INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust. Nonlinear Control 2011; **21**:197–217
Published online 12 July 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/rnc.1589

Time domain constrained H_∞-synthesis

Pierre Apkarian[1], Laleh Ravanbod-Hosseini[2,*,†] and Dominikus Noll[2]

[1]*ONERA, Department of Systems Control and Flight Dynamics, Toulouse, France*
[2]*Université Paul Sabatier, Institut de Mathématiques, Toulouse, France*

SUMMARY

Structured H_∞-synthesis subject to time domain performance specifications is introduced. Our method allows to control trajectories of the linearized system and the underlying nonlinear dynamics simultaneously. A non-smooth bundle optimization method for this class of programs is proposed and discussed. Our approach is tested against two benchmark studies: control of a rotational actuator to attenuate vibration noise, and control of a continuous crystallizer. Our algorithm gives a local convergence certificate and is suited for systems with large state dimension. Copyright © 2010 John Wiley & Sons, Ltd.

Received 30 April 2009; Revised 20 November 2009; Accepted 23 November 2009

KEY WORDS: H_∞ control; structured control laws; time domain performance specification; frequency domain performance specification; non-smooth optimization

1. INTRODUCTION

Structured H_∞-synthesis was introduced in [1] and represents a significant breakthrough in feedback control design of LTI systems. It allows to combine practically useful tuning techniques with the rationale of H_∞-control, bridging the gap between practice and theory. Structured H_∞-synthesis may for instance be used to design H_∞-optimal PID controllers or H_∞-controllers within other practically useful controller structures or architectures. This includes observer-based controllers, decentralized H_∞-controllers, multi-channel and multi-scenario control, control with feed forward structure, set-point and washout filters, and so on.

The purpose of the present paper is to apply the rationale of H_∞-control within the frame of iterative feedback tuning (IFT) techniques. IFT was popularized by Gevers [2] and co-workers since the 1990s and has developed into one of the principal research areas in control, with a large body of literature available. Boyd and Barratt [3] could with some right claim as of having put forward the whole idea. In IFT closed-loop responses to specific input signals (steps, ramps, sinusoids, etc.) are shaped using least squares methods in order to directly satisfy time-domain specifications like limits on under- or overshoot, rise- and settling times, or control constraints like saturation. Such criteria can also be influenced by the traditional frequency domain set-up, but IFT allows to address these time domain criteria more directly. Another advantage of IFT is that nonlinear and even data-driven systems can be handled with almost the same methods.

Yet, in some sense IFT throws away the baby with the bath water. Shaping responses *exclusively* in the time domain deprives one of the sophisticated frequency domain strategies, such as

*Correspondence to: Laleh Ravanbod-Hosseini, Université Paul Sabatier, Institut de Mathématiques, Toulouse, France.
†E-mail: lalehravanbod@yahoo.fr

H_∞-control. So, the idea matures to combine these two seemingly separate worlds. This is the intention of the present work.

2. CONSTRAINED H_∞-PROGRAM

What we have in mind could best be referred to as *constrained structured H_∞-synthesis*. In order to explain the set-up, let us consider the following typical situation. Suppose we are given a plant

$$P:\begin{cases}\dot{x}=Ax+B_1w+B_2u\\ z=C_1x+D_{11}w+D_{12}u\\ y=C_2x+D_{21}w+D_{22}u\end{cases} \tag{1}$$

in standard form, where $x\in\mathbb{R}^n$ is the state, $y\in\mathbb{R}^{p_2}$ the measured output, $z\in\mathbb{R}^{p_1}$ the controlled output, $u\in\mathbb{R}^{m_2}$ the control, and $w\in\mathbb{R}^{m_1}$ the exogenous input. We wish to compute an output feedback controller with a given structure

$$K(\mathbf{x}):\begin{cases}\dot{x}_K=A_K(\mathbf{x})x_K+B_K(\mathbf{x})y\\ u=C_K(\mathbf{x})x_K+D_K(\mathbf{x})y,\end{cases} \tag{2}$$

which stabilizes the plant (1) internally and optimizes the transfer $w\to z$ in the H_∞-norm among all stabilizing controllers $K(\mathbf{x})$ with the same structure. Here the term *structure* indicates that the state-space representation (A_K, B_K, C_K, D_K) of K depends smoothly on a design parameter $\mathbf{x}\in\mathbb{R}^r$.

Suppose a locally optimal structured H_∞-controller $K(\mathbf{x}^*)$ has been computed. That is, $K(\mathbf{x}^*)$ stabilizes the plant (1) internally, and the H_∞-norm of the closed-loop channel $w\to z$ is locally minimal among all internally stabilizing controllers of the same structure, i.e.

$$\|T_{w\to z}(K(\mathbf{x}^*))\|_\infty\leqslant\|T_{w\to z}(K(\mathbf{x}))\|_\infty \quad \text{for all } \mathbf{x} \text{ near } \mathbf{x}^*. \tag{3}$$

Here $T_{w\to z}(K,s)$ denotes the closed-loop transfer function $w\to z$ if controller K is used, and

$$\|T_{w\to z}(K)\|_\infty=\max_{\omega\in[0,\infty]}\overline{\sigma}(T_{w\to z}(K,j\omega))$$

is the H_∞-norm of the closed loop transfer matrix $T_{w\to z}(K,\cdot)$. In practice this controller $K(\mathbf{x}^*)$ now undergoes *a posteriori* testing. Typically this consists in displaying time-domain responses

$$z(t),\quad z=T_{w\to z}(K(\mathbf{x}^*))w_0$$

to certain test inputs $w_0(t)$ such as steps, ramps, sinusoids. If some of these responses show unsatisfactory behaviour, such as unacceptable overshoot, bad tracking, long settling times, or actuator saturation, the designer is forced to modify the closed-loop transfer channel $w\to z$ in order to achieve better results. This process usually consists in modifying frequency filters and is based on trial-and-error. Advanced intuitive skills are needed in particular for MIMO systems, where graphical tools fail and appropriate substitutes do not exist. Notwithstanding, posterior testing could also involve closed-loop transfer channels $T_{w'\to z'}$ other than the one used to compute $K(\mathbf{x}^*)$.

Even if the above process allows to identify a new closed-loop channel $T^\sharp_{w\to z}$, the corresponding new locally optimal H_∞-controller $K(\mathbf{x}^\sharp)$ will satisfy the time-domain specifications only at the cost of degrading the good performances of $K(\mathbf{x}^*)$ with respect to the original channel $T_{w\to z}$. To avoid this, we propose the following constrained optimization program

$$\begin{aligned}\text{minimize}\quad & \|T_{w\to z}(K)\|_\infty\\ \text{subject to}\quad & z_\ell(t)\leqslant z(K,t)\leqslant z_u(t)\quad \text{for all } \underline{t}\leqslant t\leqslant\overline{t}\\ & z(K,s)=T_{w\to z}(K,s)w_0(s)\\ & K=K(\mathbf{x}),\ \mathbf{x}\in\mathbb{R}^r\end{aligned} \tag{4}$$

DOI: 10.1002/rnc

where w_0 represents an input test signal as above, and where $z_\ell(t)$ and $z_u(t)$ are pre-specified lower and upper bounds on certain critical time intervals $[\underline{t}, \overline{t}\,]$.

In other words, with (4) we propose to include the time domain constraints *directly* in the design process, instead of using them *a posteriori*. The advantage is that the original criterion $\|T_{w\to z}(K)\|_\infty$ does not have to be abandoned. Instead, the two (or more) concurring objectives are combined and a trade-off is achieved.

Remark
A typical application is PID control for MIMO systems, which today uses heuristic tuning techniques. The reason is that traditional ARE-based synthesis [4] applies to unstructured (full order) controllers only. In state space PIDs can be parametrized as

$$K_{\text{pid}}(\mathbf{x}) = \left[\begin{array}{cc|c} 0 & 0 & R_i \\ 0 & -\tau I_{m_2} & R_d \\ \hline I_{m_2} & I_{m_2} & D \end{array}\right] \tag{5}$$

so the decision vector is $\mathbf{x} = [\tau, \text{vec}(R_i), \text{vec}(R_d), \text{vec}(D)]$. One would typically have a constraint $\tau \geqslant \varepsilon$ in parameter space in order to assure that the controller is realizable and in particular, implementable. (Strictly speaking, (5) is then an approximate PID.) Notice that $\dim(\mathbf{x}) = 3m_2 p_2 + 1$ while the controller is of order $n_K = 2m_2$.

Solving the H_∞-control problem with respect to the structure $K_{\text{pid}}(\mathbf{x})$ provides what we call H_∞-optimal PID controllers. Computing H_∞ PID controllers is possible via the method of [1].

3. TIME DOMAIN RESPONSES OF THE NONLINEAR SYSTEM

It is tempting to go one step further. Consider the case where P is obtained by linearizing a nonlinear time invariant system

$$\widetilde{P}: \begin{cases} \dot{x} = f(x, w, u) \\ z = g(x, w, u) \\ y = h(x, w, u) \end{cases} \tag{6}$$

about its steady state at the origin. That is, $f(0,0,0) = 0$, $g(0,0,0) = 0$ and $h(0,0,0) = 0$, $A = f'_x(0,0,0)$, $B_1 = f'_w(0,0,0)$, etc. The traditional linearizing approach neglects nonlinear effects and works with P. One computes a locally optimal structured H_∞ controller $K(\mathbf{x}^*)$ for the transfer $w \to z$ of P and hopes that $K(\mathbf{x}^*)$ will also be appropriate for $\widetilde{P}$. This is assured if the nonlinearity is mild.

Suppose, however, the system shows significant nonlinearity in some components. Then it is reasonable to work directly with the nonlinear system, by shaping the corresponding trajectories of $\widetilde{P}$:

$$z_\ell(t) \leqslant \widetilde{z}(K, t) \leqslant z_u(t) \quad \text{for all } \underline{t} \leqslant t \leqslant \overline{t} \tag{7}$$

where $\widetilde{z}(K, t) = \widetilde{T}_{w\to z}(K, t, w_0)$ now denotes the response of the nonlinear plant $\widetilde{P}$ to the input $w_0(t)$. The novelty, which we propose here, is to combine features of P and $\widetilde{P}$ in a single set-up. This leads to the optimization program

$$\begin{array}{ll} \text{minimize} & \|T_{w\to z}(K)\|_\infty \\ \text{subject to} & z_\ell(t) \leqslant \widetilde{z}(K, t) \leqslant z_u(t) \quad \text{for all } \underline{t} \leqslant t \leqslant \overline{t} \\ & \widetilde{z}(K, t) = \widetilde{T}_{w\to z}(K, t, w_0) \\ & K = K(\mathbf{x}), \mathbf{x} \in \mathbb{R}^r. \end{array} \tag{8}$$

DOI: 10.1002/rnc

Here one uses the linearized system to assess the H_∞-norm of the transfer function $w \to z$. At the same time program (8) allows to control some of the nonlinear outputs $\widetilde{z}_i$ directly on critical time intervals $[\underline{t}, \overline{t}]$, where they differ significantly from their linearizations. As in (4) we allow the controller $K(\mathbf{x})$ to have a pre-specified structure, so that the optimization variable in (8) is the structural design parameter $\mathbf{x}$, and we include parameter constraints $g(\mathbf{x}) \leqslant 0$ in (8) if required.

