Structured flight control law design using non-smooth optimization

M. Gabarrou * **D. Alazard** ** **D. Noll** ***

* *Université de Toulouse - I.S.A.E and U.P.S, France (e-mail: marion.gabarrou@math.univ-toulouse.fr)*
** *Université de Toulouse - I.S.A.E, France (e-mail: alazard@isae.fr)*
*** *Université de Toulouse - U.P.S, France (e-mail: dominikus.noll@math.univ-toulouse.fr)*

Abstract: We extend the concept of bundle methods to address non-convex and nonsmooth optimization problems arising in the design of a feedback control law for the longitudinal flight control of a civil aircraft. Our novel approach has two advantages. It allows to handle the specific structure of the control law directly, and we can express control law specifications directly as band-limited frequency-domain mathematical programming constraints.

Keywords: optimization, structured controller, flight control, H_∞, multi-objective.

1. INTRODUCTION

Practical solutions for the control of aerospace vehicles are based on architectures involving multi-level control loops. The efficiency of these architectures, referred to as *guidance, navigation and control laws* (GNC), was established in several applications of practical interest (Tischler (1996); Stevens and Lewis (1992)). The inner loop (called the control loop) aims at controlling the short term dynamics, that is, the high frequency behaviour. In contrast, the outer loop (called the guidance loop) serves to control the long term dynamics, that is, the low frequency behaviour. The method is therefore based on frequency decoupling between the rotational (short term) dynamics and the translational (long term) dynamics. In addition, it has to take into account that the system is under-actuated. In an aircraft the elevator is used to control the longitudinal attitude (rotation) and the vertical speed (translation). Here the inner loop is also called the flight control loop and the outer loop is the autopilot loop. The GNC architecture is also very interesting from the point of view of the hardware implementation, because it facilitates transitions between the manual mode and the auto-pilot mode.

For the inner flight control loop practitioners prefer simple controller structures in order to address issues like saturation, interpolation of the controller according to flight operating conditions, and feedforward compensation adapted to the various aircraft configurations. A structure often found in practice combines proportional or proportional/integral feedback with a low-pass filter on the control signal to cut-off high frequency components (roll-off) and to improve robustness with respect to unmodeled dynamics, caused mainly by flexible structural modes (Alazard (2002)). The tuning of the controller gains and the filter has then to respect the trade-off between low frequency performance and high frequency roll-off. Frequency-domain approaches, like H_∞ design and μ-extension, are as a rule suitable to achieve this trade-off by introducing suitable frequency weights on closed-loop sensitivity functions. But they have serious drawbacks if currently available software is used:

- They only provide full-order unstructured controllers. Recent research has striven on overcoming this limitation, for instance Geromel et al. (1998) aims at reduced order controllers using controller order reduction (Madelaine (1998)). A new technique which allows to design arbitrarily structured control laws is Apkarian and Noll (2006).
- A second aspect is that the H_∞ norm is not fully adapted to the flight control context, because the objective has to match frequency domain templates only in a limited frequency range. There is no need to control the low frequency behavior by the flight control loop, because this part will be controlled later by the autopilot loop. The first approach to synthesize control laws with H_∞ norms on several frequency bands was Apkarian and Noll (2007).

For these reasons the design of longitudinal flight control law is a good example to illustrate the interest for new optimization tools, allowing to take not only a given structure of the control law, but also a limited frequency range for specifications into account.

In the following section, our non-smooth optimization method is presented. In section 2.2 we specialize it to a multi-objective H_∞/H_∞ control problem. Section 3 presents the longitudinal flight model, the control law structure, the performance specifications and the results obtained by the non-smooth optimization approach.

2. NON-SMOOTH AND CONSTRAINED OPTIMIZATION

2.1 Non-smooth algorithm

We investigate the problem :

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz and potentially nonsmooth. The goal is to compute a locally optimal solution $\bar{x}$ in the sense that the first order necessary optimality condition

$$0 \in \partial f(\bar{x}) \tag{2}$$

is satisfied, $\partial f(x)$ being the Clarke subdifferential of f at x. Our algorithm uses a bundle method with proximity control (Noll et al. (2008)), which solves (1) in the sense that for an arbitrary starting point, every accumulation point of the sequence of serious iterates generated by the algorithm is a critical point of f, i.e. satisfies (2). Traditional bundle methods are conceived for convex nonsmooth programs, so we have to extend this concept to address non-convex objectives. We use a local model $\Phi(.,x)$ of f in a neighbourhood of the current iterate x, which is a non-smooth analogue of the Taylor expansion of f at x:

$$\Phi(y,x) = \phi(y,x) + \tfrac{1}{2}(y-x)^T Q(x)(y-x),$$

where $\phi(y,x)$ is the first-order part of the model. We assume that $\phi(.,x)$ is convex for fixed x, but generally non-smooth. In contrast, the second order term $\frac{1}{2}(y-x)^T Q(x)(y-x)$ is smooth, but need not be convex.

