf{x}^*$ of the sequence of serious iterates $\mathbf{x}^j$ generated by Algorithm* 1 *satisfies $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$. In other words, $\mathbf{x}^*$ is either a critical point of constraint violation, or a Karush–Kuhn–Tucker point of* (22).

Proof We will adapt the proof of Theorem 6.6 and Corollary 6.7 in [17] to our needs. For that let us recall a notion from [17, Definitions 2.1 and 6.1], which we apply here to the progress function F. We call $\phi : \mathbb{R}^n \times \mathcal{S} \to \mathbb{R}$ a strict first-order model of F on the set $\mathcal{S} \subset \mathbb{R}^n$ if for every $\mathbf{x} \in \mathcal{S}$ the function $\phi(\cdot, \mathbf{x})$ is convex and the following axioms hold:

Algorithm 1 Proximal bundle algorithm with downshifted tangents

Parameters: $0 < \gamma < \widetilde{\gamma} < \Gamma < 1, 0 < \delta \ll 1, 0 < q < T \leqslant \infty$.

▷ **Step 1** (`Initialize outer loop`)**.** Choose initial feasible guess $\mathbf{x}^1$, fix memory control parameter $\tau_1^\sharp$, and put outer loop counter $j = 1$.

◇ **Step 2** (`Stopping test`)**.** At outer loop counter j, stop if $0 \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$. Otherwise, take a symmetric matrix Q_j respecting $-qI \preceq Q_j \preceq qI$, and goto inner loop.

▷ **Step 3** (`Initialize inner loop`)**.** Put inner loop counter $k = 1$ and initialize control parameter $\tau_1 = \max\{\tau_j^\sharp, -\lambda_{\min}(Q_j) + \delta\}$, where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of a symmetric matrix. Choose initial working model $\phi_1(\cdot, \mathbf{x}^j) = g(\mathbf{x}^j)^\top(\cdot - \mathbf{x}^j)$ with $g(\mathbf{x}^j) \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$.

▷ **Step 4** (`Tangent program`)**.** At inner loop counter k, let $\Phi_k(\mathbf{y}, \mathbf{x}^j) = \phi_k(\mathbf{y}, \mathbf{x}^j) + \frac{1}{2}(\mathbf{y} - \mathbf{x}^j)^\top Q_j(\mathbf{y} - \mathbf{x}^j)$ and find solution $\mathbf{y}^k$ (`trial step`) of the tangent program

$$\min_{\mathbf{y}\in\mathbb{R}^n} \Phi_k(\mathbf{y}, \mathbf{x}^j) + \frac{\tau_k}{2}\|\mathbf{y} - \mathbf{x}^j\|^2.$$

◇ **Step 5** (`Acceptance test`)**.** Compute the quotient

$$\rho_k = \frac{F(\mathbf{y}^k, \mathbf{x}^j)}{\Phi_k(\mathbf{y}^k, \mathbf{x}^j)}.$$

If $\rho_k \geqslant \gamma$ (`serious step`), put $\mathbf{x}^{j+1} = \mathbf{y}^k$ and update memory element $\tau_{j+1}^\sharp$ as τ_k if $\rho_k < \Gamma$, and $\frac{1}{2}\tau_k$ otherwise. Reset $\tau_{j+1}^\sharp = T$ if $\tau_{j+1}^\sharp > T$, increase outer loop counter j and loop back to step 2. If $\rho_k < \gamma$ (`null step`), continue inner loop with step 6.

▷ **Step 6** (`Update working model`)**.** Generate a cutting plane $m_k(\cdot, \mathbf{x}^j)$ at null step $\mathbf{y}^k$ and counter k using downshifted tangents. Compute aggregate plane $m_k^*(\cdot, \mathbf{x}^j)$ at $\mathbf{y}^k$, and then build new working model $\phi_{k+1}(\cdot, \mathbf{x}^j)$ by adding the new cutting plane, keeping the exactness plane and using aggregation to avoid overflow.

◇ **Step 7** (`Update control parameter`)**.** Compute secondary control parameter

$$\widetilde{\rho}_k = \frac{M_k(\mathbf{y}^k, \mathbf{x}^j)}{\Phi_k(\mathbf{y}^k, \mathbf{x}^j)},$$

with $M_k(\mathbf{y}, \mathbf{x}^j) = m_k(\mathbf{y}, \mathbf{x}^j) + \frac{1}{2}(\mathbf{y} - \mathbf{x}^j)^\top Q_j(\mathbf{y} - \mathbf{x}^j)$. If $\widetilde{\rho}_k < \widetilde{\gamma}$ then keep $\tau_{k+1} = \tau_k$, otherwise step up $\tau_{k+1} = 2\tau_k$. Increase inner loop counter k and loop back to step 4.

(M_1) $\phi(\mathbf{x}, \mathbf{x}) = F(\mathbf{x}, \mathbf{x}) = 0$ and $\partial_1\phi(\mathbf{x}, \mathbf{x}) \subset \partial_1 F(\mathbf{x}, \mathbf{x})$.
($\widehat{M}_2$) If $\mathbf{y}_j \to \mathbf{x}$ and $\mathbf{x}_j \to \mathbf{x}$ then there exists $\varepsilon_j \to 0^+$ such that $F(\mathbf{y}_j, \mathbf{x}_j) - \phi(\mathbf{y}_j, \mathbf{x}_j) \leqslant \varepsilon_j\|\mathbf{y}_j - \mathbf{x}_j\|$.
(M_3) ϕ is jointly upper semicontinuous on $\mathbb{R}^n \times \mathcal{S}$, i.e., if $(\mathbf{y}_j, \mathbf{x}_j) \to (\mathbf{y}, \mathbf{x})$ then $\limsup_{j\to\infty} \phi(\mathbf{y}_j, \mathbf{x}_j) \leqslant \phi(\mathbf{y}, \mathbf{x})$.

