Gain Scheduling with Guardian Maps for Longitudinal Flight Control

David Saussié* and Lahcen Saydy†
École Polytechnique de Montréal, Montréal, Québec H3T 1J4, Canada
Ouassima Akhrif‡
École de Technologie Supérieure, Montréal, Québec H3C 1K3, Canada
and
Caroline Bérard§
Institut Supérieur de l'Aéronautique et de l'Espace, 31000 Toulouse, France

DOI: 10.2514/1.52178

A new approach to gain scheduling of linear controllers is proposed and applied to a longitudinal flight control problem. Traditionally, gain scheduling is done a posteriori by the interpolation of controller gains designed for several operating points or conditions. The method proposed here is based on guardian maps and does not require as many linear controller syntheses as there are design points. Rather, it extends the performance of an initial single controller carried out on an arbitrary operating point to the entire domain while ensuring generalized stability all along the process. The method, which uses a given fixed architecture controller, is successfully applied on the longitudinal flight control of a business jet aircraft.

Nomenclature

M	=	Mach number
n_z	=	normal load factor, g
q	=	pitch rate, deg/s
w	=	normal velocity perturbation, m/s
δ_c	=	commanded elevator angle deflection, rad
δ_e	=	elevator angle deflection, rad
ϵ	=	downwash angle, rad

I. Introduction

AMONG nonlinear control methods, gain scheduling is one of the most popular [1,2], and efforts have been made recently to put forth a unified framework for it [3]. There are mainly two approaches to designing gain-scheduled controllers. The first approach consists of designing a set of linear controllers corresponding to a given set of linearized models around equilibrium points [4–6]. These controllers are then interpolated versus scheduling variables, which may be exogenous and/or endogenous. Interpolation can, however, become a challenging issue when dynamic compensators are considered [7,8], and implementation problems can occur [5,9,10]. The second approach is based on linear parameter-varying (LPV) techniques [11,12] and ensures a priori global stability of the closed-loop system. LPV problems are formulated as linear matrix inequalities optimization problems , and controllers are directly obtained in an LPV form; consequently, no a posteriori interpolation is needed. Unfortunately, this approach can be rather conservative, especially when the operating domain becomes large, and it may lead to nonfeasible optimization problems.

The method proposed here lies somewhere in between these two categories and seeks to avoid their respective disadvantages while preserving their benefits. Unlike classic gain scheduling, for which the design has to be carried out over many equilibrium points, it only needs one initial controller. The procedure then automatically constructs controllers that satisfy the stability and performance requirements for other trim points until the entire operating domain is covered. Unlike LPV methods, it has the advantage that a fixed architecture controller can be considered. The final product is a set of automatically generated controllers for which the combined robust performance regions cover the entire evolution domain. Thereafter, these controllers can be easily interpolated with respect to scheduling variables, as they all have the same structure, or they can be implemented directly as lookup tables. The performance constraints considered here are of the generalized stability type; that is, they are set in terms of eigenvalue or pole confinement within specific subsets of the complex plane. These are handled by guardian maps [16], which provide indications on both the allowable gain regions (i.e., the gain regions within which the system's generalized stability is fulfilled) and on the controller robustness versus uncertain parameters.

A preliminary application was performed on a missile benchmark problem [17] and gave promising results compared with former papers [18–29]. But unlike the method in [17], the one presented here ensures stability and performance all along the process. A new application to the longitudinal flight control of an aircraft (*Challenger 604*) is also presented. Traditionally, flight controllers use simple controller structures derived from the wide experience acquired by flight control engineers; they have, indeed, well-studied and understood architectures [30] and work well in many practical applications. The present paper builds on the work initiated by Saydy et al. [31] and Saussié et al [32,33]. It proposes to conserve the original architecture provided by the manufacturer and to automatically schedule the gains with scheduling variables, such as Mach number M and altitude h, in order to satisfy given handling qualities; this remains one of the most important objectives in flight control design [34], and several criteria [35] may be used to this effect. A flight controller must ideally satisfy these handling qualities [36,37] over the entire flight domain. The longitudinal handling qualities mainly concern the short-period dynamics of the aircraft, and satisfying some of them can be cast in terms of eigenvalue confinement and zero location (mainly for Gibson's dropback [37]).

Received 24 September 2010; revision received 28 January 2011; accepted for publication 8 February 2011.

*Ph.D. Student, Department of Electrical Engineering; Institut Supérieur de l'Aéronautique et de l'Espace, 31000 Toulouse, France; david.saussie@polymtl.ca. Member AIAA.

†Professor, Department of Electrical Engineering; lahcen.saydy@polymtl.ca.

‡Professor, Department of Electrical Engineering; ouassima.akhrif@etsmtl.ca. Member AIAA.

§Professor, Department of Mathematics, Computer Science and Control Theory; caroline.berard@isae.fr.

A first controller satisfying the handling qualities at stake is provided for a given flight condition, and the goal is to extend its performance to the rest of the flight envelope by scheduling its gains with respect to flight condition parameters (Mach and altitude).

The paper is organized as follows. A brief overview of guardian maps is proposed in Sec. I. Next, two algorithms are proposed to extend the performance of an initial controller to the whole operating domain, and thus yield a scheduled controller. In Sec. III, the aircraft model of a Bombardier, Inc., *Challenger 604* is introduced, as well as the handling qualities of interest. Finally, Sec. IV is devoted to the application of the technique to the design of a scheduled flight controller for the longitudinal motion of the plane.

II. Guardian Maps: Brief Review

The guardian map approach was introduced by Saydy et al. [16] as a unifying tool for the study of generalized stability of parameterized families of matrices or polynomials. Here, generalized stability means confinement of matrix eigenvalues or polynomial zeros to general open subsets of the complex plane and includes the open left half-plane and the unit circle as special cases. Some of the basic concepts are introduced.

A. Guardian Maps

Basically, guardian maps are scalar-valued maps defined on the set of $n \times n$ real matrices (or nth-order polynomials) that take nonzero values on the set of Ω-stable matrices (or polynomials) and vanish on its boundary. The following description will focus on families of matrices, with the understanding that it applies to polynomials as well. The stability sets of interest are of the form

$$S(\Omega) = \{A \in \mathbb{R}^{n\times n}\colon \sigma(A) \subset \Omega\} \tag{1}$$

where Ω is an open subset of the complex plane of interest, and $\sigma(A)$ denotes the set consisting of the eigenvalues of A. Such sets $S(\Omega)$ will be referred to as generalized stability sets; thus, they represent the set of all matrices that are stable relative to Ω, i.e., which have all their eigenvalues in Ω.

Definition II.1: Let ν map $\mathbb{R}^{n\times n}$ into $\mathbb{C}$. By definition, ν guards $S(\Omega)$ if for all $A \in \bar{S}(\Omega)$, the following equivalence holds:

$$\nu(A) = 0 \Leftrightarrow A \in \bar{S}(\Omega) \tag{2}$$

Here, $\bar{S}$ denotes closure of the set S. The map is said to be polynomic if it is a polynomial function of the entries of its argument.

Example II.1: Some guardian maps are given for classical regions (Fig. 1):

1) For the Hurwitz stability, for $\Omega = \mathring{\mathbb{C}}_-$, a guardian map is

$$\nu_H(A) = \det(A \odot I)\det(A) \tag{3}$$

where $\odot$ denotes the bialternate product [38] (see Appendix A).

2) For the stability margin, the open α-shifted half-plane region has a corresponding guardian map

$$\nu_m(A) = \det(A \odot I - \alpha I \odot I)\det(A - \alpha I) \tag{4}$$

3) The conic sector with inner angle 2θ has a corresponding guardian map given by

$$\nu_d(A) = \det[A^2 \odot I + (1 - 2\zeta^2)A \odot A]\det(A) \tag{5}$$

where $\zeta \triangleq \cos\theta$ denotes the limiting damping ratio.

4) For the Schur stability, for the circle of radius $\omega > 0$, a corresponding guardian map is

$$\nu_p(A) = \det(A \odot A - \omega^2 I \odot I)\det(A^2 - \omega^2 I) \tag{6}$$

A systematic way of constructing guardian maps for various Ω regions can be found in Saydy et al. [16].

B. Stabilizing Gain Characterization

Let $\{A(r)\colon r \in U \subset \mathbb{R}^k\}$ be a continuous family of $n \times n$ matrices that depends on the (usually) uncertain parameter vector $r := (r_1, \ldots, r_k)$, where each entry lies in a given range for which only the bounds are known: say, $r \in U \subset \mathbb{R}^k$.

Theorem II.1: Let $S(\Omega)$ be guarded by the map ν_Ω. The family $\{A(r)\colon r \in U\}$ is stable relative to Ω if, and only if: a) it is nominally stable [i.e., $A(r_0) \in S(\Omega)$ for some $r_0 \in U$), and b) $\forall\, r \in U$, $\nu_\Omega[A(r^0)]\nu_\Omega[A(r)] > 0$ (i.e., $\nu_\Omega[A(r)]$ does not vanish in U).

Corollary II.1: Let $S(\Omega)$ be guarded by the map ν and consider the family $\{A(r)\colon r \in U\}$. Then, C defined by

$$C = \{r \in \mathbb{R}^k\colon \nu_\Omega[A(r)] = 0\} \tag{7}$$

and divides the parameter space $\mathbb{R}^k$ into components C_i that are either stable or unstable relative to Ω. To see which situation prevails for a given component C_i, one only has to test $A(r)$ for any one point in C_i.

Example II.2: Suppose that the closed-loop poles of a given system are specified by the polynomial:

$$p(s) = s^3 + k_1 s^2 + k_2 s + 1 \tag{8}$$

where k_1 and k_2 denote some controller gains. If the damping region $\zeta > 0.7$ is the one considered (Fig. 2), then one obtains [e.g., by applying Eq. (5) to the companion matrix corresponding to p]

$$\nu_\Omega(p) = 2k_2^3 - k_1^2 k_2^2 - 4k_1 k_2 + 2k_1^3 + 1 \tag{9}$$

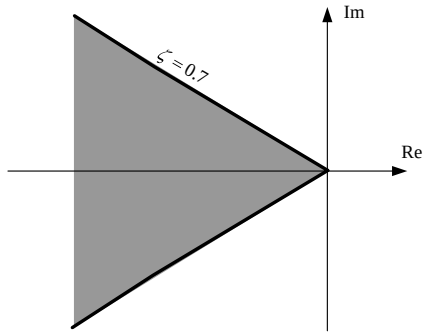


Fig. 2 Stability region Ω.

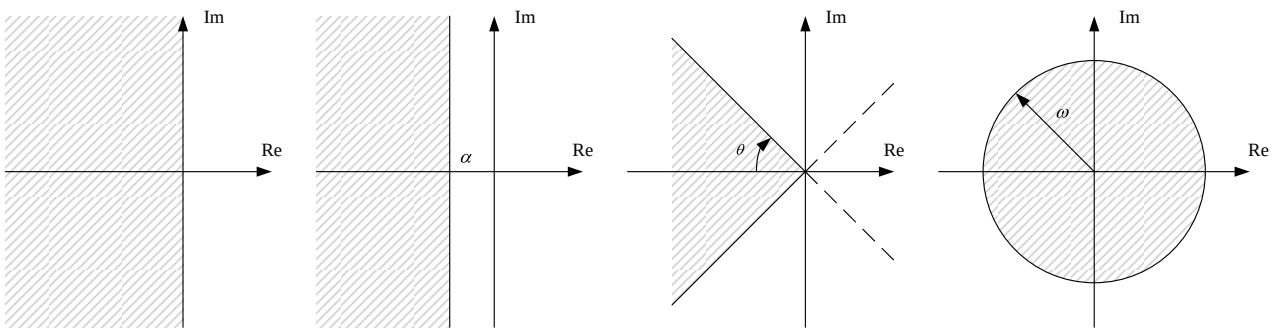


Fig. 1 Regions for Example II.1.

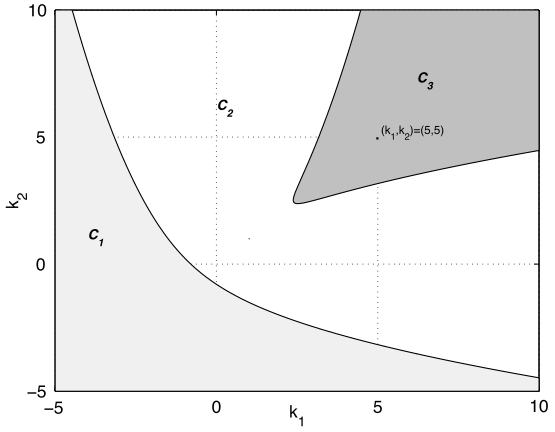


Fig. 3 Set C_3 of all gains ensuring Ω stability.

