Nonsmooth multi-objective synthesis with applications

A. M. Simões [a,*], P. Apkarian [a,b], D. Noll [b],

[a] *ONERA-CERT, Control System Department, 2 av. Edouard Belin, 31055 Toulouse, France*

[b] *Université Paul Sabatier, Institut de Mathématiques, 118, route de Narbonne, 31062 Toulouse, France*

Abstract

Nonsmooth optimization is used to design feedback controllers subject to closed-loop performance specifications both in time and frequency domains. In time-domain the nonlinear plant is submitted to a set of test input signals and the closed-loop responses so generated are called scenarios. A design technique is proposed which computes a controller with a prescribed structure that satisfies performance specifications for a given set of scenarios in tandem with robustness constraints in the frequency domain, and is locally optimal among other controllers with these properties.

Key words: Nonsmooth optimization, structured controllers, multi-objective design, time response shaping, frequency-domain shaping, tracking, decoupling, reliable control.

1 Introduction

Nonsmooth optimization has been used recently to solve with success a variety of difficult problems in structured linear controller design (Apkarian and Noll, 2006; Bompart et al., 2008; Apkarian et al., 2008; Simões et al., 2008). These design methods avoid using Lyapunov variables whose number grows quadratically with the plant state dimension. This explains why nonsmooth techniques perform satisfactorily even for sizable systems, where standard BMI or LMI

* Corresponding author.
Email addresses: alberto.simoes@cert.fr (A. M. Simões), apkarian@cert.fr (P. Apkarian), noll@mip.ups-tlse.fr (D. Noll).

methods succumb due to the curse of dimension. Yet another appealing feature of nonsmooth optimization is the ease with which controller structures and architectures favoured by practitioners may be addressed directly.

An application of specific interest is synthesis of structured controllers satisfying explicit time-domain specifications. Time domain constraints may involve rise and settling times, overshoot or undershoot, steady-state error, input amplitude and rate constraints or other operational limits on plant trajectories, but also control constraints like saturation. It is therefore deplorable that most existing linear controller synthesis methods do not allow to address these criteria directly. Frequency-domain methods like H_∞ or H_2 synthesis (Zhou et al., 1996) only allow the designer to address time-domain specifications indirectly by setting up suitable frequency weighing filters. This leads to trial-and-error and remains prone to failure. Similar comments apply to optimal control or eigenstructure assignment techniques, see Oliva and Leite Filho (2002). Finally, all these methods use linear plant models, even though most physical systems are nonlinear. This is clearly regrettable if a nonlinear plant model is available for synthesis.

On the other hand, genuine frequency-domain constraints also arise quite regularly in control design, for instance when robustness with regard to model uncertainty or exogenous disturbances is needed. In consequence, controller synthesis techniques working simultaneously in time and frequency domains are of great practical interest, and probably even more if it is allowed to compute control laws with predefined structure. The method presented here does in fact combine these three aspects in a single optimization framework.

Multi-objective controller synthesis has been discussed before. For instance, Polak and Salcudean (1989); Boyd and Barratt (1991) set forth similar ideas. The main difference with the present technique is that these approaches rely on the Youla parametrization, which leads to high-order controllers devoid of any particular practically relevant structure.

In practical applications controllers should not only perform well in the nominal situation, but exhibit some form of robustness e.g. with respect to parameter uncertainties, perturbations, and possibly in situations where the plant operates under conditions which differ significantly from the nominal behavior. Multi-scenario design addresses this situation by grouping various time and frequency-domain specifications for several scenarios of the same system into the design. The present work expands on Apkarian and Noll (2006); Bompart et al. (2008); Apkarian et al. (2008); Simões et al. (2008) in two main aspects. Time-domain responses to input test signals are those of the nonlinear plant if nonlinear dynamics are available. At the same time frequency domain criteria like the H_∞ norm continue to be based on the linearized model, and both aspects are unified in a single optimization program.

The structure of the paper is as follows. The multi-objective synthesis problem is formalized in section 2. The main ingredients of the proposed nonsmooth minimization technique are reviewed in section 3. Two realistic applications are discussed in section 4. In the first application, a robust tracking and decoupling controller under control amplitude and rate constraints is designed for the nonlinear model of a satellite launcher. As a second example, a fault-tolerant flight controller is designed for a combat aircraft in challenging flight conditions subject to wind gusts.

Notation

Let $\mathbb{R}^{n\times m}$ denote the space of $n\times m$ matrices equipped with the corresponding scalar product $\langle X, Y\rangle = X \bullet Y$. Concepts from nonsmooth analysis covered by Clarke (1983) are used. For a locally Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$, $\partial f(x)$ denotes its Clarke subdifferential at x while $f'(x; h)$ stands for the Clarke directional derivative at x in the direction h. For functions of two variables $f(x, y)$, $\partial_1 f(x, y)$ denotes the Clarke subdifferential with respect to the first variable. For differentiable functions f of two variables x and y the notation $\nabla_x f(x, y)$ stands for the gradient with respect to the first variable. The max operator applied to a vector $v \in \mathbb{R}^n$ is defined as $\max v = \max\limits_{i=1,\ldots,n} v_i$. The notation $[.]_+$ applied to a scalar ρ denotes the threshold function $[\rho]_+ = \max\{0, \rho\}$. Its generalization to a vector $v \in \mathbb{R}^n$ is defined as $[v]_+ = \max\{0, \max v\} = \max\limits_{i=1,\ldots,n}[v_i]_+$. The symbol $\mathcal{F}_l(\cdot, \cdot)$ denotes the traditional lower Linear Fractional Transformation. The symbol $\alpha(M)$ represents the spectral abscissa of a matrix $M \in \mathbb{R}^{n\times n}$ defined as $\alpha(M) := \max\{\text{Re } \lambda : \lambda \text{ eigenvalue of M }\}$. The notation conv$(S)$ refers to the convex hull of the set S.

2 Multi-objective synthesis set-up

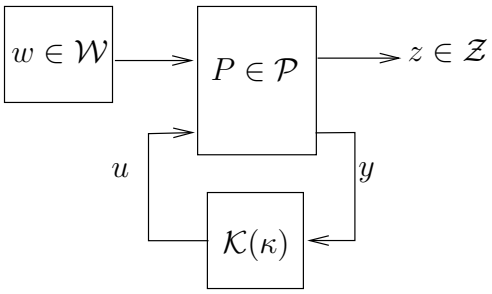

Fig. 1. Multi-scenario interconnection

The set-up for multi-objective synthesis as investigated in this paper is as follows. A structured feedback controller $\mathcal{K}(\kappa)$ is sought which simultaneously

stabilizes a finite family of plants $\mathcal{P} = \{P^1, \ldots, P^p\}$ internally in closed loop, as represented in Figure 1. Moreover, this controller optimizes closed-loop performances of the plants. The set $\mathcal{W}$ consists of test input signals (steps, ramps, sinusoidals) which can be injected into the different plants $P \in \mathcal{P}$, generating closed-loop responses $z = z(w, P, \kappa)$. Each plant $P^i \in \mathcal{P}$ is described by a system of nonlinear ordinary differential equations

$$(P^i) \quad \begin{cases} \dot{x}(t) = \ f_i\,(x, u, w^i, t) \\ z^i(t) = g_{i1}\,(x, u, w^i, t) \\ y(t) = g_{i2}\,(x, u, w^i, t) \end{cases}, \tag{1}$$

which reduces to the familiar state-space description in the particular case of linear time invariant (LTI) systems if linearized:

$$(P^i_{\text{lin}}) \quad \begin{bmatrix} \dot{x} \\ z^i \\ y \end{bmatrix} = \begin{bmatrix} A^i & B^i_1 & B^i_2 \\ C^i_1 & D^i_{11} & D^i_{12} \\ C^i_2 & D^i_{21} & D^i_{22} \end{bmatrix} \begin{bmatrix} x \\ w^i \\ u \end{bmatrix}. \tag{2}$$

The dimensions of the input and output vectors y and u must agree for all plants P^i, because a single controller is used for all P^i, but it is not required that the P^i have the same state dimension or that the (w^i, z^i) have concordant dimensions. Yet, typically all P^i derive from a single system to be controlled. It may also happen that several test signals w^{ij} with responses z^{ij} are used for the same P^i. The set of all these (w^{ij}, z^{ij}) are called scenarios. During the following the index notation is not used, as it does not contribute to readability. Instead, it will be written $z \in \mathcal{Z}$ for the responses arising in the different scenarios, assuming that the corresponding input and plant are then known.

The above somewhat abstract description is flexible enough to include situations where a single plant is submitted to various test signals, as is the case when decoupling properties must be guaranteed, but also in the case where the original system P is split into several operating conditions or faulty modes P^i, which have to be controlled simultaneously. The latter is often referred to as multi-model control (Mäkilä, 1991; Piguet et al., 1999) or reliable control (Liao et al., 2002; Pujol et al., 2007).

