Non-smooth optimization for robust control of infinite-dimensional systems*

Pierre Apkarian † Dominikus Noll ‡ Laleh Ravanbod ‡

Abstract

We use a non-smooth trust-region method for H_∞-control of infinite-dimensional systems. Our method applies in particular to distributed and boundary control of partial differential equations. It is computationally attractive as it avoids the use of system reduction or identification. For illustration the method is applied to control a reaction-convection-diffusion system, a Van de Vusse reactor, and to a cavity flow control problem.

Keywords: Robust control · non-smooth optimization · non-smooth trust-region method · infinite-dimensional system · boundary control

1 Introduction

Feedback control of partial differential equations and other infinite-dimensional systems encounters limitations due to computational issues. In state-space, PDE models are overly complex and not directly suited for controller synthesis. System reduction is required to bring the state-space down to a size where synthesis methods are applicable. Not only is this technically demanding, it also bears the risk of producing inaccurate or oversimplified models, where H_∞-performance can no longer be guaranteed.

Computing the system transfer function directly from the infinite dimensional model avoids this loss of information, but encounters a second difficulty. Customary strategies now try to fit a finite-dimensional state-space model to the infinite-dimensional transfer function. This uses optimization-based identification techniques, which are in conflict with the H_∞-objective, as the two optimization procedures in series are no longer meaningful. In addition, for unstable systems the identification often uses heuristics or ad hoc approaches, which have no certificates.

The method we propose here avoids both pitfalls. We synthesize controllers directly from the pre-computed frequency response, thereby avoiding system reduction and identification. Discretization for computation is performed in frequency space on a low-dimensional object, which avoids the loss of information. Our tests demonstrate that this works fast and reliably, once the transfer function is available. It turns out that the success of our method hinges on the use of non-smooth optimization. We use a non-smooth trust-region method first proposed in [5], which allows trial steps tailored to the specific application. We prove convergence under Kiwiel's aggregation rule, a question which had

*Dedicated to the memory of Jon Borwein

†ONERA, Control System Department, Toulouse, France

‡Université de Toulouse, Institut de Mathématiques, Toulouse, France

remained open in [5]. This given an affirmative answers to a question already posed in [26] for the convex non-smooth trust-region method. For complementary information on bundle methods see [27, 15, 22], a mix of bundle and trust-regions is [30].

Design of controllers in the frequency domain based on non-smooth optimization has already been performed in [20, 21, 23, 24, 19, 9]. Structured $H-\infty$-control for infinite-dimensional systems is addressed in [3]. H_∞-control of a heat exchange system is discussed in [29]. These approaches use either unstructured controllers, are based on matrix inequalities, or differ with regard to the optimization technique.

The structure of the paper is as follows. In section 2 we outline our approach to H_∞-control of infinite dimensional systems. In section 3 we discuss optimization and present our non-smooth trust-region method originally proposed in [5], on which the present approach rests. Convergence of the non-smooth trust-region method is discussed in section 3.2. In section 4 we point to some particularities when applying the trust-region method to H_∞-optimization. This concerns the choice of working model, trial step, and stability barrier, as well as the approximation error between the infinite-dimensional H_∞-program and its discretization.

Numerical results for applications to infinite-dimensional control problems are presented in section 5. Subsection 5.1 shows how the method is, in general, applied to a boundary control problem, subsection 5.2 illustrates this in boundary H_∞-control of a non-linear reaction-convection-diffusion equation, and subsection 5.3 for a non-linear Van de Vusse reactor. Subsection 5.4 discusses a cavity flow control problem.

2 Control strategy

We consider an abstract linear time-invariant control system of the form

$$G: \quad \left\{ \begin{array}{ll} \dot{x} & = Ax + Bu \\ y & = Cx + Du \end{array} \right. \tag{1}$$

where A is an unbounded linear operator on a Hilbert space Z generating a strongly continuous semi-group, B a closed linear operator mapping the control input space U to Z, C a closed linear operator mapping Z to the space Y of measured outputs, and D a closed linear operator mapping U to Y. For practical reasons we assume that $U \simeq \mathbb{R}^p$ and $Y \simeq \mathbb{R}^m$, which reflects the fact that the process is assessed by a finite number of sensors and actuators, rendering our control law physically implementable. This means that B, D are bounded, while C is allowed to be closed unbounded. In addition, the domains of A and C satisfy $D(A) \subset D(C)$. As a consequence of the finite rank assumption on Y, U, the transfer function $G(s) = C(sI - A)^{-1}B + D$ is defined on the resolvent set $\rho(A)$ and meromorphic on $\mathbb{C}$, with values $G(s) \in \mathbb{C}^p \times \mathbb{C}^m$, see e.g. [16, 13].

We consider a class $K \in \mathscr{K}$ of feedback control laws which have similar state-space realizations

$$K: \quad \left\{ \begin{array}{ll} \dot{x}_K & = A_K x_K + B_K y \\ u & = C_K x_K + D_K y \end{array} \right. \tag{2}$$

on a Hilbert space $Z_{\mathscr{K}}$, with input space $Y \simeq \mathbb{R}^m$ and output space $U \simeq \mathbb{R}^p$, so that we can put G and $K \in \mathscr{K}$ in lower feedback $\mathcal{F}_\ell(G, K)$ as in Figure 1. Candidate controllers $K \in \mathscr{K}$ have to stabilize $\mathcal{F}_\ell(G, K)$ internally in closed loop, by which we mean that the infinitesimal generator of the closed loop system generates an exponentially stable semi-group [16, 10].

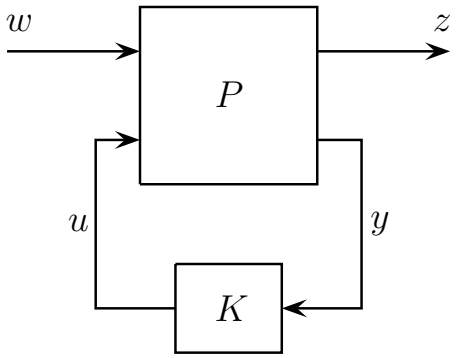

Figure 1: Lower feedback interconnection $\mathcal{F}_\ell(P, K)$ with H_∞-performance channel $w \to z$.

In H_∞-control one has not only to assure stability in closed loop, but also to guarantee good performance and robustness of the feedback system. For that purpose the system G is embedded in a plant P with a similar state-space realization

$$P: \quad \left\{ \begin{array}{rl} \dot{x} &= Ax + B_1 w + B_2 u \\ z &= C_1 x + D_{11} w + D_{12} u \\ y &= C_2 x + D_{21} w + D_{22} u \end{array} \right. \tag{3}$$

where the channel $y \to u$ is used for control, the channel $w \to z$ for performance. Closing the u-y-loop in (3) with (2) as in Figure 1 leaves us with the closed-loop transfer function $T_{wz}(K)$ from exogenous input w to regulated output z. The H_∞-control problem consists now in minimizing the L^2-L^2-operator norm of $T_{wz}(K)$ over a suitable class $K \in \mathscr{K}$ of admissible control laws. This operator norm is also known as the H_∞-norm, given as

$$\|T_{wz}(K)\|_\infty = \max_{\omega \in [0,\infty]} \overline{\sigma}\left(T_{wz}(K, j\omega)\right), \tag{4}$$

where $\overline{\sigma}(M)$ denotes the maximum singular value of a matrix M. If candidate controllers $K \in \mathscr{K}$ in (2) are parametrized as $K(\mathbf{x})$ for a finite-dimensional vector $\mathbf{x} \in \mathbb{R}^n$ of tunable parameters, then this H_∞-optimization program takes the form

$$\begin{array}{ll} \text{minimize} & f_\infty(\mathbf{x}) = \max\limits_{\omega \in [0,\infty]} \overline{\sigma}\left(T_{wz}(K(\mathbf{x}), j\omega)\right) \\ \text{subject to} & K(\mathbf{x}) \text{ stabilizes } G \text{ in closed-loop} \\ & \mathbf{x} \in \mathbb{R}^n \end{array} \tag{5}$$

For finite-dimensional real-rational $P, K(\mathbf{x})$ the objective (6) may be computed by an iterative procedure [7, 6], but for infinite dimensional $G, K(\mathbf{x})$, this is no longer possible, and we need to approximate (6) on a finite grid $\Omega_{\text{opt}} \subset [0, \infty]$. Introducing the discretized version

$$\|T_{wz}(K)\|_{\infty,d} = \max_{\omega \in \Omega_{\text{opt}}} \overline{\sigma}\left(T_{wz}(K, j\omega)\right)$$

of the H_∞-norm on the grid Ω_{opt}, this leads to the discretized H_∞-optimization program

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) = \max\limits_{\omega \in \Omega_{\text{opt}}} \overline{\sigma}\left(T_{wz}(K(\mathbf{x}), j\omega)\right) \\ \text{subject to} & K(\mathbf{x}) \text{ stabilizes } G \text{ in closed-loop} \\ & \mathbf{x} \in \mathbb{R}^n \end{array} \tag{6}$$

to which our non-smooth optimization method is applied. The overall procedure for H_∞-control is now given in algorithm 1.

Algorithm 1. Infinite-dimensional H_∞-synthesis

Parameters: Tolerance $\vartheta > 0$.

▷ **Step 1** (`Steady-state`)**.** Compute steady-state and linearize infinite-dimensional system about steady-state.

▷ **Step 2** (`Transfer function`)**.** Use linearized infinite-dimensional system to compute transfer function $G(j\omega)$, either formally for arbitrary $\omega \in [0, \infty]$, or numerically at a very high precision on a very fine grid Ω_{fine}.

▷ **Step 3** (`Plant`)**.** Set up plant P which defines the H_∞-performance channel $w \to z$.

▷ **Step 4** (`Grid for optimization`)**.** Find initially stabilizing $K(\mathbf{x}_0)$ for G and use it to compute grid for optimization Ω_{opt} such that $\|T_{wz}(K(\mathbf{x}_0))\|_\infty \leq f(\mathbf{x}_0) + \vartheta$. Either use a formal or a numerical function $G(j\omega)$, or extract Ω_{opt} from the pre-computed high precision grid Ω_{fine}.

▷ **Step 5** (`Non-smooth optimization`)**.** Use non-smooth trust-region algorithm 2 to compute locally optimal solution $K(\mathbf{x}^*)$ of (6).

▷ **Step 6** (`Refined grid`)**.** Check whether $\|T_{wz}(K(\mathbf{x}^*))\|_\infty \leq \|T_{wz}(K(\mathbf{x}^*))\|_{\infty,d} + \vartheta$. If not then add nodes to Ω_{opt} and go back to step 5.

Remark 1. It should be stressed that the computation of $G(j\omega)$ in steps 1 and 2 of algorithm 1 is the only moment where the full infinite-dimensional, or likewise, large-scale finite-dimensional model is used. Since this step is performed prior to optimization, and typically $|\Omega_{\text{opt}}| \ll |\Omega_{\text{fine}}|$, optimization is speedy. Also, the process of finding the correct P, which needs going back to step 3 of algorithm 1, is not slowed down by steps 1 and 2. In other words, once the transfer function $G(j\omega)$ is available, the complexity of the process is the same as that of a finite-dimensional structured H_∞-design procedure in the sense of [1, 8]. This will be illustrated in our experimental section.

Remark 2. Stability in closed-loop in step 4, and during optimization in step 5, uses the well-known Nyquist stability test in tandem with the barrier approach to be addressed in section 4.1. For the theoretical justification in the context of infinite-dimensional systems see [3].

We now address the individual steps of algorithm 1. The central ingredient is optimization, which is needed to solve (6), and which is discussed in the next sections 3 and 3.2. Generation of the grid Ω_{opt} in step 4, and the certificate in step 6, are discussed in section 4.

3 Non-smooth trust-region method

We work with the non-smooth trust-region method introduced in [5], which has already been successfully used in mechanical contact problems [11], and in system theory [5] for computing the worst-case H_∞-performance of a system, its stability margin, and its distance to instability. Here we use it for H_∞-control of infinite-dimensional systems. We also answer a question left open in [5, Remark 16], which concerns the theoretical justification of Kiwiel's aggregation technique [18] in non-smooth trust-regions. This question goes back to [26] for the convex trust-region method, but had until now remained open. An affirmative answer will be obtained in section 3.2.

3.1 Presentation of the algorithm

We briefly recall the essentials of the non-smooth trust-region method. For the present work it is sufficient to apply it to optimization programs of the form

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \tag{7}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz but non-smooth and non-convex. Following [22], a function $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is called a *model* of f if it satisfies the following properties:

(M_1) $\phi(\cdot, \mathbf{x})$ is convex, $\phi(\mathbf{x}, \mathbf{x}) = f(\mathbf{x})$, and $\partial_1\phi(\mathbf{x}, \mathbf{x}) \subset \partial f(\mathbf{x})$.

(M_2) If $\mathbf{y}_k \to \mathbf{x}$, then there exist $\epsilon_k \to 0^+$ such that $f(\mathbf{y}_k) \leq \phi(\mathbf{y}_k, \mathbf{x}) + \epsilon_k\|\mathbf{y}_k - \mathbf{x}\|$.

(M_3) If $\mathbf{x}_k \to \mathbf{x}, \mathbf{y}_k \to \mathbf{y}$, then $\limsup_{k\to\infty} \phi(\mathbf{y}_k, \mathbf{x}_k) \leq \phi(\mathbf{y}, \mathbf{x})$.