The interplay between f and its local model Φ is used to create descent step for f away from x by a proximity control mechanism. We do not use $\Phi(.,x)$ directly to generate search steps, because this may be too costly. Instead we build a so-called working model $\Phi_k(.,x)$, which has the form

$$\Phi_k(y,x) = \phi_k(y,x) + \tfrac{1}{2}(y-x)^T Q(x)(y-x),$$

where $\phi_k(.,x)$ is an approximation of the ideal first-order model $\Phi(.,x)$. Following the general philosophy of the cutting plane method, $\phi_k(y,x)$ is improved iteratively using cutting planes and aggregation. The role of the inner loop is to find a new serious step. The 2^{nd}-order part is kept fixed in inner loop, and is only updated between serious steps. At inner loop counter k, we compute a trial step y^{k+1} by solving the tangent program with proximity control

$$\min_{y\in\mathbb{R}^n} \Phi_k(y,x) + \tfrac{\tau_k}{2}\left\|y-x\right\|^2, \tag{3}$$

where τ_k is the proximity control parameter. Here y^{k+1} is a local solution in the sense that

$$0 \in \partial_1\phi_k(y^{k+1},x) + (Q(x)+\tau_k I)(y^{k+1}-x).$$

If y^{k+1} gives sufficient decrease below $f(x)$, it becomes the new iterate $x^+ = y^{k+1}$. In that case y^{k+1} is called a serious step. Otherwise y^{k+1} is called a null step. In this case we keep x, but use the information transmitted by y^{k+1} to improve the first order part $\phi_k(.,x) \leftarrow \phi_{k+1}(.,x)$ of the working model. We then update the proximity control parameter $\tau_k \leftarrow \tau_{k+1}$ and rerun the tangent program to obtain a better trial step y^{k+2}. The rationale of the inner loop is that after some updates $k \leftarrow k+1$, y^{k+1} will improve over the current x and become the new iterate x^+. During these on-goings the second order term is kept invariant, and changed only between serious step. Its rationale is to give our method an option to converge superlinearly (in the serious steps) if f has hidden smoothness properties, as is often the case in application.

In order to decide whether y^{k+1} can be accepted to become the new serious step, we fix a constant $0 < \gamma < 1$ and compute the quotient

$$\rho_k = \frac{f(x) - f(y^{k+1})}{f(x) - \Phi_k(y^{k+1},x)}$$

which reflects the agreement between f and $\Phi_k(.,x)$ at y^{k+1}. If $\rho_k \geq \gamma$, then y^{k+1} is accepted as a serious step $x^+ = y^{k+1}$, otherwise is is rejected, referred to as a null step. Here we have to improve $\phi_k(.,x)$ into a better first order model $\phi_{k+1}(.,x)$. But it may also become necessary to increase the proximity control parameter τ_k. In order to decide when this has to happen, we introduce a second quotient

$$\tilde{\rho_k} = \frac{f(x) - \Phi(y^{k+1},x)}{f(x) - \Phi_k(y^{k+1},x)}$$

which reflects the agreement between $\Phi(.,x)$ and $\Phi_k(.,x)$ at y^{k+1}. We fix a constant $\gamma < \tilde{\gamma} < 1$ and we say that $\Phi_k(.,x)$ is far from $\Phi(.,x)$ (at y^{k+1}) if $\tilde{\rho_k} < \tilde{\gamma}$. Let us consider the case $\rho_k < \gamma$ and $\tilde{\rho_k} > \tilde{\gamma}$, i.e. $\Phi_k(y^{k+1},x)$ is far from $f(y^{k+1})$, but at the same time near $\Phi(y^{k+1},x)$. Using aggregation and cutting plane alone will now fail because it will only drive the working model $\Phi_{k+1}(.,x)$ even closer to the ideal model (at y^{k+2}), but will not suffice to make progress. Namely in this case $\Phi(y^{k+1},x)$ is by itself too far from $f(y^{k+1})$, and this phenomenon is likely to persist if $\left\|y^{k+2}-x\right\| \approx \left\|y^{k+1}-x\right\|$. To force $\Phi(y^{k+2},x)$ closer to $f(y^{k+2})$, i.e., to do smaller steps, we have to tighten proximity control, that is, increase τ_k. On the other hand, if $\rho_k < \gamma$ and also $\tilde{\rho_k} < \tilde{\gamma}$, then nothing seems decided as yet, so here we keep the proximity control parameter unchanged and rely on cutting planes and aggregation.
The question which remains to clarify is the construction of the new working model ϕ_{k+1} after a null step. The first element is to guarantee exactness, that is $\phi_{k+1}(x,x) = f(x)$ and $\partial_1\phi_{k+1}(x,x) \subset \partial f(x)$. To do this we pick an element $g(x) \in \partial f(x)$, define the affine function $m(y) = f(x) + g(x)^T(y-x)$ and assure that $\phi_{k+1}(y,x) \geq m(y)$. Then $\partial_1\phi_{k+1}(x,x) \subset \partial f(x)$ is guaranteed.

The cutting planes technique is discussed next. In the case of a null step y^{k+1} we pick $g_{k+1} \in \partial_1\phi(y^{k+1},x)$. By convexity of $\phi(.,x)$

$$g_{k+1}^T(y-y^{k+1}) \leq \phi(y,x) - \phi(y^{k+1},x).$$

In other words, $m_{k+1}(y) = a_{k+1} + g_{k+1}^T(y-x)$ with $a_{k+1} = \phi(y^{k+1},x) + g_{k+1}^T(x-y^{k+1})$ is an affine support function of $\phi(.,x)$ at y^{k+1}. It is called the cutting plane. The effect of including the cutting plane m_{k+1} as an affine minorant of the next convex working model $\phi_{k+1}(.,x)$ is that the unsuccessful trial step y^{k+1} is cut away at the next trial $k+1$, paving the way for a better y^{k+2} to come.

