Nonsmooth optimization techniques for structured controller design

Pierre Apkarian[1], **Vincent Bompart**[2], **and Dominikus Noll**[3]

[1] ONERA and Institut de Mathématiques Université Paul Sabatier,
2 av. Edouard Belin, 31055 Toulouse, France
apkarian@cert.fr

[2] ONERA
2 av. Edouard Belin, 31055 Toulouse, France
bompart@cert.fr

[3] Institut de Mathématiques Université Paul Sabatier,
118, route de Narbonne, 31062 Toulouse, France
noll@mip.ups-tlse.fr

Abstract: Significant progress in control design has been achieved by the use of nonsmooth and semi-infinite mathematical programming techniques. In contrast with LMI or BMI approaches, these new methods avoid the use of Lyapunov variables, which gives them two major strategic advances over matrix inequality methods. Due to the much smaller number of decision variables, they do not suffer from size restrictions, and they are much easier to adapt to structural constraints on the controller. In this paper, we further develop this line and address both frequency- and time-domain design specifications by means of a nonsmooth algorithm general enough to handle both cases.

1 Introduction

Interesting new methods in nonsmooth optimization for the synthesis of controllers have recently been proposed. See [11, 8] for stabilization problems, [4, 5, 20, 3, 10] for H_∞ synthesis, and [3, 6] for design with IQCs. These techniques are in our opinion a valuable addition to the designer's toolkit:

- They avoid expensive state-space characterizations, which suffer the curse of dimension, because the number of Lyapunov variables grows quadratically with the system size.
- The preponderant computational load of these new methods is transferred to the frequency domain and consist mainly in the computation of spectra and eigenspaces, and of frequency domain quantities, for which efficient algorithms exist. This key feature is the result of the idea of the diligent use of nonsmooth criteria of the form $f(K) = \max_{\omega\in[0,\infty]} \lambda_1 (F(K,\omega))$, which are composite functions of a smooth but nonlinear operator F, and a non-smooth but convex function λ_1.

- The new approach is highly flexible, as it allows to address, with almost no additional cost, structured synthesis problems of the form $f(\kappa) = \max_{\omega\in[0,\infty]} \lambda_1 \left(F(\mathcal{K}(\kappa), \omega)\right)$, where $\mathcal{K}(\cdot)$ defines a mapping from the space of controller parameters κ to the space of state-space representations K. From a practical viewpoint, structured controllers are better apprehended by designers and facilitate implementation and re-tuning whenever performance or stability specifications change. This may be the major advantage of the new approach over matrix inequality methods.
- The new approach is general and encompasses a wide range of problems beyond pure stabilization and H_∞ synthesis. A number of important problems in control theory can be regarded as structured control problems. Striking examples are simultaneous stabilization, reliable and decentralized control, multi frequency band design, multidisk synthesis, and much else.
- Finally, the new methods are supported by mathematical convergence theory, which certifies global convergence under practically useful hypotheses in the sense that iterates converge to critical points from arbitrary starting points.

In this paper, we expand on the nonsmooth technique previously introduced in [4], and explore its applicability to structured controller design in the presence of frequency- and time-domain specifications. We show that the same nonsmooth minimization technique can be used to handle these seemingly different specifications. We address implementation details of the proposed technique and highlight differences between frequency and time domain.

We refer the reader to the articles cited above for references on controller synthesis using nonsmooth optimization. General concepts in nonsmooth analysis can be found in [12], and optimization of max functions is covered by [23]. Time response shaping is addressed at length in [13, 15, 17]. These techniques are often referred to as the Iterative Feedback Tuning (IFT) approach, mainly developed by M. Gevers, H. Hjalmarsson and co-workers.

2 Time- and frequency domain designs

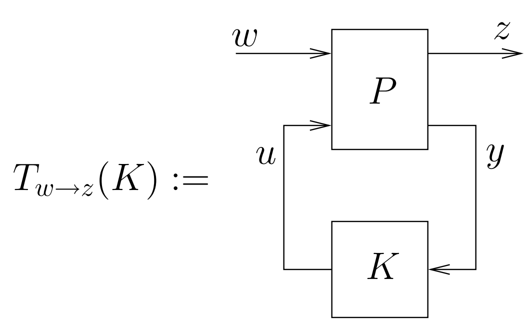

Figure 1: standard interconnection

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 or a test signal, $y \in \mathbb{R}^{p_2}$ 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 time- or frequency domain 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, $K(s)$ is such that either the closed-loop time response $z(t)$ to a test signal $w(t)$ satisfies prescribed constraints, or the H_∞ norm of transfer function $\|T_{w\to z}(K)\|_\infty$ is minimized. Here $T_{w\to z}(K)$ denotes the closed-loop transfer function from w to z, see figure 1.