Remark
Observe that we continue to seek linear controllers for nonlinear systems, so our method is not to be confused with a genuine nonlinear H_∞ approach like for instance [5]. It is less ambitious (but perhaps more realistic) to assume that only some of the states of $\widetilde{P}$ show genuine nonlinearity, and (8) allows to address these individually using constraints like (7). If successful, the advantage of (8) is that a linear controller can be used. On the other hand, the set-up can be combined with genuine nonlinear strategies. If a change of coordinates reducing nonlinearity is applied before linearizing the system, our methodology is still applicable, although the final controller will then be nonlinear.

Remark
During the rest of the paper, we shall continue to use the symbols $T_{w\to z}(K,s)$ and $z(K,s)$ for closed-loop transfer channels and responses of the linearized plant P, while $\widetilde{T}_{w\to z}$ and $\widetilde{z}(K,t)$ will denote channel and response of the nonlinear plant $\widetilde{P}$. The steady state is at 0.

4. HARD CONSTRAINTS IN IFT

The philosophy of IFT is to shape closed-loop responses $z(K,t)$ or $\widetilde{z}(K,t)$ by optimizing least squares objectives like

$$\min_K \int_{\underline{t}}^{\overline{t}} (z(K,t) - z_{\mathrm{ref}}(t))^2 \, \mathrm{d}t, \tag{9}$$

where $z_{\mathrm{ref}}(t)$ denotes a desired or reference response of the closed-loop system. Recently, we have developed an alternative approach [6–9], which allows to replace soft constraints of the form (9) by hard constraints as in (7). For this we apply non-smooth optimization methods to the constraint violation function

$$\min_{\mathbf{x}\in\mathbb{R}^r} \max_{\underline{t}\leqslant t\leqslant \overline{t}} \{0, z_\ell(t) - z(K(\mathbf{x}),t), z(K(\mathbf{x}),t) - z_u(t)\}. \tag{10}$$

Program (10) ends with an optimal value 0 if $K(\mathbf{x})$ satisfying the constraints (7) is found, while a value >0 indicates a local minimum of constraint violation, meaning failure to satisfy the constraints with the given controller structure from the chosen starting point $\mathbf{x}_0$. In [6–9] this type of non-smooth optimization program is addressed by a bundle technique. Currently, we will extend this approach to solve programs (4) and (8).

In our experiments, trajectories of the linearized system $z(K,t)$ are computed with the Matlab functions `lsim` [10]. For trajectories $\widetilde{z}(K,t)$ of the nonlinear system we have used forward differentiation based on standard Runge-Kutta solvers, which corresponds to the direct mode in automatic differentiation. As soon as the state dimension of the plant gets large, say $\geqslant 300$ states, it may be preferable to use the adjoint equation, which corresponds to the reverse mode of automatic differentiation. For details see [11].

5. SUBGRADIENTS OF THE H_∞-NORM

The H_∞-norm has been discussed at length in [1], so we can be very cursory here. Typical subgradients of the H_∞-norm as a functional on the space of stable transfer matrices are already given in [3]. Applying suitable chain rules gives the full characterization of the subdifferential of

DOI: 10.1002/rnc

$f(\mathbf{x}) = \|T_{w\to z}(K(\mathbf{x}))\|_\infty$. Clarke regularity [12] of the composite function $\mathbf{x} \mapsto \|\cdot\|_\infty \circ T_{w\to z}(\cdot) \circ K(\mathbf{x})$ allows to apply a non-smooth chain rule, which leads to the characterization of the $\phi \in \partial f(\mathbf{x})$ as

$$\phi = \frac{\mathrm{d}K}{\mathrm{d}\mathbf{x}}^* \frac{\mathrm{d}T^*_{w\to z}}{\mathrm{d}K}\Phi,$$

where $\Phi \in \partial\|\cdot\|_\infty(T_{w\to z}(K(\mathbf{x})))$ is a subgradient of the H_∞-norm in the convex sense. The form of these subgradients Φ and of the adjoint of $\mathrm{d}T_{w\to z}/\mathrm{d}K$ have been given in [1], while $\mathrm{d}K/\mathrm{d}\mathbf{x}$ and its adjoint depend on the controller structure and have to be provided by the user in each case.

Computation of the H_∞ norm as well as the peaks of the frequency curve $\omega \mapsto T_{w\to z}(K, j\omega)$ for fixed K is based on the bisection method of [13], which means that function values may occasionally be corrupted by computational noise. Subgradients of the H_∞ norm use linear algebra procedures, but require knowledge of the peak positions. Fortunately, these numerical errors are rarely observed in practice.

6. ALGORITHM

The algorithmic approach to the constraint program (4) or (8) is based on a three-stage procedure.

Algorithm for program (8)

1: **Initial phase**. Compute a locally optimal H_∞ controller $K(\mathbf{x}_1)$ with pre-assigned structure. Alternatively, compute a structured $K(\mathbf{x}'_1)$ such that the time domain constraints (7) are satisfied.
2: **Smooth optimization**. Use $\mathbf{x}_1$ or $\mathbf{x}'_1$ as initial guess to run a smooth SQP-solver in which program (8) is solved as if all functions were smooth. For every non-smooth function pick a single subgradient as a substrate for the gradient. The result of phase 2 is $K(\mathbf{x}_2)$.
3: **Non-smooth phase.** Use $\mathbf{x}_2$ as initial guess to run the non-smooth progress function method of Section 8. The result is $K(\mathbf{x}_3)$, which has a non-smooth local optimality certificate.

We expect the smooth optimization phase 2 to be fast and bring us close to a local optimum of (4) or (8) in the majority of cases. Notice, however, that even when $\mathbf{x}_2$ turns out locally optimal, the usual smooth optimality tests in SQP will not apply because (8) is non-smooth. Therefore, phase 3 is always needed, either to continue optimization if the smooth method encounters difficulties, or just to furnish a non-smooth local optimality test.

When phase 2 ends with $\mathbf{x}_2$, we have the following possibilities. There are the usual reasons for failure in smooth SQP methods, like local minima of constraint violation, or insignificant progress. To these we have to add the case where the SQP solver in phase 2 signals a local minimum (a KKT point), but erroneously so. We may indeed encounter *dead points*, where the smooth method stops, but where a non-smooth method will achieve further progress till a genuine local minimum (or KKT point) is reached. While this seems a long list of possible failure, the good news is that even if phase 2 fails to find a KKT point, it will often bring us close to a local minimum and give a good starting point for phase 3.

The individual steps of this scheme will now be discussed. Phase 1 uses structured H_∞-synthesis [1] to compute $K(\mathbf{x}_1)$ or the methods mentioned in Section 4 to compute $K(\mathbf{x}'_1)$. Phases 2 and 3 will be discussed subsequently.

Remark

H_∞ optimization as presented in [1] requires an initial closed-loop stabilizing controller, which we compute by the method of [14]. We mention that the idea to optimize stability of closed-loop systems directly via the spectral abscissa was pioneered by Burke *et al.*, see e.g. [15–17].

DOI: 10.1002/rnc

7. SMOOTH OPTIMIZATION

Phase 2 of Algorithm 1 is heuristic, because no convergence or optimality certificate whatsoever can be proved by means of a smooth technique if program data are non-smooth. And yet, phase 2 is important in practice because it often finds a solution. Even in the case of failure, phase 2 may still provide good initial guesses for phase 3. So how should phase 2 be organized?

In order to apply a sequential quadratic programming (SQP) algorithm to program (4) or (8), we are somewhat restricted in our choice due to the following facts:

1. Gradients are available almost everywhere, but *not* everywhere. As a substitute one may pick Clarke subgradients.
2. We have observed that using finite difference approximations to provide substitutes for the gradients is a bad idea. On top of being costly, they are also often of bad quality. There is no good explanation for this badness, other than recalling that gradients do not really exist.
3. Second order information is definitely unavailable. Therefore, methods using BFGS type updates appear to be a good option. Software based on these criteria is for instance Powell's [18], Schittkowski's `nlpqlp` [19], or the Matlab function `fmincon` [20].
4. A fourth aspect, maybe the most troublesome, is that (8) has a hidden constraint: closed-loop stability of $K(\mathbf{x})$. Indeed, the H_∞-norm $f(\mathbf{x})=\|T_{w\to z}(K(\mathbf{x}))\|_\infty$ is undefined for candidate controllers $K(\mathbf{x})$, which are not stabilizing. While it is in principle possible to maintain closed-loop stability of $K(\mathbf{x})$ during the iterations, for instance by a backtracking line-search if a destabilizing iterate is encountered, the difficulty is that solvers are usually not prepared for this situation, so that the tangent program will every now and then produce non-stabilizing controllers, which lead to infinite function values during line-search, causing breakdown of the SQP solver. This difficulty is best avoided if the solver uses reverse communication, where one may intervene *during* the optimization process if infinite function values $f(\mathbf{x})=\infty$ occur. (In our opinion this is one of the reasons that gives reverse communication an advantage over those strategies where the program data are passed as a function arguments into the optimizer).

The problem of destabilizing candidate controllers may to some extent be defanged if the representation of the closed loop plant $[A(K), B(K), C(K), D(K)]$ is minimal. In that case the objective $f(K)$ behaves like a barrier function as controllers approach the boundary of the region of internal stability, and this will help to keep iterates K within the region of finiteness of f. In our current testing the phenomenon was rarely observed, but in [6, 21] we had previously identified this as a potential difficulty for SQP methods in control. SQP as a general tool gains much of its potential by allowing unfeasible iterates during the optimization. In our current context the presence of time domain constraints seems rather beneficial to maintain closed-loop stability of trial points.

8. NON-SMOOTH PROGRESS FUNCTION

At the core of our scheme is phase 3, which uses non-smooth optimization. The theoretical foundations are given in [22–24]. Here we recall the main ingredients and represent the algorithm by a flowchart.

In abstract form, program (8) may be written as $\min\{f(\mathbf{x}): g(\mathbf{x})\leqslant 0\}$, where f and g are potentially non-smooth functions. We assume that Clarke subgradients [12] of f and g are available. For the criteria used in (4) and (8) these subgradients have been made explicit in the previous sections.

