Performance Enhancement of a Self-Scheduled Longitudinal Flight Control System via Multi-Objective Optimization

H. Lhachemi\(^1\), D. Saussié\(^1\) and G. Zhu\(^1\)

Abstract—The present work aims at improving the performance of robust and smooth self-scheduled controllers in the framework of structured \(H_\infty\) design. The developed procedure exploits multi-model and multi-channel capabilities of a MATLAB-based tool hinfsctruct in the design of robust and self-scheduled flight control systems. In this approach, both controller and gain-scheduling architectures are defined \textit{a priori} and are cast into the structured \(H_\infty\) synthesis framework. By formulating the considered problem in the multi-objective optimization framework, the control design amounts then to computing weakly Pareto optimal solutions in which weighting coefficients can be determined based on physical considerations or preliminary designs. The tuning of weighting coefficients can be performed by normalizing the minimization effort over the operating domain or assigning more weight to a specific sub-domain. The proposed procedure is applied to the design of a robust and self-scheduled longitudinal flight control system. Numerical analysis and simulation show an important performance improvement.

I. INTRODUCTION

As the dynamic behavior of an aircraft varies significantly with such parameters as altitude and Mach number, a single linear time-invariant (LTI) controller cannot assure the required performance for flight control within the whole flight envelope. To overcome this problem, gain-scheduling techniques \([1], [2]\) have been resorted to in the design of flight control systems. In this framework, the gains of the controller are updated in function of scheduled variables regarding the operating point. Essentially, gain-scheduling consists in designing a set of LTI controllers for a finite set of linearized models representing the dynamics of the aircraft over the flight envelope. During the flight, controller gains are then computed using interpolation algorithms \([3], [4]\). This \textit{a posteriori} procedure is time consuming and raises serious challenges in terms of performance guarantee \([5]\). Moreover, interpolation may result in inadequate values outside of the points considered in the controller design. To overcome those limitations, some techniques were developed which allow imposing \textit{a priori} the structure of the scheduled gains \([6], [7], [8]\).

Based on the framework of structured \(H_\infty\) control \([9]\), a new technique has been introduced in \([10]\) to design robust and smooth self-scheduled controllers over an operating domain with a predefined control architecture. This approach takes advantage of the MATLAB function hinfsctruct which enables to tune the gains of a controller with fixed architecture \([11]\). An important feature offered by this tool is the multi-model synthesis, which allows designing a single controller for several plants at the same time \([12]\). This technique offers also a possibility to design robust and self-scheduled controllers. One can define \textit{a priori} the structure of the scheduled gains w.r.t. scheduling variables and apply the multi-model synthesis to a set of plants chosen in the operating domain and over the uncertain domain. In this procedure, gain-scheduling can be considered as a design requirement through \(H_\infty\) constraints. This paper extends the work reported in \([10]\) by casting the multi-model design problem into the multi-objective optimization framework. In the preliminary design, all the operating points have the same weighting, which leads to the computation of a weakly Pareto optimal solution through a non-uniform optimization effort over the flight envelope. Based on physical considerations or specific requirements, it may be of interest to assign more weight to some operating points than others in the design. This leads to the computation of different weakly Pareto optimal solutions. The main benefit of the proposed approach is that it allows achieving an enhanced overall system performance through the tuning of weighting coefficients, which can be handled by the aforementioned structured \(H_\infty\) control design tool.

The rest of the paper is organized as follows. Section II describes the design procedure of a robust self-scheduled controller in the \(H_\infty\) framework. Section III depicts the aircraft longitudinal dynamics problem under mass and CG (Center of Gravity) uncertainties (model and requirements). Section IV presents the application of the design procedure to synthesize robust and self-scheduled controllers. Finally, performance evaluation and numerical simulations of the system in closed-loop are presented in Section V.

II. ROBUST AND SMOOTH SELF-SCHEDULING IN THE STRUCTURED \(H_\infty\) FRAMEWORK

We first introduce the procedure used in the design of a robust and smooth self-scheduled controller in the \(H_\infty\) framework. This problem is then reformulated in the multi-objective optimization framework.