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

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

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:

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

so that manipulations will involve a static matrix

$$\mathcal{K} := \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \in \mathbb{R}^{(k+m_2)\times(k+p_2)}. \tag{3}$$

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

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

Structural constraints on the controller will be defined by a matrix-valued mapping $\mathcal{K}(.)$ from $\mathbb{R}^q$ to $\mathbb{R}^{(k+m_2)\times(k+p_2)}$, that is $\mathcal{K} = \mathcal{K}(\kappa)$, where vector $\kappa \in \mathbb{R}^q$ denotes the independent variables in the controller parameter space $\mathbb{R}^q$. For the time being we will consider free variation $\kappa \in \mathbb{R}^q$, but the reader will be easily convinced that adding parameter restriction by means of mathematical programming constraints $g_I(\kappa) \leq 0, g_E(\kappa) = 0$ could be added if need be. We will assume throughout that the mapping $\mathcal{K}(.)$ is continuously differentiable, but otherwise arbitrary. As a typical example, consider MIMO PID controllers, given as

$$K(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{1 + \epsilon s}\,, \tag{5}$$

where K_p, K_i and K_d are the proportional, the integral and the derivative gains, respectively, are alternatively represented in the form

$$K(s) = D_K + \frac{R_i}{s} + \frac{R_d}{s + \tau}\,, \tag{6}$$

with the relations

$$D_K := K_p + \frac{K_d}{\epsilon}, \quad R_i := K_i, \quad R_d := -\frac{K_d}{\epsilon^2}, \quad \tau := \frac{1}{\epsilon}\,, \tag{7}$$

and a linearly parameterized state-space representation is readily derived as

$$\mathcal{K}(\kappa) = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right] = \left[\begin{array}{cc|c} 0 & 0 & R_i \\ 0 & -\tau I & R_d \\ \hline I & I & D_K \end{array}\right], \qquad A_K \in \mathbb{R}^{2m_2 \times 2m_2}\,. \tag{8}$$

Free parameters in this representation can be gathered in the vector κ obtained as

$$\kappa := \begin{bmatrix} \tau \\ \operatorname{vec} R_i \\ \operatorname{vec} R_d \\ \operatorname{vec} D_K \end{bmatrix} \in \mathbb{R}^{3m_2^2+1}\,.$$

We stress that the above construction is general and encompasses most controller structures of practical interest. We shall see later that interesting control problems such as reliable control are also special cases of the general structured design problem.

With the introduced notation, time-domain design is the optimization program

$$\underset{\kappa \in \mathbb{R}^q}{\text{minimize}}\, f_\infty(\kappa) \text{ with } f_\infty(\kappa) := \max_{t \in [0,T]} f(\kappa, t)$$

where the case $T = \infty$ is allowed. See section 3.1.2 for further details and other practical options.

Frequency-domain design is the standard H_∞ problem and can be cast similarly using the definition

$$f_\infty(\kappa) := \sup_{\omega \in [0,\infty]} \bar{\sigma}(T_{w \to z}(\mathcal{K}(\kappa), j\omega)) = ||T_{w \to z}(\mathcal{K}(\kappa))||_\infty\,.$$

3 Nonsmooth descent method

In this section we briefly present our nonsmooth optimization technique for time- and frequency-domain max functions. For a detailed discussion of the H_∞ norm, we refer the reader to [4, 5]. The setting under investigation is

$$\underset{\kappa}{\text{minimize}} \max_{x \in X} f(\kappa, x)\,, \tag{9}$$

where the semi-infinite variable $x = t$ or $x = \omega$ is restricted to a one-dimensional set X. Here X may be the halfline $[0, \infty]$, or a limited band $[\omega_1, \omega_2]$, or a union of such bands in the frequency domain, and similarly in the time domain. The symbol κ denotes the design variable involved in the controller parametrization $\mathcal{K}(\cdot)$, and we introduce the objective or cost function

$$f_\infty(\kappa) := \max_{x \in X} f(\kappa, x)\,.$$

At a given parameter κ, we assume that we can compute the set $\Omega(\kappa)$ of active times or frequencies, which we assume finite for the time being:

$$\Omega(\kappa) := \{x \in X : f(\kappa, x) = f_\infty(\kappa)\}\,. \tag{10}$$

For future use we construct a finite extension $\Omega_e(\kappa)$ of $\Omega(\kappa)$ by adding times or frequencies to the finite active set $\Omega(\kappa)$. An efficient strategy to construct this set for $x = \omega$ has been discussed in [4, 5].

