A general controller design framework using $\mathcal{H}_\infty$ and dynamic inversion for robust control in the presence of uncertainties and saturations

Master Thesis Report

JEREMY LESPRIER

Degree Project in Electrical Engineering
Toulouse, France (ONERA)
June - November 2012

XR-EE-RT 2012:030
A general controller design framework using $\mathcal{H}_\infty$ and dynamic inversion for robust control in the presence of uncertainties and saturations

by Jeremy LESPRIER

This thesis deals with robust controller design using recently developed methods and tools. Starting from a nonlinear model, nonlinear dynamic inversion (NDI) is applied in order to linearize the system and deal with the varying parameters. Since the resulting closed-loop model lacks of good robustness properties, an $\mathcal{H}_\infty$ scheme is used in order to improve it, by using new MATLAB© routines also allowing to fix the structure and the order of the controller. Then a next step is to consider actuator saturations, which leads to a multi-objective anti-windup design. At the end the stability and the performance properties of the closed-loop system in the presence of linear time-invariant (LTI) and linear time-varying (LTV) uncertainties are formally evaluated using $\mu$-analysis based tools and integral quadratic constraints (IQCs). All the theory is briefly exposed for each technique and is then applied on the control of the angle of attack for a simple aircraft longitudinal model. All this framework shows interesting and satisfying results which prove the effectiveness of $\mathcal{H}_\infty$-based methods and the progress that have been made in this field.
Acknowledgements

The five months of Master Thesis presented in this report brought me a huge but far from being definitive package of knowledge concerning modern automatic control. Moreover it provided me the will to continue to work on this subject during a thesis, some rigorous methods for writing a paper and deal with a research subject, and most importantly an interesting and not close-to-the-end experience at the French Aerospace Lab, ONERA\(^1\).

This experience and the writing of the report were made possible and easier thanks to some people I would like to particularly thank here. The following non exhaustive list will try to cover all the significant human contributions. I am very grateful to:

- Patrick Fabiani, director of the System Control and Flight Dynamics department (DCSD) at ONERA, Christelle Cumer, head of the Integration and Command unit and Valérie Cassignol, department’s secretary, for having introduced me in the DCSD department and allowed me to execute this Master Thesis;

- Jean-Marc Biannic and Clément Roos, my supervisors at ONERA, for their patience, their very useful explanations and huge help, their feedbacks and all the information they have given to me during this period (and this is not over) which has brought me confidence and more interests for the thesis works that will follow;

- Elling W. Jacobsen, professor in automatic control at KTH, for his acceptation to supervise my master thesis’ work, for his feedbacks and advice on this report and for any organizational help;

- Patrik Thede, KTH’s student, for having been my opponent for the presentation, who has given good constructive comments;

- Matthieu Catalon, a very good friend who has taken some time to review this report and has given his helping impressions with an outside eye;

- my ex-colleagues in my office, David Hernandez and Loïc Maudoux, ex-interns at ONERA for the good ambiance and the many discussions.

Finally I would like to thank all the people I have met and I will continue to meet at ONERA who contributed to the good working atmosphere and who gave me any kind of useful information.

---

\(^1\)Office National d’Etudes et de Recherches Aerospatiales
## Contents

Abstract ............................................. i

Acknowledgements ................................. ii

List of Figures ....................................... iv

Abbreviations ....................................... v

Symbols ............................................. vi

Introduction ......................................... 1

1 Nonlinear Dynamic Inversion ................. 3
   1.1 General principle ................................................................. 3
   1.2 Application: longitudinal model ................................................. 4

2 $\mathcal{H}_\infty$ Control Design ............... 9
   2.1 General principle ................................................................. 9
   2.2 Weighting functions design ..................................................... 10
   2.3 Application ................................................................. 12

3 Saturations and Anti-Windup Design ......... 17
   3.1 Representation of control inputs saturations ................................ 17
   3.2 Anti-windup strategy ............................................................... 18
   3.3 Application ................................................................. 20

4 Robustness Analysis ............................... 24
   4.1 LFT: Linear Fractional Transformation ......................................... 24
   4.2 Robustness analysis for LTI parameters: $\mu$-analysis .................... 25
   4.3 Robust performance ............................................................... 27
   4.4 Application ................................................................. 27
   4.5 On IQC-based analysis: a brief overview ..................................... 29

Conclusion ............................................ 30

A $\mathcal{H}_\infty$ Control Design: Matlab Code .... 33

B Multi-objective $\mathcal{H}_\infty$ Control Design for Anti-windup control: Matlab Code 38

C $\mu$-analysis and performance: Matlab Code .... 42

Bibliography .......................................... 45
# List of Figures

1.1 Longitudinal axis (body X-axis) of a military aircraft ........................................... 5  
1.2 Eigenvalues of the system with different combinations of the parametric var-
ations (covering the whole flight domain) for (a) the open-loop with the un-
certainties, (b) the closed-loop with the designed NDI controller with no ac-
tuator’s dynamics and no uncertainties, (c) the closed-loop with no actuator’s
dynamics but with uncertainties and (d) the closed-loop with both actuator’s
dynamics and parameters uncertainties. ......................................................... 7  

2.1 Standard form for $H_\infty$ optimization problems. ............................................. 10  
2.2 Typical scheme with weighting functions for multi-channel $H_\infty$ control design. 11  
2.3 SIMULINK diagram used for designing the controller via $H_\infty$ control design  
with the MATLAB routine $\texttt{hinfsstruct}$. ................................................. 13  
2.4 Bode diagrams of the closed-loop transfer functions from exogenous inputs to 
exogenous outputs (solid lines) and inverse of weighting functions (dashed lines). 14  
2.5 Step response with parameters variations and no saturations using $H_\infty$ control  
design. .............................................................................................................. 15  
2.6 Step response with parameters variations and no saturations for a PID controller. 16  

3.1 First-order actuator model. ....................................................................................... 18  
3.2 First-order saturated actuator model. ..................................................................... 18  
3.3 General anti-windup scheme. .................................................................................. 19  
3.4 SIMULINK diagram used for multi-objective $H_\infty$ control design for nomimal and  
anti-windup controllers. ...................................................................................... 20  
3.5 Step response with rate saturation using multichannel $H_\infty$ + anti-windup  
control design. .................................................................................................. 21  
3.6 Step response with rate saturation using multichannel $H_\infty$ + degraded anti-
windup control design (75 %) ................................................................. 22  
3.7 Control deflection amplitude (a) and speed (b) responses due to a step on the  
commanded input $\alpha_c$ from 30 deg to $-5$ deg at 14 seconds. ......................... 23  

4.1 The two LFT Representations. ............................................................................... 25  
4.2 Standard interconnection form for robustness analysis (LFR) ......................... 26  
4.3 Standard interconnection for robust performance analysis. ............................. 27  
4.4 SIMULINK© diagram used to build the LFT Representation with the LFR  
TOOLBOX for $\mu$-analysis. ................................................................................. 28
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IQC</td>
<td>Integral Quadratic Constraint</td>
</tr>
<tr>
<td>LFR</td>
<td>Linear Fractional Representation</td>
</tr>
<tr>
<td>LFT</td>
<td>Linear Fractional Transformation</td>
</tr>
<tr>
<td>LMI</td>
<td>Linear Matrix Inequality</td>
</tr>
<tr>
<td>LPV</td>
<td>Linear Parameter Varying</td>
</tr>
<tr>
<td>LTI</td>
<td>Linear Time Invariant</td>
</tr>
<tr>
<td>LTV</td>
<td>Linear Time Varying</td>
</tr>
<tr>
<td>NDI</td>
<td>Nonlinear Dynamic Inversion</td>
</tr>
</tbody>
</table>
Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_n )</td>
<td>Identity matrix of size ( n \times n )</td>
<td>-</td>
</tr>
<tr>
<td>( q )</td>
<td>Pitch rate</td>
<td>( \text{deg/s} )</td>
</tr>
<tr>
<td>( u )</td>
<td>Control input</td>
<td>-</td>
</tr>
<tr>
<td>( \mathcal{F}_u )</td>
<td>Upper LFT</td>
<td>-</td>
</tr>
<tr>
<td>( \mathcal{F}_l )</td>
<td>Lower LFT</td>
<td>-</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Angle of attack</td>
<td>( \text{deg} )</td>
</tr>
<tr>
<td>( \mathbb{R} )</td>
<td>Set of real numbers</td>
<td>-</td>
</tr>
<tr>
<td>( \sigma(M) )</td>
<td>Maximum singular value of matrix ( M )</td>
<td>-</td>
</tr>
<tr>
<td>( M^T )</td>
<td>Transpose of the matrix ( M )</td>
<td>-</td>
</tr>
<tr>
<td>( M^* )</td>
<td>Conjugate transpose matrix (( = \overline{M^T} ))</td>
<td>-</td>
</tr>
<tr>
<td>( \omega )</td>
<td>Frequency</td>
<td>( \text{rad/s} )</td>
</tr>
</tbody>
</table>
Introduction

In the context of my last year of specialization in aerospace engineering at the Royal Institute of Technology (KTH, Stockholm, Sweden) a Master Thesis was done to validate some concepts learnt during the studies. Due to my interests and courses I followed I chose to work on a subject dealing with automatic control at the French aerospace lab, ONERA (Toulouse, France). The following report deals with a new framework for designing robust controllers in the presence of parametric variations and nonlinearities. This framework will use a combination of recently developed methods such as nonlinear dynamic inversion and nonsmooth $\mathcal{H}_\infty$ optimization with anti-windup compensation for saturations. The aim is to demonstrate their global efficiency and flexibility to deal with aeronautical problems using a simple but realistic military aircraft’s longitudinal axis model. The objective is to control the angle of attack in an efficient and robust way by taking into account the actuator’s constraints and the whole flight domain (parameters variations depending on flight conditions such as the Mach number, the altitude, dynamic pressure etc.). This problem can be solved using classical methods such as designing many PID controllers or using poles placements with gain-scheduling (e.g. [1]). However the new methods could bring significant improvements and could yield better performance and robustness properties with only one designed controller, which will be shown in this report.

