INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2007; **17**:1320–1342
Published online 31 January 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/rnc.1175

Non-smooth structured control design with application to PID loop-shaping of a process

Pierre Apkarian[1,2,*,†], Vincent Bompart[1,2] and Dominikus Noll[2]

[1] *ONERA-CERT, Centre d'études et de recherches de Toulouse, Control System Department, 2 av. Edouard Belin, Toulouse 31055, France*
[2] *Institut de Mathématiques, Université Paul Sabatier, 118, route de Narbonne, Toulouse 31062, France*

SUMMARY

Feedback controllers with specific structure arise frequently in applications because they are easily apprehended by design engineers and facilitate on-board implementations and re-tuning. This work is dedicated to H_∞ synthesis with structured controllers. In this context, straightforward application of traditional synthesis techniques fails, which explains why only a few *ad hoc* methods have been developed over the years. In response, we propose a more systematic way to design H_∞ optimal controllers with fixed structure using local optimization techniques. Our approach addresses in principle all those controller structures which can be built into mathematical programming constraints. We apply non-smooth optimization techniques to compute locally optimal solutions, and provide practical tests for descent and optimality. In the experimental part we apply our technique to H_∞ loop-shaping proportional integral derivative (PID) controllers for MIMO systems and demonstrate its use for PID control of a chemical process. Copyright © 2007 John Wiley & Sons, Ltd.

Received 19 October 2006; Accepted 4 December 2006

KEY WORDS: non-smooth optimization; H_∞ synthesis; structured controllers; PID; *NP*-hard problems

1. INTRODUCTION

Considerable efforts have been made over the past two decades to develop new and powerful control methodologies. H_∞ synthesis [1] is certainly the most prominent outcome of this search. In spite of its theoretical success, it turns out that structured controllers such as proportional integral derivative (PID), lead-lag, observed based, and others, are still preferred in industrial

*Correspondence to: Pierre Apkarian, ONERA-CERT, Centre d'études et de recherches de Toulouse, Control System Department, 2 av. Edouard Belin, Toulouse 31055, France.
†E-mail: apkarian@cert.fr

Contract/grant sponsor: Fondation d'entreprise EADS
Contract/grant sponsor: Agence Nationale de Recherche (ANR); contract/grant number: NT05

control. The reason is that controllers designed with modern control techniques are usually of high order, difficult to implement and often impossible to re-tune in case of model changes. But those are precisely the properties which make structured controllers so popular for practitioners. Easy to implement and to understand, and easy to re-tune whenever performance or stability specifications change. The trade-off may be roughly described as high performance combined with low flexibility *versus* lower performance combined with high flexibility. The question we ask here is how the performance properties of structured controllers may be improved.

Structured control design is generally a difficult problem. Even the simple static output feedback stabilization problem is known to be *NP*-hard [2]. Due to their importance for practice, a number of innovative techniques and heuristics for structured control have been proposed in the literature. Some authors use branch-and-bound techniques to construct globally optimal solution to the design problem [3]. In the same vein, Wong and Bigras [4] proposed evolutionary optimization to reduce the computational overhead, while still aiming at globally optimal solutions. These approaches are certainly of interest for small problems, but quickly succumb when problems get sizable.

A fairly disparate set of heuristic techniques for structured control design was developed in the realm of linear matrix inequalities (LMIs) [5]. Alternating projection techniques were proposed in [6] for static controller and for the more specific structured control design problem in [7]. In the same vein, coordinate descent schemes were proposed in [8, 9, 10]. In [11], the authors suggest using a BMI formulation for solving a variety of problems including structured control. These techniques may be useful in practice, but they bear the risk of missing a local solution, because convergence to a local minimum is rarely ensured. Iterative solving of SDPs based on successive linearizations is yet another idea, but often leads to prohibitive running times. In [12], 2 h cputime was necessary to compute a decentralized PID controller for a 2×2 process on a Pentium *II* 333 MHz computer. Even longer cputimes are reported in [13] for medium size PID design problems.

A relatively rich literature addresses specific controller structures such as decentralized or PIDs. In [14], Miyamoto and Vinnicombe discuss a coordinate scheme for H_∞ loop-shaping with decentralized constraints. In [15], again in the loop-shaping context, the authors adopt a truncation procedure to reduce a full-order controller to a PID controller. Those are heuristic procedures, because closed-loop performance is not necessarily inherited by the final controller. In [16], Saeki addresses sufficient conditions under which PID synthesis reduces to solving LMIs.

In a recent interesting work, Rotkowitz and Lall [17] fully characterize a class of problems for which structured controller design can be solved using convex programming. They introduce the concept of *quadratic invariance* and show that for problems having this property, optimal structured controllers may be efficiently synthesized. In a different but related work [18], the authors identify various control structures that are amenable to convex optimization by an appropriate choice of coprime factors in the Youla–Kucera parametrization [19]. A similar analysis is made by Scherer in [20] both for structured controller design and multi-objective H_∞ control. Unfortunately, these concepts and tools only apply to very particular problem classes and controller patterns and do not easily lend themselves to generalization when finer controller structures are required.

In our opinion local optimization is the approach best suited for these difficult design problems. It should whenever possible be used in tandem with heuristic methods, as those may be useful to compute good initial points for the optimization. We mention that early approaches to structured design based on tailored optimization techniques can be traced back to the work of

DOI: 10.1002/rnc

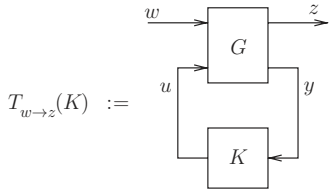

Figure 1. Standard interconnection.

Mäkilä and Toivonen [21] for parametric LQ problems, or Polak and Wardi [22] for problems with frequency domain singular value inequalities. In the latter reference, many design problems are recognized as non-differentiable, and in consequence, techniques from non-smooth analysis are employed. More recently, we have used non-smooth analysis to fully characterize the subdifferential properties of closed-loop mappings of the form $||\cdot||_\infty \circ T_{w\to z}$ acting on the controller space, where $T_{w\to z}(K)$ denotes the closed-loop transfer function from w to z at a given controller K, Figure 1. These results are used to develop non-smooth descent algorithms for various design problems [23–25]. Here, we extend our results to structured controller design and elaborate the case of multi-input multi-output (MIMO) PID controllers. We then demonstrate how the proposed technique can be used in the context of PID loop-shaping as introduced in [13, 26]. We conclude with an application to chemical process control.

2. NON-SMOOTH H_∞ DESIGN TECHNIQUE

Consider a plant P in state-space form

$$P(s): \quad \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector of P, $u \in \mathbb{R}^{m_2}$ the vector of control inputs, $w \in \mathbb{R}^{m_1}$ the vector of exogenous inputs, $y \in \mathbb{R}^{p_2}$ the vector of measurements and $z \in \mathbb{R}^{p_1}$ the controlled or performance vector. Without loss, it is assumed throughout that $D_{22} = 0$.

The focus is on H_∞ synthesis with structured controllers, which consists in designing a dynamic output feedback controller $K(s)$ with feedback law $u = K(s)y$ for the plant in (1) having the following properties:

- *Controller structure*: $K(s)$ has a prescribed structure.
- *Internal stability*: $K(s)$ stabilizes the original plant $P(s)$ in closed-loop.
- *Performance*: Among all stabilizing controllers with that structure, $K(s)$ minimizes the H_∞ norm $||T_{w\to z}(K)||_\infty$. Here $T_{w\to z}(K)$ denotes the closed-loop transfer function from w to z.

2.1. Subdifferential of the H_∞ map

For the time being we leave apart structural constraints and assume that $K(s)$ has the frequency domain representation

$$K(s) = C_K(sI - A_K)^{-1}B_K + D_K, \quad A_K \in \mathbb{R}^{k\times k} \tag{2}$$

DOI: 10.1002/rnc

where k is the order of the controller, and where the case $k = 0$ of a static controller $K(s) = D_K$ is included. A further simplification is obtained if we assume that preliminary dynamic augmentation of the plant $P(s)$ has been performed

$$A \to \begin{bmatrix} A & 0 \\ 0 & 0_k \end{bmatrix}, \quad B_1 \to \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \text{ etc}$$

so that manipulations will involve a static matrix

$$\mathscr{K} := \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \tag{3}$$

With this proviso, the following closed-loop notations will be useful:

$$\begin{bmatrix} \mathscr{A}(\mathscr{K}) & \mathscr{B}(\mathscr{K}) \\ \mathscr{C}(\mathscr{K}) & \mathscr{D}(\mathscr{K}) \end{bmatrix} := \begin{bmatrix} A & B_1 \\ C_1 & D_{11} \end{bmatrix} + \begin{bmatrix} B_2 \\ D_{12} \end{bmatrix} \mathscr{K} [C_2 \ D_{21}] \tag{4}$$

Owing to its special composite structure, the function $f = \|\cdot\|_\infty \circ T_{w\to z}$, which maps the set $\mathscr{D} \subset \mathbb{R}^{(m_2+k)\times(p_2+k)}$ of stabilizing controllers into $\mathbb{R}^+$, is Clarke subdifferentiable [23, 27, 28]. Its Clarke subdifferential can be described as follows. Introduce the set of active frequencies at a given $\mathscr{K}$

$$\Omega(\mathscr{K}) := \{\omega \in [0, +\infty] : \bar{\sigma}(T_{w\to z}(\mathscr{K}, \mathrm{j}\omega)) = f(\mathscr{K})\} \tag{5}$$

We assume throughout that $\Omega(\mathscr{K})$ is a finite set and we refer the reader to [29] for a justification of this hypothesis. We shall also need the notation

$$\begin{bmatrix} T_{w\to z}(\mathscr{K}, s) & G_{12}(\mathscr{K}, s) \\ G_{21}(\mathscr{K}, s) & \bigstar \end{bmatrix} := \begin{bmatrix} \mathscr{C}(\mathscr{K}) \\ C_2 \end{bmatrix} (sI - \mathscr{A}(\mathscr{K}))^{-1} [\mathscr{B}(\mathscr{K}) \ B_2] + \begin{bmatrix} \mathscr{D}(\mathscr{K}) & D_{12} \\ D_{21} & \bigstar \end{bmatrix} \tag{6}$$

This leads to the following result.

