Minimizing the memory of a system

Ngoc Minh DAO * and Dominikus NOLL *
* Université de Toulouse, Institut de Mathématiques, Toulouse, France

***Abstract*—Consider a stable linear time-invariant system $G(\mathbf{x})$ with tunable parameters $\mathbf{x} \in \mathbb{R}^n$, which maps inputs $w \in L^2(\mathbb{R}^m)$ to outputs $z \in L^2(\mathbb{R}^p)$. Our goal is to find a choice of the tunable parameters $\mathbf{x}^*$ which avoids undesirable responses of the system to past excitations known as system ringing. We address this problem by minimizing the Hankel norm $\|G(\mathbf{x})\|_H$ of $G(\mathbf{x})$, which quantifies the influence of past inputs on future outputs.**

I. Introduction

Ringing generally designates undesired responses of a system to a past excitation. 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.

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 transfer function, which measures the effect of past inputs on future outputs.

Consider a stable LTI system

$$G : \begin{cases} \dot{x} = Ax + Bw \\ z = Cx \end{cases} \tag{1}$$

with state $x \in \mathbb{R}^{n_x}$, input $w \in \mathbb{R}^m$, and output $z \in \mathbb{R}^p$. If we think of $w(t)$ as an excitation at the input which acts over the time period $0 \leq t \leq T$, 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

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

Here our interest is in systems (1) with tunable parameters $\mathbf{x} \in \mathbb{R}^n$, that is, systems of the form

$$G(\mathbf{x}) : \begin{cases} \dot{x} = A(\mathbf{x})x + B(\mathbf{x})w \\ z = C(\mathbf{x})x \end{cases} \tag{3}$$

where $A(\mathbf{x})$, $B(\mathbf{x})$, $C(\mathbf{x})$ 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. This leads to the Hankel norm optimization program

$$\begin{array}{ll} \text{minimize} & \|G(\mathbf{x})\|_H \\ \text{subject to} & G(\mathbf{x}) \text{ internally stable} \\ & \mathbf{x} \in \mathbb{R}^n. \end{array} \tag{4}$$

We will discuss instances, where program (4) may be of interest. Then we present an algorithm to solve (4) based on techniques from eigenvalue optimization. We present a non-smooth optimization method, and a smooth relaxation based on work of Y. Nesterov [7]. Applications are then presented in the final sections.

II. Hankel feedback synthesis

A first instance of (4) is Hankel feedback synthesis. Consider an LTI plant $P(s)$ in standard form

$$P : \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}, \tag{5}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{m_2}$ the control input, $w \in \mathbb{R}^{m_1}$ the exogenous input, $y \in \mathbb{R}^{p_2}$ the measured output, and $z \in \mathbb{R}^{p_1}$ the regulated output,

$$P(s) = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (sI - A)^{-1} \begin{bmatrix} B_1 & B_2 \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & 0 \end{bmatrix}.$$

Let $u(s) = K(s)y(s)$ be a dynamic output feedback control law for (5). Then, the closed-loop transfer function of the performance channel channel $w \to z$ is obtained as

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

Substituting $u = Ky$ into (5), we get

$$T_{w \to z}(K) : \begin{bmatrix} \dot{x} \\ z \end{bmatrix} = \begin{bmatrix} \mathcal{A}(K) & \mathcal{B}(K) \\ \mathcal{C}(K) & \mathcal{D}(K) \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix}, \tag{6}$$

where

$$\begin{aligned} \mathcal{A}(K) &= \begin{bmatrix} A + B_2 D_K C_2 & B_2 C_K \\ B_K C_2 & A_K \end{bmatrix}, \\ \mathcal{B}(K) &= \begin{bmatrix} B_1 + B_2 K_K D_{21} \\ B_K D_{21} \end{bmatrix}, \\ \mathcal{C}(K) &:= \left[C_1 + D_{12} D_K C_2 \;\; D_{12} C_K \right], \\ \mathcal{D}(K) &:= D_{11} + D_{12} D_K D_{21}. \end{aligned}$$

We assume that $\mathcal{D}(K) = 0$, which can be arranged e.g. by standard assumption, e.g., $D_{11} = 0$ and either $D_{21} = 0$ or $D_{12} = 0$ or K strictly proper.

