In the coming years, unmanned combat air vehicles will be designed to outperform most pilot-in-the-loop systems. The absence of a pilot onboard allows larger flight envelopes to be considered with, for example, higher load factors. From a control perspective, the above remark induces new needs for improved design methods to obtain robust controllers that will automatically adapt to extremely varying flight conditions. Based on the well-known concept of nonlinear dynamic inversion (NDI), the approach of this paper introduces an original and rather generic “robustification” framework, which leads to a multi-objective $H_\infty$ design problem. The latter is now easily solved by existing and efficient numerical tools based on nonsmooth optimization. The proposed design methodology is illustrated by a combat aircraft control problem.
a control law that achieves the desired response characteristics may be formulated as follows

\[ u = g(x, \theta)^{-1}(v - f(x, \theta)) \]

(2)

where \( v \) specifies the desired response and is generally produced as the output of a linear controller, remarking that – in the ideal case – the closed-loop system now simply reads:

\[ \dot{x} = v \]

(3)

Interestingly, when an accurate model is available, the control structure (2) compensates not only for the nonlinearities of the plant, but also for its parametric variations, as is further clarified below. Unfortunately, an exact compensation is never achieved in practice because of uncertainties in the model, because of noises in the measurements, because part of the states might not be available, because the control efficiency \( g(x, \theta) \) might be temporarily non-invertible and, finally, because of control saturations. For these reasons, making the desired response \( v \) requires special attention.

Within the context of Linear Parameter Varying (LPV) plants, the nonlinear differential equation (1) can be rewritten as:

\[
\begin{bmatrix}
\dot{x}
\end{bmatrix} = \begin{bmatrix}
A(\theta)x + B(\theta)u \\
z
\end{bmatrix} Lx
\]

where \( z \) denotes the signal to be tracked and \( B(\theta) \) is a square and non-singular matrix throughout the operating domain of the plant. Thus, the above “linearizing” control law (2) can be adapted as follows:

\[ u = B(\theta)^{-1}(B_0v - (A(\theta) - A_0)x) \]

(5)

so that the LPV plant now becomes LTI:

\[
\begin{bmatrix}
\dot{x}
\end{bmatrix} = \begin{bmatrix}
A_0x + B_0v \\
z
\end{bmatrix} Lx
\]

(6)

for which the new control input \( v \) in the following format:

\[ v = H(s)z_c + K(s)x \]

(7)

is easily designed by any standard approach. It is easily verified that the combination of equations (5) and (7) defines a standard LPV control law:

\[ u = B(\theta)^{-1}[B_0H(s)z_c + (B_0K(s) + A_0 - A(\theta))x] \]

(8)

It is very interesting here to point out that any difficulty related to the size of the parametric vector \( \theta \) has been removed, which, for many reasons, is clearly a weak point of most LPV or gain scheduling techniques. Note that the selection of the “central” matrices \( A_0 \) and \( B_0 \) is completely free. A standard choice consists in setting \( A_0 = A(\hat{\theta}) \) and \( B_0 = B(\hat{\theta}) \) where \( \hat{\theta} \) denotes a mean value of the varying parameter. However, in some cases, it might not be the best. Rather than considering mean values of the parameters, an interesting alternative consists in focusing on worst case combinations, for which the instability degree of \( A_0 \), for example, is maximized, or for which the control efficiency is minimized. The central idea is that the LTI system (6) should not necessarily capture the mean behavior of the LPV plant, but rather a worst case behavior. Unfortunately, there are still no general rules for the selection of the “central” model. This is still an open issue and it seems that the best choice will highly depend on the application. Another difficulty of the proposed approach is related to the assumptions regarding the matrix \( B(\theta) \), which must be square and invertible. In practice, these two requirements are rarely met. However, within the context of aerospace systems, a time scale separation technique can be used to bypass such difficulties [26]. This point is clarified in the following.