Most practical design problems require structured controllers such as PID, decentralized, fixed-order, observer-based etc. It is therefore convenient to introduce a controller parametrization in state-space

$$\kappa \in \mathbb{R}^q \to \mathcal{K}(\kappa) := \begin{bmatrix} A_K(\kappa) & B_K(\kappa) \\ C_K(\kappa) & D_K(\kappa) \end{bmatrix}, \tag{3}$$

with corresponding frequency-domain representation

$$K(s) = C_K(\kappa)(sI - A_K(\kappa))^{-1}B_K(\kappa) + D_K(\kappa)\,.$$

It is not restrictive to assume that the mapping $\mathcal{K}: \mathbb{R}^q \to \mathbb{R}^{(m_2+k)\times(p_2+k)}$ is continuously differentiable, while otherwise arbitrary. See Bompart et al. (2008) and Simões et al. (2008) for examples. In (3), κ denotes the design variables and k stands for the order of the controller, where the case $k = 0$ of a static controller is included.

The goal of multi-scenario design in the time-domain is now the following. Compute $\kappa \in \mathbb{R}^q$ such that the closed-loop time responses $z = z(w, \kappa, \cdot) \in \mathcal{Z}$ obtained with controller $\mathcal{K}(\kappa)$ satisfy shape constraints of the form

$$l_z(t) \le z(t) \le u_z(t),\ \forall t \ge 0,\ \forall z \in \mathcal{Z}, \tag{4}$$

where the lower and upper bounds l_z and u_z for each scenario $z \in \mathcal{Z}$ are usually chosen as piecewise constant. These bounds are illustrated as dashed lines in Figure 2 for a step following specification. The strategy used here is to force these constraints (4) by minimizing the constraint violation function

$$\psi(\kappa) := \max_{z\in\mathcal{Z}} \max_{t\ge 0} \left\{[z(\kappa,t) - u_z(t)]_+,\ [l_z(t) - z(\kappa,t)]_+\right\}\ . \tag{5}$$

Note that the constraints (4) are satisfied as soon as $\psi(\kappa) \le 0$, while $\psi(\kappa) > 0$ indicates constraint violation.

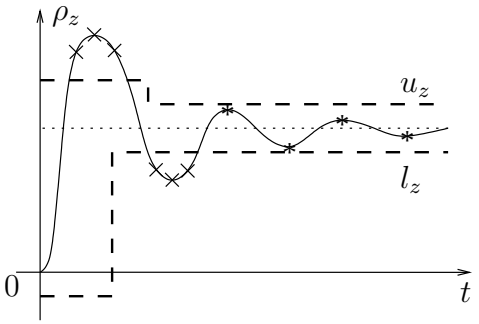

Fig. 2. Shape constraints on time-domain system response

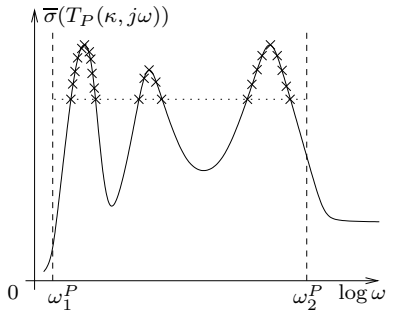

Fig. 3. Extended set $\Omega_P^e(\kappa)$ of active frequencies

The LTI models P_{lin}^i obtained by linearizing the $P^i \in \mathcal{P}$ about steady state allow us to address closed-loop frequency domain specifications. Those arise naturally in synthesis due to robustness issues. Unstructured model uncertainty, for instance, or energy-bounded disturbances require that the controller satisfies specific bounds on the largest singular value norm of certain closed-loop transfers

$$\sup_{\omega\in I_P} \overline{\sigma}\left(T_P(\kappa, j\omega)\right) \le \gamma_P,\ \forall P \in \mathcal{P}_{\text{lin}}, \tag{6}$$

where $T_P(\kappa, s) := \mathcal{F}_l\left(P(s), K(\kappa, s)\right)$, and where $\mathcal{P}_{\text{lin}}$ denotes the set of linearized plants $\{P_{\text{lin}} : P \in \mathcal{P}\}$. For each P specific frequency bands $I_P = [\omega_1^P, \omega_2^P]$, or more generally, $I_P = [\omega_1^P, \omega_2^P] \cup \ldots \cup [\omega_q^P, \omega_{q+1}^P]$, can be specified, where right interval tips may take infinite values. Dynamic weights $W_\infty^P(s)$ may also be used in (6), in which event the constraint take the normalized form

$$\sup_{\omega \in I_P} \overline{\sigma}\left(W_\infty^P(j\omega) T_P(\kappa, j\omega)\right) \leq 1,\ P \in \mathcal{P}_{\text{lin}}\ . \tag{7}$$

Transfers $T_P(\kappa, j\omega)$ may represent, for example, traditional closed-loop sensitivity functions. Constraints (7) will be satisfied as soon as the constraint violation function

$$\sigma(\kappa) := \max_{P \in \mathcal{P}_{\text{lin}}} \sup_{\omega \in I_P} \overline{\sigma}\left(W_\infty^P(j\omega) T_P(\kappa, j\omega)\right) - 1 \tag{8}$$

becomes ≤ 0. For disturbances of finite average power, the H_2 norm may be more appropriate:

$$\left\| W_2^P(s) T_P(\kappa, s) \right\|_2 \leq 1,\ P \in \mathcal{P}_{\text{lin}}. \tag{9}$$

The corresponding constraint violation function is

$$\theta(\kappa) := \max_{P \in \mathcal{P}_{\text{lin}}} \left\| W_2^P(s) T_P(\kappa, s) \right\|_2 - 1\ . \tag{10}$$

Finally, the most fundamental closed-loop specification is internal stability. If this is difficult to achieve based on the performance criteria alone, it may become necessary to include constraints on $\mathcal{K}(\kappa)$ using the closed-loop spectral abscissa

$$\alpha(\mathcal{A}_P(\kappa)) \leq \alpha_P,\ P \in \mathcal{P}_{\text{lin}}, \tag{11}$$

where $\alpha_P < 0$ and $\mathcal{A}_P(\kappa)$ is the state matrix of the closed-loop system $T_P(\kappa, s)$. The constraint violation function for the spectral abscissa is

$$\alpha(\kappa) := \max_{P \in \mathcal{P}_{\text{lin}}} \alpha(\mathcal{A}_P(\kappa)) - \alpha_P\ . \tag{12}$$

Notice that the original controller synthesis problem has been cast as a multi-objective optimization program

$$\underset{\kappa \in \mathbb{R}^q}{\text{minimize}} \quad \left(\psi(\kappa), \sigma(\kappa), \theta(\kappa), \alpha(\kappa)\right)\ , \tag{13}$$

to be optimized until all entries of the objective vector (13) become non-positive.

For practical reasons it may be useful to distinguish between hard and soft constraints in (13). For the time-domain constraints the scenario set $\mathcal{Z}$ is partitioned into disjoint subsets $\mathcal{S}$ and $\mathcal{H}$, i.e., $\mathcal{Z} = \mathcal{S} \cup \mathcal{H}$, $\mathcal{S} \cap \mathcal{H} = \emptyset$,

where $\mathcal{S}$ are soft constraints and $\mathcal{H}$ hard constraints. This gives rise to the constraint violation functions $\psi_{\mathcal{S}}$ and $\psi_{\mathcal{H}}$. A similar partition may be applied to the frequency domain constraints.

The constrained multi-objective optimization problem could then be solved as

$$\begin{aligned} \underset{\kappa\in\mathbb{R}^q}{\text{minimize}} \;\; & f(\kappa) := \max\{\psi_{\mathcal{S}}(\kappa), \sigma_{\mathcal{S}}(\kappa), \theta_{\mathcal{S}}(\kappa), \alpha_{\mathcal{S}}(\kappa)\} \\ \text{subject to } & g(\kappa) := \max\{\psi_{\mathcal{H}}(\kappa), \sigma_{\mathcal{H}}(\kappa), \theta_{\mathcal{H}}(\kappa), \alpha_{\mathcal{H}}(\kappa)\} \leq 0 \;, \end{aligned} \tag{14}$$

where objective and constraint now regroup constraint violation functions from (5), (8), (10) or (12). A solution to program (14), being feasible, necessarily meets the hard constraints, while soft constraints will be achieved only when the objective function falls below 0. In program (14), the role of individual weights for the various specifications is played by tuning parameters l_z, u_z, W_∞^P, W_2^P and α_P. The strategy adopted here is to select these weights in close spirit with the aspiration levels approach for multi-objective optimization (Boyd and Barratt, 1991, p.64). Tuning parameters are adjusted iteratively based on a few trial-and-error designs. For instance, hard constraints which are easy to satisfy in one run can be tightened in a next stage, while violated constraints which continue to resist may have to be relaxed. One of the appealing features of this approach is that tuning parameters are closely related to engineering specifications, so such parameters can be adjusted based on intuition or engineering insight.

The synthesis framework (14) gives the designer the flexibility to handle design specifications directly as posed in practice. Program (14) is, however, a difficult nonconvex and nonsmooth mathematical program with semi-infinite constraints (5) or (8). Specific nonsmooth optimization techniques have been developed in Apkarian et al. (2008) to address these problems, and the key ingredients are recalled in section 3. Standard smooth optimization packages could be used to solve programs like (14), but this bears the risk of failure if algorithms encounter so-called *dead points* (Apkarian and Noll, 2006b). Even when used to produce good starting values, smooth optimization should in the end always be completed by a nonsmooth optimization phase, to avoid missing a local optimum. There is a longstanding experience in using classical smooth methods for solving nonsmooth problems. Smooth methods invariably encounter breakdowns at points that are not local optimum (kinks, multiple active functions, etc) because local optima are typically nonsmooths points in practice.