Minimizing the effects of past inputs on future outputs by way of an appropriate feedback may now be cast as the optimization program

$$\begin{array}{ll} \text{minimize} & \|T_{w\to z}(K)\|_H \\ \text{subject to} & K \text{ stabilizes (5)} \\ & K = K(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n. \end{array} \tag{7}$$

Here $K(\mathbf{x})$ refers to a controller which is structured in the sense of [1], and $\mathbf{x}$ stands for the tunable parameters of $K(\mathbf{x})$. Typical examples of structured controllers include PIDs

$$K_{\text{pid}}(\mathbf{x}) = \left[\begin{array}{cc|c} 0 & 0 & r_i \\ 0 & -\tau & r_d \\ \hline 1 & 1 & d_K \end{array}\right],$$

where $\mathbf{x}$ regroups the parameters r_i, r_d, d_K, τ, or observer-based controllers, decentralized, fixed reduced order controllers, and more generally, control architectures combining basic building blocks like PIDs with filters, feed forward blocks, and much else (see [1], [5]). It is convenient to write $\mathcal{A}(\mathbf{x}) = \mathcal{A}(K(\mathbf{x}))$, etc., and $T_{w\to z}(\mathbf{x}) = T_{w\to z}(K(\mathbf{x}))$.

III. System reduction

System reduction is the most widely known application of Hankel norm optimization (4). It may be considered a special case of Hankel controller synthesis (7). Given a stable system

$$G : \begin{cases} \dot{x} = Ax + Bw \\ z = Cx + Dw \end{cases} \tag{8}$$

of order n, we wish to find a stable system

$$G_{\text{red}} : \begin{cases} \dot{x} = A_{\text{red}}x + B_{\text{red}}w \\ z = C_{\text{red}}x + Dw \end{cases} \tag{9}$$

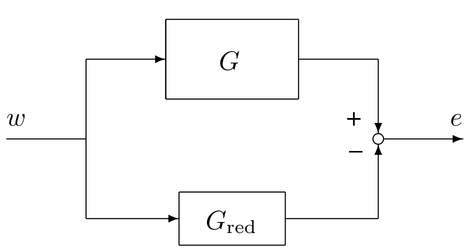
of reduced order $n_{\text{red}} < n$ with input-output behavior as close as possible to the original system G. If the model matching error $e = (G - G_{\text{red}})w$ is measured in the Hankel norm

$$\begin{array}{ll} \text{minimize} & \|G - G_{\text{red}}\|_H \\ \text{subject to} & G - G_{\text{red}} \text{ internally stable} \\ & \mathbf{x} = (A_{\text{red}}, B_{\text{red}}, C_{\text{red}}), \end{array} \tag{10}$$

then (10) is a special case of (7), where we define plant and controller as

$$P : \left[\begin{array}{c|cc} A & B & 0 \\ \hline C & D & -I \\ 0 & I & 0 \end{array}\right] \qquad K : \left[\begin{array}{c|c} A_{\text{red}} & B_{\text{red}} \\ \hline C_{\text{red}} & D \end{array}\right], \tag{11}$$

the tunable parameters $\mathbf{x}$ being the entries of A_{red}, B_{red}, C_{red}.



(12)

Glover [6] has shown how to compute an explicit solution to (10) when the matrices A_{red}, B_{red}, C_{red} are not further restricted to have specific pattern. Here we use this as a blind test of algorithm 2 by applying it to problem (10).

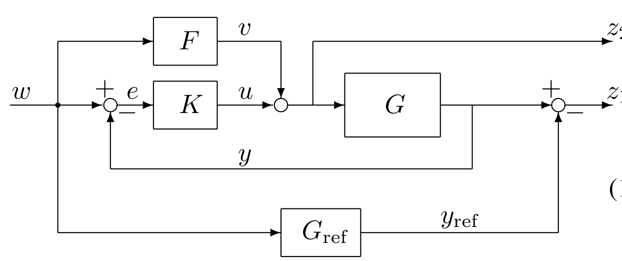
Recall that the use of the Hankel norm in system reduction (10), (12) is to some extent artificial and mainly motivated by the fact that it leads to a linear algebra solution. The more natural approach would be H_∞-system reduction

$$\begin{array}{ll} \text{minimize} & \|G - G_{\text{red}}\|_\infty \\ \text{subject to} & G - G_{\text{red}} \text{ internally stable} \\ & \mathbf{x} = (A_{\text{red}}, B_{\text{red}}, C_{\text{red}}, D_{\text{red}}), \end{array} \tag{13}$$

which is usually considered too demanding algorithmically. This has changed with the solution of the structured H_∞-control problem in [1]. The tool `hinfstruct` developed since [1], [2], [8] and made available through [14] since 2010, allows to perform H_∞-system reduction as a special case of structured H_∞-synthesis using (11). Solving (13) by a nonsmooth optimization method was first proposed in [1], and in section IX we will present tests related to (10) and (13).