The algorithm of Figure 1 generates the sequence $\mathbf{x}^j$, $j=1,2,\ldots$ of so-called serious steps, which represent the outer loop of the algorithm. In the flowchart this part is the outer box. At the current iterate $\mathbf{x}^j$ of the outer loop we consider the progress function

$$F(\mathbf{x},\mathbf{x}^j)=\max\{f(\mathbf{x})-f(\mathbf{x}^j)-\mu[g(\mathbf{x}^j)]_+, g(\mathbf{x})-[g(\mathbf{x}^j)]_+\}, \tag{11}$$

DOI: 10.1002/rnc

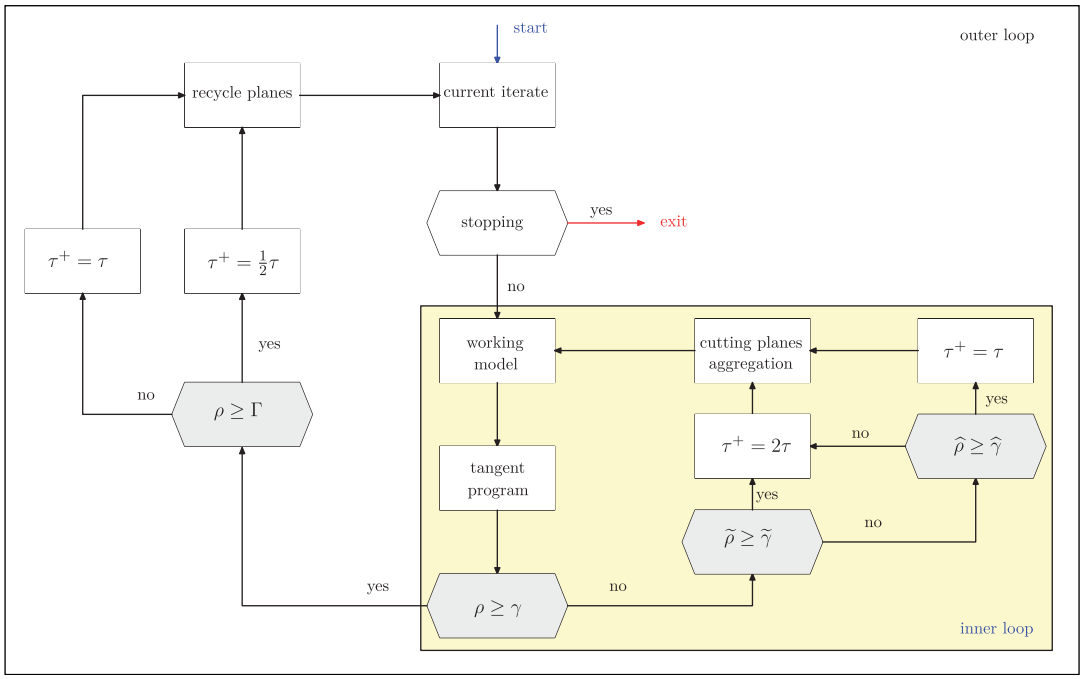

Figure 1. Flowchart of non-smooth progress function algorithm.

where $[u]_+ = \max\{u, 0\}$ denotes the positive part, and $\mu > 0$ is fixed once and for all. Our algorithm seeks a new serious iterate $\mathbf{x}^{j+1}$ such that $F(\mathbf{x}^{j+1}, \mathbf{x}^j)$ is significantly below the value of $F(\mathbf{x}^j, \mathbf{x}^j) = 0$. In the end, every accumulation point $\mathbf{x}^*$ of the sequence $\mathbf{x}^j$ will satisfy $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$, where ∂_1 denotes the Clarke subdifferential with respect to the first coordinate. It follows that every such $\mathbf{x}^*$ is a F. John critical point of (8), which is as close to being a local minimum as one can expect; see [23]. In practice, $\mathbf{x}^*$ is either a local minimum of (8), or a local minimum of constraint violation.

How do we compute the new serious step $\mathbf{x}^{j+1}$, and what is meant by sufficient decrease $F(\mathbf{x}^{j+1}, \mathbf{x}^j) < F(\mathbf{x}^j, \mathbf{x}^j) = 0$? We first build local models $\widetilde{f}$ of f and $\widetilde{g}$ of g in a neighbourhood of the current $\mathbf{x}^j$. For the H_∞ norm we know that

$$f(\mathbf{x}) = \max_{\omega \in [0,\infty]} \overline{\sigma}(T_{w\to z}(\mathbf{x}, j\omega)),$$

where $T_{w\to z}(\mathbf{x}, s)$ denotes the closed loop transfer function $w \to z$ with controller $K(\mathbf{x})$. Now we put

$$\widetilde{f}(\mathbf{x}, \mathbf{x}^j) = \max_{\omega \in [0,\infty]} \overline{\sigma}(T_{w\to z}(\mathbf{x}^j, j\omega) + D_{\mathbf{x}} T_{w\to z}(\mathbf{x}^j, j\omega)(\mathbf{x} - \mathbf{x}^j)),$$

where $D_{\mathbf{x}}$ denotes derivatives with respect to the controller parameter $\mathbf{x}$. We call $\widetilde{f}(\cdot, \mathbf{x}^j)$ the ideal model of f at $\mathbf{x}^j$. To get the idea, the ideal model of a function $f(\mathbf{x})$ of class C^1 is simply $\widetilde{f}(\mathbf{x}, \mathbf{x}^j) = f(\mathbf{x}^j) + \nabla f(\mathbf{x}^j)^T(\mathbf{x} - \mathbf{x}^j)$. For a composite function like $f = \overline{\sigma} \circ F$ with smooth $F : \mathbb{R}^r \to \mathbb{R}^{p\times m}$ we use partial linearization $\widetilde{f}(\mathbf{x}, \mathbf{x}^j) = \overline{\sigma}(F(\mathbf{x}^j) + D_{\mathbf{x}} F(\mathbf{x}^j)(\mathbf{x} - \mathbf{x}^j))$.

For time domain constraints $z_\ell(t) \leqslant \widetilde{z}(\mathbf{x}, t) \leqslant z_u(t)$ we use the non-smooth expression (10). Clarke subgradients of (10) can be found in [6]. Notice a difference with Section 7, where hard constraints (7) are discretized at certain $\underline{t} = t_1 < t_2 < \cdots < t_d = \overline{t}$. In the non-smooth case we address the semi-infinite constraint (7) *directly*, so no discretization is used. This is an advantage if response curves exhibit sharp peaks, which could be missed by the discrete set t_i.

Having defined $\widetilde{f}$ and $\widetilde{g}$, we obtain the following approximation $\widetilde{F}$ of F,

$$\widetilde{F}(\mathbf{x}, \mathbf{x}^j) = \max\{\widetilde{f}(\mathbf{x}, \mathbf{x}^j) - f(\mathbf{x}^j) - \mu[g(\mathbf{x}^j)]_+, \widetilde{g}(\mathbf{x}, \mathbf{x}^j) - [g(\mathbf{x}^j)]_+\},$$

DOI: 10.1002/rnc

called the *ideal model* of F. We consider $\widetilde{F}(\cdot,\mathbf{x}^j)$ a substitute for the linearization (first-order expansion) of $F(\cdot,\mathbf{x}^j)$ at $\mathbf{x}^j$. Such a substitute is required because f and g are non-smooth.

If f and g are smooth, the ideal model $\widetilde{F}$ is the maximum of two affine functions and can be used directly to generate descent steps. In the non-smooth case $\widetilde{F}$ is convex in the first variable, but may still be too rich to be useful in the tangent program. We need a coarse approximation of $\widetilde{F}(\cdot,\mathbf{x}^j)$, which we call the *working model* of F at $\mathbf{x}^j$. It is denoted $\widetilde{F}_k(\cdot,\mathbf{x}^j)$ and has a counter k to it because it is improved iteratively in the inner loop of our algorithm, which uses that counter k. The inner loop is shown in the box in the lower right part of the flowchart (Figure 1).

Like $\widetilde{F}(\cdot,\mathbf{x}^j)$ the working model $\widetilde{F}_k(\cdot,\mathbf{x}^j)$ is convex in the first argument. It is constructed by assembling support planes of the ideal model. In particular, this means $\widetilde{F}_k\leqslant\widetilde{F}$ for all k. We distinguish four types of such planes:

1. Exactness plane,
2. Cutting plane,
3. Aggregate plane,
4. Heuristic planes.

The exactness plane, item 1, is of the form $m_0(\mathbf{x})=F(\mathbf{x}^j,\mathbf{x}^j)+\phi^{\mathrm{T}}(\mathbf{x}-\mathbf{x}^j)$, where $\phi\in\partial_1 F(\mathbf{x}^j,\mathbf{x}^j)$. This means m_0 is an affine support function of $\widetilde{F}(\cdot,\mathbf{x}^j)$ at $\mathbf{x}^j$. Including m_0 in the model $\widetilde{F}_k(\cdot,\mathbf{x}^j)$ implies $\widetilde{F}_k(\mathbf{x}^j,\mathbf{x}^j)=F(\mathbf{x}^j,\mathbf{x}^j)$ for all k, hence the name exactness plane.

Cutting planes and aggregate planes are constructed iteratively. Suppose we have already built model $\widetilde{F}_k(\cdot,\mathbf{x}^j)$ at the kth instance of the inner loop belonging to the jth instance of the outer loop. Then, we solve the tangent program with proximity control

$$\min_{\mathbf{y}\in\mathbb{R}^n}\widetilde{F}_k(\mathbf{y},\mathbf{x}^j)+\frac{\tau_k}{2}\|\mathbf{y}-\mathbf{x}^j\|^2, \tag{12}$$

where $\tau_k>0$ is the so-called proximity control parameter. Suppose the solution of (12) is $\mathbf{y}^k$, called the trial step. Fixing $0<\gamma<\Gamma<1$ once and for all, we compute the quotient

$$\rho_k=\frac{-F(\mathbf{y}^k,\mathbf{x}^j)}{-\widetilde{F}_k(\mathbf{y}^k,\mathbf{x}^j)}, \tag{13}$$

which tests agreement between F and the working model $\widetilde{F}_k$ at $\mathbf{y}^k$. If $\rho_k\geqslant\gamma$, then we accept the trial step $\mathbf{y}^k$ as the new serious iterate, $\mathbf{x}^{j+1}=\mathbf{y}^k$, because F decreases sufficiently. This is called a serious step. If $\rho_k<\gamma$, then $\mathbf{y}^k$ is rejected, because decrease of F is insufficient, and we call this a null step. Here our working model needs to be improved at the next inner loop iteration $k+1$.

In the case of a null step, we pick a subgradient $\phi_k\in\partial_1\widetilde{F}(\mathbf{y}^k,\mathbf{x}^j)$ and define $m_k(\mathbf{x})=\widetilde{F}(\mathbf{y}^k,\mathbf{x}^j)+\phi_k^{\mathrm{T}}(\mathbf{x}-\mathbf{y}^k)$ as the cutting plane. We include $m_k(\cdot)$ as affine support function into the upcoming model $\widetilde{F}_{k+1}(\cdot,\mathbf{x}^j)$.

Finally, by the definition of $\mathbf{y}^k$, we have $0\in\partial_1\widetilde{F}_k(\mathbf{y}^k,\mathbf{x}^j)+\tau_k(\mathbf{y}^k-\mathbf{x}^j)$. Therefore, $\phi_k^*=\tau_k(\mathbf{x}^j-\mathbf{y}^k)\in\partial_1\widetilde{F}_k(\mathbf{y}^k,\mathbf{x}^j)$. But $\widetilde{F}_k(\cdot,\mathbf{x}^j)$ is by induction already built as the convex envelope of older planes of the four types. That means, the element ϕ_k^* can be written as $\phi_k^*=\sum_{v=1}^r\tau_v\phi_v$, $\tau_v\geqslant 0$, $\sum_{v=1}^r\tau_v=1$, where $m_v(\mathbf{x})=\alpha_v+\phi_v^{\mathrm{T}}(\mathbf{x}-\mathbf{x}^j)$ are planes used to build $\widetilde{F}_k(\cdot,x^j)$. Then, put $\alpha^*=\sum_{v=1}^r\tau_v\alpha_v$ and let $m_k^*(\mathbf{x})=\alpha^*+\phi_k^{*\mathrm{T}}(\mathbf{x}-\mathbf{x}^j)$ be the so-called aggregate plane.