Representing the cutting plane in (23) as $m_{\mathbf{y}^+}(\cdot, \mathbf{x}) = a + g^\top(\cdot - \mathbf{x})$ with $g \in \partial_1 F(\mathbf{y}^+, \mathbf{x})$ and $a = \min\{t_{\mathbf{y}^+}(\mathbf{x}), -c\|\mathbf{y}^+ - \mathbf{x}\|^2\}$, $t_{\mathbf{y}^+}(\cdot) = F(\mathbf{y}^+, \mathbf{x}) + g^\top(\cdot - \mathbf{y}^+)$, we define

$$\phi(\mathbf{y}, \mathbf{x}) = \sup\{m_{\mathbf{y}^+}(\mathbf{y}, \mathbf{x}) : \mathbf{y}^+ \in B(\mathbf{x}, r)\},$$

where $B(\mathbf{x}, r)$ is a fixed ball large enough to contain all possible trial steps, and where the supremum is over all possible cases of $m_{\mathbf{y}^+}(\cdot, \mathbf{x})$. It then follows

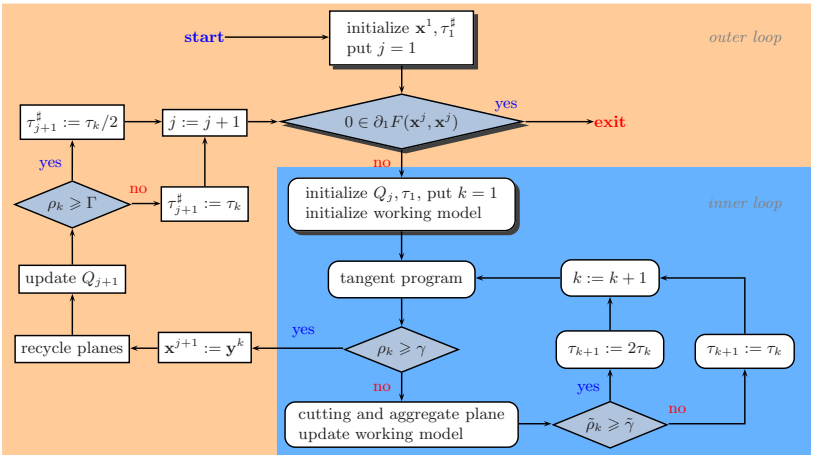


Fig. 1 Flowchart of proximal bundle algorithm. Inner loop is shown in the *lower right box*

that ϕ is a strict model of F in the sense of the above definition. This can be shown as in [16, Lemmas 7–9]. Axiom $(\widehat{M}_2)$ relies on the fact that $F(\cdot, \mathbf{x})$ is lower-C^1 by the assumptions on f and h. Furthermore, the construction of ϕ and ϕ_k also guarantees that the working models ϕ_k are lower approximations of ϕ satisfying $\phi_k(\mathbf{x}, \mathbf{x}) = \phi(\mathbf{x}, \mathbf{x}) = F(\mathbf{x}, \mathbf{x}) = 0$, $\partial_1 \phi_k(\mathbf{x}, \mathbf{x}) \subset \partial_1 \phi(\mathbf{x}, \mathbf{x})$ and $\phi_k(\cdot, \mathbf{x}) \leqslant \phi(\cdot, \mathbf{x})$. The difference with [17] is that here the cutting planes $m_k(\cdot, \mathbf{x})$ are not directly tangents of ϕ, but we shall argue that the essential link between ϕ_k and ϕ rests the same.

The proof now follows essentially [17, Theorem 6.6, Corollary 6.7], which assures that every accumulation point $\mathbf{x}^*$ of the iterates $\mathbf{x}^j$ satisfies $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$. Note that $f(\mathbf{x}^j)$ and $f(\mathbf{y}^k)$ used in [17] have to be replaced by $F(\mathbf{x}^j, \mathbf{x}^j) = 0$ and $F(\mathbf{y}^k, \mathbf{x})$. The fact that $\Phi(\mathbf{y}^{k+1})$ in the definition of $\widetilde{\rho}_k$ in [17] is changed to $M_k(\mathbf{y}^k, \mathbf{x})$ can be treated using the property that if $\mathbf{y}_j \to \mathbf{x}$ and $\mathbf{x}_j \to \mathbf{x}$ then there exists $\varepsilon_j \to 0^+$ such that $F(\mathbf{y}_j, \mathbf{x}_j) - m_{\mathbf{y}_j}(\mathbf{y}_j, \mathbf{x}_j) \leqslant \varepsilon_j \|\mathbf{y}_j - \mathbf{x}_j\|$, as follows from [16, Lemma 8], using again crucially that $F(\cdot, \mathbf{x})$ is lower-C^1. The equality $\phi_{k+1}(\mathbf{y}^{k+1}, \mathbf{x}) = \phi(\mathbf{y}^{k+1}, \mathbf{x})$ used in the proof of [17, Lemma 4.2] is now replaced by $\phi_{k+1}(\mathbf{y}^k, \mathbf{x}) \geqslant m_k(\mathbf{y}^k, \mathbf{x})$. Finally, Lemma 4 completes the last statement of the theorem. □

9 A smooth relaxation of the Hankel norm

Here, we introduce a smooth relaxation of the Hankel norm based on a result of Nesterov in [15]. He provides a fine analysis of the convex bundle method in situations where the objective $f(\mathbf{x})$ has the specific structure of a max-function, including the case of a convex maximum eigenvalue function. These findings indicate that for a given precision, such programs may be solved with lower algorithmic complexity using smooth relaxations. While these results are

a priori limited to the convex case, it may be interesting to apply Nesterov's idea as a heuristic in the nonconvex situation. This leads to the following

Proposition 4 *Let $\mathcal{Z}$ be a symmetric matrix of order m depending smoothly on a parameter $\mathbf{x} \in \mathbb{R}^n$ with eigenvalues $\lambda_1(\mathcal{Z}) \geqslant \cdots \geqslant \lambda_m(\mathcal{Z})$. Then, for a tolerance parameter $\mu > 0$, the function*

$$f_\mu(\mathbf{x}) = \mu \ln \left(\sum_{i=1}^{m} e^{\lambda_i(\mathcal{Z}(\mathbf{x}))/\mu} \right) \tag{24}$$

is a uniform smooth approximation of the nonsmooth function $f(\mathbf{x}) = \lambda_1(\mathcal{Z}(\mathbf{x}))$ in the sense that $f_\mu(\mathbf{x})$ converges uniformly to $f(\mathbf{x})$ as $\mu \to 0$.