For the ease of presentation we assume that the cost function f is differentiable with respect to κ for fixed $x \in \Omega_e(\kappa)$, so that gradients $\phi_x = \nabla_\kappa f(\kappa, x)$ are available. Extensions to the general case are easily obtained by passing to subgradients, since $f(., x)$ has a Clarke gradient with respect to κ for every $x \in X$ [12]. Following the line in Polak [23], see also [4], we introduce the optimality function

$$\theta_e(\kappa) := \min_{h \in \mathbb{R}^q} \max_{x \in \Omega_e(\kappa)} -f_\infty(\kappa) + f(\kappa, x) + h^T \phi_x + \tfrac{1}{2} h^T Q h, \tag{11}$$

Notice that θ_e is a first-order model of the objective function $f_\infty(\kappa)$ in (9) in a neighborhood of the current iterate κ. The model offers the possibility to include second-order information [2] via the term $h^T Q h$, but $Q \succ 0$ has to be assured. For simplicity, we will assume $Q = \delta\, I$ with $\delta > 0$ in our tests.

Notice 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: $\theta_e(\kappa) \leq 0$, and $\theta_e(\kappa) = 0$ if and only if $0 \in \partial f_\infty(\kappa)$, that is, κ is a critical point of f_∞. In order to use θ_e to compute descent steps, it is convenient to obtain a dual representation of θ_e. To this aim, we first replace the inner maximum over $\Omega_e(\kappa)$ in (11) by a maximum over its convex hull and we use Fenchel duality to swap the max and min operators. This leads to

$$\theta_e(\kappa) := \max_{\sum_{x \in \Omega_e(\kappa)} \tau_x = 1,\, \tau_x \geq 0} \min_{h \in \mathbb{R}^q} \sum_{x \in \Omega_e(\kappa)} \tau_x (f(\kappa, x) - f_\infty(\kappa) + h^T \phi_x) + \tfrac{1}{2} h^T Q h\,.$$

These operations do not alter the value of θ_e. The now inner infimum over $h \in \mathbb{R}^q$ is now unconstrained and can be computed explicitly. Namely, for fixed τ_x in the outer program, we obtain the solution of the form

$$h(\tau) = -Q^{-1}\left(\sum_{x\in\Omega_e(\kappa)} \tau_x \phi_x\right). \tag{12}$$

Substituting this back in the primal program (11) we obtain the dual expression

$$\theta_e(\kappa) = \max_{\tau_x \geq 0,\ \sum\limits_{x\in\Omega_e(\kappa)} \tau_x = 1} \sum_{x\in\Omega_e(\kappa)} \tau_x \left(f(\kappa, x) - f_\infty(\kappa)\right) - \tfrac{1}{2}\left(\sum_{x\in\Omega_e(\kappa)} \tau_x\phi_x\right)^T Q^{-1}\left(\sum_{x\in\Omega_e(\kappa)} \tau_x\phi_x\right). \tag{13}$$

Notice that in its dual form, computing $\theta_e(\kappa)$ is a convex quadratic program (QP). As a byproduct we see that $\theta_e(\kappa) \leq 0$ and that $\theta_e(\kappa) = 0$ implies κ is critical that is, $0 \in \partial f_\infty(\kappa)$.

What is important is that as long as $\theta_e(\kappa) < 0$, the direction $h(\tau)$ in (12) is a descent direction of f_∞ at κ in the sense that the directional derivative satisfies the decrease condition

$$f'_\infty(\kappa; h(\tau)) \leq \theta_e(\kappa) - \tfrac{1}{2}\left(\sum_{x\in\Omega_e(\kappa)} \tau_x\phi_x\right)^T Q^{-1}\left(\sum_{x\in\Omega_e(\kappa)} \tau_x\phi_x\right) \leq \theta_e(\kappa) < 0,$$

where τ is the dual optimal solution of program (13). See [5, Lemma 4.3] for a proof. In conclusion, we obtain the following algorithmic scheme:

<u>Nonsmooth descent method for $\min_\kappa f_\infty(\kappa)$</u>

Parameters $0 < \alpha < 1$, $0 < \beta < 1$.	
1.	**Initialize**. Find a structured closed-loop stabilizing controller $\mathcal{K}(\kappa)$.
2.	**Active times or frequencies**. Compute $f_\infty(\kappa)$ and obtain the set of active times or frequencies $\Omega(\kappa)$.
3.	**Add times or frequencies**. Build finite extension $\Omega_e(\kappa)$ of $\Omega(\kappa)$.
4.	**Compute step**. Calculate $\theta_e(\kappa)$ by the dual QP (13) and thereby obtain direction $h(\tau)$ in (12). If $\theta_e(\kappa) = 0$ stop. Otherwise:
5.	**Line search**. Find largest $b = \beta^k$ such that $f_\infty(\kappa + bh(\tau)) < f_\infty(\kappa) - \alpha b\theta_e(\kappa)$ and such that $\mathcal{K}(\kappa + b\,h(\tau))$ remains closed-loop stabilizing.
6.	**Step**. Replace κ by $\kappa + b\,h(\tau)$ and go back to step 2.

Finally, we mention that the above algorithm is guaranteed to converge to a critical point [4, 5], a local minimum in practice.

3.1 Nonsmooth properties

In order to make our conceptual algorithm more concrete, we need to clarify how (sub)differential information can be obtained for both time- and frequency-domain design.

3.1.1 Frequency-domain design

In the frequency domain we have $x = \omega$. The function $f_\infty(\kappa)$ becomes $f_\infty(\kappa) = ||.||_\infty \circ T_{w\to z}(.) \circ \mathcal{K}(\kappa)$, which maps $\mathbb{R}^q$ into $\mathbb{R}^+$, and is Clarke subdifferentiable as a composite function [21, 4, 3]. Its Clarke gradient is obtained as $\mathcal{K}'(\kappa)^*\partial g_\infty(\mathcal{K})$, where $\mathcal{K}'(\kappa)$ is the derivative of $\mathcal{K}(.)$ at κ, $\mathcal{K}'(\kappa)^*$ its adjoint, and where g_∞ is defined as $g_\infty := ||.||_\infty \circ T_{w\to z}(.)$ and maps the set $\mathcal{D} \subset \mathbb{R}^{(m_2+k)\times(p_2+k)}$ of closed-loop stabilizing controllers into $\mathbb{R}^+$. Introducing the notation

$$\begin{bmatrix} T_{w\to z}(\mathcal{K},s) & G_{12}(\mathcal{K},s) \\ G_{21}(\mathcal{K},s) & \star \end{bmatrix} := \begin{bmatrix} \mathcal{C}(\mathcal{K}) \\ C_2 \end{bmatrix} (sI - \mathcal{A}(\mathcal{K}))^{-1} \begin{bmatrix} \mathcal{B}(\mathcal{K}) & B_2 \end{bmatrix} + \begin{bmatrix} \mathcal{D}(\mathcal{K}) & D_{12} \\ D_{21} & \star \end{bmatrix} . \tag{14}$$

the Clarke subdifferential of g_∞ at $\mathcal{K}$ is the compact and convex set of subgradients $\partial g_\infty(\mathcal{K}) := \{\Phi_Y : Y \in \mathcal{S}(\mathcal{K})\}$ where

$$\Phi_Y = g_\infty(\mathcal{K})^{-1} \sum_{\omega\in\Omega(\mathcal{K})} \Re\left\{G_{21}(\mathcal{K},j\omega)\, T_{w\to z}(\mathcal{K},j\omega)^H Q_\omega Y_\omega (Q_\omega)^H G_{12}(\mathcal{K},j\omega)\right\}^T , \tag{15}$$

and where $S(\mathcal{K})$ is the spectraplex

$$\mathcal{S}(\mathcal{K}) = \{Y = (Y_\omega)_{\omega\in\Omega(\mathcal{K})} : Y_\omega = (Y_\omega)^H \succeq 0, \sum_{\omega\in\Omega(\mathcal{K})} \mathrm{Tr}\,(Y_\omega) = 1,\, Y_\omega \in \mathbb{H}^{r_\omega}\}.$$