A. Standard Form

The LTI system is represented under the formalism of the standard form (Fig. 1) \([13]\). We consider the configuration where the plant \(P(s, \theta, \delta)\) depends on the operating point \(\theta\) and an uncertain parameter vector \(\delta\). In this context, \(K(s, \theta)\) is supposed to be a structured controller parametrized by \(\theta\). Scheduling variables \(\theta\) lie in the operating domain \(\Theta\) and
are assumed to be known. Uncertain parameters \( \delta \) lie in the uncertain domain \( \Delta \) with known upper and lower bounds.

In this formalism, exogenous inputs are gathered in \( w \), regulated outputs in \( z \), control signals in \( u \) and measurements in \( y \). The closed-loop transfer function between the input \( w \) and the output \( z \) is given via the following LFT representation:

\[
T_{zw} = F_j(P,K).
\]  

(1)

B. Structured \( H_\infty \) Synthesis

In the classical \( H_\infty \) framework, the objective is, given an augmented plant \( P \), to find an internal stabilizing controller \( K^* \) that minimizes the \( H_\infty \) norm of the transfer function \( T_{zw} \):

\[
||T_{zw}(P,K)||_{\infty} := \max_{w \neq 0} \frac{||z||_2}{||w||_2} = \max_{\omega \in \mathbb{R}} (T_{zw}(j\omega)).
\]  

(2)

Unlike classic \( H_\infty \) solutions, structured \( H_\infty \) synthesis allows fixing \textit{a priori} the controller architecture to meet \( H_\infty \) constraints. In this framework, common requirements (tracking, noise rejection, stability margins) have to be expressed in terms of \( H_\infty \) norms. For multiple requirements, one can stack constraints of the type:

\[
||W_i T_i||_{\infty} < 1, \quad i = 1, \ldots, m,
\]  

(3)

where \( T_i \) denotes the transfer function of interest and \( W_i \) is a weighting transfer function capturing the requirements. One can build the following generalized constraint:

\[
||H||_{\infty} < 1,
\]  

(4)

where

\[
H = \text{diag} [W_1 T_1, \ldots, W_m T_m].
\]  

(5)

Structured \( H_\infty \) method amounts then to optimizing the generalized constraint \( H \) rather than the global transfer function \( T_{zw}(P,K) \) and hence, it reduces the conservatism of classic \( H_\infty \) approaches.

Another distinguished feature offered by structured \( H_\infty \) method is the multi-model synthesis capability. It enables the design of a single controller based on requirements related to multiple plants \( P \) [11], [13].

C. Design Strategies

Problems with different natures can take advantage of the multi-model capability of the \texttt{hinfsyn} function, as detailed below. Note that those strategies still require an \textit{a posteriori} validation over the uncertain or/and operating domain, with possible time-varying dependencies.

1) \textit{Robust Synthesis:} Let the uncertain plant \( G(s, \delta, \theta_0) \) with \( \delta \in \Delta \) for a fixed operating point \( \theta_0 \). The required constraints are:

\[
||W_i T_i(s, \theta_0, \delta)||_{\infty} < 1, \quad i = 1, \ldots, m, \quad \delta \in \Delta.
\]  

(6)

As the continuous problem is not tractable, we consider the approximation on a finite set of values \( (\delta_j)_{1 \leq j \leq n} \in \Delta \):

\[
||W_i T_i(s, \theta_0, \delta_j)||_{\infty} < 1, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.
\]  

(7)

One can build the generalized constraint \( H(s) \):

\[
H(s) = \text{diag} [(W_i T_i(s, \theta_0, \delta_j))_{i=1,\ldots,m; j=1,\ldots,n}].
\]  

(8)

Using \texttt{hinfsyn}, we get one controller that minimizes \( ||H(s)||_{\infty} \).

2) \textit{Smooth Self-Scheduled Synthesis:} Let \( G(s, \theta) \) be a parameter dependant plant and \( K(s, \theta) \) a scheduled controller, with \( \theta \in \Theta \). In our setup, each scalar gain \( k \) is expressed as a polynomial function of the scheduling variables:

\[
k(\theta) = \sum_{d_1+\ldots+d_p \leq d} k_{d_1,\ldots,d_p} \prod_{i=1}^{p} \theta_i^{d_i}.
\]  

(9)

Considering a finite set of operating points \( (\theta_l)_{1 \leq l \leq p} \in \Theta \), the scheduled controller should satisfy:

\[
||W_i T_i(s, \theta_l)||_{\infty} < 1, \quad i = 1, \ldots, m, \quad l = 1, \ldots, p.
\]  

(10)

Note that in general the weighting functions depend on the operating point. Embedding the gain-scheduling structure into the controller architecture, one can build the generalized constraint \( H(s) \):

\[
H(s) = \text{diag} [(W_i T_i(s, \theta_l))_{i=1,\ldots,m; l=1,\ldots,p}].
\]  

(11)

In this setup, the function \texttt{hinfsyn} can be used to tune \( k_{d_1,\ldots,d_p} \) to synthesize one controller that minimizes \( ||H(s)||_{\infty} \).