IV. Maximizing the memory of a system

Within the present framework it is also possible to maximize the memory of a system G via feedback if a reference system G_{ref} with desirable memory properties is used. As an example we consider a 2-DOF synthesis scheme of the following form:



(14)

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. In other words, 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 $\|z_2\|_H \leqslant \gamma_H$ or $\|z_2\|_\infty \leqslant \gamma_\infty$, where $z_2 = u+v$, in order to avoid exceedingly large controller actions. If $\|z_2\|_H \leqslant \gamma_H$ is used, this problem can be cast as a special case of (7), where plant and decentralized controller structure are:

$$P:\left[\begin{array}{cc|ccc} A & 0 & 0 & B & B \\ 0 & A_{\rm ref} & B_{\rm ref} & 0 & 0 \\ \hline C & -C_{\rm ref} & -D_{\rm ref} & D & D \\ 0 & 0 & 0 & I & I \\ -C & 0 & I & -D & -D \\ 0 & 0 & I & 0 & 0 \end{array}\right],$$

$$K:\left[\begin{array}{cc|cc} A_F & 0 & B_F & 0 \\ 0 & A_K & 0 & B_K \\ \hline C_F & 0 & D_F & 0 \\ 0 & C_K & 0 & D_K \end{array}\right].$$

Notice that

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

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

V. Hankel operator

As is well known, a representation of the Hankel norm $\|G\|_H$ amenable to computations, is obtained through the observability and controllability Gramians. Using $x(-\infty)=0$, the solution of (1) satisfies

$$y(t)=\int_{-\infty}^{t} Ce^{A(t-\tau)}Bu(\tau)d\tau,$$

and if we focus on input signals u_- that live for times $t<0$ and vanish for $t\geqslant 0$, then the output restricted to $t\geqslant 0$ is

$$y_+(t)=\int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)d\tau,\ t\geqslant 0.$$

Now the Hankel operator $\Gamma_G : L^2(-\infty,0] \longrightarrow L^2[0,\infty)$, defined by

$$(\Gamma_G u)(t)=\int_{-\infty}^{0} Ce^{A(t-\tau)}Bu(\tau)d\tau,\ t\geqslant 0,$$

maps past inputs u_- to future outputs $y_+=\Gamma_G u_-$. Using

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

it follows that

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

Let X and Y be the controllability and observability Gramians of the system, i.e.

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

Suppose that σ is a nonzero singular value of Γ_G and u is a eigenvector corresponding to the eigenvalue σ^2 of $\Gamma_G^*\Gamma_G$, i.e. $\Gamma_G^*\Gamma_G u=\sigma^2 u$. Set $y(t)=(\Gamma_G u)(t)=Ce^{At}x_0$ with

$$x_0=\int_{-\infty}^0 e^{-A\tau}Bu(\tau)d\tau.$$

Then

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

Therefore,

$$\begin{aligned}\sigma^2 x_0 &= \int_{-\infty}^0 e^{-A\tau}B\sigma^2 u(\tau)d\tau\\ &= \int_{-\infty}^0 e^{-A\tau}BB^\top e^{-A^\top\tau}Yx_0 d\tau = XYx_0.\end{aligned}$$

We see that $x_0\neq 0$ since otherwise $\sigma^2 u=0$ 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 $u=B^\top e^{-A\tau}Yx_0$ we obtain $u\neq 0$ and $\Gamma_G^*\Gamma_G u=\sigma^2 u$. Hence,

$$\sigma_i^2(\Gamma_G)=\lambda_i(XY),$$

where $\sigma_i(T)$ denotes the ith singular value of T and $\lambda_i(A)$ denotes the ith eigenvalue of A. In particular, since $\Gamma_G^*\Gamma_G$ is self-adjoint,

$$\|\Gamma_G\|=\sup_{0\neq u_-\in L^2(-\infty,0]}\frac{\|y_+\|_2}{\|u_-\|_2}=\sigma_1(\Gamma_G)=\sqrt{\lambda_1(XY)}. \quad (15)$$

This agrees with the definition (2) of the Hankel norm $\|G\|_H$ of the stable LTI system G if we truncate the input signal at $-T<0$, introduce a supremum over T, and use time-invariance to move to the right.

VI. Clarke subdifferentials of the Hankel norm in closed-loop

In this section we prepare our non smooth optimization approach by computing Clarke subgradients of the closed-loop objective function

$$f(\mathbf{x})=\|T_{w\to z}(\mathbf{x})\|_H^2=\lambda_1(X(\mathbf{x})Y(\mathbf{x}))$$

of (4). Here λ_1 denotes the maximum eigenvalue of a symmetric or Hermitian matrix, and $X(\mathbf{x})$ and $Y(\mathbf{x})$ are the controllability and observability Gramians that can be obtained from the Lyapunov equations

$$\mathcal{A}(\mathbf{x})X+X\mathcal{A}^\top(\mathbf{x})+\mathcal{B}(\mathbf{x})\mathcal{B}^\top(\mathbf{x})=0, \quad (16)$$

$$\mathcal{A}^\top(\mathbf{x})Y+Y\mathcal{A}(\mathbf{x})+\mathcal{C}^\top(\mathbf{x})\mathcal{C}(\mathbf{x})=0, \quad (17)$$

with $\mathcal{A}(\mathbf{x})$, etc. denoting closed loop data (6). Notice that despite the symmetry of X and Y the product XY need not be symmetric, but stability of $\mathcal{A}(\mathbf{x})$ in closed-loop guarantees $X \succ 0$, $Y \succ 0$ in (16), (17), so that we can write

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

which brings us back in the realm of eigenvalue theory of symmetric matrices.