Since analytical models can be derived, Nonlinear Dynamic Inversion (NDI) has proved its effectiveness for designing controllers in aerospace applications (see e.g. [2] and [3]). Considerable attention has been devoted to this technique since the early 1990s. This allows to compensate nonlinearities as well as parametric variations included in the initial nonlinear plant, thus to linearize it. Based on this it is then possible to design a controller using any linear methods to impose desired dynamics to the resulting decoupled linear system. For flight control it is an interesting alternative to gain-scheduling techniques mainly used in the industry which require cumbersome tuning procedures and which lack of guarantee between interpolation points. However an accurate open-loop model is never derived and an exact compensation is never reached due to uncertainties in the model and noise measurements. It
is thus of a large interest to combine nonlinear dynamic inversion with robustifying methods, e.g. $\mathcal{H}_\infty$ optimization. Based on existing tools which have been developed over the past ten years it is now possible to design fixed-order $\mathcal{H}_\infty$ controllers for example using the routine `hinfstruct` available with MATLAB© 2010b or higher. See [4] for details. This allows to have a controller with lower order (since it is fixed by the user) compared to what is normally done using standard $\mathcal{H}_\infty$ design. This report will thus in Chapters 1 and 2 explain a method for designing such a controller using NDI and $\mathcal{H}_\infty$ optimization.

However one other limiting constraint for designing controllers using NDI is nonlinearities such as saturations. In the real world the actuators can neither achieve very large amplitudes nor fast variations. Dealing with these constraints can bring significant challenge for designing controllers. Indeed saturations can make the system unstable if they are not taken into consideration during the design phase. Many methods have been developed to cope with these nonlinearities and one of the most popular is anti-windup design [5, 6], which will also be explained in this report (Chapter 3).

Afterwards it is necessary to analyze robustness of the whole system. How can it deal with uncertainties? Until which amount does it remain stable? This can be a tricky phase since uncertainties can be time-invariant or time-varying. Some efficient techniques exist especially for time-invariant parameters, e.g. $\mu$-analysis [7], which will be detailed and applied in Chapter 4. Some more sophisticated methods have been developed for linear time varying (LTV) uncertainties with bounded rates of variation using integral quadratic constraints (IQC) [8, 9] or parameter-dependent Lyapunov functions [10]. One main drawback of these latter techniques is the calculation cost which can be quite high even with powerful computers for systems with a large number of states (which is the case in industry applications). Since only the complex IQC-based method was superficially tackled during the Master Thesis, a few words concerning it will be exposed at the end of Chapter 4.

As mentioned at the beginning of this introduction all this framework will be illustrated on an aircraft’s longitudinal axis model considered and updated all along this report for better and concrete understanding of the methods.
Chapter 1

Nonlinear Dynamic Inversion

NDI is the starting point of the study case of this report. It will be used for designing a first version of the controller after having eliminated the nonlinear dynamics from the model. In fact NDI is a feedback linearization technique which decouples, linearizes and stabilizes a given nonlinear process. This introduces some auxiliary nonlinear feedback loops which allow the system to be treated as linear for control design purposes. NDI was first derived from noninteracting control [11] and from feedback linearization techniques [12]. Before NDI was of a great interest in aerospace applications for nonlinear control ([2] and [3]), linear methods such as gain scheduling were successfully used. The latter typically consists in employing a set of linearized approximations of the nonlinear process which covers the operating domain. The main problem of this technique is the validity between the interpolation points and the restrictive stability conditions [13], which gives NDI a significant advantage since it adapts automatically to the present operating point around which the system will be linearized. This chapter briefly explains the theory in a simplified way before showing the application on an aircraft’s longitudinal axis model also described for the rest of the report.

1.1 General principle

Consider the nonlinear system described in (1.1):

$$\dot{x} = f(x) + g(x)u$$

(1.1)

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input (to simplify the problem consider $m = n$, so that we have an input-affine square system) and $f(x) \in \mathbb{R}^n$ and $g(x) \in \mathbb{R}^{m \times m}$
are nonlinear function. We will also assume here that the function \( g(x) \) is invertible, i.e. \( \det(g(x)) \neq 0 \ \forall x \in \mathbb{D} \subset \mathbb{R}^m \).

Nonlinear Dynamic Inversion (NDI) allows to linearize and decouple a nonlinear system via state-feedback, using directly the functions \( f \) and \( g \). This implies that the full state \( x \) is available for the controller which can be a little restrictive in a first sight. Thus a control law that achieves the desired response can be written as follows:

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

(1.2)

where \( v \) specifies the desired response and can be designed using any linear approach. Then in the ideal case we have \( \dot{x} = v \) which is very interesting when an accurate model is available. It also supposes that the full state \( x \) is available to the controller. For instance if \( x_c \) is the commanded state, \( v \) can be of the following form:

\[
  v = C(s) \begin{bmatrix} x_c \\ x \end{bmatrix}
\]

(1.3)

The linear controller \( C(s) \) can be chosen so that it deals with the desired system requirements: having no static error, no overshoot, compensate inversion errors or having good robustness properties.

### 1.2 Application: longitudinal model

The control problem here concerns the control around the longitudinal axis of a rigid military aircraft, represented by the \( X \)-axis (body) in Figure 1.2 with the angle of attack \( \alpha \) depicted. This problem will be considered and updated all along the thesis depending on the treated method. The objective is to command the angle of attack through the elevators. The open-loop behavior is described by (1.4).

\[
\begin{align*}
  \dot{\alpha} &= w_\alpha + q \\
  \dot{q} &= w_q + m_\delta \delta_m
\end{align*}
\]

(1.4)

where \( \alpha \) denotes the angle of attack, \( q \) the pitch rate, \( \delta_m \) the elevator deflection, \( m_\delta \) the deflector efficiency (strictly negative, and thus invertible on the whole flight domain) while \( w_q \) and \( w_\alpha \) correspond to nonlinear parameters functions depending mainly on dynamic pressure and airspeed, and also on \( \alpha \) and \( q \).
Chapter 1. Nonlinear Dynamic Inversion

Figure 1.1: Longitudinal axis (body X-axis) of a military aircraft

We have a description of the parameters \( w_\alpha \) and \( w_q \) given by (1.5).

\[
\begin{align*}
    w_\alpha &= z_\alpha \alpha \\
    w_q &= m_\alpha \alpha + m_q q
\end{align*}
\]  

with:

\[
\begin{align*}
    z_\alpha &= - (1 + 0.6 \theta_{z_\alpha}) (1.3 + 0.7 \theta_v) \\
    m_\alpha &= 0.8 + (\theta_{m_\alpha} + 1.5 \delta_{m_\alpha}) (5 + 4 \theta_v) \\
    m_q &= - (1 + 0.35 \theta_{m_q}) (0.9 + 0.45 \theta_v)
\end{align*}
\]  

\( \theta_v \) is a normalized parameter directly corresponding to the airspeed, -1 corresponding to 100 m/s while 1 corresponds to 300 m/s, \( \theta_{z_\alpha} \), \( \theta_{m_\alpha} \) and \( \theta_{m_q} \) correspond to normalized parametric variations. They are equivalent to perturbation terms which represent the variations of other flight parameters such as the Mach number and the altitude and are used to cover the whole flight domain when varying from -1 to 1. The parameter \( \delta_{m_\alpha} \) is an uncertainty which is not accessible to the controller. This representation allows to cover the whole flight domain with a pessimistic approach. The deflector efficiency \( m_\delta \) is also dependent on variations in \( \theta_v \) following (1.7):

\[
m_\delta = -40 \left( 1 + 0.2 \delta_{m_\delta} \right) \left( 1 + 1.15 \theta_v + 0.35 \theta_v^2 \right)
\]  

where \( \delta_{m_\delta} \) is another uncertainty. The considered flight domain is large since the Mach number can vary from 0.2 to 0.9 and the altitude from 5000 ft to 40000 ft. The acceptable range for \( \alpha \) is around \([-30, 30]\) degrees (with reasonable transitory overshoots). The flight domain was set so that the model equations are valid. An even larger domain will be considered using this framework during the thesis that will follow.