The last element to be discussed is known as aggregation. It is used to recycle some of the information stored in ϕ_k for the new working model ϕ_{k+1}, and its essence is to prevent overflow of the algorithm. The optimality condition for program (3) implies

$$0 \in \partial_1\phi_k(y^{k+1},x) + (Q(x)+\tau_k I)(y^{k+1}-x).$$

In other words

$$g_{k+1}^* = (Q(x)+\tau_k I)(x-y^{k+1}) \in \partial_1\phi_k(y^{k+1},x).$$

That means $m_{k+1}^*(y) = a_{k+1}^* + g_{k+1}^{*T}(y-x)$ with $a_{k+1}^* = \phi_k(y^{k+1},x) + g_{k+1}^{*T}(x-y^{k+1})$, is an affine support function

of $\phi_k(.,x)$ at y^{k+1}, called the aggregate plane. Aggregation is a clever substitute for storing the full sequence of previous models $\phi_k \leq \phi_{k+1} \rightarrow \phi$, as this would lead to working models of increasing complexity. Naturally, the latter would turn out too expensive.

Altogether we have identified the following list of conditions for which the convex working model has to satisfy

- Exactness $m(.) \leq \phi_{k+1}(.,x) \leq \phi(.,x)$.
- Cutting plane $m_{k+1}(.) \leq \phi_{k+1}(.,x)$.
- Aggregation $m^*_{k+1}(.) \leq \phi_{k+1}(.,x)$.

These conditions are sufficient to ensure convergence of the method.

Parameters $0 < \gamma < \tilde{\gamma} < \Gamma < 1$, and $0 < q < \infty$
1: **Initialize outer loop**. Choose initial guess x^1 and an initial matrix $Q_1 = Q_1^T$ with $-qI \preceq Q_1 \preceq qI$. Then fix memory control parameter $\tau_1^\#$ such that $Q_1 + \tau_1^\# I \succ 0$. Put $j = 1$.
2: **Stopping test**. At outer loop counter j, stop if $0 \in \partial f(x^j)$. Otherwise goto inner loop.
3: **Initialize inner loop**. Put inner loop counter k=1 and initialize the parameter τ using memory element, i.e., $\tau_1 = \tau_j^\#$. Choose initial convex working model $\phi_1(.,x^j)$, and let $\Phi_1(y,x^j) = \phi_1(y,x^j) + \frac{1}{2}(y-x^j)^T Q_j (y-x^j)$.
4: **Trial step generation**. At inner loop counter k solve tangent program

$$\min_{y \in \mathbb{R}^n} \Phi_k(y,x^j) + \tfrac{\tau_k}{2} \left\| y - x^j \right\|^2$$

Solution is the new trial step y^{k+1}.
5: **Acceptance test**. Check whether

$$\rho_k = \frac{f(x^j) - f(y^{k+1})}{f(x^j) - \Phi_k(y^{k+1}, x^j)} \geq \gamma$$

If this is the case, put $x^{j+1} = y^{k+1}$ (serious step), quit inner loop and goto step 8. On the other hand, if this is not the case (null step), continue inner loop with step 6.
6: **Update proximity parameter**. Compute second control parameter :

$$\tilde{\rho_k} = \frac{f(x^j) - \Phi(y^{k+1}, x^j)}{f(x^j) - \Phi_k(y^{k+1}, x^j)}$$

Then put

$$\tau_{k+1} = \begin{cases} \tau_k & \text{if } \tilde{\rho}_k < \tilde{\gamma} \\ 2\tau_k & \text{if } \tilde{\rho}_k \geq \tilde{\gamma} \end{cases}$$

7: **Update working model**. Build new convex working model $\phi_{k+1}(.,x^j)$ by respecting the three rules (exactness, cutting plane, aggregation) based on null step y^{k+1}. Then increase inner loop counter k and continue inner loop with step 4.
8: **Update Q_j and memory element**. Update matrix $Q_j \rightarrow Q_{j+1}$ respecting $Q_{j+1} = Q_{j+1}^T$ and $-qI \preceq Q_{j+1} \preceq qI$. Then store new memory element

$$\tau_{j+1}^\# = \begin{cases} \tau_k & \text{if } \gamma \leq \rho_k < \Gamma \\ \frac{\tau_k}{2} & \text{if } \rho_k \geq \Gamma \end{cases}$$

Increase $\tau_{j+1}^\#$ if necessary to ensure $Q_{j+1} + \tau_{j+1}^\# I \succ 0$. Increase outer loop counter j by 1 et and loop back to step 2.