We may interpret $\phi(\cdot, \mathbf{x})$ as a substitute for the first-order Taylor expansion of f at $\mathbf{x}$. For convergence theory we need a slightly stronger type of model, which is given by the following:

Definition 1. *A first-order model ϕ of f is called* strict *if it satisfies the following stronger version of axiom (M_2):*

$(\widehat{M_2})$ *If $\mathbf{x}_k, \mathbf{y}_k \to \mathbf{x}$, then there exist $\epsilon_k \to 0^+$ such that $f(\mathbf{y}_k) \leq \phi(\mathbf{y}_k, \mathbf{x}_k) + \epsilon_k\|\mathbf{y}_k - \mathbf{x}_k\|$.*

□

The difference between (M_2) and the strict version $(\widehat{M_2})$ is analogous to the difference between differentiability and strict differentiability, hence the nomenclature. For additional information on the model concept see [22, 4, 5].

Remark 3. A typical example of a strict model ϕ is obtained when f is a maximum eigenvalue function $f(\mathbf{x}) = \lambda_1(F(\mathbf{x}))$, with $F : \mathbb{R}^n \to \mathbb{S}^m$ a class C^1-mapping into the space of $m \times m$ Hermitian matrices. We take $\phi(\mathbf{y}, \mathbf{x}) = \lambda_1(F(\mathbf{x}) + F'(\mathbf{x})(\mathbf{y} - \mathbf{x}))$. See [22, 4].

Definition 2. Let $\mathbf{x}$ be the current serious iterate of the trust-region algorithm, $\mathbf{z}$ a trial step. Let g be a subgradient of $\phi(\cdot, \mathbf{x})$ at $\mathbf{z}$. Then the affine function $m_{\mathbf{z}}(\cdot, \mathbf{x}) = \phi(\mathbf{z}, \mathbf{x}) + g^\top(\cdot - \mathbf{z})$ is called a *cutting plane* of f at serious iterate $\mathbf{x}$ and trial step $\mathbf{z}$. □

If $\mathbf{z} = \mathbf{x}$, then due to axiom (M_1) a cutting plane $m_{\mathbf{x}}(\cdot, \mathbf{x})$ at serious iterate $\mathbf{x}$ and trial step $\mathbf{z} = \mathbf{x}$ is just a tangent plane to f at $\mathbf{x}$. Since $m_{\mathbf{x}}(\mathbf{x}, \mathbf{x}) = f(\mathbf{x})$, including $m_{\mathbf{x}}(\cdot, \mathbf{x})$ in the working model $\phi_k(\cdot, \mathbf{x})$ at $\mathbf{x}$ guarantees exactness $\phi_k(\mathbf{x}, \mathbf{x}) = f(\mathbf{x})$ of the working model at all counters k. We cast this in the following

Definition 3. *A cutting plane $m_{\mathbf{x}}(\cdot, \mathbf{x})$ at serious iterate $\mathbf{x}$ and trial step $\mathbf{z} = \mathbf{x}$ is called an* exactness plane. □

As is standard in bundle and cutting plane algorithms, by storing cutting planes at unsuccessful trial steps $\mathbf{z}^k$, we accumulate information, which we use to build polyhedral models of f near $\mathbf{x}$. We use the notation $\phi_k(\cdot, \mathbf{x})$ for these *working models* of f formed by cutting planes, where k denotes the counter of the inner loop. Note that $\phi_k \leq \phi$ by construction of the cutting planes. If in addition a positive semi-definite symmetric matrix $Q(\mathbf{x}) \succeq 0$ is available as a substitute for the Hessian of f at $\mathbf{x}$, then we call $\Phi_k(\cdot, \mathbf{x}) = \phi_k(\cdot, \mathbf{x}) + \frac{1}{2}(\cdot - \mathbf{x})^\top Q(\mathbf{x})(\cdot - \mathbf{x})$ a *second-order working model* of f at serious iterate $\mathbf{x}$.

We are now ready to present the bundle trust-region algorithm. (See algorithm 2).

Algorithm 2. Non-smooth trust-region algorithm

Parameters: $0 < \gamma < \widetilde{\gamma} < 1$, $0 < \gamma < \Gamma \leq 1$, $0 < \theta \ll 1$, $M \geq 1$, $q > 0$.

▷ **Step 1** (`Initialize outer loop`)**.** Fix initial iterate $\mathbf{x}^1$ and memory trust-region radius $R_1^\sharp > 0$. Initialize $Q_1 \succeq 0$ with $\|Q_1\| \leq q$. Put outer loop counter $j = 1$.

▷ **Step 2** (`Stopping test`)**.** At outer loop counter j, stop if $\mathbf{x}^j$ is a critical point of (7). Otherwise go to inner loop.

▷ **Step 3** (`Initialize inner loop`)**.** Put inner loop counter $k = 1$ and initialize trust-region radius as $R_1 = R_j^\sharp$. Build polyhedral first-order working model $\phi_1(\cdot, \mathbf{x}^j)$, where at least one exactness plane at $\mathbf{x}^j$ is included. Possibly enrich by adding recycled planes from previous steps, or by including anticipated cutting planes. Build second-order working model $\Phi_1(\cdot, \mathbf{x}^j) = \phi_1(\cdot, \mathbf{x}^j) + \frac{1}{2}(\cdot - \mathbf{x}^j)^\top Q_j(\cdot - \mathbf{x}^j)$.

▷ **Step 4** (`Trial step generation`)**.** At inner loop counter k compute solution $\mathbf{y}^k$ of trust-region tangent program

$$\begin{array}{ll} \text{minimize} & \Phi_k(\mathbf{y}, \mathbf{x}^j) \\ \text{subject to} & \|\mathbf{y} - \mathbf{x}^j\| \leq R_k \end{array}$$

Then admit any $\mathbf{z}^k$ satisfying $\|\mathbf{z}^k - \mathbf{x}^j\| \leq M\|\mathbf{y}^k - \mathbf{x}^j\|$ and $f(\mathbf{x}^j) - \Phi_k(\mathbf{z}^k, \mathbf{x}^j) \geq \theta\left(f(\mathbf{x}^j) - \Phi_k(\mathbf{y}^k, \mathbf{x}^j)\right)$ as trial step.

▷ **Step 5** (`Acceptance test`)**.** If

$$\rho_k = \frac{f(\mathbf{x}^j) - f(\mathbf{z}^k)}{f(\mathbf{x}^j) - \Phi_k(\mathbf{z}^k, \mathbf{x}^j)} \geq \gamma$$

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

▷ **Step 6** (`Update working model`)**.** Generate a cutting plane $m_k(\cdot, \mathbf{x}^j)$ of f at the unsuccessful trial step $\mathbf{z}^k$ and add it to the polyhedral model. Possibly taper out ϕ_k by removing some of the older cuts, and build new first-order working $\phi_{k+1}(\cdot, \mathbf{x}^j)$. Then $\Phi_{k+1}(\cdot, \mathbf{x}^j) = \phi_{k+1}(\cdot, \mathbf{x}^j) + \frac{1}{2}(\cdot - \mathbf{x}^j)^\top Q_j(\cdot - \mathbf{x}^j)$ is the new second-order working model. Continue with step 7.

▷ **Step 7** (`Update trust-region radius`)**.** Compute secondary control parameter

$$\widetilde{\rho}_k = \frac{f(\mathbf{x}^j) - \phi_{k+1}(\mathbf{z}^k, \mathbf{x}^j)}{f(\mathbf{x}^j) - \Phi_k(\mathbf{z}^k, \mathbf{x}^j)}$$

and put

$$R_{k+1} = \begin{cases} R_k & \text{if } \widetilde{\rho}_k < \widetilde{\gamma} \\ \frac{1}{2}R_k & \text{if } \widetilde{\rho}_k \geq \widetilde{\gamma} \end{cases}$$

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

▷ **Step 8** (`Update memory radius`)**.** Store new memory trust-region radius

$$R_{j+1}^\sharp = \begin{cases} R_k & \text{if } \rho_k < \Gamma \\ 2R_k & \text{if } \rho_k \geq \Gamma \end{cases}$$

Update $Q_j \to Q_{j+1}$ respecting $Q_{j+1} \succeq 0$ and $\|Q_{j+1}\| \leq q$. Increase outer loop counter j and go back to step 2.

Remark 4. Before we discuss convergence of algorithm 2 in the next section, we recall the form of the tangent program in step 4 from [5]. Let the first-order working model at inner loop counter k have the form $\phi_k(\cdot, \mathbf{x}^j) = \max_{i \in I_k} a_i + g_i^\top(\cdot - \mathbf{x}^j)$ for some finite set I_k, and suppose the trust-region norm is the maximum norm. Then the tangent program at serious iterate $\mathbf{x}^j$ and inner loop instant k is the following CQP

$$\begin{array}{ll} \text{minimize} & t + \frac{1}{2}(\mathbf{y} - \mathbf{x}^j)^\top Q_j(\mathbf{y} - \mathbf{x}^j) \\ \text{subject to} & a_i + g_i^\top(\mathbf{y} - \mathbf{x}^j) \le t, \quad i \in I_k \\ & -R_k \le \mathbf{y}_i - \mathbf{x}_i^j \le R_k \quad , i = 1, \ldots, n \end{array} \tag{8}$$

with decision variable $(t, \mathbf{y}) \in \mathbb{R} \times \mathbb{R}^n$, giving rise to the solution $\mathbf{y}^k$ in step 4.

3.2 Convergence

In this section we prove convergence of the trust-region algorithm 2 toward a Clarke critical point. As is standard, we start by proving that the inner loop ends finitely if $0 \notin \partial f(\mathbf{x}^j)$, where throughout ∂f denotes the Clarke subdifferential. During this part of the proof we write $\mathbf{x} = \mathbf{x}^j$ and $Q = Q_j$, as those are fixed during the inner loop at counter j. Note that by the necessary optimality condition for the tangent program in step 4 of algorithm 2, there exists a subgradient $g_k^* \in \partial_1 \phi_k(\mathbf{y}^k, \mathbf{x})$ such that $g_k^* + Q(\mathbf{y}^k - \mathbf{x}) + \mathbf{v}_k = 0$, where $\mathbf{v}_k$ is in the normal cone to the trust-region norm ball $B(\mathbf{x}, R_k)$ at $\mathbf{y}^k$.

Definition 4. *We call g_k^* the* aggregate subgradient. *The affine function $m_k^*(\cdot, \mathbf{x}) = \phi_k(\mathbf{y}^k, \mathbf{x}) + g_k^{*\top}(\cdot - \mathbf{y}^k)$ is called the* aggregate plane. □

Lemma 1. *There exists $\sigma > 0$ depending only on the constants $\theta \in (0, 1)$ and $M > 0$ in algorithm 2 and on the trust-region norm $\|\cdot\|$, such that for every trial point $\mathbf{z}^k$ at inner loop instant k with corresponding solution $\mathbf{y}^k$ of the trust-region tangent program in step 4, and for the corresponding aggregate subgradient $g_k^* \in \partial_1 \phi(\mathbf{y}^k, \mathbf{x})$, we have the estimate*

$$f(\mathbf{x}) - \phi_k(\mathbf{z}^k, \mathbf{x}) \ge \sigma \|g_k^* + Q(\mathbf{y}^k - \mathbf{x})\| \|\mathbf{z}^k - \mathbf{x}\|. \tag{9}$$

Proof: This is essentially the same as [5, Lemma 1]. □

Lemma 2. *Suppose the inner loop at $\mathbf{x}$ turns infinitely, and $\liminf_{k\to\infty} R_k = 0$. Then $\mathbf{x}$ is a critical point of (7).*

Proof: According to step 7 of algorithm 2 we have $\widetilde{\rho}_k \ge \widetilde{\gamma}$ for infinitely many $k \in \mathcal{K}$. Since R_k is never increased during the inner loop, that implies $R_k \to 0$. Hence $\mathbf{y}^k, \mathbf{z}^k \to \mathbf{x}$ as $k \to \infty$, where we use the trial step generation rule of step 4 of algorithm 2. We argue that this implies $\phi_k(\mathbf{z}^k, \mathbf{x}) \to f(\mathbf{x})$.

Indeed, $\limsup_{k\to\infty} \phi_k(\mathbf{z}^k, \mathbf{x}) \le \limsup_{k\to\infty} \phi(\mathbf{z}^k, \mathbf{x}) = \lim_{k\to\infty} \phi(\mathbf{z}^k, \mathbf{x}) = f(\mathbf{x})$ is always true due to $\phi_k \le \phi$ and axiom (M_1). On the other hand, ϕ_k includes (i.e. dominates) an exactness plane $m_0(\cdot, \mathbf{x}) = f(\mathbf{x}) + g_0^\top(\cdot - \mathbf{x})$, hence $f(\mathbf{x}) = \lim_{k\to\infty} m_0(\mathbf{z}^k, \mathbf{x}) \le \liminf \phi_k(\mathbf{z}^k, \mathbf{x})$. The two together show $\phi_k(\mathbf{z}^k, \mathbf{x}) \to f(\mathbf{x})$, and then immediately also $\Phi_k(\mathbf{z}^k, \mathbf{x}) \to f(\mathbf{x})$. We also readily obtain $\phi_k(\mathbf{y}^k, \mathbf{x}) \to f(\mathbf{x})$ from the link between $\mathbf{z}^k, \mathbf{y}^k$ in step 4 of algorithm 2.

