A new design methodology inspired by dynamic inversion techniques is proposed in this paper. It combines partially linearizing inner-loops with structured and robust outer-loops, which are designed using a non-smooth multi-model $H_\infty$ optimization approach. The proposed methodology also includes a robustness analysis scheme providing worst-case configurations, which are then used to enrich the bank of design models and thus iteratively improve the robustness properties of the designed outer-loops. Our approach is successfully tested on a realistic nonlinear aircraft control problem subject to large parametric variations and uncertainties.

Introduction

Robust feedback linearization techniques [10] have proved their efficiency in many aerospace applications, especially to control highly maneuverable aircraft or UAVs in large operating domains [25, 2, 20, 21, 11, 4]. Interestingly, such techniques do not only permit a large class of nonlinear systems to be linearized and decoupled, but also make it possible to adapt the control laws to the operating point. Thus, they become a competitive alternative to standard gain-scheduling techniques [7], which often entail many adjustments, or to LPV control design strategies, which require high fidelity LPV models [15]. However, standard dynamic inversion methods are often criticized for their lack of robustness and the need for an accurate model. This drawback is generally bypassed via robust linear outer-loops [6], which still require difficult and possibly time-consuming robustness evaluation a posteriori [18, 19]. Severe problems are also likely to occur when the actuator dynamics and limitations prevent an exact cancellation of the nonlinear terms. As emphasized in [12], it is therefore essential to take these dynamics into account in the design process. As observed in [8], one of the main reasons why standard dynamic inversion schemes exhibit poor robustness properties is due to the fact that the linearizing inner-loops are designed to convert the nonlinear system into a generic Brunovsky’s form. Following an intuitive path, it is then proposed in [8] to design the inner-loops so that, for given operating conditions, the nonlinear system will converge to its Jacobian linearization. Hence, the design of the linear robust outer-loops is no longer based on a generic model, but now explicitly depends on the linearized dynamics of the initial plant. A similar path, consisting of promoting interactions between inner and outer loops, is followed in this paper. More precisely, as in [8] a feedback linearization step is applied so that, in some enlarged neighborhood of given trim conditions, the nonlinear plant behaves like its linearization. In a second step, a robust linear outer-loop is designed. The originality of our approach lies in the particular structure of our $H_\infty$-based outer controller, which uses a nonlinear input as a key input to further enlarge the operating domain of the nonlinear closed-loop system. Various uncertainties are also taken into account in our procedure by a $\mu$-based robustness analysis phase, during which worst-case configurations are identified and then used in an iterative multi-objective and multi-model $H_\infty$ design process. It should be pointed out here that the resolution of such highly non-convex optimization problems has considerably improved in the past few years with recent advances in non-smooth $H_\infty$ optimization algorithms [3, 9]. Indeed, the latter now make it possible to fix the order of the $H_\infty$ controller as well as its structure and to consider multi-channel objectives and multiple models simultaneously, in order to increase the robustness margins. Such algorithms will then be a key ingredient in our proposed methodology.

The paper is organized as follows. A thorough description of the proposed methodology with its three main steps is first given. An application to a realistic landing aircraft control problem is then detailed. More precisely, we focus on the longitudinal part and give many details on how the method is applied. Then we briefly deal with the lateral part of the control problem, before presenting the global landing application on the full nonlinear aircraft model using the designed controller. Finally, we conclude the paper.
Description of the methodology

Robust Nonlinear Compensation Technique

Consider a continuous-time parameter-dependent nonlinear input-affine system described as follows

\[
\begin{align*}
\dot{x}(t) &= f(x(t),u_i(t)) + G(x(t),u_i(t))u(t) \\
u(t) &= L_i(u_i(t))
\end{align*}
\]  

(1)

where \( x(t) \in \mathbb{R}^n \) denotes the physical states evolving in the admissible operating domain \( X \subset \mathbb{R}^n \). The realized control inputs \( u(t) \in \mathbb{R}^m \) are derived from the commanded inputs \( u_i(t) \in \mathbb{R}^{m_i} \) via linear time-invariant actuators – denoted by \( L_i(.) \) – with unitary static gains. The nonlinearities and parametric variations of the system are captured by \( f(...) \in \mathbb{R}^n \) and \( G(...) \in \mathbb{R}^{m \times n} \), which both nonlinearly depend not only on the state vector but also on a set of parameters \( \theta \in \Theta \subset \mathbb{R}^p \).

Let us denote \( \xi(t), \xi_i(t) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \) an equilibrium point for system (1), thus satisfying:

\[
f(\xi, \xi_i) + G(\xi, \xi_i)\bar{u} = \bar{f} + \bar{G}u = 0 \tag{2}
\]

and rewrite \( f(\xi(t), \xi_i(t)) \) as follows:

\[
f(\xi(t), \xi_i(t)) = \bar{f} + Ax(t) + \Delta_j(t) \tag{3}
\]

where:

\[
A = \frac{\partial f}{\partial \xi} \bigg|_{\xi = \xi} \quad \text{and} \quad x(t) = \xi - \xi_i(t) \tag{4}
\]

and where \( \Delta_j(t) \) denotes the deviation between the nonlinear function \( f(...) \) and its linear approximation.

Assumption 1

There exists a constant matrix \( B \in \mathbb{R}^{m \times n} \) such that, for all \( (\xi(t), \xi_i(t)) \in X \times \Theta \), a nonsingular matrix \( \Lambda \in \mathbb{R}^{m \times m} \) and a residual error matrix \( \Delta_G \in \mathbb{R}^{m \times n} \) can be found such that:

\[
G(\xi(t), \xi_i(t)) = BA + \Delta_G \tag{5}
\]

\[
\forall \nu(t) \in \mathbb{R}^n, L_j(A^{-1}\nu(t)) = L_j(A^{-1}L_i(\nu(t))) \tag{6}
\]

The square matrix \( \Lambda \) typically represents the control input efficiency. When considering aerospace systems evolving in standard operating domains, the above non-singularity assumption – connected to the notion of controllability – is not restrictive. Moreover, the variations of this diagonal-dominant matrix are mainly induced by slowly-varying terms, such as the dynamic pressure. This observation justifies the commutative property (6) between \( \Lambda^{-1} \) and the fast dynamics \( L_i(.) \) of the actuators. Yet, a possible relaxation of (6) is introduced next.