2.2 H_∞ optimization by non-smooth algorithm

Setting. We consider a plant P described by the following state space realization :

$$P : \begin{bmatrix} \dot{x}_P \\ z_1 \\ z_2 \\ y_P \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 & B \\ C_1 & D_{11} & D_{12} & D_{1u} \\ C_2 & D_{21} & D_{22} & D_{2u} \\ C & D_{y1} & D_{y2} & D_{yu} \end{bmatrix} \begin{bmatrix} x_P \\ w_1 \\ w_2 \\ u \end{bmatrix},$$

where $x_P \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ the control, $y_P \in \mathbb{R}^p$ the output, and where $w_1 \in \mathbb{R}^{m_1} \rightarrow z_1 \in \mathbb{R}^{p_1}$,$w_2 \in \mathbb{R}^{m_2} \rightarrow z_2 \in \mathbb{R}^{p_2}$ are two (concurring) H_∞ performance channels :

- $T_{z_i w_i}$ is the open-loop transfer function $w_i \rightarrow z_i$. A state space realization of $T_{z_i w_i}$ is obtained by deleting the w_i column and the z_i line in P.
- $\|G\|_{\infty,\Omega} = \sup_{\omega \in \Omega} \bar{\sigma}[G(j\omega)]$, $\Omega \subset \mathbb{R}_+$

where $\bar{\sigma}[G(j\omega)]$ is the maximum singular value of a transfer matrix $G(j\omega)$. We seek an output feedback controller $K : \begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \begin{bmatrix} x_K \\ y_P \end{bmatrix}$, such that the closed loop system, obtained by substituting K into P, satisfies the following properties :

(1) Internal stability : K stabilizes P exponentially in closed loop.
(2) Fixed H_∞ performance : $T_{z_2 w_2}(K)$ has a pre-specified performance level $\|T_{z_2 w_2}(K)\|_{\infty,\Omega} \leq \gamma$
(3) Optimal H_∞ performance : $\|T_{z_1 w_1}(K)\|_{\infty,\Omega}$ is minimized among all K satisfying (1) and (2).
(4) K must satisfy structural constraints.

To address (4) we introduce a parametrization

$$x \in \mathbb{R}^q \rightarrow K(x) = \begin{bmatrix} A_K(x) & B_K(x) \\ C_K(x) & D_K(x) \end{bmatrix},$$

where x denotes the unknown parameters in K (and not the plant state, for which we reserve the notation x_P). The mixed H_∞/H_∞ synthesis problem is now as follows:

$$\begin{array}{ll} \text{minimize} & f(x) := \|T_{z_1 w_1}(K(x))\|^2_{\infty,\Omega} \\ \text{subject to} & g(x) := \|T_{z_2 w_2}(K(x))\|^2_{\infty,\Omega} \leq \gamma^2 \end{array} \quad (4)$$

To satisfy the closed loop stability criterion, an additional constraint on the spectral abscissa is imposed, i.e. $\alpha(A(K)) < 0$ where $A(K)$ is the closed loop dynamical matrix (Burke and Overton (1994)). Previous work on mixed H_2/H_∞ control design using optimization is for instance Apkarian et al. (2008).

H_∞ norm. Objective f and constraint g have the form $\|.\|^2_{\infty,\Omega} \circ T_{w \rightarrow z}$, and can be represented as

$$h(x) = \sup_{\omega \in \Omega} \lambda_1[H(x,\omega)] = \sup_{\omega \in \Omega} h(x,\omega)$$

where $H : \mathbb{R}^q \times \Omega \rightarrow \mathbb{H}^m$ is of class C^2 in x and jointly continuous in (x,ω), and where λ_1 is the maximum eigenvalue of the Hermitian matrix $H(x,\omega)$. Simply take $H(x,\omega) = T_{z_i w_i}(K(x), j\omega) T_{z_i w_i}(K(x), j\omega)^H$ and $m = n_{z_i}$, then $h(x)$ is nonsmooth with two possible sources of nonsmoothness, the infinite maximum, and the maximum eigenvalue function. To use our algorithm, we have to compute the Clarke subdifferential of h.

Lemma 1. Let $K(x)$ be closed-loop stabilizing. Then $h(x) = \|T(K(x))\|^2_{\infty,\Omega} < \infty$, and the set of active frequen-

cies at x defined as $\mathcal{I}(x)=\{\omega \in \Omega : h(x) = h(x,\omega)\}$ is either finite or equals Ω.

Lemma 2. Suppose $K(x)$ is closed-loop stabilizing and $\mathcal{I}(x)$ finite. Then the Clarke subdifferential $\partial h(x)$ is

$$\mathrm{co}\left\{\mathrm{v}\in\mathbb{C}^{\mathrm{q}} : \mathrm{v_i} = \left\langle \mathrm{Q(x,\omega)Y_\omega Q(x,\omega)^H}, \partial \mathrm{H(x,\omega)}/\partial \mathrm{x_i}\right\rangle,\right.$$
$$\left. i\in\{1,...,q\}, Y_\omega \in \mathbb{H}^p, \mathrm{Tr}(Y_\omega)=1, Y_\omega \succeq 0, \omega\in\mathcal{I}(x)\right\}.$$

Here the columns of the matrix $Q(x,\omega)$ form an orthonormal basis of the eigenspace of $H(x,\omega)$ associated with its maximum eigenvalue $\lambda_1(H(x,\omega))$ of multiplicity p.