Setting this quantity to zero yields the three components in the parameter space (k_1, k_2) of Fig. 3. It can be verified that the set of all gains (k_1, k_2) that place all the closed-loop poles within the preceding damping zone is the component C_3. Any other choice of the gains outside of C_3 yields closed-loop poles outside the damping conic region. One arrives to this conclusion simply by testing Ω stability of $p(s)$ for any three pairs (k_1, k_2) in C_1, C_2, and C_3, respectively.

C. Robust Stability

For the application at hand, the model is given as linearized state-space models for several operating points that depend on two parameters, namely, Mach number M and altitude h. Consequently, the stability of one- and two-parameter families of real matrices relative to a domain Ω is considered.

1. *One-Parameter Family Stability Test*

Single-parameter polynomial matrices of the following form are considered:

$$A(r) = A_0 + rA_1 + \cdots + r^k A_k \tag{10}$$

with A_i given constant matrices, and such that $A(r_0)$ is Ω stable. The corresponding guardian map $\nu_\Omega[A(r)]$ is a polynomial in r. Let

$$r^- \doteq \sup\{r < r_0\colon \nu_\Omega[A(r)] = 0\}(\text{or} - \infty \text{ if none exist})$$
$$r^+ \doteq \inf\{r > r_0\colon \nu_\Omega[A(r)] = 0\}(\text{or} + \infty \text{ if none exist})$$

be the maximal perturbation bounds for nonsingularity of matrices around $r = r_0$.

Lemma II.1: Let $A(r) = A_0 + rA_1 + \cdots + r^k A_k$ be a polynomial matrix in the uncertain parameter r real with given constant matrices A_i, such that $A(r_0)$ is stable with respect to Ω, and let $S(\Omega)$ be guarded by a map ν_Ω. Then, $A(r)$ is stable relative to Ω for all $r \in (r^-, r^+)$. Furthermore, this interval is the largest one containing r_0.

2. *Two-Parameter Family Stability Test*

Let us consider the stability of two-parameter polynomial families of real matrices (or polynomials) relative to a domain Ω for which $S(\Omega)$ is endowed with a polynomic guardian map ν_Ω:

$$A(r_1, r_2) = \sum_{i=0}^{i=k} \sum_{j=0}^{j=l} r_1^i r_2^j A_{ij} \tag{11}$$

with r_1 and r_2 real parameters. The theorem by Saydy et al. [16] gives necessary and sufficient conditions for the stability of $A(r_1, r_2)$ over a specific rectangle-shaped domain $(r_1, r_2) \in [\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$. A novel corollary is stated here in order to find the largest stable open rectangle with a fixed side, as will be explained next.

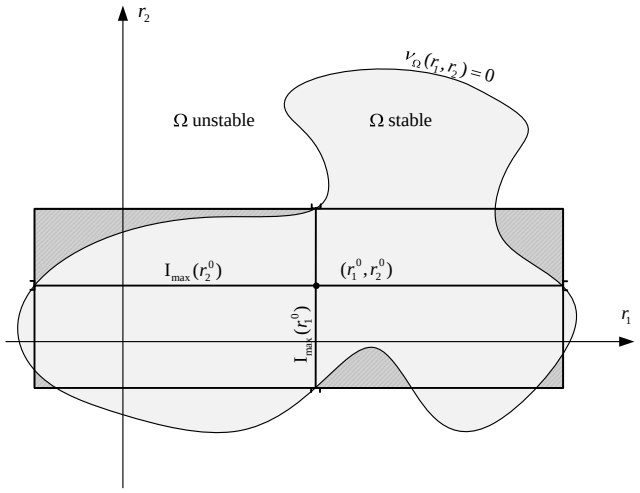


Fig. 4 Instability of $I_{max}(r_2^0) \times I_{max}(r_1^0)$.

Let $r^0 \triangleq (r_1^0, r_2^0)$ be a nominal parameter vector such that $A(r_1^0, r_2^0)$ is Ω stable. By freezing one of the parameters (say, $r_2 = r_2^0$), one can easily compute the largest (open) interval containing r_1^0, $I_{\max}(r_2^0) \triangleq (r_1^-, r_1^+)$, such that $\forall\ r_1 \in I_{\max}(r_2^0), A(r_1, r_2^0) \in S(\Omega)$ (Lemma II.1). Similarly, for $r_1 = r_1^0$ fixed, one can obtain the largest (open) interval containing r_2^0, $I_{\max}(r_1^0) \triangleq (r_2^-, r_2^+)$, such that $\forall\ r_2 \in I_{\max}(r_1^0)$, $A(r_1^0, r_2) \in S(\Omega)$. When both parameters vary, it is, of course, not true in general that the family is Ω stable within $I_{\max}(r_2^0) \times I_{\max}(r_1^0)$ (see Fig. 4 for an illustration).

Therefore, given a nominal parameter vector $r^0 \triangleq (r_1^0, r_2^0)$ and fixed side $[\alpha_1, \beta_1] \subset I_{\max}(r_2^0)$ containing r_1^0, the largest open interval (α_2^*, β_2^*) is sought, such that

$$\forall\ (r_1, r_2) \in [\alpha_1, \beta_1] \times (\alpha_2^*, \beta_2^*), A(r_1, r_2) \in S(\Omega) \tag{12}$$

as illustrated in Fig. 5. Naturally, one can choose $[\alpha_2, \beta_2] \subset I_{\max}(r_1^0)$ and seek the largest open interval (α_1^*, β_1^*), such that

$$\forall\ (r_1, r_2) \in (\alpha_1^*, \beta_1^*) \times [\alpha_2, \beta_2], A(r_1, r_2) \in S(\Omega) \tag{13}$$

Before stating the corollary, which enables statements such as the one in Eq. (12) or in Eq. (13), some notations are introduced. Let

$$\nu_1(r_2) \triangleq p_{[r_2]}(\alpha_1) p_{[r_2]}(\beta_1) \tag{14}$$

and

$$\nu_2(r_2) \triangleq \det B(p_{[r_2]}, p'_{[r_2]}) \tag{15}$$

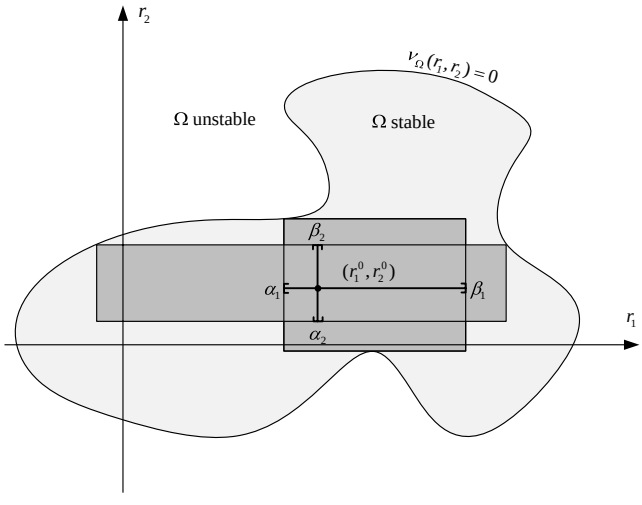


Fig. 5 Largest stable rectangle with a fixed side.

where, for each r_2, $p_{[r_2]}(r_1) \triangleq \nu_\Omega[A(r_1, r_2)]$ denotes the univariate polynomial in r_1, the coefficients of which are themselves polynomials in r_2, $p'_{[r_2]}$ denotes its derivative with respect to r_1, and $B(a, b)$ designates the Bezoutian of two polynomials, a and b (see Appendix B).

Corollary II.2: Let $A(r_1^0, r_2^0) \in S(\Omega)$ and let ν_Ω be a guardian map for $S(\Omega)$. Choose $[\alpha_1, \beta_1] \subset I_{\max}(r_2^0)$. Define $\underline{r}_2$ as the largest root of $\nu_1(r_2)$ strictly less than r_2^0 (set to $-\infty$ if no such root exists), and $\underline{r}_2$ as the smallest root of $\nu_1(r_2)$ strictly larger than r_2^0 (set to $+\infty$ if no such root exists)

Also define α_2^* as the largest root μ of $\nu_2(r_2)$, $\underline{r}_2 \le \mu < r_2^0$, for which $p_{[\mu]}(r_1)$ has a root in $[\alpha_1, \beta_1]$ (set to $\underline{r}_2$ if none exists); and β_2^* as the smallest root μ of $\nu_2(r_2)$, $r_2^0 < \mu \le \bar{r}_2$, for which $p_{[\mu]}(r_1)$ has a root in $[\alpha_1, \beta_1]$ (set to $\bar{r}_2$ if none exists).

Thus, (α_2^*, β_2^*) is the largest open interval, such that

$$\forall\ (r_1, r_2) \in [\alpha_1, \beta_1] \times (\alpha_2^*, \beta_2^*), A(r_1, r_2) \in S(\Omega) \tag{16}$$

Proof: First, let $[\alpha_2, \beta_2] \subset (\alpha_2^*, \beta_2^*)$. Let us prove that the family $A(r_1, r_2)$ is stable on $[\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$; that is, $\nu_\Omega(r_1, r_2) = p_{[r_2]}(r_1)$ does not vanish for any $(r_1, r_2) \in [\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$.

Q6 According to Theorem [16], $A(r_1, r_2)$ is stable for all $(r_1, r_2) \in [\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$ if, and only if, $U_{\text{cr}}^1 = \varnothing$ and the univariate polynomials p_{α_2} and $p_{[r_2]}$, $r_2 \in U_{\text{cr}}^2$, have no zeros in $[\alpha_1, \beta_1]$, where

$$U_{\text{cr}}^1 := \{r_2 \in [\alpha_2, \beta_2]\colon\ \nu_1(r_2) = 0\} \tag{17}$$

and

$$U_{\text{cr}}^2 := \{r_2 \in [\alpha_2, \beta_2]\colon\ \nu_2(r_2) = 0\} \tag{18}$$

According to the definitions of $\underline{r}_2$, $\bar{r}_2$, α_2^*, and β_2^*, one has $\underline{r}_2 \le \alpha_2^* < \alpha_2 \le \beta_2 < \beta_2^* \le \bar{r}_2$. Consequently, it follows by construction that $U_{\text{cr}}^1 = \varnothing$, as $\underline{r}_2$ and $\bar{r}_2$ are, respectively, the largest (less than r_2^0) and the smallest (greater than r_2^0) roots of $\nu_1(r_2)$. Similarly, p_{α_2} and $p_{[r_2]}$, $r_2 \in U_{\text{cr}}^2$ have no zeros in $[\alpha_1, \beta_1]$ for, indeed, α_2^* and β_2^* are, respectively, the largest (less than r_2^0) and the smallest (greater than r_2^0) roots of $\nu_1(r_2)$ for which such a property holds. This proves Ω stability of $A(r_1, r_2)$ for all $[\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$ with $\alpha_2^* < \alpha_2 \le \beta_2 < \beta_2^*$.