We include m_0,m_k,m_k^* among the affine minorants of $\widetilde{F}_{k+1}(\cdot,\mathbf{x}^j)$. In addition, we allow to include all sorts of heuristic planes in the model to enrich it. This may be recycled planes from previous steps $k,k-1,k-2,\ldots$, or anticipated cutting planes, as used in [24].

Before re-running the tangent program (12) with the new working model, we have to modify the proximity control parameter τ_k. It is the management of $\tau_k\to\tau_{k+1}$ that makes the inner loop (Figure 1 lower right part) somewhat technical. The general idea is to avoid increasing τ_k prematurely, because large τ_k leads to small trial steps. So we set forth maximum resistance before we ease up and increase τ_k. In order to decide, a secondary control parameter $\widetilde{\rho}_k$ is introduced

$$\widetilde{\rho}_k=\frac{-\widetilde{F}(\mathbf{y}^k,\mathbf{x}^j)}{-\widetilde{F}_k(\mathbf{y}^k,\mathbf{x}^j)}.$$

DOI: 10.1002/rnc

It tests agreement between working model and ideal model at $\mathbf{y}^k$. Fixing $0<\gamma<\widetilde{\gamma}<1$ once and for all, one checks whether $\widetilde{\rho}_k \geqslant \widetilde{\gamma}$. If this is the case, τ_k is increased, for instance, $\tau_{k+1}=2\tau_k$. This is referred to a tightening proximity control. Otherwise, the third control parameter $\widehat{\rho}_k=\rho_k/\widetilde{\rho}_k$ is compared to $\widehat{\gamma}\in(0,1)$. If $\widehat{\rho}_k \geqslant \widehat{\gamma}$, then $\tau_{k+1}=\tau_k$, otherwise τ_k is increased. It can be shown that updating $\widetilde{F}_k \to \widetilde{F}_{k+1}$ and $\tau_k \to \tau_{k+1}$ in this way guarantees finding a serious step after a finite number of trials k. In other words, the inner loop is finite if $0 \notin \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$.

The last ingredient of the algorithm is when $\mathbf{x}^{j+1}=\mathbf{y}^k$ is accepted at inner loop counter k_j at the jth outer loop, i.e. when $\rho_k \geqslant \gamma$. If $\rho_{k_j}>\Gamma$, then the proximity control parameter is re-started with $\tau_1=\frac{1}{2}\tau_{k_j}$ in the $j+1$st outer loop. That is, proximity control is slackened. On the other hand, if $\gamma \leqslant \rho_{k_j} \leqslant \Gamma$ then we initialize as $\tau_1=\tau_{k_j}$, so proximity control is maintained.

Remark

A detailed convergence theory of our method is not within the scope of the present work, but can be found in [23]. The control problems treated here have so far only be addressed by heuristic techniques, and our method gives a first rigourous approach. This may to some extent account for the rather complex structure of the algorithm.

9. ROTATIONAL ACTUATOR TO STABILIZE TRANSLATIONAL MOTION

The rotational actuator shown in Figure 2 was proposed as a benchmark test example in [25]. Its nonlinear dynamics are

$$\begin{aligned}(M+m)\ddot{Z}+kZ &= -me(\ddot{\theta}\cos\theta-\dot{\theta}^2\sin\theta)+F\\ (I+me^2)\ddot{\theta} &= -me\ddot{Z}\cos\theta+N\end{aligned} \tag{14}$$

The cart moves horizontally with position Z [m], and the pendulum with angular position θ [rad] is used as a control torque N. This type of device is for instance used to suppress oscillatory motions of the cart or vibration noise represented by the external disturbance F. The physical parameters are given in Table I.

Coupling of the rotational and translational motions is represented by the dimension free parameter

$$\varepsilon=\frac{me}{\sqrt{(I+me^2)(M+m)}}.$$

The system has four states $(Z, \dot{Z}, \theta, \dot{\theta})$, the measured output is $y=(Z,\theta)$, the control is $u=N$. The regulated outputs are $z=W_z(Z,\theta,N)$, where $W_z=\mathrm{diag}[16.1, 0.3162, 14.1]$ reflects that we want the off-sets Z and θ small. Including the control N among the outputs is standard to avoid exceedingly

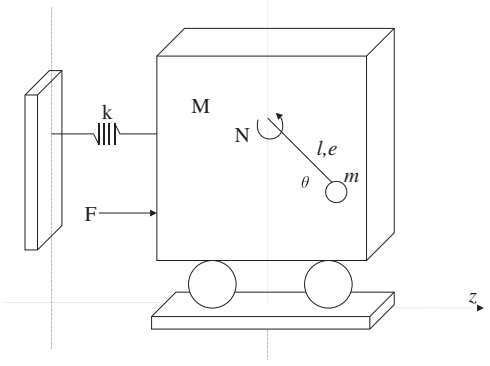

Figure 2. Rotational actuator.

DOI: 10.1002/rnc

Table I. Physical parameters of rotational actuator.

Mass of cart	$M = 1.3608\,\mathrm{kg}$
Mass of pendulum	$m = 0.096\,\mathrm{kg}$
Pendulum eccentricity	$e = 0.0592\,\mathrm{m}$
Moment of inertia	$I = 2.175e-4\,\mathrm{kg\,m^2}$
Spring stiffness	$k = 186.3\,\mathrm{N/m}$
Coupling parameter	$\varepsilon = 0.2001$

Table II. Initial and final controllers for low frequency constraint.

Test #	Controller	Order	$\|T_{w\to z}\|_\infty$	Constraint
—	H_∞	4	5.2327	—
1	$K_{1,\mathrm{init}}$	1	5.2340	0.0390
	$K_{1,\mathrm{final}}$	1	5.2960	$-1.8521e-5$
2	$K_{2,\mathrm{init}}$	1	5.5232	$-3.2603e-4$
	$K_{2,\mathrm{final}}$	1	5.2866	$1.3174e-7$

Test 1 starts with unfeasible but good H_∞-norm, test 2 starts with feasible but bad H_∞-norm. The H_∞ norm of $T_{F\to W_z(Z,\theta,N)}$ is optimized.

Table III. Rotational actuator. Initial and final controllers for time domain constraints on response to low frequency sinusoidal input.

Test #	Controller
1	$K_{1,\mathrm{init}} = [-4.837\ -9.233\ -19.27\ -2.287\ 5.113\ -9.694]$ $K_{1,\mathrm{final}} = [-3.0942\ -17.4656\ -15.5944\ -1.9478\ -5.2216\ -11.3253]$
2	$K_{2,\mathrm{init}} = [-1.8077\ 0.9335\ -2.5380\ -1.0310\ 1.0772\ -1.7093]$ $K_{2,\mathrm{final}} = [-1.5738\ 0.8750\ -2.5509\ -1.0299\ 1.7263\ -1.9793]$

Table IV. Rotational actuator. Initial and final first-order controllers for time domain constraint on high frequency sinusoidal input.

Test #	Controller
3	$K_{3,\mathrm{final}} = [-7.2111\ -9.3198\ -18.1361\ -4.3229\ -2.3483\ -11.8707]$
4	$K_{4,\mathrm{init}} = [-3.3632\ 4.9686\ -4.1849\ -1.9526\ 3.7555\ -2.5335]$ $K_{4,\mathrm{final}} = [-3.3159\ 4.9527\ -4.1239\ -1.7697\ 3.9251\ -2.6908]$

In test 3 the initial guess was $K_{1,\mathrm{init}}$.

large control actions. The external force $w = F$ acting on the cart represents a disturbance, which is the source of the vibratory motion to be attenuated.

In a first test we computed the full order H_∞ controller and a locally optimal H_∞ controller $K_{1,\mathrm{init}}$ of order $n_K = 1$. The H_∞ norm of the closed-loop channel $F \to W_z(Z,\theta,N)$ is shown in column 4 of Table II. Lines 1 and 2 in Table II show that the closed-loop gain of the optimal first-order H_∞ controller is very close to what may be achieved with the full order H_∞ controller.

Given this observation, the goal of the study is to achieve satisfactory H_∞ control subject to time domain constraints with a first-order controller. The design parameter is therefore $\mathbf{x} \in \mathbb{R}^6$ with

$$K(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \\ \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 \end{bmatrix}, \tag{15}$$

while a full order (unstructured) controller would have 30 parameters. In Tables III and IV we display this as $K = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_6]$ for space reasons.

DOI: 10.1002/rnc

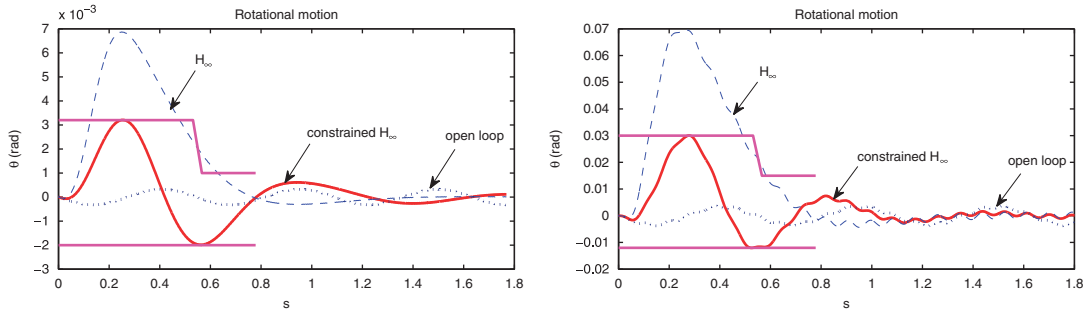

Figure 3. Rotational actuator. Closed-loop response $\widetilde{\theta}(t)$ to high (right) and low (left) frequency sinusoidal input signal $w_0(t)=\sin\omega t$. Curves with the smallest amplitude (left and right) show the open-loop, top curves show the closed-loop with locally optimal first-order H_∞-controller $K_{1,\text{init}}$. In both cases the constraint is violated. The middle curves show responses with $K_{1,\text{final}}$ (left) and $K_{3,\text{final}}$ (right) obtained by (8).

Table V. Rotational actuator. Initial and final first-order controllers for constraints on time domain response to high frequency sinusoidal input.

Test #	Controller	$\|T_{w\to z}\|_\infty$	Constraint
3	$K_{1,\text{init}}$	5.2340	$3.7000e-3$
	$K_{3,\text{final}}$	5.2717	$-4.5593e-5$
4	$K_{4,\text{init}}$	5.3827	$-1.9040e-7$
	$K_{4,\text{final}}$	5.2591	$-3.0050e-8$

Test 3 uses initial $K_{1,\text{init}}$ from test 1. Test 4 starts with feasible K.