Proof Following [15, Section 4], f_μ is smooth in $\mathcal{Z}$ and

$$\nabla f_\mu(\mathcal{Z}) = \left(\sum_{i=1}^{m} e^{\lambda_i(\mathcal{Z})/\mu} \right)^{-1} \sum_{i=1}^{m} e^{\lambda_i(\mathcal{Z})/\mu} q_i q_i^\top,$$

where q_i is the ith column of the orthogonal matrix $Q(\mathcal{Z})$ from the eigendecomposition of the symmetric matrix $\mathcal{Z} = Q(\mathcal{Z})D(\mathcal{Z})Q(\mathcal{Z})^\top$. This implies that f_μ is smooth at $\mathbf{x}$ with the gradient given by

$$\nabla f_\mu(\mathbf{x}) = \left[\operatorname{Tr}(\nabla f_\mu(\mathcal{Z}(\mathbf{x}))^\top \mathcal{Z}_1(\mathbf{x})) \ \ldots \ \operatorname{Tr}(\nabla f_\mu(\mathcal{Z}(\mathbf{x}))^\top \mathcal{Z}_m(\mathbf{x})) \right]^\top.$$

On the other hand, we have the estimate

$$f(\mathbf{x}) \leqslant f_\mu(\mathbf{x}) \leqslant f(\mathbf{x}) + \mu \ln m,$$

which says that $f_\mu(\mathbf{x})$ is a uniform approximation of the function $f(\mathbf{x})$. □

Now, we can try to solve problem (2) and (12) on replacing the function $f(\mathbf{x}) = \lambda_1(\mathcal{Z}(\mathbf{x}))$ by its smooth approximation $f_\mu(\mathbf{x})$ in (24). Due to the estimate in the above proof, to find an ε-solution $\bar{\mathbf{x}}$ of problem (2) and (12), we have to find an $\frac{\varepsilon}{2}$-solution of the smooth problem

$$\min\{f_\mu(\mathbf{x}) : h(\mathbf{x}) \leqslant 0\} \tag{25}$$

with $\mu = \frac{\varepsilon}{2 \ln m}$. Here, we use a local solution of (25) to initialize the nonsmooth Algorithm 1. The smooth problem (25) can be solved using standard NLP software.

10 Numerical experiments

In this section, we apply our approach to a variety of problems. Let us start by commenting on practical ways to implement the stopping test $0 \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$ in step 2 of the algorithm. In practice, this is delegated to the inner loop. If the inner loop at $\mathbf{x}^j$ finds a new feasible serious iterate $\mathbf{x}^{j+1}$ satisfying

$$\frac{|f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j)|}{1 + |f(\mathbf{x}^j)|} < \mathrm{tol}_1, \tag{26}$$

then we accept $\mathbf{x}^{j+1}$ as optimal. This corresponds to stopping the algorithm in step 2 of the $(j+1)$st outer loop. In our experiments, we have used $\mathrm{tol}_1 = 10^{-8}$.

On the other hand, if the inner loop has difficulties finding a serious step and provides three unsuccessful trial steps satisfying

$$\frac{\|\mathbf{x}^j - \mathbf{y}^k\|}{1 + \|\mathbf{x}^j\|} < \mathrm{tol}_2, \tag{27}$$

then we interpret this in the sense that $\mathbf{x}^j$ is already optimal. This corresponds to stopping the algorithm in step 2 of the jth outer loop. Here, we have used $\mathrm{tol}_2 = 10^{-7}$. Theoretically, both tests are based on the observation that $0 \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$ if and only if $\mathbf{y}^k = \mathbf{x}^j$ is solution of the tangent program in the trial step generation (see [11] for theoretical results).

In general, our stopping strategy is similar to recommendations in smooth optimization, see e.g., [10, Chapter 7], where the goal is to obtain scale independent choices of the tolerances tol_1 and tol_2. Nonetheless, one has to accept that a nonsmooth algorithm converges very slowly at the final stages, which makes stopping a delicate task.

Before applying Algorithm 1 to solve examples of (2), note that internal stability is not a constraint in the usual sense of mathematical programming since the set $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : G(\mathbf{x}) \text{ internally stable}\}$ is open. The stability of the system can be formulated as a constraint $\alpha(A(\mathbf{x})) \leqslant -\varepsilon$ using the spectral abscissa $\alpha(A) = \max\{\mathrm{Re}(\lambda) : \lambda \text{ eigenvalue of } A\}$ in the continuous time case, and as $\rho(A(\mathbf{x})) \leqslant 1 - \varepsilon$ using the spectral radius $\rho(A) = \max\{|\lambda| : \lambda \text{ eigenvalue of } A\}$ in the discrete time case, for $\varepsilon > 0$ some small threshold. Theoretical properties of the spectral abscissa and the spectral radius have been studied in [7]. In general, before optimization can start, one has, indeed, to find a stabilizing $\mathbf{x}$. Using the method in [4], this can be achieved by an initial phase where $\alpha(A(\mathbf{x}))$ is minimized until an iterate $\mathbf{x}^1$ with $\alpha\left(A(\mathbf{x}^1)\right) \leqslant -\varepsilon$ is found.

10.1 Hankel feedback synthesis

We introduce an application of program (14) to a classical 1-DOF control system design, using an example from [5, Section 2.4]. The open-loop system

G, exogenous input w and regulated output z, are given by

$$G = \frac{10-s}{s^2(10+s)}, \quad w = \begin{bmatrix} d \\ n_y \\ r \end{bmatrix}, \quad z = \begin{bmatrix} y_p \\ u \end{bmatrix}.$$

The corresponding plant is

$$P : \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right],$$

where

$$\begin{aligned} A &= \begin{bmatrix} -10 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & B_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & B_2 &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ C_1 &= \begin{bmatrix} 0 & -1 & 10 \\ 0 & 0 & 0 \end{bmatrix} & & & D_{12} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ C_2 &= \begin{bmatrix} 0 & 1 & -10 \end{bmatrix} & D_{21} &= \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}. \end{aligned}$$

Inspired by a manually tuned controller

$$K_b = \frac{219.6s^2 + 1973.95s + 724.5}{s^3 + 19.15s^2 + 105.83s + 965.95},$$

proposed in [5, Section 2.4], we compute the optimal Hankel controller K_H with the same proposed structure and compare it to K_b and also to the optimal H_∞-controller K_∞ of that same structure

$$K(\mathbf{x}) = \frac{as^2 + bs + c}{s^3 + ms^2 + ns + p} = \left[\begin{array}{ccc|c} -m & -n & -p & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline a & b & c & 0 \end{array}\right],$$

where $\mathbf{x} = [m, n, p, a, b, c]^\top$ regroups the unknown tunable parameters. Using the Matlab function `hinfstruct` based on [1], we obtain

$$K_\infty = \frac{7941.9s^2 + 13028.4s + 3611.6}{s^3 + 3206.2s^2 + 12528.3s + 11078.3}.$$