By definition, α_2^* and β_2^* are either ∞ ($-\infty$ and $+\infty$, respectively) or are such that either $A(r_1, \alpha_2^*)$ or $A(r_1, \beta_2^*)$ is Ω unstable for some value of $r_1 \in [\alpha_1, \beta_1]$. Therefore, any choice of $\alpha_2 \le \alpha_2^*$ or $\beta_2 \ge \beta_2^*$ would lead, with the same proof, to the instability of $A(r_1, r_2)$ over $[\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$. Consequently, (α_2^*, β_2^*) is the largest open interval, such that

$$\forall\ (r_1, r_2) \in [\alpha_1, \beta_1] \times (\alpha_2^*, \beta_2^*), A(r_1, r_2) \in S(\Omega) \tag{19}$$

□

Example II.3: To illustrate the preceding, let the closed-loop poles of a given system be specified by the uncertain polynomial

$$d(s) = s^3 + s^2 + s + r_1^2 + r_2^2 \tag{20}$$

where r_1 and r_2 denote uncertain parameters with nominal values $r_1^0 = r_2^0 = 0.5$. If the open left half-plane $\Omega = \mathring{\mathbb{C}}_-$ is the stability region considered, then a guardian map is

$$\nu_\Omega(r_1, r_2) = (r_1^2 + r_2^2)(1 - r_1^2 - r_2^2) \tag{21}$$

By setting ν_Ω to zero, we easily see that the stable component is the interior of the unit disk, except the origin (light gray region in Fig. 6). It follows that

$$p_{[r_2]}(r_1) = -r_1^4 + (1 - 2r_2^2)r_1^2 + r_2^2 - r_2^4 \tag{22}$$

$$p'_{[r_2]}(r_1) = -4r_1^3 + (2 - 4r_2^2)r_1 \tag{23}$$

and the corresponding Bezoutian matrix is

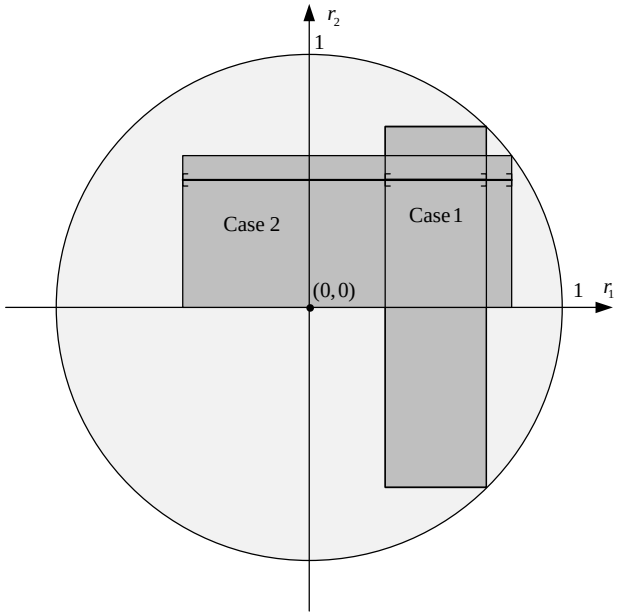


Fig. 6 Example.

$$B(p_{r_2}, p'_{r_2}) = \begin{bmatrix} 0 & 4r_2^2 - 4r2^4 & 0 & -2r_2^2 + 6r_2^4 - 4r_2^6 \\ -2 + 4r_2^2 & 0 & 2 - 4r_2^2 + 4r_2^4 & 0 \\ 0 & 2 - 4r_2^2 & 0 & 4r_2^2 - 4r_2^4 \\ 4 & 0 & -2 + 4r_2^2 & 0 \end{bmatrix} \tag{24}$$

which yields

$$\nu_2(r_2) = \det B(p_{[r_2]}, p'_{[r_2]}) = 16r_2^2(r_2^2 - 1) \tag{25}$$

Letting $(r_1^0, r_2^0) = (0.5, 0.5)$ yields $I_{\max}(r_2^0) \approx (-0.866, 0.866)$. By choosing, for instance, $[0.3, 0.7] \subset I_{\max}(r_2^0)$, one obtains

$$\nu_1(r_2) \triangleq p_{[r_2]}(0.3)p_{[r_2]}(0.7) = (r_2^4 - 0.82r_2^2 - 0.0819)(r_2^4 - 0.02r_2^2 - 0.2499) \tag{26}$$

the real roots of which are $\{-0.9539, -0.7141, 0.7141, 0.9539\}$. Therefore, $\underline{r}_2 = -0.7141$ and $\bar{r}_2 = 0.7141$. The roots of $\nu_2(r_2)$ are $\{-1, 0, 1\}$. Since none of the polynomials, $p_{[0]}(r_1)$ and $p_{[1]}(r_1)$, has a root in $[0.3, 0.7]$, one can conclude, according to Corollary II.2, that $(-0.7141, 0.7141)$ is the largest open interval, such that

$$\forall\ (r_1, r_2) \in [0.3, 0.7] \times (-0.7141, 0.7141), \qquad d(s) \in S(\Omega) \tag{27}$$

If, instead, the interval $[-0.5, 0.8] \subset I_{\max}(r_2^0)$ is chosen, one obtains

$$\nu_1(r_2) \triangleq p_{[r_2]}(-0.5)p_{[r_2]}(0.8) = (r_2^4 - 0.5r_2^2 - 0.1875)(r_2^4 + 0.28r_2^2 - 0.2304) \tag{28}$$

with roots $\{-0.866, -0.6, 0.6, 0.866\}$. In this case, $\underline{r}_2 = -0.6$ and $\bar{r}_2 = 0.6$. The roots of $\nu_2(r_2)$ are $\{-1, 0, 1\}$, and $p_{[0]}(r_1)$ has a root (namely zero) in $[-0.5, 0.8]$. Therefore, according to Corollary II.2, $(0, 0.6)$ is the largest open interval, such that

$$\forall\ (r_1, r_2) \in [-0.5, 0.8] \times (0, 0.6), \qquad d(s) \in S(\Omega) \tag{29}$$

III. Gain-Scheduling Procedure

In typical gain-scheduling techniques, LTI controllers have to be designed on different linearized models; controller interpolation is Q7

done a posteriori or switching laws are implemented between the various controllers. In the interpolation approach case, even if the controllers designed on each operating point fulfill the requirements locally, there is no guarantee that between the synthesis points stability is retained, especially if the designer did not take enough synthesis points [4,9]. Moreover, depending on controller complexity, interpolation problems may arise. For example, in the case of H_∞ control or μ synthesis, an initial order reduction phase is often required on each LTI controller in order to ensure that all the controllers have the same order and structure [21]. In the switching approach case, the designed controllers must cover the entire domain and, again, the number of synthesis points is crucial. So the three major issues are the number of synthesis points, the controller structure, and the stability and performance satisfaction on the entire domain [1,2]. The present method proposes to address these issues.

The basic idea is as follows. Let $\boldsymbol{K}^0$, an initial fixed-structure controller, be designed for a particular trim condition. This controller naturally presents some robust performance margins with respect to trim variables; that is, it will no longer ensure performance for a limit trim condition. By adjusting $\boldsymbol{K}^0$ (according to a method based on guardian maps; see Appendix C), we obtain a new controller $\boldsymbol{K}^1$ having its own robust performance margins. By taking another worst-case condition where $\boldsymbol{K}^1$ fails, another controller $\boldsymbol{K}^2$ is designed, and so on, until covering the entire operating domain.

In what follows, two gain-scheduling algorithms are introduced: one for the single-parameter case (one scheduling variable) and one for the two-parameter case (two scheduling variables). In virtue of Lemma II.1 and Corollary II.2, generalized stability will be ensured all along the procedure. The algorithms automatically produce a set of controllers for which the combined robust performance regions cover the entire operating domain. They will prove efficient in the longitudinal motion control problem, as only two parameters, Mach number and altitude, are involved.

A. Single-Parameter Gain-Scheduling Algorithm

The first algorithm proposed is applied in the case of a system with one single parameter r. Let Ω be the region of eigenvalue confinement of interest and ν_Ω be a corresponding guardian map. Let $A(r, \boldsymbol{K})$ denote the closed-loop state-space matrix with $\boldsymbol{K} = [K_i]$ as the gain vector. With a slight abuse in notation, let us denote $\nu_\Omega(r, \boldsymbol{K}) := \nu_\Omega[A(r, \boldsymbol{K})]$. If $A(r, \boldsymbol{K})$ depends polynomially on the parameters, and the boundary of Ω is defined polynomially, then $\nu_\Omega(r, \boldsymbol{K})$ is a multivariable polynomial as well. We seek to find $\boldsymbol{K}$ (scheduled with respect to r) that stabilizes the system for $r \in [r_{\min}, r_{\max}]$.

For an initial parameter value $r_0 = r_{\min}$, let $\boldsymbol{K}^0$ be a nominal choice of stabilizing gains: that is, such that the eigenvalues of $A(r_0, \boldsymbol{K}^0)$ are inside Ω. With $\boldsymbol{K} = \boldsymbol{K}^0$, Lemma II.1 is applied to find the largest stability interval $]\underline{r}_0, \bar{r}_0[$. Thus, the vector $\boldsymbol{K}^0$ stabilizes the system for any parameter r in $]\underline{r}_0, \bar{r}_0[$. If $\bar{r}_0 = +\infty$ or $\bar{r}_0 > r_{\max}$, one can stop as $\boldsymbol{K}^0$ ensures stability $\forall\ r \in [r_{\min}, r_{\max}]$. Moreover, $\nu_\Omega(\underline{r}_0, \boldsymbol{K}^0) = \nu_\Omega(\bar{r}_0, \boldsymbol{K}^0) = 0$ if $\underline{r}_0$ and $\bar{r}_0$ happen to be finite.

If $\bar{r}_0 \le r_{\max}$, one proceeds as follows. Fix $r_1 = \bar{r}_0$. The equation $\nu_\Omega(r_1, \boldsymbol{K}) = 0$ defines new components in the space of gain parameters that are either stable or unstable (Corollary II.1). By definition of $\bar{r}_0$, $\nu_\Omega(\bar{r}_0, \boldsymbol{K}^0) = 0$ and $\boldsymbol{K}^0$ lies on the boundary of a stable component. In [39] (see Appendix C), an algorithm was developed to search inside a component in order to find a new vector $\boldsymbol{K}^1$ that places closed-loop poles strictly inside Ω. This new choice leads to a new stability interval $]\underline{r}_1, \bar{r}_1[$ with $\bar{r}_0 \in]\underline{r}_1, \bar{r}_1[$. The same steps are repeated until possibly covering all values of parameter $r \in [r_{\min}, r_{\max}]$.

If the algorithm succeeds, it yields a sequence of controllers $\{\boldsymbol{K}^0, \ldots, \boldsymbol{K}^i, \ldots, \boldsymbol{K}^n\}$ satisfying all the criteria on the corresponding intervals $\{[r_{\min}, \bar{r}_0[, \ldots,]\underline{r}_i, \bar{r}_i[, \ldots,]\underline{r}_n, r_{\max}]\}$. Moreover, the entire parameter range $[r_{\min}, r_{\max}]$ is covered since, by construction, $\underline{r}_{i+1} < \bar{r}_i$. The user is then free to exploit this set of controllers, depending on the way they will be implemented: lookup tables, switching controllers, or interpolation of the data to name a few.

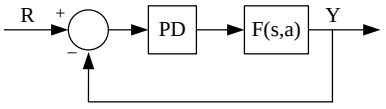


Fig. 7 Feedback configuration.

Remark III.1: Going "rightward" from $r_{\min}$ to $r_{\max}$ is an arbitrary choice, and one can easily adapt the algorithm to make it work "leftward". This would lead to different results in general.

Example III.1: The following simple example illustrates the procedure for the synthesis of a one-parameter fixed-structure scheduled controller: namely, a PD controller. The system at stake is described by a second-order one-parameter transfer function:

$$F(s, a) = \frac{1}{s^2 + 0.2a(a-10)s + a^2} \tag{30}$$

where the parameter $a \in [0, 10]$. The objective is to design a scheduled PD controller (Fig. 7) $C(s) = K_p(a) + K_d(a)s$ that places the closed-loop poles inside the region Ω defined by Fig. 8, where open-loop poles (unstable) are indicated for some values of a.