Application to a generic aircraft control problem

Let us consider a longitudinal aircraft short-term dynamics control problem, where the objective is to track the angle-of-attack over a large flight envelope. Using standard flight dynamics notation (see [10] for further details), the equations of interest are as follows, where \( \alpha \) and \( q \) denote the angle-of-attack and the pitch rate, respectively:

\[
\begin{align*}
\dot{\alpha} &= q + w_\alpha(\theta_p) \\
J_{yy} \dot{q} &= w_q(\theta_p) + \lambda(\theta_p) \delta_m \\
&= q_dSLC(M_{\alpha}) + q_dCLC(M_{\alpha})/V + (J_{zz} - J_{xx})pr + J_{xz}(r^2 - p^2) \\
&= q_dSLC(M_{\alpha})
\end{align*}
\]

with:

\[
\begin{align*}
w_\alpha(\theta_p) &= \frac{g}{V \cos \beta}(\cos \alpha \cos \theta \cos \phi + \sin \alpha \sin \theta) \\
&+ \alpha_z \cos \alpha - \alpha_x \sin \alpha - p_r \tan \beta \\
w_q(\theta_p) &= q_dSLC(M_{\alpha}) + q_dCLC(M_{\alpha})/V \\
&+ (J_{zz} - J_{xx})pr + J_{xz}(r^2 - p^2)
\end{align*}
\]

and:

\[
\begin{align*}
p_r &= p \cos \alpha + r \sin \alpha \\
\alpha_x &= ga_\alpha &+ L_\alpha g(q^2 - r^2) \\
\alpha_z &= ga_{\alpha z} &+ L_\alpha g(q - pr)
\end{align*}
\]

In the above representation, all parametric variations of the system (mainly induced by the variations of velocity) are captured by the parametric vector \( \theta_p \). The notation \( \theta_p \), widely used in the LPV control literature, is no longer appropriate here, since it now denotes the pitch angle (see Equation 10). The longitudinal short-term motion of the aircraft is essentially controlled through the second equation in (9), via the control input \( \delta_m \), denoting the elevator deflection. The latter also impacts the first equation through a small effect on the accelerations \( \alpha_x \) and \( \alpha_z \). However, the latter is small enough to be neglected.

Interestingly, the expressions of the nonlinear inputs \( (w_w \text{ and } w_q) \) and of the control efficiency \( \lambda(\theta_p) \) given in (10) depend on known and on-line measurable data. Consequently, these three parameter-varying terms can be used by the controller. Observing that the control efficiency verifies \( \lambda(\theta_p) < 0 \) (and is thus invertible) over the entire flight envelope, a standard NDI approach consists in inverting the moment equation, so as to control the pitch rate. Assuming that the actuator dynamics are much faster than the desired response on \( \delta_m \), they are temporarily neglected. It is then readily verified that the following control law:
\[
\delta_{mc} = \lambda(\theta_p)^{-1}(J_{yy}^{-1}(q_c - q) - w_u)
\]

yields:

\[
\dot{q} \approx \tau_q^{-1}(q_c - q)
\]

Since the pitch rate evolves much faster than the angle-of-attack, (this also can be enforced by choosing \(\tau_q\) small enough), one further considers that \(q = q_c\), so that the first equation in (9) is now controlled via \(q\), and the following choice for the commanded pitch rate:

\[
q_c = \alpha_s^2 \int_0^\infty (\alpha_c - \alpha) \, d\tau - 2\xi_\omega \alpha_s \, w_u
\]

enforces a second-order behavior on the angle-of-attack:

\[
\frac{\alpha_c(s)}{\alpha_e(s)} \approx \frac{\alpha_s^2}{s^2 + 2\xi_\omega s + \omega_\alpha^2}
\]

where the desired closed-loop pulsation \(\omega_\alpha\) is chosen as a function of the calibrated airspeed. Combining the above equations, one observes that the nonlinear parameter-varying control law may be summarized as:

\[
\begin{align*}
\dot{u}_c &= \lambda(\theta_p)\dot{\delta}_{mc} = H(s)\left(\alpha_c \, w_c, w_q\right) + K(s)\left(\alpha \, q\right) \\
H(s) &= \begin{bmatrix} J_{yy} \alpha_s^2 & J_{yy} \end{bmatrix} - \frac{J_{yy}}{\tau_q} - 1, \quad K(s) = -\frac{J_{yy}}{\tau_q} \begin{bmatrix} \alpha_s^2 & 2\xi_\omega, \alpha_s \end{bmatrix} 1
\end{align*}
\]

Note that Equation (16) can be viewed as an alternative formulation of (8). With this approach, rather sophisticated parameter-varying control laws covering the entire flight domain of interest are then easily obtained after a very short design process. However, the efficiency of such control laws relies strongly on the availability of the nonlinear inputs \(w_c\) and \(w_q\). In practice, uncertainties affect these two signals and the control efficiency \(\lambda(\theta_P)\) is not known precisely. Moreover, because of the actuator dynamics, the actual deflection \(\delta_m\) may differ sometimes significantly from the commanded variable \(\dot{\delta}_{mc}\). Within the context of dynamic inversion, several techniques have been proposed to cope with these limitations, by improving the robustness of the controller [26]. The central idea of these techniques consists in mixing the concept of dynamic inversion with robust control theory. Based on this idea, in the design procedure which is detailed next, a multi-objective \(H_\infty\) design framework is proposed to generalize the above control structure and to better optimize the gains \(K(s)\) and \(H(s)\).

### A novel tuning procedure

As mentioned above, the desired elevator deflection is not produced instantaneously, but rather is delivered by an actuator of limited capacity. Within our context, its dynamics are accurately described by a linear second-order transfer function. Here, to further simplify the following discussion, let us temporarily reduce it to a first-order system so that \(\dot{\delta}_m = \tau^{-1}(\delta_m - \delta_m)\) and let us define the new variable \(u = \lambda(\theta_p)\delta_m\). Thus, one obtains:

\[
\dot{u} = \lambda(\theta_p)\dot{\delta}_m + \dot{\lambda}(\theta_p)\delta_m = \lambda(\theta_p)\dot{\delta}_m + w_u = \tau^{-1}(u_c - u) + w_u
\]

where the commanded input \(u_c\) is defined in (16) and the perturbation term \(w_u\) may further be characterized as:

\[
w_u = \lambda(\theta_p)u = \mu(\theta_p)u
\]

Hence, the nonlinear aircraft model of equation (9), including the actuator, can be drawn as shown in figure 1, where the state-space data of the linear system \(G(s) = C(sI - A_c)^{-1}B_c\) is initially given by:

\[
A_G = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & J_{yy}^{-1} & J_{yy}^{-1} \end{bmatrix}, \quad C_G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

Figure 1 - Description of a nonlinear plant as a linear system with nonlinear inputs

**Remark** As already discussed above, the linear system \(G(s)\) should at least locally represent a realistic behavior of the aircraft, which is definitely not the case in (20), which corresponds to a double-integrator. For given values \(\theta_p\) of the varying parameters, a linearization technique yields:

\[
\begin{bmatrix} \dot{w}_c \dot{w}_q \end{bmatrix} = \begin{bmatrix} \frac{\partial w_c}{\partial \alpha} (\theta_p)\alpha + \frac{\partial w_c}{\partial q} (\theta_p)q + \dot{w}_c \\ \frac{\partial w_q}{\partial \alpha} (\theta_p)\alpha + \frac{\partial w_q}{\partial q} (\theta_p)q + \dot{w}_q \end{bmatrix}
\]