Posterior inspection of the optimal H_∞ controller in closed loop (top curve in Figure 3 left and right) shows that the time domain response $\widetilde{\theta}(t)$ of the nonlinear system, excited by a sinusoidal input $w_0(t)=\sin\omega t$, shows unsatisfactory transient behaviour between 0 and 3 s, whereas the steady-state response is acceptable. We therefore added time domain constraints on $\widetilde{\theta}(t)$ and $\widetilde{Z}(t)$ together with a constraint on closed-loop poles. This leads to the optimization program

$$\begin{aligned}
&\text{minimize} && \|T_{F\to W_z(Z,\theta,N)}(K)\|_\infty \\
&\text{subject to} && \theta_{\min}(t)\leqslant\widetilde{\theta}(K,t,w_0)\leqslant\theta_{\max}(t) \quad \text{as in Figure 3} \\
& && Z_{\min}(t)\leqslant\widetilde{Z}(K,t,w_0)\leqslant Z_{\max}(t) \quad \text{as in Figure 5} \\
& && \alpha(A(K))\leqslant -0.15 \\
& && K \text{ as in (15)}
\end{aligned} \tag{16}$$

where $A(K)$ denotes the closed-loop system matrix. Here $\alpha(A)=\max\{Re(\lambda):\lambda\in\text{spec}(A)\}$ is the spectral abscissa of A, and the constraint $\alpha(A(K))\leqslant -0.15$ guarantees the closed-loop stability.

According to our previous terminology, $\widetilde{\theta}$ denotes the angular position of the nonlinear system, in response to the sinus $w_0(t)=\sin\omega t$, while θ is as usual the off-set of the linearized system about the steady state $\widetilde{\theta}=0$. The objective and the constraint $\alpha\leqslant -0.15$ clearly refer to the linearized system. Similar arrangements hold for Z and $\widetilde{Z}$. Each instance of program (8) has six unknown variables and 93 inequality constraints (2×23 for $\widetilde{\theta}$, 2×23 for $\widetilde{Z}$, and one for α).

The time domain constraints of (16) are represented in column 5 of Tables II and V, which displays the maximum violation of all constraints. The optimal H_∞-controller $K_{1,\text{init}}$ does *not* satisfy the time domain constraints, hence the value >0 in column 5. Alternatively, we have used other initial controllers $K_{2,\text{init}}$ respectively $K_{4,\text{init}}$, which do satisfy the time domain constraints strictly, hence the values <0 in columns 5 of Tables II and V. On the other hand, these controllers

DOI: 10.1002/rnc

have large H_∞ norm (column 3). Optimal controllers (solutions of (8)) behave well in *both* respects, and activate some of the constraints. This leads to the values 0 in columns 5 of the two tables.

Using (8) allows to significantly reduce the unacceptable transient peak in $\widetilde{\theta}(t)$ (Figure 3), at the cost of only a very mild increase of the H_∞ norm (Figure 6). The control input is shown in Figure 4. This comes with a significant reduction of the H_2 norm under a performance constraint on the H_∞-norm, a result which one could try to obtain directly using mixed H_2/H_∞ optimization as in [23].

Remark

Figure 6 shows that (8) reduces the L_2-norm and also L_1-norm in frequency domain. Reduction of the frequency L_1 norm indicates reduction of the time domain L_∞-norm, and therefore a reduction of the peak, but there is no tight duality relation between the two norms, as opposed to the time and frequency L_2 norms. Reducing the time domain peak by optimizing the frequency L_1 or L_2 norm is therefore justified heuristically, but less stringent than optimizing the time domain constraint explicitly. Optimizing the peak-to-peak norm would be an alternative, but this norm is harder to compute than the H_2 and H_∞-norms [26], and no attempt to optimize this criterion has to date been reported.

10. CASE STUDY: CONTINUOUS CRYSTALLIZER

Our second study uses the model of a continuous crystallizer presented by Vollmer and Raisch [27] (see Figure 7), where crystal growth is modelled by mole and population balance equations. The mole balance is

$$\begin{aligned} M\frac{\mathrm{d}c}{\mathrm{d}t} &= \frac{q(\rho-Mc)}{V}+\frac{\rho-Mc}{\varepsilon}\frac{\mathrm{d}\varepsilon}{\mathrm{d}t}+\frac{qMc_f}{V\varepsilon}-\frac{q\rho}{V\varepsilon}\left(1+k_v\int_0^\infty (h_p(L)-1)n(L,t)L^3\,\mathrm{d}L\right)\\ \varepsilon(t) &= 1-k_v\int_0^\infty n(L,t)L^3\,\mathrm{d}L \end{aligned} \tag{17}$$

with initial condition

$$c(0)=c_0,$$

and the population balance

$$\frac{\partial n(L,t)}{\partial t}=-G(c(t))\frac{\partial n(L,t)}{\partial L}-\frac{q}{V}(h_f(L,t)+h_p(L))n(L,t), \tag{18}$$

with initial and boundary conditions

$$n(L,0)=n_0(L),\quad n(0,t)=\frac{B(c(t))}{G(c(t))},\quad G(c)=k_g(c-c_s)^g,\quad B(c)=k_b(c-c_s)^b.$$

Here $c(t)$ is the solute concentration and $n(L,t)$ the crystal size distribution (no. of crystals per crystal length L per volume) at time t, also known as nuclei density. Both c and n are assumed space independent due to ideal mixing. The classification functions are

$$h_f(L,t)=\begin{cases} R_1(t) & \text{if } L<L_f\\ 0 & \text{if } L\geqslant L_f\end{cases}\qquad h_p(L)=\begin{cases} 1 & \text{if } L<L_p\\ 1+R_2 & \text{if } L\geqslant L_p.\end{cases}$$

The parameter values are gathered in Table VI and are originally adopted from [28]. Some typical values are given in Table VIII.

In industrial crystallizers operated in open loop, the crystal size distribution (CSD) may exhibit bad damping or sustained oscillations (see Figure 8 left, and Figure 10 top), which affect product quality, hence the need for feedback control. The challenge addressed in [27] (study 1 in Table VII) was to control the system based on measuring only solute concentration $c(t)$ and by acting on the

DOI: 10.1002/rnc

Table VI. Parameters of continuous crystallizer.

Feed rate	q	0.05	$\frac{\ell}{\min}$
Total volume	V	10.5	ℓ
Fines removal cut size	L_f	0.2	mm
Product cut size	L_p	1	mm
Fines removal constant	$R_{1,\mathrm{ss}}$	5	—
Product removal constant	R_2	2	—
Growth rate constant	k_g	0.0305	$\frac{\mathrm{mm}\,\ell}{\min \mathrm{mol}}$
Growth rate exponent	g	1	—
Nucleation rate constant	k_b	$8.36e9$	$\frac{\ell^3}{\min \mathrm{mol}^4}$
Nucleation rate exponent	b	4	—
KCl crystal density	ρ	1989	$\frac{\mathrm{g}}{\ell}$
KCl mole mass	M	74.551	$\frac{\mathrm{g}}{\mathrm{mol}}$
Volumetric shape factor	k_v	$1.112e-7$	$\frac{\ell}{\mathrm{mm}^3}$
Saturation concentration	c_s	4.038	$\frac{\mathrm{mol}}{\ell}$
Crystal size distribution	$n(L,t)$	—	$\frac{\#}{\mathrm{mm}\,\ell}$
Steady-state CSD	$n_{\mathrm{ss}}(L)$	—	idem
Slurry rate	$m(L,t)$	—	$\frac{\mathrm{g}}{\mathrm{mm}}$
Overall crystal mass	$m(t)$	—	g
Solute concentration	$c(t)$	—	$\frac{\mathrm{mol}}{\ell}$
Steady-state SC	c_{ss}	4.0918	idem
Solute feed concentration	$c_f(t)$	—	idem
Steady-state SFC	$c_{f\mathrm{ss}}$	4.4	idem
Fines removal rate	$R_1(t)$	—	—

Table VII. Possible inputs and outputs.

Study	Input(s)	Output(s)	
1	Solute feed concentration $c_f(t)-c_{f\mathrm{ss}}$	Solute concentration $c(t)-c_{\mathrm{ss}}$	SISO [27]
2	Solute feed concentration fines dissolution rate $R_1(t)-R_{1\mathrm{ss}}$	solute concentration overall crystal mass $\rho k_v \int_0^\infty (n(L,t)-n_{\mathrm{ss}}(L))L^3\,\mathrm{d}L$	MIMO

Table VIII. Typical values.

Crystal size (mm)	$0 \leqslant L \leqslant 2$
Slurry rate (g/mm)	$0 \leqslant m(L,t) \leqslant 400$
CSD (1/mm l)	$0 \leqslant n(L,t) \leqslant 1e7$
Feed concentration (mol/l)	$4.3 \leqslant c_f(t) \leqslant 4.5$

solute feed rate $c_f(t)$ alone, while older strategies control the process through the fines dissolution rate $R_1(t)$.

The control strategy in [27] is original. The SISO system $c_f - c_{f\mathrm{ss}} \to c - c_{\mathrm{ss}}$ is considered. The authors do not discretize the population balance (18) with respect to L. Instead, they linearize the system about the steady state $(c_{\mathrm{ss}}, n_{\mathrm{ss}}(L))$ and Laplace transform with respect to time t. The steady state can be computed analytically. The linear ODE in L so obtained is solved analytically. This provides the infinite dimensional transfer function $G(s) = \Delta c(s)/\Delta c_f(s)$. After approximating exponential terms by polynomials, a finite dimensional system is obtained and used in H_∞ control.

A characteristic parameter of the crystallization process is the mass density function $m(L,t) = \rho k_v n(L,t) L^3$, also known as the slurry density or slurry rate. In study 2 we assume that not only solute concentration, $c(t)$, but also the overall crystal mass $m(t) := \int_0^\infty m(L,t)\,\mathrm{d}L$ are measured. In total this leads to two outputs (Δc, Δm) and two inputs (Δc_f, ΔR_1). It is clear that the infinite

DOI: 10.1002/rnc

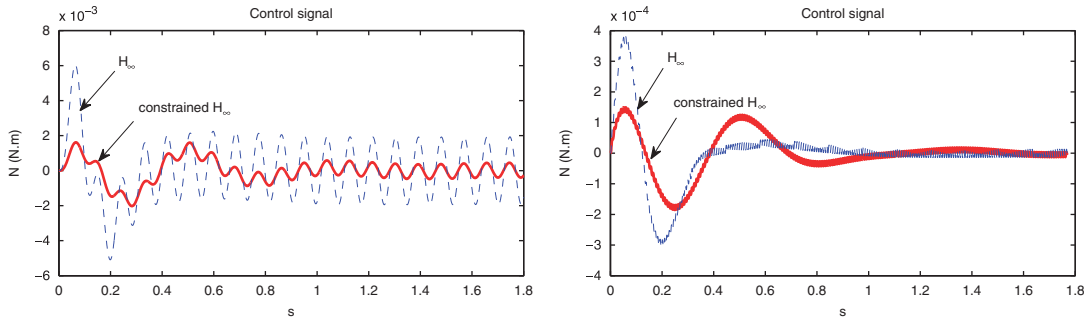

Figure 4. Rotational actuator. Control torque $u = N$ for high frequency (right) and low frequency (left) sinusoidal input. Dashed curves show optimal first-order H_∞-controller $K_{1,\text{init}}$, continuous curves show solutions $K_{1,\text{final}}$ and $K_{3,\text{final}}$ of (8).