We now prove that $\liminf_{k\to\infty} \|g_k^*\| = 0$. Assume on the contrary that $\|g_k^*\| \ge \eta > 0$ for all k. Choose k large enough to have $\|g_k^* + Q(\mathbf{y}^k - \mathbf{x})\| \ge \frac{1}{2}\|g_k^*\|$. This is possible because the sequence g_k^* is bounded away from 0 and $Q(\mathbf{y}^k - \mathbf{x}) \to 0$. Then by estimate (9)

we have $f(\mathbf{x})-\phi_k(\mathbf{z}^k,\mathbf{x}) \geq \frac{1}{2}\sigma\eta\|\mathbf{z}^k-\mathbf{x}\|$. Therefore, for k large enough, $f(\mathbf{x})-\Phi_k(\mathbf{z}^k,\mathbf{x}) \geq \frac{1}{4}\sigma\eta\|\mathbf{z}^k-\mathbf{x}\|$, as the quadratic term in Φ_k is of the order $\|\mathbf{z}^k-\mathbf{x}\|^2$. Since $\mathbf{z}^k \to \mathbf{x}$, by axiom $(\widehat{M}_2)$ there exist $\epsilon_k \to 0^+$ such that $f(\mathbf{z}^k)-\phi(\mathbf{z}^k,\mathbf{x}) \leq \epsilon_k\|\mathbf{z}^k-\mathbf{x}\|$. Now we estimate

$$\widetilde{\rho}_k = \rho_k + \frac{f(\mathbf{z}^k)-\phi(\mathbf{z}^k,\mathbf{x})}{f(\mathbf{x})-\Phi_k(\,\mathbf{z}^k,\mathbf{x})} \leq \rho_k + \frac{\epsilon_k\|\mathbf{z}^k-\mathbf{x}\|}{\frac{1}{4}\sigma\eta\|\mathbf{z}^k-\mathbf{x}\|} = \rho_k + 4\epsilon_k/(\sigma\eta).$$

Since $\epsilon_k \to 0$ and $\rho_k < \gamma$, we have $\limsup \widetilde{\rho}_k \leq \gamma < \widetilde{\gamma}$, a contradiction with $\widetilde{\rho}_k > \widetilde{\gamma}$ for the infinitely many $k \in \mathcal{K}$. That proves $g_k^* \to 0$ for a subsequence $k \in \mathcal{N}$.

Next observe that by the subgradient inequality and $\phi_k \leq \phi$ we have

$$g_k^{*\top}\mathbf{h} \leq \phi_k(\mathbf{y}^k+\mathbf{h},\mathbf{x}) - \phi_k(\mathbf{y}^k,\mathbf{x}) \leq \phi(\mathbf{y}^k+\mathbf{h},\mathbf{x}) - \phi_k(\mathbf{y}^k,\mathbf{x}).$$

Since $\phi_k(\mathbf{y}^k,\mathbf{x}) \to f(\mathbf{x}) = \phi(\mathbf{x},\mathbf{x})$, passing to the limit $k \in \mathcal{N}$ and using $g_k^* \to 0$, $\mathbf{y}^k \to \mathbf{x}$ implies

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

Since $\mathbf{h}$ was arbitrary, we have $0 \in \partial_1\phi(\mathbf{x},\mathbf{x}) \subset \partial f(\mathbf{x})$ by (M_1). That proves the Lemma. □

This result needs only the fact that $\phi_k \leq \phi$, so it is not in conflict with any rule in step 6 used to taper out $\phi_k \to \phi_{k+1}$, as long as some exactness plane is present in ϕ_k at all inner loop instants k. In the next two lemmas we examine the more involved case when R_k is bounded away from 0. Here we require not only that an exactness plane is present at all times, but also that the latest cutting plane is added into ϕ_{k+1}.

However, this leads to a tangent program of size growing with k, which raises the question whether it is in principle possible to limit the number of planes included in the working model $\phi_k(\cdot,\mathbf{x})$ in step 6. For the convex bundle method this question is answered in the affirmative by Kiwiel's aggregation rule [18], according to which only three planes are required, an exactness plane, the latest cut, and the aggregate plane to account for the past. For the non-convex bundle method, an affirmative answer was first given in [22]. The aggregate plane is a convex combination of active cuts at the trial point $\mathbf{y}^k$, and can be described as follows. Had we removed from the last tangent program all active planes, and substituted instead the aggregate plane, the solution $\mathbf{y}^k$ would have been the same. The idea of the aggregation technique is to add the aggregate plane into the working model after an unsuccessful trial step $\mathbf{y}^k$, which allows to remove the active planes for the next sweep. Inactive planes may leave the model in any case.

Remark 5. Ruszczyński [26] stresses that the situation is more delicate for the convex non-smooth trust region method, and asks whether convergence could be proved under the aggregation rule, or under any other rule allowing to limit the number of planes included in the ϕ_k. Here we address this question in the general non-convex case. In [5] we had shown that the number of cuts in step 6 may at least be limited to $n+2$ using Carathéodory's theorem, but we remarked that it would be far more attractive to have a maximum number independent of the dimension n as in Kiwiel's rule. In [5, Remark 16] we observed that the question whether Kiwiel's rule could also be justified for the trust-region method was still open. Here we shall answer this question in the affirmative by proving convergence under the aggregation rule.

The following result justifies the use of aggregation in the first place for the special case $\mathbf{z}^k = \mathbf{y}^k$. Note that the trivial choice $\mathbf{z}^k = \mathbf{y}^k$ in step 4 is always authorized (due to

$M \geq 1$ and $\theta \leq 1$), but of course we want to use the additional freedom offered by $\mathbf{z}^k$ to improve performance of our method, so $\mathbf{z}^k = \mathbf{y}^k$ is rather restrictive, and we will remove it later.

Lemma 3. *Suppose the inner loop at* $\mathbf{x}$ *turns infinitely and the trust-region radius* R_k *stays bounded away from* 0. *Let* $Q \succ 0$ *and suppose the aggregation rule is used to taper out the models in step* 6. *Suppose the* $\mathbf{y}^k$ *are chosen as trial steps. Then* $\mathbf{x}$ *is a critical point of* (7).

Proof: Since the trust-region radius is frozen $R_k = R_{k_0}$ from some counter k_0 onwards, we write $R := R_{k_0}$. According to step 7 of the algorithm that means $\widetilde{\rho}_k < \widetilde{\gamma}$ for $k \geq k_0$. The only progress in the working model as we update $\phi_k \to \phi_{k+1}$ is now the addition of the cutting plane and the aggregate plane. The working models ϕ_k now contain at least three planes, an exactness plane, the latest cut from the last unsuccessful trial step, and the aggregate plane. They may contain more planes, but those will not be used in our argument below.

We want to prove $\mathbf{y}^k \to \mathbf{x}$. Since R_k stays bounded away from 0, this is more involved than in the previous Lemma. Since $Q \succ 0$ is fixed, we introduce the Euclidian norm $|\mathbf{x}|_Q^2 = \mathbf{x}^\top Q \mathbf{x}$. With this arrangement the objective function of the tangent program becomes

$$\Phi_k(\cdot, \mathbf{x}) = \phi_k(\cdot, \mathbf{x}) + \tfrac{1}{2}|\cdot - \mathbf{x}|_Q^2.$$

We know that the cutting plane $m_k(\cdot, \mathbf{x})$ at trial step $\mathbf{y}^k$ satisfies $m_k(\mathbf{y}^k, \mathbf{x}) = \phi(\mathbf{y}^k, \mathbf{x})$, so it memorizes the value $\phi(\mathbf{y}^k, \mathbf{x})$, while the aggregate plane $m_k^*(\cdot, \mathbf{x})$ satisfies $m_k^*(\mathbf{y}^k, \mathbf{x}) = \phi_k(\mathbf{y}^k, \mathbf{x})$, so it memorizes in turn the value $\phi_k(\mathbf{y}^k, \mathbf{x})$. The latter gives

$$\Phi_k(\mathbf{y}^k, \mathbf{x}) = m_k^*(\mathbf{y}^k, \mathbf{x}) + \tfrac{1}{2}|\mathbf{y}^k - \mathbf{x}|_Q^2. \tag{10}$$

Now we introduce the quadratic function

$$\Phi_k^*(\cdot, \mathbf{x}) = m_k^*(\cdot, \mathbf{x}) + \tfrac{1}{2}|\cdot - \mathbf{x}|_Q^2,$$

then from what we have just seen in (10)

$$\Phi_k^*(\mathbf{y}^k, \mathbf{x}) = \Phi_k(\mathbf{y}^k, \mathbf{x}). \tag{11}$$

Moreover, we have

$$\Phi_k^*(\cdot, \mathbf{x}) \leq \Phi_{k+1}(\cdot, \mathbf{x}), \tag{12}$$

because according to the aggregation rule we include the aggregate plane $m_k^*(\cdot, \mathbf{x})$ in the built of the new model ϕ_{k+1}, that is, we have $m_k^*(\cdot, \mathbf{x}) \leq \phi_{k+1}(\cdot, \mathbf{x})$, and hence (12). Expanding the quadratic function $\Phi_k^*(\cdot, \mathbf{x})$ at $\mathbf{y}^k$ gives

$$\Phi_k^*(\cdot, \mathbf{x}) = \Phi_k^*(\mathbf{y}^k, \mathbf{x}) + \nabla\Phi_k^*(\mathbf{y}^k, \mathbf{x})^\top(\cdot - \mathbf{y}^k) + \tfrac{1}{2}|\cdot - \mathbf{y}^k|_Q^2,$$

where $\nabla\Phi_k^* = g_k^* + Q(\mathbf{y}^k - \mathbf{x})$. From the optimality condition of the tangent program at $\mathbf{y}^k$ we get $g_k^* + Q(\mathbf{y}^k - \mathbf{x}) = -\mathbf{v}_k$ with $\mathbf{v}_k$ in the normal cone to the ball $B(\mathbf{x}, R)$ at $\mathbf{y}^k$, hence

$$\Phi_k^*(\cdot, \mathbf{x}) = \Phi_k^*(\mathbf{y}^k, \mathbf{x}) - \mathbf{v}_k^\top(\cdot - \mathbf{y}^k) + \tfrac{1}{2}|\cdot - \mathbf{y}^k|_Q^2. \tag{13}$$

Now we argue as follows:

$$\begin{aligned}
\Phi_k(\mathbf{y}^k,\mathbf{x}) &= \Phi_k^*(\mathbf{y}^k,\mathbf{x}) && \text{(by (11))}\\
&\le \Phi_k^*(\mathbf{y}^k,\mathbf{x}) + \tfrac{1}{2}|\mathbf{y}^{k+1}-\mathbf{y}^k|_Q^2 &&\\
&= \Phi_k^*(\mathbf{y}^{k+1},\mathbf{x}) + \mathbf{v}_k^\top(\mathbf{y}^{k+1}-\mathbf{y}^k) && \text{(by (13))}\\
&\le \Phi_k^*(\mathbf{y}^{k+1},\mathbf{x}) && \text{(since } \mathbf{v}_k^\top(\mathbf{y}^{k+1}-\mathbf{y}^k)\le 0\text{)}\\
&\le \Phi_{k+1}(\mathbf{y}^{k+1},\mathbf{x}) && \text{(by (12))}\\
&\le \Phi_{k+1}(\mathbf{x},\mathbf{x}) && (\mathbf{y}^{k+1}\text{ minimizer of } \Phi_{k+1}(\cdot,\mathbf{x}))\\
&= \phi(\mathbf{x},\mathbf{x}) = f(\mathbf{x}).
\end{aligned} \tag{14}$$

Therefore the sequence $\Phi_k(\mathbf{y}^k,\mathbf{x})$ is increasing and bounded above, and converges to a limit $\Phi^* \le f(\mathbf{x})$. Going back with this information to the estimation chain (14) shows $\frac{1}{2}|\mathbf{y}^{k+1}-\mathbf{y}^k|_Q^2 \to 0$ and also $\mathbf{v}_k^\top(\mathbf{y}^{k+1}-\mathbf{y}^k)\to 0$. Then also

$$\tfrac{1}{2}|\mathbf{y}^{k+1}-\mathbf{x}|_Q^2 - \tfrac{1}{2}|\mathbf{y}^{k}-\mathbf{x}|_Q^2 \to 0,$$

because $|\cdot|_Q$ is a Euclidian norm. In consequence

$$\phi_{k+1}(\mathbf{y}^{k+1},\mathbf{x}) - \phi_k(\mathbf{y}^k,\mathbf{x}) = \Phi_{k+1}(\mathbf{y}^{k+1},\mathbf{x}) - \Phi_k(\mathbf{y}^k,\mathbf{x}) - \tfrac{1}{2}|\mathbf{y}^{k+1}-\mathbf{x}|_Q^2 + \tfrac{1}{2}|\mathbf{y}^{k}-\mathbf{x}|_Q^2 \to 0.$$

Now recall that the cutting plane $m_k(\cdot,\mathbf{x})$ is an affine support function of $\phi_{k+1}(\cdot,\mathbf{x})$ at $\mathbf{y}^k$. Hence by the subgradient inequality

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

Since $\phi_{k+1}(\mathbf{y}^k,\mathbf{x}) = \phi(\mathbf{y}^k,\mathbf{x})$, we deduce

$$\phi(\mathbf{y}^k,\mathbf{x}) + g_k^\top(\cdot-\mathbf{y}^k) \le \phi_{k+1}(\cdot,\mathbf{x}). \tag{15}$$

Now using (15) we estimate as follows:

$$\begin{aligned}
0 &\le \phi(\mathbf{y}^k,\mathbf{x}) - \phi_k(\mathbf{y}^k,\mathbf{x})\\
&= \phi(\mathbf{y}^k,\mathbf{x}) + g_k^\top(\mathbf{y}^{k+1}-\mathbf{y}^k) - \phi_k(\mathbf{y}^k,\mathbf{x}) - g_k^\top(\mathbf{y}^{k+1}-\mathbf{y}^k)\\
&\le \phi_{k+1}(\mathbf{y}^{k+1},\mathbf{x}) - \phi_k(\mathbf{y}^k,\mathbf{x}) - g_k^\top(\mathbf{y}^{k+1}-\mathbf{y}^k).
\end{aligned}$$

Since $\mathbf{y}^{k+1}-\mathbf{y}^k\to 0$ and the g_k are bounded, we have $g_k^\top(\mathbf{y}^{k+1}-\mathbf{y}^k)\to 0$, hence we deduce $\phi(\mathbf{y}^k,\mathbf{x}) - \phi_k(\mathbf{y}^k,\mathbf{x})\to 0$, and also $\Phi(\mathbf{y}^k,\mathbf{x}) - \Phi_k(\mathbf{y}^k,\mathbf{x})\to 0$.