3) \textit{Robust and Smooth Self-Scheduled Synthesis:} Finally, it is possible to combine the above two strategies to synthesize in a single procedure a robust and self-scheduled controller. One can define \textit{a priori} the structure of the controller and the dependency of each scalar gain \( k \) versus scheduled variables (9). Based on a gridding of both uncertain domain \( (\delta_j)_{1 \leq j \leq n} \in \Delta \) and operating domain \( (\theta_l)_{1 \leq l \leq p} \in \Theta \), the robust and self-scheduled controller should satisfy for all \( i = 1, \ldots, m \):

\[
||W_i T_i(s, \theta_l, \delta_j)||_{\infty} < 1, \quad j = 1, \ldots, n, \quad l = 1, \ldots, p.
\]  

(12)

One can build the generalized constraint \( H(s) \) and minimize it with \texttt{hinfsyn}:

\[
H(s) = \text{diag} [(W_i T_i(s, \theta_l, \delta_j))_{i=1,\ldots,m; j=1,\ldots,n; l=1,\ldots,p}].
\]  

(13)
D. Reformulation in the Multi-Objective Optimization Framework

All the strategies described previously can be reformulated in the multi-objective optimization framework [14]. Indeed, let us consider the design of a self-scheduled controller over an operating domain. The operating domain is first discretized into a finite set \( \{ \theta_i \}_{1 \leq i \leq p} \subset \Theta \) and controller architecture is defined \textit{a priori}. For each operating point \( \theta_i \), all the \( H_m \) constraints (possibly including uncertain requirements) are concatenated in the generalized constraint \( H_l(s) \). The minimization of each individual constraint \( H_l(s) \) via \texttt{hinfsstruct} leads to the computation of the optimal gain \( \gamma \) such that:

\[
\gamma = \min_{K \in \mathbb{K}} \| H_l(s) \|_\infty, \tag{14}
\]

where \( \mathbb{K} \) is the set of controllers \( K \) with respect to the predefined architecture.

For the self-scheduled controller with a predefined gain-scheduling structure, we define the multi-objective function \( F = [F_1, \ldots, F_p] \) as:

\[
F : \mathbb{K}_{GS} \rightarrow \mathbb{R}^p, \quad K \rightarrow \left[ \| H_1(s) \|_\infty, \ldots, \| H_p(s) \|_\infty \right]^T, \tag{15}
\]

where \( \mathbb{K}_{GS} \) denotes the set of admissible self-scheduled controllers with respect to both predefined controller and gain-scheduling structures. Based on (14), one can then define the \textit{utopia point} \( \Gamma^0 = [\gamma_1, \ldots, \gamma_p]^T \) that represents the minimum reachable gain of each individual operating point:\n\[
\forall \theta \in \Theta, \Gamma^0 \leq F(\theta). \tag{17}
\]

For practical applications, \( \Gamma^0 \) is generally unattainable. The classic approach consists in finding a solution that is as close\(^2\) as possible to \( \Gamma^0 \). Such a solution, denoted by \( K^* \), is a \textit{Pareto optimal} point, i.e. there does not exist another controller \( K \in \mathbb{K}_{GS} \) such that \( F(K) \leq F(K^*) \), and \( F_l(K) < F_l(K^*) \) for at least one function. Considering the computational issue, numerical solutions provided by optimization algorithms may not be Pareto optimal but \textit{weakly Pareto optimal}, i.e. there does not exist a controller \( K \in \mathbb{K}_{GS} \) such that \( F(K) < F(K^*) \).