The closed-loop corresponding state-space matrix is given by

$$A(a, K_p, K_d) = A_0 + A_1 a + A_2 a^2 \tag{31}$$

with

$$A_0 = \begin{bmatrix} 0 & 1 \\ -K_p & -K_d \end{bmatrix} \qquad A_1 = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 0 & 0 \\ -1 & -0.2 \end{bmatrix} \tag{32}$$

A corresponding guardian map ν_Ω is

$$\nu_\Omega(A) = \nu_m(A)\nu_p(A)\nu_d(A) = \nu_\Omega(a, K_p, K_d) \tag{33}$$

with, by applying Eqs. (4–6),

$$\nu_m(A) = \tfrac{1}{2}(10 - K_d + 2a - 0.2a^2)(K_p + 10a - 5K_d + 25) \tag{34}$$

$$\begin{aligned}\nu_p(A) = {} & (a^2 + K_p - 144)(144 - 24a + K_p + 3.4a^2 \\ & + 12K_d)(144 + 24a + K_p - 1.4a^2 - 12K_d)\end{aligned} \tag{35}$$

$$\nu_d(A) = \tfrac{1}{2}[(K_d + 0.2a^2 - 2a)^2 - 2a^2 - 2K_p](a^2 + K_p) \tag{36}$$

For a fixed value $a = a_0$, Corollary II.1 states that vanishing $\nu_\Omega(a_0, K_p, K_d)$ divides the gain parameter space $\mathbb{R}^2$ into components that are either stable or unstable relative to Ω. One can then obtain the admissible gain regions, ensuring that all closed-loop poles are inside

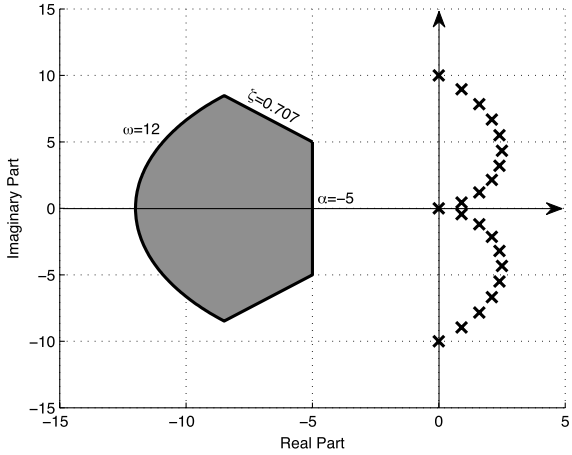
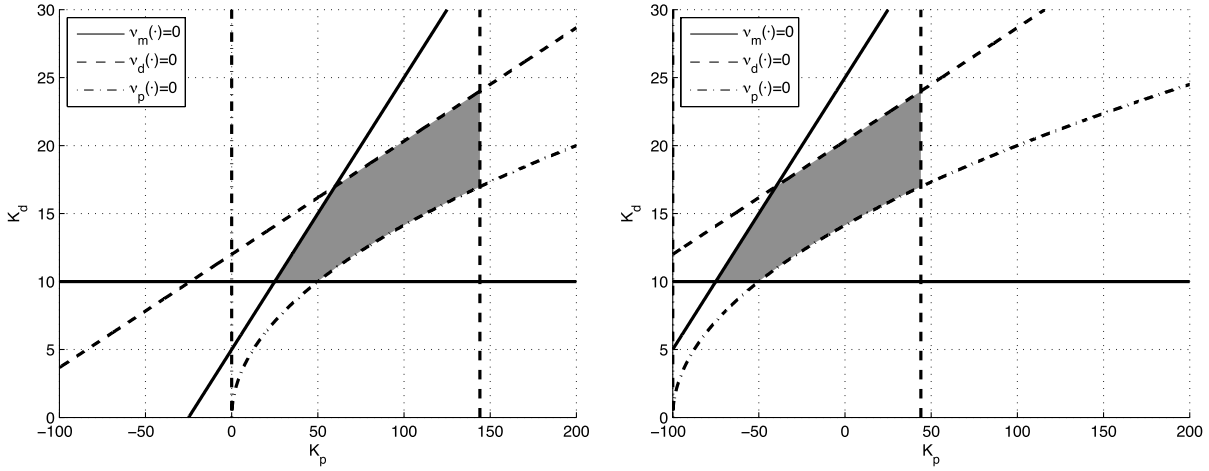


Fig. 8 General stability target region and open-loop poles.



a) a = 0 **b) a = 10**

Fig. 9 Stabilizing gain regions for different values of $a = \{0, 10\}$.

Table 1 Algorithm steps

Step i	K_p^i	K_d^i	Stability domain $]\underline{a}_i \bar{a}_i[$
0	106.3	17.7	$]-1.447, 1.778[$
1	103.1	20.6	$]-0.0243, 4.479[$
2	86.2	22.6	$]1.964, 7.599[$
3	48.6	21.3	$]3.327, 9.768[$
4	10.9	18.1	$]5.494, 11.535[$

the target region Ω (see Fig. 9, where shaded areas are Ω stable). For the $a = 0$ and $a = 10$ cases, the intersection is empty, implying that there exists no single PD controller satisfying the constraints for both extreme values. Scheduling the controller gains is therefore necessary.

For step 0, a first controller $(K_p^0, K_d^0) = (106.3, 17.7)$ is designed for $a = 0$, which satisfies the constraints with closed-loop poles $-8.85 \pm 5.29j$ ($\zeta = 0.858$, $\omega_n = 10.3$). The application of Lemma II.1 yields that this controller is robust for $a \in]-1.447, 1.778[$.

For step 1.1, we set $a = 1.778$ as the upper robust stability limit and look for another gain (K_p, K_d) inside the stable component defined by $\nu_\Omega(1.778, K_p, K_d) = 0$ [Eq. (33)]. The search algorithm yields the new controller $(K_p^1, K_d^1) = (103.1, 20.6)$.

For step 1.2, the PD controller (K_p^1, K_d^1) is found to be stabilizing with respect to Ω for $a \in]-0.0243, 4.479[$.

Three other iterations lead to the results compiled in Table 1. Finally, one has a set of five controllers for which the stability domain union covers the entire range of a. This sequence of stabilizing controllers can then be exploited to find a polynomial interpolation through the gains (K_p^i, K_d^i). The scheduled PD gains with respect to a are those of Eqs. (37) and (38) and are illustrated in Fig. 10. Figure 11 shows that the closed-loop poles are inside the target region Ω for different values of a. After replacing the gains by their scheduled expressions, one must check that the polynomial $\nu_\Omega(A) = \nu_\Omega(a)$ does not vanish for any $a \in [0, 10]$ assessing that the closed-loop poles remain inside the region Ω:

$$K_p(a) = -10.12a + 118.78 \tag{37}$$

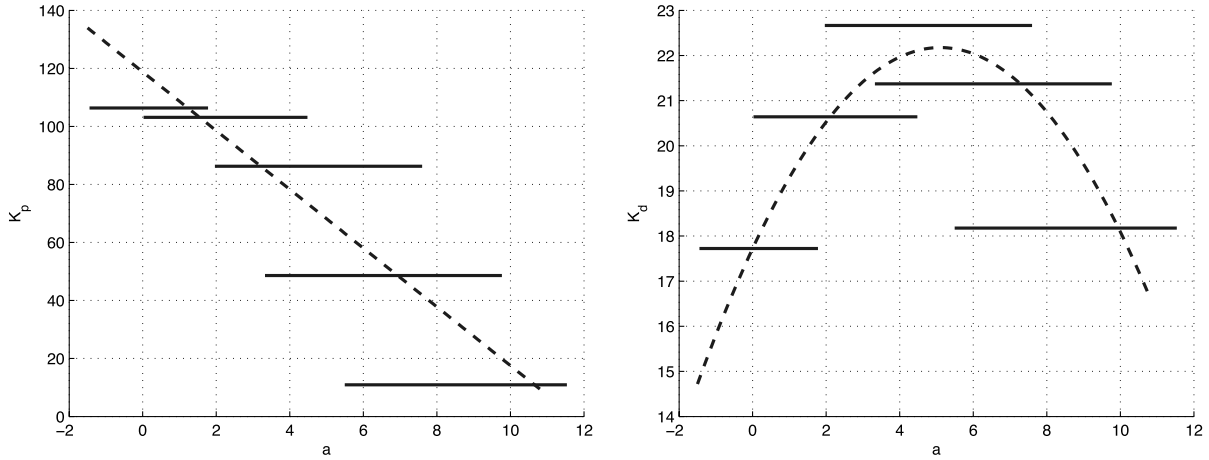
$$K_d(a) = -0.17a^2 + 1.74a + 17.72 \tag{38}$$

B. Two-Parameter Gain-Scheduling Algorithm

A new algorithm is introduced to handle two-parameter families. Let $A(r_1, r_2, \boldsymbol{K})$ denote the closed-loop state-space matrix, with $\boldsymbol{K} = [K_i]$ as the gain vector and Ω as the target stability region. A controller $\boldsymbol{K}(r_1, r_2)$ scheduled with respect to r_1 and r_2 is sought; it must place the poles within Ω for all

$$(r_1, r_2)[r_{1,\min}, r_{1,\max}] \times [r_{2,\min}, r_{2,\max}]$$

The algorithm is mainly based upon Algorithm 1 and alternates between synthesis of scheduled controllers with respect to r_1 and robustness analysis of these controllers along r_2.



a) Scheduled gain K_p with respect to a **b) Scheduled gain K_d with respect to a**

Fig. 10 Algorithm results and scheduled gains with respect to a.

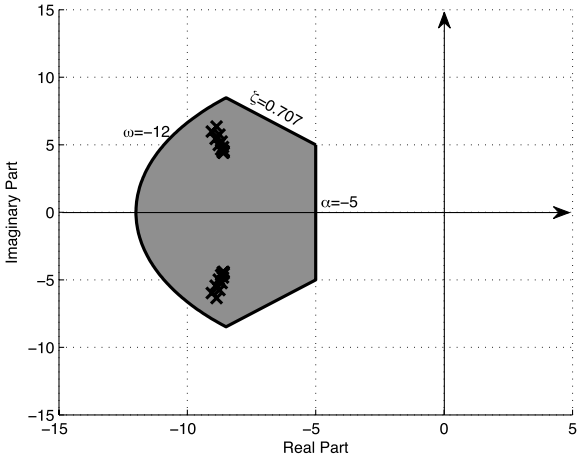


Fig. 11 Eigenvalue value confinement with scheduled controller.

The main idea of the algorithm is as follows. For a fixed $r_2 = r_{2,0} = r_{2,\min}$, use Algorithm 1 to find a stabilizing scheduled gain $\boldsymbol{K}^0(r_1)$ over $[r_{1,\min}, r_{1,\max}]$, and then use Corollary II.2 to find the largest open interval $(\underline{r}_{2,0}, \bar{r}_{2,0})$ containing $r_{2,0}$. Therefore, the scheduled gain $\boldsymbol{K}^0(r_1)$ stabilizes the system on $[r_{1,\min}, r_{1,\max}] \times (\underline{r}_{2,0}, \bar{r}_{2,0})$. Set $r_2 = r_{2,1} = \bar{r}_{2,0}$ and find a scheduled gain $\boldsymbol{K}^1(r_1)$ $[r_{1,\min}, r_{1,\max}]$; it stabilizes the system over $[r_{1,\min}, r_{1,\max}] \times (\underline{r}_{2,1}, \bar{r}_{2,1})$. The procedure must be repeated until it covers the entire domain. This leads to Algorithm 2, for which the first steps are illustrated in Fig. 12.

The algorithm yields a sequence of scheduled controllers $\{\boldsymbol{K}^0(r_1), \ldots, \boldsymbol{K}^i(r_1), \ldots, \boldsymbol{K}^n(r_1)\}$ stabilizing the system relative to Ω on

$$[r_{1,\min}, r_{1,\max}] \times \{[r_{2,\min}, \bar{r}_{2,0}), \ldots, (\underline{r}_{2,i}, \bar{r}_{2,i}), \ldots, (\underline{r}_{2,n}, r_{2,\max}]\}$$

Moreover, the entire parameter range $[r_{2,\min}, r_{2,\max}]$ is covered since, by construction, $\underline{r}_{2,i+1} < \bar{r}_{2,i}$. The interpolation technique is left to the user.

IV. Flight Controller Problem

The Bombardier, Inc., *Challenger 604* aircraft longitudinal control problem is described. After a short description of the open-loop model and its LPV description, the retained controller architecture and the requirements, also known as handling qualities, are introduced. A scheduled controller that will be effective on the entire flight envelope is sought.