In this section, the key ingredients of the nonsmooth optimization method used in the experimental section 4 are recalled. The reader is referred to Apkarian et al. (2008) for details.

For the sake of clarity, program (14) is represented in the more abstract form

$$\begin{array}{ll} \underset{\kappa}{\text{minimize}} & f(\kappa) \\ \text{subject to} & g(\kappa) \leq 0\,. \end{array} \tag{15}$$

where both objective and constraints can contain a mix of frequency and time domain elements as in (14). To solve the constrained program (15), the following progress function is introduced, following an idea in Polak (1997):

$$F(\kappa^+, \kappa) = \max\{f(\kappa^+) - f(\kappa) - \mu g(\kappa)_+; g(\kappa^+) - g(\kappa)_+\}, \tag{16}$$

where $\mu > 0$ is some fixed parameter (with $\mu = 1$ a typical value). Here κ represents the current iterate, κ^+ is the next iterate or a candidate for the next iterate. Except the case where $\bar{\kappa}$ is a local minimum of the constraint violation $g(\bar{\kappa}) > 0$, it is shown in Polak (1997) that critical points $\bar{\kappa}$ of $F(\cdot, \bar{\kappa})$ will also be critical points of the original program (15). Refer to Polak (1997) and Apkarian et al. (2008) for an in-depth discussion of this property.

Minimizing the progress function in (16) leads to a so-called phase I/phase II method. As long as the constraint $g(\kappa) \leq 0$ is not satisfied, the right hand term in (16) is dominant and reducing it means reducing constraint violation. This is phase I, which ends successfully as soon as a feasible iterate $g(\kappa^k) \leq 0$ has been found. Now phase II begins, and from now on iterates stay (strictly) feasible, while the objective function is minimized at each step. Notice that the choice of the constant $\mu > 0$ may have an influence on the behavior of the method in phase I (Apkarian et al., 2008), but has been fixed as $\mu = 1$ in the numerical implementation.

The search for a point $\bar{\kappa}$ with $0 \in \partial_1 F(\bar{\kappa}, \bar{\kappa})$ is based on an iterative descent procedure. Suppose the current iterate κ has $0 \notin \partial_1 F(\kappa, \kappa)$. Then it is possible to further reduce the function $F(\cdot, \kappa)$ in a neighborhood of κ, that is, one can find κ^+ such that $F(\kappa^+, \kappa) < F(\kappa, \kappa)$. Replacing κ by κ^+, the procedure is repeated. Unless $0 \in \partial_1 F(\kappa^+, \kappa^+)$, in which case the search is over, it is possible to find κ^{++} such that $F(\kappa^{++}, \kappa^+) < F(\kappa^+, \kappa^+)$, etc. The sequence $\kappa, \kappa^+, \kappa^{++}, \ldots$ so generated is expected to converge to the sought local minimum $\bar{\kappa}$ of (15) if the reduction is substantial in a sense made precise in the above references.

Descent steps κ^+ away from the current κ are found by solving a tangent

program at κ. Its name derives from the fact that a first-order approximation $\widehat{F}(\cdot,\kappa)$ of $F(\cdot,\kappa)$ is built, which provides a descent direction $d\kappa$ at κ, that is, $d_1F(\kappa,\kappa;d\kappa) < 0$, where d_1F denotes the directional derivative of $F(\cdot,\kappa)$ at κ in direction $d\kappa$. The next iterate is then $\kappa^+ = \kappa + d\kappa$, or possibly $\kappa^+ = \kappa + \xi d\kappa$ for a suitable stepsize $\xi \in (0,1)$ found by backtracking.

3.1 Search directions from the tangent program

In order to generate a first-order approximation $\widehat{F}(\cdot,\kappa)$ of $F(.,\kappa)$ around κ, the specific structure of the criteria is exploited, and the concept of active times and active frequencies is needed. For time domain constraints this is explained for the function $\psi_{\mathcal{S}} = \max_{z\in\mathcal{S}} \psi_z$. Consider a scalar-valued scenario $z \in \mathcal{S}$. Then the violation function is of the form $\psi_z(\kappa) = \max_{t\geq 0} \psi_z(\kappa,t)$, where

$$\begin{aligned}\psi_z(\kappa,t) &:= \max\{[z(\kappa,t) - u_z(\kappa,t)]_+,\ [l_z(t) - z(\kappa,t)]_+\} \\ &= \max\{z(\kappa,t) - u_z(t),\ l_z(t) - z(\kappa,t),\ 0\}\,. \end{aligned} \tag{17}$$

Thus, the set of active times for this entry $z \in \mathcal{S}$ at κ is given as

$$A_z(\kappa) := \{t \geq 0 : \psi_z(\kappa,t) = \psi_{\mathcal{S}}(\kappa)\}.$$

For vector-valued scenarios $z \in \mathcal{S}$ the definition is applied to each coordinate of z. Finally the set $A_{\psi_{\mathcal{S}}}(\kappa)$ of active times for $\psi_{\mathcal{S}}$ is simply the union of the $A_z(\kappa)$, $z \in \mathcal{S}$. Of course an analogous definition holds for $A_{\psi_{\mathcal{H}}}$.

The concept of active frequencies is now defined. Consider a function of the form

$$\sigma_{\mathcal{S}}(\kappa) = \max_{P\in\mathcal{S}} \max_{\omega\geq 0} \sigma_P(\kappa,\omega),$$

where

$$\sigma_P(\kappa,\omega) := \overline{\sigma}\left(W_\infty^P(j\omega)T_P(\kappa,j\omega)\right) - 1\,, \tag{18}$$

for $\omega \geq 0$, $P \in \mathcal{P}_{\text{lin}}$. Clearly $\mathcal{S}$ now denotes the subset of $\mathcal{P}_{\text{lin}}$ chosen for soft constraints. The set of active frequencies for $P \in \mathcal{S}$ is given as

$$\Omega_P(\kappa) := \{\omega \geq 0 : \sigma_P(\kappa,\omega) = \sigma_{\mathcal{S}}(\kappa)\},$$

while the set $\Omega_{\sigma_{\mathcal{S}}}(\kappa)$ of active frequencies of $\sigma_{\mathcal{S}}$ at κ is the union of the $\Omega_P(\kappa)$, $P \in \mathcal{S}$.

As a rule, the sets $A_{\psi_{\mathcal{S}}}(\kappa)$ and $\Omega_{\sigma_{\mathcal{S}}}(\kappa)$ are finite or can at least be finitely generated, and similarly for subscripts $\mathcal{H}$. In order to increase the efficiency of the method, finite extensions $A^e_{\psi_{\mathcal{S}}}(\kappa)$ of $A_{\psi_{\mathcal{S}}}(\kappa)$ and $\Omega^e_{\sigma_{\mathcal{S}}}(\kappa)$ of $\Omega_{\sigma_{\mathcal{S}}}(\kappa)$ are used, where typically near-active times or frequencies are added. For every

$t \in A^e_{\psi_\mathcal{S}}(\kappa)$ and $\omega \in \Omega^e_{\sigma_\mathcal{S}}(\kappa)$ certain subgradients will be picked for the tangent program and will be stored in a subgradient set $\mathcal{G}$. It will now be explained how to pick these subgradients for the model functions $\psi_\mathcal{S}$ and $\sigma_\mathcal{S}$.

To begin with, consider a time domain function $\psi_\mathcal{S} = \max_{z\in\mathcal{S}} \psi_z$. Closed-loop time responses $\kappa \to z(\cdot, t)$ are assumed to be continuously differentiable. Clarke's formula for the subdifferential of a max-function applies and gives the representation

$$\partial\psi_\mathcal{S}(\kappa) = \text{conv} \cup \{\partial\psi_z(\kappa, t) : t \in A_z(\kappa), z \in \mathcal{S}\},$$

where the specific structure of the ψ_z for a scalar $z \in \mathcal{S}$ gives

$$\partial\psi_z(\kappa,t) = \begin{cases} \{\nabla z(\kappa,t)\}, & \text{if } z(\kappa,t) > u_z(t) \\ \text{conv}\{\nabla z(\kappa,t)\}, 0\} & \text{if } z(\kappa,t) = u_z(t) \\ \{0\} & \text{if } l_z(\kappa) < z(\kappa,t) < u_z(\kappa) \\ \text{conv}\{-\nabla z(\kappa,t)\}, 0\} & \text{if } z(\kappa,t) = l_z(t) \\ \{-\nabla z(\kappa,t)\} & \text{if } z(\kappa,t) < l_z(t). \end{cases}$$