The interest in this example is also to show that parametrizations $\mathbf{x}$ may arise naturally in the frequency domain. Note also that the closed-loop has no direct transmission term since $D_{11} = 0$ and K is strictly proper. To compute K_H, we solve (14) with the standard Hankel norm (1) and start Algorithm 1 at an initial stabilizing controller

$$\mathbf{x}^1 = [2.1460, 12.7448, 7.4208, 1.2271, 1.8013, 0.3517]^\top$$

with $f(\mathbf{x}^1) = 455.2874$, using the stability constraint $h(\mathbf{x}) = \alpha(A(\mathbf{x})) + \varepsilon \leqslant 0$ with a typical value $\varepsilon = 10^{-8}$. The stopping tests were (26) and (27). The algorithm came to a halt due to (26) and returned the optimal solution

$$\mathbf{x}^* = [77.0614, 255.2324, 74.6195, 188.0709, 133.9333, 22.2401]^\top$$

with $f(\mathbf{x}^*) = 10.8419$, meaning $\|T_{w\to z}(P, K_H)\|_H = 3.2927$ and

$$K_H := K(\mathbf{x}^*) = \frac{77.0614s^2 + 255.2324s + 74.6195}{s^3 + 188.0709s^2 + 133.9333s + 22.2401}.$$

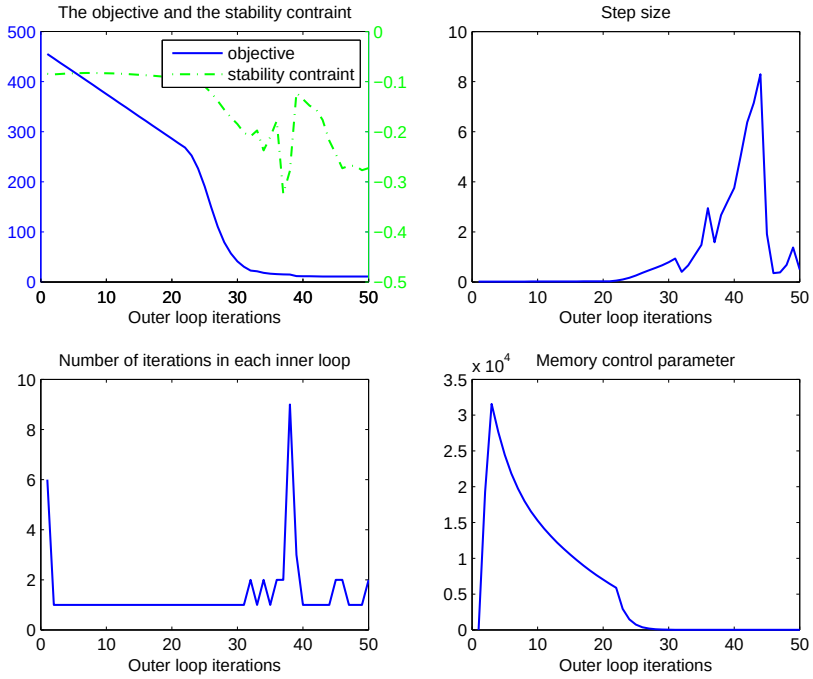


Fig. 2 Hankel feedback synthesis. Bearing of the algorithm. *Top left* shows $j \mapsto f(\mathbf{x}^j)$ and $j \mapsto \alpha(A(\mathbf{x}^j)) + 10^{-8}$. *Top right* shows $j \mapsto \|\mathbf{x}^j - \mathbf{x}^{j+1}\|$. *Lower left* shows $j \mapsto k_j$, *lower right* shows $j \mapsto \tau_j^\sharp$, the evolution of the memory control parameter at serious steps

The algorithm needed 50 serious iterates with 2.3 s CPU to reach the local minimum K_H. Bearing of the algorithm is shown in Fig. 2. The improvement of $\|T_{w\to z}(P, K_H)\|_H = 3.2927$ over $\|T_{w\to z}(P, K_\infty)\|_H = 3.3265$ is moderate, while the improvement over $\|T_{w\to z}(P, K_b)\|_H = 109.52$ is plain. Step responses and magnitude plots of the controllers K_b, K_H and K_∞ are shown in Fig. 3. Posterior testing displays ringing effects caused by various input signals w, including $w =$ unit step, white noise and sinc, shown in Fig. 4. As can be seen e.g., in Fig. 4, middle image, for a truncated white noise function $w_T = w\chi_{[0,T]}$, with $T = 3$, comparison of the responses $z_H = T_{w\to z}(K_H)w_T$ and $z_\infty = T_{w\to z}(K_\infty)w_T$, while confirming optimality $\|z_\infty\|_\infty = 0.5413 < \|z_H\|_\infty = 0.5498$, reveals that the bulk of energy in z_∞ has a wider spread over time, and $\|z_H\|_{2,[T,\infty)} = 1.1626 < \|z_\infty\|_{2,[T,\infty)} = 1.1878$ corroborating that the memory effects in K_H are reduced by the use of program (14).

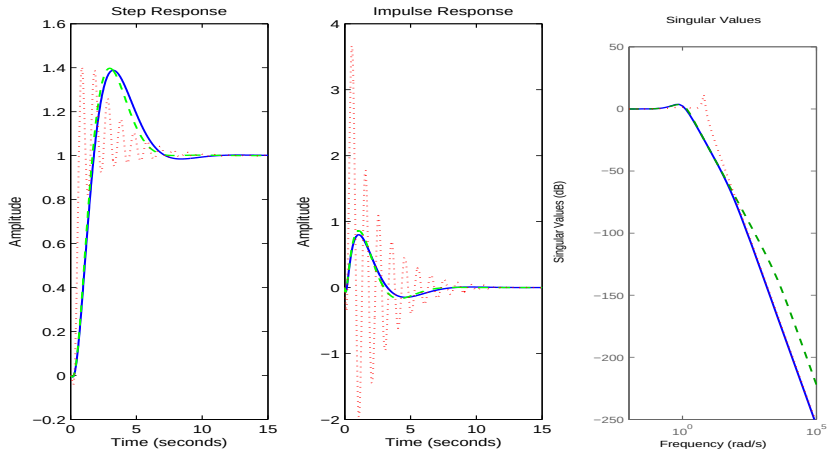


Fig. 3 Hankel feedback synthesis. Step responses (*left*), impulse responses (*middle*), magnitude plot (*right*) for controllers K_b (*dotted*), K_∞ (*dashed*), and K_H (*solid*)