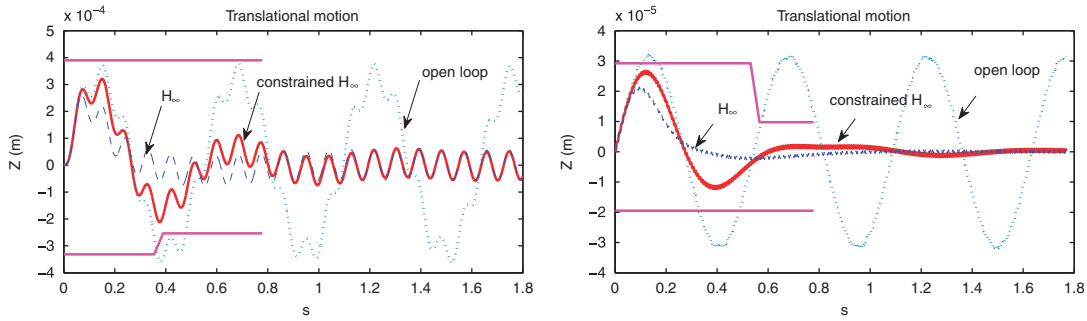

Figure 5. Rotational actuator. Closed-loop system response $\widetilde{Z}(t)$ to sinusoidal input (high frequency right, low frequency left). Dotted curve shows open-loop system, dashed curves show first-order optimal H_∞-controller $K_{1,\text{init}}$, continuous curves show optimal first-order H_∞-controller $K_{1,\text{final}}$ (left) and $K_{3,\text{final}}$ (right) with additional time-domain constraint (solutions of (8)).

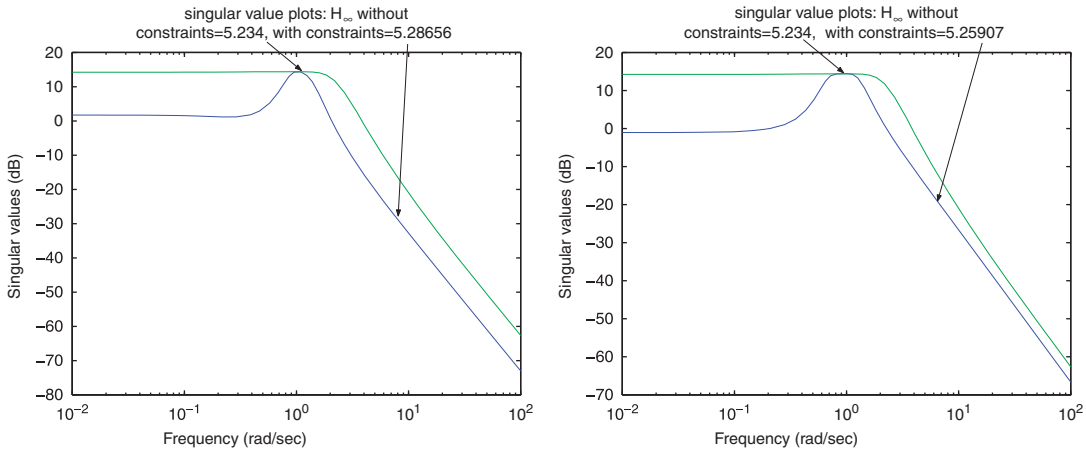

Figure 6. Rotational actuator. Maximum singular value plot for closed-loop transfer channel $T_{F\to W_\xi(Z,\theta,N)}$ at $K_{1,\text{init}}$ (left and right upper curves) versus $K_{1,\text{final}}$ left and $K_{3,\text{final}}$ right lower curves. Satisfying the time domain constraints reduces the H_2-norm, with only a slight increase of the H_∞ norm.

dimensional strategy outlined above is more complicated in the MIMO case (study 2 in Table VII), but still in reach. Here we follow a different line, where control is to be achieved with the most natural discretization techniques.

DOI: 10.1002/rnc

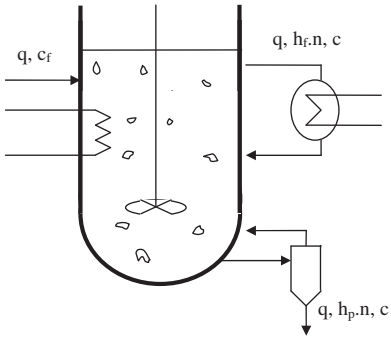

Figure 7. Continuous crystallizer model from [27]. Ideal mixing is assumed, particle breakage and agglomeration are neglected. Growth rate is size independent. Fines are recycled at variable rate $R_1(t)$. Particles are removed at constant rate R_2. Solute feed and fines removal rate are possible controls. Solute concentration and overall crystal mass are possible measurements.

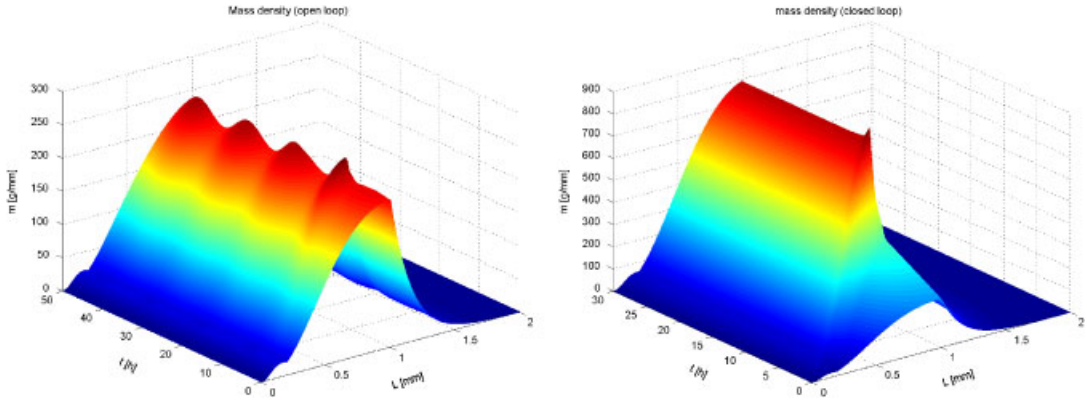

Figure 8. Mass density $m(L,t)$ in open loop (on the left) shows sustained oscillations. Closed loop with structured constrained H_∞ controller (on the right) shows stable behaviour as the system moves from (c_0, n_0) to (c_{ss}, n_{ss}), witnessed by the cylindric form of the surface for large times t. The old steady-state profile $m_0(L)$ can still be seen along the $t=0$ axis, while the new steady-state curve $m_{ss}(L)$ curve is almost established around $t=5$.

The most straightforward and probably also the most natural approach to control a system like (17), (18) is to discretize (18) with respect to crystal size L and apply standard control techniques to the system of ODEs. Surprisingly, this has a bad reputation. The reason is the following: In order to obtain a realistic simulation of the system, between 1000 and 2000 time steps Δt and spatial steps ΔL have to be chosen e.g. for a second-order upwind scheme, which appears too large for controller synthesis. One may argue that for feedback control, a realistic description of the input–output behaviour of the system is sufficient, where high-frequency components will be useless. System reduction should give a better suited model. Figure 12 shows that one should succeed in eliminating most of the open-loop poles, as they extend toward $-\infty$.

Model reduction could be based on the Matlab function `modred` [29]. As the system is unstable in open loop, the function `balreal` (see [29]), which one would typically like to use to identify states that could be ignored, is not available. (Notice that `modred` itself can be used for unstable plants; cf. [29].)

We obtain a reduced-order open-loop system by discretizing the population balance equation with only 250 spatial steps ΔL (as opposed to the 1000 spatial steps used for simulations of the system). In addition, the corners of the functions h_f and h_p had to be mildly smoothed using $h_p(L)=1+0.5R_2(\tanh(\gamma(L-L_p))-\tanh(\gamma(L_p-L)))$ and $h_f(L,R_1)=R_1(0.5+0.5\tanh\gamma(L_f-L))$, because otherwise spurious uncontrollable nodes were introduced in

DOI: 10.1002/rnc

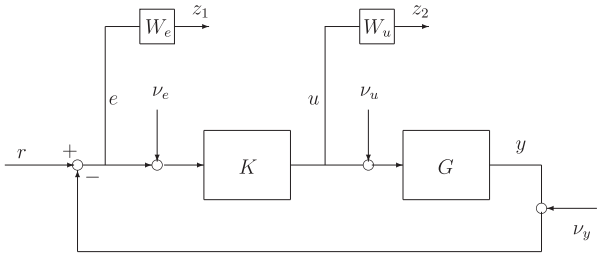

Figure 9. Control scheme for continuous crystallizer.

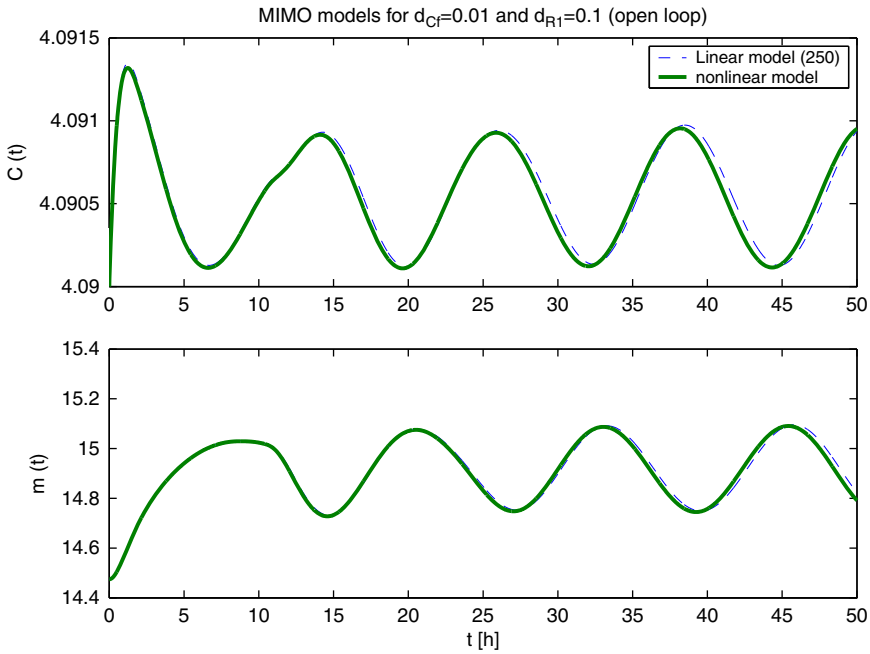

Figure 10. MIMO study. Comparison between nonlinear and linear model in open loop for $\Delta c_f = 0.01$ and $\Delta R_1 = 0.1$.

particular if L_f or L_p were part of the grid. Surprisingly, this is now where traditional synthesis (like LQG or nominal H_∞ synthesis) encounters its bottleneck. Even 250 states are still far beyond the reach of current design techniques, at least in the MIMO case. Functions like `hinfric`, `hinflmi`, `h2syn` or `h2lqg` certainly fail at state-space dimension 250, `hinflmi` already encounters difficulties at state dimension 50 (see [29] for these functions). Moreover, these tools provide full order controllers only, which means that state dimension 50 is still far too big for a practical design.

Now if we reduce the system further in order to comply with the limitations of existing design techniques, let us say to state dimension 30, then we will be in trouble, because the model will be very approximative, and the risk to fail to control or at least to degrade performances will be high. What is worse is that even when this works at size 30, then we will get a 30th order dynamic controller, which, on top of being hard to implement, will have bad performance.

With structured H_∞-control [1] we are no longer hampered by these considerations. We are free to design a second-order locally optimal H_∞ structured controller with two inputs and two outputs from a version of (17), (18) discretized with 250 states. The bad reputation of the above straightforward control strategy is undeserved. The failure is to be blamed on ARE-based design techniques, which are not sufficiently flexible.