A. Open-Loop Linear Parameter-Varying Description

For design purposes, only the short-period equations of motion are considered:

$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\epsilon} \end{bmatrix} = \begin{bmatrix} Z_w & U_0 & Z_\epsilon \\ M_w & M_q & M_\epsilon \\ E_w & 0 & E_\epsilon \end{bmatrix} \begin{bmatrix} w \\ q \\ \epsilon \end{bmatrix} + \begin{bmatrix} Z_{\delta_e} \\ M_{\delta_e} \\ 0 \end{bmatrix} \delta_e \tag{39}$$

At trimmed level flight, the dimensional coefficients Z_w, Z_ϵ, M_w, M_q, M_ϵ, E_w, E_ϵ, Z_{δ_e}, and M_{δ_e} depend mainly on Mach number M and altitude h. The state variable ϵ denotes the tail downwash angle. The downwash is experienced by a horizontal tail placed in the flow behind the wing. This vortex diminishes the incidence angle of the tail by an angle ϵ. The available measurements are pitch rate q and normal acceleration n_z:

$$\begin{bmatrix} q \\ n_z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \tilde{Z}_w & \tilde{Z}_q & \tilde{Z}_\epsilon \end{bmatrix} \begin{bmatrix} w \\ q \\ \epsilon \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{Z}_{\delta_e} \end{bmatrix} \delta_e \tag{40}$$

where

$$n_z = (\dot{w} - U_0 q - l_x \dot{q})/g \tag{41}$$

so $\tilde{Z}_w$, $\tilde{Z}_q$, $\tilde{Z}_\epsilon$, and $\tilde{Z}_{\delta_e}$ can be deduced from the other coefficients. Twenty flight conditions at different altitudes and Mach numbers were provided, as illustrated in Fig. 13. An LPV description is then deduced by interpolating the data:

$$\dot{x}_a = A_a(M, h) x_a + B_a(M, h) \delta_e \tag{42}$$

Algorithm 1 Single-parameter gain-scheduling algorithm

Step 0: initialization
Let Ω be a region of the complex plane; ν_Ω be a corresponding guardian map; and $A(r, \boldsymbol{K})$ be a closed-loop matrix, depending polynomially on the single parameter $r \in [r_{\min}, r_{\max}]$ and the gain vector $\boldsymbol{K} = [K_j] \in \mathbb{R}^p$. Obtain a controller $\boldsymbol{K}^0$ designed for the nominal case $r_0 = r_{\min}$, which ensures nominal stability relative to Ω. Set $n \leftarrow 0$. Using Lemma II.1 on $A(r, \boldsymbol{K}^0)$, find the largest stability interval $]\underline{r}_0, \bar{r}_0[$ containing r_0.
If $\bar{r}_0 > r_{\max}$, **then Stop, else** set the counter $n \leftarrow n + 1$.
Step n.1: synthesis phase
Find a new gain vector $\boldsymbol{K}^n$ inside a component defined by $\nu_\Omega(\bar{r}_{n-1}, \boldsymbol{K}) = 0$ using a search algorithm (see Appendix C) with an initial vector $\boldsymbol{K}^{n-1}$.
Step n.2: Robustness analysis
Using Lemma II.1 on $A(r, \boldsymbol{K}^n)$, find the largest stability interval $]\underline{r}_n, \bar{r}_n[$ containing $\bar{r}_{n-1}$.
If $\bar{r}_n > r_{\max}$, **then** go to **Final Step** or **Stop, else** set the counter $n \leftarrow n + 1$ and go to **Step n.1**.
Final step: interpolation
If an interpolation $\boldsymbol{K} = \boldsymbol{K}(r)$ is sought, use Lemma II.1 to check if stability is preserved $\forall\ r \in [r_{\min}, r_{\max}]$.

Algorithm 2 Two-parameter gain-scheduling algorithm

Step 0: Initialization
Let Ω be a region of the complex plane; ν_Ω be a corresponding guardian map; and $A(r_1, r_2, \boldsymbol{K})$ be a matrix depending polynomially on the two parameters $(r_1, r_2) \in [r_{1,\min}, r_{1,\max}] \times [r_{2,\min}, r_{2,\max}] \subset \mathbb{R}^2$ and the gain vector $\boldsymbol{K} = [K_j] \in \mathbb{R}^p$. Let $\boldsymbol{K}^0$, such that $A(r_{1,\min}, r_{2,\min})$ is Ω stable.
Set $n \leftarrow 0$, $r_{2,0} = r_{2,\min}$
Step n.1: synthesis of a scheduled controller with respect to r_1
Use Algorithm 3 to find a scheduled gain vector $\boldsymbol{K}^n(r_1)$ with fixed value $r_2 = r_{2,n}$. The scheduled controller $\boldsymbol{K}^n(r_1)$ ensures nominal stability for $(r_1, r_2) \in [r_{1,\min}, r_{1,\max}] \times \{r_{2,n}\}$.
Step n.2: robustness analysis of $\boldsymbol{K}^n(r_1)$ **along** r_2
Use Corollary II.2 to find the largest open interval $(\underline{r}_{2,n}, \bar{r}_{2,n})$ containing $r_{2,n}$, such that the scheduled gain vector $\boldsymbol{K}^n(r_1)$ stabilizes the system $\forall\ (r_1, r_2) \in [r_{1,\min}, r_{1,\max}] \times (\underline{r}_{2,n}, \bar{r}_{2,n})$.
If $\bar{r}_{2,n} > r_{2,\max}$, **then** go to **Final Step** or **Stop, else** set the counter $n \leftarrow n + 1$, $r_{2,n} = \bar{r}_{2,n-1}$ and go to **Step n.1**.
Final step: interpolation
If an interpolation $\boldsymbol{K} = \boldsymbol{K}(r_1, r_2)$ is sought, **Then** use Corollary II.2 to check if stability is preserved $\forall\ (r_1, r_2) \in [r_{1,\min}, r_{1,\max}] \times [r_{2,\min}, r_{2,\max}]$.

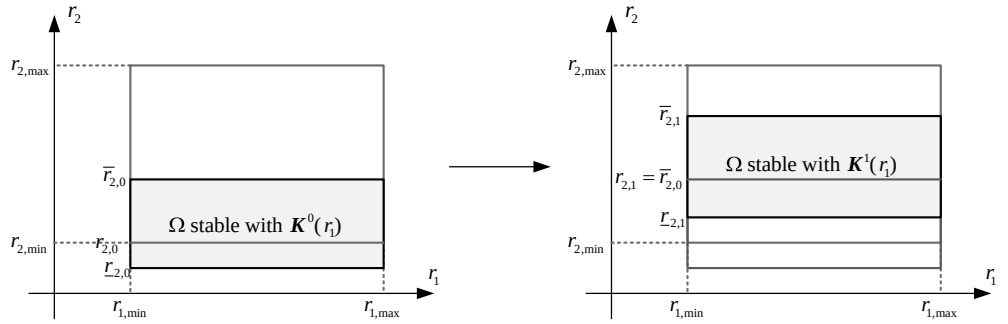


Fig. 12 Algorithm 2 illustration.

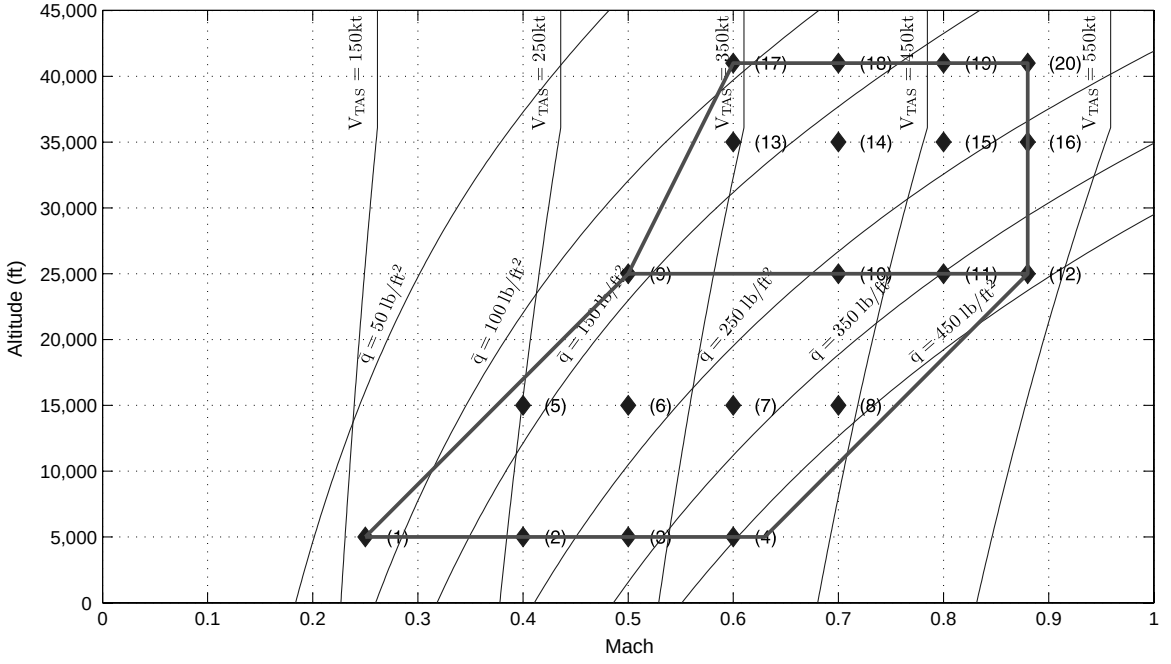


Fig. 13 Flight envelope.

$$y = C_a(M,h)x_a + D_a(M,h)\delta_e \tag{43}$$

with

$$x_a = [w \quad q \quad \epsilon]^T \qquad y = [q \quad n_z]^T$$

$$A_a(M,h) = \sum_{i=0}^{2}\sum_{j=0}^{2} A_{ij}M^i h^j \qquad B_a(M,h) = \sum_{i=0}^{2}\sum_{j=0}^{2} B_{ij}M^i h^j$$

$$C_a(M,h) = \sum_{i=0}^{2}\sum_{j=0}^{2} C_{ij}M^i h^j \qquad D_a(M,h) = \sum_{i=0}^{2}\sum_{j=0}^{2} D_{ij}M^i h^j \tag{44}$$

Even if more accurate interpolations can be found, the present one is sufficient for the problem at stake. To diminish interpolation errors, the flight envelope is split into two regions leading to two LPV descriptions: one for low to middle altitudes ($h \in [5,000, 25,000$ ft]) and another one for middle to high altitudes ($h \in [25,000, 41,000$ ft]). This improves the quality of interpolation on a more restricted zone. Special attention is given to the interpolation at the common border $h = 25,000$ ft in order to maintain continuity between the two LPV descriptions. Figure 14 shows the short-period mode poles at different altitudes. In general, natural frequency increases with Mach number and damping ratio diminishes with altitude.

Because of the presence of multiple delays in the actuator and sensor dynamics, both the actuator and IRU sensors (q and n_z) are modeled by high-order transfer functions (each 15th), as provided by Bombardier, Inc. (Fig. 15). Since the characteristics of both the actuator and sensors do not vary with Mach number and altitude, no interpolations with respect to M and h are needed. After balanced reduction of these elements and incorporation in the short-period model, a suitable eighth-order LPV description is finally obtained in the form of

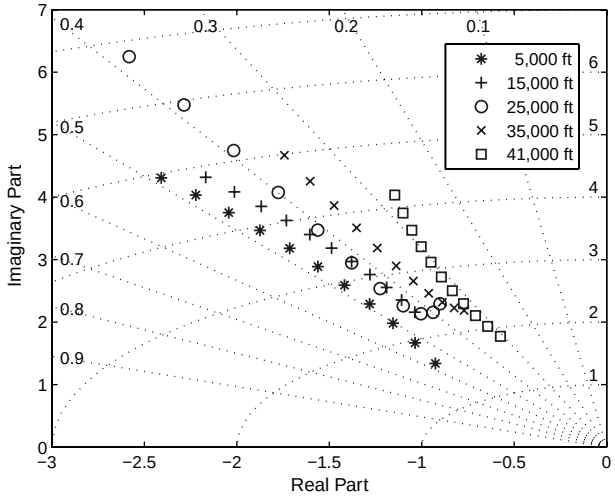


Fig. 14 Short-period mode evolution within flight envelope.

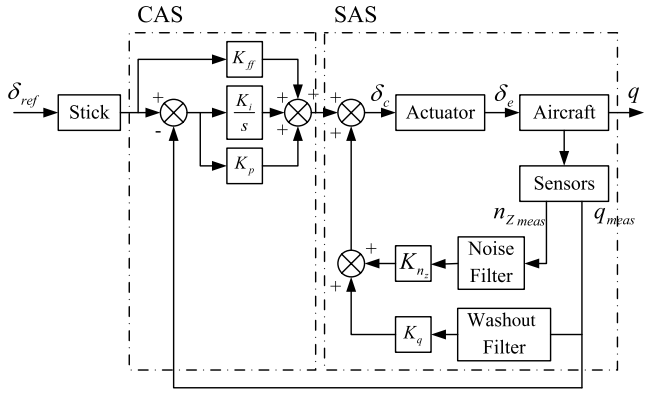


Fig. 15 Attitude hold control system.

$$\dot{x} = A(M,h)x + B\delta_c \tag{45}$$

$$q = C_q x \tag{46}$$

$$n_z = C_{n_z}(M,h)x \tag{47}$$

where x is the concatenation of the state vector x_a and the states of the actuator and sensor dynamic approximations, and δ_c is the actuator input. Note that the new input and output matrices B and C_q are both independent of M and h.