Now we claim that $\phi_k(\mathbf{y}^k,\mathbf{x})\to f(\mathbf{x})$. Since $\phi_k(\mathbf{y}^k,\mathbf{x}) \le \Phi_k(\mathbf{y}^k,\mathbf{x}) \to \Phi^* \le f(\mathbf{x})$, it remains to prove $\liminf \phi_k(\mathbf{y}^k,\mathbf{x}) \ge f(\mathbf{x})$. Suppose that this is not the case, and let $\phi_k(\mathbf{y}^k,\mathbf{x})\to f(\mathbf{x})-\eta$ for a subsequence and some $\eta>0$. Then also $\phi(\mathbf{y}^k,\mathbf{x})\to f(\mathbf{x})-\eta$ for that subsequence. (Here we use that $\phi_k(\mathbf{y}^k,\mathbf{x}) - \phi(\mathbf{y}^k,\mathbf{x})\to 0$ proved above). Passing to yet another subsequence, and using boundedness of the $\mathbf{y}^k$, we may assume $\frac{1}{2}|\mathbf{y}^k-\mathbf{x}|_Q^2 \to \ell \ge 0$. Choose $\delta>0$ such that $\delta < (1-\widetilde{\gamma})\eta$. From what we have just seen there exists k_1 such that

$$\phi(\mathbf{y}^k,\mathbf{x}) - \phi_k(\mathbf{y}^k,\mathbf{x}) < \delta$$

for all $k\ge k_1$. Now recall that $\widetilde{\rho}_k \le \widetilde{\gamma}$ for every $k\ge k_0$, hence

$$\widetilde{\gamma}\left(\Phi_k(\mathbf{y}^k,\mathbf{x}) - f(\mathbf{x})\right) \le \phi(\mathbf{y}^k,\mathbf{x}) - f(\mathbf{x}) \le \phi_k(\mathbf{y}^k,\mathbf{x}) - f(\mathbf{x}) + \delta.$$

Passing to the limit gives $-\widetilde{\gamma}\eta + \widetilde{\gamma}\ell \leq -\eta + \delta$, hence $(1-\widetilde{\gamma})\eta + \ell\widetilde{\gamma} \leq \delta$, which contradicts the choice of δ. Hence $\phi_k(\mathbf{y}^k, \mathbf{x}) \to f(\mathbf{x})$. We immediately deduce that $\Phi_k(\mathbf{y}^k, \mathbf{x}) \to f(\mathbf{x})$ and $\Phi(\mathbf{y}^k, \mathbf{x}) \to f(\mathbf{x})$.

We now argue that $\mathbf{y}^k \to \mathbf{x}$. This follows from the estimates

$$\phi_k(\mathbf{y}^k, \mathbf{x}) \leq \Phi_k(\mathbf{y}^k, \mathbf{x}) = \phi_k(\mathbf{y}^k, \mathbf{x}) + \tfrac{1}{2}|\mathbf{y}^k - \mathbf{x}|_Q^2 \leq \Phi^* \leq f(\mathbf{x})$$

because $\phi_k(\mathbf{y}^k, \mathbf{x}) \to f(\mathbf{x})$ now shows that all terms go to $f(\mathbf{x})$, and that implies $|\mathbf{y}^k - \mathbf{x}|_Q \to 0$. Since $Q \succ 0$ we deduce $\mathbf{y}^k \to \mathbf{x}$, and this is where the proof no longer works if only $Q \succeq 0$. Note that this shows that $\mathbf{y}^k$ is in the interior of $B(\mathbf{x}, R)$ from some counter onward, so that $\mathbf{v}_k = 0$.

Let us now show that $0 \in \partial f(\mathbf{x})$. From the subgradient inequality we have

$$g_k^{*\top}(\mathbf{x} + \mathbf{h} - \mathbf{y}^k) \leq \phi_k(\mathbf{x} + \mathbf{h}, \mathbf{x}) - \phi_k(\mathbf{y}^k, \mathbf{x}) \leq \phi(\mathbf{x} + \mathbf{h}, \mathbf{x}) - \phi_k(\mathbf{y}^k, \mathbf{x}).$$

Passing to the limit using $g_k^* \to 0$, $\phi_k(\mathbf{y}^k, \mathbf{x}) \to f(\mathbf{x}) = \phi(\mathbf{x}, \mathbf{x})$, we obtain

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

and since $\mathbf{h}$ is arbitrary and $\phi(\cdot, \mathbf{x})$ is convex, this gives $0 \in \partial_1\phi(\mathbf{x}, \mathbf{x}) \subset \partial f(\mathbf{x})$. □

Remark 6. For the general case $Q \succeq 0$ and $\mathbf{z}^k$ different from $\mathbf{y}^k$ it is still not known whether aggregation is justified, but with [5, Lemma 2] we can prove convergence if we keep all cuts in the model, or if we use the Carathéodory type argument of that reference to limit the number of cutting planes in the inner loop to $\leq n + 2$.

Remark 7. Note that $Q \succ 0$ is not a restriction in practice, but $\mathbf{z}^k = \mathbf{y}^k$ is. Fortunately the aggregation technique may still be justified in the case $\mathbf{z}^k \neq \mathbf{y}^k$ if we proceed as follows. In the first place we allow $\mathbf{z}^k$ as a trial point in step 4. If acceptance fails, then we perform step 7. However, if step 7 gives no reduction of R_k, then we are in the difficult case. We then do the following. We fall back on $\mathbf{y}^k$ as the trial point, i.e., we forget about $\mathbf{z}^k$. When $\mathbf{y}^k$ is not accepted, we proceed with step 6 and apply aggregation. This is now justified because we are in the situation covered by Lemma 3. Note that the additional work required in steps 6 and 7 is marginal, so we do not waste time by this evasive maneuver. We could even perform this maneuver as default (i.e. checking $\mathbf{y}^k$ if $\mathbf{z}^k$ fails). We have proved the following

Lemma 4. *Suppose the inner loop turns infinitely. Suppose $Q \succ 0$ and that the aggregation rule is used in step 6 to limit the size of ϕ_k to any pre-defined fixed number $N \geq 3$. Suppose we accept to fall back on $\mathbf{y}^k$ if $\mathbf{z}^k$ fails in step 5 with $\widetilde{\rho}_k < \widetilde{\gamma}$ in step 7. Then $0 \in \partial f(\mathbf{x})$.* □

Remark 8. In the classical trust-region method failure of the trial step always leads to reduction of the trust-region radius. One occasionally sees non-smooth versions where authors do the same. As we have already shown in [5, section 5.5], that must fail. The example in that reference also shows that the Cauchy point fails in the non-smooth case.

We are now ready to state the main convergence result for our optimization method. The proof may be adapted from [5] with minor changes, so we skip it here.

Theorem 1. *Suppose $\mathbf{x}^1$ is such that the level set $\{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \leq f(\mathbf{x}^1)\}$ is bounded. Let $\mathbf{x}^j$ be the sequence of serious iterates generated by the bundle trust-region algorithm. Then every accumulation point $\mathbf{x}^*$ of the $\mathbf{x}^j$ is a critical point of (7).* □

4 The case of H_∞-synthesis

The general purpose algorithm 2 is readily applicable to the H_∞-design problem (6), as this is a special case of (7). Since f is the square-root of a maximum eigenvalue function, the ideal model ϕ in algorithm 2 is chosen as in remark 3. There are, however, some particularities in the application of algorithm 2 to (6), on which we comment in this section.

4.1 Stability barrier

The hidden constraint of closed-loop stability may occasionally lead to unacceptable trial points $\mathbf{y}^k$, but this can be avoided by complementing the objective in (6) by the following barrier function: $f(\mathbf{x}) = \max\{\|T_{wz}(K(\mathbf{x}))\|_{\infty,d}, c\|S(K(\mathbf{x}))\|_{\infty,d}\}$, where $S = (I + GK)^{-1}$ is the closed-loop sensitivity function, and $c > 0$ a small constant. It is well-known that $\|S\|_\infty^{-1}$, also known as the modulus margin, is an indicator for the distance of the Nyquist curve to the point -1, and since the Nyquist curve crosses -1 when iterates become destabilizing, the term $\|S\|_\infty$ becomes large as iterates approach the limit of the region of stability. In other words, $c\|S\|_\infty$ has the effect of a barrier function at the boundary of the hidden constraint. Note that the maximum of two H_∞-norms is again a H_∞-norm, i.e., we may still represent the modified objective as $f(\mathbf{x}) = \|T_{w',z'}(K)\|_{\infty,d}$ for a modified channel $w' \to z'$. Note that other closed-loop transfer function, which may contribute a similar complementary barrier effect include $GS = G(I + GK)^{-1}$, $(I - KG)^{-1}$ and $K(I + KG)^{-1}$, see e.g. [8, p.36].

4.2 Exploiting freedom in steps 3 and 4

An important practical aspect of program (6) is to use an adapted initialization of the model $\phi_1(\cdot, \mathbf{x})$ at the beginning of the inner loop, the idea being that in the vast majority of cases the first trial step will then be successful. This is achieved by including not only active frequencies ω from $T_{wz}(K(\mathbf{x}), j\omega)$ at $\mathbf{x}$ in the model, but also branches belonging to secondary peaks, which are susceptible to become active at the next trial step. Selecting near active frequencies of the H_∞-norm is decisive for the quality of the working models ϕ_k and was already discussed in [1] and [4] in the context of the bundle method, and this is shown schematically in Figure 2. We refer to this type of affine functions as anticipated cutting planes, and their integration into the working models is covered by convergence theory as long as they are affine minorants of ϕ.

Remark 9. We next exploit the freedom in step 4 of algorithm 2 in the case of program (6). In case of failure of the solution $\mathbf{y}^k$ of the tangent program it is attractive to use a backtracking linesearch to generate trial steps of the form $\mathbf{z}^k = \mathbf{x} + t(\mathbf{y}^k - \mathbf{x})$, $0 < t < 1$. By convexity, $\Phi_k(\mathbf{x}, \mathbf{x}) - \Phi_k(\mathbf{x} + t(\mathbf{y}^k - \mathbf{x}), \mathbf{x}) \geq t\left(\Phi_k(\mathbf{x}, \mathbf{x}) - \Phi_k(\mathbf{y}^k, \mathbf{x})\right)$, hence every $\mathbf{z}^k$ with $t \geq \theta$ gives automatically a trial point in the sense of step 4, and we only have to check acceptance $\rho_k \geq \gamma$. Even for smaller steps $t < \theta$ it may still be possible to have $f(\mathbf{x}^j) - \Phi_k(\mathbf{z}^k, \mathbf{x}^j) \geq \theta\left(f(\mathbf{x}^j) - \Phi_k(\mathbf{y}^k, \mathbf{x}^j)\right)$, in which case $\mathbf{z}^k$ remains a candidate.

4.3 Performance certificate

The strategy in algorithm 1 is to perform discretization at the level of the system transfer function $G(s)$, and not before, avoiding system reduction or identification. To justify this

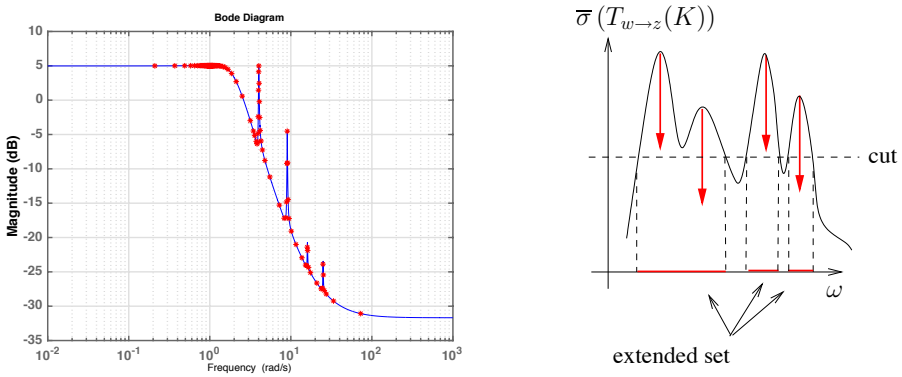

Figure 2: Left: grid $\Omega_{\rm opt}$ based on the criterion (16). Right: selection of extended set of frequencies around active frequencies with primary and secondary peaks (i.e., global and local maxima of $\omega \mapsto \overline{\sigma}(T_{wz}(j\omega))$.

we have to select a discretization $\Omega_{\rm opt}$ for optimization (6) such that the optimal value $f(\mathbf{x}^*) = \|T_{wz}(K(\mathbf{x}^*))\|_{\infty,d}$ is not too far from the true infinite-dimensional value $f_\infty(\mathbf{x}^*) = \|T_{wz}(K(\mathbf{x}^*))\|_\infty$. This hinges on a suitable grid generation technique. This is crucial in our approach, but we stress that this takes place in a low-dimensional space, whereas system reduction and identification need heavy large scale linear algebra machinery, and yet remain on the level of heuristics.