The approach presented in Subsection II-C.3 consists in finding a weakly Pareto optimal solution for \( F \) through the minimization of the generalized constraint \( H(s) \):

\[
\min_{K \in \mathbb{K}_{GS}} \| H(s) \|_\infty = \min_{K \in \mathbb{K}_{GS}} \max_{1 \leq l \leq p} F_l(K). \tag{16}
\]

Thus we have:

\[
\max_{1 \leq l \leq p} \gamma_l \leq \min_{K \in \mathbb{K}_{GS}} \| H(s) \|_\infty. \tag{17}
\]

As a consequence, the effort on minimizing \( H(s) \) is limited by the worst performance of each operating point considered individually. Thus, the minimization effort is not uniform over the flight envelope. Consequently, certain operating points could be well optimized during the synthesis, whereas those for which \( \gamma_l \) are smaller may suffer from performance degradation.

It is then of interest to look for an other weakly Pareto optimal solution making the optimization effort better distributed over the whole flight envelope. Indeed, based on physical considerations or specific requirements, it is possible to assign more weight in the design to a specific subdomain. In this way, certain weighting scalar coefficients \( (\alpha_i)_{1 \leq i \leq p} \in [0, 1]^p \) can be injected in the original problem to compute:

\[
\min_{K \in \mathbb{K}_{GS}} \max_{1 \leq l \leq p} \alpha_l F_l(K). \tag{18}
\]

A controller minimizing (18) can be computed by \texttt{hinfsstruct} via the minimization of the following generalized constraint:

\[
H(s) = \text{diag} \left[ (\alpha_i H_l(s))_{l=1,\ldots,p} \right]. \tag{19}
\]

In this approach, weakly Pareto optimal solutions computed by \texttt{hinfsstruct} depend on the weighting coefficients used in the design. Performance improvement with the proposed approach will be illustrated through the design of a longitudinal flight control system.

III. AIRCRAFT MODEL AND REQUIREMENTS

In this section, the short-term longitudinal dynamics of an aircraft are introduced. Closed-loop system requirements are also presented.

A. Aircraft Dynamic Model

The control of the longitudinal dynamics of an aircraft is considered for short term movements. In this problem, both mass \( m \) and longitudinal CG position \( \Delta x \) are supposed to be uncertain parameters. For short term movements, the classical Short Period approximation leads to the following reduced model [10]:

\[
\begin{bmatrix}
\dot{\alpha} \\
\dot{q}
\end{bmatrix} = \begin{bmatrix}
A(m, \Delta x) & B(m, \Delta x) \\
0 & 1
\end{bmatrix} \begin{bmatrix}
\alpha \\
q
\end{bmatrix} + \begin{bmatrix}
0 \\
\Delta \delta_e
\end{bmatrix}, \tag{20}
\]

where \( \alpha \) is the angle of attack, \( q \) the pitch rate and \( \delta_e \) the elevator angle. The outputs of the system are the pitch rate and the load factor \( n_c \) at pilot’s location expressed in g-units:

\[
n_c \approx \left( U_e \dot{\alpha} - U_e q - \dot{\alpha} q \right) / g. \tag{21}
\]

Finally, the model is completed with the dynamics of elevator through a first order transfer function between the actual deflection \( \delta_e \) and the command \( \delta_{ec} \).

For numerical applications, we consider the F-16 Fighting Falcon whose physical characteristics are retrieved from a NASA technical report [15], [16].

B. Requirements

Load factor \( n_c \) control is considered over the flight envelope \( \Theta \) shown in Fig. 2 for the following mass and CG uncertainties: \( m \in [8300, 10300] \text{kg} \) and \( \Delta x \in [-4.35, 4.35] \text{m} \). The requirements over the flight envelope \( \Theta \) and for all mass and CG variations are:

- internal stability;
IV. APPLICATION TO LOAD FACTOR CONTROL

A. Controller Architecture and Augmented Plant

1) Controller Architecture: The controller structure [13] is depicted in Fig. 3 where \( \varepsilon = 0.001 \) and the block \( F_{roll} \) is chosen as a second order roll-off filter for noise rejection:

\[
F_{roll}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}.
\]  

(22)

2) Weighting functions: Performance requirements are cast into \( H_\infty \) constraints as follows.

