Control Design in the Time- and Frequency Domain using Nonsmooth Techniques

Vincent Bompart [*] Pierre Apkarian [†] Dominikus Noll [‡]

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. Numerical experiments are presented for reliable or fault-tolerant control, and for time response shaping.

Keywords: Nonsmooth optimization, H_∞ synthesis, structured controllers, PID tuning, time-domain constraints, fault tolerance control, time response shaping.

Notation

Let $\mathbb{R}^{n\times m}$ be the space of $n\times m$ matrices, equipped with the corresponding scalar product $\langle X,Y\rangle = \mathrm{Tr}(X^TY)$, where X^T is the transpose of the matrix X, $\mathrm{Tr}\,(X)$ its trace. The notation $\mathbb{H}^r$ stands for the set of Hermitian matrices of size r. 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, $X \succeq Y$ 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. We use concepts from nonsmooth analysis covered by [12]. For a locally Lipschitz function $f:\mathbb{R}^n\to\mathbb{R}$, $\partial f(x)$ denotes its Clarke subdifferential at x while the notation $f'(x;h)$ stand for its directional derivative at x in the direction h.

[*]ONERA-CERT, Centre d'études et de recherches de Toulouse, Control System Department, 2 av. Edouard Belin, 31055 Toulouse, France - and - Institut de Mathématiques, Université Paul Sabatier, Toulouse, France - Email: `bompart@cert.fr` - Tel: +33 5.62.25.22.11 - Fax: +33 5.62.25.27.64.

[†]Corresponding author, ONERA-CERT, Centre d'études et de recherches de Toulouse, Control System Department, 2 av. Edouard Belin, 31055 Toulouse, France - and - Institut de Mathématiques, Université Paul Sabatier, Toulouse, France - Email: `apkarian@cert.fr` - Tel: +33 5.62.25.27.84 - Fax: +33 5.62.25.27.64.

[‡]Université Paul Sabatier, Institut de Mathématiques, 118, route de Narbonne, 31062 Toulouse, France - Email: `noll@mip.ups-tlse.fr` - Tel: +33 5.61.55.86.22 - Fax: +33 5.61.55.83.85.

1 Introduction

Interesting new methods in nonsmooth optimization for the synthesis of controllers have recently been proposed. See [10, 8] for stabilization problems, [4, 5, 20, 3, 11] 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 $f(\kappa) = \max_{\omega\in[0,\infty]} \lambda_1 (F(\mathcal{K}(\kappa),\omega))$, 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. Finally, we present numerical results in observer-based control design and reliable control, as well as for time-response shaping.

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

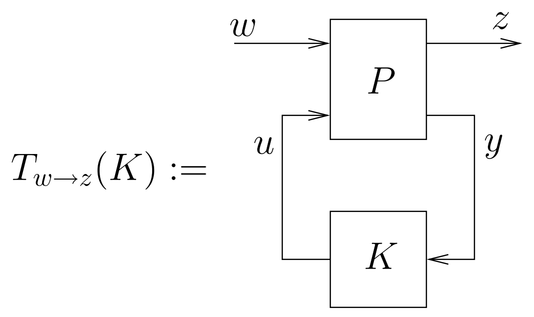


Figure 1: standard interconnection

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

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

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

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

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

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

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

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

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 $x = t$ some ideas will be given in section 4.3.

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

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

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^TQh, 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

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

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

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

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:

Nonsmooth descent method for $\min_\kappa f_\infty(\kappa)$

```
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 $\kappa$ 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 $K'(\kappa)$ is the derivative of $\mathcal{K}(.)$ at κ, $K'(\kappa)^*$ 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

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

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

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

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

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

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:

$$z_{min}(t) \le z(\kappa,t) \le z_{max}(t) \quad \text{for all } t \ge 0 \quad (16)$$

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

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

is well defined, $f_\infty(\kappa) \geq 0$, 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

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

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 $t \in \Omega(\kappa)$,

$$\underset{i \in \mathcal{I}(\kappa, t)}{\text{co}} \nabla_\kappa f_i(\kappa, t) = \left\{ \begin{array}{ll} \{\nabla_\kappa z(\kappa, t)\} & \text{if } z(\kappa, t) > z_{max}(t) \\ \{-\nabla_\kappa z(\kappa, t)\} & \text{if } z(\kappa, t) > z_{min}(t) \\ \{0\} & \text{if } z_{min}(t) < z(\kappa, t) < z_{max}(t) \\ [\nabla_\kappa z(\kappa, t), 0] & \text{if } z(\kappa, t) = z_{max}(t) > z_{min}(t) \\ [-\nabla_\kappa z(\kappa, t), 0] & \text{if } z(\kappa, t) = z_{min}(t) < z_{max}(t) \\ [-\nabla_\kappa z(\kappa, t), \nabla_\kappa z(\kappa, t)] & \text{if } z(\kappa, t) = z_{min}(t) = z_{max}(t) \end{array} \right. \tag{19}$$

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

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

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.

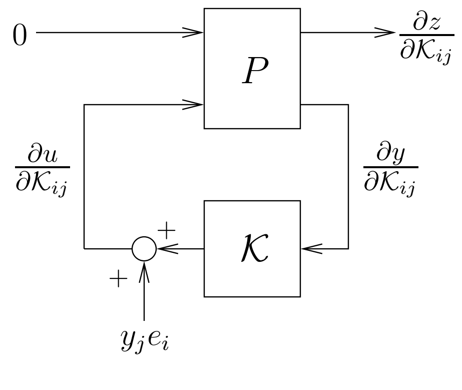

0
P
$\frac{\partial z}{\partial \mathcal{K}_{ij}}$
$\frac{\partial u}{\partial \mathcal{K}_{ij}}$
$\frac{\partial y}{\partial \mathcal{K}_{ij}}$
$\mathcal{K}$
+
+
$y_j e_i$

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 $\big(y(t^l)\big)$ 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 Applications

In this section, we illustrate our nonsmooth technique on a variety of examples including structured H_∞ control, reliable control and time response shaping. We insist on the high flexibility of the proposed nonsmooth techniques and its excellent success in solving hard practical problems.

4.1 Application to observer-based controller design

Observer-based controllers suit many realistic control problems, whenever the system state variables are not directly available for feedback synthesis. Unlike unstructured dynamic controllers, they have a straightforward physical interpretation, since controller states are simply estimates of the plant states. In addition, observer-based controllers are easier to implement and to re-tune.