Given any two signals \( \nu(t) \in \mathbb{R}^n \) and \( \zeta(t) \in \mathbb{R}^n \), let us now define the intermediate, input linearizing, control law

\[
u_i(t) = L_i(\xi_i(t))^{-1}(\nu(t) - \zeta(t)) + \bar{u} \tag{7}
\]

Combining equations (7) and (1), one readily obtains with the above notation in mind:

\[
\dot{x} = Ax + BL_i(v) + w_f + w_c \tag{8}
\]

where:

\[
w_f = \Delta_j - BL_i(\zeta) \tag{9}
\]

\[
w_c = (G - \bar{G})\bar{u} + \Delta_G A^{-1}L_i(v - \zeta) \tag{10}
\]

As is clear from Equation (8), where parametric-dependence and time-dependence have been omitted to alleviate notation, the parameter-dependent nonlinear system (1) has been reduced to a linear model with a new control input \( v \) and two measured perturbations \( w_f \) and \( w_c \). As is usual in dynamic inversion schemes, \( w_f \) can be partly canceled by an optimal choice of the auxiliary input signal \( \zeta(t) \):

\[
\hat{\zeta}(t) = \arg \min_{\zeta \in \mathbb{R}^n} \left\| \Delta_j(t) - BL_i(\zeta(t)) \right\| \tag{11}
\]

Remark 1

In the special case of square systems with idealized actuators (i.e., \( L_i(a(t)) = u(t) \)), one easily obtains \( w_f = 0 \) with \( \zeta(t) = B^{-1}\Delta_j(t) \).

Let us denote by \( w = w_f + w_c \) the vector of remaining input perturbations, which cannot be canceled, and assume that the latter is available for feedback, via estimation, at least on a limited bandwidth. The following structure for the linear outer-loop may thus be considered:

\[
v = K(s) \begin{bmatrix} w_f \\ w_c \\ y \end{bmatrix} \tag{12}
\]

where \( \hat{w} \), \( w_c \) and \( y \) denote respectively the estimation of \( w \), the target on the variables \( z \) to be tracked and the measurement signal. Without any significant loss of generality in most applications, both \( y(t) = Cz(t) \in \mathbb{R}^n \) and \( z(t) = Lx(t) \in \mathbb{R}^n \) are assumed to depend linearly on the state vector. The output feedback gain \( K(s) \) in (12) is to be designed so as to satisfy the following requirements:

- good tracking properties, by minimizing the error between \( z \) and the reference signal \( z_r = R(s)w_f \), where the LTI model \( R(s) \) describes the nominal reference closed-loop dynamics,

- a reasonable control activity, which is indirectly obtained by limiting \( v \) to avoid control input saturations,

- good rejection of the perturbations \( w = w_f + w_c \) that could not be entirely removed by the linearizing inner loop to enlarge the operating domain.

Denoting by \( \Sigma_j(s) \) the transfer matrix associated to the linear operator \( L_i(.) \) and by \( \Sigma(s) \) the linearized plant interconnection:

\[
\Sigma(s) = \begin{bmatrix} L_i & \{sI - A\}^{-1}B \\ C \end{bmatrix} \tag{13}
\]

the above outer-loop design issue can be recast into a linear framework as a multi-objective \( H_\infty \) minimization problem. More precisely,
considering the linear interconnection of Figure 1, it is proposed to compute $K(s)$ as follows:

$$\hat{K}(s) = \arg\min_{K(s) \in \mathcal{L}(s)} \left\| F_{w \rightarrow z}^s(s) \right\|_s$$

(14)

with:

$$\left\| F_{w \rightarrow z}^s(s) \right\|_s \leq \gamma_w$$

$$\left\| F_{z \rightarrow w}^s(s) \right\|_s \leq \gamma_z$$

(15)

The $H_s$-norm minimization considered in (14) corresponds to the nominal tracking requirement. As is standard in $H_s$ design approaches, a weighting function – typically a low-pass filter – $W_s^z(s)$ is introduced to specify the frequency domain where the tracking performance should be more efficient. Next, the second and third requirements are respectively taken into account by the additional two constraints in (15). The first one involves a high-pass weighting function $W_s^z(s)$ to minimize the control activity in the high-frequency domain. Finally, note the presence of a low-pass function denoted by $F(s)$, which can be viewed as an approximation of the estimation process ($\hat{w} \approx F(s)w$).

Problem (14)-(15) is a difficult non-convex and non-smooth optimization problem. However, it is now efficiently solved thanks to recent advances in non-smooth optimization techniques [3, 9]. Moreover, the structure of the controller as well as its order can also be fixed a priori, which makes the implementation easier. Last but not least, multiple models can be considered simultaneously during the design procedure. This flexibility will be used below to improve the robustness properties of our proposed design scheme.

**Preliminary LTI Robustness Analysis**

After an initial controller has been obtained as the solution of the above multi-objective design problem, closer attention is now paid to the validity of a few approximations that were made and their potential impact on the closed-loop behavior. More precisely, it is shown here how LTI robustness analysis tools (such as $\mu$ or skew-$\mu$ bounds) can be used to detect potential difficulties induced by three main sources of uncertainties.

**Plant uncertainties**

When combining Equations (7) and (1) to obtain (8), it is assumed that both $f$ and $G$ are well known. However, in practice, uncertainties $\delta$ are always present so that Equation (8) is now written as:

$$\dot{x} = A(\delta)x + B(\delta)L_s^z(\nu) + w$$

(16)

where $\delta = \left[ \delta_1, ..., \delta_s \right] \in \mathbb{R}^s$. Assuming that both $A$ and $B$ rationally depend on $\delta$ it is then possible through an LFT modeling approach [14] to rewrite $\Sigma(s)$ as the following Linear Fractional Representation (LFR), as depicted in Figure 2:

$$\Sigma(s) = \mathcal{F}_s(M(s), \Delta_u)$$

(17)

where $\mathcal{F}_s(\cdot)$ denotes the upper Linear Fractional Transformation (LFT), $M(s)$ is an LTI system and:

$$\Delta_u \in \Delta_u = \left\{ \text{diag} \left( \delta_1, ..., \delta_s \right), \delta_i \in \mathbb{R} \right\}$$

(18)

**Actuator uncertainties**

In Assumption 1, the commutative property (6) might not be valid in all cases. Following [5], it is then proposed to relax it as follows.