Thus, the nonlinear inputs of \(G(s)\) become \(\dot{w}_c\) and \(\dot{w}_q\) and its \(A_G\) matrix is updated as follows:

\[
A_{G}(\theta_p) = \begin{bmatrix} \frac{\partial w_c}{\partial \alpha} (\theta_p) & 1 + \frac{\partial w_c}{\partial q} (\theta_p) \\ \frac{\partial w_q}{\partial \alpha} (\theta_p) & \frac{\partial w_q}{\partial q} (\theta_p) \end{bmatrix}
\]

**Formulation as a multi-channel \(H_\infty\) design problem**

Merging the above actuator description and \(G(s)\), an augmented linear interconnection model \(M(s)\) can be obtained as the linear multi-variable transfer matrix from inputs \(w_c, \dot{w}_c, \dot{w}_q\) and \(u_c\) to the outputs \(z_c = \alpha, z_q = u\) and \(z = y\). This linear interconnection is the central element of the \(H_\infty\) design problem visualized in figure 2.

![Figure 2 - Design model](image-url)
Basically, the general idea consists in computing the best controller $K(s)$, such that a few relevant weighted transfers from $w_1$ to $z_1$ are minimized. More precisely, the transfer from $w_1$ to $z_1$ can be associated with the nominal performance, since $w_1$ corresponds to a control input on the angle-of-attack while $z_1$ denotes the error between the actual output and that of the reference model $R(s)$. Thus, the problem to be solved takes the form of the following multi-channel $H_\infty$ optimization program:

$$\min_{K(s)} \left\| \begin{bmatrix} T_{w_1 \rightarrow z_1} \end{bmatrix} \right\|_{\infty} \quad \text{with} \quad \left\| \begin{bmatrix} T_{w_1 \rightarrow z_1} \end{bmatrix} \right\|_{\infty} \leq c_{11},$$

$$\left\| \begin{bmatrix} T_{w_2 \rightarrow z_2} \end{bmatrix} \right\|_{\infty} \leq c_{23},$$

$$\left\| \begin{bmatrix} T_{w_3 \rightarrow z_3} \end{bmatrix} \right\|_{\infty} \leq c_{32},$$

$$\left\| \begin{bmatrix} T_{w_3 \rightarrow z_3} \end{bmatrix} \right\|_{\infty} \leq c_{22},$$

where constant terms $c_i$ can be tuned to quantify various robustness levels:

- $c_{11}$: stability robustness against the neglected term $w_1 = \mu(\theta)\mu$
- $c_{23}$: performance robustness against perturbations $\tilde{w}_\alpha$ and $\tilde{w}_q$
- $c_{32}$: bound on the nominal control activity
- $c_{22}$: bound on the "perturbed" (by $\tilde{w}_\alpha$ and $\tilde{w}_q$) control activity

### Interpretation and controller structure

As is visible in the diagram of figure 2 and from (23), the nonlinear control design problem has been re-formulated as a rejection problem of on-line estimated nonlinear input perturbations. Assume that a controller $K(s)$ has been computed, then the control law to be implemented will read:

$$\delta_{mc} = \frac{1}{\lambda(\theta^p)} K(s)[\tilde{w}_\alpha, \tilde{w}_q, \alpha_c - \alpha, y]'$$

The above expression generalizes (16). Here, the unique compensator $K(s)$ includes both the feedforward (previously denoted $H(s)$) and feedback (previously denoted $K(s)$) paths. Note that the signal $w$ is not used by the controller, since its estimation might be very poor (because of possibly fast variations). Moreover, it has been observed in practice that this signal is often very small. As is clear from (19), its magnitude is directly related to the rate-of-change in the varying parameters.

### Weighting function tuning procedure

The most challenging task in $H_\infty$ design approaches, once the controller structure has been defined, consists in tuning the weighting functions correctly. This step is often very tricky. Fortunately, the situation is quite favorable here, since an initial solution can be obtained by a standard dynamic inversion approach (see Equations (16) and (17)). As a result, the weighting functions can be tuned by a frequency-domain analysis of the design model in feedback loop with this standard solution. Thus, all that remains is to iterate from this starting point, to improve the standard controller. Alternatively, the tuning procedure may also be started from scratch when the standard dynamic inversion based controller fails to satisfy any of the desired closed-loop properties.