Let $\hat{x}$ be an asymptotic estimate of the state vector x of the plant P (1). We assume that (C_2, A) is detectable, and that (A, B_2) is stabilizable. As is well known, the estimate $\hat{x}$ is obtained through the full-order Luenberger observer:

$$\dot{\hat{x}} = A\hat{x} + B_2 u + K_f(y - C_2\hat{x}), \tag{24}$$

where $K_f \in \mathbb{R}^{n\times p_2}$ is the state estimator gain. The controlled input is $u = -K_c\hat{x}$, where $K_c \in \mathbb{R}^{m_2\times n}$ is the state feedback gain. With the notations in (3), we get

$$\mathcal{K}(\kappa) = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right] = \left[\begin{array}{c|c} A - B_2K_c - K_fC_2 & K_f \\ \hline -K_c & 0 \end{array}\right] \tag{25}$$

where the controller parameters are described as

$$\kappa := \begin{bmatrix} \operatorname{vec} K_f \\ \operatorname{vec} K_c \end{bmatrix} \in \mathbb{R}^{n(p_2+m_2)}\,.$$

This state-space realization of the standard full-order observer-based controller is linear in κ. It follows that the Jacobian matrix $\mathcal{K}'(\kappa)$ only depends on the plant data A, B_2 and C_2.

Nonsmooth H_∞ synthesis with observer-based controllers has been tested on the following problem discussed in [1], taken from the Matlab library of mu-synthesis examples for Simulink. The state-space equations of the controlled plant are:

$$G(s): \quad \begin{bmatrix} \dot{x_G} \\ y_G \end{bmatrix} = \left[\begin{array}{cc|c} -36.6 & -18.923 & -0.414 \\ -1.9 & 0.983 & -77.8 \\ \hline 0 & 57.3 & 0 \end{array}\right] \begin{bmatrix} x_G \\ u_G \end{bmatrix} \tag{26}$$

The exogenous inputs are the disturbances w_1 (on the actuator signal) and w_2 (on the measurement). Two frequency domain weights $W_u(s)$ and $W_y(s)$ are applied to $u = u_G - w_1$ respectively to $y = y + w_1$, defining the regulated outputs z_1 and z_2. They are

$$W_u(s) = \frac{50s + 5000}{s + 10000}, \text{ and } \quad W_y(s) = \frac{0.5s + 1.5}{s + 10000}. \tag{27}$$

The resulting augmented model (figure 3) has order 4, with state vector $x = [x_G, x_u, x_y]^T$, and the realization used in our computations is the following:

$$P(s): \quad \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \left[\begin{array}{cccc|cc|c} -36.6 & -18.923 & 0 & 0 & -0.414 & 0 & -0.414 \\ -1.9 & 0.983 & 0 & 0 & -77.8 & 0 & -77.8 \\ 0 & 0 & -10000 & 0 & 0 & 0 & 512 \\ 0 & 3667.2 & 0 & -10000 & 0 & 64 & 0 \\ \hline 0 & 0 & -966.8 & 0 & 0 & 0 & 50 \\ 0 & 28.65 & 0 & -78.102 & 0 & 0.5 & 0 \\ \hline 0 & 57.3 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} x \\ w \\ u \end{bmatrix} \tag{28}$$

Notice that the static gains K_f and K_c depend on the state-space realization chosen for P, since the controller state variable $\hat{x}$ estimates x.

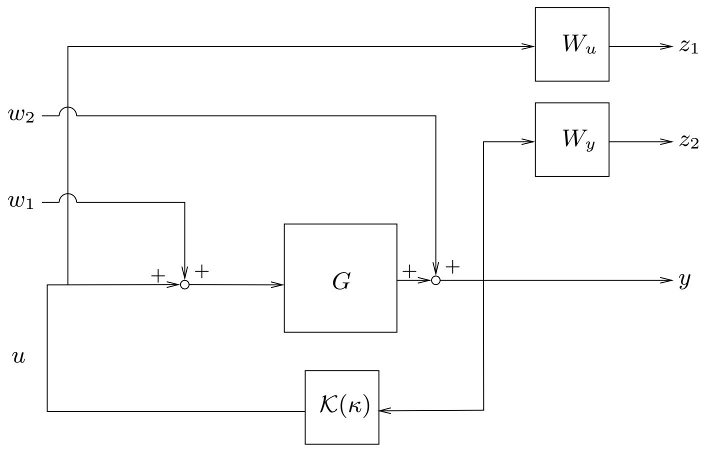


Figure 3: observer-based feedback with augmented model