Given a continuously differentiable function $\phi : [0,\infty] \to \mathbb{R}_+$, we want to construct a finite grid $\Omega_{\rm opt}$ such that $\max_{\omega\in[0,\infty]} |\phi(\omega) - P_\phi(\omega)| \le \vartheta$ for a fixed tolerance ϑ, where P_ϕ is the piecewise linear function interpolating $\{\phi(\omega) : \omega \in \Omega_{\rm opt}\}$. We call $M[\cdot,\cdot]$ a first-order bound of $\phi : \mathbb{R} \to \mathbb{R}$ if $|\phi'(\omega)| \le M[\omega^-,\omega^+]$ for all $\omega^- < \omega^+$ and all $\omega \in [\omega^-,\omega^+]$. We need the following preparatory:

Lemma 5. *Suppose we have constructed a grid Ω on $[0,\infty]$ such that for two consecutive nodes $\omega_i, \omega_{i+1} \in \Omega$ and some $\gamma^* \ge \max\{\phi(\omega_i), \phi(\omega_{i+1})\}$ the inequality*

$$M[\omega_i,\omega_{i+1}](\omega_{i+1}-\omega_i) < 2\gamma^* + 2\vartheta - \phi(\omega_i) - \phi(\omega_{i+1}). \tag{16}$$

is satisfied. Then $\phi(\omega) < \gamma^* + \vartheta$ for every $\omega \in [\omega_i,\omega_{i+1}]$.

Proof: Suppose on the contrary that there exists $\omega^* \in [\omega_i,\omega_{i+1}]$ such that $\phi(\omega^*) \ge \gamma^*+\vartheta$. Then the polygon through $\phi(\omega_i)$, $\phi(\omega^*)$, $\phi(\omega_{i+1})$ has length greater than or equal to L, where

$$L = \sqrt{A^2 + (\omega^*-\omega_i)^2} + \sqrt{B^2 + (\omega_{i+1}-\omega^*)^2}$$

with $A = \gamma^* + \vartheta - \phi(\omega_i)$ and $B = \gamma^* + \vartheta - \phi(\omega_{i+1})$. Now $L \ge \ell$, where

$$\ell := \min_{\omega\in[\omega_i,\omega_{i+1}]} \sqrt{A^2 + (\omega-\omega_i)^2} + \sqrt{B^2 + (\omega_{i+1}-\omega)^2} = \sqrt{(A+B)^2 + (\omega_{i+1}-\omega_i)^2},$$

the minimum being attained at $\omega = \frac{\omega_i B + \omega_{i+1} A}{A+B}$. On the other hand the curve $\{(\omega,\phi(\omega)) : \omega \in [\omega_i,\omega_{i+1}]\}$ has length $\mathscr{L} = \int_{\omega_i}^{\omega_{i+1}} \sqrt{1+\phi'(\omega)^2}d\omega \le \sqrt{1+M[\omega_i,\omega_{i+1}]^2}(\omega_{i+1}-\omega_i)$, and $\mathscr{L} \ge L$, so in combining the two estimates we get $\mathscr{L} \ge \ell$, which yields the estimate $\sqrt{1+M[\omega_i,\omega_{i+1}]^2} \ge \sqrt{(A+B)^2/(\omega_{i+1}-\omega_i)^2+1}$. We deduce $M[\omega_i,\omega_{i+1}] \ge (A+B)/(\omega_{i+1}-\omega_i)$, and since $A+B = 2\gamma^* + 2\vartheta - \phi(\omega_i) - \phi(\omega_{i+1})$, this contradicts

(16). □

We would now like to apply this to the function $\phi(\omega) = \overline{\sigma}\left(T_{wz}(K^*, j\omega)\right)$, where $K^* = K(\mathbf{x}^*)$ is the optimal H_∞-controller computed by algorithm 2. For that we have to prove differentiability of ϕ. We have the following:

Lemma 6. [6, Theorem 2.3] *The function ϕ has only a finite number of points of non-smoothness, and in particular, is of class C^2 in the neighborhood of all primary and secondary peaks (all global and local maxima).*

Proof: By [17, Thm. 6.1] the one-parameter family of Hermitian matrices $\omega \mapsto \mathscr{T}(\omega) = T_{wz}(K^*, j\omega)^H T_{wz}(K^*, j\omega)$ has real analytic eigenvalue functions $\lambda_\nu(\omega)$, hence $\phi^2(\omega)$ is a finite maximum of real analytic functions, and then also ϕ because $\phi > 0$. The rest of the argument is now as in [6]. □

Remark 10. In consequence, ϕ is twice continuously differentiable in a neighborhood of each peak, and in particular on a set $\{\omega \in [0,\infty] : \phi(\omega) > \|T_{wz}(K^*)\|_\infty - \vartheta_0\}$ for some $\vartheta_0 > 0$. This means Lemma 5 is applicable.

We use this to construct the grid Ω_{opt} used in (6) as follows. Start with $\omega_0 = 0$. Having constructed ω_i, compute an extrapolation $\omega_i^\sharp > \omega_i$ and obtain $M = \max\{M[\omega_i, \omega] : \omega \in [\omega_i, \omega_i^\sharp]\}$. Then choose $\omega_{i+1} \in (\omega_i, \omega_i^\sharp]$ such that

$$M(\omega_{i+1} - \omega_i) < 2\max\{\phi(\omega_i), \phi(\omega_{i+1})\} + 2\vartheta - \phi(\omega_i) - \phi(\omega_{i+1}). \tag{17}$$

If $G(s)$ is available analytically, then $M[\cdot,\cdot]$ is computable. In the numerical approach we use a finite difference estimation $\phi'(\omega) \approx (\phi(\omega^+) - \phi(\omega))/(\omega^+ - \omega)$. Since ϕ is continuously differentiable near the peak values, this gives excellent results. In our experience, the method rarely leads to grids with more than a few hundred of nodes, which allows an efficient solution of the optimization program. A typical example is shown in Figure 2.

The following result justifies our method theoretically.

Theorem 2. *If $0 < \vartheta \le \vartheta_0$ and if a first-order bound $M[\cdot,\cdot]$ for $\phi = \overline{\sigma}\left(T_{wz}(K^*, \cdot)\right)$ in tandem with rule* (17) *is used in step* 4 *of algorithm* 1 *to construct Ω_{opt}, then the gain γ^* achieved by the solution K^* of* (6) *is* certified *to satisfy $\gamma^* \ge \|T_{wz}(K^*)\|_\infty - \vartheta$.*

Proof: Since $\phi(\omega) = \overline{\sigma}\left(T_{wz}(K^*, j\omega)\right)$ and $\|T_{wz}(K^*)\|_\infty = \max_{\omega\in[0,\infty]} \phi(\omega)$, we have to show that $\gamma^* \ge \phi(\omega) - \vartheta$ for every i and all $\omega \in [\omega_i, \omega_{i+1}]$, where ω_i are the nodes of the grid Ω_{opt} constructed in step 4 of algorithm 1 based on the rule (17). Since γ^* is the gain achieved by the solution K^* of (6) on that grid, it satisfies $\gamma^* = \max_i \phi(\omega_i)$. Hence $\gamma^* \ge \max\{\phi(\omega_i), \phi(\omega_{i+1})\}$, and so condition (16) of Lemma 5 is satisfied. Since by Lemma 6 we may apply Lemma 5 to ϕ, we obtain the conclusion $\phi(\omega) \le \max\{\phi(\omega_i), \phi(\omega_{i+1})\} + \vartheta \le \gamma^* + \vartheta$ on $[\omega_i, \omega_{i+1}]$, as claimed. □

Remark 11. In rule (17) we apply (16) with $\gamma^* = \max\{\phi(\omega_i), \phi(\omega_{i+1})\}$ on each interval $[\omega_i, \omega_{i+1}]$. When it comes to just certifying the optimal value $f(\mathbf{x}^*) = \|T_{wz}(K(\mathbf{x}^*))\|_\infty = \phi(\omega^*)$ in step 6 of algorithm 1, then we can construct an even coarser grid by applying (16) with $\gamma^* = \phi(\omega^*)$ the same for all $[\omega_i, \omega_{i+1}]$. Namely, our grid can be very coarse at frequencies ω where $\phi(\omega) \ll \phi(\omega^*)$, and still capture sharp peaks, as illustrated in Figure 2. We refer to this as a verification grid Ω_{ver}. According to our experience, the outlined method to construct Ω_{opt} is well-adapted to discretize the controller design problem.

We can further exploit lemma 5 to obtain information on how close the values γ^* of (6) and γ_∞ of the infinite-dimensional program (5) are. Writing as before $f(\mathbf{x}) = \|T_{wz}(K(\mathbf{x}))\|_{\infty,d}$ for the discrete H_∞-norm on Ω_{opt}, and $f_\infty(\mathbf{x}) = \|T_{wz}(K(\mathbf{x}))\|_\infty$ for the true H_∞-norm, we compare the discretized H_∞-program $\min_{\mathbf{x}} f(\mathbf{x})$, i.e., (6), to the underlying infinite-dimensional $\min_{\mathbf{x}} f_\infty(\mathbf{x})$, i.e. (5).

Corollary 1. *Let $\mathbf{x}_\infty$ be a local minimum of* (5) *with value γ_∞, and $\mathbf{x}^*$ a local minimum of* (6) *with value γ^*. Suppose a first-order bound in tandem with rule* (17) *has been used in step* 6 *of algorithm* 1. *Then if $\mathbf{x}^*$, $\mathbf{x}_\infty$ are within neighborhoods of local optimality of each other, we have $f(\mathbf{x}_\infty) \geq f(\mathbf{x}^*) \geq f_\infty(\mathbf{x}^*) - \vartheta \geq f_\infty(\mathbf{x}_\infty) - \vartheta \geq f(\mathbf{x}_\infty) - \vartheta$.*

Proof: Indeed, $f(\mathbf{x}_\infty) \geq f(\mathbf{x}^*)$ because $\mathbf{x}^*$ is a minimum of f on a neighborhood $U(\mathbf{x}^*)$, and $\mathbf{x}_\infty \in U(\mathbf{x}^*)$ by hypothesis. Next $f(\mathbf{x}^*) \geq f_\infty(\mathbf{x}^*) - \vartheta$ by Lemma 5, because construction of the grid uses the bound $M[\cdot,\cdot]$ and rule (16). Next $f_\infty(\mathbf{x}^*) \geq f_\infty(\mathbf{x}_\infty)$, because $\mathbf{x}_\infty$ is a minimum of f_∞ on a neighborhood $U(\mathbf{x}_\infty)$, and $\mathbf{x}^* \in U(\mathbf{x}_\infty)$ by hypothesis. The last inequality is satisfied because $f \leq f_\infty$. □

This means comparable locally optimal values of the infinite dimensional H_∞-program (5) and its approximation (6) differ by at most ϑ, our apriori chosen tolerance. Since most of the time our algorithm finds the global minimum of (6), this is a very useful result in practice, as it determines the value of the infinite dimensional H_∞-program (5) within a prior tolerance level ϑ.

The argument remains valid if $\mathbf{x}_\infty$, $\mathbf{x}^*$ are only approximate local minima, say up to the same tolerance ϑ in the values. Then we get the chain $f(\mathbf{x}_\infty) \geq f(\mathbf{x}^*) - \vartheta \geq f_\infty(\mathbf{x}^*) - 2\vartheta \geq f_\infty(\mathbf{x}_\infty) - 3\vartheta \geq f(\mathbf{x}_\infty) - 3\vartheta$, so here our approximation (6) gives the correct value up to an error of 3ϑ in the values.

5 Applications

In this section, we apply our method to several challenging studies in control of infinite dimensional system, and in particular, to boundary and distributed control of systems of parabolic partial differential equations.