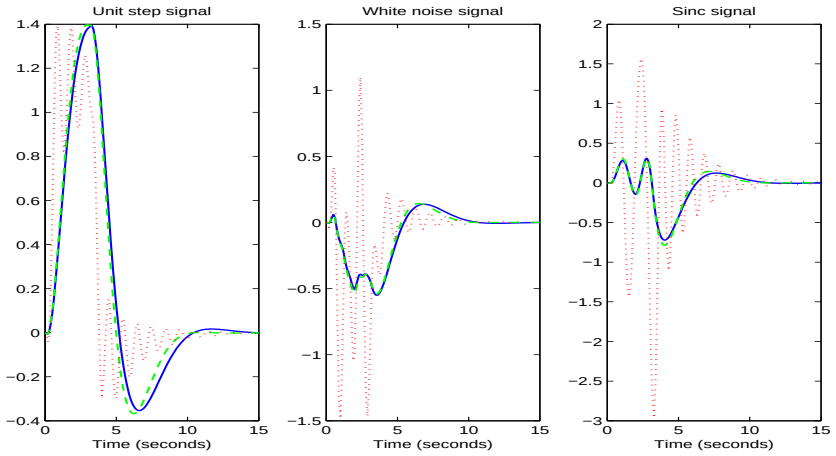


Fig. 4 Hankel feedback synthesis. Ringing for controllers K_b (*dotted*), K_∞ (*dashed*), and K_H (*solid*). Inputs: Unit step signal (*left*), white noise signal (*middle*), sinc signal (*right*)

10.2 Hankel system reduction

In this section, we solve program (15) with the usual Hankel norm, where our tests use the 15th order Rolls-Royce Spey gas turbine engine model described in [23, Chapter 11], with data available for download on I. Postlethwaites's homepage as `aero0.mat`. The goal of this study is to use the theoretical values to perform a blind testing of our algorithm. For $k = 1, 2, \ldots, 14$, using Algorithm 1, we computed Hankel reduced-order systems G_k of order k, and compared the achieved objective $f(\mathbf{x}^*) = \|G - G_k(\mathbf{x}^*)\|_H$ of (15) with the theoretically known optimal Hankel norm approximation errors $\|G - G_k\|_H = \sigma_{k+1}$,

the $(k+1)$st Hankel singular value of G. As can be seen in columns 2 and 3 of Table 1, this error is within the limits of numerical precision.

Table 1 Hankel system reduction. Comparison of optimal values $\|G - G_k(\mathbf{x}^*)\|_H$ with theoretical values σ_{k+1}

k	σ_{k+1}	$\|G - G_{\text{red}}\|_H$	No of iterations	Time
1	4.046418	4.046418	26	3.5
2	2.754623	2.754624	71	21.0
3	1.763527	1.763529	124	47.3
4	1.296531	1.299542	151	101.5
5	0.629640	0.629640	88	118.0
6	0.166886	0.166887	183	197.3
7	0.093407	0.093408	93	185.8
8	0.022193	0.022201	76	132.4
9	0.015669	0.015675	162	203.7
10	0.013621	0.013624	175	191.3
11	0.003997	0.003997	140	380.0
12	0.001179	0.001179	57	488.4
13	0.000324	0.000324	24	224.2
14	0.000033	0.000033	68	372.5

In each run, the algorithm was started from a random initial guess, and no information as to the specific structure of problem (15) was provided. On average, the algorithm needed about 103 serious steps to reach the optimal objective function value within a tolerance of $< 10^{-10}$. See Table 1 for number of iterations and running times in seconds.

Remark 7 The results show no clear relation between running times and the order of the reduced system, as one might have expected. This is due to the fact that local optimization techniques depend very sensibly on the initial guess, which in this comparison was chosen randomly.

Remark 8 In [9], we have used the same example to give a comparison between Hankel system reduction and H_∞-system reduction, which is compared to the H_∞-bound (see [12]).

10.3 Maximizing the memory of a system

We use here an illustrative example for (18), where G and G_{ref} are defined as

$$G(s) = \frac{1}{s-1}, \quad G_{\text{ref}} = \frac{11.11}{s^2 + 6s + 11.11}.$$

The filter F is chosen of order 2,

$$F(s) = \frac{as^2 + bs + c}{s^2 + ds + e} = \left[\begin{array}{cc|c} -d & -e & 1 \\ 1 & 0 & 0 \\ \hline b - ad & c - ae & a \end{array}\right],$$

which leads to 5 tunable parameters, whereas K is a PID

$$K(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{T_f s + 1} = \left[\begin{array}{cc|c} 0 & 0 & k_i \\ 0 & -\frac{1}{T_f} & -\frac{k_d}{T_f^2} \\ \hline 1 & 1 & k_p + \frac{k_d}{T_f} \end{array}\right],$$

adding another 4 unknowns. We have added a low-pass filter $W_1(s) = \frac{0.25s+0.6}{s+0.006}$ to the output z_1 to asses the tracking error $y - y_{\text{ref}}$ in low-frequency, and a high-pass filter $W_2(s) = \frac{s}{s+0.001}$ on the control output z_2 to reduce high-frequency components of the control signal $u + v$.

Due to the choice of the performance channel $w \to z = (W_1 z_1, W_2 z_2)$, the closed-loop has a non-vanishing direct transmission term. We therefore solve problem (14) for the setup (18) using the extended Hankel program (2) with (10), and also using the constraint program (12). Running Algorithm 1 from the same starting point, these two methods give Hankel controllers (F_{eH}, K_{eH}) and (F_{cH}, K_{cH}) with

$$F_{eH}(s) = \tfrac{-3.4778s^2-13.9996s-0.0546}{s^2+1.9202s+0.0001},\ K_{eH}(s) = 6.3078 + \tfrac{3.6689}{s} - \tfrac{1.0924}{0.4739s+1},$$
$$F_{cH}(s) = \tfrac{-3.6552s^2-13.6987s-0.0522}{s^2+1.9588s+0.0001},\ K_{cH}(s) = 6.1959 + \tfrac{3.8435}{s} - \tfrac{0.7121}{0.3644s+1},$$

where we used the constraint $\sigma_1(D) \leqslant \eta$ with $\eta = 1$. For comparison, we also synthesized the usual Hankel norm controller, where the direct transmission is ignored, and the H_∞-controller, both with the same architecture:

$$F_H(s) = \tfrac{-2.2376s^2-1.9738s-2.4161}{s^2+0.9054s+0.9836},\ K_H(s) = 2.4482 + \tfrac{0.7883}{s} + \tfrac{0.8023}{0.7817s+1},$$
$$F_\infty(s) = \tfrac{-9.9366s^2-1.5077s-0.0349}{s^2+0.9969s+0.0273},\ K_\infty(s) = 11.5131 + \tfrac{0.2673}{s} - \tfrac{0.5507}{1.0117s+1}.$$

Figure 5 compares step responses y and step reference responses y_{ref} for these controllers. The evolution of the optimization method for the three Hankel controllers can be traced in Fig. 6. The achieved Hankel norms are

$$\begin{aligned}\|T_{w\to z}(F_{eH}, K_{eH})\|_H = 0.8767 &< \|T_{w\to z}(F_{cH}, K_{cH})\|_H = 0.8862 \\ &< \|T_{w\to z}(F_H, K_H)\|_H = 1.0160 < \|T_{w\to z}(F_\infty, K_\infty)\|_H = 1.0277.\end{aligned}$$

This example is again interesting in so far as the parametrization of F and K arises naturally in the frequency domain.