Running our nonsmooth algorithm to minimize $\|T_{w\to z}(\mathcal{K}(\kappa))\|_\infty$, yields the gains

$$\begin{array}{ll} K_f & = [3886,\ 7203,\ 163.3,\ 29.22]^T \\ K_c & = [-484,\ 219.2,\ 1876,\ -152.9] \end{array}. \tag{29}$$

Correspondingly, the achieved H_∞ norm is 0.5168, which is about the same as the H_∞ performance of the standard full-order H_∞ controller $\gamma = 0.5109$.

4.2 Application to reliable control

Reliable or fault-tolerant control has to guarantee satisfactory stability and performance in nominal conditions as well as in scenarios, where some system components turn faulty or deviate from nominal conditions. The active approach uses a fault detection scheme or identification procedure to adjust the controller in real-time to ensure safe operation. Here the focus is on the passive approach, where a single controller is required to handle most plausible default scenarios. See [25] and the survey [22] for a comprehensive view. We reformulate this problem as a doubly structured design problem and use our nonsmooth method to derive a local solution. In figure (4) we show on the left side various synthesis scenarios, where a single (unstructured or already structured) controller $\mathcal{K}(\kappa)$ must simultaneously minimize the closed-loop performance channels $||T_{w_i \to z_i}(\mathcal{K}(\kappa))||_\infty$ for $i = 1, \ldots, N$. This is readily formulated as a single structured H_∞ synthesis problem, where the sought controller $\mathbf{K}(\kappa)$ has a repeated block-diagonal structure $\mathbf{K}(\kappa) = \mathcal{R} \circ \mathcal{K}(\kappa)$. See the right-hand side of figure 4. A state-space representation of the plant $P(s)$ is readily derived as

$$\left[\begin{array}{c|c|c} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{11} \\ \hline C_2 & D_{21} & D_{22} \end{array}\right] = \left[\begin{array}{c|c|c} \operatorname*{diag}_{i=1,\ldots,N} A^i & \operatorname*{diag}_{i=1,\ldots,N} B_1^i & \operatorname*{diag}_{i=1,\ldots,N} B_2^i \\ \hline \operatorname*{diag}_{i=1,\ldots,N} C_1^i & \operatorname*{diag}_{i=1,\ldots,N} D_{11}^i & \operatorname*{diag}_{i=1,\ldots,N} D_{12}^i \\ \hline \operatorname*{diag}_{i=1,\ldots,N} C_2^i & \operatorname*{diag}_{i=1,\ldots,N} D_{21}^i & \operatorname*{diag}_{i=1,\ldots,N} D_{22}^i \end{array}\right].$$

where the superscript i refers to the ith scenario and where appropriate dynamic augmentations have been performed beforehand for dynamic controllers. Note that the controller is now doubly-structured in the sense that the already structured data $\mathcal{K}(\kappa)$ are repeated in the block-diagonal structure $I_N \otimes \mathcal{K}(\kappa)$, which we describe by the diagonal block repetition operator $\mathcal{R}$ above (see figure 4). This fits nicely into the general framework presented in sections 2 and 3, because the chain rule can obviously be used to cope with a diagonal augmentation $\mathcal{R} \circ \mathcal{K}$ of controller parameterizations.