5.1 Computation of $G(s)$ in boundary control

We illustrate how our procedure is applied to boundary control of parabolic PDEs. Consider a boundary problem of the form

$$\begin{aligned} z_t(x,t) = \sum_{|\alpha|,|\beta|\leq m} (-1)^{|\alpha|} D^\alpha (a_{\alpha\beta}(x) D^\beta z)(x,t) = 0 \qquad & (x,t) \in Q \times [0,\infty) \\ D_\nu^{i-1} z(x,t) = \mathcal{U}_i(x,t) \qquad & x \in \partial Q,\, i = 1,\ldots,m, \end{aligned} \tag{18}$$

where $\mathcal{U}_i$ are abstract controls acting on the boundary ∂Q. Here for convenience Q is bounded open with ∂Q a compact orientable C^∞-manifold, the coefficients are $a_{\alpha,\beta} \in C^\infty(\overline{Q})$, and uniform ellipticity $\sum_{|\alpha|,|\beta|\leq m} a_{\alpha\beta}(x)\xi^\alpha\xi^\beta \geq c|\xi|^2$ is assumed for $x \in Q$. Then by [28] problem (18) may be represented in the abstract form (1), where however the input space U is potentially still infinite dimensional. To comply with our assumption

that $K(s)$ should be finite-rank, i.e., that input and output spaces U, Y should be finite-dimensional, we select basis functions ϕ_{ik} on the boundary ∂Q and replace the boundary control action $\mathcal{U}$ in (18) by a finite-dimensional version

$$D_\nu^{i-1} z(x,t) = \sum_{k=1}^{N} \phi_{ki}(x) u_{ik}(t), \quad x \in \partial Q,\ i = 1, \dots, m,$$

which now has input space $U \simeq \mathbb{R}^{Nm}$. A finite-dimensional output space $Y \simeq \mathbb{R}^p$ could be obtained by taking measurements of the form

$$y_i(t) = \int_Q \psi_i(x) z(x,t)\, dx, \quad i = 1, \dots, p,$$

with another set of basis functions ψ_i on Ω representing sensors. For one-dimensional Q point evaluations on ∂Q are possible. This case will be used in our numerical experiments.

Referring to [28, 10, 29] for the correct setup of (1), we directly pass to the computation of $G(s)$. Laplace transforming (18) with initial condition $z(x,0) = 0$ leads for fixed $s \in \mathbb{C}$ to the elliptic boundary value problem

$$\begin{aligned} sz(x,s) &= \sum_{|\alpha|,|\beta| \le m} (-1)^{|\alpha|} D^\alpha (a_{\alpha\beta}(x) D^\beta z)(x,s) = 0 \qquad x \in Q \\ D_\nu^{i-1} z(x,s) &= \sum_{k=1}^{N} \phi_{ki}(x) u_{ik}(t) \qquad x \in \partial Q,\ i = 1, \dots, m. \end{aligned} \tag{19}$$

Then $G_{ikr}(s)$ is obtained by solving (19) with $u_{ik} = 1$ $u_{i',k'} = 0$ for $(i',k') \neq (i,k)$, and by computing $y_r(s) = \int_Q \psi_r(x) z(x, j\omega)\, dx$. For the one-dimensional case $Q = [0,1]$, point evaluations $y_r(s) = z_r(0,s)$, $y_r(s) = z_r(1,s)$, or linear combinations of those, are possible.

Remark 12. Computation of $G'(s)$ can also be obtained by solving an elliptic boundary value problem, which is (19) differentiated with respect to s. Since K is known explicitly, this is useful when computing the bound $M[\cdot,\cdot]$ of ϕ in Lemma 5.

Remark 13. In those cases where computations are not performed formally, a high spatial resolution is used to solve (19) accurately. One such solve can then be interpreted as a function evaluation $G_{ikr}(s)$. If carried out numerically, we perform the computation of $G(s)$ for fixed $s = j\omega$ with the highest spatial discretization available in our setup. Typically, this is at least as accurate spatially as our final simulation of the closed-loop system. Since the number of inputs and outputs is not very large, pre-computing $G(s)$ will not seriously burden the overall performance of algorithm 1. Since pre-computing $G(j\omega)$ is done off-line, it neither impedes the optimization phase, nor the plant modeling phase.

5.2 Reaction-convection-diffusion equation

We consider a non-linear reaction-convection-diffusion equation with Danckwaerts and von Neumann boundary conditions

$$\begin{aligned} \frac{\partial C(z,t)}{\partial t} &= D\frac{\partial^2 C(z,t)}{\partial z^2} - U(t)\frac{\partial C(z,t)}{\partial z} - kC(z,t) \qquad (z,t) \in [0,L] \times [0,\infty) \\ D\frac{\partial C(0,t)}{\partial z} - U(t)(C(0,t) - C_{\text{in}}) &= 0, \qquad \frac{\partial C(L,t)}{\partial z} = 0. \end{aligned} \tag{20}$$

The process represents a chemical reaction in a cylindrical plug flow reactor with time-varying flow velocity $U(t)$, constant axial dispersion D, and constant reaction rate k. The dynamics of the reaction $A \xrightarrow{k} B$ are described by the spatially and temporally varying concentration $C(z,t)$ of reactant A, the concentration of product B being a dependent state. Using online measurement $y(t) = C(L,t) - C_{ss}(L)$ of the concentration of ingredient A at the outflow position $z = L$ we steer the plug flow velocity $U(t)$ to maintain the process in steady-state U_{ss}, $C_{ss}(z)$, y_{ss}, while attenuating measurement noise and a disturbance of the flow velocity, and to enable speedy tracking of set-point changes in the steady-state concentration. We refer to [29] or [10, Example 3.3.5] for the correct setup of this problem as a Hilbert space linear system (1).

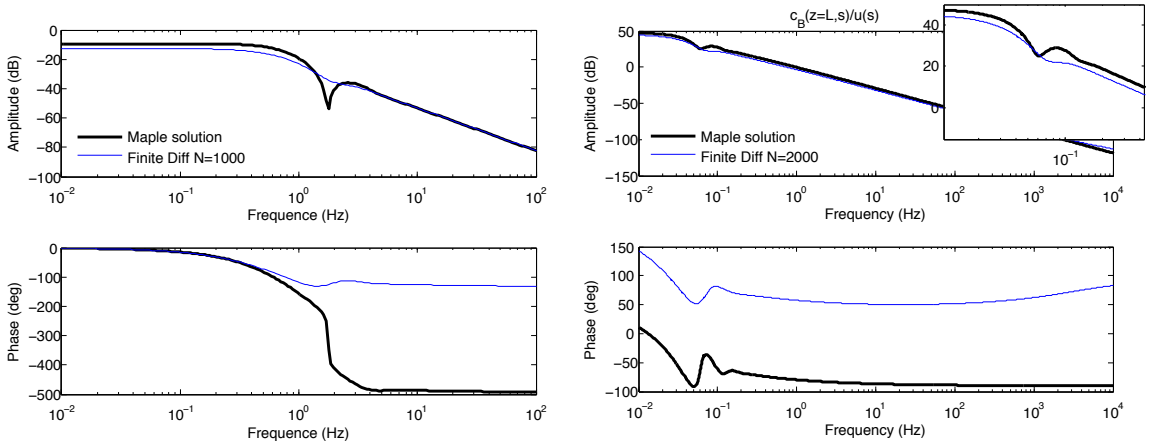

Figure 3: Bode plot of infinite-dimensional transfer function $G(s)$ compared with finite-difference based $G_{fd}(s)$. Left shows study 1 computed with Maple, compared to finite-differences of order $N = 1000$. Even with 5000 states the transmission zero at frequency ω1.6Hz, which is missed by the discretization of order $N = 1000$. Right shows Van de Vusse study with $G(s)$ computed via numerical Maple solve of (29), compared to finite-differences of order $N = 2000$.

Fixing a steady-state flow velocity U_{ss}, we compute the corresponding steady-state concentration $C_{ss}(z)$ by solving the one-dimensional boundary value problem

$$\begin{aligned} DC''_{ss}(z) - U_{ss}C'_{ss}(z) - kC_{ss}(z) &= 0 \\ DC'_{ss}(0) - U_{ss}(C_{ss}(0) - C_{\text{in}}) = 0, \qquad C'_{ss}(L) &= 0. \end{aligned} \tag{21}$$

Linearization about steady-state with $U(t) = U_{ss} + u(t)$ and $C(z,t) = C_{ss}(z) + c(z,t)$ leads now to the linearized boundary and distributed control problem

$$\begin{aligned} c_t(z,t) &= Dc_{zz}(z,t) - U_{ss}c_z(z,t) - C'_{ss}(z)u(t) - kc(z,t) \\ Dc_z(0,t) - U_{ss}c(0,t) + (C_{in} - C_{ss}(0))u(t) = 0, \qquad c_z(L,t) &= 0. \end{aligned} \tag{22}$$

The linearized output is $y(t) = c(L,t)$. In this case steady-state $C_{ss}(z)$ and transfer function $G(s) = y(s)/u(s)$ of the linearized equation can be computed formally using Maple. We obtain

$$C_{ss}(z) = C_{in}\, b\, \frac{(b-f)e^{\frac{f(1-z/L)-bz/L}{2a}} - (b+f)e^{\frac{f(z/L-1)-bz/L}{2a}}}{(-bf-2ak-b^2)e^{\frac{-f}{2a}} - (bf-2ak-b^2)e^{\frac{f}{2a}}} \tag{23}$$

where

$$a = \frac{D}{L^2}, \quad b = \frac{-U_{ss}}{L}, \quad f = \sqrt{b^2 + 4ak}. \tag{24}$$

Then the transfer $u(s) \to c(z,s)$ is obtained analytically as $C_{in}\,\frac{P_1(z,s)}{P_2(z,s)}$, where

$$\begin{aligned}
P_1(z,s) = &\left[2La(s+k)\int_0^1 f_1(x)dx - LT_2\int_0^{\frac{z}{L}} f_2(x)dx - T_3(m-1)\right]e^{T_4(z)} + \\
&\left[LT_6\int_0^{\frac{z}{L}} f_1(x)dx + 2La(s+k)\int_0^1 f_2(x)dx + T_8(m-1)\right]e^{T_9(z)} + \\
&e^{T_5(z)}LT_2\int_1^{\frac{z}{L}} f_1(x)dx + e^{T_7(z)}LT_6\int_{\frac{z}{L}}^1 f_2(x)dx \\
P_2(z,s) = &\; LT_1(T_2e^{\frac{T_1}{2a}} + T_6e^{\frac{-T_1}{2a}})
\end{aligned}$$

with

$$\begin{aligned}
&T_1 = \sqrt{b^2+4a(s+k)}, \quad T_2 = bT_1 + 2a(-k-s) - b^2, \quad T_3 = bT_1 + 4a(-k-s) - b^2, \\
&T_4(z) = \frac{\frac{z}{L}b + (\frac{z}{L}-1)T_1}{-2a}, \quad T_5(z) = \frac{-\frac{z}{L}b + (\frac{z}{L}+1)T_1}{2a}, \quad T_6 = bT_1 + 2a(s+k) + b^2, \\
&T_7(z) = \frac{\frac{z}{L}b + (\frac{z}{L}+1)T_1}{2a}, \quad T_8 = bT_1 + 4a(s+k) + b^2, \quad T_9(z) = \frac{-\frac{z}{L}b + (\frac{z}{L}-1)T_1}{2a} \\
&f_1(z) = C_{ss_z}(z)e^{\frac{(b-T_1)z}{2aL}}, \quad f_2(z) = C_{ss_z}(z)e^{\frac{(b+T_1)z}{2aL}}, \quad m = \frac{C_{ss}(0)}{C_{in}}.
\end{aligned}$$

The transfer $u \to y$ is then obtained at $z = L$.

Adopting numerical values $D = 1.05\,\mathrm{m}^2/\mathrm{min}$, $U_{ss} = 1.24\,\mathrm{m/min}$, $C_{\mathrm{in}} = 0.5\,\mathrm{mol/m}^3$, $k = 0.25\,\mathrm{m}^3/\mathrm{mol}$, and $L = 6.36\,\mathrm{m}$ from a study in [14], we can compare the the infinite-dimensional transfer function $G(s)$ with a finite-difference approximation $G_{\mathrm{fd}}(s)$. Figure 3 left shows the comparison of $G(s)$ and $G_{\mathrm{fd}}(s)$.

The scheme for synthesis is shown in Figure 4 and uses the filters in Figure 5 (right), which are defined as

$$W_e(s) = \frac{0.00001s + 5}{s + 0.25}, \qquad W_n(s) = \frac{0.00125s^2 + 0.00035s + 0.00005}{0.000025s^2 + 0.007s + 1}, \qquad W_u = 0.1.$$

The controller structure $\mathscr{K}_{\mathrm{pi}}$ includes SISO PI-controllers with two parameters $K(s) = k_p + \frac{k_i}{s}$, so $\mathbf{x} = (k_p, k_i)$. With the mixed performance-robustness channel $w = (r,n) \to z = (z_e, z_u) = (W_e e, W_u u)$ we have now defined the objective $\|T_{wz}(K(\mathbf{x}))\|_\infty$ of our problem, and according to section 4.1 the objective f is complemented by the barrier function $c\|S(K(\mathbf{x}))\|_\infty$. Note that in the scheme of Figure 4 the sensitivity function S equals the unfiltered closed-loop transfer function T_{re}, so altogether with regard to the general form (6) in this study the channel $(r,n) \to (z_e, z_u, ce)$ is optimized, where $c = 0.2$. The optimal solution obtained by algorithm 2 is $\mathbf{x}^* = (k_i^*, k_p^*) = (9.93 \cdot 10^{-5}, 7.13 \cdot 10^2)$.