10.4 Control of flow in a graph

Here, we give an application of program (21). Let $\mathscr{V}_{\text{stay}} = \{1, 2, \ldots, n_x\}$, $\mathscr{V}_{\text{in}} = \{1, 2, \ldots, m\}$, $\mathscr{V}_{\text{out}} = \{1, 2, \ldots, p\}$. Let $\mathbf{x}$ regroup the unknown tunable parameters $a_{jj'}, b_{ij}$ and set $A(\mathbf{x}) = [a_{jj'}]_{n_x \times n_x}^\top, B(\mathbf{x}) = [b_{ij}]_{m \times n_x}^\top, C = [c_1, \ldots, c_{n_x}]$, where $a_{jj'} = 0$ if $(j, j') \notin \mathscr{A}, b_{ij} = 0$ if $(i, j) \notin \mathscr{A}$. We have a discrete linear time-invariant system

$$G(\mathbf{x}) : \begin{cases} x(t+1) = A(\mathbf{x})x(t) + B(\mathbf{x})w(t) \\ z(t) \quad\quad\ = Cx(t). \end{cases}$$

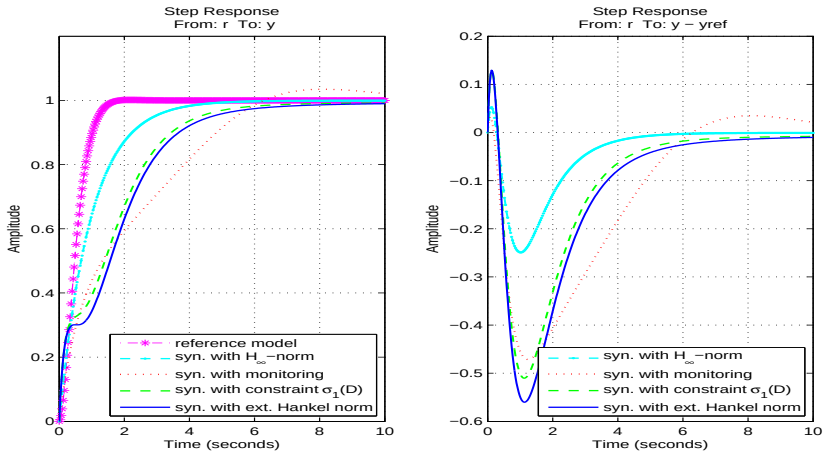


Fig. 5 Maximizing memory. Comparison between step responses y and y_{ref} for H_∞-controller and Hankel controllers computed by programs (2) with monitoring (*dotted*), (12) (*dashed*) and (2) with (10) (*solid*)

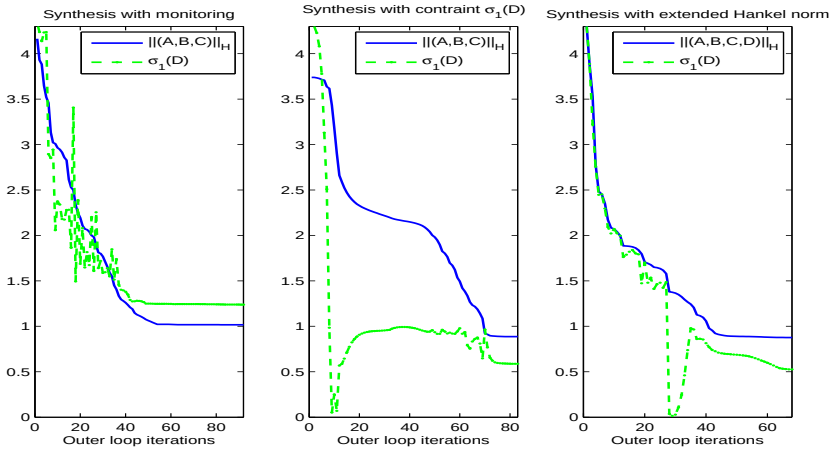
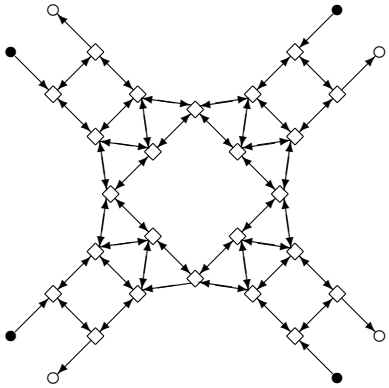


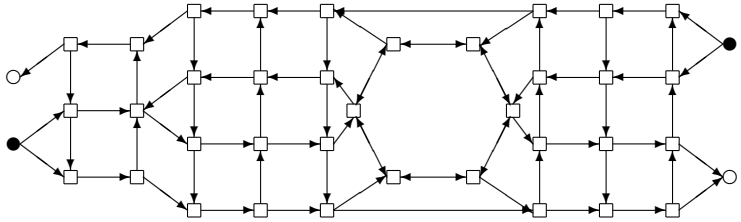
Fig. 6 Maximizing memory. Comparison between standard Hankel program (2) with monitoring (*left*), constraint program (12) (*middle*), and extended Hankel program (2) with (10) (*right*). While (2) with (10) and (12) give comparable results, minimization of $\|(A, B, C)\|_H$ alone (*left*) gives a large direct transmission

Remark that the linear constraint conditions in (21) can be transferred to the form $A_{\text{eq}}\mathbf{x} = b_{\text{eq}}, \mathbf{x} \geqslant 0$, which are added in each trial step generation of Algorithm 1.

We now take the following graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ with $n_x = 24, m = 4$ and $p = 4$.



Let $z(t)$ be the total number of people on the fairground, which corresponds to the weights $c_1 = \cdots = c_{n_x} = 1$. We start Algorithm 1 at the uniform distribution $\mathbf{x}^1$, where $f(\mathbf{x}^1) = 714.8634$, and $\|G(\mathbf{x}^1)\|_H = 26.7369$. After 2469 serious iterates with 8768 s CPU, our algorithm returns the optimal $\mathbf{x}^*$ with $f(\mathbf{x}^*) = 8.6056$, meaning $\|G(\mathbf{x}^*)\|_H = 2.9335$. For comparison, with the Matlab function `fmincon` started at $\mathbf{x}^1$, we obtain $\mathbf{x}^\dagger$ with $f(\mathbf{x}^\dagger) = 12.5994 > f(\mathbf{x}^*) = 8.6056$. However, if we take $\mathbf{x}^\dagger$ as initial for Algorithm 1, the result is $f(\mathbf{x}^*) = 8.6056$, meaning $\|G(\mathbf{x}^*)\|_H = 2.9335$, which is achieved very fast (29 serious iterates, 87 s CPU).