Using the spectral abscissa $\alpha(A) = \max\{\mathrm{Re}(\lambda) : \lambda \text{ eigenvalue of } A\}$ of a square matrix A, we can replace program (7) by the following program

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) := \|T_{w\to z}(\mathbf{x})\|_H^2 \\ \text{subject to} & c(\mathbf{x}) := \alpha(\mathcal{A}(\mathbf{x})) + \varepsilon \leqslant 0 \end{array} \tag{18}$$

for some fixed small $\varepsilon > 0$. We have $X(\mathbf{x}) \succ 0$ and $Y(\mathbf{x}) \succ 0$ on the feasible set $\mathcal{C} = \{\mathbf{x} : c(\mathbf{x}) \leqslant 0\}$, so that f is well-defined and locally Lipschitz on $\mathcal{C}$.

Let $\mathbb{M}_{n,m}$ be the space of $n \times m$ matrices, equipped with the corresponding scalar product $\langle X, Y\rangle = \mathrm{Tr}(X^\top Y)$. The space of $m \times m$ symmetric matrices is denoted $\mathbb{S}_m$. We define

$$\mathbb{B}_m := \{X \in \mathbb{S}_m : X \text{ is positive semidefinite}, \mathrm{Tr}(X) = 1\}.$$

Set $Z := X^{\frac{1}{2}}YX^{\frac{1}{2}}, Z_i(\mathbf{x}) := \frac{\partial Z(\mathbf{x})}{\partial \mathbf{x}_i}, i = 1, \ldots, n$ and pick Q to be a matrix whose columns form an orthonormal basis of the t-dimensional eigenspace associated with $\lambda_1(Z)$. By [9, Theorem 3], the Clarke subdifferential of f at $\mathbf{x}$ is the set

$$\partial f(\mathbf{x}) = \{(\mathrm{Tr}(QUQ^\top Z_1(\mathbf{x})), \ldots, \mathrm{Tr}(QUQ^\top Z_n(\mathbf{x}))) : U \in \mathbb{B}_t\}.$$

It now remains to compute $Z_i(\mathbf{x})$. We have

$$Z_i(\mathbf{x}) = D_K Z(\mathbf{x}) K_i(\mathbf{x}) = \varphi_i Y X^{\frac{1}{2}} + X^{\frac{1}{2}}\psi_i X^{\frac{1}{2}} + X^{\frac{1}{2}} Y \varphi_i, \tag{19}$$

where $K_i(\mathbf{x}) := \frac{\partial K(\mathbf{x})}{\partial \mathbf{x}_i}, \varphi_i := D_K X^{\frac{1}{2}} K_i(\mathbf{x}), \psi_i := D_K Y K_i(\mathbf{x})$. From (16) and (17), and putting $\phi_i := D_K X K_i(\mathbf{x})$, we obtain

$$\begin{aligned} \mathcal{A}\phi_i + \phi_i\mathcal{A}^\top &= -B_2K_i(\mathbf{x})C_2X - X(B_2K_i(\mathbf{x})C_2)^\top \\ &\quad - B_2K_i(\mathbf{x})D_{21}\mathcal{B}^\top - \mathcal{B}(B_2K_i(\mathbf{x})D_{21})^\top \end{aligned} \tag{20}$$

$$\begin{aligned} \mathcal{A}^\top\psi_i + \psi_i\mathcal{A} &= -(B_2K_i(\mathbf{x})C_2)^\top Y - YB_2K_i(\mathbf{x})C_2 \\ &\quad - (D_{12}K_i(\mathbf{x})C_2)^\top\mathcal{C} - \mathcal{C}^\top D_{12}K_i(\mathbf{x})C_2, \end{aligned} \tag{21}$$

using $D_K\mathcal{A}K_i(\mathbf{x}) = B_2K_i(\mathbf{x})C_2$, $D_K\mathcal{B}K_i(\mathbf{x}) = B_2K_i(\mathbf{x})D_{21}$, $D_K\mathcal{C}K_i(\mathbf{x}) = D_{21}K_i(\mathbf{x})C_2$. Since $X^{\frac{1}{2}}X^{\frac{1}{2}} = X$,

$$X^{\frac{1}{2}}\varphi_i + \varphi_i X^{\frac{1}{2}} = \phi_i. \tag{22}$$

Altogether, we obtain algorithm 1 to compute subgradients of f at $\mathbf{x}$. Here, depending on the controller structure $K(\mathbf{x})$, which is usually affine in $\mathbf{x}$, the $K_i(\mathbf{x})$ may be pre-calculated to accelerate the algorithm. Similar pre-calculations are possible in (19)-(22).