a) Tracking: Time-response and modal performance requirements are converted in a model following objective. The reference model \( G_{ref} \) is chosen to be a second order system with a damping coefficient set to \( \zeta_{ref} = 0.7 \) and a natural frequency \( \omega_{n,ref} \) which is adjusted according to the operating point. In this setup, the objective is to minimize the transfer function \( T_1 \) between the load factor command \( n_{zc} \) and the error \( e \) for low and medium frequencies. This specification is captured through the requirement \(| |W_1 T_1||_\infty < 1\), where \( W_1 \) is a low-pass filter:

\[
W_1 = 15 \left( \frac{s}{s + 5} \right)^2.
\]

(23)

b) Noise Rejection: Noise rejection is required to prevent high frequency signals from entering in the actuator. This is captured by minimizing the transfer function \( T_2 \) between the noise inputs \( n_q \) and \( n_n \) and the control signal \( \delta_{ec} \) for high frequencies. It is expressed as \(| |W_2 T_2||_\infty < 1\), where \( W_2 \) is a high-pass filter:

\[
W_2 = \frac{s^2/80^2 + \sqrt{2}s/80 + 1}{20s^2 + 80^2/s^2 + \sqrt{2}s/8000 + 1}.
\]

(24)

c) Stability Margin: Phase and gain margins requirements are formulated in the \( H_\infty \) framework through the constraint \(| |W_3 T_3||_\infty < 1\). \( T_3 \) denotes the transfer function between \( w \) and \( \delta_{ec} \) and \( W_3 = 0.8 \). This specification guarantees a gain margin of at least 13.97dB or -5.1dB and a phase margin \(|PM| \geq 47.1^\circ \).

B. Preliminary Design and Baseline Self-Scheduled Controller Synthesis

The present work is based-on the design of a robust and self-scheduled controller reported in [10] described in this subsection.

1) Preliminary Design: As mentioned earlier, a preliminary design is necessary to:

- Adjust (and fix) the natural frequency of the reference model \( \omega_{n,ref} \) depending on the current operating point;
- Define an adequate shape of the gains versus scheduling variables;
- Perform an \textit{a posteriori} gain-scheduling that will be used as an initial condition for the design of the robust and self-scheduled controller.

Firstly, both flight envelope and uncertain domain are discretized. In our case, 12 operating points (Fig. 2) and 5 uncertain configurations, i.e. the nominal one and the four extremes, are considered. At each operating point, the strategy described in Subsection II-C.1 is applied to design 12 robust LTI controllers. The 12 final \( H_\infty \) norms of \( H_1(s) \) are summarized in Table I.

Secondly, based on the analysis of the gains, quadratic polynomial functions are used to schedule \textit{a posteriori} the 5 gains of the controller versus altitude and Mach number. This procedure results in a performance degradation of the controller to a global maximum \( H_\infty \) norm of 3.86 over the 12 operating points used in the design.

### Table I

<table>
<thead>
<tr>
<th>h (m)</th>
<th>M</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td></td>
<td>1.44</td>
<td>1.28</td>
<td>1.16</td>
<td>1.14</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5000</td>
<td></td>
<td>1.45</td>
<td>1.35</td>
<td>1.28</td>
<td>1.25</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10000</td>
<td></td>
<td>-</td>
<td>-</td>
<td>1.48</td>
<td>1.47</td>
<td>1.45</td>
<td>1.41</td>
</tr>
</tbody>
</table>

Fig. 2. Flight envelope Θ

Fig. 3. Controller architecture
2) A Priori Gain-Scheduling: Based on the result of the preliminary design, each tunable gain of the controller is expressed as a quadratic polynomial function of altitude and Mach number. The scheduling function is then directly embedded into the controller structure, leading to a total of 30 tunable gains. Finally, the procedure described in Subsection II-C.3 is used to tune the 30 scalar gains to meet the robustness and scheduling requirement in the $H_{\infty}$ framework, which are given by:

\[
K_p(h, M) = k_{00} + k_{10}h + k_{20}h^2 + k_{11}M + k_{21}hM + k_{22}h^2
\]

\[
K_p(h, M) = k_{00} + k_{10}h + k_{20}h^2 + k_{11}M + k_{21}hM + k_{22}h^2
\]

\[
K_p(h, M) = k_{00} + k_{10}h + k_{20}h^2 + k_{11}M + k_{21}hM + k_{22}h^2
\]

\[
\alpha^2(h, M) = \alpha_{00}^2 + \alpha_{10}^2h + \alpha_{20}^2h^2 + \alpha_{11}^2M + \alpha_{21}^2hM + \alpha_{22}^2h^2
\]

\[
\gamma(h, M) = \alpha_{00} + \alpha_{10}h + \alpha_{20}h^2 + \alpha_{11}M + \alpha_{21}hM + \alpha_{22}h^2
\]

where $\alpha = 2\xi / \omega_n$ and $\omega_n$ are considered rather than $\xi$ and $\omega_n$ for practical reasons.