We next consider an example using the second graph with $n_x = 36, m = 2$ and $p = 2$. Let $z(t)$ quantify the number of people on the fairground, where the 6 central nodes are counted twice. In this example, we will directly compare our nonsmooth method to the heuristic in Sect. 9. Optimization starts again at the uniform distribution $\mathbf{x}^1$. Minimizing smooth function $f_\mu(\mathbf{x})$ in (24) with initial $\mathbf{x}^1$ leads to $\mathbf{x}^\dagger$, where $f(\mathbf{x}^\dagger) = 21.7291, \|G(\mathbf{x}^\dagger)\|_H = 4.6614$, while $f(\mathbf{x}^1) = 578.6875, \|G(\mathbf{x}^1)\|_H = 24.0559$. We now use $\mathbf{x}^\dagger$ to initialize the nonsmooth Algorithm 1. After 44 serious steps with 168 s CPU, our algorithm returns the optimal $\mathbf{x}^*$ with $f(\mathbf{x}^*) = 14.8353$, meaning $\|G(\mathbf{x}^*)\|_H = 3.8517$.

For the two displayed graphs, Figs. 7 and 8 compare ringing effects in unit step and white noise responses truncated at $T = 30$ for the three systems $G(\mathbf{x}^1), G(\mathbf{x}^\dagger)$ and $G(\mathbf{x}^*)$. We can see that ringing for $G(\mathbf{x}^\dagger)$ and $G(\mathbf{x}^*)$ is substantially reduced.

Tables 2 and 3 show a simulated study, where we compare the effects of the transition probability distributions $\mathbf{x}^1, \mathbf{x}^\dagger, \mathbf{x}^*$ by recording the evacuation of people from the fairground. We simulate crowd entering through the gates $1, \ldots, 4$ for different scenarios w. We then close the entrance gates at time $T = 15$, when in the first study 6994 people have entered the ground, and

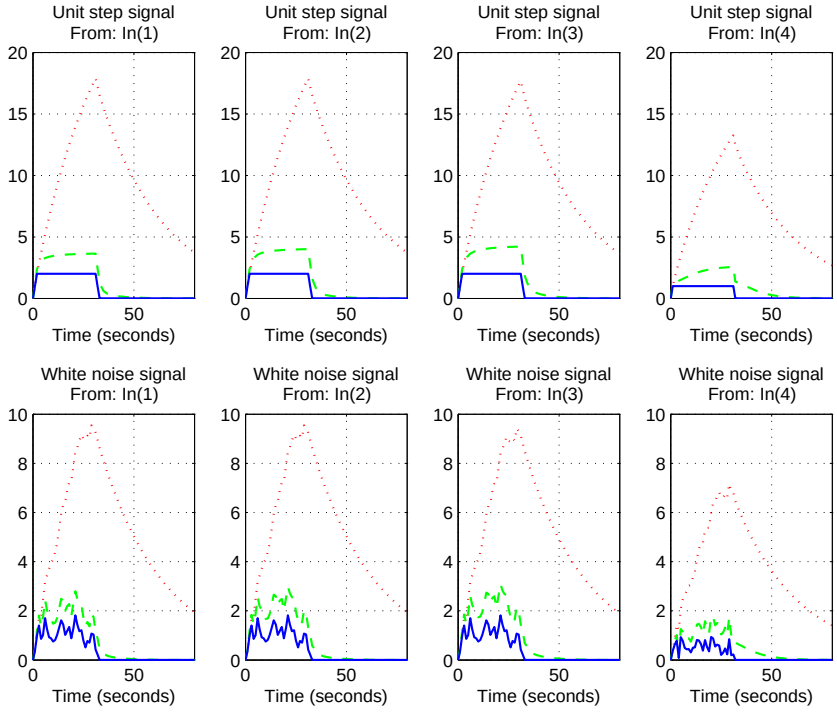


Fig. 7 Ringing effects of three systems $G(\mathbf{x}^1)$ (*dotted*), $G(\mathbf{x}^\dagger)$ (*dashed*) and $G(\mathbf{x}^*)$ (*solid*) for the first graph. Input: Unit step signal (*top*) and white noise signal (*bottom*)

Table 2 First graph, three distributions $\mathbf{x}^1$, $\mathbf{x}^\dagger$, $\mathbf{x}^*$. Times when 90% of crowd in fairground has been evacuated

Input signal	People	$z^1(T)$	$G(\mathbf{x}^1)$	$z^\dagger(T)$	$G(\mathbf{x}^\dagger)$	$z^*(T)$	$G(\mathbf{x}^*)$
	Entering	Remain	Evac. time	Remain	Evac. time	Remain	Evac. time
$[w_1; w_2; w_3; 0]$	6994	4680	78	1478	18	1141	17
$[w_1; w_2; 0; w_3]$	6994	4375	75	1293	18	941	17
$[w_1; 0; w_2; w_3]$	6994	4367	75	1306	18	941	17
$[0; w_1; w_2; w_3]$	6994	4367	75	1374	18	941	17

Entry gates are closed at $T = 15$

Table 3 Second graph, three distributions. Times when 90% of crowd in the fairground has been evacuated

Input signal	People	$z^1(T)$	$G(\mathbf{x}^1)$	$z^\dagger(T)$	$G(\mathbf{x}^\dagger)$	$z^*(T)$	$G(\mathbf{x}^*)$
	Entering	Remain	Evac. time	Remain	Evac. time	Remain	Evac. time
$[w_1; w_2]$	4994	3794	63	1530	20	1216	19
$[w_1; w_3]$	5200	3901	63	1546	20	1227	19
$[w_2; w_3]$	3794	2704	63	1034	20	804	20