Now considering all this we can apply nonlinear dynamic inversion in order to find a general expression for the controller to design.
Starting from (1.4) the main objective is to obtain an expression for the control variable $\delta_m$ depending on the commanded variable $\alpha_c$. We first have, inverting the model equation, that:

$$\delta_m = m_\delta^{-1}(\dot{q} - w_q)$$

(1.8)

Considering that the pitch rate $q$ has a dynamic similar to a first order, i.e. $\dot{q} = \frac{1}{\tau_q}(q_c - q)$ where $q_c$ is the commanded pitch rate and $\tau_q$ is a function of $\theta_v$: $\tau_q = 0.4/(7 + 3\theta_v)$, we thus have:

$$\delta_m = m_\delta^{-1}\left(\frac{1}{\tau_q}(q_c - \dot{\alpha} + w_\alpha) - w_q\right)$$

(1.9)

A physical dynamic behavior approximation for the angle attack is given by a second order:

$$\frac{\alpha}{\alpha_c}(s) = \frac{\omega_r^2}{s^2 + 2\xi\omega_r s + \omega_r^2}$$

(1.10)

giving that $\dot{\alpha} = \omega_r^2 \int (\alpha_c - \alpha) - 2\xi\omega_r\alpha$, with $\omega_r$ the pulsation chosen as a function of the normalized calibrated airspeed $\theta_v$: $\omega_r = 3\theta_v + 7$. Using this expression combined with (1.9) we finally get:

$$\delta_c = m_\delta^{-1}\left(K(s) [\alpha q]^T + H(s) [\alpha_c w_\alpha w_q]^T\right)$$

(1.11)

where $\delta_c$ is the commanded elevator deflection and:

$$\left\{\begin{array}{c}
K(s) = \frac{1}{\tau_q}\left[2\xi\omega_r + \frac{\omega_r^2}{s} 1\right] \\
H(s) = \left[\frac{-\omega_r^2}{\tau_q s} \frac{1}{\tau_q} - 1\right]
\end{array}\right.$$ 

(1.12)

With this expression we have a parameter-varying control law which works well with the defined model with imposed desired dynamics for the pitch rate and the angle of attack. This relies on the availability for the controller of the signals $w_\alpha$ and $w_q$. One must see that this also depends on $m_\delta$ which will make the control more or less efficient depending on the calibrated airspeed. Such a control is thus strongly influenced by uncertainties on the varying parameters. Moreover the deflection $\delta_m$ differs from the commanded one $\delta_c$ since there is actuator’s dynamic:

$$\delta_m = \frac{\omega_a^2}{\omega_a^2 + 2\xi_a\omega_a s + s^2}\delta_c$$

(1.13)

where $\omega_a = 60\text{rad/s}$ and $\xi_a = 0.6$ in order to reflect the real actuator’s dynamics. In the following pictures (Figure 1.2) are displayed the eigenvalues of the system on a parametric grid for all possible combinations of the values $\theta_{z_a}$, $\theta_{m_a}$, $\theta_{m_q}$ and $\theta_v$, varying from $-1$ to $1$ with a step of $0.5$. First Figure 1.2a shows the eigenvalues of the open-loop system with the normalized uncertainties $\delta_{m_a}$ and $\delta_{m_q}$ included. It clearly appears that the open-loop system
is unstable for some flight conditions and uncertainties values since some eigenvalues are in
the right-half plane. Then Figures 1.2b-1.2d show the plots for the closed-loop system with
the previously designed controller with or without the normalized uncertainties \( \delta_{m_a} \) and \( \delta_{m_d} \).
One can thus see that with the nominal model the system is fast and very stable (Figure
1.2b). However with uncertainties (absolute values bounded by 1) and then with the control
input actuator described by the second order model in equation (1.13) some worst cases for
which the closed-loop poles are poorly damped can be clearly identified (Figure 1.2c and
1.2d).

![Figure 1.2: Eigenvalues of the system with different combinations of the parametric vari-
ations (covering the whole flight domain) for (a) the open-loop with the uncertainties, (b)
the closed-loop with the designed NDI controller with no actuator’s dynamics and no un-
certainties, (c) the closed-loop with no actuator’s dynamics but with uncertainties and (d)
the closed-loop with both actuator’s dynamics and parameters uncertainties.]

As a conclusion one strong limitation of dynamic inversion is its lack of robustness. To cope
with this limitation a generalization of this control structure combined with \( \mathcal{H}_\infty \) optimization
will be proposed in the next chapter.
Chapter 2

$\mathcal{H}_\infty$ Control Design

2.1 General principle

The main objective of the $\mathcal{H}_\infty$ control design is to have a controller both having good performance and good robustness properties following some user-defined criteria. $\mathcal{H}_\infty$ control design is based on the minimization of the largest singular value of the closed-loop system $G_{ec}(s) = \mathcal{F}_l(P(s), C(s))$ containing the nominal plant and the controller (see Figure 2.1 and Chapter 4 for details about lower LFT $\mathcal{F}_l(M, N)$).

\[
\|G_{ec}(s)\|_\infty = \max_{\omega \in \mathbb{R}} \sigma(G_{ec}(i\omega)) \tag{2.1}
\]

In particular we will search for a controller minimizing $\gamma$ such that:

\[
\|G_{ec}(s)\|_\infty \leq \gamma \tag{2.2}
\]

The $\mathcal{H}_\infty$ design objective will thus be to find the minimum $\gamma$ such that (2.2) holds and thus to seek for the best optimized controller $C(s)$ since the plant $P(s)$ is fixed. In the history of $\mathcal{H}_\infty$ design two approaches have been developed to solve this problem. The first one, the classical approach, uses coupled Ricatti equations or Linear Matrix Inequalities (LMI). The obtained controller is generally full-order and unstructured (see [14]). The most recent approach, based on nonsmooth optimization, consists in directly solving the problem without putting it in a convex form. This uses some complex algorithms with generalized gradients and bundling techniques suited for the $\mathcal{H}_\infty$-norm and other nonsmooth performance criteria. The method, rather complex and not described in details here since it is not the goal of this report, is explained in [15] and in the references therein. This generally works fine.
for high-order systems and allows to have a structured controller with fixed-order and to simultaneously consider many weighted transfers (multi-channel optimization), which is of high interest for our control problem. This approach is thus easier to implement and to construct (especially thanks to new existing routines on MATLAB©, see [4]) and will be used here.

Thus after applying some NDI as shown in the previous section we can design a controller described by (1.3), with $C(s)$ to be tuned and with fixed order. Consider the standard form with output-feedback of Figure 2.1. $P(s)$ represents the augmented plant (open-loop plant + actuator + sensors + weighting functions) specifying the $H_\infty$ control problem, $C(s)$ the controller to be tuned, $w$ the exogenous inputs (e.g. commanded inputs and nonlinear parameters without the uncertainties), $z$ the weighted exogenous outputs (error between the output $y$ and the reference for instance), $u$ the control input and $y$ the available states for the controller.

\[
\begin{align*}
\begin{bmatrix}
w \\
y
\end{bmatrix} & = \begin{bmatrix} P(s) & C(s) \end{bmatrix} \begin{bmatrix} w \\
u \\
y
\end{bmatrix} \\
\end{align*}
\]

**Figure 2.1:** Standard form for $H_\infty$ optimization problems.

One tricky task is to use some well-chosen weighting functions in order to design a controller having both good performance and robustness capabilities (a trade-off has to be found depending on the requirements, since it is impossible to have both on all frequencies, see next section). The problem can then be solved using existing tools on MATLAB© especially with the MATLAB© CONTROL DESIGN TOOLBOX.

### 2.2 Weighting functions design

The $H_\infty$ control problem consists in specifying the transfer functions to be minimized. This is done by tuning some so-called weighting functions which specify the frequency domains where the minimization should mainly occur in order to give the final system the desired characteristics that will shape the closed-loop.

This task is often difficult but essential since the weighting functions will reflect the requirements concerning stability and performance. These requirements can be translated from the
time domain (rise time, step overshoot and settling time) to the frequency domain (cut-off frequency and frequency gains). It is rather impossible to achieve both robustness and performance requirements at all frequencies since the closed-loop must have high gain for good setpoint tracking and good disturbance rejection, and low gain to obtain sufficient stability margins and be insensible to neglected dynamics or noises. Typically in order to ensure good disturbance rejection and performance level the closed-loop gain should be higher at low frequencies. On the other hand the noises or neglected dynamics appear at high frequency, which will involve a low gain at these bandwidths.

![Figure 2.2: Typical scheme with weighting functions for multi-channel $H_\infty$ control design.](image)

Given the scheme of Figure 2.2 with $\Sigma(s)$ the open-loop system, $A(s)$ the actuator, $w_c$ the commanded input, $w_f$ some perturbations and $\tilde{w}$ the nonlinear parameters, one can see the three considered weighting functions in dashed lines boxes $W_w(s)$, $W_u(s)$ and $W_p(s)$. Since the main focus here is the minimization of the influence of the nonlinearities $\tilde{w}$ on the closed-loop system and the tracking of a reference model (given by $R(s)$ and equation (1.10)) the weighting functions can be designed as follows:

- $W_p(s)$ weights the output $z_e$ which is the tracking error. It should be chosen so that the low-frequency gain is high and the high-frequencies are well attenuated. This will ensure a good performance level at low frequencies since the optimization process will seek to minimize the size of the error signal between the reference model $R(s)$ and the control objective in the desired bandwidth.