Optimality conditions. We address program (4) by introducing a progress function :

$$F(y,x) = \max\{f(y)-f(x)-\mu\left[g(x)-\gamma^2\right]_+ ; \quad (5)$$
$$\left[g(y)-\gamma^2\right]-\left[g(x)-\gamma^2\right]_+\}$$

where $\mu>0$ is fixed and $g(x)_+ = \max(g(x),0)$.

Lemma 3. Suppose $0\in\partial_1 F(\bar{x};\bar{x})$ for some $\bar{x}\in\mathbb{R}^n$. Then we have the following possibilities :

(1) Either $g(\bar{x})>\gamma^2$, then $\bar{x}$ is a critical point of g, called a critical point of constraint violation.
(2) Or $g(\bar{x})\le\gamma^2$, then $\bar{x}$ satisfies the F. John necessary optimality conditions for program (4). In addition, there are two sub-cases:
   (a) Either $\bar{x}$ is a Karush-Kuhn-Tucker point of (4).
   (b) Or $\bar{x}$ fails to be Karush-Kuhn-Tucker point. The latter could only happen when $g(\bar{x})=\gamma^2$ and at the same time $0\in\partial g(\bar{x})$.

First local model. We need a local model for F in a neighbourhood of the current iterate x. An approximation $\tilde{h}$ of h in a neighbourhood of x can be obtained by linearizing the operator $y\mapsto H(y,\omega)$ around x.

$$\tilde{h}(y;x) = \max_{\omega\in\Omega}\lambda_1[H(x,\omega)+H'(x,\omega)(y-x)]$$
$$= \max_{\omega\in\Omega}\max_{Z\in\mathcal{C}}\langle Z, H(x,\omega)+H'(x,\omega)(y-x)\rangle,$$

where $\langle A,B\rangle=\mathrm{Tr}(A^HB)$ is the dot product of $\mathcal{M}_m(\mathbb{C})$, $H'(x,\omega)$ is the derivative of $H(.,\omega)$ with respect to x, and where $\mathcal{C}=\{Z\in\mathbb{H}^m : Z\succeq 0,\ \mathrm{Tr}(Z)=1\}$. By Taylor's theorem, we expect $\tilde{h}(.;x)$ to be a good approximation of h in a neighbourhood of x. Notice that $\tilde{h}(x,x)=h(x)$. Altogether we have the following approximation of $F(.,x)$

$$\tilde{F}(y,x) = \max\{\tilde{f}(y,x)-f(x)-\mu\left[g(x)-\gamma^2\right]_+ ; \quad (6)$$
$$\left[\tilde{g}(y,x)-\gamma^2\right]-\left[g(x)-\gamma^2\right]_+\}.$$

Lemma 4. Let $B\subset\mathbb{R}^n$ be a bounded set. Then there exists $L>0$ such that $\forall\, x,y\in B$

$$|F(y;x)-\tilde{F}(y;x)|\le L\|y-x\|^2.$$

It is convenient to represent the local model differently. Let $H_i(x,\omega)=T_{z_iw_i}(K(x),j\omega)T_{z_iw_i}(K(x),j\omega)^H$, and

$$\alpha_1(\omega,Z)=\langle Z,H_1(x,\omega)\rangle - f(x)-\mu\left[g(x)-\gamma^2\right]_+\in\mathbb{R}$$
$$\phi_1(\omega,Z)=H_1'(x,\omega)^HZ\in\mathbb{R}^{n_{z_1}}$$
$$\alpha_2(\omega,Z)=\langle Z,H_2(x,\omega)\rangle - \gamma^2-\left[g(x)-\gamma^2\right]_+\in\mathbb{R}$$
$$\phi_2(\omega,Z)=H_2'(x,\omega)^HZ\in\mathbb{R}^{n_{z_2}}$$

In this way the right and left hand branch of $\tilde{F}(y;x)$ may be written as the envelopes of cutting planes

$$\tilde{f}(y,x)-f(x)-\mu\left[g(x)-\gamma^2\right]_+ =$$
$$\sup_{\omega\in\Omega}\sup_{Z\in\mathcal{C}}\alpha_1(\omega,Z)+\phi_1(\omega,Z)^T(y-x),$$
$$[\tilde{g}(y;x)-\gamma^2]-[g(x)-\gamma^2]_+ =$$
$$\sup_{\omega\in\Omega}\sup_{Z\in\mathcal{C}}\alpha_2(\omega,Z)+\phi_2(\omega,Z)^T(y-x).$$

We can introduce

$$\mathcal{G}=\mathrm{co}(\{(\alpha_1(\omega,\mathrm{Z}),\phi_1(\omega,\mathrm{Z})) : \omega\in\Omega,\mathrm{Z}\in\mathcal{C}\}$$
$$\cup\{(\alpha_2(\omega,Z),\phi_2(\omega,Z)) : \omega\in\Omega, Z\in\mathcal{C}\}).$$

Then the local model may be written as

$$\tilde{F}(y,x)=\max\left\{\alpha+\phi^T(y-x) : (\alpha,\phi)\in\mathcal{G}\right\} \quad (7)$$

The advantage of (7) over (6) is that elements (α,ϕ) of $\mathcal{G}$ are easier to store than elements (ω,Z) of $\Omega\times\mathcal{C}$. In addition it is also more convenient to construct approximations $\mathcal{G}_k$ of $\mathcal{G}$. This is addressed in the next section.