Entry gates are closed at $T = 15$

record the time which passes until 90% of the crowd has been evacuated. In

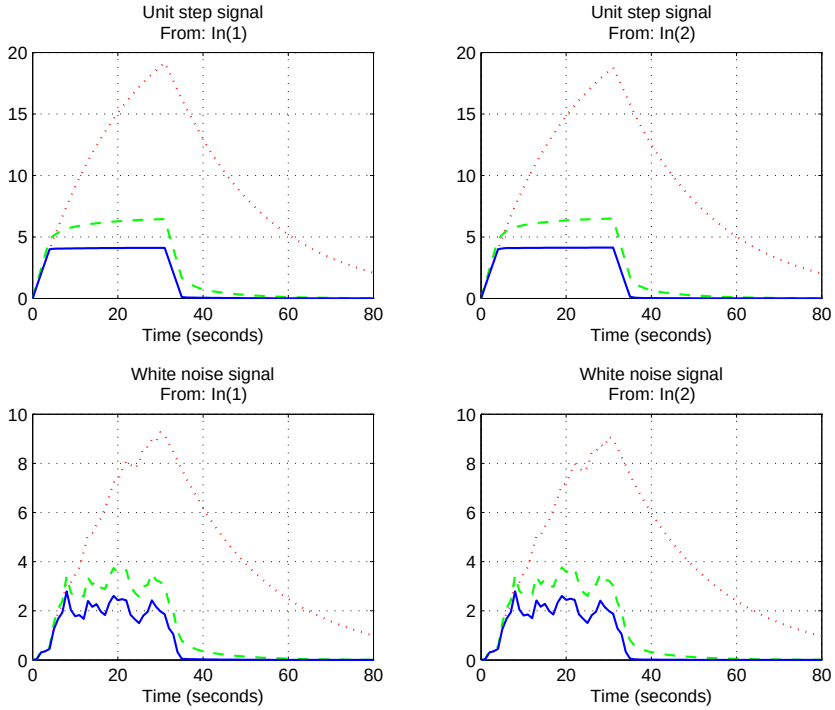


Fig. 8 Ringing effects of three systems $G(\mathbf{x}^1)$ (*dotted*), $G(\mathbf{x}^\dagger)$ (*dashed*) and $G(\mathbf{x}^*)$ (*solid*) for the second graph. Input: Unit step signal (*top*) and white noise signal (*bottom*)

our tests w_1 is a step signal, w_2 is a sine wave, and w_3 is a square wave. A similar approach is chosen in the second graph.

Column $z^1(T)$ gives the number of people still present on the fairground at time T when distribution $\mathbf{x}^1$ is used, and column $G(\mathbf{x}^1)$ gives the time which then elapses until this crowd is reduced below 10% of the total number 6994. Columns 5–8 are analogous. As compared to $\mathbf{x}^1$, the optimal strategy $\mathbf{x}^*$ reduces the evacuation time to close to 1/5 in the first graph, and to close to 1/3 in the second graph.

11 Conclusion

We have proposed a new methodology to reduce unwanted ringing effects in a tunable linear time-invariant system. The problem was addressed by minimizing the Hankel norm of the system, a problem which leads to an eigenvalue optimization program for the associated Hankel operator. A proximal bundle algorithm was presented to solve a variety of test problems successfully, and a smooth heuristic, based on work of Nesterov [15], was added and used to initialize the algorithm with a favorable initial seed.

Acknowledgements The authors acknowledge helpful discussions with Dr. Armin Rainer (University of Vienna).

References

1. Apkarian P, Noll D (2006) Nonsmooth H_∞ synthesis. IEEE Trans Automat Control 51(1):71–86
2. Apkarian P, Noll D, Rondepierre A (2008) Mixed H_2/H_∞ control via nonsmooth optimization. SIAM J Control Optim 47(3):1516–1546
3. Bellman R (1959) Kronecker products and the second method of Lyapunov. Math Nachr 20: 17–19
4. Bompart V, Apkarian P, Noll D (2007) Non-smooth techniques for stabilizing linear systems. In: Proceedings of the American Control Conference, New York, pp 1245–1250
5. Boyd S, Barratt C (1991) Linear controller design: limits of performance. Prentice Hall, New York
6. Bronshtein MD (1979) Smoothness of roots of polynomials depending on parameters. Sibirsk Mat Zh 20(3): 493–501. English Transl. in (1980) Siberian Math J, vol 20, pp 347–352
7. Burke JV, Overton ML (1994) Differential properties of the spectral abscissa and the spectral radius for analytic matrix-valued mappings. Nonlinear Anal 23(4):467–488
8. Clarke FH (1983) Optimization and nonsmooth analysis. John Wiley & Sons, Inc., New York
9. Dao MN, Noll D (2013) Minimizing the memory of a system. In: Proceedings of the Asian Control Conference, Istanbul
10. Dennis JE Jr, Schnabel RB (1983) Numerical methods for unconstrained optimization and nonlinear equations. Prentice Hall, New Jersey
11. Gabarrou M, Alazard D, Noll D (2013) Design of a flight control architecture using a non-convex bundle method. Math Control Signals Syst 25(2):257–290
12. Glover K (1984) All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. Int J Control 39(6):1115–1193
13. Mangasarian OL (1969) Nonlinear programming. McGraw-Hill Book Co., New York-London-Sydney
14. Mifflin R (1982) A modification and extension of Lemaréchal's algorithm for nonsmooth minimization. Nondifferential and variational techniques in optimization (Lexington, Ky., 1980). Math Programming Stud 17:77–90
15. Nesterov Y (2007) Smoothing technique and its applications in semidefinite optimization. Math Program, Ser A 110(2):245–259
16. Noll D (2010) Cutting plane oracles to minimize non-smooth non-convex functions. Set-Valued Var Anal 18(3-4):531–568
17. Noll D, Prot O, Rondepierre A (2008) A proximity control algorithm to minimize nonsmooth and nonconvex functions. Pac J Optim 4(3):571–604
18. Overton ML (1992) Large-scale optimization of eigenvalues. SIAM J Optim 2(1):88–120
19. Parusiński A, Rainer A (2014) A new proof of Bronshtein's theorem. arXiv:1309.2150v2
20. Polak E (1997) Optimization: algorithms and consistent approximations. Applied Mathematical Sciences 124. Springer-Verlag, New York
21. Rainer A (2011) Smooth roots of hyperbolic polynomials with definable coefficients. Israel J Math 184: 157–182
22. Rockafellar RT, Wets RJ-B (1998) Variational analysis. Springer-Verlag, Berlin
23. Skogestad S, Postlethwaite I (2005) Multivariable feedback control: analysis and design. John Wiley & Sons, Chichester
24. Spingarn JE (1981) Submonotone subdifferentials of Lipschitz functions. Trans Amer Math Soc 264(1):77–89
25. van den Dries L (1998) Tame topology and o-minimal structures. London Math Soc Lecture Note Ser 248. Cambridge University Press, Cambridge
26. Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, New Jersey