In the above expressions, Q_ω is a matrix whose columns span the eigenspace of $T_{w\to z}(\mathcal{K},j\omega)T_{w\to z}(\mathcal{K},j\omega)^H$ associated with its largest eigenvalue $\lambda_1\left(T_{w\to z}(\mathcal{K},j\omega)T_{w\to z}(\mathcal{K},j\omega)^H\right)$ of multiplicity r_ω.
We also deduce from expression (15) the form of the subgradients of $f(\kappa,\omega) := \bar{\sigma}(T_{w\to z}(\mathcal{K}(\kappa),j\omega))$ at κ with fixed ω, which are used in the primal and dual programs (11) and (13), respectively

$$\phi_x = \Phi_{Y_\omega} = \mathcal{K}'(\kappa)^* f(\kappa,\omega)^{-1} \Re\left\{G_{21}(\mathcal{K},j\omega)\, T_{w\to z}(\mathcal{K},j\omega)^H Q_\omega Y_\omega (Q_\omega)^H G_{12}(\mathcal{K},j\omega)\right\}^T$$

where Q_ω is as before and $Y_\omega \in \mathbb{H}^{r_\omega}$, $Y_\omega \succeq 0$, $\mathrm{Tr}\,(Y_\omega) = 1$. Finally, we note that all subgradient formulas are made implementable by expliciting the action of the adjoint operator $\mathcal{K}'(\kappa)^*$ on elements $F \in \mathbb{R}^{(m_2+k)\times(p_2+k)}$. Namely, we have

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

In the general case, where some of the maximum eigenvalues at some of the frequencies in the extended set $\Omega_e(\kappa)$ has multiplicity > 1, the formulas above should be used, and the dual program in (13) becomes a linear SDP [4, 5]. This is more expensive than a QP, but the size of the SDP remains small, so that the method is functional even for large systems. When max eigenvalues are simple, which seems to be the rule in practice, matrices Y_ω are scalars, and the primal and dual subproblems become much faster convex QPs. This feature, taken together with the fact that Lyapunov variables are never used, explains the efficiency of the proposed technique.

3.1.2 Time-domain design

We now specialize the objective function f_∞ to time-domain specifications. For simplicity of the exposition, we assume the performance channel $w \to z$ is SISO, that is $m_1 = p_1 = 1$, while the controller channel $y \to u$ remains unrestricted.

As noted in [9], most specifications are in fact envelope constraints:

$$z_{min}(t) \leq z(\kappa, t) \leq z_{max}(t) \quad \text{for all } t \geq 0 \tag{16}$$

where $z(\kappa, .)$ is the closed-loop time response to the input signal w (typically a unit step command), when controller $\mathcal{K} = \mathcal{K}(\kappa)$ is used, and where $-\infty \leq z_{min}(t) \leq z_{max}(t) \leq +\infty$ for all $t \geq 0$. This formulation offers sufficient flexibility to cover basic step response specifications such as rise and settling times, overshoot and undershoot, or steady-state tracking. Several constraints of this type can be combined using piecewise constant envelope functions z_{min} and z_{max}. A model following specification is easily incorporated by setting $z_{min} = z_{max} = z_{ref}$, where z_{ref} is the desired closed-loop response.

For a stabilizing controller $\mathcal{K} = \mathcal{K}(\kappa)$, the maximum constraint violation

$$f_\infty(\kappa) = \max_{t \geq 0} \max \left\{ [z(\kappa, t) - z_{max}(t)]^+, \; [z_{min}(t) - z(\kappa, t)]^+ \right\}, \tag{17}$$

where $[.]^+$ denotes the threshold function $[x]^+ = \max\{0, x\}$, is well defined. We have $f_\infty(\kappa) \geq 0$, and $f_\infty(\kappa) = 0$ if and only if $z(\kappa, .)$ satisfies the constraint (16). Minimizing f_∞ is therefore equivalent to reducing constraint violation, and will as a rule lead to a controller $\mathcal{K}(\bar{\kappa})$ achieving the stated time-domain specifications. In the case of failure, this approach converges at least to a local minimum of constraint violation.

The objective function f_∞ is a composite function with a double $\max$ operator. The outer $\max$ on $t \geq 0$ makes the program in (17) semi-infinite, while the inner $\max$, for all $t \geq 0$, is taken over $f_1(\kappa, t) = z(\kappa, t) - z_{max}(t)$, $f_2(\kappa, t) = z_{min}(t) - z(\kappa, t)$ and $f_3(\kappa, t) = 0$.