Theorem 2.1
Assume the controller $K(s)$ stabilizes $P(s)$ in (1), that is, $\mathscr{K} \in \mathscr{D}$. With notations (5) and (6), let Q_ω be a matrix whose columns form an orthonormal basis of the eigenspace of $T_{w\to z}(\mathscr{K}, \mathrm{j}\omega) T_{w\to z}(\mathscr{K}, \mathrm{j}\omega)^H$ associated with the largest eigenvalue $\lambda_1(T_{w\to z}(\mathscr{K}, \mathrm{j}\omega) T_{w\to z}(\mathscr{K}, \mathrm{j}\omega)^H) = \bar{\sigma}(T_{w\to z}(\mathscr{K}, \mathrm{j}\omega))^2$. Then, the Clarke subdifferential of the mapping f at $\mathscr{K} \in \mathscr{D}$ is the compact and convex set $\partial f(\mathscr{K}) = \{\Phi_Y : Y \in \mathscr{S}(\mathscr{K})\}$, where

$$\Phi_Y = f(\mathscr{K})^{-1} \sum_{\omega \in \Omega(\mathscr{K})} \Re\{G_{21}(\mathscr{K}, \mathrm{j}\omega) \, T_{w\to z}(\mathscr{K}, \mathrm{j}\omega)^H Q_\omega Y_\omega (Q_\omega)^H G_{12}(\mathscr{K}, \mathrm{j}\omega)\}^{\mathrm{T}} \tag{7}$$

and $S(\mathscr{K})$ is the spectraplex set

$$\mathscr{S}(\mathscr{K}) = \{Y = (Y_\omega)_{\omega \in \Omega(\mathscr{K})} : Y_\omega = (Y_\omega)^H \succcurlyeq 0, \ \sum_{\omega \in \Omega(\mathscr{K})} \mathrm{Tr}\ Y_\omega = 1\} \tag{8}$$

Proof
The proof is based on the chain rule for the Clarke gradient of the composite mapping $f = \|\cdot\|_\infty \circ T_{w\to z}$ and we refer the reader to [30] and [23, 28, 31] for a proof and further details. □

DOI: 10.1002/rnc

In geometric terms, the subdifferential of f is a linear image of the spectraplex set $S(\mathscr{K})$. We shall see later that it reduces to a more familiar geometric set under additional assumptions on the multiplicity of the singular values.

2.2. Structured controllers

Note that we have assumed so far that controllers have no specific structure. We now extend the results in Section 2.1 to structured controllers using chain rules.

Assume $\mathscr{K}$ defined in (3) depends smoothly on a free parameter $\kappa \in \mathbb{R}^q$, that is, $\mathscr{K} = \mathscr{K}(\kappa)$, where $\mathscr{K}(\cdot)$ is smooth. Then the subgradients with respect to κ of the mapping $g = ||\cdot||_\infty \circ T_{w\to z}(\cdot) \circ \mathscr{K}(\cdot)$ at κ are obtained as $\mathscr{K}'(\kappa)^* \partial f(\mathscr{K})$, where $\partial f(\mathscr{K})$ is given in Theorem 2.1, $\mathscr{K}'(\kappa)$ is the derivative of $\mathscr{K}(\cdot)$ at κ, and where $\mathscr{K}'(\kappa)^*$ is its adjoint. This is a direct application of the chain rule in [30]. Note that the adjoint $\mathscr{K}'(\kappa)^*$ acts on elements $F \in \mathbb{R}^{(m_2+k)\times(p_2+k)}$ *via*

$$\mathscr{K}'(\kappa)^* F = \left[\mathrm{Tr}\left(\frac{\partial \mathscr{K}(\kappa)}{\partial \kappa_1}^{\mathrm{T}} F\right), \ldots, \mathrm{Tr}\left(\frac{\partial \mathscr{K}(\kappa)}{\partial \kappa_q}^{\mathrm{T}} F\right)\right]^{\mathrm{T}}$$

We infer the following.

Corollary 2.2
Assume the controller $\mathscr{K}(\kappa)$ stabilizes $P(s)$ in (1), that is, $\mathscr{K}(\kappa) \in \mathscr{D}$. With the notations of Theorem 2.1, the Clarke subdifferential of the mapping $g = ||\cdot||_\infty \circ T_{w\to z}(\cdot) \circ \mathscr{K}(\cdot)$ at $\kappa \in \mathbb{R}^q$ is the compact and convex set

$$\partial g(\kappa) = \left\{ \left[\mathrm{Tr}\left(\frac{\partial \mathscr{K}(\kappa)}{\partial \kappa_1}^{\mathrm{T}} \Phi_Y\right), \ldots, \mathrm{Tr}\left(\frac{\partial \mathscr{K}(\kappa)}{\partial \kappa_q}^{\mathrm{T}} \Phi_Y\right)\right]^{\mathrm{T}} : \Phi_Y \in \partial f(\mathscr{K}(\kappa)) \right\}. \tag{9}$$

Using vectorization, the subgradients in (9) can be rewritten as

$$\left[\mathrm{vec}\,\frac{\partial \mathscr{K}(\kappa)}{\partial \kappa_1}, \ldots, \mathrm{vec}\,\frac{\partial \mathscr{K}(\kappa)}{\partial \kappa_q}\right]^{\mathrm{T}} \mathrm{vec}\,\Phi_Y \tag{10}$$

An important special case in practice is when the maximum singular values $\bar{\sigma}(T_{w\to z}(\mathscr{K}(\kappa), \mathrm{j}\omega))$ have multiplicity one for every $\omega \in \Omega(\mathscr{K}(\kappa))$. Then, the subgradients Φ_Y reduce in vector form to $\mathrm{vec}\,\Phi_Y = \Psi\xi$ where $\sum_{\omega\in\Omega(\mathscr{K}(\kappa))} \xi_\omega = 1$, $\xi_\omega \geqslant 0, \forall \omega \in \Omega(\mathscr{K}(\kappa))$ and matrix Ψ is constructed columnwise as

$$\Psi := (\mathrm{vec}\,\Re\{G_{21}(\mathscr{K}, \mathrm{j}\omega)\, T_{w\to z}(\mathscr{K}, \mathrm{j}\omega)^H Q_\omega (Q_\omega)^H G_{12}(\mathscr{K}, \mathrm{j}\omega)\}^{\mathrm{T}})_{\omega\in\Omega(\mathscr{K}(\kappa))}$$

Combining this expression with (10), the subdifferential $\partial g(\kappa)$ at κ admits a simpler representation in the form of a linear image of a simplex

$$\partial g(\kappa) = \left\{ \left[\mathrm{vec}\,\frac{\partial \mathscr{K}(\kappa)}{\partial \kappa_1}, \ldots, \mathrm{vec}\,\frac{\partial \mathscr{K}(\kappa)}{\partial \kappa_q}\right]^{\mathrm{T}} \Psi\xi : \sum_{\omega\in\Omega(\mathscr{K}(\kappa))} \xi_\omega = 1, \xi_\omega \geqslant 0, \forall \omega \in \Omega(\mathscr{K}(\kappa)) \right\}$$

DOI: 10.1002/rnc

2.3. PID controllers

In this section we specialize the above results to PID controllers. A common representation of MIMO PID controllers is

$$K(s) = K_{\mathrm{p}} + \frac{K_{\mathrm{i}}}{s} + \frac{K_{\mathrm{d}} s}{1 + \epsilon s} \tag{11}$$

where K_{p}, K_{i} and K_{d} are the proportional, the integral and the derivative gains, respectively. The PID gains K_{p}, K_{i} and K_{d} all belong to $\mathbb{R}^{m_2 \times m_2}$ for a square plant with m_2 inputs and outputs. ϵ is a small scalar which determines how close the last term in (11) comes to a pure derivative action. Using partial fraction expansion, an alternative representation can be obtained in the form

$$K(s) = D_K + \frac{R_{\mathrm{i}}}{s} + \frac{R_{\mathrm{d}}}{s + \tau} \tag{12}$$

with the correspondence

$$D_K \coloneqq K_{\mathrm{p}} + \frac{K_{\mathrm{d}}}{\epsilon}, \quad R_{\mathrm{i}} \coloneqq K_{\mathrm{i}}, \quad R_{\mathrm{d}} \coloneqq -\frac{K_{\mathrm{d}}}{\epsilon^2}, \quad \tau \coloneqq \frac{1}{\epsilon}$$

Note that these two representations are in one-to-one correspondence *via*

$$K_{\mathrm{d}} = -\epsilon^2 R_{\mathrm{d}}, \quad K_{\mathrm{p}} = D_K + \epsilon R_{\mathrm{d}}, \quad K_{\mathrm{i}} = R_{\mathrm{i}}, \quad \epsilon = \frac{1}{\tau}$$

From (12) we obtain a linearly parameterized state-space representation of a MIMO PID controller

$$\mathscr{K} = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right] = \left[\begin{array}{cc|c} 0 & 0 & R_{\mathrm{i}} \\ 0 & -\tau I & R_{\mathrm{d}} \\ \hline I & I & D_K \end{array}\right], \quad A_K \in \mathbb{R}^{2m_2 \times 2m_2} \tag{13}$$