It has been shown in [10] that the obtained robust and self-scheduled controller met the requirements over the flight envelope. The final $H_{\infty}$ norm of the generalized constraint $H(s)$ reaches 1.56, compared to the worst case of 1.48 in the preliminary design (Tab. I).

C. Normalized Synthesis

In spite of good results, the previous approach may present some limitations. Indeed, it can be seen from Table I that the optimization effort is not uniform over the flight envelope. As a consequence, the previous approach assures a good optimization for low speeds and high altitudes but it compromises the performance for flights at low altitude and high Mach number.

Weighting coefficients can be introduced to normalize the minimization effort over the flight envelope. A contrario, one can adjust the weighting coefficients to assign more weight to specific regions of the flight envelope.

1) First Approach: Let $N_i$ be the final $H_{\infty}$ norm obtained in the preliminary design for each operating point $\theta_i$ (see, e.g., Tab. I). A natural way to make the minimization effort uniform is to introduce the following weighting coefficients:

\[
P_i = 1/N_i, \quad l = 1, \ldots, 12,
\]

and to minimize the weighted generalized constraint (19). The execution has been stopped after 400 iterations with a final gain of 1.14. The final $H_{\infty}$ norm of $H_l(s)$ for each operating point is given in Tab. II. Compared to 1.56 obtained with the previous design, it shows a significant improvement at high speed and low altitude region. Whereas, the performance at low speed and high altitude is degraded as expected.

2) Second Approach: Through an ad hoc procedure, several weightings $(P_l)_{1 \leq l \leq 12}$ are tested. The objective is to improve the performance at low altitude and high Mach number while keeping good performance for low Mach numbers and high altitudes. According to our experimentation, the best result is obtained for:

\[
P_l = 2/(N_l + 1), \quad l = 1, \ldots, 12.
\]

Note that compared to the weight coefficients defined in (25), (26) tends to make the minimization effort to be more smoothly distributed over the whole flight envelope. The execution has been stopped after 400 iterations. Starting from an initial value of 3.61, the norm of the generalized constraint reaches 1.30 for a calculation time of 182 hours $^3$ to simultaneously minimize the $H_{\infty}$ norm of 240 scalar transfer functions (12 operating points $\times$ 5 mass and CG configurations $\times$ 4 scalar transfer function requirements). The final $H_{\infty}$ norm of $H_l(s)$ for each operating point is given in Tab. III.

It can be seen from Table III that this setup exhibits a good compromise between the original and the first normalized approaches (Tab. I and Tab. II). The performance is considerably improved for flights at low altitude and high mach number ($H_{\infty}$ norm of $H_l(s)$ significantly less than 1.56 of the original approach). Flights at low speeds and high altitudes are also improved compared to the first normalized approach, although the performance is slightly degraded from the original design.

To avoid the ad hoc procedure in weighting coefficients adjustment, one can define the weighting coefficients as a function of a tunable vector $p$. The weighting coefficients adjustment can be then directly cast into the synthesis process. Nevertheless, such an approach requires a careful choice of the structure of the weighting functions.

V. PERFORMANCE EVALUATION

A. Performance over the Flight Envelope

The characteristics of the three self-scheduled controllers over the flight envelope are depicted from Fig. 5 to Fig. 9, based on a gridding of both flight envelope and uncertain domain. For each operating point, only the worst-case performance characteristics over the uncertain domain are illustrated. It can be seen that the minimal time constant is lower than 1s over the main part of the flight envelope (Fig. 5) and gain margin requirements are met over the whole flight envelope.