**Assumption 2**

There exist a nonlinear bounded operator $\Gamma(\cdot)$ and a positive bound $\kappa_\nu \in \mathbb{R}^+$ such that $\forall \nu(t) \in \mathbb{R}^n$:

$$L_s^z(\nu(t)) = \Lambda^{-1}L_s^z(\nu(t)) + \Gamma(\nu(t))$$

$$\left\| \Gamma(\nu(t)) \right\| \leq \kappa_\nu \left\| \nu(t) \right\|$$

(19)

(20)

It results from Assumption 2 that the linear model $\Sigma_s^z(s)$ in Figure 1 should now be replaced by the nonlinear uncertain version $\Sigma_s^z(s) + \Gamma(\cdot)$, as illustrated in Figure 2.

**Estimation uncertainties**

Finally, one should notice that only an estimated $\hat{w}$ of the nonlinear input perturbations $w$ is available to the outer-loop controller $K(s)$. While this estimation process has been taken into account in the design phase through the approximation $\hat{w} \approx F(s)w$, it might be too optimistic in practice. A diagonal perturbation block $\Delta_w$ will then be considered to introduce some multiplicative uncertainties on the signal so that $\hat{w}$ is now transformed into $(I + \Delta_w)\hat{w}$. This is also visualized in the robustness analysis diagram in Figure 2.

As is usual in any standard LFT modeling process, the three uncertain blocks that have been described above are normalized, merged into a single block-diagonal operator $\Delta = \text{diag} \left( \Delta_u, \Gamma(\cdot), \Delta_w \right)$ and pulled out to generate an augmented linear model $P(s)$. These operations are summarized in Figure 3. The APRICOT Library from the Matlab...
Figure 3 – LFT of the system for LTI robustness analysis.

Toolbox SMAC [22] can be used to generate such an LFT, from a set of LTI models.

Temporarily assuming that \( \Delta \) is a time-invariant operator, \( \mu \) or skew-\( \mu \) analysis can be used to quantify, respectively, the stability and performance properties of the uncertain closed-loop depicted in Figure 3. Using the SMART Library from the Matlab Toolbox SMAC [22] allows bounds on the structured singular value [24] to be computed. Since \( \Delta \) has been normalized, the system is said to be robustly stable to LTI perturbations if the \( \mu \) upper-bound verifies \( \mu < 1 \). Otherwise, a destabilizing worst-case perturbation \( \Delta^* \) might exist such that \( \sigma(\Delta^*) < 1 \).

Multi-model Design

Using the above robustness analysis results, two main ways exist to improve the initial controller \( \hat{K}_s(s) \). Since it was designed within the \( H_\infty \) framework, a natural way consists in using a \( \mu \)-synthesis approach. While this strategy, already implemented in MATLAB™ [26], works well in the case of complex-valued uncertainties, numerical difficulties are often reported with real-valued uncertain parameters. Moreover, results might be quite conservative in that case. For these reasons, the alternative path, consisting of a multi-model design approach, will be preferred next. First considered in [1], this intuitive approach leads to non-convex optimization problems. However, as has already been pointed out, it has regained interest recently thanks to the flexibility of non-smooth \( H_\infty \) optimization algorithms. The latter indeed offer new perspectives in this direction, since multiple models can be considered simultaneously. From this observation, a simple iterative algorithm can be proposed. Starting from a single-model design, the principle consists of analysis & multi-model design iterations. During the analysis step, worst-case configurations (associated with \( \Delta^* \)) are isolated, in order to enrich the bank of models to be considered in the next design step. This is summarized below.

Algorithm 1

Robust multi-model design

1. Define a nominal configuration model \( \Sigma_0(s) \) using the input linearizing inner-loop control law (7), leading to (8).
2. Set \( i = 0 \) and solve (14)-(15) to compute an initial controller \( \hat{K}_s(s) \).
3. Perform LFT modeling & robustness analysis pursuant to previous Subsection. Extract a destabilizing perturbation \( \Delta^* \). If \( \sigma(\Delta^*) < 1 \), then go to Step 4, otherwise go to Step 5.
4. Enrich the bank of models with \( \Delta^*_i \), set \( i \leftarrow i + 1 \), compute \( \hat{K}_s(s) \) via multi-model \( H_\infty \) design and return to Step 3.
5. Perform a final robustness analysis with \( \mu \) upper-bound evaluation.

Remark 2

Unlike \( \mu \)-synthesis based approaches, the above algorithm implements necessary conditions for robust stability with respect to LTI perturbations, which become sufficient if the \( \mu \) upper-bound in Step 1 is less than 1.

Towards a Global Robustness Analysis

In the simplified robustness analysis approach of page 3, the nonlinear input signal \( w \) is considered as an external perturbation. Yet, considering Equations (8) and (9), it is clear that \( w \) may depend on \( x \) and \( \theta_r \) in a quite complicated way. Consequently, robustness analysis becomes tricky in the most general case. Fortunately, with a good knowledge of the studied process, reasonably simpler approximations can be obtained in practice, such as:

\[
\begin{align*}
\dot{w} &= H(\theta_r)x + W(x) \\
&= (21)
\end{align*}
\]

with the following assumptions:

- \( H(\cdot) \) is assumed to depend rationally on \( \theta_r \), and can thus be rewritten as an LFT.
- the nonlinear operator \( W(x) \) satisfies Lipschitz conditions.