DOI: 10.1002/rnc

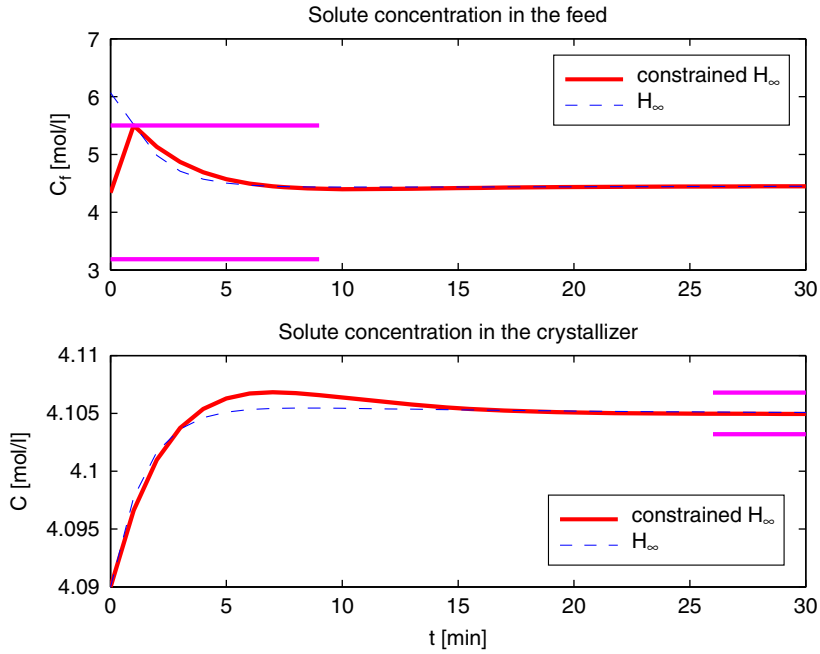

Figure 11. Comparison of responses of structured H_∞ controller (dashed curves) and time constrained structured H_∞ controller (continuous curves) in closed-loop. The anti saturation constraint on $\widetilde{c}_f$ is violated for the structured H_∞ controller (top). The constrained structured H_∞ controller renders the controller less performing (bottom). The time response $\widetilde{c}(t)$ (bottom) has slightly longer rise time and increased overshoot compared to the dotted line. The steady-state behaviour is still within the allowed tube. Introducing the time constraints leads to a reduction of 30% for the maximum value $\widetilde{c}_{f,\max}$.

The scheme for the synthesis we have chosen is the traditional 1-DOF, see Figure 9. The reference signal is $r=(\Delta c_{\text{ref}}, \Delta m_{\text{ref}})\in\mathbb{R}^2$. The control is $u=(\Delta c_f, \Delta m)\in\mathbb{R}^2$, based on the measured output $y=(\Delta c, \Delta m)\in\mathbb{R}^2$. In order to enhance the robustness of the design, we have added measurement noise v_c, v_m and actuator noise v_{c_f}, v_{R_1}, so altogether the exogenous input vector is $w=(r_1, r_2, v_c, v_m, v_{c_f}, v_{R_1})\in\mathbb{R}^6$. Controlled outputs concern the tracking error $e=r-y\in\mathbb{R}^2$, the output y and the control u and are of the form $z=(W_e e, W_u u)\in\mathbb{R}^4$. The 2×2 filters are chosen as

$$W_e=\begin{bmatrix}\dfrac{0.02}{s+0.01} & 0\\ 0 & \dfrac{0.01}{s+0.01}\end{bmatrix},\quad W_u=\begin{bmatrix}\dfrac{0.01s}{s+0.01} & 0\\ 0 & \dfrac{0.01s}{s+0.01}\end{bmatrix},$$

and the filter $W_2=\text{diag}(0.01, 0.01)$ is used at input v_e.

Finally, in order to study time domain constraints for a system of this size, a saturation constraint on the control input $c_f(t)$ is included. As can be seen in Figure 11, this constraint is violated by the structured H_∞-controller, so program (8) is invoked with the H_∞-norm as objective, time domain constraints

$$(1-0.27)c_{\text{fss}}\leqslant\widetilde{c}_f(t)\leqslant(1+0.27)c_{\text{fss}},\ \text{mol/l}\quad\text{during } 0\leqslant t\leqslant 10 \text{ minutes}$$

and

$$(1-0.12)c_{\text{ss}}\leqslant\widetilde{c}(t)\leqslant(1+0.12)c_{\text{ss}},\ \text{mol/l}\quad\text{for } t\geqslant 30 \text{ minutes}$$

were introduced, and a closed-loop stability constraint $\alpha\leqslant -1e-7$ was added.

DOI: 10.1002/rnc

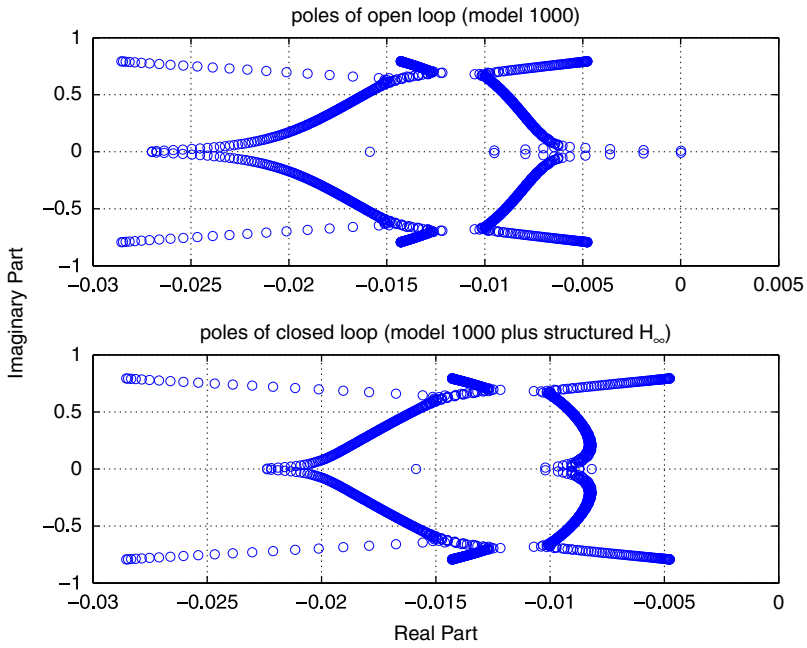

Figure 12. Continuous crystallizer. Open-loop poles top, closed-loop poles bottom.

In the SISO study we start with a second-order H_∞ controller K_1 with $\|T_{c_f\to c}(K_1)\|_\infty = 1.5114$. The time domain constraints are not satisfied and constraint violation is 1.1308. Our algorithm uses K_1 as initial guess and computes a second-order controller K_2 with $\|T_{c_f\to c}(K_2)\|_\infty = 1.6080$ and negligible constraint violation of 1e–3 in 411.984 s cpu. The non-smooth algorithm identifies this as locally optimal within acceptable numerical precision, i.e. $K_2 \approx K_3$. In phase 2 this study incorporates 93 time domain constraints, where

$$c_{\max} = c_{\text{ref}} + 0.12\Delta c, \quad c_{\min} = c_{\text{ref}} - 0.12\Delta c \quad c_{f,\max} = c_{\text{fss}}(1+0.27), \quad c_{f,\min} = c_{\text{fss}}(1-0.27).$$

This second-order controller is tested *a posteriori* by moving the system from a prior steady state $(c_0, n_0(L))$ into the desired steady state $(c_{\text{ss}}, n_{\text{ss}}(L))$ in order to increase crystal production. This could be interpreted as a step response.

In the MIMO study the initial second-order H_∞ controller K_1 satisfies $\|T_{(c_f,R_1)\to(c,m)}(K_1)\|_\infty = 1.6248$ obtained in 691 s cpu (system with 250 states). This controller has maximum control action $R_1 = 56$ and violates the saturation constraint $R_1 \leqslant 30$. It is used as initial guess in phase 2. Our algorithm finds a new second-order controller K_2 with $\|T_{(c_f,R_1)\to(c,m)}(K_2)\|_\infty = 1.7543$ and constraint violation 4.22e–6 within 1975.6 s cpu. Phase 3 shows again optimality within an acceptable numerical precision. This study incorporates 181 mathematical programming constraints, which are $\alpha \leqslant -1e-3$, $0 \leqslant R_1(t) \leqslant 30$, and

$$c_{\max} = c_{\text{ref}} + 0.15\Delta c, \quad c_{\min} = c_{\text{ref}} - 0.15\Delta c, \quad m_{\max} = m_{\text{ref}} + 0.15\Delta m, \quad m_{\min} = m_{\text{ref}} - 0.15\Delta m.$$

A snapshot of the process can be seen in Figure 8 on the right, where the slurry rate $m_0(L) = k_v L^3 n_0(L)$ of the old steady state is still visible along the $t = 0$ axis, whereas the new steady-state $m_{\text{ss}}(L)$ can be recognized for later times $t \geqslant 5$ h, where the cylindric form characterizing a stable steady state is about to establish itself (Figure 8, right). Step responses are shown and compared in Figures 13 and 14.

DOI: 10.1002/rnc

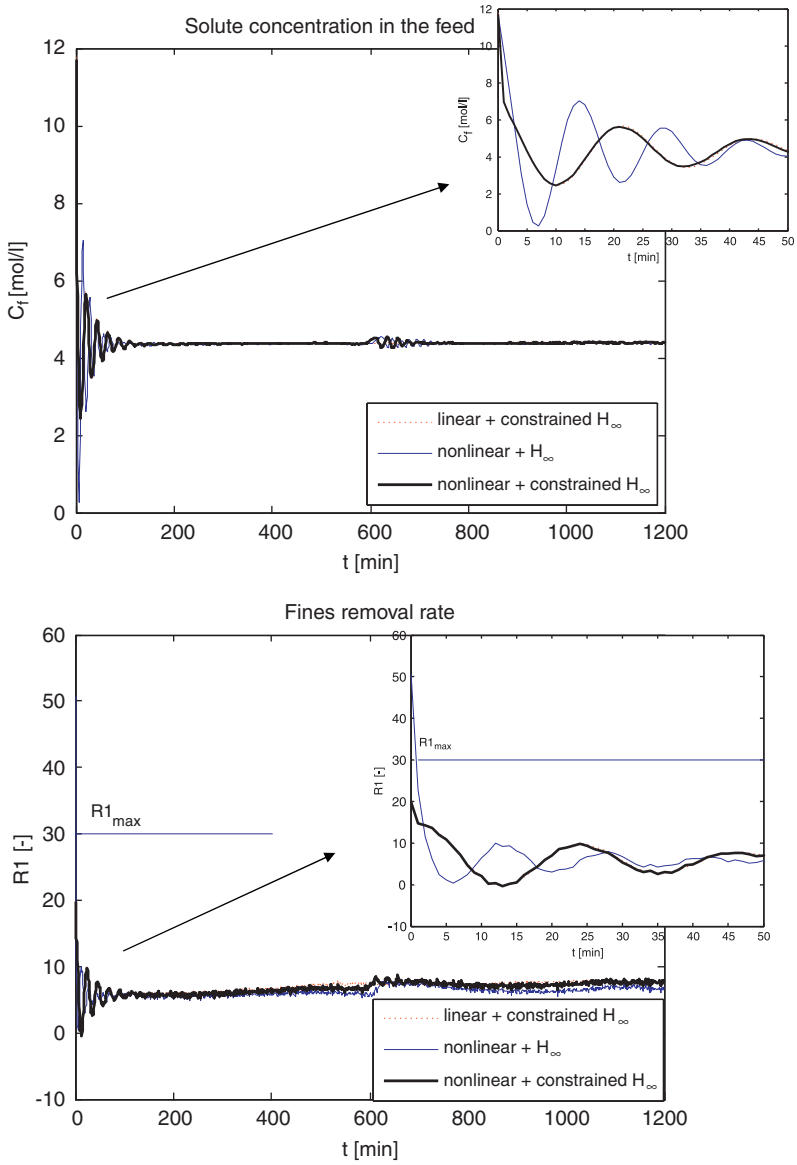

Figure 13. MIMO study. Closed-loop controls $\widetilde{c}_f(t)$ and $\widetilde{R}_1(t)$ for structured H_∞ controller with and without time constraints, linear and nonlinear. Inclusion of the time domain constraints leads to a 42.9% reduction of the fines dissolution rate overshoot. Namely $R_{1,ss}=5$, $\widetilde{R}_{1,\max,\text{constr}}=20$, $\widetilde{R}_{1,\max,\text{unconstr}}=51$, meaning that the gap in the unconstrained case is reduced from $|51-30|=21$, to $|21-30|=9$. The transient responses are also improved in the time constrained case.