- $W_u(s)$ weights the output $z_u$ which is the control input. It should be chosen so that the high-frequency gain is high. The optimization process will thus diminish the size of the control input signal in the desired bandwidth thus reducing the activity at high frequencies.
Chapter 2. $\mathcal{H}_\infty$ Control Design

- $W_w(s)$ is used as a filter for the exogenous nonlinear inputs $\tilde{w}$ (parametric variations). It translates the fact that the system nonlinearities are not significant on an infinite frequency spectrum. Indeed the nonlinear dynamics are mainly present at low-frequencies which could involve $W_w(s)$ to be a low-pass filter.

2.3 Application

In the previous part on our longitudinal axis model (see Section 1.2 in Chapter 1) the NDI technique was described leading to a controller with good performance but with poor robustness. Using the theory described above the goal was to tune weighting functions adapted to our model and to design a new controller with state-feedback using an $\mathcal{H}_\infty$ method. In our example parameters variations are not taken into account during the design phase and a nominal model on which the controller will be computed is thus defined. The good robustness and performance properties of this controller will then be assessed later on the whole flight domain. Using (1.6) and (1.7) with uncertainties and variations equal to zero and a fixed value $\theta_{v0}$ of $\theta_v$ we get nominal values $\bar{z}_{\alpha 0}$, $m_{\alpha 0}$, $m_{q0}$ and $m_{q0}$. The nominal value for $\theta_v$ is based on a worst-case design, i.e. for large speeds ($\theta_{v0} = 1$). Figure 2.3 displays the Simulink diagram used for the design of such a controller. The blocks called AC LONG and DYN PARAM represent the equations (1.4) and (1.5) respectively (with nominal values as stated above). The block REF1 corresponds to the reference to be followed, represented by equation (1.10). The exogenous inputs and outputs are in blue color while those going from and into the controller are in yellow. All the other blocks correspond to the weighting functions. They have been chosen in the following way, given what was discussed in the previous section:

$$W_w = W_{\tilde{w} \rightarrow z_e} = \frac{\bar{W}}{1 + \tau_{\tilde{w}} s} \quad (2.3)$$

$$W_p = W_{\epsilon \rightarrow z_e} = \frac{K_i (\tau_1 s + 1)}{\tau_2 s + 1} \quad (2.4)$$

$$W_u = W_{u \rightarrow z_u} = \frac{K_{rob} (\tau_{rob1} s + 1)}{\tau_{rob2} s + 1} \quad (2.5)$$

with $\bar{W}$, $K_i$ and $K_{rob}$ some gains, $\tilde{w}$ one of the inputs $w_\alpha$ or $w_q$, $\tau_i$ and $\tau_{rob}$ some time constants ($i \in \{1, 2\}$). $W_w$ focuses on the influence of the nonlinear inputs $w_\alpha$ and $w_q$ on the performance error $z_e$. Since they are slowly varying with time there is no need to consider them on the whole frequency grid. Thus a low-pass filter was chosen with a cut-off frequency around 1-2 rad/s. $W_p$ was also chosen as a low-pass filter since it directly focuses on performance (minimal error between the reference input $\alpha_r$ and the resulting output $\alpha$).
The analysis of the open-loop model gives an approximation of the time constant value $\tau_2$ which corresponds approximately to the inverse of the bandwidth of the system, while $\tau_1$ was chosen such that $\tau_2 \gg \tau_1$. $W_u$ was chosen to limit the high frequencies corresponding to high and fast control input solicitations, so as a high-pass filter ($\tau_{\text{rob}1} \gg \tau_{\text{rob}2}$).

As mentioned in the previous section these choices allow to keep good tracking of the reference and also the minimization of the influence of the non-linearities for robustness (whole flight domain considered). Thus the final closed-loop system is expected to have both good robust performance and robust stability.

Using the routine `hinfstruct` from the Robust Control Toolbox in combination with the diagram in Figure 2.3 it is possible to compute the best first-order controller $C(s)$ such that each weighted transfer (from $\dot{\bar{w}}$ to $z_e$, from $\alpha_c$ to $z_e$ and from $u$ to $z_e$) is minimized. This leads to a first-order controller with 7 inputs and 1 output. The 7 inputs are here the nonlinear inputs $w_\alpha$ and $w_q$, the error between the commanded input $\alpha_c$ and the computed $\alpha$, the one between the reference input $\alpha_r$ and the computed $\alpha$, the integral of this error, the calculated pitch rate $q$ and the variation speed of the reference model $\dot{\alpha}_r$. The more information the controller can get the better the performance and the robustness, and we
Chapter 2. $H_\infty$ Control Design

suppose that all this information can be obtained in the process.

Many trial-and-errors on the weighting functions parameters values were necessary before having a priori good results. Some singular values analysis of the weighted transfers were necessary in order to chose the right parameters in the weighting functions. The goal was mainly to normalize the performance and disturbances rejection transfers while trying to limit as much as possible the control input (which was not necessary normalized). The MATLAB code is available in Appendix A. Figure 2.4 shows the transfer functions from all inputs to all exogenous outputs for the closed-loop system (solid lines) as well as the inverse of the weighting functions used to design the expected frequency responses (dashed lines). The specifications are well fulfilled but with a low constraint on the control input ($K_{rob}$ chosen small such that it does not limit the amplitude of control needed). The final $\gamma$ obtained (equation (2.2)) is equal to $\gamma = 0.8$ which is really good since we have $\gamma < 1$ (which proves we have all the requirements fulfilled concerning the constraints).

Figure 2.5 presents the simulation using the designed controller. Note that no actuator saturations are considered yet since it will be detailed in Chapter 3. Two steps of respectively $+30$ degrees and $-35$ degrees are given on the commanded $\alpha_c$ at 1 second and 14 seconds.
respectively. The plot shows the curves for five values of the longitudinal normalized airspeed $\theta_v$ (from 100 m/s to 300 m/s). The normalized parameters $\theta_{m_a}$, $\theta_{m_q}$ and $\theta_{z_a}$ are also slowly varying with time. Their influence is clearly well taken into account (very small variations of the angle of attack during the whole simulation). A step on the uncertainties $\delta_{m_q}$ and $\delta_{z_a}$ appears at 10 seconds in order to see the influence which is again very well counteracted by the controller with almost no loss of performance.

What is interested to see in this case is a comparison with a simple PID controller, quickly designed using tools from MATLAB® (PID Control Design). For the same configuration the result is displayed on Figure 2.6. Even if a PID is at least as efficient as the $\mathcal{H}_\infty$ designed controller it is much less robust. The parameters variations are not well countered and the uncertainties yield considerable performance losses (see the $-35$ degrees second step at 14 seconds). One can also conclude that one PID controller is not enough and, for the same robust properties as the only one $\mathcal{H}_\infty$ previously designed controller, many PID controllers should probably be tuned.
Figure 2.6: Step response with parameters variations and no saturations for a PID controller.
Chapter 3

Saturations and Anti-Windup Design

In previous chapters NDI and $\mathcal{H}_\infty$ control design were mentioned in order to satisfy some given objectives concerning both performance and robustness. The simulations on our control problem have shown that combining both can achieve very satisfying results. However in real life there are actuators dynamics and saturations which limit the control inputs. Putting these magnitude and rate saturations in the simulation model with the previously designed controller leads to instability and in particular to windup problems. This phenomenon occurs when an integrator gets large output values during control signal saturation (constant value being integrated) which yield either larger response time or even instability. This chapter will briefly explain the modeling of saturations and the technique used to cope with them. Anti-windup design, as it is called, was a method rigorously developed in the beginning of the 90’s (see e.g. [16]). This strategy will be performed in our longitudinal axis control problem with saturations implemented, showing its efficiency.

3.1 Representation of control inputs saturations

Generally it is a good approximation to model an actuator by a first-order transfer function like in (3.1) with $\tau_a$ chosen small.

$$A(s) = \frac{1}{\tau_a s + 1} \quad (3.1)$$

A block diagram representation of this is shown in Figure 3.1. $u_c$ is the commanded control input while $u_r$ is the real output delivered by the actuator.
Chapter 3. Saturations and Anti-Windup Design

It is thus very simple to take into account in this diagram the amplitude and rate saturations. The new diagram displayed in Figure 3.2 shows the new representation with only static nonlinearities.

The saturation block can be explicitly described by the following equation (3.2), assuming that it is symmetric (which will be the case for our control problem).

\[
    u_{cL} = \begin{cases} 
    L & \text{if } u_c > L \\
    u_c & \text{if } |u_c| \leq L \\
    -L & \text{if } u_c < -L 
\end{cases}
\]  

(3.2)

### 3.2 Anti-windup strategy

One of the best ways to cope with saturations is to use an anti-windup strategy which has been proved to be very efficient. The basic principle is to limit the effects of saturations by introducing a new controller \( J(s) \). This controller acts only when the saturation block is active and then injects a correction signal into the nominal controller \( C(s) \) and into the commanded control input \( u_c \). This correction decreases the amplitude of the signal delivered by the controller when saturations happen [17]. However in order to take into account the saturations measurements which are not necessarily available the anti-windup control loop must contain what is called a control limiter. This limiter will mainly contain the saturation blocks inside the anti-windup control loop just before the ”real” actuator block \( A(s) \). The anti-windup control loop integrated with the actuator \( A(s) \), the nominal controller \( C(s) \) (also
augmented with the anti-windup output) and the open-loop system $\Sigma(s)$ are shown in Figure 3.3.