Hence, the LFT \( F_s^*(P(s),\Delta) \) of Figure 3 – further denoted \( F_s^*(\hat{P}(s),\hat{\Delta}) \) – is modified to include additional blocks in \( \hat{\Delta} \) so that \( w \) will no longer appear as an external perturbation:

\[
\hat{\Delta} = diag(\Delta_m, \Gamma(\cdot), \Delta_n, W(\cdot), \Theta_r) \\
&= (22)
\]

The extended uncertain operator \( \hat{\Delta} \) now clearly contains uncertainties \( \Delta_m, \Delta_n, \) and time-varying parameters \( \Theta_r \), as well as memoryless nonlinearities \( \Gamma(\cdot) \) and \( W(\cdot) \). Robustness analysis then must be performed with more general tools based, for example, on the IQC framework [16].

Application to longitudinal aircraft control design

The above robust nonlinear compensation framework is now applied to a longitudinal aircraft control problem. Note that the lateral motion is not yet considered, assuming a steady flat-wing aircraft with no sideslip. Lateral controller design will be described while global simulations with both combined motions will be presented.

Nonlinear Longitudinal Aircraft Model

The longitudinal motion of a civil aircraft can be described by the following 4-state model [7]:

\[
\begin{align*}
\dot{m}V &= -\bar{q}SC_\mu(\alpha) - mgsin\gamma + F\cos\alpha \\
\dot{m}V\dot{\gamma} &= \bar{q}SC_\mu(\alpha, q, V, \delta_e) - m\cos\gamma + F\sin\alpha \\
J\dot{q} &= \bar{q}SLC_\mu(\alpha, q, V, \delta_e) + z_eF \\
\dot{\theta} &= q
\end{align*}
\]

Application to longitudinal aircraft control design

The above robust nonlinear compensation framework is now applied to a longitudinal aircraft control problem. Note that the lateral motion is not yet considered, assuming a steady flat-wing aircraft with no sideslip. Lateral controller design will be described while global simulations with both combined motions will be presented.
where \( V \) is the airspeed, \( \gamma \) is the flight path angle, \( q \) is the pitch rate, \( \theta \) is the pitch angle, \( p = \frac{1}{2} \rho V^2 \) is the dynamic pressure, \( J \) is the longitudinal inertia, \( m \) is the mass, \( S \) is the reference surface, \( L \) is the reference length (often taken as the half span), \( \rho \) is the air density, \( \alpha \) is the angle of attack \( (\alpha = \theta - \gamma) \), \( F \) is the engine thrust on the longitudinal axis and \( z_e \) is the vertical shift between the position of the center of gravity and the thrust application point. The drag, lift and pitching coefficients \( \hat{C}_d \), \( \hat{C}_l \) and \( \hat{C}_n \) can be expressed as follows:

\[
\begin{align*}
\hat{C}_d(\alpha) &= C_{d0} + C_{d\alpha}\alpha + C_{d\alpha^2}\alpha^2 \\
\hat{C}_l(\alpha, q, V, \delta_e) &= C_{l0} + C_{l\alpha}\alpha + C_{l\alpha q} q + C_{l\delta_e}\delta_e \\
\hat{C}_n(\alpha, q, V, \delta_e) &= C_{n0} + C_{n\alpha}\alpha + C_{n\alpha q} q + C_{n\delta_e}\delta_e
\end{align*}
\] (24)

where \( \delta_e \) is the elevator deflection angle and the \( C_n \) are fixed aerodynamic coefficients, whose values can be obtained from the benchmark library of the SMAC Toolbox\(^1\) and do not depend on the Mach number. Here \( \xi = \begin{bmatrix} V & \gamma & \alpha \end{bmatrix}^T \) and \( \theta = \begin{bmatrix} m \gamma & \alpha \end{bmatrix}^T \), where \( x_m \) is the center of gravity position and \( V \) is the initial airspeed. The operating domain for a landing application is such that \( V \in [60 \, 120] \text{ m/s}, \ m \in [123 \, 180] \text{ tons} \) and \( x_m \in [15 \, 41] \% \), defining the admissible set \( \Theta \). \( F \) and \( \delta_e \) follow the first-order actuator dynamics \( \Sigma_1(s) \):

\[
\Sigma_1(s) = \begin{bmatrix} \frac{1}{s+1} & 0 \\ 0 & \frac{1}{1+s/s} \end{bmatrix}
\] (25)

**Remark 3**

Using first-order models for the actuators is common practice in the aeronautical industry, and it is sufficiently representative here. However, rate limitations are neglected, but it would be possible to add an anti-windup controller to take them into account.

The aerodynamic coefficients can have up to 30\% of multiplicative uncertainties. Using the notation of (1) and applying (5) from Assumption 1, (23) is rewritten the same way with:

\[
f = \begin{bmatrix} \frac{-\pi S C_{l0}}{m} - x m g \sin y \\ \frac{\pi S C_{l\alpha}}{m} \frac{x_m \cos y}{q} \\ \frac{\pi S C_{l\alpha}}{q} \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 & 0 \ y \\ 0 & 0 & 1 \end{bmatrix}
\]

\[
A = \begin{bmatrix} \cos \alpha \sin \gamma & \sin \gamma \\ \frac{\pi S C_{n0}}{m} \end{bmatrix}, \quad \Delta_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sin \gamma \sin \alpha \cos \gamma & \sin \gamma \cos \gamma \cos \alpha \end{bmatrix}
\]

**Remark 4**

Given the operating domain of the system, \( \Lambda \) is non-singular. Furthermore, \( \Lambda \) is slowly varying, since its dynamics mainly come from the dynamic pressure \( \bar{p} \). These two conditions verify Assumption 1.