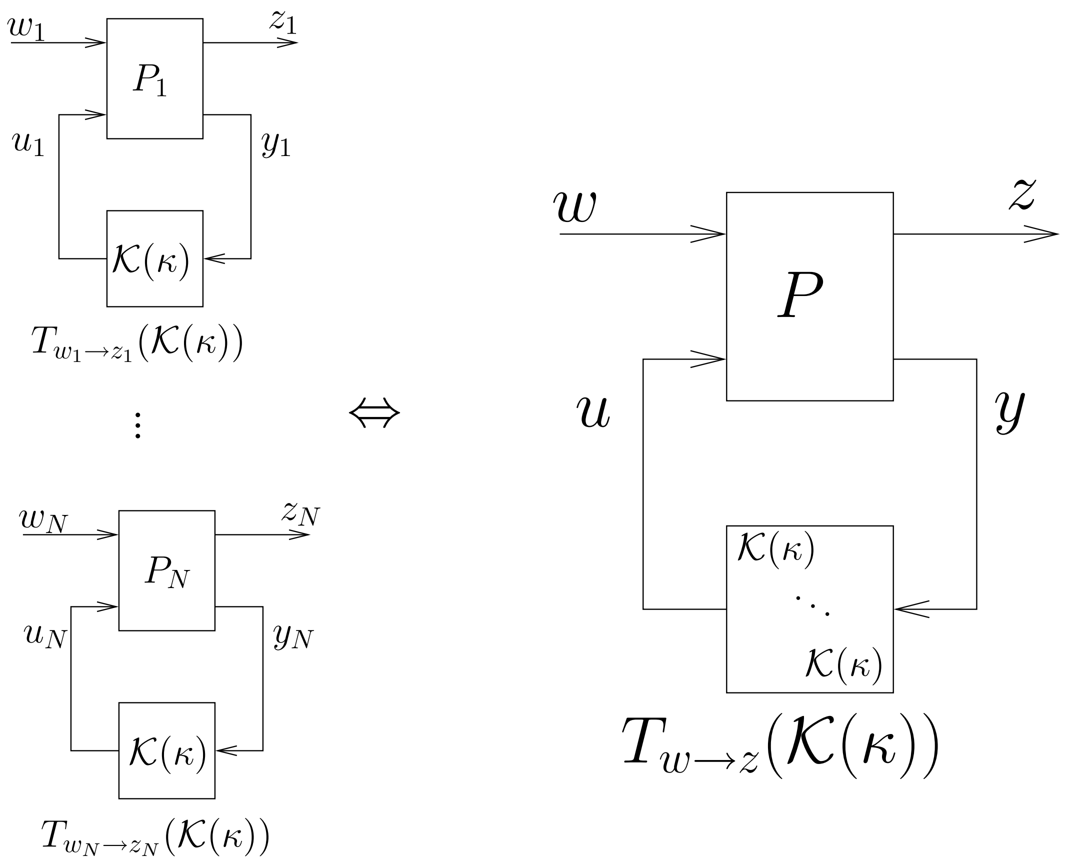


Figure 4: reliable control based on N scenarios

We illustrate the synthesis technique by means of the following example from [27]. It consists of 10 plants G_0 to G_9 with $G_0(s) = \frac{1}{s-1}$, the nominal model and

$$\begin{array}{llllllll} G_1(s) &=& \frac{6.1}{s^2+5.1s-6.1}, & G_2(s) &=& \frac{1.425}{s-1.425} & G_3(s) &=& \frac{0.67}{s-0.67}, \\ G_4(s) &=& \frac{-(s-14.29)}{(s+14.29)(s-1)}, & G_5(s) &=& \frac{4900}{(s-1)(s^2+21s+4900)}, & G_6(s) &=& \frac{4900}{(s+777.7)(s+6.301)(s-1)}, \\ G_7(s) &=& \frac{1562510^6}{(s+50)^6(s-1)}, & G_8(s) &=& \frac{-2.9621(s-9.837)(s+0.7689)}{(s+32)(s-1)(s+0.5612)}, & G_9(s) &=& \frac{4.991(s^2+3.672s+34.85)}{(s+32)(s+7.241)(s-1)}, \end{array}$$

as faulty modes. The design objective is to minimize

$$\max_{i=0,\dots,9} ||W(s)T_{w^i\to z^i}(K)||_\infty, \text{where } T_{w^i\to z^i} := (I + G^i(s)K(s))^{-1} \tag{30}$$

and $W(s) = \frac{0.25s+0.6}{s+0.006}$ is a weighting function penalizing the low frequency range to achieve good tracking properties in response to a step command. Different controller structures $\mathcal{K}(\kappa)$ will be considered in the sequel. The author in [27] investigated various options including a controller with good nominal performance $K_1(s) = \frac{10(0.9s+1)}{s}$, a robust controller $K_2(s) = \frac{2.8s+1}{s}$, and a generalized internal model control (GIMC) implementation based on both $K_1(s)$ and $K_2(s)$, which we shall denote $K_{\text{gimc}}(s)$. The latter is a 6th order controller with Youla parameter

$$Q := \frac{-0.68889s(s+1.452)(s+1)}{(s+1.111)(s^2+1.8s+1)}.$$

Step responses of these controllers for nominal and faulty modes of the plant are displayed in figures 6 to 8. Clearly, K_1 has good nominal performance, but is not satisfactory on faulty modes. On the other hand, controller K_2 shows very robust performance, but responses are slower and the controller performs poorly in the nominal case. The GIMC implementation dramatically improves the situation, since good nominal performance is maintained and the controller also gives good performance on faulty modes.

We have used the proposed nonsmooth technique to minimize the worst case objective in (30) over the set of PID feedback controllers with parametrization given in (6). The following parameter values were obtained:

$$\tau = 0.0100, R_i = 1.1589, R_d = 0.4883, D_K = 4.0263\,.$$

The achieved max H_∞ norm is 0.4886, and step responses with this controller are shown in figure 9. This controller shows clearly better performance than $K_1(s)$ or $K_2(s)$, but performs slightly worse than $K_{\text{gimc}}(s)$. Note that the GIMC controller can also be viewed as a two-degree of freedom controller [26], which suggests using a two-degree of freedom architecture to improve the PID feedback controller. We have used the model following synthesis interconnection shown in figure 5, where G_{ref} is a reference model to compute a second-order feedforward controller $F(s)$ with fixed feedback controller $K(s) = K_{\text{PID}}$. As before, the feedforward controller is required to minimize the worst-case tracking error from r to e over simultaneously the nominal model and the faulty modes