Assuming that the time response $\kappa \mapsto z(., t)$ is continuously differentiable, f_∞ is Clarke regular and its subdifferential is

$$\partial f_\infty(\kappa) = \mathrm{co}_{\, t \in \Omega(\kappa)} \left\{ \mathrm{co}_{\, i \in \mathcal{I}(\kappa, t)} \nabla_\kappa f_i(\kappa, t) \right\}, \tag{18}$$

where $\Omega(\kappa)$ is the set of active times defined by (10), and $\mathcal{I}(\kappa, t) = \{i \in \{1, 2, 3\} : \; f(\kappa, t) = f_i(\kappa, t)\}$. More precisely, for all $t \in \Omega(\kappa)$,

$$\mathrm{co}_{\, i \in \mathcal{I}(\kappa, t)} \nabla_\kappa f_i(\kappa, t) = \begin{cases} \{\nabla_\kappa z(\kappa, t)\} & \text{if } z(\kappa, t) > z_{max}(t) \\ \{-\nabla_\kappa z(\kappa, t)\} & \text{if } z(\kappa, t) > z_{min}(t) \\ \{0\} & \text{if } z_{min}(t) < z(\kappa, t) < z_{max}(t) \\ [\nabla_\kappa z(\kappa, t), 0] & \text{if } z(\kappa, t) = z_{max}(t) > z_{min}(t) \\ [-\nabla_\kappa z(\kappa, t), 0] & \text{if } z(\kappa, t) = z_{min}(t) < z_{max}(t) \\ [-\nabla_\kappa z(\kappa, t), \nabla_\kappa z(\kappa, t)] & \text{if } z(\kappa, t) = z_{min}(t) = z_{max}(t) \end{cases} \tag{19}$$

Clearly, as soon as the envelope constraint is satisfied for one active time $t \in \Omega(\kappa)$, either one of the last four alternatives in (19) is met, we have $f_\infty(\kappa) = 0$ for all $t \geq 0$ so that $0 \in \partial f_\infty(\kappa)$ and κ is a global minimum of program (9). The computation of the descent step only makes sense in the first two cases, i.e., when $f_\infty(\kappa) > 0$. Notice then that the active times set $\Omega(\kappa)$ can be partitioned into

$$\begin{array}{rcl} \Omega_1(\kappa) & := & \{t: \; t \in \Omega(\kappa), f_1(\kappa, t) = f_\infty(\kappa)\} \\ \Omega_2(\kappa) & := & \{t: \; t \in \Omega(\kappa), f_2(\kappa, t) = f_\infty(\kappa)\} \end{array} \tag{20}$$

and the Clarke subdifferential $\partial g_\infty(\mathcal{K})$ is completely described by the subgradients

$$\Phi_Y(\mathcal{K}) = \sum_{t \in \Omega_1(\mathcal{K})} Y_t \nabla_{\mathcal{K}} z(\mathcal{K}, t) - \sum_{t \in \Omega_2(\mathcal{K})} Y_t \nabla_{\mathcal{K}} z(\mathcal{K}, t) \tag{21}$$

where $Y_t \geq 0$ for all $t \in \Omega(\mathcal{K})$, and $\sum_{t \in \Omega(\mathcal{K})} Y_t = 1$.

Remark. The hypothesis of a finite set $\Omega(\kappa)$ may be unrealistic in the time domain case, because the step response trajectory $z(\cdot, t)$ is not necessarily analytic or piecewise analytic, and may therefore attain the maximum value on one or several contact intervals $[t_-, t_+]$, where t_- is the entry time, t_+ the exit time, and where it is reasonable to assume that there are only finitely many such contact intervals. In that case, our method is easily adapted, and (11) remains correct in so far as the full contact interval can be represented by three pieces of information: the gradients ϕ_x of the trajectory at $x = t_-$, $x = t_+$, and one additional element $\phi_x = 0$ for say $x = (t_- + t_+)/2$ on the interior of the contact interval. (This is a difference with the frequency domain case, where the functions $\omega \mapsto f(\kappa, \omega)$ are analytic, so that the phenomenon of a contact interval could not occur).

A more systematic approach to problems of this form with infinite active sets would consist in allowing choices of finite sets $\Omega_e(\kappa)$, where $\Omega(\kappa) \not\subset \Omega_e(\kappa)$ is allowed. This leads to a variation of the present algorithm discussed in [6, 24, 7], where a trust region strategy replaces the present line search method.