\(^1\) http://w3.onera.fr/smac/

**Nonlinear Compensation Technique**

Using the previous notation it is now possible to apply the method described in page 2. First applying the control input (7) leads to the linearized system (8). \( \tilde{Z} \) is chosen so that \( W_p \) only preserves the nonlinear terms on the flight path angle dynamics \( \gamma \), canceling the nonlinear dynamics on \( V \) and \( q \). The nominal configuration \( \Theta_0 \) is set for a mass of 150 tons, a center of gravity position at 21\%, an initial airspeed of 70 m/s and an altitude of 300 m (landing configuration). This configuration is “central” in terms of the system pole location, ensuring that the other configurations are covered as much as possible when designing the robust controller. Choosing a worst-case configuration could also be a possibility, but the nominal performance is highly degraded in this case and the “opposite” worst-case configurations may be harder to control. According to the specifications of the robust nonlinear compensation technique; the frequency weightings \( W_p(s) \) and \( W_n(s) \) are chosen respectively as low-pass and high-pass filters:

\[
W_p(s) = \begin{bmatrix} \frac{s/2+0.15}{s+1.5 \times 10^{-4}} & 0 \\ 0 & \frac{s/20+0.4}{s+4 \times 10^{-4}} \end{bmatrix}
\] (27)

\[
W_n(s) = \begin{bmatrix} 0 & 0 \\ 0 & \frac{0.00125}{s+0.2} \end{bmatrix}
\] (28)

The filter \( F(s) \) is such that:

\[
F(s) = \begin{bmatrix} \frac{1}{s+1} & 0 & 0 & 0 \\ 0 & \frac{1}{s+1} & 0 & 0 \\ 0 & 0 & \frac{1}{s+1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\] (29)

The second term in (29) on the diagonal is higher than the others, since it is not compensated by the input signal \( \tilde{Z} \) (which only compensates for the first and the third nonlinear terms). This enables the controller to focus more on this one than on the others. The last term is 0 since the last equation in (23) does not contain any nonlinear terms.

A third-order controller is chosen, since it offers a good compromise between the achievable performance and a preferable low-order controller for easier implementation. Solving the multi-objective \( H_\infty \) minimization problem (14)-(15) with the routine hinfsctruct of the Robust Control Toolbox for Matlab [3] yields an \( H_\infty \)-norm of \( \gamma_c = 0.9653 \) after a few iterations, thus insuring that the frequency-domain

**Remark 5**

The choice of \( B \) was made based on the maximum control efficiency. Indeed, as is clear from (23), the thrust input \( F \) mainly affects the airspeed \( V \) and the elevator deflection \( \delta_e \) has a large impact on the pitch rate \( q \).

The goal is to track the airspeed \( V \) and the flight path angle \( \gamma \), and the state is assumed to be fully available to the controller, yielding \( C = I \). The reference model \( R(s) \) for the airspeed (1\st order dynamics) and the flight path angle (3\rd order dynamics) is given by:

\[
R(s) = \begin{bmatrix} \frac{1}{s+0.35} \\ 0 & 0 \\ \frac{0.35}{(s+1)(s+2)(s+0.7)(s+0.35)+s} \end{bmatrix}
\] (26)
specifications are fulfilled. Nonlinear simulations are performed with the obtained initial controller $\hat{K}_s(s)$, and the corresponding results are depicted in Figure 4 and Figure 5.

In Figure 4, the aircraft responses to a 3 degree step demand on the flight path angle $\gamma$ are visualized for various configurations of mass $m \in \{120, 150, 180\}$ tons, initial airspeed $V \in \{60, 70, 80\}$ m/s, and center-of-gravity locations $x_g \in \{15, 20, 40\}$ %.

Similarly, the responses to a 3 m/s variation step demand on the longitudinal airspeed $V$ are shown in Figure 5. In both cases, the dashed red plots correspond to the reference signals to be tracked.

Figure 4 – Nonlinear Simulations with a 3° step on $\gamma$, for different initial aircraft configurations and no aerodynamic coefficient uncertainties, with the initial controller $\hat{K}_s(s)$.

Figure 5 – Nonlinear Simulations with a 3 m/s step on $V$, for different initial aircraft configurations and no aerodynamic coefficient uncertainties, with the initial controller $\hat{K}_s(s)$.
Both simulations prove the effectiveness of the robust nonlinear compensation scheme with regard to performance and robustness properties. The flight domain is indeed very large thanks to the nonlinear compensation part of the controller being aimed at maintaining as much as possible the nominal performance. The controller manages to follow the reference models quite well, while also ensuring a good decoupling. Note however that no uncertainties in the aerodynamic coefficients have been considered yet. These will have significant impact on the performance and stability of the system, as will be shown in the next section dealing with robust stability analysis.

Robustness analysis and multi-model design

The above controller $\hat{K}_w(s)$ has been designed to cope with a large operating domain under nominal conditions. The uncertain operators represented by "gray boxes" in Figure 2 have thus not been considered yet. The objective of this section is twofold. First, the effects of the aforementioned uncertainties are studied within the LTI frameworks with the help of $\mu$-analysis tools. Next, the identified worst cases are used in a multi-model design strategy detailed in Algorithm 1 to improve the robustness properties of the initial controller.

LTI modeling and $\mu$-analysis

LFT modeling is a key step in our robustness analysis process. Starting from the set of nonlinear equations (23), parametric uncertainties are first introduced in the aerodynamic coefficients:

$$C_D = (1 + \delta_{c_D}) \hat{C}_D$$
$$C_L = (1 + \delta_{c_L}) \hat{C}_L$$
$$C_n = (1 + \delta_{c_n}) \hat{C}_n$$

and variations are also introduced in the airspeed $V$, mass $m$ and center of gravity location: $V = V_0 + \delta_{x}$, $m = m_n + \delta_{m}$, $x_c = x_{c_0} + \delta_{x_c}$. Next, the equations are linearized so that a bank of parameter-dependent linear models, as described in (16), is obtained. From this continuous set, a low-order LFT model $\mathcal{F}_w(M(s), \Delta_u)$ (see Eq. (17)) is rather easily obtained with the help of the most recent algorithms implemented in the APRICOT Library of the SMAC Toolbox [23]. These algorithms, using orthogonal least square techniques, are based on low-order polynomial interpolation methods. For this application the size of the $\Delta_u$-block is kept reasonably low:

$$\Delta_u = \text{diag}(\delta_{c_D}, \delta_{c_L}, \delta_{c_n}, \delta_{x_0}, \delta_{m_0}) \in \mathbb{R}^{15 \times 15}$$

As a result, the size of the global $\Delta$-block including $\Gamma(.)$ and $\Delta_u$ (see Figure 3) verifies:

$$\Delta = \text{diag}(\Delta_u, \Delta_v, \Delta_T) \in \mathbb{R}^{20 \times 20}$$

A normalization step is finally applied so that variations of $\Delta$ in the unit ball induce 30% uncertainties in the aerodynamic coefficients, and 30% variations in $\Delta_u$ and $\Delta_T$. Note that this last step is easily achieved with the LFT modeling library available in the SMAC Toolbox. Based on this normalized LFT object, both upper and lower bounds of the structured singular value $\mu$ are evaluated with the SMART Library of the SMAC Toolbox. With the nominal controller $\hat{K}_w(s)$, a lower-bound $\mu < 1$ is found together with its corresponding worst-case configuration $\Delta_w^*$ such that $\sigma(\Delta_w^*) = \mu^{-1} < 1$.