Therefore, for every active $t \in A_z(\kappa)$ a pair (ϕ_t, Φ_t) with $\phi_t := \psi_z(\kappa, t) = \psi_\mathcal{S}(\kappa)$ and $\Phi_t \in \partial\psi_z(\kappa, t)$ is stored in the set $\mathcal{G}_{\psi_\mathcal{S}}(\kappa)$. For $t \in A^e_z(\kappa) \setminus A_z(\kappa)$ the same pair (ϕ_t, Φ_t) is stored in $\mathcal{G}_{\psi_\mathcal{S}}(\kappa)$, even though $\phi_t = \psi_z(\kappa, t) < \psi_\mathcal{S}(\kappa)$ in this case. In this way first-order approximations of $\psi_z(\kappa + d\kappa, t)$ in the neighborhood of κ are obtained: $\psi_z(\kappa + d\kappa, t) \approx \alpha_t + \Phi_t^T d\kappa$ if $(\phi_t, \Phi_t) \in \mathcal{G}_{\psi_\mathcal{S}}$. Consequently, the desired first-order approximation for $\psi_\mathcal{S}$ around κ is

$$\psi_\mathcal{S}(\kappa + d\kappa) \approx \max_{(\phi_t,\Phi_t)\in\mathcal{G}_{\psi_\mathcal{S}}(\kappa)} \phi_t + \Phi_t^T d\kappa. \tag{19}$$

Notice that the term on the right is the upper envelope of affine functions and is therefore convex as a function of $d\kappa$. If only the sets $A_z(\kappa)$ were used to build this model, then all these affine lines would be generalized tangents to the graph of $\psi_\mathcal{S}$ at κ. Having enlarged $A_z(\kappa)$ into $A^e_z(\kappa)$ makes that some of the lines in the model pass strictly below $\psi_\mathcal{S}(\kappa)$, so do not contribute at κ. However, these lines can quickly become active as one moves from κ to a nearby $\kappa + d\kappa$, hence the significance of the extension $A^e_z(\kappa)$ for the local model (19).

Building a first-order approximation of $\sigma_\mathcal{S}(\kappa + d\kappa)$ around κ is slightly more involved, because the largest singular value functions $\sigma_P(\kappa, \omega)$ are nonsmooth in κ even for fixed ω and P. Instead of working with the maximum singular value $\overline{\sigma}(WT)$ as in (6), it will be convenient to work with the maximum eigenvalue $\lambda_1(WTT^HW^H)$. Therefore, the symmetrization $S_P(\kappa, s) :=$

$(W_\infty^P(s)T_P(\kappa,s))(W_\infty^P(s)T_P(\kappa,s))^H$ is introduced, as well as $\lambda_{\mathcal{S}}(\kappa) := \overline{\sigma}_{\mathcal{S}}(\kappa)^2 = \max_{P\in\mathcal{S}} \max_{\omega\geq 0} \lambda_1\left(S_P(\kappa, j\omega)\right)$.

A first-order approximation of $\lambda_P(\kappa+d\kappa, j\omega)$ in the neighborhood of κ is given by (Apkarian and Noll, 2006; Simões et al., 2008)

$$\lambda_P(\kappa + d\kappa, j\omega) \approx \sup_{\substack{Y_w\geq 0,\\ \mathrm{Tr}(Y_w)=1}} Q_w Y_w Q_w^H \bullet \left(S_P(\kappa, j\omega) + S_P'(\kappa, j\omega)d\kappa\right) - 1 \qquad (20)$$

where $S_P'(\kappa, j\omega)$ is the Fréchet derivative of $S_P(\cdot, j\omega)$ at κ, and Q_ω is an orthonormal basis of the eigenspace of $\lambda_1\left(S_P(\kappa, j\omega)\right)$ at κ. Define

$$\phi_g := Q_\omega Y_\omega Q_\omega^H \bullet S_P(\kappa, j\omega) - 1, \qquad \Phi_g := S_P'(\kappa, j\omega)^\star Q_\omega Y_\omega Q_\omega^H$$

and store the pairs $(\phi_\omega, \Phi_\omega)$ in the set $\mathcal{G}_{\sigma_{\mathcal{S}}}(\kappa)$ for all $\omega \in \Omega^e_{\sigma_{\mathcal{S}}}(\kappa)$. As previously observed, enlarging Ω into Ω^e gives the model some robustness. Notice, however, a difference with the model building of ψ. The pairs in $\mathcal{G}_{\sigma_{\mathcal{S}}}$ are indexed by Y_ω ranging over the set of matrices $\{Y_\omega : Y_\omega \succeq 0, \mathrm{Tr}(Y_\omega) = 1\}$, which is infinite if $\dim(Y_\omega) > 1$. If all these subgradients are kept, the tangent program will be a small size semidefinite program. However, it has be shown in Apkarian et al. (2008) that one can get by with a finite set of subgradients, so tangent programs will turn out to be small to medium size convex quadratic programs, which can be solved very efficiently.

The function $\theta(\kappa)$ being smooth, one can keep the choice of $(\phi_\theta(\kappa), \Phi_\theta(\kappa))$ simple by taking $(\theta(\kappa), \nabla\theta(\kappa))$. Enlarging is possible, but not mandatory due to smoothness. Similarly, for the spectral abscissa the function value and a single subgradient suffice, which can be computed as outlined in Bompart et al. (2007).

The tangent model of $F(., \kappa)$ at κ is now constructed by assembling the first-order approximations of all the branches of the max-function in (16). With the above preparation, suppose finitely many pairs $(\phi_f, \Phi_f) \in \mathcal{G}_f(\kappa)$ and $(\phi_g, \Phi_g) \in \mathcal{G}_g(\kappa)$ have been constructed for objective and constraint. Then a first-order approximation of the progress function is

$$\widehat{F}(\kappa + h, \kappa) := \max\left\{ \max_{(\phi_f,\Phi_f)\in\mathcal{G}_f(\kappa)} \phi_f - f(\kappa) - \mu g(\kappa)_+ + \Phi_f^T h, \right.$$
$$\left. \max_{(\phi_g,\Phi_g)\in\mathcal{G}_g(\kappa)} \phi_g - g(\kappa)_+ + \Phi_g^T h \right\}, \qquad (21)$$

where h is the displacement in the controller parameter space $\mathbb{R}^q$. This gives the tangent program

$$\underset{h\in\mathbb{R}^q}{\text{minimize}}\ \widehat{F}(\kappa + h, \kappa) + \tfrac{\delta}{2}\|h\|^2, \qquad (22)$$

with $\delta > 0$ a fixed parameter. It is worth noting that an equivalent formulation for (22) is the following

$$\begin{array}{ll} \underset{t,\, h \in \mathbb{R}^q}{\text{minimize}} & t + \frac{\delta}{2}\|h\|^2 \\ \text{subject to} & \phi_f - f(\kappa) - \mu g(\kappa)_+ + \Phi_f^T h \leq t,\ \forall (\phi_f, \Phi_f) \in \mathcal{G}_f(\kappa)\,, \\ & \phi_g - g(\kappa)_+ + \Phi_g^T h \leq t,\ \forall (\phi_g, \Phi_g) \in \mathcal{G}_g(\kappa)\,. \end{array} \tag{23}$$

When the eigenvalue multiplicity of all maximum eigenvalue functions equals 1, one can select $Y_w = 1$ in (20). Then (23) is a standard convex quadratic program (CQP), and can be efficiently solved using currently available codes. Current state-of-the-art CQP codes solve problems involving several hundreds of variables and constraints in less than a second. Note that the quadratic term in (22) can be used to capture second-order information, or it may be interpreted as a trust region radius management parameter. The reader is referred to Apkarian and Noll (2006), Apkarian et al. (2008) and Simões et al. (2008) for more elaborate variations of the present technique, and to Polak (1997) for a general view on phase I/phase II methods. The key facts about (22) or (23) have been established in Apkarian and Noll (2006) and are stated here without proof:

- The fact that the extended sets contain the active sets ensures that the solution to (22) is a descent direction of $F(.,\kappa)$ at κ. If $h = 0$ then $0 \in \partial_1 F(\kappa,\kappa)$, and the search is over. Clearly, a stopping test may be based on the smallness of the solution h to the tangent program.
- The direction h can be used in an Armijo line search (J.E. Dennis, Jr. and Schnabel, 1996) defined by a step ξ in direction h with:

$$F(\kappa + \xi\, h, \kappa) - F(\kappa,\kappa) < \gamma \xi F'(.,\kappa)(\kappa; h),$$

  where $0 < \gamma < 1$, which terminates after finitely many steplength trials $\xi \in (0, 1]$.

Both items use the fact that $\partial_1 \widehat{F}(\kappa,\kappa) = \partial_1 F(\kappa,\kappa)$. Having described the main features of the algorithm, its pseudo-code is as follows:

Algorithm 1. Nonsmooth algorithm for program (15)

Parameters: $\delta > 0$, $0 < \beta < 1$, $0 < \gamma < 1$.

1: **initialize**. Select initial κ^1. Put counter $j = 1$.

2: **stopping test**. At counter j, stop if $0 \in \partial_1 \widehat{F}(\kappa^j, \kappa^j)$ and return κ^j. Otherwise continue.

3: **compute descent direction.** At counter j solve tangent programs (22) or (23)

$$\min_{h \in \mathbb{R}^q} \widehat{F}(\kappa^j + h, \kappa^j) + \tfrac{\delta}{2}\|h\|^2.$$

Solution is the search direction h^j.