Gradient computation By differentiating the state-space equations (1) with respect to $\mathcal{K}_{ij}$, we get

$$\left\{ \begin{array}{rcl} \dot{\frac{\partial x}{\partial \mathcal{K}_{ij}}}(\mathcal{K}, t) & = & A \frac{\partial x}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) + B_2 \frac{\partial u}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) \\ \frac{\partial z}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) & = & C_1 \frac{\partial x}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) + D_{12} \frac{\partial u}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) \\ \frac{\partial y}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) & = & C_2 \frac{\partial x}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) \end{array} \right. \tag{22}$$

controlled by

$$\begin{array}{rcl} \frac{\partial u}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) & = & \frac{\partial \mathcal{K}}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) y(\mathcal{K}, t) + \mathcal{K} \frac{\partial y}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) \\ & = & y_j(\mathcal{K}, t) e_i + \mathcal{K} \frac{\partial y}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t) \end{array} \tag{23}$$

where e_i stands for the i-th vector of the canonical basis of $\mathbb{R}^{m_2}$. It follows that the partial derivative of the output signal $\frac{\partial z}{\partial \mathcal{K}_{ij}}(\mathcal{K}, t)$ is the simulated output of the interconnection in figure 2, where the exogenous input w is held at 0, and the vector $y_j(\mathcal{K}, t)e_i$ is added to the controller output signal. We readily infer that $n_u \times n_y$ simulations are required in order to form the sought gradients.

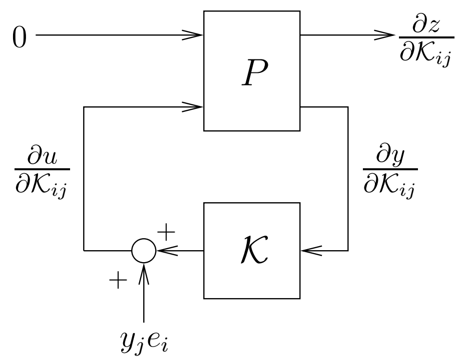

Figure 2: interconnection for gradient computation

This way of computing output signal gradients by performing closed-loop simulations is at the root of the Iterative Feedback Tuning (IFT) method, intially proposed in [17] for SISO systems and controllers. This optimization technique has originated an extensive bibliography (see [16, 15, 13] and references therein) and was extended to multivariable controllers [14]. Most of these papers illustrate the IFT with a smooth quadratic objective function, minimized with the Gauss-Newton algorithm. In [18], the nonsmooth absolute error is used, but a differentiable optimization algorithm (DFP) is applied. Our approach here differs both in the choice of the nonsmooth optimization criterion f_∞, and in the design of a tailored nonsmooth algorithm as outlined in section 3.

Practical aspects The active time sets $\Omega_1(\mathcal{K})$ and $\Omega_2(\mathcal{K})$ are computed via numerical simulation of the closed-loop system in response to the input signal w, see figure 1. This first simulation determines the time samples $(t^l)_{0\leq l\leq N}$ that will be used throughout the optimization phase. Measured output values $\left(y(t^l)\right)$ must be stored for subsequent gradient computation. The extension $\Omega_e(\mathcal{K})$ is built from $\Omega(\mathcal{K})$ by adding time samples with largest envelope constraint violation (16), up to n_Ω elements in all are retained. According to our experiments the set extension generally provides a better model of the original problem as captured by the optimality function θ_e (11) and thus descent directions (12) with better quality are obtained. The gradients $\nabla_\mathcal{K} z(\mathcal{K}, t^l)$ (for $t \in \Omega_e(\mathcal{K})$) result from $n_u \times n_y$ additional simulations of the closed-loop (figure 2) at the same time samples $(t^l)_{0\leq l\leq N}$.

4 Conclusion

We have described a general and very flexible nonsmooth algorithm to compute locally optimal solutions to synthesis problems subject to frequency- or time-domain constraints. Our method offers the new and appealing possibility to integrate controller structures of practical interest in the design. We have now several encouraging reports of successful experiments, which advocate the use of nonsmooth mathematical programming techniques when it comes to solving difficult (often NP-hard) design problems. The results obtained in this paper corroborate previous studies on different problem classes. Extension of our nonsmooth technique to problems involving a mixture of frequency- and time-domain constraints seems a natural next step, which is near at hand. For time-domain design, we have noticed that the proposed technique assumes very little about the system nature, except the access to simulated responses. A more ambitious goal would therefore consider extensions to nonlinear systems.

References

[1] D. Alazard and P. Apkarian. Exact observer-based structures for arbitrary compensators. *Int. J. Robust and Nonlinear Control*, 9(2):101–118, 1999.

[2] P. Apkarian, V. Bompart, and D. Noll. Nonsmooth structured control design with application to PID loop-shaping of a process. *Int. J. Robust and Nonlinear Control*, 17(14):1320–1342, 2007.