B. Controller Architecture

Figure 15 shows a classic longitudinal flight controller architecture designed to track pitch rate commands: mainly, the short-term dynamics. The architecture consists of a stability augmentation system (SAS) and a control augmentation system (CAS).

Q11 The variable parameters are the SAS gains K_q and K_{n_z}, the PI gains K_p and K_i, and the feedforward gain K_{ff}. The noise and washout filters are fixed first-order filters: a low-pass filter $10/(s+10)$ on the n_z measure and a washout filter $s/(s+3)$ on the q channel. By adding the two filter states and the integrator state to the open-loop state vector, the closed-loop state-space model is then

$$\begin{bmatrix} \dot{x} \\ \dot{x}_q \\ \dot{x}_{n_z} \\ \dot{x}_i \end{bmatrix} = \underbrace{\begin{bmatrix} A(M,h) + (K_q - K_p)BC_q & -K_q B & K_{n_z} B & K_i B \\ 3C_q & -3 & 0 & 0 \\ 10C_{n_z}(M,h) & 0 & -10 & 0 \\ -C_q & 0 & 0 & 0 \end{bmatrix}}_{A_{\text{cl}}} \begin{bmatrix} x \\ x_q \\ x_{n_z} \\ x_i \end{bmatrix} + \underbrace{\begin{bmatrix} (K_p + K_{\text{ff}})B \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{B_{\text{cl}}} \delta_{\text{ref}} \tag{48}$$

$$q = \underbrace{[C_q \quad 0 \quad 0 \quad 0]}_{C_{\text{cl}}} \begin{bmatrix} x \\ x_q \\ x_{n_z} \\ x_i \end{bmatrix} \tag{49}$$

where x_i denotes the integrator state, x_{n_z} denotes the n_z noise filter state, and x_q denotes the washout filter state. The closed-loop A_{cl} matrix depends on M, h, and the controller gains (except for the feedforward gain K_{ff}). The controller gain matrix is denoted by

Table 2 Handling quality boundaries

HQs	Level 1	Good Level 1
ζ_{sp}	$0.35 < \zeta_{\text{sp}} < 1.35$	$0.7 < \zeta_{\text{sp}} < 1.35$
ST	ST(2%) ≤ 3 s	ST(1%) ≤ 3 s
Drb	$-0.2 \leq Drb \leq 0.5$	$0.0 \leq Drb \leq 0.3$
M_G	>6 dB	
M_φ	>45°	

Q22

$$\boldsymbol{K} = [K_q \quad K_{n_z} \quad K_p \quad K_i \quad K_{\text{ff}}] \tag{50}$$

The problem consists, therefore, in tuning the gain matrix $\boldsymbol{K}$ in order to satisfy specific handling qualities.

C. Handling Qualities

The overall performance objective is to track pitch rate commands with predicted level 1 handling qualities and desired time-domain response behavior. The handling quality criteria considered in this paper are the short-period mode damping ratio ζ_{sp}, the settling time (ST), Gibson's dropback [37] Drb (see Appendix D), the gain margin M_G, and the phase margin M_φ of the CAS loop. The boundaries of these criteria are defined by military standards [35]. Even though handling qualities are primarily defined for military aircraft, they are usually applied to commercial aircraft with slight modifications derived from manufacturer experience. Table 2 summarizes the considered handling quality boundaries. More handling qualities may be taken into account, like frequency response-based criteria (i.e., pitch attitude bandwidth $\omega_{\text{BW}_\theta}$ and phase delay τ_p).

V. Controller Design

The scheduled controller will be designed in two steps in order to adequately illustrate the one- and two-parameter cases. First, a controller scheduled with respect to Mach number M at fixed altitude $h = 5000$ ft is designed. Then, the Mach scheduled controller will be extended to the full flight envelope with varying altitude.

A. Constraints

The time-domain objectives are simply expressed via pole constraints. More precisely, the issue is to maintain low-frequency closed-loop poles inside a target region Ω (Fig. 16): the damping condition is set to $\zeta > 0.7$ and the stability degree must be greater than 1 (i.e., $\alpha < -1$).

A corresponding guardian map ν_Ω, which depends on M, h, K_p, K_i, K_q, and K_{n_z} is then built by applying Eqs. (4) and (5):

$$\nu_\Omega(A_{\text{cl}}) = \nu_m(A_{\text{cl}})\nu_d(A_{\text{cl}}) \tag{51}$$

As the gain K_{ff} has no influence on the closed-loop pole location, it is dedicated to exactly tune the dropback value after tuning the other gains with the gain-scheduling algorithm. To have a null dropback (refer to Saussié et al. [40]), one has to take

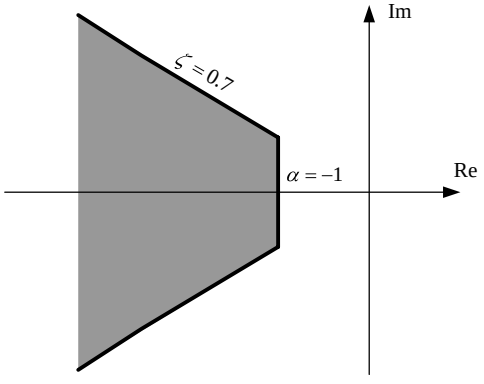


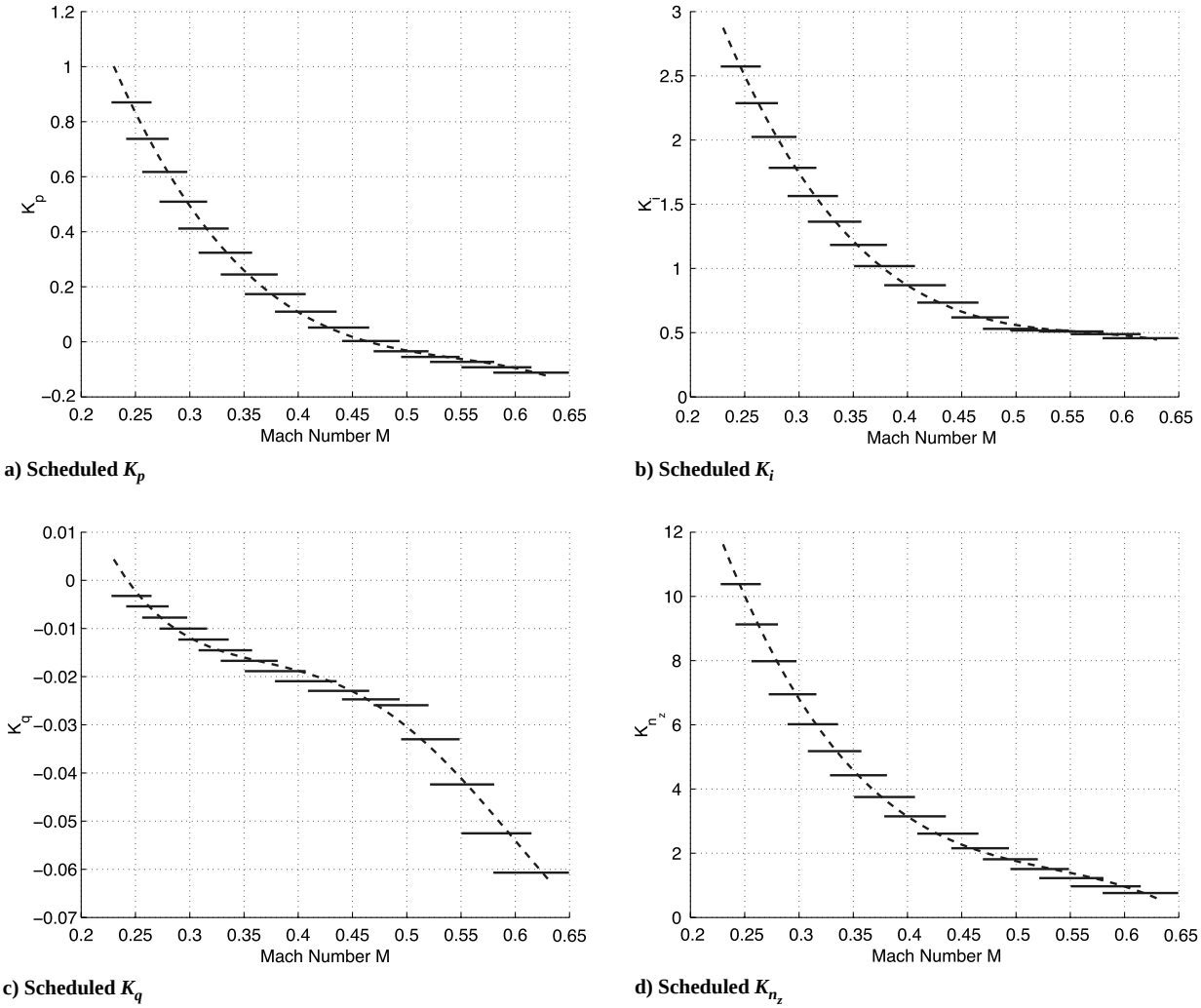
Fig. 16 Target zone Ω.

Algorithm 3 Algorithm to find a new vector gain inside an active component

0. **Initialization.**
Let Ω and $\boldsymbol{K}^0 \in \mathbb{R}^p$, such that $v_\Omega(\boldsymbol{K}^0) = 0$ and n are the maximum desired number of main loop iterations. Set the counter $m \leftarrow 0$.
1. **Main loop.**
While $m \leq n$
For i from one to p
a) Fix all the gains $\boldsymbol{K}$ to their current values except K_i.
b) The guardian map $v_\Omega(\boldsymbol{K})$ now only depends on K_i.
c) Using Lemma II.1, find the largest stability interval $]\underline{K}_i, \bar{K}_i[$ containing K_i^m.
d) Set $K_i^{m+1} \leftarrow (\underline{K}_i + \bar{K}_i)/2$.
End
A new gain vector $\boldsymbol{K}^{m+1}$ is then obtained.
a) If $\|\boldsymbol{K}^m - \boldsymbol{K}^{m+1}\| \leq \epsilon_K(1 + \|\boldsymbol{K}^m\|)$ (with ϵ_K arbitrary small positive value), then **Stop**.
b) Else, set $m \leftarrow m + 1$.
End

Table 3 Nominal eigenvalue assignment on $(M_0, h_0) = (0.25, 5000)$

Open-loop poles		Closed-loop poles
0		$-1.69 \pm 0.781i (\xi = 0.9, \omega = 1.86)$
$-0.926 \pm 1.34i (\xi = 0.569, \omega = 1.63)$		-2.44
-3	$\Rightarrow$	-3.85
-8.43		$-5.80 \pm 3.90i (\xi = 0.83, \omega = 6.99)$
-10		-8.52
-15.1		



a) Scheduled K_p

b) Scheduled K_i

c) Scheduled K_q

d) Scheduled K_{n_z}

Fig. 17 Scheduled gains with respect to M at $h_0 = 5000$ ft.

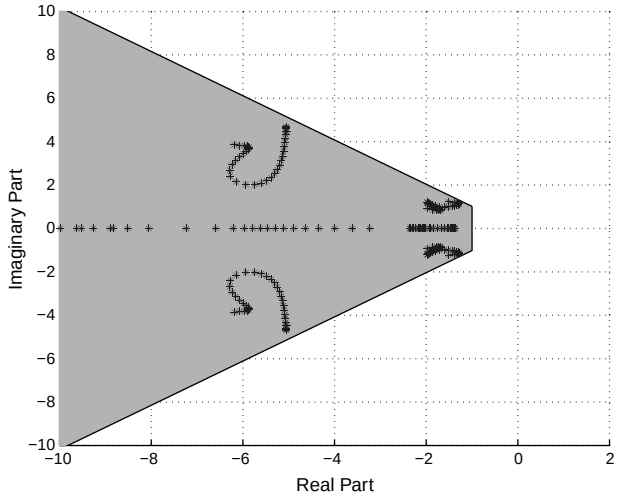


Fig. 18 Closed-loop pole confinement for $M \in [0.25 \quad 0.6]$ at $h_0 = 5000$ ft.

$$K_{\text{ff}} = -\frac{1 + K_{n_z} C_{n_z}(M,h) A(M,h)^{-1} B}{C_q A(M,h)^{-1} B} \tag{52}$$

B. Scheduled Controller with Respect to M at $h_0 = 5000$ feet

Algorithm 3 is used to find a scheduled controller with respect to M at $h_0 = 5000$ ft.