4: **line search.** Find $\xi = \beta^\nu$, $\nu \in \mathbb{N}$, satisfying the Armijo condition

$$F(\kappa^j + \xi h^j, \kappa^j) - F(\kappa^j, \kappa^j) \leq \gamma \xi F'(\cdot, \kappa^j)(\kappa^j, h^j) < 0.$$

5: **update**. Put $\kappa^{j+1} = \kappa^j + \xi h^j$, increase counter j by 1 and loop back to step 2.

3.2 Implementation details

Similarly to *iterative feedback tuning* (IFT), the proposed technique relies on simulations to compute function values as well as trajectory gradients for time-domain constraints. A comprehensive discussion on how this can be done is presented in Hjalmarsson (2002), Hjalmarsson et al. (1998) and Bompart et al. (2008). This is generally the costly part of the technique since $\dim \kappa = q$ simulations for each scenario may be required in order to form the trajectories gradients. Simulations serving to compute trajectory gradients for nonlinear plants require the auxiliary nonlinear system

$$\begin{cases} \frac{\partial \dot{x}}{\partial \kappa_j}(t) = \frac{\partial f}{\partial x}(x,u,w,t)\,\frac{\partial x}{\partial \kappa_j}(t) + \frac{\partial f}{\partial u}(x,u,w,t)\,\frac{\partial u}{\partial \kappa_j}(t) \\ \frac{\partial z}{\partial \kappa_j}(t) = \frac{\partial g_1}{\partial x}(x,u,w,t)\,\frac{\partial x}{\partial \kappa_j}(t) + \frac{\partial g_1}{\partial u}(x,u,w,t)\,\frac{\partial u}{\partial \kappa_j}(t) \\ \frac{\partial y}{\partial \kappa_j}(t) = \frac{\partial g_2}{\partial x}(x,u,w,t)\,\frac{\partial x}{\partial \kappa_j}(t) + \frac{\partial g_2}{\partial u}(x,u,w,t)\,\frac{\partial u}{\partial \kappa_j}(t) \end{cases}, \tag{24}$$

where κ_j is the j-th free design variable. As in practice the plant trajectories are only inspected on a finite horizon, the half-line $t \geq 0$ should be replaced with $t \in [0, T]$ everywhere in the text.

Nonlinear simulations can be performed using a general-purpose ordinary differential equation solver, like in the MATLAB function `SIM`. For LTI systems, however, it is computationally more efficient to use the classical discrete state-propagation approach as in the MATLAB function `LSIM`. This method is particularly appealing here because the simulation scenarios for a given plant in

the family only differ by input signals and, consequently, the dynamic equation

$$\dot{x} = Ax + B_1 w + B_2 u$$

needs only be discretized once to get the simulation dynamics

$$x_{k+1} = A^d x_k + B_1^d w_k + B_2^d u_k \,.$$

A reduction in execution time is then achieved since the data A^d, B_1^d and B_2^d can be recycled for each scenario and the rest of the computation amounts to simple matrix vector products. Notice further that the outlined procedure is amenable to parallel computing because scenarios are typically independent.

Another important question is how to build the extension sets A_z^e and Ω_P^e which determine the tangent program (22) and thereby the behavior of the nonsmooth algorithm. To construct A_z^e, different strategies are used for soft and hard constraints, see Figure 2. In the soft constraints case, A_z^e contains the set of active times plus some extra samples for which constraints are violated, represented in the figure by '$\times$' symbols. This is easily obtained by decimating samples provided by the numerical integrator. In the hard constraint case, A_z^e is built similarly, but includes also extrema satisfying the constraint envelope, represented by '$*$' symbols. The idea is to feed the tangent program with first-order information about ψ even during phase II. In return this helps preventing iterates getting stuck on the feasibility boundary. The set $\Omega_P^e(\kappa)$ includes active frequencies plus some extra nearly active frequencies, see Figure 3.

4 Applications

The simulations and computations for the case studies presented in this section have been performed with the Matlab environment running on a 2.4GHz Core 2 Quad processor with 4Gb RAM. The code has been developed essentially using Matlab, with Fortran being used for the CQP tangent problem (22) to minimize the main performance bottlenecks.

4.1 Tracking and decoupling control for a satellite launcher

In this first example, a tracking and decoupling controller is designed for the Brazilian satellite launcher vehicle VLS during the atmospheric flight phase. This case study has been initially investigated in Oliva and Leite Filho (2002). For the sake of completeness, the nonlinear equations for the vehicle are re-

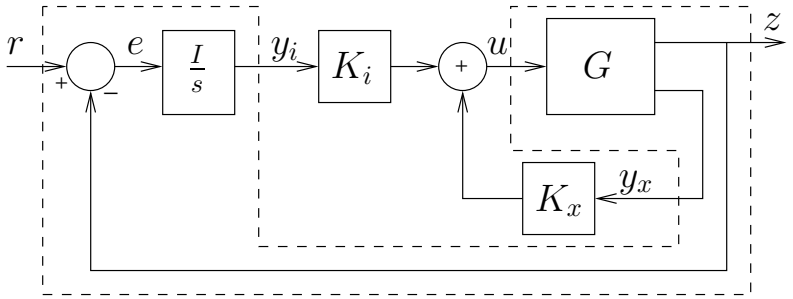

Fig. 4. closed-loop system block diagram representation

produced below:

$$
\begin{aligned}
\dot{v} =& Y_v v - g\sin(\theta)\sin(\phi) + g\cos(\theta)\sin(\psi)\cos(\phi) + Y_r r + pw + Y_{\beta_y}\beta_y \\
\dot{w} =& Z_w w - g\sin(\theta)\cos(\phi) - g\cos(\theta)\sin(\psi)\sin(\phi) + Z_q q - pv + Z_{\beta_z}\beta_z \\
\dot{p} =& L_p p + L_{qr} qr + L_{\beta_r}\beta_r \\
\dot{q} =& M_w w + M_q q + M_{pr} pr + M_{\beta_z}\beta_z \\
\dot{r} =& N_v v + N_r r + N_{pq} pq + N_{\beta_y}\beta_y \\
\dot{\theta} =& \cos(\phi)\sec(\psi)q - \sin(\phi)\sec(\psi)r, \\
\dot{\psi} =& \sin(\phi)q + \cos(\phi)r \\
\dot{\phi} =& p - \tan(\psi)\cos(\psi)q + \tan(\psi)\sin(\phi)r
\end{aligned}
$$

where v is the sideslip velocity, w is the z-body axis vehicle velocity, p, q, r are respectively the roll, pitch and yaw rates, and θ, ψ, ϕ are the attitude angles in pitch, yaw and roll, respectively. Variations of the x-body velocity need not be considered since there is no thrust control on the vehicle. The motion is controlled by deflections β_z, β_y and β_r of pitch, yaw and roll nozzle actuators, respectively. Coefficients of the nonlinear equations vary with flight time, but are supposed to be fixed here: $Y_v = -0.0162$, $Y_r = -87.9$, $Y_{\beta_y} = -10.87$, $Z_w = -0.0162$, $Z_q = 87.9$, $Z_{\beta_z} = 10.87$, $L_p = -0.0289$, $L_{qr} = 0$, $L_{\beta_r} = 25.89$ $M_w = 0.0022$, $M_q = 0.0148$, $M_{pr} = 0.8333$, $M_{\beta_z} = 4.08$, $N_v = -0.0022$, $N_r = 0.0151$, $N_{pr} = -0.9231$, $N_{\beta_y} = 4.08$.

The design objective is to synthesize a stabilizing controller where each attitude angle in the nonlinear model must track as closely as possible a step reference signal. The control law is a standard feedback with integral action in order to eliminate the steady-state tracking error, and is depicted in Figure 4, where

$$
z = \begin{bmatrix} \theta \\ \psi \\ \phi \end{bmatrix},\ r = \begin{bmatrix} \theta_{ref} \\ \psi_{ref} \\ \phi_{ref} \end{bmatrix},\ u = \begin{bmatrix} \beta_z \\ \beta_y \\ \beta_r \end{bmatrix},\ y_x^T = \begin{bmatrix} q\ \theta\ r\ \psi\ p\ \phi \end{bmatrix}^T.
$$

States w and v are not available for feedback. The controller variables to be

determined are the static gains $K_x \in \mathbb{R}^{3\times 6}$ and $K_i \in \mathbb{R}^{3\times 3}$ in Figure 4.

A difficulty of the problem is that the launcher has strongly coupled responses when performing angular maneuvers, and consequently the tracking controller must also achieve decoupling between the three channels. Available techniques to solve such tracking and decoupling problems include for instance the Linear Quadratic Regulator (LQR) and eigenstructure assignment techniques for linear systems investigated in Oliva and Leite Filho (2002). Nonetheless, this kind of problem fits nicely into the multi-scenario framework with appealing advantages. First, realistic time-domain performance criteria such as rise-time and overshoot, coupling amplitude limitations, as well as control amplitude and rate constraints are easily handled using the proposed nonsmooth method. This is in contrast with the existing approaches, where such performance specifications must be addressed indirectly by an iterative trial-and-error adjustment of auxiliary design variables such as modes and eigenvector structures for the eigenstructure assignment method or such as the quadratic weights Q and R with the LQR method. Moreover, the nonsmooth design technique does not assume full state measurement as is the case for the latter methods. Oliva and Leite Filho (2002) assume full state measurement and cancel out gains corresponding to unmeasured states afterwards which bears the risk of performance deterioration or even of a loss of stability. And last but not least, time-domain criteria can be imposed on the actual nonlinear model, thus leading to more realistic results.