Since the state-space representation of the PID controller is affine in the parameters τ, R_{i}, R_{d} and D_k, the same is true for its vectorized form and we can write

$$\mathrm{vec}\left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right] = \mathrm{vec}\left[\begin{array}{cc|c} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline I & I & 0 \end{array}\right] + T \overbrace{\begin{bmatrix} \tau \\ \mathrm{vec}\, R_{\mathrm{i}} \\ \mathrm{vec}\, R_{\mathrm{d}} \\ \mathrm{vec}\, D_K \end{bmatrix}}^{\kappa}$$

for a suitable matrix $T \in \mathbb{R}^{(k+m_2)(k+p_2) \times (3m_2^2+1)}$. The linear part of (13) can be expanded as

$$\begin{pmatrix} 0 \\ -I \\ 0 \end{pmatrix} \tau I \begin{pmatrix} 0 \\ I \\ 0 \end{pmatrix}^{\mathrm{T}} + \begin{pmatrix} I \\ 0 \\ 0 \end{pmatrix} R_{\mathrm{i}} \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix}^{\mathrm{T}} + \begin{pmatrix} 0 \\ I \\ 0 \end{pmatrix} R_{\mathrm{d}} \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix}^{\mathrm{T}} + \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix} D_K \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix}^{\mathrm{T}}$$

DOI: 10.1002/rnc

In consequence, using the Kronecker product to convert matrix products into vectors, we obtain the transformation

$$T = \left[\begin{pmatrix} 0 \\ I \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ -I \\ 0 \end{pmatrix} \operatorname{vec} I \quad \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix} \otimes \begin{pmatrix} I \\ 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix} \otimes \begin{pmatrix} 0 \\ I \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix} \right]$$

Finally, the subdifferential of the mapping $g = \|\cdot\|_\infty \circ T_{w\to z}(\cdot) \circ \mathscr{K}(\cdot)$ at κ, where $\mathscr{K}(\kappa)$ describes a MIMO PID controller (11) or (12) above, is the compact and convex set of subgradients

$$\partial g(\kappa) = \{T^{\mathrm{T}} \operatorname{vec} \Phi_Y : \Phi_Y \in \partial f(\mathscr{K}(\kappa))\} \tag{14}$$

For a decentralized MIMO PID controller, R_{i}, R_{d} and D_K reduce to diagonal matrices. Introducing the linear transformation

$$J \coloneqq [e_1 \otimes e_1 \quad \ldots \quad e_{m_2} \otimes e_{m_2}]$$

where $(e_i)_{i=1,\ldots,m_2}$ is the canonical basis of $\mathbb{R}^{m_2}$, it is easily verified that

$$\operatorname{vec} M = J[M_{11}, M_{22}, \ldots, M_{m_2 m_2}]^{\mathrm{T}}$$

for any square matrix M of size m_2. This leads to

$$\begin{bmatrix} \tau \\ \operatorname{vec} R_{\mathrm{i}} \\ \operatorname{vec} R_{\mathrm{d}} \\ \operatorname{vec} D_K \end{bmatrix} = \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & J & 0 & 0 \\ 0 & 0 & J & 0 \\ 0 & 0 & 0 & J \end{bmatrix}}^{L} \overbrace{\begin{bmatrix} \tau \\ \operatorname{vec}\operatorname{diag} R_{\mathrm{i}} \\ \operatorname{vec}\operatorname{diag} R_{\mathrm{d}} \\ \operatorname{vec}\operatorname{diag} D_K \end{bmatrix}}^{\kappa}$$

with the new parameter vector κ as indicated above. Again by the chain rule for Clarke subdifferentials, we obtain that the subdifferential of the mapping $g = \|\cdot\|_\infty \circ T_{w\to z}(\cdot) \circ \mathscr{K}(\cdot)$ at κ, where $\mathscr{K}(\kappa)$ describes a MIMO decentralized PID controller stabilizing (1), is the compact and convex set of subgradients

$$\partial g(\kappa) = \{L^{\mathrm{T}} T^{\mathrm{T}} \operatorname{vec} \Phi_Y : \Phi_Y \in \partial f(\mathscr{K}(\kappa))\}$$

We emphasize that the outlined procedure to determine subdifferentials of various types of PID controllers is general and encompasses most controller structures. In particular, this includes all structures $\mathscr{K} = \mathscr{K}(\kappa)$ with a differentiable parametrization $\mathscr{K}(\cdot)$. In addition, in some cases the parameter $\kappa \in \mathbb{R}^q$ may be restricted to a constraint subset of $\mathbb{R}^q$.

2.4. Setpoint filter design

Despite the improvement in performance achieved by our new technique, using PID feedback alone may not be sufficient to meet suitable time-domain constraints. Traditionally, this difficulty is overcome by using a two-degree of freedom strategy including feedback and

DOI: 10.1002/rnc

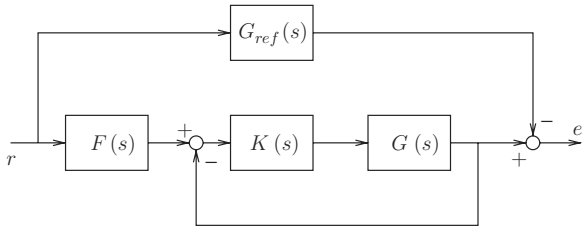

Figure 2. Setpoint filter design.

prefiltering or setpoint filtering. Setpoint filters operate on the reference signals to improve responses properties such as settling-time, overshoot and decouplings. In Figure 2, a typical model following strategy is shown. The setpoint filter $F(s)$ is used in such a way that the responses of the feedback controlled plant $G(s)$ match as closely as possible those of a reference model $G_{\text{ref}}(s)$. Finding such a filter could also be cast as an H_∞ synthesis problem, where the transfer function from the reference signal r to the tracking error e is minimized

$$\underset{F(s)}{\text{minimize}} \; ||T_{r\to e}(F)||_\infty \tag{15}$$

To solve the setpoint filter design problem, we suggest once again the use of non-smooth optimization methods. In order to illustrate the construction, consider the case of a two-input two-outputs system. To achieve decoupling and good quality responses, the setpoint filter is sought in the form [15]

$$F(s) = \begin{bmatrix} \dfrac{1}{\tau_1 s+1} & \dfrac{a_1 s}{b_1 s+1} \\ \dfrac{a_2 s}{b_2 s+1} & \dfrac{1}{\tau_2 s+1} \end{bmatrix} \tag{16}$$

Setting

$$\kappa_1 = \frac{1}{\tau_1}, \quad \kappa_2 = \frac{1}{b_1}, \quad \kappa_3 = \frac{a_1}{b_1}$$
$$\kappa_4 = \frac{1}{\tau_2}, \quad \kappa_5 = \frac{1}{b_2}, \quad \kappa_6 = \frac{a_2}{b_2}$$

a state-space representation of the filter is obtained as

$$\mathscr{F}(\kappa) := \left[\begin{array}{c|c} A_F & B_F \\ \hline C_F & D_F \end{array}\right] = \left[\begin{array}{cccc|cc} -\kappa_1 & 0 & 0 & 0 & \kappa_1 & 0 \\ 0 & -\kappa_2 & 0 & 0 & 0 & -\kappa_3 \\ 0 & 0 & -\kappa_4 & 0 & 0 & \kappa_4 \\ 0 & 0 & 0 & -\kappa_5 & -\kappa_6 & 0 \\ \hline 1 & \kappa_2 & 0 & 0 & 0 & \kappa_3 \\ 0 & 0 & 1 & \kappa_5 & \kappa_6 & 0 \end{array}\right]$$

DOI: 10.1002/rnc

This means there exists a matrix U such that

$$\operatorname{vec} \mathscr{F}(\kappa) = \operatorname{vec} \mathscr{F}(0) + U\kappa, \quad \kappa \in \mathbb{R}^6$$

We immediately deduce the relevant subgradient formulas for program (15). With $v := ||\cdot||_\infty \circ T_{r\to e}(\cdot) \circ \mathscr{F}(\cdot)$, the subdifferential of v at κ, where $\mathscr{F}(\kappa)$ is a setpoint filter, is the compact and convex set of subgradients

$$\partial v(\kappa) = \{U^{\mathrm{T}} \operatorname{vec} \Phi_Y : \Phi_Y \in \partial(||\cdot||_\infty \circ T_{r\to e})(\mathscr{F}(\kappa))\} \tag{17}$$

The remaining expression for the subdifferential is directly obtained from Theorem 2.1.

2.5. *Non-smooth descent method*

In this section we briefly present our non-smooth optimization technique for composite functions of the H_∞-norm. For a more detailed discussion we refer the reader to [23, 24]. We start by representing the composite functions $f = ||\cdot||_\infty \circ T_{w\to z}$ or more generally $g = ||\cdot||_\infty \circ T_{w\to z} \circ \mathscr{K}(\cdot)$ under the form

$$g(\kappa) = \max_{\omega\in[0,+\infty]} g(\kappa, \omega)$$

where each $g(\kappa, \omega)$ is a composite maximum singular value function

$$g(\kappa, \omega) = \bar{\sigma}(\mathscr{G}(\kappa, \mathrm{j}\omega))$$

Here $\mathscr{G}(\kappa, \mathrm{j}\omega) = T_{w\to z}(\mathscr{K}(\kappa), \mathrm{j}\omega)$. At a given parameter κ, we can compute the set $\Omega(\kappa) := \Omega(\mathscr{K}(\kappa))$ of active frequencies, which is either finite, or coincides with $[0, +\infty]$ in those rare cases where the closed-loop system is all-pass. Excluding this case, we assume $\Omega(\kappa)$ finite and construct a finite extension $\Omega_e(\kappa)$ by adding frequencies according to the strategy presented in [23, 24]. See Figure 3 for a typical choice.