Multi-model design

Following step 4 of Algorithm 1, $\Delta_w^*$ is used to initialize our multi-model design procedure. Note that this case corresponds to a retracted positioning of the center of gravity, maximum weight and high values of the aerodynamic coefficient uncertainties. After 5 more iterations, a new controller $\hat{K}_w(s)$ is obtained that significantly improves the worst-case configuration. For this controller one indeed obtains $\mu_w < 1$. However, using standard algorithms, the upper-bound $\mu_w$ remains above 1, which does not make it possible to conclude on stability. Refined versions of the algorithm using branch-and-bound techniques [17], now available in the SMART Library, are then used, which greatly improves the accuracy of the upper-bound. One eventually obtains $\mu_w < 1$, as is summarized in Table 1.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Lower-bound $\mu_w$</th>
<th>Upper-bound $\mu_w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal – DG-scaling</td>
<td>1.4610</td>
<td>1.5110</td>
</tr>
<tr>
<td>Multi-model – DG-scaling</td>
<td>0.8957</td>
<td>1.0780</td>
</tr>
<tr>
<td>Multi-model – branch-and-bound</td>
<td>0.8999</td>
<td>0.9449</td>
</tr>
</tbody>
</table>

Table 1 – Values of the $\mu$-bounds for the nominal and multi-model controllers. The computations for the multi-model controller are made using the initial DG-scaling method and the improved branch-and-bound algorithm.

Time-domain evaluation

To conclude this section, a few nonlinear simulations are performed with the above multi-model based controller. The flight path angle ($\gamma$) and airspeed ($V$) step demands are first applied for various aircraft configurations without uncertainties. The results, to be compared with those obtained with the nominal controller, are displayed in Figures 6 and 7.

As expected, the decoupling is a bit degraded and the responses are slightly slower but major improvements will be shown in the presence of uncertainties.

Let us now introduce 25% uncertainties in the aerodynamic coefficients (not 30%, so that the nominal controller is not completely unstable). The aircraft responses to a step demand on $\gamma$ are visualized in Figure 8. The left subplot (Figure 8a) reveals poor robustness properties of the nominal controller, while significant improvements are clearly observed on the right subplot (Figure 8b) with the multi-model controller. This clearly demonstrates the efficiency of the proposed methodology.

Application to lateral aircraft control design

Now that the method has been successfully applied on the longitudinal part of the aircraft, the lateral part will be dealt with using the exact same strategy as explained below.
Figure 6 – Nonlinear Simulations with a 3° step on $\gamma$, for different initial aircraft configurations and no aerodynamic coefficient uncertainties, with the multi-model controller $\hat{K}(s)$.

Figure 7 – Nonlinear Simulations with a 3 m/s step on $V$, for different initial aircraft configurations and no aerodynamic coefficient uncertainties, with the multi-model controller $\hat{K}(s)$.
Nonlinear Lateral Aircraft Model

The lateral motion of a civil aircraft can be described by the following 4-state model [7]:

\[
\begin{align*}
\dot{mV} = mV (p \sin \alpha - r \cos \alpha) + mg \left(\sin \theta \cos \alpha \sin \beta + \cos \beta \cos \phi \sin \beta - \cos \theta \sin \alpha \cos \alpha \sin \beta \right) \\
+ \frac{qS}{\lambda} \dot{C}_y - F \sin \beta \cos \alpha \\
\phi = p + \tan \theta (q \sin \phi + r \cos \phi) \\
I_{xx} \dot{r} = \frac{qS}{\lambda} \dot{C}_i + (I_{xx} - I_{yy}) pq \\
I_{yy} \dot{p} = \frac{qS}{\lambda} \dot{C}_a + (I_{yy} - I_{xx}) rq
\end{align*}
\]

(30)

where \(\delta_a\) and \(\delta_r\) are, respectively, the aileron and rudder deflection angles, and the \(C_{xy}\) are aerodynamic coefficients specific to the given aircraft. Using similar notations to those used in the longitudinal case, let us define:

\[
\xi = [\beta \phi r p]^T, \quad u = [\delta_a \delta_r]^T \quad \text{and} \quad \vartheta = [m \bar{x} V]^T.
\]

\(\delta_a\) and \(\delta_r\) follow the first-order actuator dynamics

\[
A_s = \begin{bmatrix}
0 & 0.06 & 1 \\
0 & 1 & 0.2 \\
0 & 1 & 0.1
\end{bmatrix}
\]

(32)

Using the notation of (1) and applying (5) from Assumption 1, (23) is rewritten in the same way with:

\[
\Lambda = \begin{bmatrix}
\frac{qS}{\lambda} C_{xy} & \frac{qS}{\lambda} C_{xy} \\
\frac{qS}{\lambda} C_{iy} & \frac{qS}{\lambda} C_{iy}
\end{bmatrix}
\]

\[
\Delta_G = \begin{bmatrix}
0 & \frac{qS}{mV} C_y \\
0 & 0 \\
0 & 0 \\
0 & 0
\end{bmatrix}
\]

where \(\beta\) and \(\phi\) are, respectively, the sideslip and roll angles, and \(\bar{x}\) and \(\bar{y}\) are the inertias along the \(x\)-body and \(z\)-body axes. The side force, roll and yaw coefficients \(\dot{C}_y\), \(\dot{C}_i\) and \(\dot{C}_a\) are expressed as follows:

\[
\begin{align*}
\dot{C}_y & = \frac{C_y}{\lambda} \beta + C_y \delta_a \\
\dot{C}_i & = \frac{C_i}{\lambda} \dot{C}_a + \frac{L}{V} \left( C_{ia} p + (C_{ia} + C_{ic} \alpha) r \right) + C_{ia} \delta_a + C_{ic} \delta_i \\
\dot{C}_a & = (C_{sa} + C_{ia}) \beta + \frac{L}{V} \left( C_{ia} r + (C_{sa} + C_{ia} \alpha) p \right) \\
& \quad + C_{sa} \delta_a + C_{ia} \delta_i
\end{align*}
\]