The present tracking and decoupling problem is easily described by three distinct test scenarios being applied to the single model defined by the dashed box in Figure 4. Each test scenario consists in a unit step command applied to one of the reference inputs, while the other two are kept to zero. Altogether, one has 3 test inputs $w = r^1$, r^2 or r^3 described as follows:

$$r^1(t) = \begin{bmatrix} \sigma(t) \\ 0 \\ 0 \end{bmatrix}, \; r^2(t) = \begin{bmatrix} 0 \\ \sigma(t) \\ 0 \end{bmatrix}, \; r^3(t) = \begin{bmatrix} 0 \\ 0 \\ \sigma(t) \end{bmatrix}, \tag{25}$$

where $\sigma(t)$ stands for the unit step. Design specifications for each scenario are good tracking performance for the corresponding attitude angle, limited couplings with the other two angles, and control effort and rate limitations. All these performance criteria translate into time-domain envelope constraints as illustrated by dashed lines in Figures 5 to 7.

Exploiting the particular structure of the feedback configuration, it is readily established that control rate constraints can be turned into simple bounds on the integral gains. To see this, consider, for instance, a unit pitch step represented by the case $r(t) = r^1(t)$. With zero initial condition for the plant

states and integrators, the pitch control rate $\dot{\beta}_z(t)$ attains its largest amplitude at the initial instant $t = 0^+$:

$$\begin{aligned}\max_{t\geq 0} |\dot{\beta}_z(t)| = |\dot{\beta}_z(0^+)| &= |K_x \dot{y}_x(0^+) + K_i \dot{y}_i(0^+)| \\ &= |K_i(r(0^+) - z(0^+))| = |K_i r(0^+)| = |K_i^{1,1}|,\end{aligned}$$

where $K_i^{1,1}$ is the $(1,1)$ entry of K_i in Figure 4. Hence, the pitch control rate will be bounded by directly limiting the gain $K_i^{1,1}$ of the controller, since gains $K_i^{1,2}$ and $K_i^{1,3}$ are usually smaller. A similar reasoning applies to $K_i^{2,2}$ and $K_i^{3,3}$ for yaw and roll control rate limitations, respectively.

Additionally, internal stability and robustness specifications are captured by considering the linear closed-loop system obtained by the linearization of the original nonlinear equations. Firstly, the largest singular value of the complementary sensibility function $T := GK(I + GK)^{-1}$ is bounded in the high-frequency range in order to achieve robustness against unmodeled flexible modes. Secondly, the linear closed-loop system is enforced to be internally stable through the spectral abscissa criterion (11). The idea here is that achieving closed-loop stability with the linearized model is likely to increase the stability domain of the actual nonlinear closed-loop system, though it must be admitted that this can only be checked *a posteriori*.

Design specifications may be summarized as:

- good tracking performance: piecewise constant envelopes in the left column of Figure 5;
- coupled response limitation: upper and lower bounds in the right column of Figure 5;
- control effort limitation: see Figure 6, where only the controls presenting the largest amplitudes are depicted;
- control rate limitation: the constraints $|K_i^{1,1}| < 4$, $|K_i^{2,2}| < 4$ and $|K_i^{3,3}| < 1$ are enforced. See Figure 7, where only the controls with the largest rates have been depicted.
- robustness against high-frequency unmodeled dynamics: see Figure 8, $\overline{\sigma}(GK(I + GK)^{-1}) \leq 0.4,$ for $\omega \geq 100$ rad/s.
- internal stability for the linear closed-loop system.

In this study the spectral abscissa constraint was considered as a hard constraint while all other constraints were viewed as soft. Since nonlinear simulations are computationally more expensive than using `LSIM`, it seems to be a good strategy to perform an initial synthesis considering the linearized model also for time-domain constraints, and then to use that designed controller as the seed for a final synthesis with the nonlinear model. The nonsmooth algorithm finds a locally optimal solution for the initial linear problem after 493 iterations within 4.7 minutes of cputime. The locally optimal solution for

the final nonlinear problem is found after 62 iterations within 261 minutes of cputime. The final constraints violation falls below 2×10^{-4}, so all design specifications are achieved.

The final controller is given as

$$\left[\begin{array}{cc} K_i & K_x \end{array}\right] =$$

$$\left[\begin{array}{ccc|ccccccc} \mathbf{4.0002} & 0.62315 & -0.2718 & -1.3421 & -3.6319 & -0.23305 & -0.52236 & 0.14273 & 0.19974 \\ -0.60671 & \mathbf{3.8959} & -0.2111 & 0.23207 & 0.50359 & -1.3144 & -3.553 & 0.077188 & 0.22132 \\ 0.01865 & -0.044744 & \mathbf{1.0002} & -0.02392 & -0.011085 & -0.0008793 & 0.046795 & -0.41614 & -0.96903 \end{array}\right] . \tag{26}$$

The time-domain closed-loop responses with the nonsmooth controller (26) and the nonlinear model are shown in Figures 5 to 7, while Figure 8 gives the closed-loop frequency-domain response. Also depicted are the closed-loop responses for the LQR controller in Oliva and Leite Filho (2002). In the design fast tracking responses are sought for pitch and yaw steps, leading to an increase of both control effort and rate. The LQR controller in contrast exhibits unsatisfactory decoupling of nonlinear system responses. In conclusion it may be emphasized that the proposed technique solves the design specifications as posed in practice, without taking recourse to delicate tuning of auxiliary design parameters.

4.2 Reliable flight controller

In the next example, a reliable flight control system is designed for an F-16 aircraft performing high angle-of-attack maneuvers subject to wind gusts. This problem has been studied in Liao et al. (2002) from where the model data are borrowed. The primary design goal is to synthesize a stabilizing controller achieving tracking performances for the stability axis roll rate $\dot{\mu}_{rat}$, the angle-of-attack α and the sideslip angle β of the aircraft. The control system configuration is again that of Figure 4, with $z^T = [\,\dot{\mu}_{rat}\ \alpha\ \beta\,]$. All the aircraft states are assumed available for feedback:

$$x^T = y_x^T = \left[\begin{array}{cccccc} u & w & q & v & p & r \end{array}\right], \tag{27}$$

where p, q, r are respectively the roll, pitch and yaw rates, and v, w and u are the y, z and x-body axis velocities. The aircraft model also includes an exogenous disturbance w_g which represents vertical wind gusts. The control vector is given as

$$u^T = \left[\begin{array}{ccccc} \delta_{hr} & \delta_{hl} & \delta_{ar} & \delta_{al} & \delta_r \end{array}\right], \tag{28}$$