Following the general trend of Polak [32], we now define the optimality function

$$\theta_e(\kappa) := \min_{h\in\mathbb{R}^q} \max_{\omega\in\Omega_e(\kappa)} \max_{Y_\omega \succcurlyeq 0, \operatorname{Tr}(Y_\omega)=1} -g(\kappa) + g(\kappa, \omega) + h^{\mathrm{T}}\phi_{Y_\omega} + \tfrac{1}{2}h^{\mathrm{T}}Qh \tag{18}$$

where for every fixed ω, ϕ_{Y_ω} is a subgradient of $g(\kappa, \omega)$ at κ obtained as

$$\phi_{Y_\omega} := \left[\operatorname{Tr}\left(\frac{\partial \mathscr{K}(\kappa)}{\partial \kappa_1}^{\mathrm{T}} \Phi_{Y_\omega}\right), \ldots, \operatorname{Tr}\left(\frac{\partial \mathscr{K}(\kappa)}{\partial \kappa_q}^{\mathrm{T}} \Phi_{Y_\omega}\right)\right]^{\mathrm{T}}$$

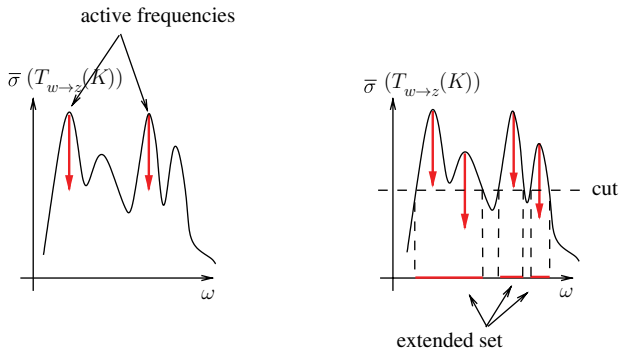

Figure 3. Selection of frequencies: (1) LHS active only and (2) RHS active and secondary peaks.

DOI: 10.1002/rnc

where

$$\Phi_{Y_\omega} = g(\kappa,\omega)^{-1}\Re G_{21}(\mathscr{K},\mathrm{j}\omega)\, T_{w\to z}(\mathscr{K},\mathrm{j}\omega)^H Q_\omega Y_\omega (Q_\omega)^H G_{12}(\mathscr{K},\mathrm{j}\omega), \quad Y_\omega \succcurlyeq 0, \quad \mathrm{Tr}\; Y_\omega = 1$$

The model of the objective function represented by θ_e is in principle of first order, but the quadratic term $h^{\mathrm{T}}Qh$ may in some cases be used to include second-order information. In [23, 24] we had worked with the basic choice $Q = \delta I \succ 0$, but we shall propose a more sophisticated choice here using BFGS updates.

Note that independently of the choices of $Q \succ 0$ and the finite extension $\Omega_e(\kappa)$ of $\Omega(\kappa)$ used, the optimality function has the following property: $\theta_e(\kappa) \leqslant 0$ and $\theta_e(\kappa) = 0$ if and only if $0 \in \partial g(\kappa)$, that is, κ is a critical point of g. In order to use θ_e to compute descent steps, it is convenient to obtain a dual representation of θ_e. To do this we use Fenchel duality to swap the max and min operators in (18). This means that we first replace the first inner supremum by a supremum over a convex hull which does not alter the value of θ_e. Then, after swapping max and min, the now inner infimum over $h \in \mathbb{R}^q$ becomes unconstrained and can be computed explicitly. Namely, for fixed Y_ω and τ_ω in the outer program, we obtain the solution of the form

$$h(Y,\tau) = -Q^{-1}\left(\sum_{\omega\in\Omega_e(\kappa)} \tau_\omega \phi_{Y_\omega}\right) \tag{19}$$

Substituting this back we obtain the dual expression

$$\theta_e(\kappa) = \max_{\tau_\omega \geqslant 0, \sum_{\omega\in\Omega_e(\kappa)} \tau_\omega = 1} \; \max_{Y_\omega \succcurlyeq 0, \mathrm{Tr}(Y_\omega)=1} \tau_\omega (g(\kappa,\omega) - g(\kappa))$$

$$-\frac{1}{2}\left(\sum_{\omega\in\Omega_e(\kappa)} \tau_\omega \phi_{Y_\omega}\right)^{\mathrm{T}} Q^{-1}\left(\sum_{\omega\in\Omega_e(\kappa)} \tau_\omega \phi_{Y_\omega}\right) \tag{20}$$

Note that in its dual form, computing $\theta_e(\kappa)$ leads to a semi-definite program. Indeed, substituting $Z_\omega = \tau_\omega Y_\omega$, program (20) becomes

$$\theta_e(\kappa) = \max_{Z_\omega \succcurlyeq 0, \sum_{\omega\in\Omega_e(\kappa)} \mathrm{Tr}(Z_\omega)=1} \mathrm{Tr}(Z_\omega)(g(\kappa,\omega) - g(\kappa)) - \frac{1}{2}\left(\sum_{\omega\in\Omega_e(\kappa)} \phi_{Z_\omega}\right)^{\mathrm{T}} Q^{-1}\left(\sum_{\omega\in\Omega_e(\kappa)} \phi_{Z_\omega}\right) \tag{21}$$

The latter program is converted to an LMI problem using a Schur complement argument. As a byproduct we see that $\theta_e(\kappa) \leqslant 0$ and that $\theta_e(\kappa) = 0$ implies κ is critical that is, $0 \in \partial g(\kappa)$.

What is important is that the direction $h(Y,\tau) = h(Z)$ in (19) is a descent direction of g at κ in the sense that the directional derivative satisfies the decrease condition

$$g'(\kappa; h(Z)) \leqslant \theta_e(\kappa) - \frac{1}{2}\left(\sum_{\omega\in\Omega_e(\kappa)} \phi_{Z_\omega}\right) Q^{-1}\left(\sum_{\omega\in\Omega_e(\kappa)} \phi_{Z_\omega}\right)$$

DOI: 10.1002/rnc

where Z is the dual optimal solution. See [24, Lemma 4.3] for a proof. In conclusion, we obtain the following algorithmic scheme:

Non-smooth descent method for $\min_\kappa g(\kappa)$

Parameters $0<\alpha<1$, $0<\beta<1$, $0<\delta\ll 1$

1. **Initialization**. Find a structured closed-loop stabilizing controller $\mathscr{K}(\kappa)$.
2. **Active frequencies**. Compute $g(\kappa)$ using the algorithm of [33] in its refined version [28] and obtain set of active frequencies $\Omega(\kappa)$.
3. **Add frequencies**. Build finite extension $\Omega_e(\kappa)$ of $\Omega(\kappa)$ as proposed in [23, 24], and choose $Q\succcurlyeq\delta I$.
4. **Step computation**. Compute $\theta_e(\kappa)$ by the dual SDP (20) and thereby obtain direction $h(Z)$ in (19). If $\theta_e(\kappa)=0$ stop. Otherwise:
5. **Line search**. Find largest $t=\beta^k$ such that $g(\kappa+th(Z))<g(\kappa)-\alpha t\theta_e(\kappa)$ and such that $\mathscr{K}(\kappa+th(Z))$ remains stabilizing.
6. **Step**. Replace κ by $\kappa+th(Z)$ and go back to step 2.

Note that the line search in step 5 is successful because $t^{-1}(g(\kappa+th(Z))-g(\kappa))\to g'(\kappa;h(Z))$ as $t\to 0^+$, and because $\theta_e(\kappa)<0$ and $0<\alpha<1$. Choosing t under the form $t=\beta^k$ with the largest possible k comes down to doing a backtracking line search, which safeguards against taking too small steps.

The final elements to be provided is computation of the matrix Q^{-1}. Since Q is supposed to carry second-order information on the objective function, it may seem appropriate to do a BFGS update

$$Q^+=Q+\frac{yy^{\mathrm{T}}}{y^{\mathrm{T}}s}-\frac{Qss^{\mathrm{T}}Q}{s^{\mathrm{T}}Qs}$$

where $s=\kappa^+-\kappa$ and $y=\phi^+-\phi$, where ϕ is the subgradient of minimal norm in $\partial g(\kappa)$, g^+ the subgradient of minimal norm in $\partial g(\kappa^+)$. Here, the notation x^+ and x stands for the current and past iterates, respectively. Since the inverse Q^{-1} is required, an alternative is to use the inverse BFGS update. Here we maintain the matrix $P\approx Q^{-1}$ in step 4 of the algorithm through

$$P^+=P+\frac{(s-Py)s^{\mathrm{T}}+s(s-Py)^{\mathrm{T}}}{s^{\mathrm{T}}y}-\frac{y^{\mathrm{T}}(s-Py)ss^{\mathrm{T}}}{(s^{\mathrm{T}}y)^2}$$

As is well-known, $P\succ 0$ is guaranteed as long as $s^{\mathrm{T}}y>0$. If this fails, or if $P\not\succ 0$ for numerical reasons, we modify P^+ or restart the procedure.

Note that computing minimal norm elements $\phi\in\partial g(\kappa)$ amounts to computing minimal norm elements in the LMI set (9) and therefore reduces to an LMI problem. Finally, we emphasize the important fact that when singular values $\bar\sigma(\mathscr{G}(\kappa,\mathrm{j}\omega))$ are simple on $\Omega_e(\kappa)$, which is the rule in practice, we have $Z_\omega=\operatorname{Tr} Z_\omega$ so that SDP (21) simplifies to a much faster convex QP. A fact that can also be exploited for computing minimal norm elements.

3. NON-SMOOTH LOOP-SHAPING DESIGN

In this section we recall some key facts from the loop-shaping design technique introduced by McFarlane and Glover [26], and we discuss how it can be merged with our non-smooth framework to arrive at a new practical PID loop-shaping design method.