$$\underset{F(s)}{\text{minimize}} \max_{i=0,\ldots,9} ||T^i_{r\to e}(F(s))||_\infty\,.$$

With a slow reference model

$$G_{\text{ref}} = \frac{11.11}{s^2+6s+11.11},$$

the following feedforward controller was obtained

$$F(s) = \frac{-3.0642(s+1.976)(s+1.273)}{(s+3.796)(s+1.049)}.$$

With a faster reference model

$$G_{\text{ref}} = \frac{28.7}{s^2+7.5s+28.7},$$

we have obtained

$$F(s) = \frac{-1.1684(s^2+1.46s+18.51)}{s^2+5.015s+18.39}.$$

Note that the overall controller has order 4 since it consists in a PID controller in the feedback path and a second-order controller for feedforward action. Steps responses are given in figures 10 and 11, respectively, for slow and fast feedforward controllers. Note that the latter controller appears to be the best obtained so far in terms of settling time and overshoot.

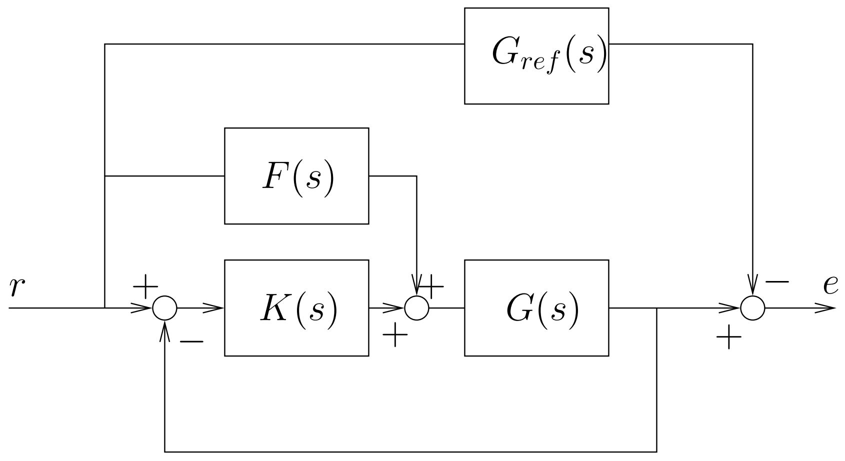
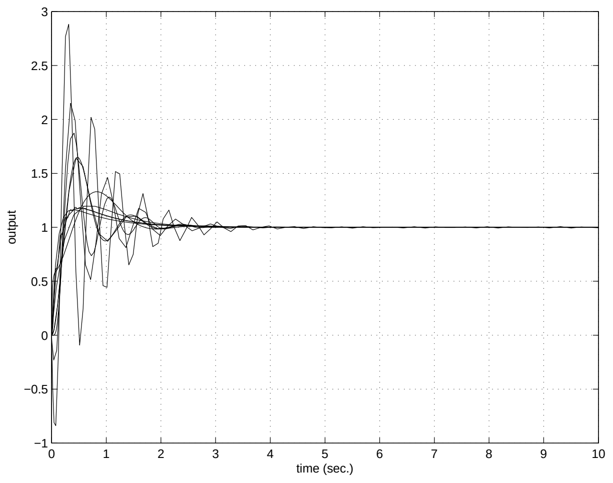


Figure 5: feedforward design with reference model



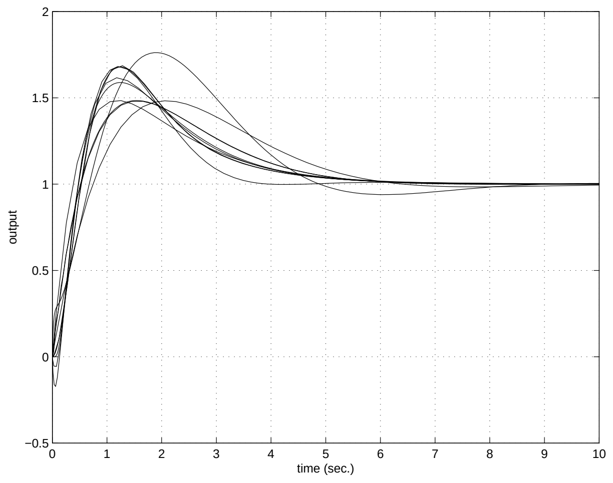


Figure 6: step response with performance controller

Figure 7: step response with robust controller

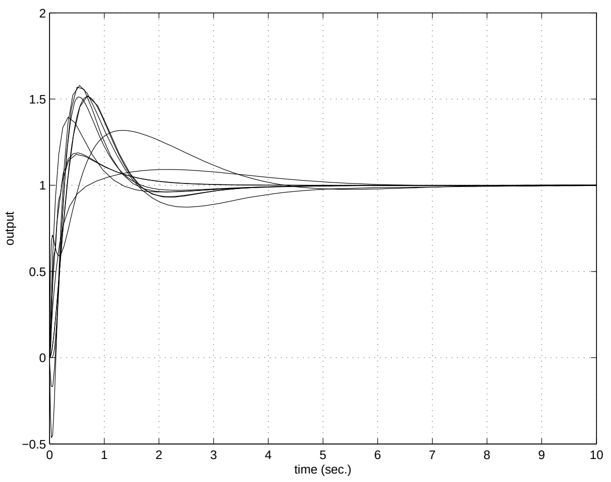


Figure 8: step response with generalized internal model controller

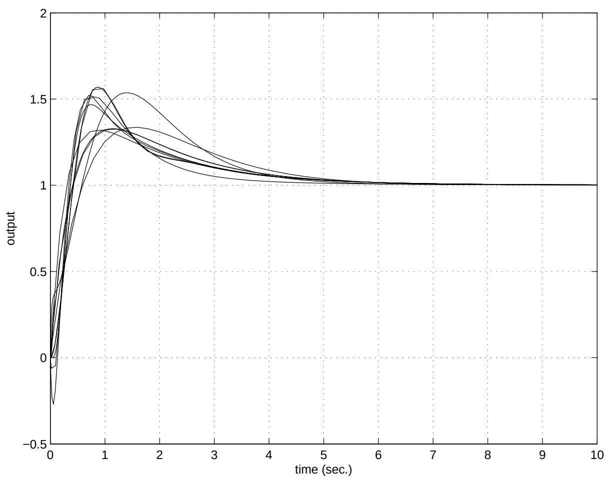


Figure 9: step response with nonsmooth PID controller

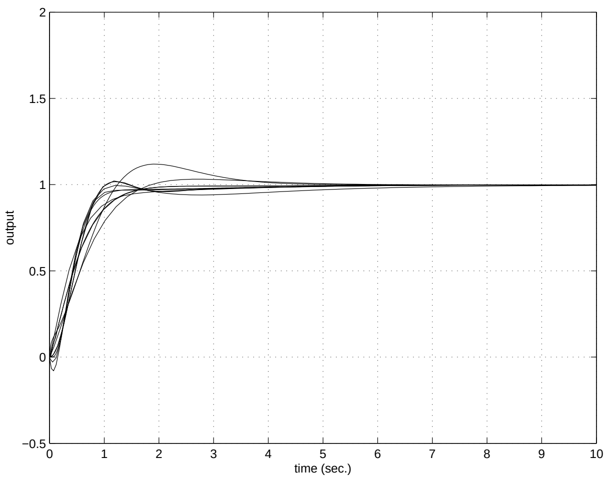


Figure 10: step response with PID + feedforward control, slow reference model

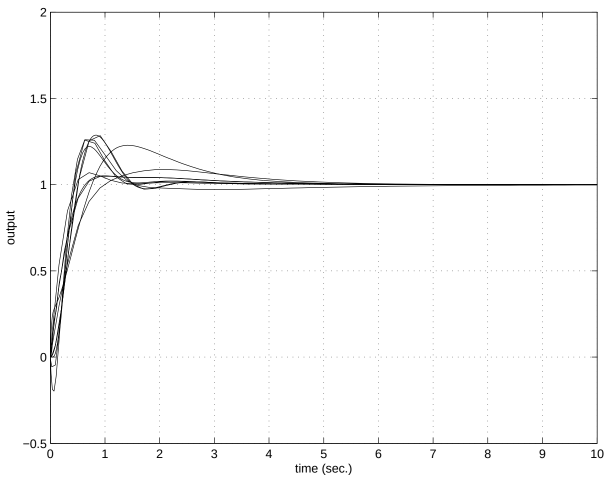


Figure 11: step response with PID + feedforward control, fast reference model

4.3 Application to PID time response shaping

In order to illustrate the ability of the proposed nonsmooth technique to handle time-domain constraints and controller structure, the following test examples from [19] have been used

$$\begin{array}{rclrcl} G_1(s) & = & \frac{1}{20s+1}\mathrm{e}^{-5s}, & G_2(s) & = & \frac{1}{20s+1}\mathrm{e}^{-20s} \\ G_3(s) & = & \frac{1}{(10s+1)^8}, & G_4(s) & = & \frac{-5s+1}{(10s+1)(20s+1)}\,. \end{array} \tag{31}$$

The time delays are replaced by a third-order Padé approximation, which generates nonminimum phase zeros in the transfer functions G_1 and G_2. Each of these plants is regulated by a two-degree-of-freedom (2-DOF) PID controller, as shown in figure 12, with $K_r(s) = K_p + \frac{K_i}{s}$ and $K_{\tilde{y}}(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{1+\varepsilon s}$.

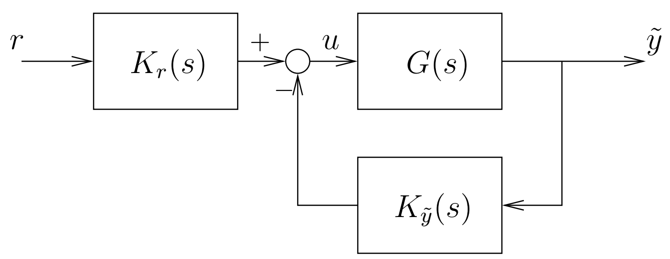


Figure 12: 2-DOF with PID

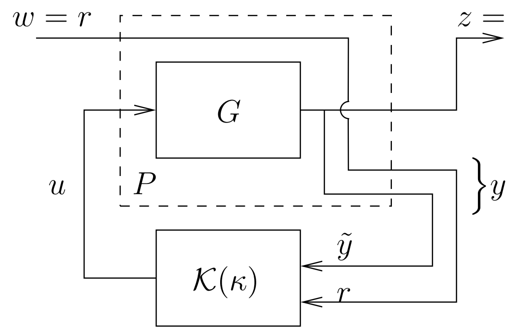