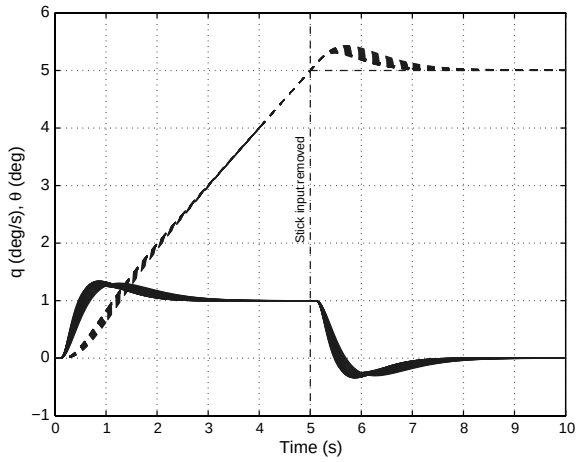


Fig. 19 Time responses for $M \in [0.25 \quad 0.6]$ at $h_0 = 5000$ ft.

1. Phase 0: Initial Static Controller

As K_{ff} has no influence on the pole assignment, a first static output feedback controller $\boldsymbol{K}^{(0)}$ is designed on the flight condition $(M_0, h_0) = (0.25, 5000)$ (the lowest left corner of the flight envelope) using the technique introduced in [40]. Table 3 shows the eigenvalue assignment. The gains are

$$\boldsymbol{K}^{(0)} = [-0.0048 \qquad 10.29 \qquad 0.865 \quad 2.552 \quad 0]$$

.

2. Phase 1: Pole Confinement

With the knowledge of a first stabilizing controller, Algorithm 3 finds successive controllers that fulfill the pole confinement requirements for $M \in [0.25, 0.6]$. The algorithm alternates between synthesis phases and robustness analysis phases:

1) For the synthesis phase, M is fixed and the guardian map $\nu_\Omega(A_{\text{cl}})$ [Eq. (51)] depends on the gains K_p, K_i, K_{n_z}, and K_q, where A_{cl} is of the form

$$A_0 + K_p A_1 + K_i A_2 + K_{n_z} A_3 + K_q A_4$$

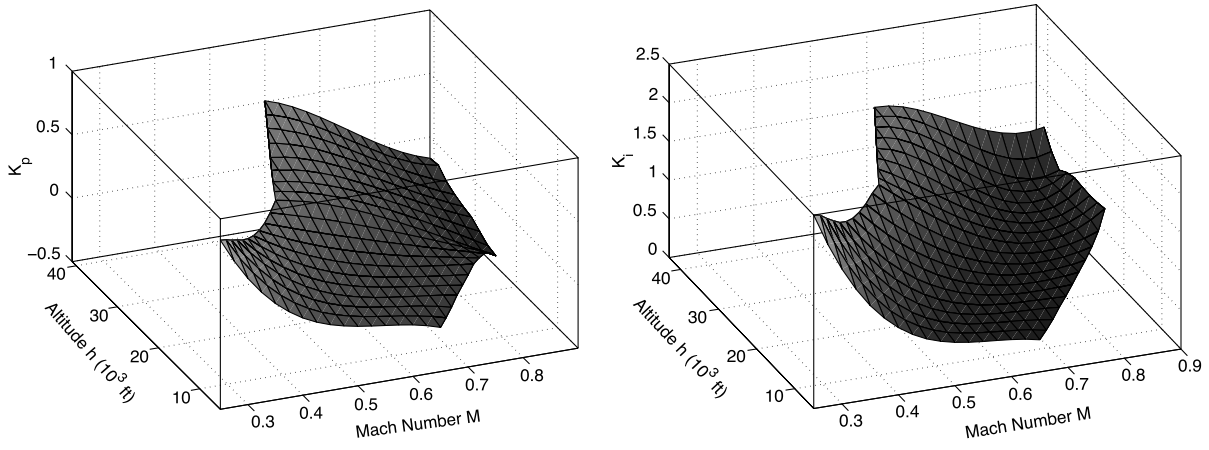
Starting from $\boldsymbol{K}^{(n-1)}$, which verifies $\nu_\Omega(A_{\text{cl}}) = 0$, a new gain vector $\boldsymbol{K}^{(n)}$ is found inside a stable component defined by $\nu_\Omega(A_{\text{cl}}) = 0$.

2) For the robustness analysis phase, the gain vector $\boldsymbol{K}^{(n)}$ is fixed and the guardian map $\nu_\Omega(A_{\text{cl}})$ [Eq. (51)] depends only on the Mach number M, where A_{cl} is of the form $A_0 + MA_1 + M^2 A_2$. The analysis results in the segment $(\underline{M}_n, \bar{M}_n)$ on which the controller $\boldsymbol{K}^{(n)}$ stabilizes the system.

Figures 17a–17d show the resulting successive gains found with Algorithm 3. These sequential gains ensure the stability condition on the entire Mach number domain [0.25, 0.6]. At this point, one can decide to switch between these gains, depending on the value of M, or continuous functions in M can be sought by interpolating data. The shape of the interpolating functions is free to the user (piecewise linear or polynomials). In the present case, third-order polynomial interpolations are obtained with the `polytool` MATLAB function (dashed lines). Figure 18 illustrates the closed-loop poles for $M \in [0.25, 0.6]$, showing that all pole confinement constraints are satisfied.

3. Phase 2: Dropback Adjustment and Time Responses

$K_{\text{ff}}(M)$ is calculated to ensure null dropback by using Eq. (52) with $h = h_0$ and $M \in [0.250.6]$. Figure 19 shows the time responses for a 5 s unit step and then a null input for different Mach number values at $h_0 = 5000$ ft. Settling times remain around 3 s, and dropback is zero for each value as predicted.



a) Scheduled K_p **b) Scheduled K_i**

Fig. 20 Scheduled gains K_p and K_i with respect to M and h.

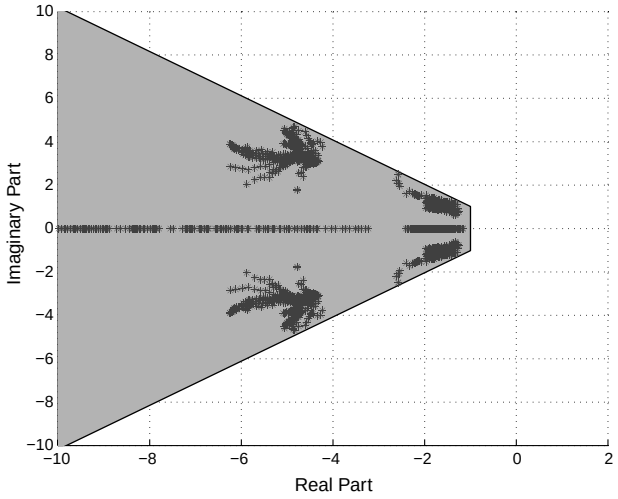


Fig. 21 Closed-loop pole confinement for full flight envelope.

C. Scheduled Controller, with respect to M and h

A first scheduled controller with respect to M has been found in the previous section for the lowest altitude $h = 5000$ ft. It now remains to extend it to the entire flight envelope.

1. Phase 1: Pole Confinement

Q12 Algorithm IIIB produces scheduled feedback gains with respect to M and h that ensure the pole confinement constraints on the entire flight envelope. As before, it alternates between synthesis phases at a fixed altitude and robustness analysis phases.

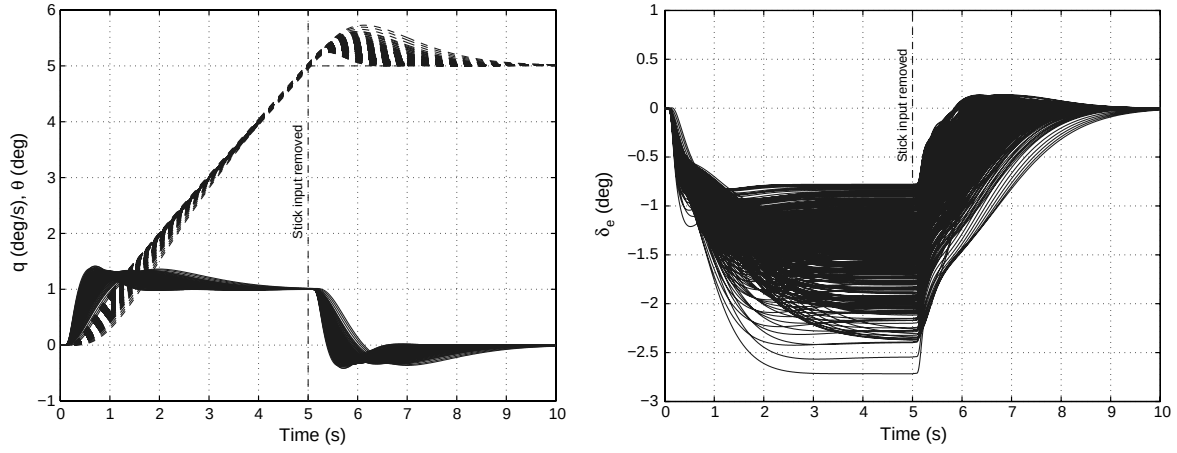
1) For the synthesis phase, altitude h_n is fixed and Algorithm 3 provides a scheduled controller $\boldsymbol{K}_n(M)$ with respect to M at altitude $h = h_n$.

2) For the robustness analysis phase, the scheduled controller $\boldsymbol{K}_n(M)$ is stable at h_n for $M \in [\underline{M}(h_n), \bar{M}(h_n)]$. Note that, as the flight domain is not rectangular, the Mach number minimum and maximum limits are functions of the altitude h. The analysis results in the segment $]\underline{h}_n, \bar{h}_n[$, such that $\boldsymbol{K}_n(M)$ stabilizes the system on the domain $[\underline{M}(h_n), \bar{M}(h_n)] \times (\underline{h}_n, \bar{h}_n)$, and the next synthesis is made at $h_{n+1} = \bar{h}_n$.

Figures 20a and 20b show the examples of resulting gain surfaces (K_p and K_i) found with Algorithm IIIB. These scheduled gains ensure the stability condition on the entire flight domain. No simple polynomial interpolations were actually sought but, instead, gains were implemented in lookup tables with multilinear interpolation. Such a technique is fairly standard, and dedicated algorithms are available in the MATLAB/Simulink environment. Figure 21 illustrates that all closed-loop poles are inside Ω_i.

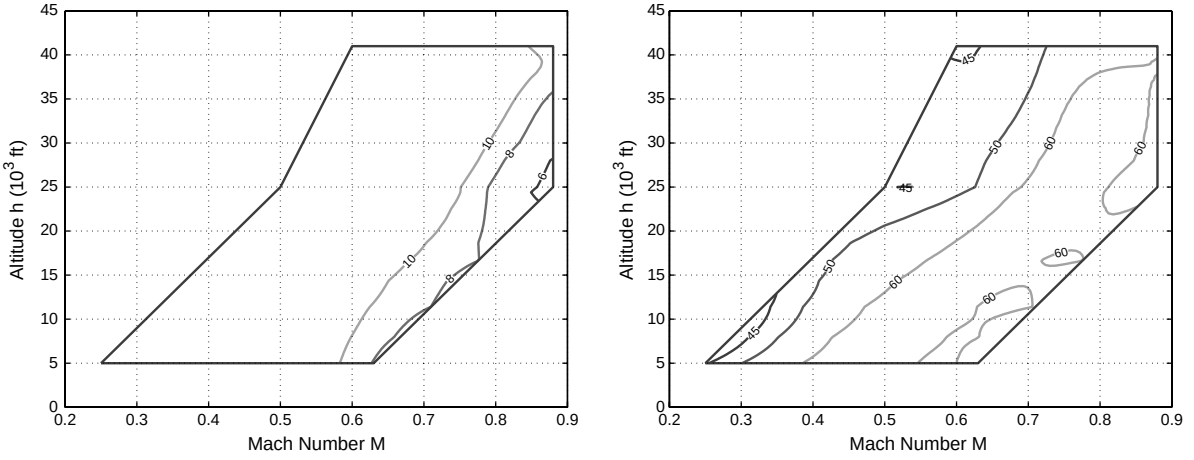
2. Phase 2: Dropback Adjustment, Time Responses, and Robustness Margins

Once again, Eq. (52) is used to obtain a scheduled gain $K_{\text{ff}}(M, h)$ that ensures null dropback on the entire flight envelope. For time simulations, a gridding of the entire flight envelope is considered. Figures 22a and 22b show time responses for a 5 s unit-step entry in δ_{ref}. They are satisfying for the entire flight envelope and, as



a) q(–), θ(– –) **b) Elevator deflection δ_e**

Fig. 22 Time responses for full flight envelope.



a) Gain margin **b) Phase margin**

Fig. 23 Gain and phase margins for full flight envelope.

predicted, Gibson's dropback [37] is almost null for every configuration. Figures 23a and 23b illustrate the gain and phase margin levels on the complete flight envelope when the CAS loop is opened. The gain margin requirement (greater than 6 dB) is largely satisfied for most part of flight envelope. The same applies for the phase margin requirement (greater than 45°), where worst cases have a phase margin above 42°. Thus, even if gain and phase margin requirements were not part of the synthesis, they are mostly fulfilled. Finally, Table A1 shows the handling qualities for the 20 flight conditions originally provided by Bombardier, Inc. (Fig. 13).