As a consequence the nominal controller $C(s)$ will contain another entry which corresponds to the output of the anti-windup control $J(s)$. In a very standard approach the anti-windup controller is optimized after the nominal controller in a two-step formulation. This method will not be described here but the focus will be on a new method using the routine `hinfsyn` of MATLAB®. This will allow to simultaneously design the feedback controller $C(s)$ and the anti-windup compensator $J(s)$ using multi-objective $\mathcal{H}_\infty$ synthesis ([6] and mostly [5]).

The same logic as in Chapter 2 will be used. However there will be two blocks to minimize via $\mathcal{H}_\infty$ norm: $G_{ec,nom}$ and $G_{ec,aw}$. $G_{ec,nom}$ contains the nominal augmented plant with no anti-windup control loop focusing only on nominal performance (it is more or less the same as $G_{ec}$ in Chapter 2). $G_{ec,aw}$ denotes the plant also including anti-windup related signals but without considering the performance transfers. Both $G_{ec,nom}$ and $G_{ec,aw}$ are in LFR (Linear Fractional Representation) form (see [18] and Chapter 4 for more details) with:

$$G_{ec,nom}(s) = F_l(P_{nom}(s), C(s))$$
$$G_{ec,aw}(s) = F_l(P_{aw}(s), \text{diag}(J(s), C(s)))$$

where $F_l(M, N)$ denotes the lower Linear Fractional Transformation and is equal to $F_l(M, N) := M_{11} + M_{12}N(I - M_{22}N)^{-1}M_{21}$ for two matrices $M$ and $N$ with appropriate dimensions. $P_{nom}(s)$ and $P_{aw}(s)$ are the corresponding plants with the well-chosen inputs and outputs. Thus the multi-objective formulation is the following: find $\hat{C}(s)$ and $\hat{J}(s)$ such that:

$$\left(\hat{C}(s), \hat{J}(s)\right) = \operatorname{argmin}_{C(s),J(s)} \left\{ \begin{array}{l} \|F_l(P_{nom}(s), C(s))\|_\infty \leq c \\ \|F_l(P_{aw}(s), \text{diag}(J(s), C(s)))\|_\infty \leq c \end{array} \right. $$

(3.5)

$\hat{C}(s)$ and $\hat{J}(s)$ are then the optimized controllers to implement.

**Figure 3.3:** General anti-windup scheme.
3.3 Application

Going back to our longitudinal axis control problem the objective is to define a new $\mathcal{H}_\infty$ procedure in order to take into account what was discussed in the previous section, and especially equation (3.5). We still have our nominal model derived via NDI techniques and still looking for the most performant and robust controller. But this time some anti-windup control must be added to take into account the saturations. In reference to what was done in Chapter 2 a new Simulink© diagram is created with new inputs and outputs, some linked to the anti-windup controller $J$ (orange in Figure 3.4) and some others linked to the "Rate Limiter" block (blue). Indeed only rate saturation is considered here (not the amplitude), and the actuator has a rate limitation of $L_r = 60 deg/s$. The whole Simulink© diagram used for the hinfstruct routine is shown Figure 3.4.

Thus with reference to this figure the plants $P_{nom}$ (and $P_{aw}$) previously mentioned have inputs [1 2 3 6] (resp. [4 5 6]) and outputs [1 2 6 7 8 9 10 11 12] (resp. [3 4 5 6 7 8 9 10 11 12]). The weighting functions are still the same as in Chapter 2 with one added for limiting the anti-windup control (with output $z_{aw}$), otherwise the impact of the anti-windup

![Simulink Diagram](image-url)
controller was too strong, which led to a much slower response. The nominal controller $C(s)$ was chosen as a first-order model while the anti-windup controller as a static gain $J(s)$ for simplicity (some tests were done with a first-order anti-windup controller leading to no significant improvements). The added weighting function mentioned above (to $z_{aw}$) was thus a simple static gain. The whole MATLAB® code with used values is available on Appendix B.

The results are shown in Figure 3.5. One can see that the response is slower compared to Figure 2.5, which shows that the rate saturation is well taken into account. However it exhibits quite good performance and no instability. As an interesting comparison the step responses with a degraded anti-windup control ($J' = 0.75J$) are shown in Figure 3.6. This clearly shows the efficiency of the anti-windup controller since in this figure there are much more oscillations or even instability for a value of $\theta_v$ equal to $-1$ (and we know that without any anti-windup controller the system would be unstable for any $\theta_v$).

Figure 3.7 shows the control deflection response for the second step on $\alpha$ from $30\text{deg}$ to $-5\text{deg}$ in terms of magnitude ($\delta_m$, Figure 3.7a) and of speed ($\dot{\delta}_m$, Figure 3.7b). This clearly shows the limited rate of variations of the control input (defined as $L_r = 60\text{deg/s}$) and that the magnitudes are in the defined reasonable range even with a large commanded step. The final value of $\gamma$ after the optimization process via hinfsstruct is good since it is equal to
Figure 3.6: Step response with rate saturation using multichannel \( \mathcal{H}_\infty \) + degraded anti-windup control design (75\%).

\( \gamma = 1.01 \) (good completion of specifications since its value is near to 1). As a conclusion this method yields very interesting results and since it is based on fixed-order and structured \( \mathcal{H}_\infty \) control design it is quite simple to implement.
Figure 3.7: Control deflection amplitude (a) and speed (b) responses due to a step on the commanded input $\alpha_c$ from 30 deg to $-5$ deg at 14 seconds.
Chapter 4

Robustness Analysis

When designing a controller for a specific model containing some uncertainties, parameters variations or nonlinearities it is necessary to check the global stability for the whole domain of variations. A lot of research has been devoted during the past few years to take into account many kinds of uncertainties (LTI and LTV) and nonlinearities (e.g. saturations). A specific technique has reached a maturity phase for dealing with structured LTI uncertainties: $\mu$-analysis. This allows to evaluate the robustness and performance properties of a LTI system with structured LTI uncertainties such as parametric uncertainties, neglected dynamics or delays. This technique will be described in this chapter after having introduced some necessary notions about LFTs. Like the previous chapters it will then be applied on our longitudinal axis control problem. For LTV parameters or nonlinearities $\mu$-analysis can no longer be applied. Some other techniques are then applied which rely on IQC or parameter-dependent Lyapunov functions. They are still on research phase, since even if many improvements have been made during the past few years [10], they are still hard to adapt for industry applications (too many states in the models imply too much computation time). Only IQC-based analysis will briefly be described in section 4.5 since it will be important for a future thesis work.

4.1 LFT: Linear Fractional Transformation

The LFT Representation (or LFR) plays an important role in both linear and nonlinear control theory. Two types of linear fractional transformation exist both depicted in Figure 4.1.
LFTs can be defined as lower or upper depending on the part of matrix $M(s)$ where exogenous inputs and outputs are connected. Considering a matrix $M$ such that:

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \quad (4.1)$$

the lower and upper LFTs are written respectively as:

$$\mathcal{F}_l(M, N_2) = M_{11} + M_{12}N_2(I - M_{22}N_2)^{-1}M_{21} \quad (4.2)$$

$$\mathcal{F}_u(M, N_1) = M_{22} + M_{21}N_1(I - M_{11}N_1)^{-1}M_{12} \quad (4.3)$$

More details about LFTs are given in [19].

Putting a system under LFT form is sometimes not an easy task. However some tools are now available and some progress has been made on the subject. This is for instance the case with the LFR Toolbox for MATLAB/SIMULINK© developed by S. Hecker, J-F Magni and A. Varga [20] which allows to simplify the construction of the $M - \Delta$ diagram shown in the next section for robustness analysis.

### 4.2 Robustness analysis for LTI parameters: $\mu$-analysis

For many robustness analysis techniques the transformation using LFT is a required step which consists in isolating the uncertainties and the nonlinearities in a so-called $\Delta$-block. This leads to the interconnection in Figure 4.2. Once the system has been put under LFT form it is possible to use $\mu$-analysis to study robustness with respect to parametric uncertainties contained in the $\Delta$-block. The $\Delta$-block is of the following form in the case of $\mu$-analysis (equation (4.4)):

$$\Delta(s) = \text{diag}(\delta_1 I_{k_1}, \ldots, \delta_r I_{k_r}, \Delta_1(s), \ldots, \Delta_q(s)) \quad (4.4)$$
with $\delta_i$ representing $k_i$-repeated real scalars for parametric uncertainties and $\Delta_i(s)$ denoting stable transfer matrices for neglected dynamics. The goal with $\mu$-analysis is to find the smallest perturbation $\Delta$ with admissible structure $\Delta$ for which the matrix $I - M(j\omega)\Delta$ is singular (i.e. the smallest perturbation $\Delta$ which brings an eigenvalue of the interconnection of Figure 4.2 on the imaginary axis (unstable system)). This quantity is defined as the structured singular value $\mu_\Delta(M(j\omega))$. Once it is computed on the whole frequency range we can define the robustness margin $k_{max}$:

$$k_{max} = \left[ \max_\omega \mu_\Delta(M(j\omega)) \right]^{-1} \quad (4.5)$$

The stability of the interconnection of Figure 4.2 is then ensured for all $\Delta$ with admissible structure such that $\sigma(\Delta) \leq k_{max}$. Thus if $k_{max}$ is more than one the stability is ensured for all $\Delta$ in the unit ball. Computing the exact value of $\mu_\Delta$ is non-polynomial (NP) hard in the general case, but both upper and lower bounds can be determined using polynomial time algorithms. An upper bound provides a guaranteed but conservative value of $k_{max}$, while a lower bound allows to measure conservatism. If the gap between the upper and the lower bound of $\mu_\Delta$ is low then the conservatism is very limited which is a good point. The algorithms used to compute these bounds are described in [7].