(31)

\[
f = \begin{bmatrix}
p \sin \alpha - r \cos \alpha + \frac{g}{V} (\sin \theta \cos \alpha \sin \beta + \cos \beta \cos \phi \sin \beta - \cos \theta \sin \alpha \cos \alpha \sin \beta)
\end{bmatrix}
\]

\[
= \begin{bmatrix}
p + \tan \theta (q \sin \phi + r \cos \phi) \\
\frac{qS}{\lambda} \dot{C}_y - \frac{I_{xx}}{I_{xx}} - I_{yy} pq \\
\frac{qS}{\lambda} \dot{C}_i + \frac{I_{yy} - I_{xx}}{I_{xx}} rq \\
B = \begin{bmatrix} 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \end{bmatrix}
\end{bmatrix}
\]
The goal is to track the roll and sideslip angles \( \phi \) and \( \beta \), respectively. The reference model \( R(s) \) is given by:

\[
R(s) = \begin{pmatrix}
\frac{1}{s + 0.85} & 0 \\
0 & \frac{\frac{1}{s + 0.75} \cdot 0.75}{s + 1}
\end{pmatrix}
\] (33)

**Remark 6**

Note that some longitudinal state variables are present in the lateral dynamics equations. For the computation of the lateral controller it will be assumed that these variables maintain a predefined equilibrium. The simulations will be performed on the full model with the previous satisfactory longitudinal controller, in order to maintain the longitudinal equilibrium despite the lateral motion.

**Lateral Controller Design**

The same method as for the computation of the longitudinal controller is used here (some details will be omitted). The nominal configuration \( \theta_p \) is the same as before. For the lateral case, the frequency weightings used for the \( H_\infty \)-design procedure \( W_p(s) \) and \( W_u(s) \) are chosen respectively as:

\[
W_p(s) = \begin{pmatrix}
\frac{s + 2}{s + 5} & 0 \\
0 & \frac{s + 5}{s + 10}
\end{pmatrix}
\] (34)

\[
W_u(s) = \begin{pmatrix}
\frac{s + 2}{0.001s + 20} & 0 \\
0 & \frac{s + 0.5}{0.001s + 5}
\end{pmatrix}
\] (35)

The filter \( F(s) \) is such that: (see the explanations for the choice of (29) for a better understanding)

\[
F(s) = \begin{pmatrix}
\frac{1}{s + 1} & 0 & 0 & 0 \\
0 & \frac{1}{s + 1} & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{0.02}{s + 1}
\end{pmatrix}
\] (36)

For the same reasons as for the longitudinal controller, a second-order controller is chosen. Solving the multi-objective \( H_\infty \) minimization problem (14)-(15) with the routine hinfsyn of the Robust Control Toolbox for Matlab [3] yields a \( H_\infty \)-norm \( \gamma = 1.03 \) after a few iterations. The stability of the closed-loop system with the initial "nominal" controller \( \hat{K}_s \) is checked. Using the same strategy as for the longitudinal case, an LFT is computed yielding a "\( \Delta \)-block" of dimension 24 x 24. Then, both the upper and the lower bounds of the structured singular value \( \mu \) are evaluated yielding \( \mu \in [1.007 1.118] \) which does not prove stability. Following Algorithm 1, after 3 iterations, the final controller \( \hat{K}_s \) greatly improves the worst-case stability since now \( \mu \in [0.7948 0.8845] \) which proves stability with respect to LTI uncertainties. Note that, unlike the longitudinal case, no branch-and-bound techniques were needed here to reduce the gap between the lower and upper bounds.

Simulations were run for various initial configurations (different values of \( \theta_p \)) and with \( \pm 30\% \) uncertainties on the lateral aerodynamic coefficients (729 runs were executed). As expected, the lateral multi-model controller performs very well as shown in Figures 9.
and 10, where respectively the references of the roll and sideslip angles $\phi$ and $\beta$ (red dashed line) are well tracked with reasonable control inputs $\delta_a$ and $\delta_r$. Furthermore, no sign of even slight instability is present and the decoupling is greatly ensured.

**Global application: aircraft landing**

Finally, the longitudinal and lateral controllers are tested on a realistic landing application. The Instrument Landing System (ILS) allows the aircraft to operate and automatically land despite difficult weather conditions (e.g., restricted visibility). It is composed of ground-based signal transmitters and on-board receiving equipment. The ground-based equipment, located at the edges of the runway, comprises localizer and glide path radio transmitters and some marker beacons. On board, the receivers allow the emitted radio signals to be processed.

The glide and localizer signals are the ones of interest for performing an automatic landing. They both allow the position of the aircraft relative to the ideal trajectory for landing to be known. Thus, a guidance control law can be designed so that the aircraft trajectory follows the centerline of the runway with a nominal descent path angle of $\gamma_{nom} = -3$ degrees. The lateral deviation is given by the localizer beam, while the longitudinal deviation is given by the glide beam. Figures 11 and 12 present the principles. Further details on autoland can be found in [13].

The aim is thus to track the glide and localizer signals and to perform the landing, via the addition of simple guidance outer loops which provide necessary inputs orders to the controller. The aircraft must hit the ground with a vertical speed $V_z = [2-3]$ ft/s not further than 500 meters after the runway threshold. Furthermore, if a lateral wind is blowing the aircraft must be able to have a final azimuth angle $\Psi'$ of 0° without being out of the axis of the runway during the process.
These challenging goals will be verified by running many simulations on the full nonlinear model with, as before, model uncertainties, different initial conditions and wind.