Remark 14. The functional analytic setup for (22) is as follows. On the Hilbert space $H = \mathcal{L}^2([0,L])$ define the differential operator $\mathscr{A} = D\frac{d^2}{dz^2} - U_{ss}\frac{d}{dz} - k$ with domain $D(\mathscr{A}) = \{h \in H : h, \frac{d}{dz}h \text{ a.c.}, \frac{d^2}{dz^2}h \in \mathcal{L}^2([0,L])\}$, and the boundary control operator by $\mathscr{P} = \left(D\frac{d}{dz} - U_{ss}\right)/(C_{ss}(0) - C_{in})$ with domain $D(\mathscr{P}) = \{h \in H : \frac{d}{dz}h \in \mathcal{L}^2([0,L]), h \text{ a.c.}\}$,

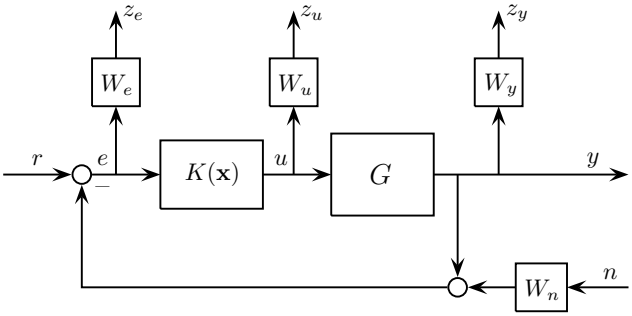

Figure 4: Scheme for synthesis. In studies 5.2 and 5.3 the H_∞-norm of the performance channel $(r, n) \to (z_e, z_u)$ is minimized, which assures that the system reacts to a set-point change r, and attenuates noise with bandwidth specified by the filter W_n. Tracking e is in the low-frequency range specified by W_e. In study 5.4 the channel $r \to (z_e, z_y)$ is optimized.

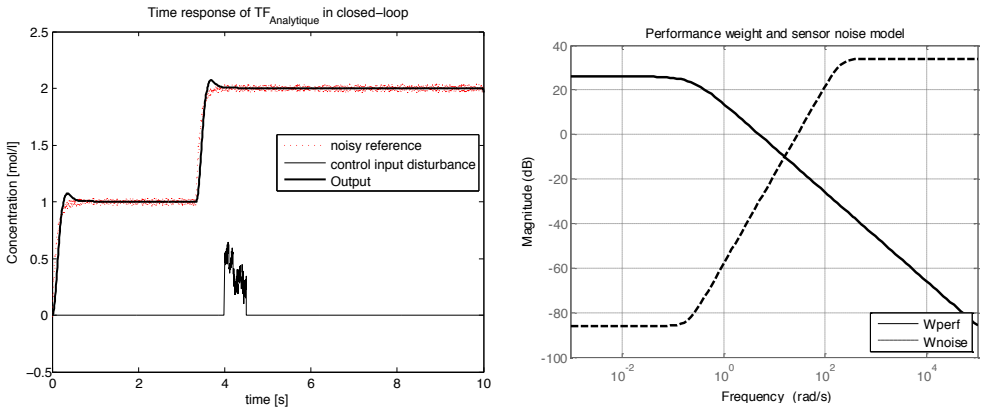

Figure 5: Left: Time response to a noisy reference signal of analytic closed loop. Right: Filter W_e for synthesis and noise filter W_n.

so that $D(\mathscr{A}) \subset D(\mathscr{P})$. One defines $\alpha = (C_{ss}(0) - C_{in}) \,/\, (D - U_{ss}/2)$ and the function $b(z) = \frac{\alpha}{2}(z+1)^2$, then the multiplication operator $Bu = b(z)u$ satisfies $\mathscr{P}(Bu) = u$, and now we have a boundary control problem in the sense of [10, Def. 3.3.2], which can be brought to the form (1).

5.3 Van de Vusse reactor

The following more challenging study uses a parallel *Van de Vusse process*

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$
$$A + A \xrightarrow{k_3} D$$

which we want to operate in continuous mode in an isothermal reactor. In [32] Van de Vusse considers A = cyclopentadiene, B = cyclopentenol, C = cyclopentanediol, D = dicyclopentadiene. The desired output product is B, the inlet is the primary product A, whereas C, D are waste products. The reaction takes place in a steady-plug-flow cylindrical chemical reactor of length L, through which a fluid with velocity U is flowing,

where ingredients are mixing axially with dispersion coefficient D, while the now non-linear chemical reaction is described by the rates k_i. In ODE-based models the axial dispersion is often neglected, or replaced by a singular perturbation approach [12]. Here we discuss the full non-linear model. For the functional analytic setup of the problem see again [29].

Assuming radially homogeneous conditions in the tube, the system can be described by one spatial dimension z, and the reaction for ingredients A and B is governed by the following diffusion-convection-reaction system of parabolic PDEs:

$$\begin{aligned}\frac{\partial C_A(z,t)}{\partial t} &= D\frac{\partial^2 C_A(z,t)}{\partial z^2} - U\frac{\partial C_A(z,t)}{\partial z} - k_1 C_A(z,t) - k_3 C_A^2(z,t)\\ \frac{\partial C_B(z,t)}{\partial t} &= D\frac{\partial^2 C_B(z,t)}{\partial z^2} - U\frac{\partial C_B(z,t)}{\partial z} + k_1 C_A(z,t) - k_2 C_B(z,t)\end{aligned} \tag{25}$$

for $(z,t) \in [0,L] \times [0,\infty)$, with Danckwaerts and von Neumann boundary conditions

$$\begin{aligned}&D\frac{\partial C_A(0,t)}{\partial z} - U\left(C_A(0,t) - C_{Ain}\right) = 0, \quad D\frac{\partial C_B(0,t)}{\partial z} - UC_B(0,t) = 0\\ &\frac{\partial C_A(L,t)}{\partial z} = 0, \qquad \frac{\partial C_B(L,t)}{\partial z} = 0\end{aligned} \tag{26}$$

for all $t \in [0,\infty)$. The meaning of these boundary conditions at $z = 0$ is that as soon as the feed enters the reactor at $z = 0$, it will be diluted by the axial mixing caused by the flow. At $z = L$ we have Neumann boundary conditions, which simply require that the concentration stops changing at the point where the flow leaves the reactor.

The goal of the study is to operate the reactor at a steady-state flow U_{ss} leading to a steady outflow $C_{Bss}(L)$ of product B at the outlet of the reactor. This steady-state flow has to be controlled by feedback, where we have the possibility to act on the velocity $U(t) = U_{ss} + u(t)$, and where we use the deviation $y(t) = C_B(L,t) - C_{Bss}(L)$ from the steady-state production as our online measurement at the outlet. It is assumed that changes of the axial flow velocity do not affect the axial dilution D assumed constant. Control has to maintain a stable steady-state, attenuate measurement noise and disturbances at the inflow, and enable the system to react to set-point changes in the flow velocity U.

Our procedure starts by computing the steady-state, which leads to solving the system of ODEs

$$\begin{aligned}DC''_{Ass}(z) - U_{ss}C'_{Ass}(z) - k_1 C_{Ass}(z) - k_3 C^2_{Ass}(z) = 0\\ DC''_{Bss}(z) - U_{ss}C'_{Bss}(x) + k_1 C_{Ass}(z) - k_2 C_{Bss}(z) = 0\end{aligned} \tag{27}$$

with steady-state boundary conditions

$$\begin{aligned}DC'_{Ass}(0) - U_{ss}\left(C_{Ass}(0) - C_{Ain}\right) = 0, \quad DC'_{Bss}(0) - U_{ss}C_{Bss}(0) = 0,\\ C'_{Ass}(L) = 0, \quad C'_{Bss}(L) = 0.\end{aligned} \tag{28}$$

This defines a mapping $U_{ss} \to (C_{Ass}(L), C_{Bss}(L))$, which allows us to see what flow U_{ss} gives the largest output. In the numerical study we fix $U_{ss} = 6.175e\text{-}3$, which leads to the solutions in Figure 6.

Remark 15. According to our strategy we assume that $C_{Ass}(z), C_{Bss}(z)$ are computed with a very high precision, representing a quasi-analytic solution. Indeed, it is in principle possible to solve the steady-state system formally using Taylor series expansions, but since the difference with a high precision numerical solution is marginal, we proceed with the numerical approach.

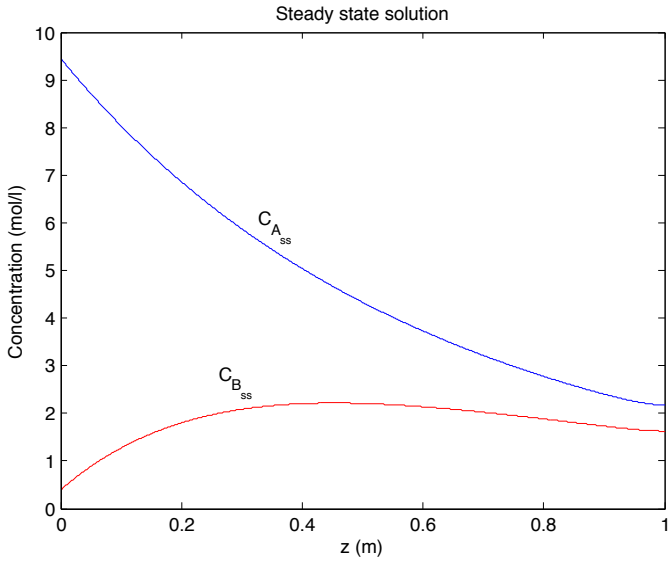

Figure 6: Steady-state flow computed by Maple for $U_{ss} = 6.175e\text{-}3$. The parameters were `maxmesh`=1e6, `abserr` = 1e-7. The plots show $C_{Ass}(z_i)$, $C_{Bss}(z_i)$ for 500 equidistant nodes on $[0, 1]$.

Once the steady-state is computed, we linearize the system by putting $C_A(z,t) = C_{Ass}(z) + c_A(z,t)$, $C_B(z,t) = C_{Bss}(z) + c_B(z,t)$, $U(t) = U_{ss} + u(t)$ with off-sets c_A, c_B, u, which leads to the linearized system

$$\begin{aligned}
\frac{\partial c_A(z,t)}{\partial t} &= D\frac{\partial^2 c_A(z,t)}{\partial z^2} - U_{ss}\frac{\partial c_A(z,t)}{\partial z} - (k_1 + 2k_3 C_{Ass}(z))c_A(z,t) - \frac{\partial C_{Ass}(z)}{\partial z}\,u(t) \\
\frac{\partial c_B(z,t)}{\partial t} &= D\frac{\partial^2 c_B(z,t)}{\partial z^2} - U_{ss}\frac{\partial c_B(z,t)}{\partial z} + k_1 c_A(z,t) - k_2 c_B(z,t) - \frac{\partial C_{Bss}(z)}{\partial z}u(t)
\end{aligned} \tag{29}$$

with left boundary conditions

$$\begin{aligned}
D\frac{\partial}{\partial z}c_A(0,t) - U_{ss}c_A(0,t) + (C_{Ain} - C_{Ass}(0))u(t) = 0 \\
D\frac{\partial}{\partial z}c_B(0,t) - U_{ss}c_B(0,t) - C_{Bss}(0)u(t) = 0.
\end{aligned} \tag{30}$$

and right boundary conditions

$$\frac{\partial c_A(L,t)}{\partial z} = 0, \qquad \frac{\partial c_B(L,t)}{\partial z} = 0. \tag{31}$$

The measured output is $y(t) = c_B(L,t)$. The transfer function $G(s) = y(s)/u(s)$ is in principle also available analytically, but we continue with the high precision numerics strategy. We first compute the transfers $c_A(z,s)/u(s)$ and $c_B(z,s)/u(s)$, which we obtain by Laplace transforming the linearized system. This leads to the linear boundary value problem

$$\begin{aligned}
D(c_A)_{zz}(z,s) - U_{ss}(c_A)_z(z,s) - (k_1 + 2k_3 C_{Ass}(z) - s)c_A(z,s) - (C_{Ass})_z(z)u(s) = 0 \\
D(c_B)_{zz}(z,s) - U_{ss}(c_B)_z(z,s) + k_1 c_A(z,s) - (k_2 + s)c_B(z,s) - (C_{Bss})_z(z)u(s) = 0
\end{aligned}$$

with Laplace transformed boundary conditions

$$\begin{aligned} D(c_A)_z(0,s) - U_{ss}c_A(0,s) + (C_{Ain} - C_{Ass}(0))u(s) &= 0, \\ D(c_B)_z(0,s) - U_{ss}c_B(0,s) + (C_{Bin} - C_{Bss}(0))u(s) &= 0 \\ D(c_A)_z(L,s) &= 0 \\ D(c_B)_z(L,s) &= 0 \end{aligned}$$

Solving this boundary value problem for fixed $s = j\omega$ with $u(s) \equiv 1$ gives the value $G(j\omega)$ of the transfer function $y(s) = c_B(L,s) = G(s)u(s)$. This is in fact the method presented in section 5.1. The magnitude of $G(s)$ is shown in Figure 7 (right).