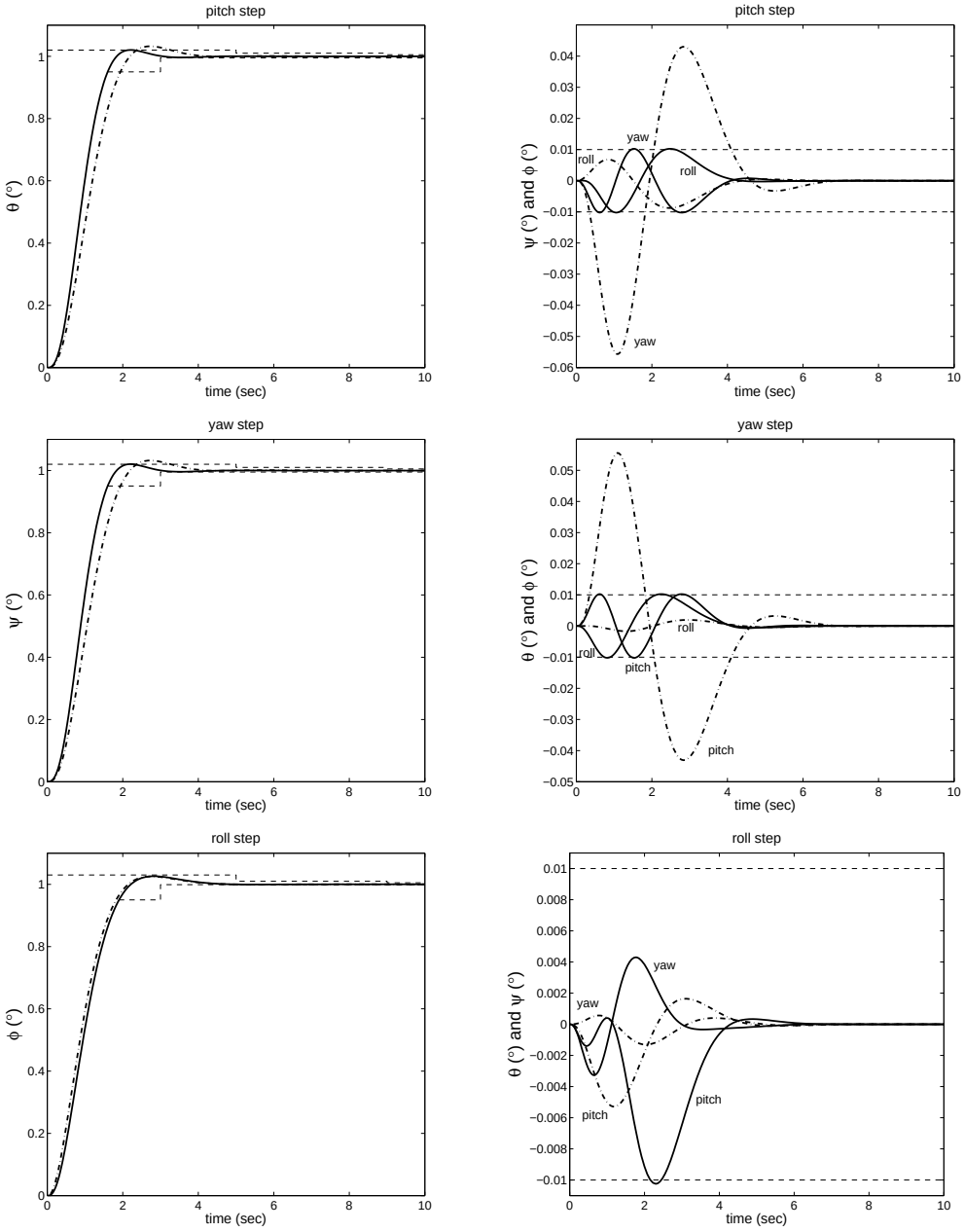

Fig. 5. Nonlinear step responses: nonsmooth (solid) and LQ (dash-dot) controllers

where δ_{hr}, δ_{hl}, δ_{ar}, δ_{al} and δ_r are the deflections of the right and left stabilators, the right and left ailerons and the rudder, respectively, which yields $K_x \in \mathbb{R}^{5\times 6}$ and $K_i \in \mathbb{R}^{5\times 3}$.

Given that the combat aircraft evolves in critical high angle-of-attack flight conditions, kinematics and inertial coupling phenomena become important and the control law must achieve substantial decoupling of the various channels. Additionally, the solution must guarantee closed-loop stability and satisfactory performance for any of the operational modes in table 1 in order to

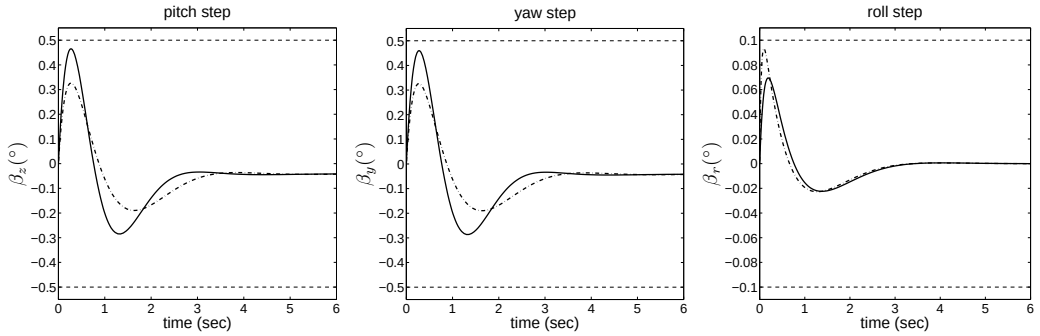

Fig. 6. Pitch, yaw and roll controls for a pitch, yaw and roll steps, respectively: nonsmooth (solid) and LQ (dash-dot) controllers

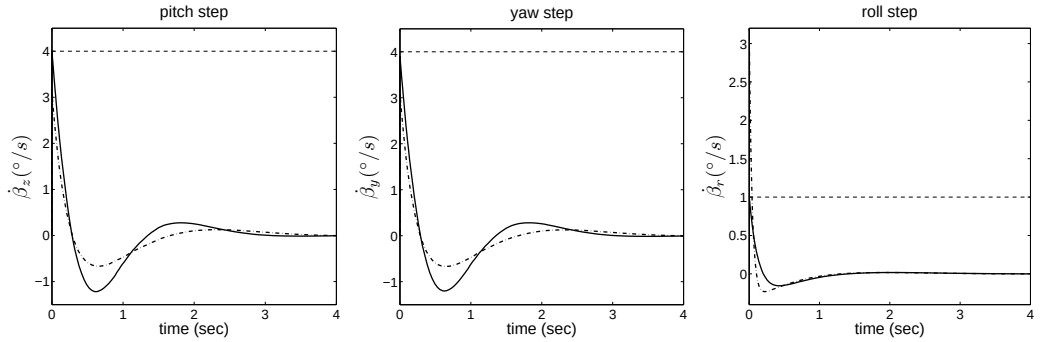

Fig. 7. Pitch, yaw and roll control rates for a pitch, yaw and roll steps, respectively: nonsmooth (solid) and LQ (dash-dot) controllers

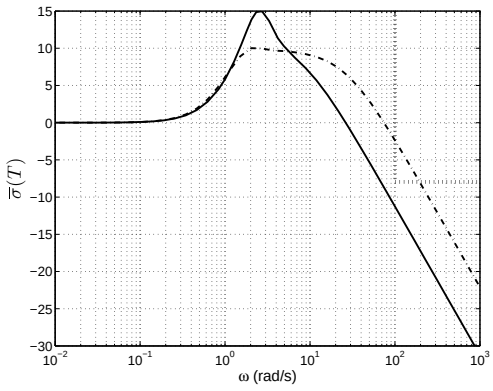

Fig. 8. Closed-loop complementary sensibility: nonsmooth (solid) and LQ (dash–dot) controllers

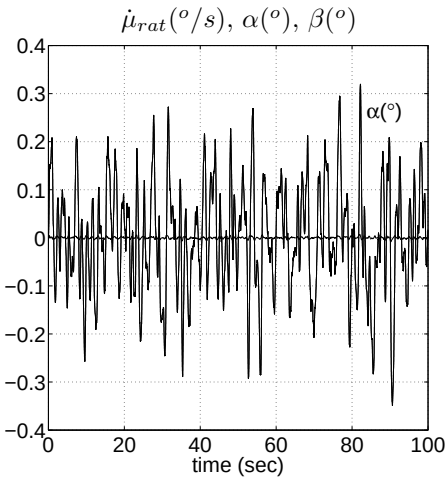

Fig. 9. Closed-loop system responses under vertical wind gusts

be a reliable controller. The linearized models P^0_{lin} and P^3_{lin} are given in Liao et al. (2002).

Table 1
Nominal and failure modes for the F-16 aircraft

mode type	description
nominal operation:	$P(s) = P^0_{\text{lin}}(s)$,
failure of the right stabilator:	$P(s) = P^0_{\text{lin}}(s) \times \text{diag}(0,1,1,1,1)$,
failure of the left stabilator:	$P(s) = P^0_{\text{lin}}(s) \times \text{diag}(1,0,1,1,1)$,
failure of the right aileron:	$P(s) = P^0_{\text{lin}}(s) \times \text{diag}(1,1,0,1,1)$,
failure of the left aileron:	$P(s) = P^0_{\text{lin}}(s) \times \text{diag}(1,1,1,0,1)$,
75 % impairment of the stabilators:	$P(s) = P^3_{\text{lin}}(s)$.

Note that the controller must achieve adequate performance not only in the nominal mode, but also when any of the failures in Table 1 occurs. This leads to 3 scenarios for each mode in order to assess tracking and decoupling properties for $\dot{\mu}_{rat}$, α and β, leading to a total of 18 scenarios. Clearly this a complicated problem involving multiple plant modes as well as multiple test inputs. It is readily incorporated in the general framework of section 2. In order to guarantee stability and robustness against the possible failures, a spectral abscissa constraint (11) is introduced for each closed-loop system associated with the various operational modes in table 1.

Another design specification is satisfactory vertical wind gust load alleviation in the nominal mode. Wind gusts are modeled as the output of a Dryden filter (Aouf et al., 2000)

$$G_w(s) = 2.5046 \frac{s + 0.1517}{(s + 0.2628)^2},$$

driven by a unit-intensity zero-mean gaussian white-noise $\hat{n}$, so that $w_g = G_w \hat{n}$. The RMS vertical gust velocity is 5 m/s. Having connected the filter $G_w(s)$ to the disturbance input w_g of the aircraft model, a bound constraint is imposed on the H_2 norm of the nominal transfer from $\hat{n}$ to the regulated output z.

Control rates are limited to $15^\circ/s$ for a unit step, using once again the strategy of limiting the integral gains K_i. In this application all entries of K_i have been constraint to $|K_i^{k,l}| \leq 15$, $k = 1 \ldots 5, l = 1 \ldots 3$.