Second local model and tangent program. Suppose x is the current iterate of our non-smooth algorithm. In order to generate trial steps away from x, we will recursively generate approximations $\tilde{F}_k(y;x)$ of $\tilde{F}(y;x)$ referred to as working models. Using (7), these will be of the form

$$\tilde{F}_k(y;x)=\max\left\{\alpha+\phi^T(y-x) : (\alpha,\phi)\in\mathcal{G}_k\right\},$$

where $\mathcal{G}_k\subset\mathcal{G}$. In particular, $\tilde{F}_k(y;x)\le\tilde{F}(y;x)$, with exactness $\tilde{F}_k(x;x)=\tilde{F}(x;x)=F(x;x)=0$ at $y=x$. Moreover, our construction assures that $\partial_1\tilde{F}_k(x;x)\subset\partial F(x;x)$ for all k and that $\tilde{F}_k$ gets closer to $\tilde{F}$ as k increases. In tandem with the proximity control management, this will assure that the $\tilde{F}_k$ get closer to the true F. Once the set $\mathcal{G}_k$ is formed, a new trial step y^{k+1} is computed via the tangent program

$$\min_{y\in\mathbb{R}^q}\tilde{F}_k(y;x)+\tfrac{\tau_k}{2}\|y-x\|^2.$$

3. APPLICATION TO LONGITUDINAL FLIGHT CONTROL

3.1 Longitudinal model

The longitudinal motion of an aircraft linearized around equilibrium (Mach= 0.7, Altitude= 5000 ft) is described by the following state space model (Boiffier (1998)):

$$\begin{bmatrix}\dot{x}_P\\ y_P\end{bmatrix}=\begin{bmatrix}A & B\\ C & D\end{bmatrix}\begin{bmatrix}x_P\\ u\end{bmatrix} \quad (8)$$

with:

- $x_P=[V,\ \gamma,\ \alpha,\ q,\ H]^T$,
- $u=[\delta_x,\ \delta_m]^T$,
- $y_P=[V,\ \gamma,\ N_z,\ q,\ H]^T$.

Altogether we have:

- 5 states. V is aerodynamic speed (m/s), γ is the climb angle (rd), α is the angle of attack (AoA, rd), $q=\dot{\theta}=\dot{\alpha}+\dot{\gamma}$ is the pitch rate (rd/s), H is the altitude (m),
- 2 controls. Engine thrust δ_x (% of the maximal thrust) and elevator deflection δ_m (rd),

- 5 measurements. The vertical load factor N_z (m/s^2) and $[V, \gamma, q, H]$.

The longitudinal dynamics are characterized by 5 eigenvalues:

- $\lambda_{1,2} = -0.56 \pm 1.61\,j$ (i.e. pulsation: $1.7\,rd/s$ and damping ratio: 0.33) is the AoA oscillation, also called short term mode. This mode mainly impacts the states α and q.
- $\lambda_{3,4} = -0.0039i \pm 0.064\,j$ (i.e. pulsation: $0.064\,rd/s$ and damping ratio 0.06) is the phugoid mode, also called long term mode. It mainly impacts the states γ and V.
- $\lambda_5 = -0.0026$ is the altitude convergence mode (a very long term mode). It mainly impacts altitude H.

3.2 Structured control law

In the longitudinal flight control study, the multi level control loop architecture discussed in the introduction is presented in Figure 1. This block diagram highlights:

- the sub-structure of the guidance loop in 2 parts. The altitude servo-loop and the auto-pilot loop (that is, the speed servo-loop and the climb angle servo-loop),
- the auto/manual switch on the input of the low-level control loop (also called flight control loop). During manual mode, the commands of the human pilot through the side-stick, are interpreted as vertical load factor input references N_{z_c} and sent directly to the flight control loop.

A commonly used flight control law reads

$$\delta_m = F(s)\Big((K_p + \frac{K_i}{s+\varepsilon})(N_{z_c} - N_z) - K_v\, q\Big) \qquad (9)$$

(where s is the frequency domain variable) and involves:

- a pitch rate feedback, through the gain K_v, which aims to damp the AoA oscillation,
- a proportional/integral feedback to servo-loop the vertical load factor N_z ($K_p + \frac{K_i}{s+\varepsilon}$: in fact, a pseudo-integrator is preferred),
- a low pass filter $F(s)$ to filter high frequency components in the control signal δ_m and to prevent spill-over on unmodeled dynamics.

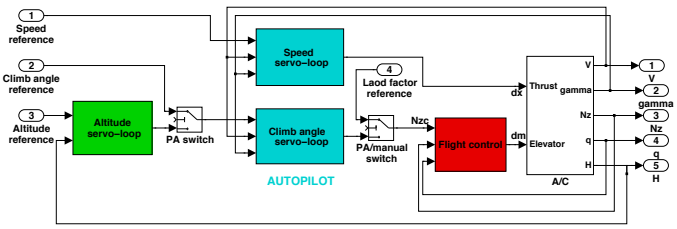

Fig. 1. Longitudinal flight control structure: inner flight control loop and outer auto-pilot loop.