DOI: 10.1002/rnc

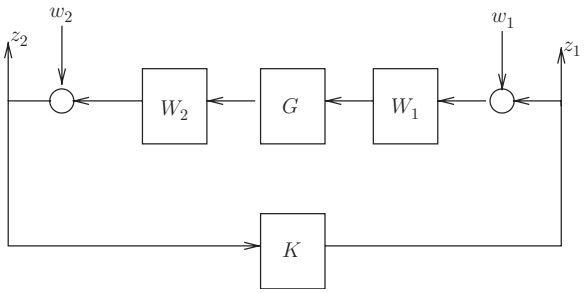

Figure 4. Loop-shaping H_∞ design.

3.1. Loop-shaping design

Loop-shaping design is an efficient and practical technique which has been applied successfully to a variety of difficult design problems, see [34–36] to cite a few. It proceeds as presented in Figure 4. Firstly, the open-loop plant G is altered by pre- and post-compensators W_1 and W_2, respectively, to achieve the desired open-loop shapes. Roughly speaking, the shaped plant W_2GW_1 should have large gains at low frequencies for performance and small gains at high frequencies for noise attenuation. Also, the roll-off rate should not be too large in the mid-frequency range. Secondly, an H_∞ synthesis is performed to minimize the objective

$$||T_{(w_1,w_2)\to(z_1,z_2)}(K)||_\infty := \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - W_2GW_1)^{-1} [G \quad I] \right\|_\infty \tag{22}$$

which yields a controller K. The final controller is then implemented in the form W_1KW_2 and has no specific structure. Note that the H_∞ norm in (22) is greater than one for the optimal controller K. Put differently, $\varepsilon := 1/\gamma$ is bounded above by unity. The scalar ε is therefore an indicator of success of the procedure. Usually values greater than 0.2 or 0.3 are deemed acceptable in the sense that the controller K does not significantly alter the desired loop shape W_2GW_1 [26]. Moreover, in this situation, the closed-loop system will be robust against coprime factor uncertainties [26]. A smaller ε, on the contrary, tells us that the desired loop shape is incompatible with robustness requirements, and the pre- and post-compensators must be modified to reflect less ambitious performance requirements. A central element of the design technique is therefore to select appropriate compensators for the trade-off between performance and robustness. In practice, this is generally accomplished by trials-and-errors.

3.2. PID loop-shaping design

For loop-shaping design with a PID controller we have adopted the strategy introduced in [13]. In this approach, the controller K is sought in the form $K = W_1^{-1}K_{\mathrm{PID}}$, assuming that the pre-compensator W_1 is invertible and K_{PID} is a PID controller. The synthesis problem is then of the form

$$\underset{K_{\mathrm{PID}}\ \text{stabilizing}}{\text{minimize}} \left\| \begin{bmatrix} W_1^{-1}K_{\mathrm{PID}} \\ I \end{bmatrix} (I - W_2GK_{\mathrm{PID}})^{-1} [W_2GW_1 \quad I] \right\|_\infty \tag{23}$$

DOI: 10.1002/rnc

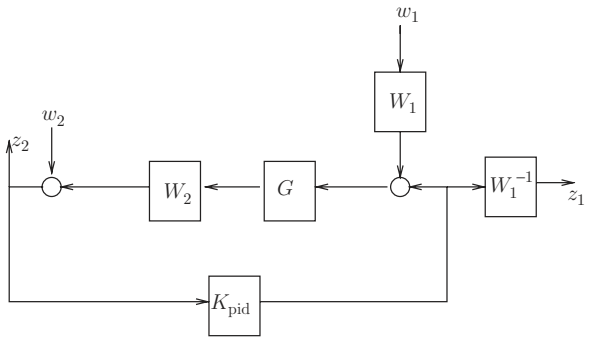

Figure 5. PID loop-shaping H_∞ design.

Note that this scheme retains all the benefits of the H_∞ loop-shaping design technique of Section 3.1. As before $\varepsilon := 1/\gamma$, where γ is the optimal H_∞ cost in (23), will serve to diagnose whether performance and robustness requirements have been achieved. The final controller is obtained as the series connection of the post-compensator W_2 and the PID controller in the form $K_{\text{PID}} W_2$. An immediate consequence is that the final controller has better noise attenuation in the high-frequency range than a pure PID controller, whenever W_2 is strictly proper. To sum up, we have to solve program (23), see also Figure 5, for a PID controller structure and the non-smooth technique and tools discussed in Sections 2 will be used to that purpose.

4. APPLICATION TO A SEPARATING TOWER

In this section, we consider the application of the non-smooth technique to the control design for a chemical process from the literature [13]. It consists of a 24-tray tower for separating methanol and water. The transfer function for controlling the temperature on the fourth and 17th trays is given as

$$\begin{bmatrix} t_{17} \\ t_4 \end{bmatrix} = \begin{bmatrix} \dfrac{-2.2e^{-s}}{7s+1} & \dfrac{1.3e^{-0.3s}}{7s+1} \\ \dfrac{-2.8e^{-1.8s}}{9.5s+1} & \dfrac{4.3e^{-0.35s}}{9.2s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{24}$$

Settling times of about 10 s are desired for the closed-loop process in response to step inputs, as well as good decoupling between the temperatures t_{17} and t_4. A good robustness margin is also required to account for process model uncertainties. The latter will be assessed using the coprime factor uncertainty margin ε as defined in Section 3. The actual plant in (24) is approximated by a rational model using second-order Pade approximation of the delays. This leads to a 12th-order model. Pre- and post-compensators are taken from [13]

$$W_1(s) = \begin{bmatrix} \dfrac{5s+2}{s+0.001} & 0 \\ 0 & \dfrac{5s+2}{s+0.001} \end{bmatrix}, \quad W_2(s) = \begin{bmatrix} \dfrac{10}{s+10} & 0 \\ 0 & \dfrac{10}{s+10} \end{bmatrix}$$

DOI: 10.1002/rnc

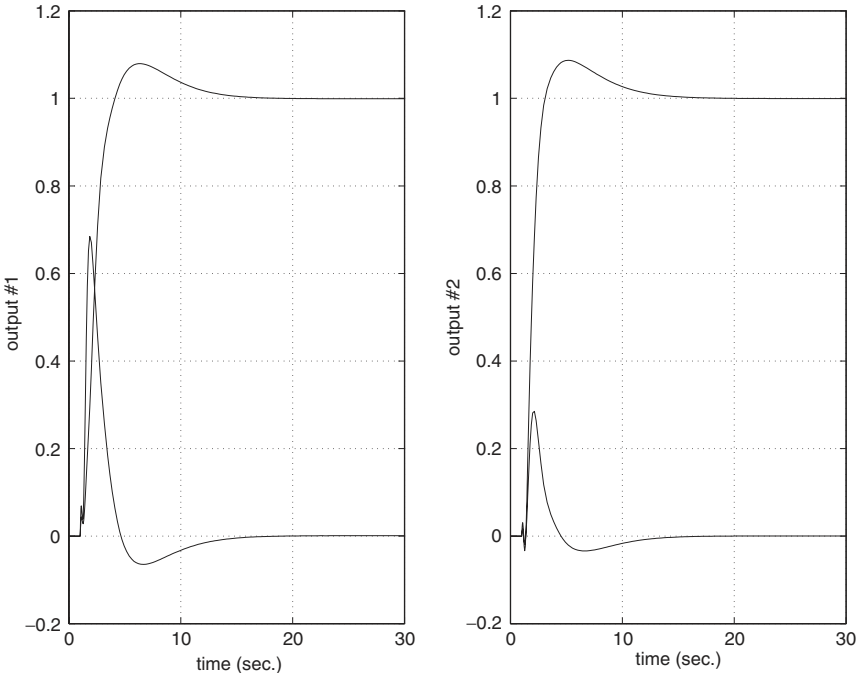

Figure 6. Time-domain simulations of H_∞ controller.

The standard form of Figure 5 incorporating the compensators is therefore of 18th order. For comparison, we have synthesized an unstructured full-order controller, whose associated step responses and singular-value frequency responses are displayed in Figures 6 and 7, respectively. For full order the robustness was found to be $\varepsilon := 1/\gamma = 0.361$. This is very satisfactory in terms of stability and gives good agreement with the prescribed open-loop shapes. This is confirmed by the fast and well-damped step responses in Figure 6. We also observe short-term couplings between t_{17} and t_4, an unpleasant behaviour which we try to reduce later when dealing with PID controllers. For future use, we keep in mind that the optimal $\gamma = 2.77$ achieved at this stage is globally optimal if controller structural constraints are disregarded.