The nonsmooth technique finds a locally optimal solution after 621 iterations corresponding to 23 minutes of cputime. The final controller found by the nonsmooth algorithm is

$$\left[\begin{array}{cc} K_i & K_x \end{array}\right] =$$

$$\left[\begin{array}{ccc|ccc|ccc} -0.5983 & -3.63 & -0.126 & 0.2544 & 0.8396 & 20.9 & -0.01052 & 2.692 & -2.703 \\ -0.5983 & -3.63 & -0.126 & 0.2544 & 0.8396 & 20.9 & -0.01052 & 2.692 & -2.703 \\ \mathbf{-15} & -0.2078 & 1.916 & 0.5747 & 0.008635 & 1.167 & -0.7743 & 58.95 & 6.538 \\ \mathbf{15} & 0.2078 & -1.916 & -0.5747 & -0.008635 & -1.167 & 0.7743 & -58.95 & -6.538 \\ -5.461 & -0.08635 & \mathbf{15} & 0.7486 & 0.04805 & 0.6288 & -2.577 & 14.59 & 99.34 \end{array}\right] . \tag{29}$$

Figures 10 to 12 show the closed-loop responses with the designed controller (29) for each of the six operational modes, together with the closed-loop responses under 25% and 50% impairment of the stabilators. The synthesized controller guarantees good closed-loop nominal behavior, but also closed-loop stability with contained performance deterioration in the event of extreme failures, indicating that a reliable design has been obtained. The worst performance degradation corresponds to the angle-of-attack tracking response under 75% impairment of the stabilators, a rather critical situation, see the central plot in Figure 11. As expected, closed-loop responses remain satisfactory under 25% and 50% impairment of the stabilators, even though these scenarios have not been explicitly included in the synthesis requirements. As can be seen in Figure 9, the nonsmooth control also attains acceptable wind gust load alleviation despite the severe gain constraint. Finally, all spectral abscissa constraints were formulated as hard constraints and consequently are met at the optimum, which means that the closed-loop system remains stable in all operational modes.

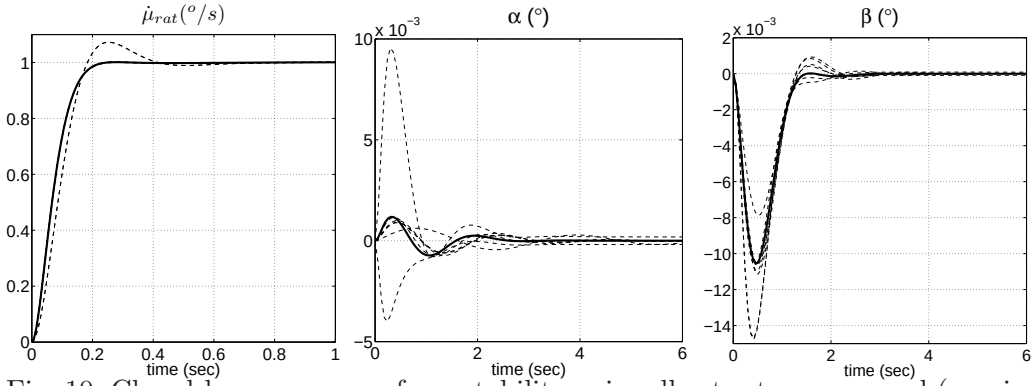

Fig. 10. Closed-loop responses for a stability-axis roll rate step command (nominal: solid)

5 Conclusion

In practical applications designers prefer feedback controllers that perform well not only in the nominal operational mode, but possibly for a collection

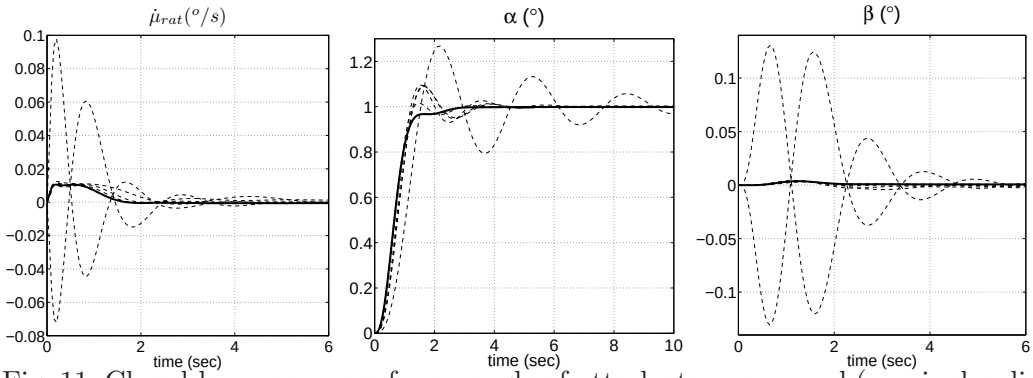

Fig. 11. Closed-loop responses for an angle-of-attack step command (nominal: solid)

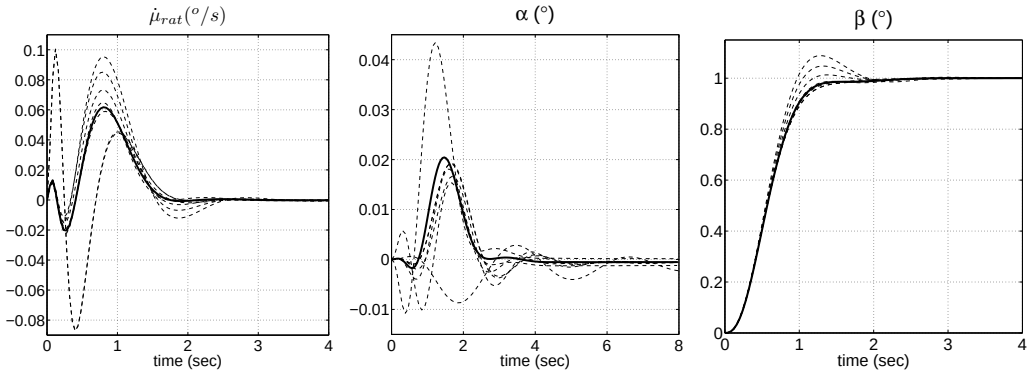

Fig. 12. Closed-loop responses for a sideslip angle step command (nominal: solid)

of scenarios representing performance specifications in time-domain. Moreover, the controller should also meet frequency-domain specifications which are routinely required in closed-loop. A nonsmooth optimization technique have been presented and discussed which allows to address this challenging class of synthesis problems quite successfully. This leads to a highly flexible design tool which allows to go beyond what can be achieved with BMI- and LMI-techniques or with simple tuning heuristics. As a by-product, the proposed approach also furnishes a detailed analysis of each closed-loop performance specification, revealing possible design difficulties. Multi-scenario time-domain design under frequency-domain constraints is a very challenging problem for which only locally optimal solutions can be computed. Despite this principled obstacle, it is shown by the case studies that the local convergence theory on which the proposed method is based produces good results in practice and can with some right be considered an efficient design tool.

Acknowledgement

This research was supported by grants from Fondation de Recherche pour l'Aéronautique et l'Espace (FNRAE) under contract *Survol*, and by Agence Nationale de la Recherche (ANR) under contract *Controvert*.

References

N. Aouf, B. Boulet, and R. Botez. H_2 and H_∞-optimal gust load alleviation for a flexible aircraft. *American Control Conference, 2000. Proceedings of the 2000*, 3:1872–1876 vol.3, 2000.

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

Pierre Apkarian and Dominikus Noll. Controller design via nonsmooth multidirectional search. *SIAM J. Control Optim.*, 44(6):1923–1949, 2006.

P. Apkarian, D. Noll, and A. Rondepierre. Mixed H_2/H_∞ control via nonsmooth optimization. *SIAM J. on Control and Optimization*, 47(3):1516–1546, 2008.

V. Bompart, P. Apkarian, and D. Noll. Nonsmooth techniques for stabilizing linear systems. *American Control Conference, 2007. ACC '07*, pages 1245–1250, July 2007.

V. Bompart, P. Apkarian, and D. Noll. Control design in the time- and frequency-domain using nonsmooth techniques. *Systems and Control Letters*, 57(3):271–282, 2008.

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

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

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

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

J.E. Dennis, Jr. and R. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM's Classics in Applied Mathematics. SIAM, 1996.

F. Liao, J. L. Wang, and G.-H. Yang. Reliable robust flight tracking control: An LMI approach. *IEEE Trans. Control Sys. Tech.*, 10:76–89, 2002.

P. Mäkilä. Multiple models, multiplicative noise and linear quadratic control-algorithm aspects. *Int. J. Control*, 54(4):921–941, Oct. 1991.

A. P. Oliva and W. C. Leite Filho. Eigenstructure versus optimal control for decoupling. *Control Engineering Practice*, 10:1059–1079, 2002.

Y. Piguet, U. Holmberg, and R. Longchamp. A minimax approach for multi-objective controller design using multiple models. *Int. J. Control*, 72(7): 716–726, may 1999.

E. Polak and S. E. Salcudean. On the design of linear multivariable feedback systems via constrained nondifferentiable optimization in H_∞ spaces. *Automatic Control, IEEE Transactions on*, 34(3):268–276, Mar 1989.

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

G. Pujol, J. Rodellar, J. Rossell, and F. Pozo. Decentralised reliable guaran-

teed cost control of uncertain systems: an LMI design. *IET Control Theory & Applications*, 1:779–785, 2007.

A. M. Simões, P. Apkarian, and D. Noll. A nonsmooth progress function for frequency shaping control design. *IET Control Theory & Applications*, 2 (4):323–336, April 2008.

K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.