[3] P. Apkarian and D. Noll. IQC analysis and synthesis via nonsmooth optimization. *Syst. Control Letters*, 55(12):971–981, 2006.

[4] P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Trans. Aut. Control*, 51(1):71–86, 2006.

[5] P. Apkarian and D. Noll. Nonsmooth optimization for multidisk H_∞ synthesis. *European J. of Control*, 12(3):229–244, 2006.

[6] P. Apkarian, D. Noll, and O. Prot. Trust region spectral bundle method for nonconvex eigenvalue optimization. *submitted*, 2007.

[7] P. Apkarian, D. Noll, and A. Rondepierre. Mixed H_2/H_∞ control via nonsmooth optimization. *to appear in CDC*, 2007.

[8] V. Bompart, D. Noll, and P. Apkarian. Second-order nonsmooth optimization for H_∞ synthesis. *Numerische Mathematik*, 107(3):433–454, 2007.

[9] S. Boyd and C. Barratt. *Linear Controller Design: Limits of Performance*. Prentice-Hall, 1991.

[10] J. V. Burke, D. Henrion, A. S. Lewis, and M. L. Overton. HIFOO - a matlab package for fixed-order controller design and H_∞ optimization. In *5th IFAC Symposium on Robust Control Design*, Toulouse, France, July 2006.

[11] J.V. Burke, A.S. Lewis, and M.L. Overton. Optimizing matrix stability. In *Proceedings of the American Mathematical Society*, volume 129, pages 1635–1642, 2001.

[12] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Canadian Math. Soc. Series. John Wiley & Sons, New York, 1983.

[13] M. Gevers. A decade of progress in iterative process control design: from theory to practice. *J. Process Control*, 12:519–531, 2002.

[14] H. Hjalmarsson. Efficient tuning of linear multivariable controllers using iterative feedback tuning. *Int. J. Adaptive Contr. and Sig. Process.*, 13:553–572, 1999.

[15] H. Hjalmarsson. Iterative feedback tuning–an overview. *Int. J. Adaptive Contr. and Sig. Process.*, 16:373–395, 2002.

[16] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin. Iterative feedback tuning: theory and applications. *IEEE Control Syst. Mag.*, 18(4):26–41, 1998.

[17] H. Hjalmarsson, S. Gunnarsson, and M. Gevers. A convergent iterative restricted complexity control design scheme. In *Proc. of the 33rd IEEE Conference on Decision and Control*, pages 1735–1740, Orlando, FL, 1994.

[18] L. C. Kammer, F. De Bruyne, and R. R. Bitmead. Iterative feedback tuning via minimization of the absolute error. In *Proc. of the 38th IEEE Conference on Decision and Control*, pages 4619–4624, Phoenix, AZ, 1999.

[19] O. Lequin, M. Gevers, M. Mossberg, E. Bosmans, and L. Triest. Iterative feedback tuning of PID parameters: comparison with classical tuning rules. *Control Engineering Practice*, 11:1023–1033, 2003.

[20] M. Mammadov and R. Orsi. H_∞ synthesis via a nonsmooth, nonconvex optimization approach. *Pacific Journal of Optimization*, 1(2):405–420, 2005.

[21] D. Noll and P. Apkarian. Spectral bundle methods for nonconvex maximum eigenvalue functions: first-order methods. *Mathematical Programming Series B*, 104(2):701–727, November 2005.

[22] R. J. Patton. Fault-tolerant control: the 1997 situation. In *Proc. of IFAC Symp. on Fault Detection, Supervision and Safety*, pages 1033–1055, Hull, UK, aug 1997.

[23] E. Polak. *Optimization : Algorithms and Consistent Approximations*. Applied Mathematical Sciences, 1997.

[24] O. Prot, P. Apkarian, and D. Noll. Nonsmooth methods for control design with integral quadratic constraints. *to appear in CDC*, 2007.

[25] J. Stoustrup and V. D. Blondel. Fault tolerant control: A simultaneous stabilization result. *IEEE Trans. Aut. Control*, 49(2):305–310, Feb 2001.

[26] K. Zhou. A new controller architecture for high performance, robust, and fault-tolerant control. *IEEE Trans. Aut. Control*, 46(10):1613 – 1618, Oct 2001.

[27] K. Zhou. A natural approach to high performance robust control: another look at the youla parameterization. In *Proceedings of the SICE 2004 Annual Conference*, pages 869–874, Sapporo, Japan, 2004.