In Reference [13], a state-space BMI formulation is used to characterize PID solutions of the H_∞ synthesis problem in (23). The proposed algorithmic strategy is a coordinate descent scheme, which alternates between minimizing over Lyapunov variables and over PID controller parameters. Such techniques are often referred to as D–K or V–K iterations, see [8] and references therein. Unfortunately, coordinate descent does not come with any convergence guarantee [37] and breakdown is often observed in practice. In our case, Reference [13] reported 38 min of cputime to obtain the following solution with such a technique:

$$\epsilon = 0.060, \quad K_{\mathrm{p}} = \begin{bmatrix} 2.4719 & -1.2098 \\ -1.1667 & -2.4766 \end{bmatrix}$$

DOI: 10.1002/rnc

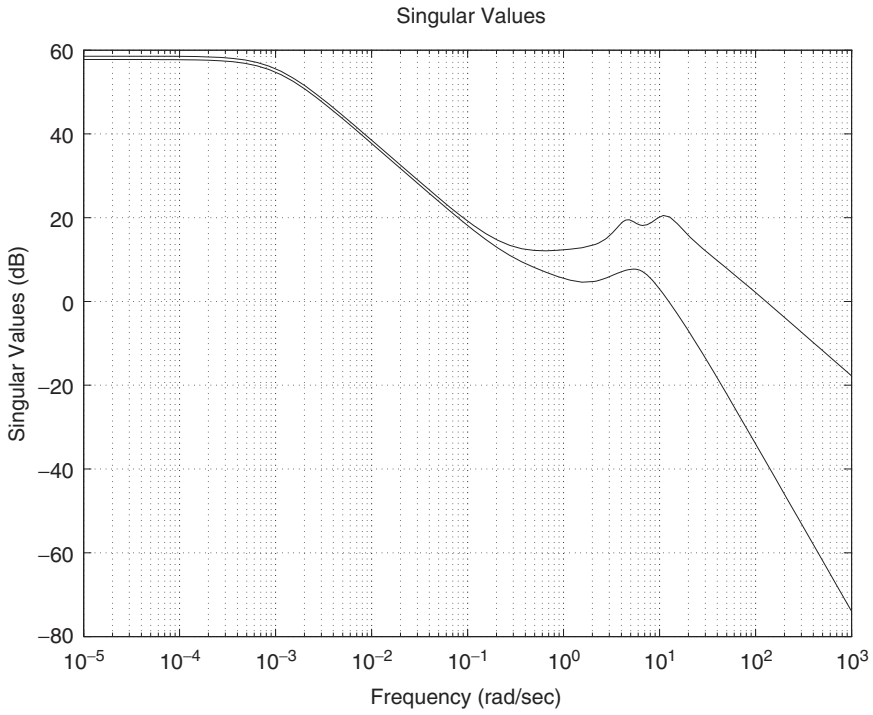

Figure 7. Frequency response of H_∞ controller.

$$K_i = \begin{bmatrix} 0.4657 & -0.31 \\ -0.2329 & -0.487 \end{bmatrix}, \quad K_d = \begin{bmatrix} 0.0534 & -0.0072 \\ -0.015 & -0.0434 \end{bmatrix}$$

The corresponding robustness margin is $\varepsilon = \frac{1}{4.02} = 0.249$. Time domain and frequency responses are shown in Figures 8 and 9, respectively. Using the tools from Section 2.5, we now show that this PID controller is not a local minimum of program (23). Our local optimality certificate θ evaluated at the PID controller above takes a negative value $\theta = -0.119$, revealing failure of the D–K iteration scheme to reach local optimality. This also indicates that further progress can be achieved by running our non-smooth method initialized at the point of failure. Ultimately, the following PID controller is obtained:

$$\epsilon = 0.1527, \quad K_p = \begin{bmatrix} 2.6047 & -0.6543 \\ -1.1253 & -2.3226 \end{bmatrix}$$

$$K_i = \begin{bmatrix} 0.8527 & -0.2591 \\ 0.0701 & -0.9362 \end{bmatrix}, \quad K_d = \begin{bmatrix} 0.7414 & -0.2551 \\ -1.5610 & -0.0331 \end{bmatrix}$$

This represents 38% improvement of the robustness margin over the value $\varepsilon = \frac{1}{2.91} = 0.343$. Better time responses are also observed, see Figure 10. The optimal PID controller also exhibits

DOI: 10.1002/rnc

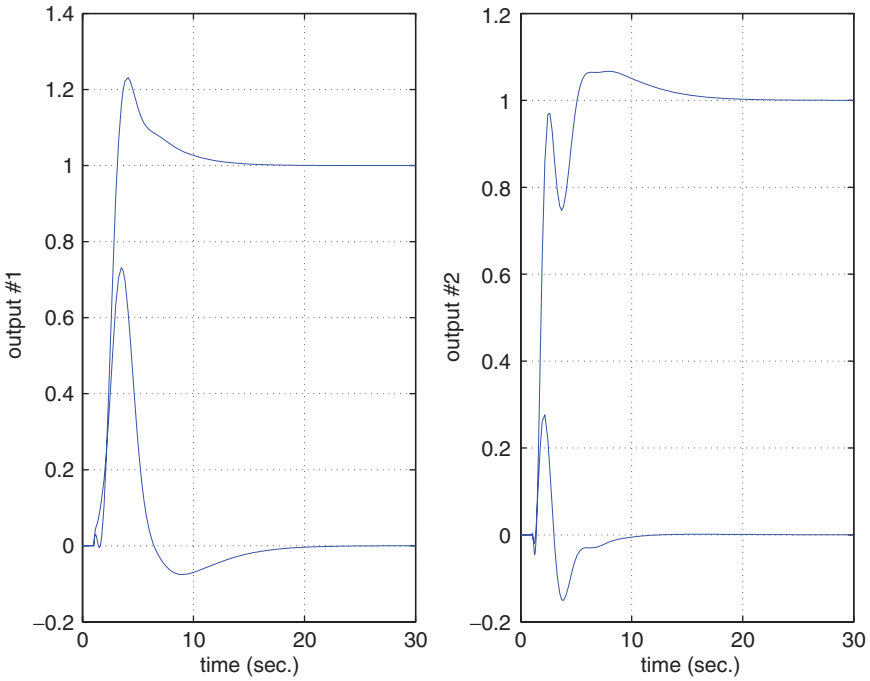

Figure 8. Time-domain simulations of PID controller in [13].

higher gains in the low and medium frequency ranges, Figure 11. On the other hand, it can be seen in Figure 10 that step responses of t_{17} and t_4 are strongly coupled, which is undesirable in this application. To reduce the coupling we added a setpoint filter using the model reference approach discussed in Section 2.4, with the filter structure in (16). The reference model was selected as

$$G_{\text{ref}}(s) := \begin{bmatrix} \dfrac{1}{s+1} & 0 \\ 0 & \dfrac{1}{s+1} \end{bmatrix}$$

Solving program (15) using our non-smooth technique produced the following setpoint filter:

$$F(s) = \begin{bmatrix} \dfrac{1.045}{s+1.045} & \dfrac{-0.3428s}{s+2.25} \\ \dfrac{-0.3666s}{s+0.6147} & \dfrac{3.675}{s+3.675} \end{bmatrix}$$

Step responses with PID controller in tandem with the setpoint filter are presented in Figure 12.

In a second experiment, we do not use the result in [13] and initialize the algorithm from scratch, by first computing a stabilizing PID controller, using an extension of our method to minimize the spectral abscissa [38]. Note that with a different initial seed, there is no

DOI: 10.1002/rnc

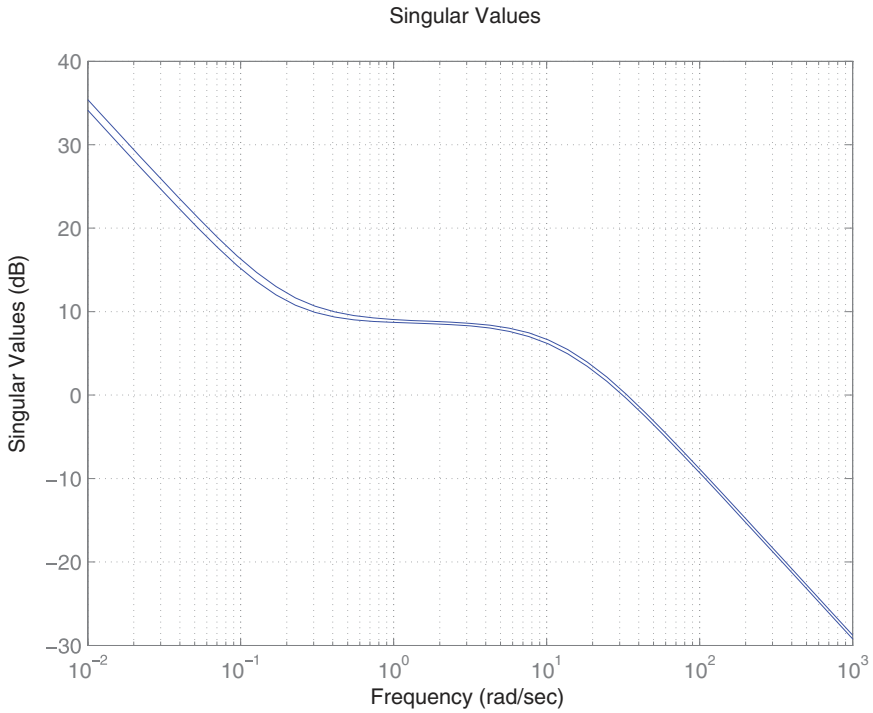

Figure 9. Frequency response of PID controller in [13].

reason why we would reach the same local solution. And indeed, a new local minimum is reached

$$\epsilon = 0.3780, \quad K_{\mathrm{p}} = \begin{bmatrix} 2.4798 & -0.5153 \\ -0.8850 & -2.1737 \end{bmatrix}$$

$$K_{\mathrm{i}} = \begin{bmatrix} 0.7428 & -0.1873 \\ -0.0455 & -0.7698 \end{bmatrix}, \quad K_{\mathrm{d}} = \begin{bmatrix} 0.2640 & -0.2347 \\ -2.1424 & 0.0723 \end{bmatrix}$$

The robustness margin is now $\varepsilon = \frac{1}{3.05} = 0.3279$, which is only marginally worse than the previously synthesized PID controller and again improves the solution given in [13] by 24%. Time- and frequency-domain evaluations are given in Figures 13 and 14. As before, a setpoint filter was computed using the reference model

$$G_{\mathrm{ref}}(s) := \begin{bmatrix} \dfrac{0.25}{s^2 + 0.4s + 0.25} & 0 \\ 0 & \dfrac{1}{s+1} \end{bmatrix}$$

DOI: 10.1002/rnc

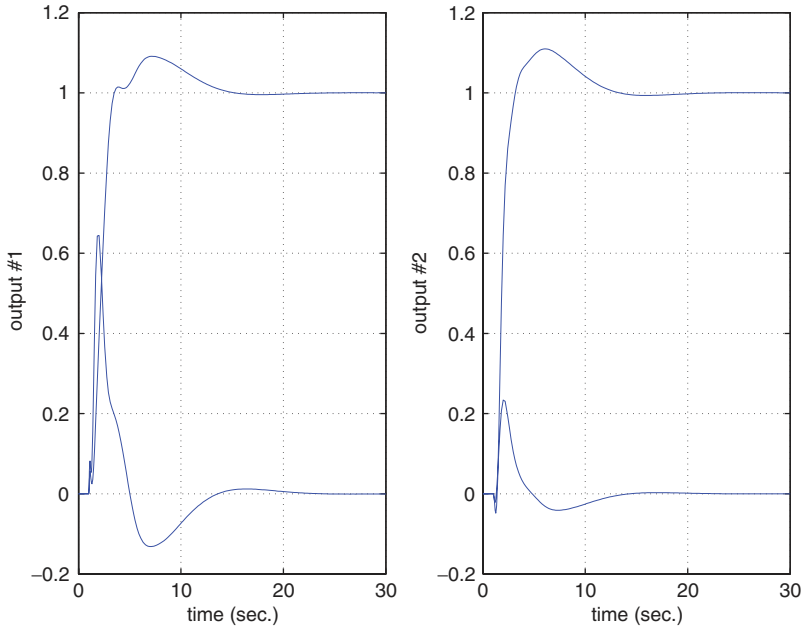

Figure 10. Time-domain simulations of first PID controller.