VII. Proximity control algorithm

In this section we present the main algorithm 2 to solve (4), respectively, (7), where the constraint of internal stability in (4) is addressed through (18). Our approach uses a progress function at the current iterate $\mathbf{x}$,

$$F(\cdot, \mathbf{x}) = \max\{f(\cdot) - f(\mathbf{x}) - \mu c(\mathbf{x})_+, c(\cdot) - c(\mathbf{x})_+\},$$

Algorithm 1 . Computing subgradients.

Input: $\mathbf{x} \in \mathbb{R}^n$. **Output:** $g \in \partial f(\mathbf{x})$.

1: Compute $K_i(\mathbf{x}) = \frac{\partial K(\mathbf{x})}{\partial \mathbf{x}_i}, i = 1, \ldots, n$ and X, Y solutions of (16), (17), respectively.
2: Compute $X^{\frac{1}{2}}$ and $Z = X^{\frac{1}{2}}YX^{\frac{1}{2}}$.
3: For $i = 1, \ldots, n$ compute ϕ_i and ψ_i solutions of (20) and (21), respectively.
4: For $i = 1, \ldots, n$ compute φ_i solution of (22) and $Z_i(\mathbf{x})$ using (19).
5: Determine a matrix Q whose columns form an orthonormal basis of the t-dimensional eigenspace associated with $\lambda_1(Z)$.
6: Pick $U \in \mathbb{B}_t$, and return $g := (\mathrm{Tr}(QUQ^\top Z_1(\mathbf{x})), \ldots, \mathrm{Tr}(QUQ^\top Z_n(\mathbf{x})))$, a subgradient of f at $\mathbf{x}$.

which is successively minimized. Antecedents of this idea can for instance be found in [10] , [11], and in our own contributions [5], [8].

In [7] Nesterov proposes a relaxation which solves specific convex nonsmooth programs by a smooth relaxation with a lower complexity than the convex bundle method. His method replaces $f(\mathbf{x}) = \lambda_1(Z(\mathbf{x}))$ by the smooth approximation

$$f_\mu(\mathbf{x}) := \mu \ln \left[\sum_{i=1}^m e^{\lambda_i(Z(\mathbf{x}))/\mu}\right]$$

with a tolerance parameter $\mu > 0$. We see that

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

Therefore, to find an ϵ-solution $\bar{\mathbf{x}}$ of problem (18), we find an $\frac{\epsilon}{2}$-solution of the smooth problem

$$\min\{f_\mu(\mathbf{x}) : c(\mathbf{x}) \leqslant 0\} \tag{23}$$

with $\mu = \frac{\epsilon}{2\ln m}$. We have used this idea to initialize the non smooth algorithm 2. The smoothed problem (23) can be solved using standard NLP software.

VIII. Experiment 1: Hankel feedback synthesis

In this section we apply program (7) to a classical 1-DOF control system design, using an example from [4, Chapter 2]. 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

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

Algorithm 2. Proximity control with downshifted tangents

Parameters: $0 < \gamma < \widetilde{\gamma} < 1$, $0 < \gamma < \Gamma < 1$, $0 < q < \infty$, $0 < c < \infty$.

1: **Initialize outer loop.** Choose initial iterate $\mathbf{x}^1$ and matrix $Q_1 = Q_1^\top$ with $-qI \preceq Q_1 \preceq qI$. Initialize memory control parameter $\tau_1^\sharp$ such that $Q_1 + \tau_1^\sharp I \succ 0$. Put $j = 1$.

2: **Stopping test.** At outer loop counter j, stop if $0 \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$. Otherwise, goto inner loop.

3: **Initialize inner loop.** Put inner loop counter $k = 1$ and initialize $\tau_1 = \tau_j^\sharp$. Build initial working model

$$F_1(\cdot, \mathbf{x}^j) = g_{0j}^\top(\cdot - \mathbf{x}^j) + \tfrac{1}{2}(\cdot - \mathbf{x}^j)^\top Q_j(\cdot - \mathbf{x}^j),$$

where $g_{0j} \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$.