**Guidance system**

Thanks to an ILS model block the vertical and lateral shifts $dZ$ and $dY$ respectively between the actual position of the aircraft and the glide and localizer signals are recovered. PID controllers, which have as inputs $dZ$ and $dY$, have been designed to provide the longitudinal and lateral controllers with the necessary values of $\gamma_c$ and $\phi_c$ to cancel these shifts. The design is not detailed in the paper, since it really consists of classical design techniques (linearization of the plant from $dZ$ and $dY$ to the controller inputs $\gamma_c$ and $\phi_c$, and PID design from frequency domain specifications). The nominal airspeed will be kept constant ($V = 70$ m/s). The general principle of the landing guidance + control system is given in Figure 13. Note that the flare + decrab control system block will be described in the following subsection.

**Flare and decrab phases**

Just before landing, thanks to the ILS guidance detailed above, the aircraft should fly with a fixed airspeed of $–3°$ m/s, a flight path angle of $–3°$ and be aligned with the runway. The objective of the flare control system, usually activated at 50 ft (15 m) above ground consists in generating an appropriate slope angle $\gamma_c$ to be tracked so that the vertical speed at touchdown approaches 2.5 ft/s (0.75 m/s) and the aircraft hits the ground 400 m after the runway threshold. In order to do so, a corresponding trajectory $h(x)$ is determined using some geometric considerations and $\gamma_c$, is given in real time using the following relation:

$$\dot{h}(x(t)) = \frac{dh(x)}{dx}V_{\gamma'}(t) = -V(t)\sin(\gamma_c(t))$$

(where $V_{\gamma'}$ is the ground speed). The procedure is shown in Figure 14.

Table 2 summarizes the landing minimum, maximum and mean values for the relevant landing parameters considering all of the simulations.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>min</th>
<th>max</th>
<th>mean</th>
<th>standard deviation</th>
<th>objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azimuth angle $\gamma'$ (deg)</td>
<td>$–1.06$</td>
<td>$0.59$</td>
<td>$0.01$</td>
<td>$0.32$</td>
<td>$0 \pm 1$</td>
</tr>
<tr>
<td>Vertical speed $V_c$ (ft/s)</td>
<td>$0.88$</td>
<td>$4.46$</td>
<td>$2.33$</td>
<td>$0.82$</td>
<td>$2.5 \pm 1$</td>
</tr>
<tr>
<td>Pitch angle $\phi$ (deg)</td>
<td>$–2.81$</td>
<td>$13.01$</td>
<td>$3.95$</td>
<td>$4.19$</td>
<td>$&gt; 0$</td>
</tr>
<tr>
<td>Pitch rate $\dot{\phi}$ (deg/s)</td>
<td>$–0.73$</td>
<td>$0.25$</td>
<td>$–0.25$</td>
<td>$0.26$</td>
<td>$&gt; 0$</td>
</tr>
<tr>
<td>Landing distance $X_l$ (m)</td>
<td>$190$</td>
<td>$567$</td>
<td>$393$</td>
<td>$115$</td>
<td>$0 &lt; X_l &lt; 500$</td>
</tr>
</tbody>
</table>

Figure 13 – Guidance and control for landing scheme.

Figure 14 – Trajectory scheme for the flare procedure.

Figure 14 – Trajectory scheme for the flare procedure.
Conclusion

Inspired by dynamic inversion techniques, an original methodology is proposed in this paper to design nonlinear controllers over possibly large flight envelopes. The procedure combines a partial feedback linearization of the plant with a fixed-structure multi-model $H_{\infty}$ design technique. Our methodology also includes a preliminary $\mu$-based validation phase, during which worst-case models are obtained and then used to enrich the set of design models. Finally, a global nonlinear robustness analysis strategy is briefly sketched and the paper concludes with a detailed application of the methodology to a realistic aircraft landing problem. Future work will be devoted to further improvements of the design strategy and its application to a small autonomous aircraft, including flight tests.

Figure 15 – Complete landing simulations in the presence of lateral wind, for various initial aircraft configurations.
References


Jean-Marc Biannic graduated from SUPAERO Engineering School in 1992 and received the PhD degree in Robust Control Theory with the highest honors in 1996 from SUPAERO as well. He joined ONERA as a research scientist in 1997 and received the HDR degree (French habilitation as PhD supervisor) from Paul Sabatier’s University of Toulouse in 2010. Jean-Marc Biannic has supervised 6 PhD students. He is the author or co-author of 20 journal papers, beyond 50 conference papers, many book chapters, teaching documents, a tutorial book on multivariable control and Matlab toolboxes. He received in 2011 the second research distinction grade (MR2) from ONERA and the “ERE” distinction from ISAE (Aeronautics and Space Institute) thanks to which he is recognized as a full-professor in PhD committees. Jean-Marc Biannic has participated to several European projects and Garteur Groups (on PIO and nonlinear control). From 2012 to 2016, he has led a research project involving 10 research scientists for the development of the SMAC toolbox (w3.onera.fr/smac) for systems modeling, analysis and control.

Jérémy Lesprier is currently working as a professional engineer in the Attitude and Orbit Control Systems Development & Studies Department at Airbus Defence and Space, Toulouse, France. He graduated from the French engineering school Arts et Métiers ParisTech with a double degree in Aerospace Engineering at KTH (Stockholm, Sweden) in 2012. He received his Ph.D degree in automatic control from the University of Toulouse in 2015, for his work on robust control laws design, robustness analysis, and modeling for aerospace applications.

Clément Roos graduated from ISAE-SUPAERO in 2004 and holds a PhD in Automatic Control. He joined ONERA as a research scientist in 2007. He takes part in industrial projects with Airbus, Dassault and Safran. He is or was also involved in several European projects such as GARTEUR-AG17, COFCLUO, CLEANSKY, ERA and VISION. His research interests focus on aircraft modeling, robustness analysis and control laws validation, as well as robust and nonlinear control. He is the author or co-author of several papers, book chapters, teaching documents and Matlab tools implemented in the Systems Modeling, Analysis and Control toolbox.