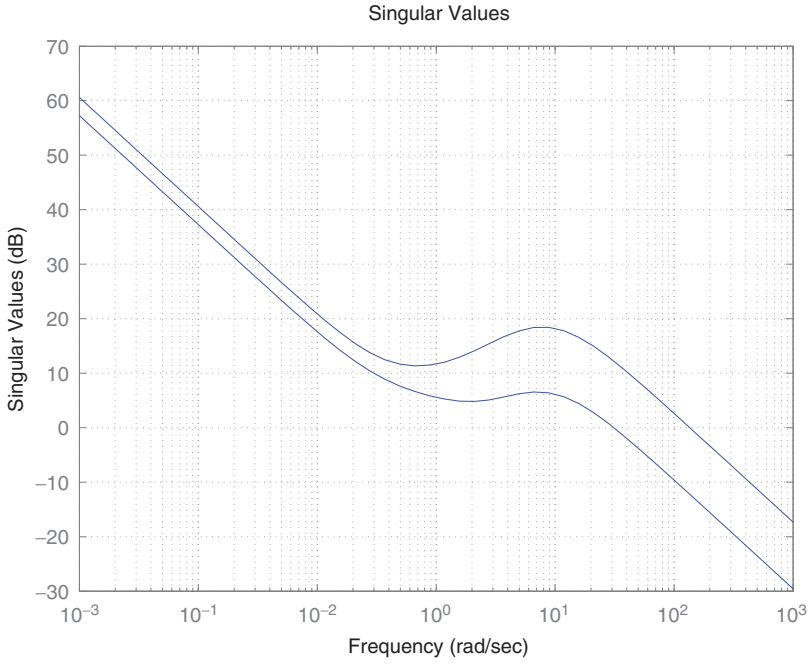

Figure 11. Frequency response of first PID controller.

DOI: 10.1002/rnc

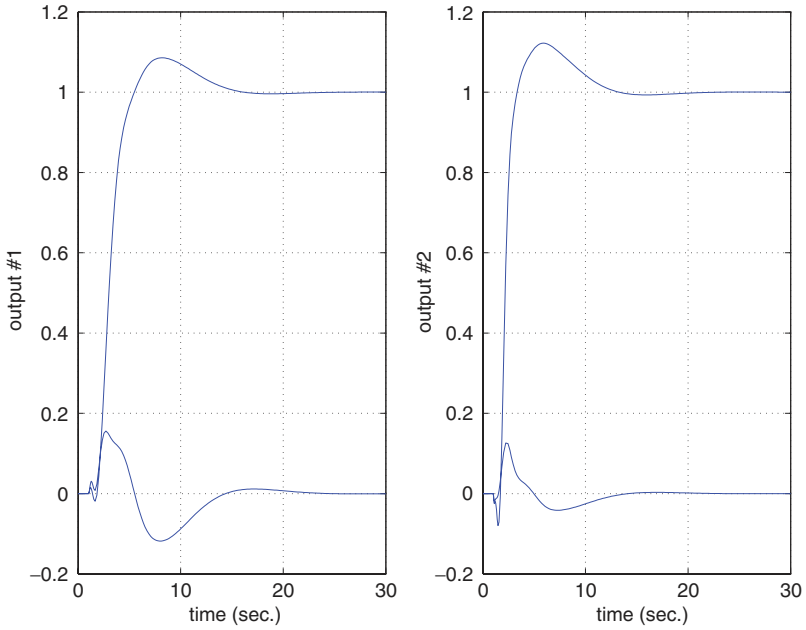

Figure 12. Time-domain simulations of first PID controller with setpoint filter.

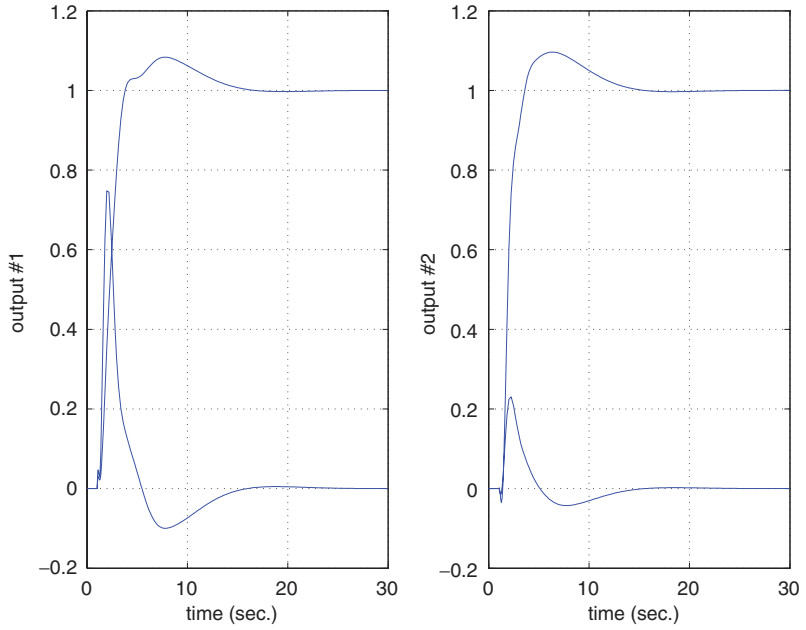

Figure 13. Time-domain simulations of second PID controller.

DOI: 10.1002/rnc

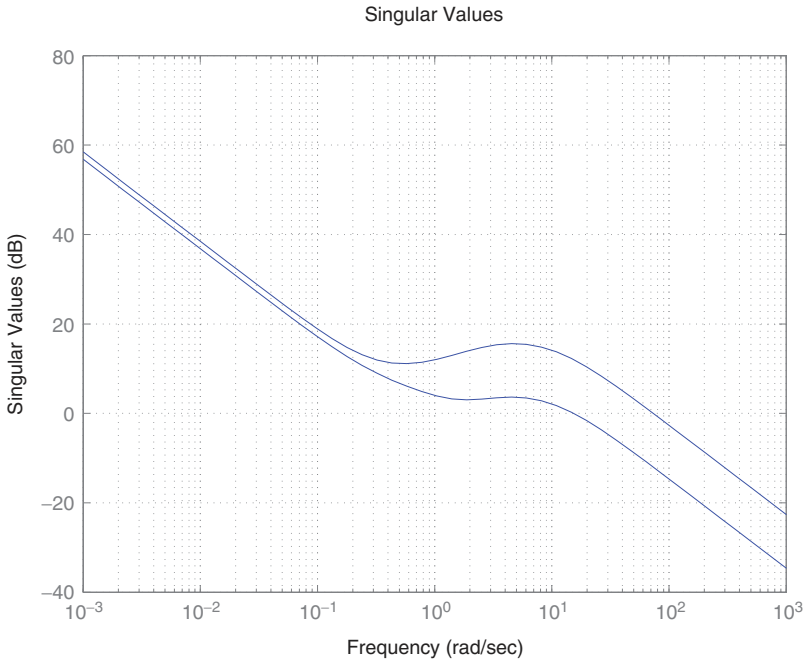

Figure 14. Frequency response of second PID controller.

The setpoint filter is described as

$$F(s) = \begin{bmatrix} \dfrac{0.7333}{s+0.7333} & \dfrac{35.81s}{s+297.7} \\ \dfrac{-0.5452s}{s+0.31} & \dfrac{215.1}{s+215.1} \end{bmatrix}$$

and step responses are given in Figure 15.

As one might expect for a local optimization technique, our experiment underlines that different local solutions are generally obtained when different initial guesses are used. Such guesses may be obtained by heuristic techniques developed in PID synthesis. If many local optima are observed, using semi-global techniques may be indicated. Such methods try to improve on a given set of local solutions [39].

A strong point of our method is that it practically always finds local optimal solutions, as theoretically expected. Also, the running times are very fast. In fact, in our test we never exceed a minute cputime. This is in strong contrast with state-space sequential LMI or BMI approaches, which require substantial numerical efforts. In those cases where a local minimum is not satisfactory, we have to do restart with different initial seeds, to explore whether further improvements is possible. How this can be organized in general will be investigated in the future.