4: **Trial step generation.** Compute

$$\mathbf{y}^k = \operatorname{argmin} F_k(\mathbf{y}, \mathbf{x}^j) + \frac{\tau_k}{2}\|\mathbf{y} - \mathbf{x}^j\|^2.$$

5: **Acceptance test.** If

$$\rho_k = \frac{F(\mathbf{y}^k, \mathbf{x}^j)}{F_k(\mathbf{y}^k, \mathbf{x}^j)} \geqslant \gamma,$$

put $\mathbf{x}^{j+1} = \mathbf{y}^k$ (serious step), quit inner loop and goto step 8. Otherwise (null step), continue inner loop with step 6.

6: **Update working model.** Generate a cutting plane $m_k(\cdot, \mathbf{x}^j) = a_k + g_k^\top(\cdot - \mathbf{x}^j)$ at null step $\mathbf{y}^k$ and counter k, where

$$\begin{aligned} g_k &\in \partial_1 F(\mathbf{y}^k, \mathbf{x}^j) - Q_j(\mathbf{y}^k - \mathbf{x}^j), \\ a_k &= t_k(\mathbf{x}^j) - s_k, \\ t_k(\cdot) &= F(\mathbf{y}^k, \mathbf{x}^j) - \tfrac{1}{2}(\mathbf{y}^k - \mathbf{x}^j)^\top Q_j(\mathbf{y}^k - \mathbf{x}^j) \\ &\qquad + g_k^\top(\cdot - \mathbf{y}^k), \\ s_k &= t_k(\mathbf{x}^j)_+ + c\|\mathbf{y}^k - \mathbf{x}^j\|^2. \end{aligned}$$

Compute aggregate plane $m_k^*(\cdot, \mathbf{x}^j) = a_k^* + g_k^{*\top}(\cdot - \mathbf{x}^j)$ at $\mathbf{y}^k$, where

$$\begin{aligned} g_k^* &= (Q_j + \tau_k I)(\mathbf{x}^j - \mathbf{y}^k), \\ a_k^* &= F_k(\mathbf{y}^k, \mathbf{x}^j) - \tfrac{1}{2}(\mathbf{y}^k - \mathbf{x}^j)^\top Q_j(\mathbf{y}^k - \mathbf{x}^j) \\ &\qquad + g_k^{*\top}(\mathbf{x}^j - \mathbf{y}^k). \end{aligned}$$

Build new working model

$$\begin{aligned} F_{k+1}(\cdot, \mathbf{x}^j) = \max\{ &F_k(\cdot, \mathbf{x}^j), m_k(\cdot, \mathbf{x}^j) + F^{[2]}(\cdot, \mathbf{x}^j), \\ &m_k^*(\cdot, \mathbf{x}^j) + F^{[2]}(\cdot, \mathbf{x}^j)\}, \end{aligned}$$

with $F^{[2]}(\cdot, \mathbf{x}^j) = \frac{1}{2}(\cdot - \mathbf{x}^j)^\top Q_j(\cdot - \mathbf{x}^j)$.

7: **Update proximity control parameter.** Compute secondary control parameter

$$\widetilde{\rho_k} = \frac{F_{k+1}(\mathbf{y}^k, \mathbf{x}^j)}{F_k(\mathbf{y}^k, \mathbf{x}^j)}$$

and put

$$\tau_{k+1} = \begin{cases} \tau_k & \text{if } \widetilde{\rho_k} < \widetilde{\gamma}, \\ 2\tau_k & \text{if } \widetilde{\rho_k} \geqslant \widetilde{\gamma}. \end{cases}$$

Increase inner loop counter k and loop back to step 4.

7: **Update Q_j and memory element.** Update matrix $Q_j \to Q_{j+1}$ respecting $Q_{j+1} = Q_{j+1}^\top$ and $-qI \preceq Q_{j+1} \preceq qI$. Then store new memory element

$$\tau_{j+1}^\sharp = \begin{cases} \frac{1}{2}\tau_{k+1} & \text{if } \rho_k < \Gamma, \\ \frac{1}{4}\tau_{k+1} & \text{if } \rho_k \geqslant \Gamma. \end{cases}$$

Increase $\tau_{j+1}^\sharp$ if necessary to ensure $Q_{j+1} + \tau_{j+1}^\sharp I \succ 0$. Increase outer loop counter j and loop back to step 2.

where

$$A = \begin{bmatrix} -10 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad B_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & -1 & 10 \\ 0 & 0 & 0 \end{bmatrix} \qquad D_{12} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 & 1 & -10 \end{bmatrix} \quad D_{21} = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}.$$

Following [4], we use the controller 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. This allows us to compare our results with the methods in [4], e.g. with the controller