3.3 Specifications and initial tuning

If the filter $F(s)$ is not considered, it is very easy to tune K_p, K_v, K_i in order to damp correctly the AoA oscillation and to meet good performances. The performance of the flight control law can be enhanced by a template $W_1(s)$ on the closed-loop sensitivity function T_1 between the input reference N_{z_c} and the servo-loop error d_{N_z}, that is the transfer between the first input and the first output of the plant $P(s)$ described on Figure 2:

$$T_1(s) = T_{N_z N_{z_c}}(s)\ .$$

Considering the template

$$W_1(s) = \frac{s^2 + 4s}{s^2 + 4s + 7},$$

the tuning # 1 defined by

$$\text{Tuning \# 1}: K_p = -0.1,\ K_v = -1,\ K_i = -0.15,\ F(s) = 1,$$

allows to match the template as seen in Figure 3. The N_{z_c} step response given in Figure 5 shows that the servo-loop performances are correct. In order to cut-off the control signal in high frequency (roll-off specification), a low pass template $W_2(s)$ is specified on the closed-loop transfer T_2 from the pitch rate measurement noise n_q (input # 2 of $P(s)$) and the load factor measurement noise (acting as the input reference N_{z_c}, input #1 of $P(s)$) to the control signal δ_m (output # 2 of $P(s)$). That is:

$$T_2(s) = T_{\delta_m [N_{z_c}, n_q]}(s).$$

The frequency-domain response of the template:

$$W_2(s) = 25/(s^2 + 5\sqrt{2}\,s + 25)\ ,$$

which aims at shaping a second order roll-off beyond $5\,rd/s$ is presented in Figure 4. To this end, one can try to choose $F(s) = W_2(s)$, that is:

$$\text{Tuning \# 2}: K_p = -0.1,\ K_v = -1,\ K_i = -0.15,\ F(s) = W_2(s).$$

Unfortunately this type of tuning does not meet the specifications, as can be seen in Figure 4. Indeed, the W_2 filter cut-off frequency ($5\,rd/s$) is too close to the N_z servo-loop bandwidth (around $2\,rd/s$). That leads to a badly damped closed-loop mode around $5\,rd/s$, that is, a resonance on the T_2 frequency-domain response (Figure 4, green plot) or to oscillations on the time-domain response (Figure 5, green plot). Setting up a trial-and-error procedure to tune K_p, K_v ,K_i and $F(s)$ simultaneously in order to satisfy the two antagonist templates on $T_1(s)$ and $T_2(s)$ appears a very cumbersome task. One can also notice on Figures 3 and 4 that frequency-domain templates do not need to be satisfied for pulsation under $0.1\,rd/s$, that is, flight control performances concern only the short-term dynamics and are not affected even when templates are violated in very low frequency.

Last but not least, a second order filter $F(s)$ can be parameterized by:

$$F(s) = \frac{a}{s^2 + bs + a},$$

and the control design problem can be stated in the following way: Compute the control law $x = [K_p,\ K_i,\ K_v,\ a,\ b]^T$ such that the closed-loop system is stable and satisfies

$$\left\|W_1^+ T_1\right\|_{\infty,[0.1;+\infty]} \leq 1, \qquad (10)$$

$$\left\|W_2^+ T_2\right\|_{\infty,[0.1;+\infty]} \leq 1, \qquad (11)$$

where $W_1^+(s)$ and $W_2^+(s)$ are the inverses of $W_1(s)$ and $W_2(s)$, regularized to be stable and implementable:

$$W_1^+(s) = \frac{s^2 + 4s + 7}{s^2 + 4s + 0.001}$$

$$W_2^+(s) = \frac{s^2 + 5\sqrt{2}\,s + 25}{25\Big((s/500)^2 + \sqrt{2}(s/500) + 1\Big)}\ .$$

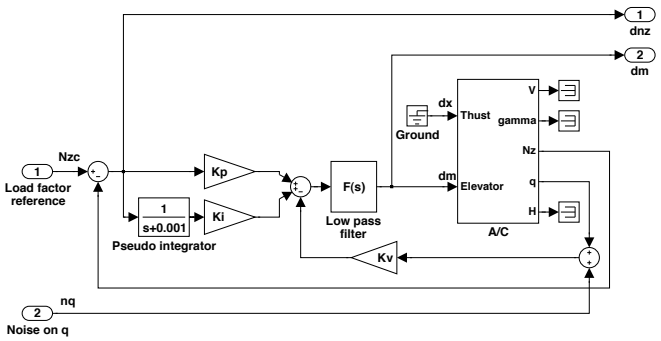

Fig. 2. The flight control loop: $P(s)$.

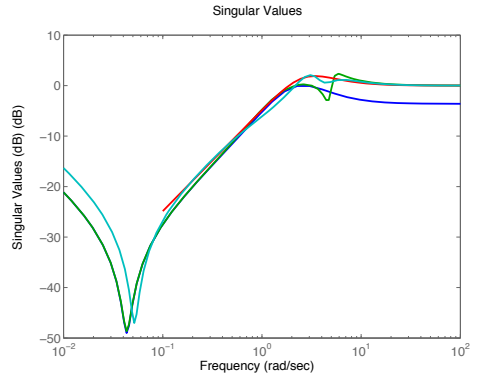

Fig. 3. Performance frequency-domain responses: template $W_1(s)$ (red) and $T_1(s)$ in the case of tuning # 1 (blue), tuning # 2 (green)and optimal tuning (cyan).