DOI: 10.1002/rnc

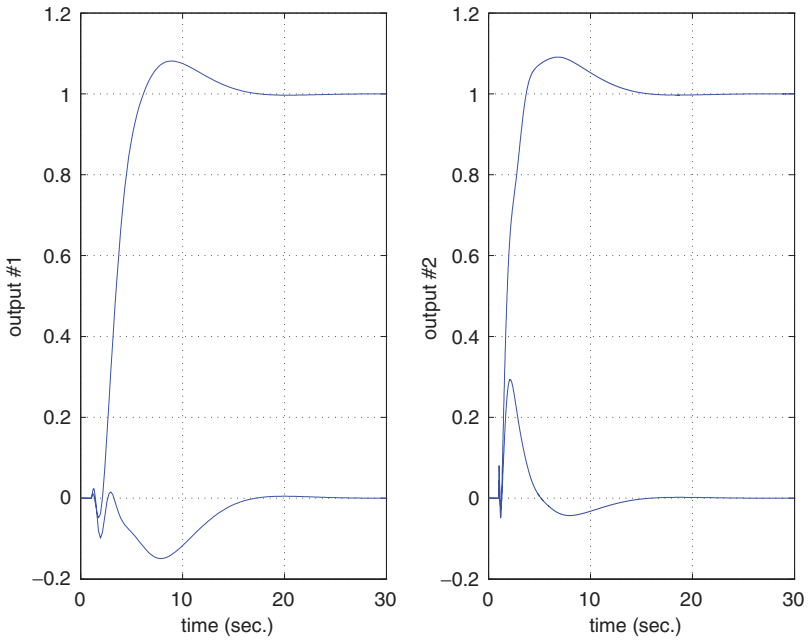

Figure 15. Time-domain simulations of second PID controller with setpoint filter.

5. CONCLUSION

We have presented and discussed a non-smooth optimization technique for the synthesis of finely structured controllers with an H_∞ objective. Our approach is general and encompasses most controller structures of practical interest. The proposed technique is endowed by an easily tractable convergence certificate, which may either serve to validate a given controller, or to drive the iterative descent method to termination. A specialization to loop-shaping design with MIMO PID controllers is also introduced. Application to a process system indicates that the technique is a practical and effective numerical tool in structured controller design.

NOTATION

Let $\mathbb{R}^{n\times m}$ be the space of $n \times m$ matrices, equipped with the corresponding scalar product $\langle X, Y\rangle = \mathrm{Tr}(X^{\mathrm{T}} Y)$, where X^{T} is the transpose of the matrix X, $\mathrm{Tr}(X)$ its trace. For complex matrices X^H stands for its conjugate transpose. For Hermitian or symmetric matrices, $X \succ Y$ means that $X - Y$ is positive definite, $X \succcurlyeq Y$ that $X - Y$ is positive semi-definite. We use the symbol λ_1 to denote the maximum eigenvalue of a symmetric or Hermitian matrix. Given an operator T, T^* is used to denote its adjoint operator on the appropriate space. The notation vec applied to a matrix stands for the usual column-wise vectorization of a matrix. The operator diag applied to a matrix produces a vector whose entries are the diagonal elements of the matrix. We use concepts from non-smooth analysis covered by Clarke [30]. For a locally Lipschitz function f: $\mathbb{R}^n \to \mathbb{R}$, $\partial f(x)$ denotes its Clarke subdifferential at x.

DOI: 10.1002/rnc

ACKNOWLEDGEMENTS

The authors acknowledge financial support from the Fondation d'entreprise EADS (under contract *Solving challenging problems...*) and from the Agence Nationale de Recherche (ANR) (project NT05 *Guidage*).

REFERENCES

1. Doyle JC, Glover K, Khargonekar P, Francis BA. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control* 1989; **AC-34**(8):831–847.
2. Tsitsiklis JN, Blondel V. NP-hardness of some linear control design problems. *SIAM Journal on Control* 1997; **35**(6):2118–2127.
3. Balakrishnan V, Boyd S. Global optimization in control system analysis and design. In *Control and Dynamic Systems*: *Advances in Theory and Applications*, Leondes CT (ed.), vol. 53. Academic Press: New York, 1992.
4. Wong T, Bigras P. Structured controller design with evolutionary optimization. *AIA2003*, Benalmadena, Spain, September 2003; 163–168.
5. Boyd S, ElGhaoui L, Feron E, Balakrishnan V. *Linear Matrix Inequalities in Systems and Control Theory*. SIAM Studies in Applied Mathematics, vol. 15. SIAM: Philadelphia, 1994.
6. Grigoriadis KM, Skelton RE. Low-order control design for LMI problems using alternating projection methods. *Automatica* 1996; **32**(8):1117–1125.
7. Ebihara Y, Tokuyama K, Hagiwara T. Structured controller synthesis using LMI and alternating projection method. *International Journal of Control* 2004; **77**(12):1137–1147.
8. El Ghaoui L, Balakrishnan V. Synthesis of fixed-structure controllers via numerical optimization. *Proceedings of IEEE Conference on Decision and Control*, Lake Buena Vista, FL, 1994; 2678–2683.
9. Iwasaki T. The dual iteration for fixed-order control. *IEEE Transactions on Automatic Control* 1999; **44**(4):783–788.
10. Han J, Skelton RE. An LMI optimization approach for structured linear controllers. *Proceedings of IEEE Conference on Decision and Control*, Maui, HI, 2003; 5143–5148.
11. Hassibi A, How J, Boyd S. A pathfollowing method for solving BMI problems in control. *Proceedings of American Control Conference*, San Diego, U.S.A., 1999; 1385–1389.
12. Bao J, Forbes JF, McLellan PJ. Robust multiloop PID controller design: a successive semidefinite programming approach. *Industrial and Engineering Chemistry Research* 1999; **38**(9):3407–3413.
13. Genc AU. A state-space algorithm for designing H_∞ loop shaping PID controllers. *Technical Report*, Cambridge University, Cambridge, U.K., October 2000.
14. Miyamoto S, Vinnicombe G. Robust decentralized control design based on coprime factorization representations and LMI optimization. *Proceedings of the 36th SICE Annual Conference*, Tokushima, Japan, 1997; 1007–1012.
15. Tan W, Chen T, Marquez HJ. Robust controller design and PID tuning for multivariable processes. *Asian Journal of Control* 2002; **4**(4):439–451.
16. Saeki M. Fixed structure PID controller design for standard H_∞ control problem. *Automatica* 2006; **42**(1):93–100.
17. Rotkowitz M, Lall S. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control* 2006; **51**(2):274–286.
18. Xin Q, Salapaka MV, Voulgaris PG, Khammash M. Structured optimal and robust control with multiple criteria: a convex solution. *IEEE Transactions on Automatic Control* 2004; **49**(10):1623–1640.
19. Boyd S, Barratt C. *Linear Controller Design*: *Limits of Performance*. Prentice-Hall: Englewood Cliffs, NJ, 1991.
20. Scherer C. Structured finite-dimensional controller design by convex optimization. *Linear Algebra and Applications* 2002; **351–352C**:639–669.
21. Mäkilä PM, Toivonen HT. Computational methods for parametric LQ minimization: a survey. *IEEE Transactions on Automatic Control* 1987; **AC-32**(8):658–671.
22. Polak E, Wardi Y. A nondifferentiable optimization algorithm for the design of control systems subject to singular value inequalities over a frequency range. *Automatica* 1982; **18**(3):267–283.
23. Apkarian P, Noll D. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control* 2006; **51**(1):71–86.
24. Apkarian P, Noll D. Nonsmooth optimization for multidisk H_∞ synthesis. *European Journal of Control* 2006; **12**(3):229–244.
25. Apkarian P, Noll D. Nonsmooth optimization for multiband frequency domain control design. *Automatica* 2006, to appear.
26. McFarlane D, Glover K. A loop shaping design procedure using H_∞ synthesis. *IEEE Transactions on Automatic Control* 1992; **37**(6):759–769.
27. Noll D, Apkarian P. Spectral bundle methods for nonconvex maximum eigenvalue functions: first-order methods. *Mathematical Programming Series B* 2005; **104**(2):701–727.

DOI: 10.1002/rnc

28. Apkarian P, Noll D. IQC analysis and synthesis via nonsmooth optimization. *Systems and Control Letters* 2006; **55**(12):971–981.
29. Bompart V, Noll D, Apkarian P. Second-order nonsmooth optimization for H_∞ synthesis. *5th IFAC Symposium on Robust Control Design*, Toulouse, France, July 2006.
30. Clarke FH. *Optimization and Nonsmooth Analysis*. Canadian Mathematical Society Series. Wiley: New York, 1983.
31. Polak E, Salcudean S. On the design of linear multivariable feedback systems via constrained nondifferentiable optimization in H_∞ spaces. *IEEE Transactions on Automatic Control* 1989; **AC-34**(3):268–276.
32. Polak E. *Optimization*: *Algorithms and Consistent Approximations*. Applied Mathematical Sciences. Springer: New York, 1997.
33. Boyd S, Balakrishnan V. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $\mathbf{L}_\infty$-norm. *Systems and Control Letters* 1990; **15**:1–7.
34. Cheang SU, Chen WJ. Stabilizing control of an inverted pendulum system based on H_∞ loop shaping design procedure. *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, Hefei, China, vol. 5, June 2000; 3385–3388.
35. Fujita M, Hatake K, Matsumura F. Loop shaping based robust control of a magnetic bearing. *IEEE Control Systems Magazine* 1993; **13**(4):57–65.
36. Raisch JJ, Lang L, Groebel M. Loop shaping controller design for a binary distillation column. *Control'91*, Edinburgh, U.K., vol. 2, March 1991; 1271–1276.
37. Helton JW, Merino O. Coordinate optimization for bi-convex matrix inequalitites. *Proceedings of IEEE Conference on Decision and Control*, San Diego, CA, 1997; 3609–3613.
38. Bompart V, Apkarian P, Noll D. Nonsmooth technique for stabilizing linear systems. 2006, submitted for publication.
39. Suykens J, Vandewalle K, De Moor B. Intelligence and cooperative search by coupled local minimizers. *International Journal of Bifurcation and Chaos* 2001; **11**(8):2133–2144.

DOI: 10.1002/rnc