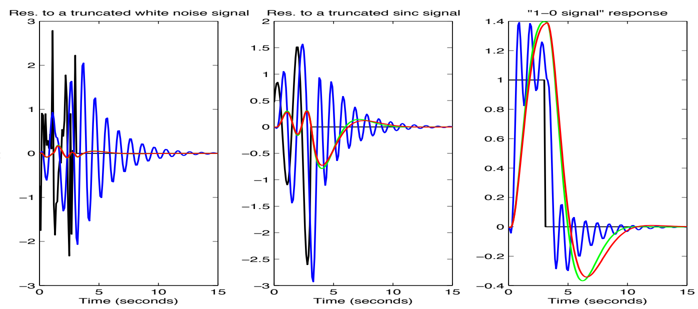


Fig. 1. White noise (left), sinc (middle), and 0-1 signal, truncated at $T = 3$. Plots show ringing for controllers K_b (blue), K_H (red), and K_∞ (green)..

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

synthesized in that reference. We also compare the optimal Hankel controller K_H to the optimal H_∞ controller K_∞ with the same structure,

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

which we computed using the Matlab function `hinfstruct` based on [1].

In order to compute K_H, we start algorithm 2 at an initial stabilizing controller

$$\mathbf{x}^1 = [3206.2, 12528.3, 11078.3, 7941.9, 13028.4, 3611.6]^\top$$

with $f(\mathbf{x}^1) = 11.0658$, using the stability constraint $c(\mathbf{x}) = \alpha(\mathcal{A}(\mathbf{x})) + \varepsilon$ with $\varepsilon = 10^{-8}$. We used the following two-stage stopping test. If the inner loop at $\mathbf{x}^j$ finds a serious iterate $\mathbf{x}^{j+1}$ satisfying

$$\frac{|f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j)|}{1 + |f(\mathbf{x}^j)|} < 10^{-5},$$

then $\mathbf{x}^{j+1}$ is accepted as the final solution. On the other hand, if the inner loop is unable to find a serious step and provides three consecutive unsuccessful trial steps $\mathbf{y}^k$ satisfying

$$\frac{\|\mathbf{x}^j - \mathbf{y}^k\|}{1 + \|\mathbf{x}^j\|} < 10^{-5}$$

or if a maximum number of 50 allowed steps k in the inner loop is reached, then we decide that $\mathbf{x}^j$ is already optimal. 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 [5] for theoretical results). The optimal controller obtained was

$$\mathbf{x}^* = [3202.0, 12990.8, 11497.7, 7650.0, 12408.9, 3513.9]^\top$$

with $f(\mathbf{x}^*) = 10.7494$ meaning $\|T_{w\to z}(P, K_H)\|_H = 3.2786$, where

$$K_H := K(\mathbf{x}^*) = \frac{7650.0s^2 + 12408.9s + 3513.9}{s^3 + 3202.0s^2 + 12990.8s + 11497.7}.$$

IX. Experiment 2: Hankel system reduction

Our tests use a 15th order Rolls-Royce Spey gas turbine engine model, decribed in [12, Chapter 11], with data available for download on I. Postlethwaites's homepage as `aero0.mat`. For $k = 1, 2, \dots, 14$, using algorithm 2, we computed reduced-order systems $G_{\text{red},k}$ of order k, and compared the achieved objective value in (7), which in this case is $f(\mathbf{x}) = \|G - G_k(\mathbf{x})\|_H$, with the theoretically known optimal Hankel norm approximation error $\|G - G_{\text{red},k}\|_H = \sigma_{k+1}(G)$, the $(k+1)$-st Hankel singular value of G. This error is equal to the error obtained by using the Hankel norm approximation algorithm.

For illustration we also used the gas turbine engine model to test H_∞-norm system reduction (13) by computing a model $G_{\text{red},\infty}$ of reduced order $k = 6$ for the 15th order model G. After defining P according to (11), we use the function `hinfstruct` [14], built on the prototype [1], to compute $\|G - G_{\text{red},\infty}\|_\infty = 0.169$. The procedure may be accelerated by choosing a Hankel norm reduction G_{red} to initialize the optimization, so that the minimum is reached very fast. It is interesting to notice that $\|G - G_{\text{red}}\|_\infty = 0.181$, so contrary to what is often claimed in the literature, in the present case the Hankel norm reduction G_{red} is not a good approximation of $G_{\text{red},\infty}$, as can also be seen in Fig. 2.