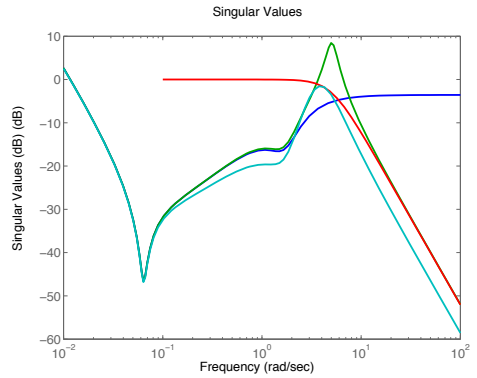

Fig. 4. Roll-off frequency-domain responses: template $W_2(s)$ (red) and $T_2(s)$ in the case of tuning # 1 (blue), tuning # 2 (green) and optimal tuning (cyan).

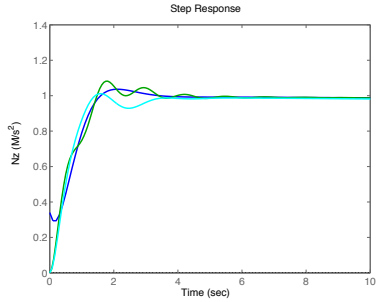

Fig. 5. Step response to N_{z_c} in the case of tuning # 1 (blue), tuning # 2 (green) and optimal tuning (cyan).

3.4 Results

The H_∞/H_∞ control design of section 2.2 was applied to this problem. The minimized channel $T_{w_1\to z_1}$ and the constrained channel $T_{w_2\to z_2}$ in the general expression (4) corresponded to $W_1^+T_1$ and $W_2^+T_2$. The optimized parametric vector found is

$$x = [-0.077603; -0.086731; -0.44784; 6.3451; 25.908]^T$$

with performance

$$\left\|W_1^+P_1\right\|_{\infty,[0.1;+\infty]} = 1.02, \quad \left\|W_2^+P_2\right\|_{\infty,[0.1;+\infty]} = 0.9\ .$$

Both constraints (11) (10) were active within numerical precision. Notice that (10) cannot decrease below 1. Responses obtained with this optimal tuning are plotted in Figures 3 to 5.

4. CONCLUSION

A non-smooth algorithm was proposed and applied to the longitudinal flight control of a carrier aircraft. It allows to take several frequency-domain specifications and a fixed structure of the control law into account. Preliminary results are encouraging and future work will address the following issues:

- In longitudinal flight control the simultaneous design (optimization) of the flight-control loop and the autopilot loop will be investigated.
- Our non-smooth approach will be extended to include time-domain specifications. This extension can be illustrated in the lateral flight control when time domain yaw/roll decoupling is required.
- Our approach will also be tested on control design problems involving flexible aircraft models (i.e. including structural bending modes) and load alleviation specifications.
- Finally, we will compare our innovative nonsmooth optimization algorithm with the pre-existing tools in matlab toolboxes for the resolution of the multi-objective H_∞ control problem on band-limited frequency-domain.

REFERENCES

Alazard, D. (2002). Robust $\mathcal{H}_2$ design for lateral flight control of a highly flexible aircraft. *Journal of Guidance, Control and Dynamics*, Vol. 25, No. 6, 502–509.

Apkarian, P. and Noll, D. (2006). Nonsmooth h infinity synthesis. *IEEE Transactions on Automatic Control*, 51 No. 1, 71–86.

Apkarian, P. and Noll, D. (2007). Nonsmooth optimization for multiband frequency domain control design. *Automatica*, 43 No. 4, 724–731.

Apkarian, P., Noll, D., and Rondepierre, A. (2008). Mixed h_2/h_∞ control via nonsmooth optimization.. *SIAM Journal on Control and Optimization*, 47 No. 3, 1516–1546.

Boiffier, J. (1998). *The Dynamics of Flight, the Equations*. John Wiley & Sons.

Burke, J. and Overton, M. (1994). Differential properties of the spectral abscissa and the spectral radius for analytic matrix-valued mappings. *Nonlinear Analysis: Theory, Methods and Applications*, 23 No. 4, 467–488.

Geromel, J., Souza, S., and Skelton, R. (1998). Static output feedback controllers: stability and convexity. *IEEE Trans. Autom. Control*, AC-43 (1), 120–125.

Madelaine, B. (1998). *Détermination d'un modèle pertinent pour la commande: de la réduction à la construction*. Thèse de Doctorat de Supaéro.

Noll, D., Prot, O., and Rondepierre, A. (2008). A proximity control algorithm to minimize nonsmooth and nonconvex functions. *Pacific Journal of Optimization*, 4, No. 3, 569–602.

Stevens, B.L. and Lewis, F.L. (1992). *Aircraft Control and Simulation*. John Wiley & Sons, Inc.

Tischler, M.B. (1996). *Advances in Aircraft Flight Control*. Taylor & Francis.