Figure 13: Equivalent standard form to 2-DOF with PID

This 2-DOF controller structure is equivalent to the following state-space representation, in the framework shown in figure 13, with the notations from (7):

$$\mathcal{K}(\kappa) = \left[\begin{array}{ccc|cc} 0 & 0 & 0 & R_i & 0 \\ 0 & -\tau & 0 & R_d & 0 \\ 0 & 0 & 0 & 0 & R_i \\ \hline -1 & -1 & 1 & -D_K & D_K + \frac{R_d}{\tau} \end{array}\right], \tag{32}$$

where the search space is $\kappa = [\tau,\, R_i,\, R_d,\, D_K]^T \in \mathbb{R}^4$.

For each of the four systems (31), the envelope functions were chosen piecewise constant, in order to constrain the overshoot z_{os} and the settling time at $\pm 2\%$ of the steady state. Using the indicatrix function $\mathbb{1}_{[a,b[}$ to restrict to intervals, we define

$$\begin{array}{rcl} z_{max} & = & (1 + z_{os})\,\mathbb{1}_{[0,t_s[} + 1.02\,\mathbb{1}_{[t_s,+\infty[} \\ z_{min} & = & \left\{\begin{array}{ll} -\infty & \text{on } [0, t_s[\\ 0.98 & \text{on } [t_s, +\infty[\end{array}\right. \end{array} \tag{33}$$

The vector κ was initialized with the PID parameters obtained with the Ziegler-Nichols (ZN) tuning rules in [19], a heuristic which gives unsatisfactory step responses for the plants (31), with high overshoot and slow settling times. The results are summarized in table 1, where “NS” indicates the controller parameters obtained from the nonsmooth optimization. They are compared to those from [19], denoted by “IFT”. The corresponding step responses are drawn in figures 14 to 17.

Obviously, arbitrary low z_{os} or t_s are not necessarily achievable with the structured controller $\mathcal{K}(\kappa)$. If the chosen time envelope is too restrictive, the algorithm will converge to a minimizer $\bar{\kappa}$ of f_∞ (i.e. $\theta_e(\bar{\kappa}) \approx 0$) with $f_\infty > 0$.

Conversely, as the algorithm returns a *local* minimizer $\bar{\kappa}$, $f_\infty(\bar{\kappa}) > 0$ does not necessary mean that the time constraints could not be satisfied. A restart with another initial controller κ may very well lead to $f_\infty = 0$. This difficulty appeared for plant G_2 (resp. G_3): with the ZN controller as starting point, we reached $f_\infty = 3.4033 \cdot 10^{-2}$ (resp. $f_\infty = 1.3907 \cdot 10^{-2}$) at convergence, which was an unsatisfactory local minimum. We got around this difficulty by restarting the algorithm from a new point, denoted by “NEW” in table 1.

plant	contr.	ε	K_p	K_i	K_d	t_s (s)	z_{os} (%)
G_1	ZN (init)	10^{-3}	4.0588	0.4388	9.3860	*46.22*	*46.92*
G_1	IFT	10^{-3}	3.6717	0.1324	7.7311	*21.34*	*5.37*
G_1	NS	10^{-3}	3.3315	0.1221	9.3874	*20.94*	*1.95*
G_2	NEW (init)	10^{-3}	0.5000	0.0500	5.0000	*319.45*	*45.72*
G_2	IFT	10^{-3}	0.9303	0.0309	5.6332	*50.15*	*0.90*
G_2	NS	$9.9992 \cdot 10^{-4}$	0.9361	0.0305	4.9920	*49.35*	*1.73*
G_3	NEW	10^{-3}	0.5000	0.0100	10.0000	*235.54*	*0.11*
G_3	IFT	10^{-3}	0.6641	0.0123	12.0959	*131.99*	*1.00*