11. CONCLUSION

We have introduced constrained structured H_∞ synthesis, a set-up which allows to optimize frequency domain and time domain performance specifications simultaneously in a single optimization program. Expanding on non-smooth techniques developed earlier for structured H_∞ synthesis [1, 30–33], we have proposed and validated a three-stage algorithm which combines SQP techniques with non-smooth optimization. Our method finds solutions in surprisingly small CPU times even for sizeable systems. Two benchmark examples have been tested and the results corroborate that our method is more versatile than synthesis techniques based on algebraic Riccati

DOI: 10.1002/rnc

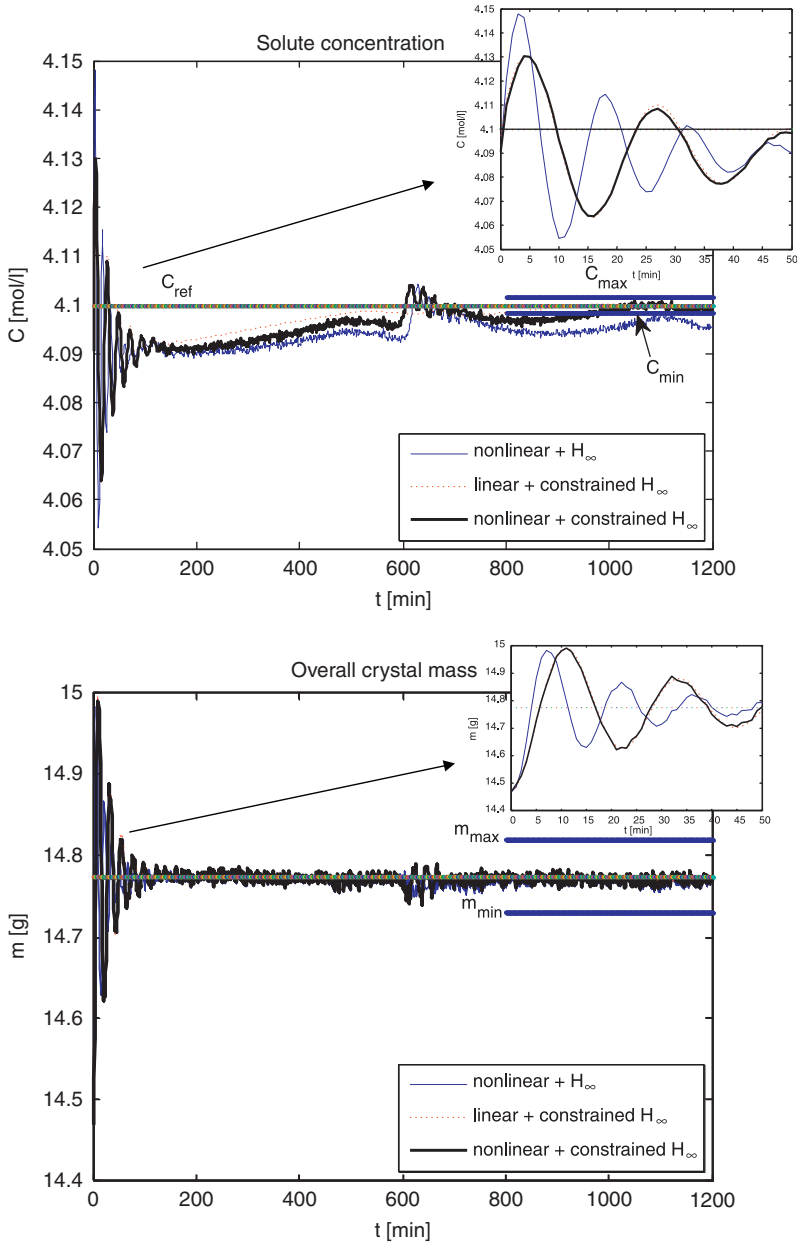

Figure 14. MIMO study. Outputs $\widetilde{c}(t)$, and $\widetilde{m}(t)$ in closed loop with various controllers. With time constraints the regulation error in $\widetilde{c}$ is reduced from $(4.1-4.096)/(4.1-4.092)=50\%$ in the unconstrained case down to $(4.1-4.099)/(4.1-4.092)=12.5\%$ after adding constraints. The transient responses are also improved.

equations or matrix inequalities. In particular, our approach makes the H_∞ control paradigm open to arbitrary controller structures and time domain specifications.

ACKNOWLEDGEMENTS

This work was supported by research grant *Guidage* from Agence Nationale de Recherche (ANR), research grant *Survol* from Fondation de Recherche pour l'Aéronautique et l'Espace (FNRAE), and research grant *Technicum* from Foundation EADS.

DOI: 10.1002/rnc

REFERENCES

1. Apkarian P, Noll D. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control* 2006; **51**:71–86.
2. Gevers M. A decade of progress in iterative process control design: from theory to practice. *Journal of Process Control* 2002; **12**:519–531.
3. Boyd S, Barratt CH. *Linear Controller Design. Limits of Performance*. Prentice-Hall: Englewood Cliffs, NJ, 1991.
4. Doyle J, Glower K, Kargonekar PP, Francis B. State-space solutions to the standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control* 1989; **34**(8):831–847.
5. Helton WT, James MR. *Extending H_∞ Control to Nonlinear Systems*. SIAM Series in Design and Control. SIAM: Philadelphia, 1999.
6. Apkarian P, Bompart V, Noll D. Control design in the time and frequency domain using nonsmooth techniques. *Systems and Control Letters* 2008; **57**(3):271–282.
7. Apkarian P, Noll D, Simões A. A nonsmooth progress function algorithm for frequency shaping control design. *IET Control Theory and Applications* 2008; **2**(4):323–336.
8. Apkarian P, Noll D, Simões A. Time domain control design: a nonsmooth approach. *IEEE Transactions on Control Systems Technology* 2009; **17**(6):1439–1445.
9. Simões AM, Apkarian P, Noll D. Nonsmooth multi-objective synthesis with applications. *Control Engineering Practice* 2009; **17**(11):1338–1348.
10. The Control System Toolbox. The MathWorks Inc.: Natick, MA.
11. Griewank A. *Evaluating Derivatives*. SIAM Frontiers in Applied Mathematics.SIAM: Philadelphia, 2000.
12. Clarke F. *Optimization and Nonsmooth Analysis*. Canadian Mathematical Society Series. Wiley: New York, 1983.
13. Boyd S, Balakrishnan V. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm. *Systems and Control Letters* 1990; **15**:1–7.
14. Bompart V, Apkarian P, Noll D. Nonsmooth techniques for stabilizing linear systems. *American Control Conference*, New York, Conference Proceedings, 2007; 1245–1250.
15. Burke JV, Lewis AS, Overton ML. Two numerical methods for optimizing matrix stability. *Linear Algebra and its Applications* 2002; **351**:117–145.
16. Burke JV, Henrion D, Lewis AS, Overton ML. Stabilization via nonsmooth nonconvex optimization. *IEEE Transactions on Automatic Control* 2006; **51**:1760–1769.
17. Gumussoy S, Millstone M, Overton ML. H_∞ strong stabilization via HIFOO, a package for fixed-order controller design. *Proceedings of 47th IEEE Conference on Decision and Control* (CDC 2008), Cancun, 9–11 December 2008; 4135–4140.
18. Powell MJD. *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations*. Lecture Notes in Mathematics, vol. 630. Springer: Berlin, 1978.
19. Schittkowski K. The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function I and II. *Numerische Mathematik* 1981; **38**:83–114 and 115–127.
20. The Optimization Toolbox. The MathWork Inc.: Natick, MA.
21. Bompart V, Noll D, Apkarian P. Second order nonsmooth optimization for feedback control. *Numerische Mathematik* 2007; **107**(3):433–454.
22. Apkarian P, Noll D, Prot O. A trust region spectral bundle method for nonconvex eigenvalue optimization. *SIAM Journal on Optimization* 2008; **19**(1):281–306.
23. Apkarian P, Noll D, Rondepierre A. Mixed H_2/H_∞ control via nonsmooth optimization. *SIAM Journal on Control and Optimization* 2008; **47**(3):1516–1546.
24. Noll D, Prot O, Rondepierre A. A proximity control algorithm to minimize nonsmooth and nonconvex functions. *Pacific Journal of Optimization* 2008; **4**(3):569–602.
25. Bupp RT, Bernstein DS, Coppola VT. A benchmark problem for nonlinear control design. *International Journal of Robust and Nonlinear Control* 1998; **8**:307–310.
26. Balakrishnan V, Boyd S. On computing the worst-case peak gain of linear systems. *Systems and Control Letters* 1992; **19**(4):265–269.
27. Vollmer U, Raisch J. H_∞ control of a continuous crystallizer. *Control Engineering Practice* 2001; **9**(8):837–845.
28. Menges M. Modellierung und dynamische Simulation von Populationssystemen. *Masters Thesis*, Universitat Stuttgart, 1993.
29. The Robust Control Toolbox. The MathWorks Inc.: Natick, MA.
30. Apkarian P, Noll D. Nonsmooth optimization for multidisk H_∞ synthesis. *European Journal of Control* 2006; **12**(3):229–244.
31. Apkarian P, Noll D. Controller design via nonsmooth multi-directional search. *SIAM Journal of Control and Optimization* 2006; **44**:1923–1949.
32. Apkarian P, Noll D. Nonsmooth optimization for multiband frequency domain control design. *Automatica* 2007; **43**(4):724–731.
33. Apkarian P, Bompart V, Noll D. Nonsmooth structured control design with applications to PID loop-shaping of a process. *International Journal of Robust and Nonlinear Control* 2007; **17**(14):1320–1342.

DOI: 10.1002/rnc