Concerning the practical aspect the computation of the bounds can be done using MATLAB© and some functions available in the Skew Mu Toolbox by J.-M. Biannic and G. Ferreres developed at ONERA (available online at [21]). This uses algorithms described in [7] which yield very efficient results in terms of computation time and conservatism even for high-order systems. This will be demonstrated in our control problem in section 4.4.
4.3 Robust performance

After having evaluated the robustness of the system one interesting further step is robust performance analysis. The robust performance problem consists in computing the largest $\mathcal{L}_2\text{-induced norm}$ of the transfer $T_{y_r \to y}$ between the reference input and the computed output in presence of structured uncertainties $\Delta(s)$. The considered interconnection is defined in Figure 4.3. Assuming that the robust stability of the interconnection of Figure 4.2 is ensured, i.e. $k_{max} > 1$ (see previous section), the performance level $\gamma_{opt}$ is thus defined as the smallest value $\gamma$ such that $\|T_{y_r \to y}\| < \gamma$ for every structured operators $\Delta$ satisfying $\sigma(\Delta) \leq 1$. The algorithm used is again described in [7] and will not be detailed here. For practical aspects the same MATLAB© toolbox as before is used.

\[ \Delta(s) \]
\[ M(s) \]
\[ w \]
\[ y_r \]
\[ z \]
\[ y \]

**Figure 4.3:** Standard interconnection for robust performance analysis.

4.4 Application

Using $\mu$-analysis implies that only LTI uncertainties can be considered. This means that the saturations and time variations are not taken into account in this part for our longitudinal axis control problem. Another method will be briefly explained in the next section for the latter. The first objectives were thus to put the system into LFT form with a $\Delta$ block containing only LTI parametric uncertainties. This was done using the LFR Toolbox with both MATLAB© and SIMULINK© programs. The SIMULINK© diagram to use with the LFR Toolbox is in Figure 4.4 while the MATLAB© code for robustness and performance analysis is available in Appendix C. The composition of the $\Delta$-block in our application problem is on the following form ($11 \times 11$ block):

\[ \Delta = \text{diag}(\delta_{m_{1}}, \delta_{m_{4}}, I_{1}, \delta_{m_{a}}, I_{1}, \theta_{m_{q}}, I_{1}, \theta_{v}, I_{6}, \theta_{z_{a}}, I_{1}) \]  \hspace{1cm} (4.6)

One should notice that $\theta_{v}$ is repeated 6 times since it is present in many equations in the model while the other parameters appear only once. The upper and lower bounds of $\mu$ are
shown in Table 4.1. This needed less than 50 iterations for the upper bound computation and the computational time was also very short (i.e. less than 10 seconds). Furthermore the gaps between the lower and upper bounds are largely tolerable which shows good conservatism properties (the real value is indeed between these two bounds and we have a good approximation here). Moreover as expected (see simulations Figure 2.5 page 15) the results concerning both robustness and robust performance are really satisfying. The value of the structured singular value is indeed less that 1 (involving $k_{max} > 1$) implying stability for all $\Delta$ in the unit ball and also good performance. One should not forget that it was only for LTI uncertainties without saturations, so an ideal unrealistic case but still interesting since it shows better results compared with a simple PID controller ($\gamma$ bounds values for robust performance were almost three times the values of Table 4.1). An interesting future step would be to use other methods based on IQC framework or parameter-dependent Lyapunov functions in order to include LTV uncertainties and nonlinearities, which will be partially exhibited in the next part.

<table>
<thead>
<tr>
<th></th>
<th>Upper bound</th>
<th>Lower bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robustness Analysis</td>
<td>$\mu = 0.79$</td>
<td>$\mu = 0.42$</td>
</tr>
<tr>
<td>Robust Performance</td>
<td>$\gamma = 0.37$</td>
<td>$\gamma = 0.19$</td>
</tr>
</tbody>
</table>

Table 4.1: Results of $\mu$-analysis for robustness and performance.
4.5 On IQC-based analysis: a brief overview

Before beginning this section the reader must be aware that it has for only objective to briefly expose the IQC-based analysis method since some work has been done during this Master Thesis concerning robustness analysis with LTV parameters. It is however hard if not impossible to detail this complex notion with a few words, but rather possible to give a small overview of it and its results. Some further details can be found in [22] for a complete theory description, [9] and [8] for the methods.

In the previous sections µ-analysis has exhibited interesting results and was shown as a powerful and mature method to analyze robustness in presence of LTI perturbations. Since it can only deal with LTI parameters other methods exist in order to be more precise by taking into account LTV perturbations and nonlinearities. One of them uses the so-called IQCs.

The principle is to replace a system time-varying coefficients, parametric uncertainties and nonlinearities by Integral Quadratic Constraints characterizations. A relaxed representation containing all possible solutions of the real system is then analyzed. All is based on the definition of multipliers Π which characterize the considered nonlinearities or perturbations, in a more or less conservative way. Then considering the general interconnection of Figure 4.2 the bounded operator ∆(.) satisfies the IQC σΠ defined by Π if for all ω in ℝ the following condition holds:

\[ \sigma_\Pi = \int_{-\infty}^{\infty} \left[ \begin{array}{c} \hat{z}(j\omega) \\ \hat{w}(j\omega) \end{array} \right]^* \Pi(j\omega) \left[ \begin{array}{c} \hat{z}(j\omega) \\ \hat{w}(j\omega) \end{array} \right] d\omega \geq 0 \] (4.7)

where the symbol \( \star \) denotes the conjugate transpose, \( \hat{z} \) and \( \hat{w} \) are the Fourier transforms of the signals \( z \) and \( w \). Then if the previous condition as well as a well-posedness condition is assumed and if, \( \forall \omega \in \mathbb{R} \):

\[ \left[ \begin{array}{cc} M(j\omega) & \Pi(j\omega) \\ I & I \end{array} \right] < 0 \] (4.8)

then the interconnection depicted in Figure 4.2 is stable.

Some IQC characterizations for a wide variety of operators contained in \( \Delta(.) \) can be found in [9], as well as much more details. The focus here will mainly concern the application with the same context as in section 4.4 but with slow-varying LTV perturbations. To do so some computing tools have been used and one of them is the MATLAB\textcopyright IQCβ Toolbox [23].

The \( \Delta(.) \) block in our analysis problem will then be characterized by the routines \texttt{iqc.slowtv} for slow time-varying parameters and \texttt{iqc.ltitgain} for the parametric uncertainties. Thus the bounded parameters \( \theta_{\alpha}, \theta_{\alpha'}, \theta_{m_\alpha}, \theta_{m_\delta} \) and \( \theta_v \) isolated in the \( \Delta(.) \) block will be considered as slowly varying with time (bounded rate of variation defined as \( |\dot{\delta}| < 5 \)) while the parametric
uncertainties $\delta_{m_\alpha}$ and $\delta_{m_\beta}$ are LTI. Using the IQC/β Toolbox has led to the computation of an upper-bound of the $L_2$-gain (solved via convex optimization). The value was equal to 0.8055. The most important thing is the fact that we have found a solution, which shows the problem is feasible and thus that the stability of the system can be proved (which was not surprising considering the simulations in Figure 2.5 page 15).

Other refined characterizations of some operators (i.e. the search for new multipliers) are currently a subject of study. This is the case for representing nonlinearities such as saturations in a less conservative way. This will require a much more complete attention in the months to come during the thesis that will follow this work since IQC-based analysis method is a complex, still in research phase but promising method. At that moment a test was done considering the controllers $K(s)$ and $J$ for the anti-windup strategy (see Chapter 3) introducing another IQC characterizing the sector nonlinearity to describe the saturation (see the routines iqc_sector and iqc_popov in [23]). Using the latter together with an adapted scheme with the previous LTI and LTV with slow rate of variations, IQCs has allowed to prove stability for all $\Delta$ such that $\|\Delta\|_\infty \leq 0.9$ which is not the unit ball. This does not mean that it is unstable for higher values, but it was the best achievable value with this method. The selected IQCs cannot prove stability but the closed-loop system could be stable anyway. The finite $H_2$-value is only a sufficient condition for stability. Moreover one should be aware that selecting poles for the multipliers allows to improve the results but this is still a very empiric step. Here 3 poles were selected for the IQC describing the slow-rate varying parameters which have made the computational times become very long (about 15 minutes are needed each time). Furthermore even if they have allowed to get good results they are certainly not the best choices but there are currently no ways to know (it would need weeks of trial-and-errors phases to test many possibilities...).