TABLE II

| Normalized approach $P_l = 1/N_l$: Final $H_{\infty}$ norm of $H_l(s)$ |
|-----------------------------|--------|--------|--------|--------|--------|--------|--------|
| $h$ (m) | $M$ | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| 1000  | 1.64 | 1.46 | 1.32 | 1.30 | -   | -   |
| 5000  | -  | 1.65 | 1.54 | 1.46 | 1.43 | -   |
| 10000 | -  | -   | 1.69 | 1.68 | 1.65 | 1.61 |

TABLE III

| Normalized approach $P_l = 2/(1 + N_l)$: Final $H_{\infty}$ norm of $H_l(s)$ |
|-----------------------------|--------|--------|--------|--------|--------|--------|
| $h$ (m) | $M$ | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| 1000  | 1.59 | 1.46 | 1.41 | 1.39 | -   | -   |
| 5000  | -  | 1.58 | 1.51 | 1.48 | 1.46 | -   |
| 10000 | -  | -   | 1.61 | 1.61 | 1.59 | 1.57 |
flight envelope (Fig. 7 and 8). The most significant differences between these three self-scheduled controllers appear at the level of the damping (Fig. 6) and the phase margin (Fig. 9). Compared to the first self-scheduled controller, the first normalized approach allows improving the damping of the closed-loop system at low altitude and high velocity. Nevertheless, it leads to a degradation of the phase margin at low Mach number, going below the required 40°. The second normalized approach overcomes this limitation. The damping is considerably improved over the flight envelope while the phase margin is greater than 40° except at high altitude and high Mach number.

B. Closed-Loop Response

The self-scheduled controller designed in Subsection IV-C.2 has been validated by simulations using nonlinear dynamic model. The aircraft response in closed-loop to a control doublet is considered for 35 operating points over the flight envelope and for 5 mass and CG configurations (the nominal and the four extremes). In this test, the speed of the aircraft is supposed to be constant. The altitude is computed during the simulation based on initial condition and flight dynamic equations. Figure 4(a) depicts the temporal behavior of the load factor. It can be seen that time response and overshoot requirements are all met over the flight envelope within the considered parametric uncertainty domain. The constraint on the pitch rate velocity is respected with an overshoot less than 100%, 92% in the worst case (Fig. 4(b)). The temporal behavior of the actuator shows that there is no position or rate saturations on elevator deflection.

Note that for completing the clearance phase, the overall performance and the global stability of the closed-loop system have been validated for time invariant and time varying flight conditions and uncertain configurations, including mass parameters and aerodynamic coefficients, using μ-analysis and IQC (Integral Quadratic Constraints) analysis. Due to space limitation, the results are not reported in this paper.

VI. CONCLUSIONS AND FURTHER DEVELOPMENTS

The present work demonstrated an application of structured $H_{\infty}$ tools to the design of robust and self scheduled control systems with an a priori fixed architecture. The reformulation of the considered problem in the multi-objective optimization framework led to the computation of weakly Pareto optimal solutions. It is shown that the introduction of weighting coefficients in the design process can significantly improve the overall performance of the closed-loop system. Note that the results presented in this paper are obtained based on a ad hoc procedure to adjust the weighting coefficients. By representing the weighting coefficients as functions of a vector parameter, the tuning of weighing coefficients can be cast into control synthesis process. Thus, further performance improvement can be expected.

ACKNOWLEDGMENT

This work is supported in part by Industrial Innovation Scholarships program jointly offered by NSERC and FORNT (Fonds de Recherche du Québec Nature et Technologies) through a university-business partnership between Polytechnique Montréal and Bombardier Aerospace Inc. The authors would like to thank the industrial partners, Yann Le Masson and Jessica Allard-Bernier, from Bombardier Aerospace Inc.

REFERENCES

Fig. 5. Real part of the dominant pole

Fig. 6. Damping

Fig. 7. Positive gain margin

Fig. 8. Negative gain margin

Fig. 9. Phase margin

(a) Gain-scheduling
(b) Normalized approach 1
(c) Normalized approach 2

(a) Gain-scheduling
(b) Normalized approach 1
(c) Normalized approach 2

(a) Gain-scheduling
(b) Normalized approach 1
(c) Normalized approach 2

(a) Gain-scheduling
(b) Normalized approach 1
(c) Normalized approach 2

(a) Gain-scheduling
(b) Normalized approach 1
(c) Normalized approach 2

(a) Gain-scheduling
(b) Normalized approach 1
(c) Normalized approach 2

(a) Gain-scheduling
(b) Normalized approach 1
(c) Normalized approach 2

(a) Gain-scheduling
(b) Normalized approach 1
(c) Normalized approach 2

(a) Gain-scheduling
(b) Normalized approach 1
(c) Normalized approach 2