### Dealing with saturations

A key improvement axis in the above procedure consists in minimizing the control activity by "playing" on the weighting function $W_s(s)$, for example. It is then expected that magnitude and rate limits will no longer induce loss of performance or stability. In a next step, further improvements can be obtained by plugging in an anti-windup compensator $J(s)$, which can also be optimized by $H_\infty$ norm minimization, together with the previous feedback gain $K(s)$. Further details on such an approach can be found in [71] and [9].

### Resolution aspects

Last but not least, the resolution of the multi-channel $H_\infty$ optimization problem (23) deserves a few comments. Unlike standard full-order $H_\infty$ design problems, the latter is non-convex, because of the multi-channel aspect. Moreover, as shown by Figure 2, numerous weighting functions have been introduced in the design model, which contributes to a significant increase in its number of states. As a result, the optimization of a full-order controller would certainly result in non-implementable control laws. It is thus strongly recommended here to search for fixed-order and structured controllers, which is a second source of non-convexity. Until recently, these problems were very hard to solve, which certainly explains why formulations such as those stated by (23) have rarely been considered. However, over the last few years, thanks to very recent progress on nonsmooth optimization techniques [4, 5, 11], new efficient tools dedicated to the local optimization of fixed-order and fixed structure $H_\infty$ controllers have appeared. Let us first cite the public domain software HIFOO [15, 16, 14] for use with MATLAB, which is backed up by the theoretical advances described in [11]. Then, appeared the routine HINFSTRUCT, whose theoretical foundations are described in [4]. The latter has been directly integrated to MATLAB by Mathworks Inc., and is available with the Robust Control Toolbox [22].

### Results on the flight control problem

The above strategy is now applied to the flight control problem, but here both the longitudinal and the lateral axes are considered. Baseline controllers are preliminarily designed for each axis. Next, both controllers are plugged into design models, as shown in figure 2, where all weighting functions are first set to identity. A singular value analysis is then performed in order to initialize the weighting functions and the fixed-order multi-channel $H_\infty$ (23) is preliminarily solved for each axis. In each case, the initial set of weighting functions is chosen so that all constraints $c_i$ are normalized. The best achieved $H_\infty$ norm of the main transfer associated to the nominal performance also verifies: $\left\| T_{w_1 \rightarrow z_1} (s) \right\|_\infty < 1$. Then, an iterative procedure is applied to decrease the constants $c_i$, while preserving the nominal performance constraint. During this procedure, one essentially tries to minimize $c_{ii}$, which reflects the capacity of the controllers to reject the nonlinear input signals, thus extending their operating domains. For each axis, this value is approximately divided by 3. During this iterative tuning procedure, the transfer $T_{w_1 \rightarrow z_1} (s)$ from $w_1$ to $z_1$ will also receive particular attention, as well as the selection of the weighting function $W_s(s)$, in order to minimize the control activity.
Both longitudinal and lateral controllers are then implemented in a nonlinear SIMULINK diagram including a complete description of the aircraft, which remains valid over the entire subsonic flight domain. Ten simulations are then performed to evaluate both the longitudinal and lateral controllers throughout the flight domain, in which 5 points are selected (see table 1).