As a conclusion the IQCs reveal themselves quite efficient in considering many kinds of uncertainties, nonlinearities and parametric variations for robustness and performance analysis. The few tests on our model show interesting but perfectible results, especially concerning the nonlinearities. The main limitations are the computation loads which can become large (depending on the order of the system + the number of poles selected for the multipliers), the conservatism of the technique sometimes (the sector nonlinearity for the description of the saturation is really conservative since it considers much more values than necessary) and the selection of the poles of some multipliers which is still based on experimentation with no clear foundations (from which a result can be significantly different depending on the multiplier and its chosen poles).
Conclusion

As a conclusion this report has shown a brief framework on how to design a robust controller for a nonlinear system with nonlinearities like actuators rate saturations. First nonlinear dynamic inversion technique was applied in order to linearize and decouple the system nominally composed of nonlinear varying parameters. Based on a simple realistic aircraft’s longitudinal axis model the method has shown good results as long as there are no uncertainties on some parameters and no actuator’s dynamics. This has proved that simple NDI lacks of robustness properties.

Using a generalized version of NDI it was then possible to design a controller using $\mathcal{H}_\infty$ robustifying methods thanks to new powerful MATLAB® routines using nonsmooth optimization techniques for structured controllers. After some tuning of weighting functions depending on the transfers we wanted to minimize this method has brought a new controller which has shown very nice step responses in presence of parameters variations and uncertainties in our model, exhibiting very good robustness properties.

The last step was to include rate saturations on the actuator which, if not considered in a nominal design, can bring significantly poorer responses or instability. This has led to the development of an anti-windup strategy. Multi-channel $\mathcal{H}_\infty$ design was used in order to compute both the anti-windup controller and the nominal controller at the same time with the same method as before. This strategy was successfully tested on our longitudinal axis model thanks to powerful MATLAB® tools. The simulations have exhibited good performance results in presence of rate saturations, which has shown the efficiency of anti-windup control.

At the end it was also interesting to apply a very mature technique to analyze robustness in presence of LTI uncertainties. Considering the case with no saturations but with only LTI parameters representing the uncertainties and parameters used to cover the whole flight domain, the robust and performance analysis has unsurprisingly shown very good results with the $\mathcal{H}_\infty$ designed controller on our model.
Since most of the perturbations are clearly slowly time-varying in our model (those representing the flight domain variations) it could be interesting to use other robustness analysis methods to deal with them, such as parameter-dependent Lyapunov functions or IQCs. The latter has shown good results with the $\mathcal{H}_\infty$ designed controller with the non-saturated model illustrating what was exhibited with the simulations. Furthermore analyzing robustness and robust performance of the anti-windup based controllers requires these kinds of methods since they can deal with nonlinearities such as saturations (at the price of introducing some conservatism). This was briefly tested, leading to some interesting good results but still perfectible. Some improvements concerning selections of the poles for the multipliers and computational times are needed. This will require much more attention during the thesis that will follow this work.

As a general conclusion of all this, with the progress made during the past few years concerning $\mathcal{H}_\infty$ control design one can hope that these tools will one day be used in the industry because of the very interesting properties and the flexibility the technique shows. However some progress still needs to be done concerning the whole process and especially the robustness analysis. Indeed the currently developed methods dealing with nonlinearities and LTV perturbations have large difficulties to handle models with a high number of states leading to extremely long computing time. This issue will be a part of my future thesis research topic applied on a much larger and complete model (longitudinal and lateral axis, flexible modes), hoping that it will make a contribution to the huge, complex and interesting world of modern automatic control.
Appendix A

$\mathcal{H}_\infty$ Control Design: Matlab Code

```matlab
% Synthese Hinf

%% Init and OL model
clear all
clc

Deltaw = 1;  % Amplitude of variations on ma, za, mq for simulation
thetaV=1;  % Nominal thetaV
wlon=3*thetaV+7;  % Omega for Reference Model

% Nominal Open Loop (no weighting functions)
Wa=1;
Wq=1;
gainI=1;
tau1=0;
tau2=0;
k=1;
tauwa=0;
tauwq=0;
Wrob=1;
taurob1=0;
taurob2=0;

% State-space Open Loop for plotting
[a0,b0,c0,d0]=linmod('syndiag_3');
OL=ss(a0,b0,c0,d0);
OL1=OL([1 2],[1 2 3]);
bode(OL1)
```
grid on;

%% Weighting Functions Tuning
% Robust functions tuning
Wa=45;
Wq=20;
tauwa=2;
tauwq=1;
Wafcn=tf([ Wa ],[ tauwa 1 ]); % Transfer function Wa
Wqfcn=tf([ Wq ],[ tauwq 1 ]); % Transfer function Wq

% Perfo Tuning
tau1=1;
tau2=0.5;
gainI=20
k=0; % for having a 1st order low pass filter
Wefcn=tf([gainI*tau1 gainI*k],[tau2 1 0]); % Transfer function We

% Controller limitation
taurob1=50;
taurob2=1;
Wrob=0.3*(sqrt(2)/sqrt(1+(taurob1/taurob2)^2));
Wufcn=tf([Wrob*taurob1 Wrob],[taurob2 1 ]); % Transfer function Wu

%% Hinf synthesis
% Linear model of the whole system
[a,b,c,d]=linmod('syndiag_3');
ny=7; nu=1;
Ps=ss(a,b,c,d);

% Options for the hinfstruct routine
opt=hinfstructOptions('StableOffset',0.5,'RandomStart',4);

% 1st order controller
Kr=ltiblock.ss('Kstruct',1,nu,ny);
[K1,gaml]=hinfstruct(Ps,Kr,opt);
gaml

% Closed-loop eigenvalues for stability checking
CL1=feedback(OL,K1,4,[3 4 5 6 7 8 9],+1);
CL1=CL1([1 2],[1 2 3]);
damp(CL1)

% Plots of all bode graphs (perf, rob, control)
figure(1); clf ;hold on;
subplot(2,3,1)
Appendix A. $\mathcal{H}_\infty$ Control Design: Matlab Code

```matlab
bodemag(CL1(1,1), 'r', Wefcn^(-1), 'b--')
subplot(2,3,2)
bodemag(CL1(1,2), 'r', Wafcn^(-1)*Wefcn^(-1), 'b--')
subplot(2,3,3)
bodemag(CL1(1,3), 'r', Wqfcn^(-1)*Wefcn^(-1), 'b--')
subplot(2,3,4)
bodemag(CL1(2,1), 'r', Wufcn^(-1), 'b--')
subplot(2,3,5)
bodemag(CL1(2,2), 'r', Wafcn^(-1)*Wufcn^(-1), 'b--')
subplot(2,3,6)
bodemag(CL1(2,3), 'r', Wqfcn^(-1)*Wufcn^(-1), 'b--')

%% Hinf Norm for perf and robustess

normhinf(CL1(1,1))
normhinf(CL1(2,1))
normhinf(CL1(1,2))
normhinf(CL1(1,3))

%% Singular values for performance analysis

om=logspace(-2,3,100);
sig1=sigma(CL1(1,1),om);
sig2=sigma(CL1(2,1),om);
figure(2); clf;
semilogx(om,sig1);
hold on;
semilogx(om,sig2,'r');
xlabel('pulsation (rad/s)');
ylabel('sigma');

%% Singular values for robustess analysis

om=logspace(-2,3,100);
sig3=sigma(CL1(1,2),om);
sig4=sigma(CL1(1,3),om);
figure(3); clf;
semilogx(om,sig3);
hold on;
semilogx(om,sig4,'r');
xlabel('pulsation (rad/s)');
ylabel('sigma');

%% Simulation
```
set(0,'defaultAxesFontName', 'Courier 10 Pitch')
set(0,'defaultTextFontName', 'Courier 10 Pitch')
K=K1;
simdiag_2;
i=0;
clf(figure(4))
for thetaV=-1:0.5:1;
  i=i+1;
  colorplot=['m' 'b' 'c' 'g' 'y'];
sim('simdiag_2')
figure(4); hold on;
plot(simout(:,1),simout(:,2),colorplot(i),'LineWidth',1.7)
%time(:,i)=simout(:,1);
%values(:,i)=simout(:,2);
end
plot(simout(:,1),simout(:,3),'-r','LineWidth',1.7)
legend('theta_v=-1','theta_v=-0.5','theta_v=0','theta_v=0.5','theta_v=1','ref');
axis([0 20 -10 34]);
xlabel('\fontsize{15} Time [s]')
ylabel('Angle of Attack [deg]','FontSize',15)
set(gca,'FontSize',14);
grid on;
hold off;

%% Simulation with PID controller for comparison

set(0,'defaultAxesFontName', 'Courier 10 Pitch')
set(0,'defaultTextFontName', 'Courier 10 Pitch')
man=1;
simdiag_2PID
i=0;
clf(figure(5))
for thetaV=-1:0.5:1;
  i=i+1;
  colorplot=['m' 'b' 'c' 'g' 'y'];
sim('simdiag_2PID')
figure(5); hold on;
plot(simout(:,1),simout(:,2),colorplot(i),'LineWidth',1.7)
%time(:,i)=simout(:,1);
%values(:,i)=simout(:,2);
end
plot(simout(:,1),simout(:,3),'-r','LineWidth',1.7)
legend('theta_v=-1','theta_v=-0.5','theta_v=0','theta_v=0.5','theta_v=1','ref');
axis([0 20 -10 34]);
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>161</td>
<td>xlabel('Time [s]', 'FontSize', 15);</td>
</tr>
<tr>
<td>162</td>
<td>ylabel('Angle of Attack [deg]', 'FontSize', 15);</td>
</tr>
<tr>
<td>163</td>
<td>set(gca, 'FontSize', 14);</td>
</tr>
<tr>
<td>164</td>
<td>grid on;</td>
</tr>
<tr>
<td>165</td>
<td>hold off;</td>
</tr>
</tbody>
</table>
Appendix B