G_3	NS	10^{-3}	0.6590	0.0121	9.9997	*129.91*	*1.97*
G_4	ZN	10^{-3}	3.5294	0.2101	14.8235	*69.22*	*53.82*
G_4	IFT	10^{-3}	3.0279	0.0654	18.4075	*28.31*	*0.53*
G_4	NS	10^{-3}	2.8947	0.0615	14.8247	*25.58*	*1.67*

Table 1: PID parameters, settling time and overshoot

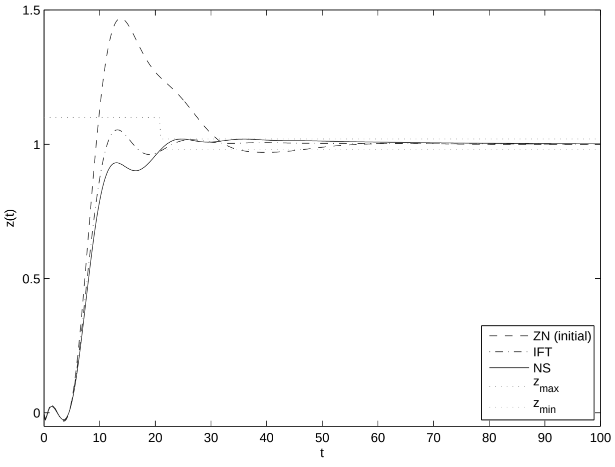


Figure 14: G_1 step responses with PIDs

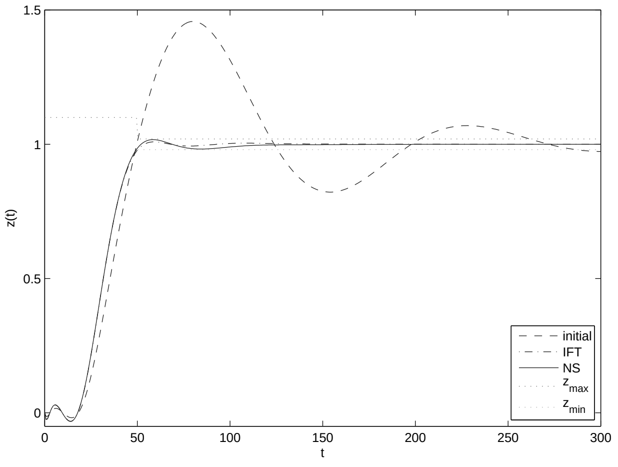


Figure 15: G_2 step responses with PIDs

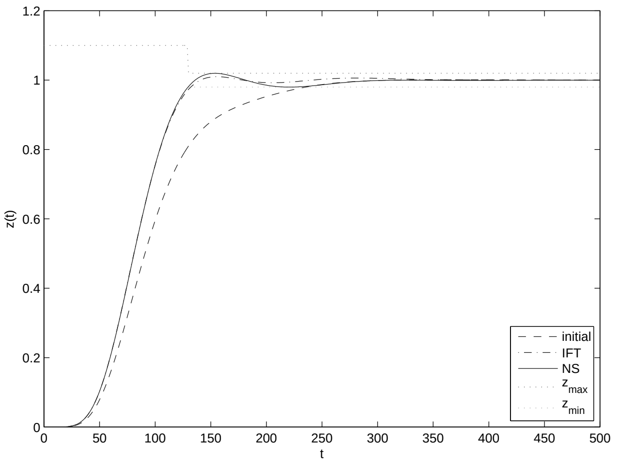


Figure 16: G_3 step responses with PIDs

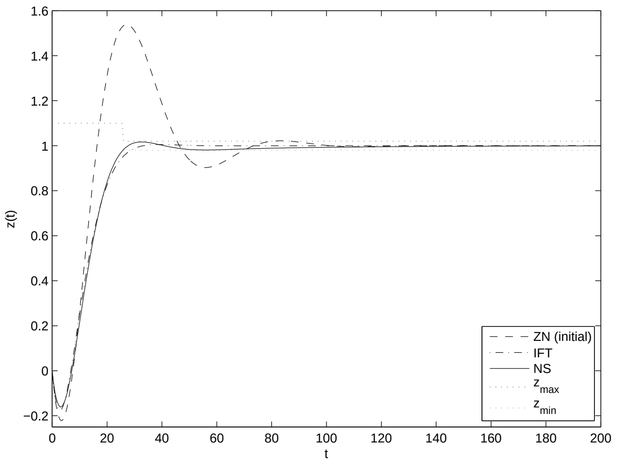


Figure 17: G_4 step responses with PIDs

Finally, we mention that we did not use any temporal mask as is classically done in the IFT approach to avoid spurious solutions.

5 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.

Ackowledgements

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

References

[1] D. Alazard and P. Apkarian, *Exact Observer-Based Structures for Arbitrary Compensators* , Int. J. Control, (1999).

[2] P. Apkarian, V. Bompart, and D. Noll, *Nonsmooth structured control design with application to PID loop-shaping of a process*, to appear in IJRNC, (2007).

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

[4] ——, *Nonsmooth H_∞ synthesis*, IEEE Trans. Aut. Control, 51 (2006), pp. 71–86.

[5] ——, *Nonsmooth optimization for multidisk H_∞ synthesis*, European J. of Control, 12 (2006), pp. 229–244.

[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*, submitted, (2007).

[8] V. Bompart, D. Noll, and P. Apkarian, *Second-order nonsmooth optimization for H_∞ and H_2 syntheses*, submitted, (2005).

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

[10] J. Burke, A. Lewis, and M. Overton, *Optimizing matrix stability*, in Proceedings of the American Mathematical Society, vol. 129, 2001, pp. 1635–1642.

[11] 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.

[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 (2002), pp. 519–531.

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

[15] ——, *Iterative feedback tuning–an overview*, Int. J. Adaptive Contr. and Sig. Process., 16 (2002), pp. 373–395.

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

[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, Orlando, FL, 1994, pp. 1735–1740.

[18] L. C. Kammer, F. D. 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, Phoenix, AZ, 1999, pp. 4619–4624.

[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 (2003), pp. 1023–1033.

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

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

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

[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*, submitted, (2007).

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

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

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