Table 1: Test points in the flight domain

<table>
<thead>
<tr>
<th>Point #</th>
<th>Mach number</th>
<th>Altitude (ft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.25</td>
<td>5000</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>20000</td>
</tr>
<tr>
<td>3</td>
<td>0.7</td>
<td>10000</td>
</tr>
<tr>
<td>4</td>
<td>0.9</td>
<td>36000</td>
</tr>
<tr>
<td>5</td>
<td>0.9</td>
<td>5000</td>
</tr>
</tbody>
</table>
For each of these points, the aircraft is preliminarily trimmed (initial thrust and elevator deflections are set to ensure steady state flight conditions) and two maneuvers are performed:

- **Longitudinal maneuver:** this sequence consists of two steps on the angle-of-attack. The first is applied after 1s. Its magnitude is tuned according to the flight point, so that the vertical load factor does not exceed the maximum value of 9g. Next, after 6s a new step is applied, so that the final angle-of-attack is now between 0 and \(-10\,\text{deg}\). Here again, the magnitude is adapted as a function of the flight point, so that the vertical load factor remains above its minimum value, which is fixed to \(-3g\). The total length of this maneuver is 10s. The simulation results are visible in figure 3.

- **Lateral maneuver:** during this sequence, the lateral behavior of the aircraft control laws is evaluated through their capacity to track roll-rate commands. For this purpose, a series of roll-rate steps is applied, as shown in figure 4. During these steps, the objective is to maintain the sideslip angle around 0.

**Concluding remarks**

In this paper, an original control design methodology combining the concepts of dynamic inversion and LPV control techniques has been described. The proposed strategy, which essentially consists in revisiting NDI control as a linear control problem with measured (or estimated) nonlinear disturbing inputs, is particularly well-suited to aerospace applications. The proposed design approach has been validated on a realistic and complete aircraft control problem over a large flight envelope.

A key advantage of this last parameter-varying control strategy resides in its capacity for handling many parameters without critical impact during the design process.

However, when the nonlinear input signals – which in most cases have to be estimated on-line – differ significantly from reality, it becomes difficult to predict whether the closed-loop properties will be guaranteed or not. A controller validation phase is then required. Such validations generally consist of extensive nonlinear simulations for many flight conditions, many parametric configurations and many different types of maneuvers. This unavoidable process takes a lot of time. This is why many efforts have been recently devoted to the development of numerically cheaper validation techniques for parameter-varying flight control laws. The interested reader may consult references [28, 27, 29, 12] and the book [41] and the references therein.

**Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPV</td>
<td>Linear Parameter Varying</td>
</tr>
<tr>
<td>LTI</td>
<td>Linear Time Invariant</td>
</tr>
<tr>
<td>NDI</td>
<td>Nonlinear Dynamic Inversion</td>
</tr>
</tbody>
</table>
References


AUTHORS

Jean-Marc Biannic is a senior research scientist at Onera, the French Aerospace Lab. He has supervised 6 PhD students and has authored a tutorial book on automatic control, more than 50 papers, several book chapters, teaching documents and Matlab toolboxes. He graduated from Supaero in 1992, received his PhD in control theory in 1996 and his HDR degree (french habilitation to supervise PhD students) in 2010. He received in 2011 the "ERE" distinction from Isae (Aeronautics and Space Institute) with which he is recognized as a full professor in PhD committees.

Laurent Burlion is currently working as a research scientist in the Systems Control and Flight Dynamics Department at the French Aerospace Lab. He graduated from the French engineering school Ensieta (now Ensta Bretagne) in 2003 and received the same year the M.Sc. degree in Control at the IRRCyN in Nantes. Working on hybrid systems and nonlinear sampled-data controlled systems within the L2S-CNRS, Dr. Burlion received his Ph.D degree from the University of Paris Sud in 2007. His current research interests are focused on the development of nonlinear control design methods with recent applications to visual servoing, flexible modes attenuation or manoeuvre load control.

Henry de Plinval graduated from Ecole Polytechnique, France, in 2006 and received his MSc in Aeronautics and Astronautics in 2006 from MIT (USA) and his PhD in automatic control from the University of Toulouse in 2014. Since 2008, he is with the Systems Control and Flight Dynamics Department within Onera - the French Aerospace Lab. His areas of interest include guidance, navigation and control, mostly for UAVs, with a particular focus on visual servoing for VTOL UAVs.