Multi-objective $\mathcal{H}_\infty$ Control Design for Anti-windup control: Matlab Code

```matlab
1
2       %-------- Synthese Hinf -------------%
3       %--------Multi-objective Anti-windup-----%  
4
5       %%% Init and OL Model
6       clear all
7       clc
8
9       Deltaw = 0;       % Amplitude of variations on ma, za, mq for simulation
10      thetaV=1;         % Nominal thetaV
11      wlon=3*thetaV+7;  % Omega for Reference Model
12
13       % Nominal Open Loop (no weighting functions)
14      Wa=1;
15      Wq=1;
16      gainI=1;
17      tau1=0;
18      tau2=0;
19      k=1;
20      tauwa=0;
21      tauwq=0;
22      WJ=1;
23      WJ=ss(WJ);
```
Appendix B. Multi-objective H\text{\textsubscript{\infty}} Control Design for Anti-windup control: Matlab Code

```matlab
Wrob=1;
taurob1=0;
taurob2=0;

% State-space Open Loop for plotting
[a0,b0,c0,d0]=linmod('syndiag_aw');
OL=ss(a0,b0,c0,d0);
OL1=OL([1 2],[1 2 3 6]);
bode(OL1)
grid on;

%% Weighting Functions Tuning
% Robust functions tuning (same as Hinf with no sat)
Wa=45;
Wq=20;
tauwa=2;
tauwq=1;
Wafcn=tf([Wa],[tauwa 1]);
Wqfcn=tf([Wq],[tauwq 1]);

% Perfo Tuning (slight modifications compared to Hinf with no sat)
tau1=1;
tau2=0.5;
gainI=70;
k=0; % 0 for having a 1st order low pass filter
Wefcn=tf([gainI*tau1 gainI*k],[tau2 1 0]);

% WJ Tuning (Simple gain since J(s) = J)
s=tf('s');
WJ0=55;
WJ=WJ0;
WJfcn=tf(WJ);
WJ=ss(WJ);

% Controller Limitation (does not matter that much here due to RL)
taurob1=50;
taurob2=1;
Wrob=0.1*(sqrt(2)/sqrt(1+(taurob1/taurob2)^2));

%% Multi-objective H\text{\textsubscript{\infty}} synthesis
[a,b,c,d]=linmod('syndiag_aw');
ny=7; nu=1; % Number of inputs/outputs for the nominal controller
Ps=ss(a,b,c,d);
Plen=Ps([1 2 6 7 8 9 10 11 12],[1 2 3 6]); % Nominal plant (no AW)```
Prob=Ps([3 4 5 6 7 8 9 10 11 12],[4 5 6]); % Robust plant (AW and no perf)

opt=hinfstructOptions('StableOffset',0.5,'RandomStart',3);

% First-order nominal controller C(s) + gain anti-windup J
Kr=ltiblock.ss('Kstruct',1,nu,ny);
J0=ltiblock.gain('J',1,1);

% Definition of the closed loop G_{ec,nom} and G_{ec,rob}
CLnom=lft(Pnom,Kr,nu,ny);
CLrob=lft(Prob,blkdiag(J0,Kr));

% Multi objective hinfstruct
[CL,gam1]=hinfstruct(blkdiag(CLnom,CLrob),opt);

K1=ss(CL.Block.Kstruct);
J=ss(CL.Block.J);

CL1=feedback(OL,blkdiag(J,K1),[5 6],[5 6 7 8 9 10 11 12],+1);
damp(CL1)

%% Simulation and Plotting
set(0,'defaultAxesFontName', 'Courier 10 Pitch')
set(0,'defaultTextFontName', 'Courier 10 Pitch')
K=K1;
simdiag_aw;
i=0;
cl(figure(4))
for thetaV=-1:0.5:1;
    i=i+1;
    colorplot=['m' 'b' 'c' 'g' 'y'];
    sim('simdiag_aw')
```matlab
figure(4); hold on;
plot(simout(:,1),simout(:,2),colorplot(i),'LineWidth',1.7)
end
plot(simout(:,1),simout(:,3),'-r','LineWidth',1.7)
legend('$\theta V=-1$', '$\theta V=-0.5$', '$\theta V=0$', '$\theta V=0.5$', '$\theta V=1$', 'ref');
axis([0 20 -10 35]);
xlabel('Time [s]', 'FontSize', 15);
ylabel('Angle of Attack [deg]', 'FontSize', 15);
set(gca,'FontSize',14);
grid on;
hold off;
```
Appendix C

μ-analysis and performance:
Matlab Code

1 % LONGITUDINAL AXIS MODEL %
2 % LFR and μ-analysis %
3 clear all
4 clc
5 close all
6
7 % LFR Toolbox Path Definition
8 path_root = '/home/jlesprie/Documents/LFRTandSLK/';
9 path([path_root,'lfr'],path);
10 path([path_root,'lfr/demo'],path);
11 path([path_root,'lfr/simulink'],path);
12 path([path_root,'lfr/simulink/demo'],path);
13 path([path_root,'lfr/simulink/sub'],path);
14
15 % Skew-μ Toolbox Path Definition
16 my_root='/home/jlesprie/Documents/Stage ONERA 2012/Skew Mu Toolbox/';
17 if strcmp(my_root,'...')
18     error('The first line of path2SMT.m should be edited first ! ');
19 end
20 path(path,[my_root 'SMTv4']);
21
22
23 % Nominal Model with parameters and uncertainties
24
25 %Parameters Definition
Appendix C. \( \mu \)-analysis and performance: Matlab Code

\[ lfrs \ tv \ tza \ tma \ tmq \ dma \ dmd \ 'real' \ [-1 \ -1 \ -1 \ -1 \ -1] \ [1 \ 1 \ 1 \ 1 \ 1] \ ... \]

\[ [0 \ 0 \ 0 \ 0 \ 0]; \]

%Parameters Equations
za = -(1+0.6*tza)*(1.3+0.7*tv);
ma = 0.8 + (tma+1.5*dma)*(5+4*tv);
mq = -(1+0.35*tmq)*(0.9+0.45*tv);
md = -40*(1+0.25*dmd)*(1+1.15*tv+0.35*tv^2);
wlon = 3*tv+7;

%Definition of the SS-representation of the model
A = [za 1;ma mq];
B = [0;md];
C = [1 0;0 1];
D = zeros(2,1);

%Definition of the LFR object
sysnom = abcd2lfr([A B;C D],2,1);

%Reference Block
xi=0.7;
kp=2*xi*wlon;
ki=wlon^2;
Aref = [0 1;-ki -kp];
Bref = [0;ki];
Cref = [1 0;0 1];
Dref = zeros(2,1);
sysref = abcd2lfr([Aref Bref;Cref Dref],2,1);

%% wat & wqt for controller feedback
thetaV_init = tv;
zao = -(1.3+0.7*thetaV_init);
mao = 0.8;
mqo = -(0.9+0.45*thetaV_init);
mdo = -40*(1+1.15*thetaV_init+0.35*thetaV_init^2);
Aw = zeros(2,2);
Bw = [za-zao 0;ma-mao mq-mqo];
Cw = [1 0;0 1];
Dw = zeros(2,2);
sysw = abcd2lfr([Aw Bw;Cw Dw],2,1);

%% Controller loading (from Hinf computation)
Appendix C. \( \mu \)-analysis and performance: Matlab Code

```matlab
load('good_controller_nosat');
K=K1;

%% Global LFR computation using SLK2LFR (LFR Toolbox)
globalLFR=slk2lfr('LFR_build_1',1);

%% \( \mu \)-analysis, using new tools from ONERA (Skew-\( \mu \) Toolbox)
[M,blk] = convert_data(globalLFR);
[M,blk] = make_square(M,blk);

%Robust Analysis
A_Mrob = M.a;
B_Mrob = M.b(:,1:size(M.b,2)-1);
C_Mrob = M.c(1:size(M.c,1)-1,:);
D_Mrob = M.d(1:size(M.d,1)-1,1:size(M.d,2)-1);
Mrob = ss(A_Mrob,B_Mrob,C_Mrob,D_Mrob);
blkrob = blk(1:size(blk,1)-1,1:size(blk,2)-1);
clear opt;
opt.lmi=1;
[ubrob,wcubrob,tabubrob,pbrob] = muub_mixed(Mrob,blkrob,opt);
[lbrob,wclbrob,pertlbrob,tablbrob] = mulb_mixed(Mrob,blkrob);
fprintf('The Robust Analysis gives the following results: 
Upper bound: %6.4g
Lower bound: %6.4g
', ubrob, lbrob)
pause

%Performance Analysis
blkperf = blk(1:length(blk)-1,1:size(blk,2)-1);
[ubperf,wcubperf,tabupbperf,pbupf] = muub_mixed(M,blk,opt);
[lbperf,wclbperf,pertlbperf,tablperf] = hinflb_real(M,blkperf);
fprintf('The Robust Performance Analysis gives the following results: 
Upper bound: %6.4g
Lower bound: %6.4g
', ubperf, lbperf)
```
Bibliography