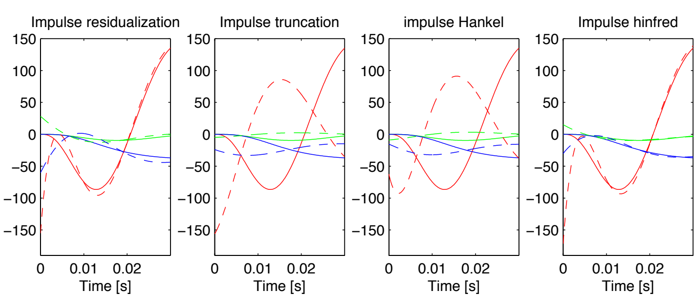


Fig. 2. Impulse responses for three channels of G (continuous lines), compared with with reduced models (dashed): Match DC (left), truncated (middle left), Hankel (middle right), H_∞ (right).

X. Experiment 3: Maximizing memory

Here we use an illustrative example (14), where G and G_{ref} are defined as

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

Filter F and controller K are structured as:

$$F(s) = \frac{as + b}{s^2 + cs + d}, \quad K(s) = \frac{m}{s + n}.$$

We have added a low-pass filter $W_1(s) = \frac{0.25s+0.6}{s+0.006}$ to the tracking signal z_1, and a weighting filter $W_2 = 0.01$ on the control output $z_2 = u + v$. The Hankel controller K_H computed by algorithm 2 delivers

$$F_H(s) = \frac{-10.3967s - 337.9222}{s^2 + 152.5995s + 337.8865}, K_H(s) = \frac{14.7058}{s + 7.0369}.$$

For comparison, we have also synthesized the optimal H_∞ control architecture of the same structure, which leads to

$$F_\infty(s) = \frac{-10.3956s - 337.9303}{s^2 + 152.6028s + 337.8770}, K_\infty(s) = \frac{14.7358}{s + 6.9270}.$$

The achieved H_∞ norms are $\|T_{w\to(W_1z_1,W_2z_2))}(K_\infty)\|_\infty = 0.0193 < \|T_{w\to(W_1z_1,W_2z_2))}(K_H)\|_\infty = 0.0200$ while the achieved H_∞ norms are $\|T_{w\to(W_1z_1,W_2z_2))}(K_H)\|_H = 0.0158 < \|T_{w\to(W_1z_1,W_2z_2))}(K_\infty)\|_H = 0.0175$.

XI. Conclusion

We have proposed a new methodology to reduce unwanted ringing effects in a tunable system $G(\mathbf{x})$. The problem was addressed by minimizing the Hankel norm $\|G(\mathbf{x})\|_H$ of $G(\mathbf{x})$, cast as an eigenvalue optimization program for the associated Hankel operator. We have proposed a non-smooth algorithm, which was shown to solve a variety of test problems successfully, and whose convergence to a critical point was previously established [5]. A smooth heuristic, based on work of Nesterov [7], was added and used to initialize the algorithm with a favorable initial seed.

References

[1] P. Apkarian, D. Noll. Nonsmooth H_∞ synthesis. *IEEE Trans. Automat. Control*, 51 (2006), no. 1, 71 – 86.
[2] P. Apkarian, D. Noll. Nonsmooth optimization for multidisk H_∞-synthesis. *European Journal of Control*, 12 (2006), no. 3, 229 – 244.
[3] V. Bompart, P. Apkarian, D. Noll. Nonsmooth techniques for stabilizing linear systems. *Proc. American Control Conf.*, New York, 2007.
[4] S. Boyd, C. Barratt. *Linear Controller Design*. Prentice Hall, 1991.
[5] M. Gabarrou, D. Alazard, D. Noll. Design of a flight control architecture using a non-convex bundle method. *Mathematics of Control, Signals, and Systems*, to appear.
[6] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *Int. J. Control*, 39 (1984), no. 6, 1115 – 1193.
[7] Y. Nesterov. Smoothing technique and its applications in semidefinite optimization. *Math. Programming, Ser. A*, 110 (2007), no. 2, 245 – 259.
[8] D. Noll, O. Prot, A. Rondepierre. A proximity control algorithm to minimize nonsmooth and nonconvex functions. *Pacific J. Optim.* 4 (2008), no. 3, 569 – 602.
[9] M.L. Overton. Large-scale optimization of eigenvalues. *SIAM J. Optimization*, 2 (1992), no. 1, 88 – 120.
[10] E. Polak. *Optimization: Algorithms and Consistent Approximations*. Appl. Math. Sci. 124, Springer-Verlag, New York, 1997.
[11] E. Polak, Y. Wardi. Nondifferentiable optimization algorithm for designing control systems having singular value inequalities. *Automatica*. 18 (1982), no. 3, 267 – 283.
[12] S. Skogestad, I. Postlethwaite. *Multivariable Feedback Control*. Wiley, 2005.
[13] K. Zhou, J. Doyle, K. Glover. *Robust and Optimal Control*. Prentice Hall, New York, 1996.
[14] The MathWorks, Inc. *Robust Control Toolbox*.