Note that for every fixed $s = j\omega$ we have to solve a static elliptic problem associated with the dynamic equation (29), and we perform these computations with the finest scale available, so that $G(s)$ is essentially lossless. In Figure 3 it can be seen that in order to achieve the accuracy in $G(s)$ with a finite-difference discretization $G_{fd}(s)$ we would requires at least 2000 states. So in state-space we would have to perform synthesis on a system of order 2000 to be sure that we do not lose information. This size is beyond existing synthesis techniques. In the same vein, any approach based on system reduction would run the risk of losing information in forming a transfer function based on a reduced model.

constant	denomination	numerical value	unit
k_1	exchange rate $A \to B$	1.39×10^{-2}	s^{-1}
k_2	exchange rate $B \to C$	2.78×10^{-2}	s^{-1}
k_3	exchange rate $A + A \to D$	2.77×10^{-4}	l/mol/s
D	axial dispersion coefficient	3.33×10^{-4}	m^2/s
U_{ss}	steady-state velocity	6.175×10^{-3}	m/s
C_{Ain}	inlet concentration of component A	10.0	mol/l
L	length of reactor	1.0	m

The spatiotemporally, spatially, and temporally varying quantities are

Quantity	denomination	unit
$C_A(z,t)$	concentration of reactant A	mol/l
$C_B(z,t)$	concentration of reactant B	mol/l
$C_{Ass}(z)$	steady-state concentration of A	mol/l
$C_{Bss}(z)$	steady-state concentration of B	mol/l
$U(t)$	superficial velocity	m/s

In the synthesis scheme of Figure 4 we use again the channel $(r,n) \to (z_e, z_u)$, now with the filters

$$W_e(s) = \frac{10^{-5}s + 1.502}{s + 0.07509}, \quad W_n(s) = \frac{0.00125s^2 + 0.00035s + 5 \cdot 10^{-5}}{2.5 \cdot 10^{-5}s^2 + 0.007s + 1}, \quad W_u = 0.1.$$

Optimization is now over the class $\mathscr{K}_3$ of third order controllers, which leads to a tunable vector $\mathbf{x} \in \mathbb{R}^{14}$, as the system matrix A_K of the controller (2) is parametrized in tridiagonal form. The channel is again complemented by the barrier $c\|S(K(\mathbf{x}))\|_\infty$, where $S = T_{re}$

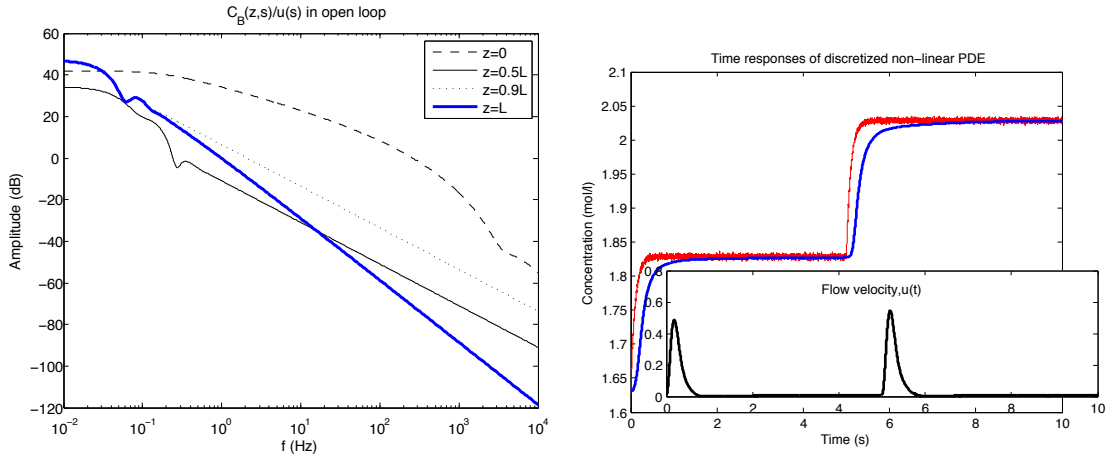

Figure 7: Left: Comparison of transfer function magnitude for different positions of the sensor. The blue curve ($z = L$) is the one chosen in the experiment. Right: Response of closed-loop system to a noisy reference for optimal controller computed by algorithm 1.

and $c = 0.2$. The optimal H_∞-controller $K(\mathbf{x}^*)$ with $\mathbf{x}^* \in \mathbb{R}^{14}$ computed by algorithm 1 is obtained in the form (2) as

$$A_K = \begin{bmatrix} -.8946 & -36.65 & 0 \\ -1.324 & -50.27 & -18.4 \\ 0 & 82.2 & 12.14 \end{bmatrix}, B_K = \begin{bmatrix} -8.32 \\ 4.728 \\ 2.019 \end{bmatrix}, C_K = \begin{bmatrix} -1.839 & -3.129 & -9.019 \end{bmatrix}, D_K = -.08686.$$

the optimal gain being $f(\mathbf{x}^*) = 0.464$. The final number of nodes required for a certified result was $|\Omega_{\text{opt}}| = 101$, where one update of the grid in step 6 of algorithm 1 was needed. The study ends with a non-linear simulation of the optimal controller. In Figure 7 (right) the response if the nonlinear system to a noisy reference signal is displayed.

5.4 Cavity flow

We consider a challenging cavity flow study from [33], where the infinite-dimensional transfer function is available analytically and of the form

$$G(s) = \frac{e^{-\tau_1 s}}{p_2(s) + q_2(s)e^{-\tau_2 s} + ce^{-\tau_3 s}}$$

with quadratic polynomials $p_2(s)$, $q_2(s)$ and delay parameters $\tau_i > 0$. Figure 8 (left) shows the magnitude plot of G in blue, indicating a large number of resonant peak frequencies.

As H_∞-objective we have chosen the channel $\|(W_1S, W_2T)\|_\infty$ with S the closed-loop sensitivity function, T the complementary sensitivity functions, and with the frequency weighing filters $W_1(s) = (0.01s + 177.4)/(s + 50.68)$, $W_2(s) = (100s + 500)/(s + 50000)$. Optimization (6) is over the class $\mathscr{K}_2$ of order 2 controllers, which features 9 tunable parameters. The optimal controller $K(\mathbf{x}^*)$ with $\mathbf{x}^* \in \mathbb{R}^9$ computed by algorithm 1 is given in transfer function form as $K^*(s) = (0.5069s^2 + 119.2s + 1419)/(s^2 + 308s + 1.276e04)$, and achieves a gain of $\gamma^* = 1.937$, which improves over the value 1.948 obtained in [33] using a coprime factorization approach. The final grid size is $|\Omega_{\text{opt}}| = 382$. The fact that the class $\mathscr{K}_2$ with which we achieve $\gamma^* = 1.937$ is much simpler than the infinite-dimensional controller structure used in [33] explains why our novel approach represents a significant improvement in this study.

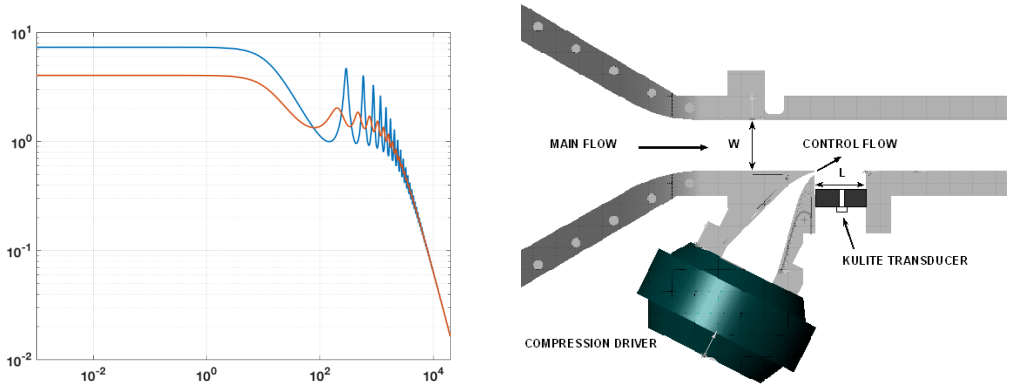

Figure 8: Cavity flow study from [33]. Left image shows magnitude of $G(j\omega)$ (blue), and of GS in closed loop (red). Sharp peaks are significantly reduced by the synthesized control law.

6 Conclusion

We have presented a novel method to compute H_∞-controllers for infinite dimensional systems and in particular for boundary and distributed control of PDEs. At the core our approach uses a non-smooth trust-region bundle algorithm to solve a frequency discretized version of the infinite-dimensional problem. The method was justified theoretically and tested numerically on a reaction-convection-diffusion equation, on a Van de Vusse reactor, and for control of a cavity flow. A convergence certificate for the non-smooth trust-region algorithm under Kiwiel's aggregation rule was proved, allowing to limit the number of cuts in the tangent program (8) to any fixed number $N \geq 3$, and answering in the affirmative a question left open in [5].

References

[1] P. Apkarian, D. Noll. Nonsmooth H_∞ synthesis. IEEE Trans. Automat. Control 51(1) (2006), 71-86.

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

[3] P. Apkarian, D. Noll. Structured H_∞-control of infinite-dimensional systems. Int. J. Robust. Nonlin. Control, to appear.

[4] P. Apkarian, D. Noll, O. Prot. A proximity control algorithm to minimize non-smooth and non-convex semi-infinite maximum eigenvalue functions. Journal of Convex Analysis, vol. 16, 2009, pp. 641 - 666.

[5] P. Apkarian, D. Noll, L. Ravanbod. Nonsmooth bundle trust-region algorithm with applications to robust stability. Set-Valued and Variational Analysis, vol. 24, no. 1, 2016, pp. 115-148.

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

[7] S. Boyd, V. Balakrishnan, P. Kabamba. A bisection method for computing the H_∞-norm of a transfer matrix and related problems. Mathematics of Control, Signals, and Systems, 2 (1989), pp. 207-219.

[8] S. Boyd, C. Barratt. Linear controller design. Limits of performance. Prentice Hall, 1991.

[9] J. Cullum, W. Donath, P. Wolfe. The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices. Math. Programming Stud., 3 (1975), pp. 35-55.

[10] R.F. Curtain, H.J. Zwart. An introduction to infinite-dimensional linear system theory. Springer Texts in Applied Mathematics 21, 1995.

[11] M.N. Dao, J. Gwinner, D. Noll, N. Ovcharova. Nonconvex bundle method with application to a delamination problem Computational Optimization and Applications, vol. 65, no. 1, 2016, pp. 173 - 203.

[12] D. Dochain, B. Bouaziz. Approximation of the dynamical model of fixed bed reactors via a singular perturbation approach. Mathematics and Computers in Simulation, 37, 1994, 165 – 172.

[13] K.-J. Engel, R. Nagel. One-parameter semigroups for linear evolution equations. Springer Graduate Texts in Mathematics 194, 2000.

[14] H.S. Fogler. Elements of Chemical Reaction Engineering, Fourth Edition. Prentice Hall International Series in the Physical and Chemical Engineering Sciences.

[15] W. Hare, C. Sagastizábal. A redistributed proximal bundle method for nonconvex optimization. SIAM J. Optim. vol. 20, no. 5, 2010, pp. 2442 - 2473.

[16] C.A. Jacobson, C.N. Nett. Linear state-space systems in infinite-dimensional space: the role and characterization of joint stabilizability/detectability. IEEE TAC 33 (1988), 541 - 549.

[17] T. Kato. Perturbation theory for linear operators; 2nd ed. Ser. Grundlehren Math. Wiss. Berlin: Springer, 1976.

[18] C. Kiwiel. An aggregate subgradient method for nonsmooth convex minimization. Math. Programming, vol. 27, 1983, p. 320 - 341.

[19] K. C. Kiwiel. A linearization algorithm for computing control systems subject to singular value inequalities. IEEE Trans. Automat. Control, AC-31 (1986), 595-602.

[20] D. Mayne, E. Polak. Algorithms for the design of control systems subject to singular value inequalities. Math. Programming Stud., 18 (1982), 112-134.

[21] D. Mayne, E. Polak, A. Sangiovanni. Computer aided design via optimization. Automatica, 18 (1982), pp. 147-154.

[22] D. Noll, O. Prot, A. Rondepierre. A proximity control algorithm to minimize nonsmooth and nonconvex functions. Pac. J. Optim. 4 (2008), no. 3, 571-604.

[23] E. Polak, Y. Wardi. A nondifferential optimization algorithm for the design of control systems subject to singular value inequalities over the frequency range. Automatica, 18 (1982), pp. 267-283.

[24] E. Polak, S. Salcudean. On the design of linear multivariable feedback systems via constrained nondifferentiable optimization in H_∞ space. IEEE Trans. Autom. Control, AC-34 (1989), 268-276.

[25] L. Ravanbod, D. Noll, P. Apkarian. Computing the structured distance to instability. Proceedings of the SIAM Conference on Control and Applications, Paris, 2015, pp. 423 - 430.

[26] A. Ruszczyński. Nonlinear Optimization. Princeton University Press, 2007.

[27] C. Sagastizábal. Composite proximal bundle method. Math. Progr. 140(1), 2013, 189-233.

[28] D. Salamon. Infinite dimensional linear systems with unbounded control and observation: a functional analytic approach. Transactions of the American Mathematical Society, vol. 300, no. 2, pp. 383-431, 1987.

[29] H. Sano. H_∞-control of a parallel-flow heat exchange process. IFAC-PapersOnLIne 48-25(2015), 050-055.

[30] H. Schramm, J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual ideas, convergence analysis, numerical results. SIAM J. Opt. 2, 1992, 121-152.

[31] J.E. Spingarn. Submonotone subdifferentials of Lipschitz functions. Trans. Amer. Math. Soc. 264, 1981, 77-89.

[32] J.G. Van de Vusse. Plug-flow type reactor versus tank reactor. Chemical Engineering Science 19, 1964, 994-998.

[33] P.Yan, M.Debiasi, X.Yuan, J.Little, H. Özbay, M.Samimy. Experimental study of linear closed-loop control of subsonic cavity flow, AIAA Journal, vol. 44, no. 5, pp. 929 – 938, 2006.