VI. Conclusions

This paper presents a novel technique for scheduling controllers with application to the longitudinal flight of a business aircraft. It applies to systems with one or two parameters. From a single controller designed for one trim condition, the algorithm automatically varies the controller gains to extend its stability property (i.e., essentially pole confinement constraints) to the rest of the parametric domain. Contrary to classic gain-scheduling techniques, there is no need to multiple controller synthesis on several trim conditions, as the algorithms deal automatically with that. Moreover, stability is guaranteed all along the process. Last but not the least, the technique can handle any fixed controller architecture. The technique was successfully applied on the control of the longitudinal flight of a Bombardier, Inc., *Challenger 604*. Based on a classic control scheme, the flight scheduled controller with respect to altitude and Mach number operates well on the full flight envelope. The pole confinement constraint is satisfied through the entire flight envelope, showing the efficiency of the method. Additional criterion (i.e., Gibson's dropback) was handled by a previously stated theoretical formula. Phase and gain margins were checked a posteriori and proved to be adequate. In spite of the fixed architecture constraint, the current procedure manages to produce an efficient scheduled controller through the entire flight envelope with less computational effort than classic techniques.

Appendix A: Bilaternate Product

Let A and B be $n \times n$ matrices. To introduce the bialternate product of A and B, we first establish some notation. Let V^n be the $\frac{1}{2}n(n-1)$-tuple consisting of pairs of integers (p, q), $p = 2, 3, \ldots, n$, $q = 1, \ldots, p-1$, listed lexicographically. That is,

$$V^n = [(2,1), (3,1), (3,2), (4,1), (4,2), (4,3), \ldots, (n, n-1)]$$

Denote by V_i^n the ith entry of V^n. Denote

$$f[(p,q);(r,s)] = \frac{1}{2}\left(\det\begin{bmatrix} a_{pr} & a_{ps} \\ b_{qr} & b_{qs} \end{bmatrix} + \det\begin{bmatrix} b_{pr} & b_{ps} \\ a_{qr} & a_{qs} \end{bmatrix}\right) \quad \text{(A1)}$$

where the dependence of f on A and B is kept implicit for simplicity. The bialternate product of A and B, denoted by $A \odot B$, is $\frac{1}{2}n(n-1)$-dimensional square matrix, for which the ijth entry is given by $(A \odot B)_{ij} = f(V_i^n, V_j^n)$. For example, the bialternate product of a 3×3 matrix $A = (a_{ij})$ with the identity matrix is a 3×3 matrix given by

$$A \odot I = \begin{bmatrix} a_{11}+a_{22} & a_{23} & -a_{13} \\ a_{32} & a_{11}+a_{33} & a_{12} \\ -a_{31} & a_{21} & a_{22}+a_{33} \end{bmatrix} \quad \text{(A2)}$$

The bialternate product has some interesting spectral properties. In particular, if $\sigma(A) = \lambda_1, \ldots, \lambda_n$, then

$$\sigma[2(A \odot I)] = \{\lambda_1+\lambda_2, \lambda_1+\lambda_3, \ldots, \lambda_1+\lambda_n, \lambda_2 \quad \text{(A3)}$$

$$+\lambda_3, \ldots, \lambda_2+\lambda_n, \ldots, \lambda_{n-1}+\lambda_n\} \quad \text{(A4)}$$

Thus, $2(A \odot I)$ becomes singular if, and only if, A has opposite eigenvalues.

Table A1 Handling quality results for the 20 original flight conditions

Case	ζ_{sp}	Drb	ST(2%)	M_G (dB)	M_φ(°)
1	0.90	0	3.32	11.31	44.46
2	0.86	0	2.35	15.19	61.72
3	0.85	0	2.50	12.29	63.80
4	0.77	0	2.40	9.33	59.96
5	0.88	0	3.18	12.49	48.84
6	0.86	0	2.66	14.41	57.35
7	0.85	0	2.44	11.78	63.37
8	0.72	0	1.34	8.99	60.46
9	0.90	0	3.53	12.09	45.28
10	0.84	0	2.30	13.55	61.64
11	0.81	0	2.31	7.56	61.08
12	0.83	0	1.75	5.69	59.07
13	0.88	0	3.82	12.21	45.52
14	0.81	0	2.96	13.47	50.54
15	0.85	0	2.42	11.63	59.07
16	0.79	0	2.52	7.64	57.98
17	0.88	0	3.41	12.79	44.31
18	0.86	0	3.34	12.76	48.45
19	0.81	0	2.83	12.12	52.12
20	0.83	0	2.49	8.77	54.55

Appendix B: Bezoutian

Given any polynomial $a(s) = a_n s^n + \ldots + a_1 s + a_0$, $a_n \neq 0$, define the polynomial

$$\hat{a}(s) := s^n a(s^{-1}) = a_0 s^n + \ldots + a_{n-1} s + a_n$$

and the matrix

$$S(a) := \begin{bmatrix} a_1 & a_2 & \ldots & \ldots & a_n \\ a_2 & \vdots & \ldots & a_n & 0 \\ \vdots & a_n & \ldots & \vdots & \vdots \\ a_n & 0 & \ldots & \ldots & 0 \end{bmatrix} \quad \text{(B1)}$$

The Bezoutian $B(a, b)$ of two polynomials a and b may then be expressed as the $n \times n$ matrix, n being the largest of the degrees of a and b, given by

$$B(a,b) := S(a)S(\hat{b})P - S(b)S(\hat{a})P \quad \text{(B2)}$$

with P as a certain permutation matrix [41]. Our interest in the Bezoutian stems from the following result.

Lemma B.1: The polynomials $a(s)$ and $b(s)$ have no common zeros if, and only, if the associated Bezoutian $B(a, b)$ [or, equivalently, the matrix $S(a)S(\hat{b}) - S(b)S(\hat{a})$] is nonsingular.

Appendix C: Search Algorithm

Let Ω and $\boldsymbol{K}^0$ denote an initial gain vector, such that $\nu_\Omega(\boldsymbol{K}^0) = 0$. The goal of the search algorithm presented in [40] is to pick a new gain vector $\boldsymbol{K}$ inside a stable component $\mathcal{C}_1$ (defined by $\nu_\Omega(\boldsymbol{K}) = 0$), knowing an initial gain vector $\boldsymbol{K}_0$ lying on the border of $\mathcal{C}_1$. According to Corollary II.1, any choice of gains $\boldsymbol{K}$ strictly inside $\mathcal{C}_1$ ensures that all the eigenvalues of $A_{cl}(\boldsymbol{K})$ are strictly inside Ω. The main and intuitive idea is that, by picking a new vector $\boldsymbol{K}$ well inside the active component $\mathcal{C}_1$, hence sufficiently far from its border, the resulting closed-loop poles will also move well inside Ω, hence far from its border.

Example C.1: The same system and constraints as in Example II.2 are considered with $(k_1^0, k_2^0) \in [0, 10] \times [0, 10]$. Figure D1 illustrates the algorithm behavior with the starting point $(k_1, k_2) = (3, 2.5)$.

Remark C.1: The main search directions can possibly be enriched with other search directions as in direct search methods [42,43]. Another alternative would be to consider the local gradient as the unique search direction and to use Lemma II.1.

Appendix D: Dropback

Besides the classical time-domain criteria, such as settling time, overshoot, or rising time, Gibson's dropback [37] is commonly used by flight control engineers. It is a short-term measure of the pitch attitude changes, and it is calculated based on the reduced-order attitude θ response (i.e., without the phugoid mode) to a stick step input removed after a few seconds. Figure D2 illustrates how to calculate the dropback Drb. The quantity q_{ss} is the pitch rate steady-state value. Ideally, having a zero dropback value means piloting a pure integrator in θ after a short time. As it is preferred to have positive dropback values instead of negative ones, this comes with some significant overshoots in the pitch rate q response.

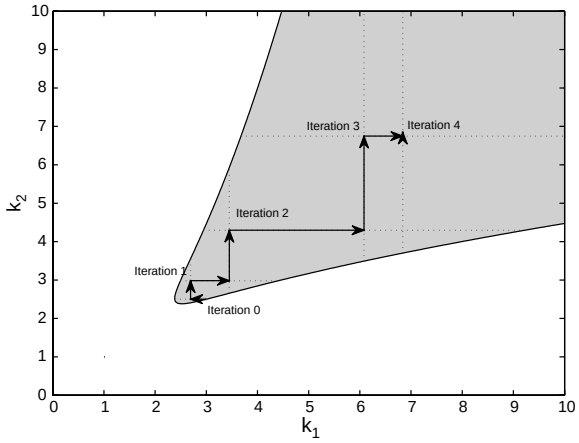


Fig. D1 Search algorithm illustration.

References

[1] Rugh, W. J., and Shamma, J. S., "Research on Gain-Scheduling," *Automatica*, Vol. 36, No. 10, Oct. 2000, pp. 1401–1425. doi:10.1016/S0005-1098(00)00058-3

[2] Leith, D. J., and Leithead, W. E., "Survey of Gain-Scheduling Analysis and Design," *International Journal of Control*, Vol. 73, No. 11, 2000, pp. 1001–1025. doi:10.1080/002071700411304

[3] Fromion, V., and Scorletti, G., "A Theoretical Framework for Gain Scheduling," *International Journal of Robust and Nonlinear Control*, Vol. 13, No. 10, 2003, pp. 951–982. doi:10.1002/rnc.748

[4] Rugh, W. J., "Analytical Framework for Gain Scheduling," *IEEE Control Systems Magazine*, Vol. 11, No. 1, Jan. 1991, pp. 79–84. doi:10.1109/37.103361

[5] Lawrence, D. A., and Rugh, W. J., "Gain Scheduling Dynamic Linear Controllers for a Nonlinear Plant," *Automatica*, Vol. 31, No. 3, March 1995, pp. 381–390. doi:10.1016/0005-1098(94)00113-W

[6] Shamma, J. S., and Athans, M., "Analysis of Gain-Scheduled Control for Nonlinear Plants," *IEEE Transactions on Automatic Control*, Vol. 35, No. 8, 1990, pp. 898–907. doi:10.1109/9.58498

[7] Stilwell, D. J., and Rugh, W. J., "Stability Preserving Interpolation Methods for the Synthesis Of Gain-Scheduled Controllers," *Automatica*, Vol. 33, No. 7, 2000, pp. 1263–1271. Q17 Q18

[8] Clément, B., and Duc, G., "An Interpolation Method for Gain-Scheduling," *40th IEEE Conference on Decision and Control*, Vol. 2, IEEE Publ., Piscataway, NJ, 2001, pp. 1310–1315.

[9] Leith, D. J., Leithead, W. E., Lawrence, D. A., and Rugh, W. J., "Comments on Gain Scheduling Dynamic Linear Controllers for a Non Linear Plant," *Automatica*, Vol. 34, No. 8, 1998, pp. 1041–1043. doi:10.1016/S0005-1098(98)00038-7

[10] Mehendale, C. S., and Grigoriadis, K. M., "A New Approach to LPV Gain-Scheduling Design and Implementation," *43rd IEEE Conference on Decision and Control*, Vol. 3, Paradise Island, Bahamas, IEEE Publ., Piscataway, NJ, Dec. 2004, pp. 2942–2947.

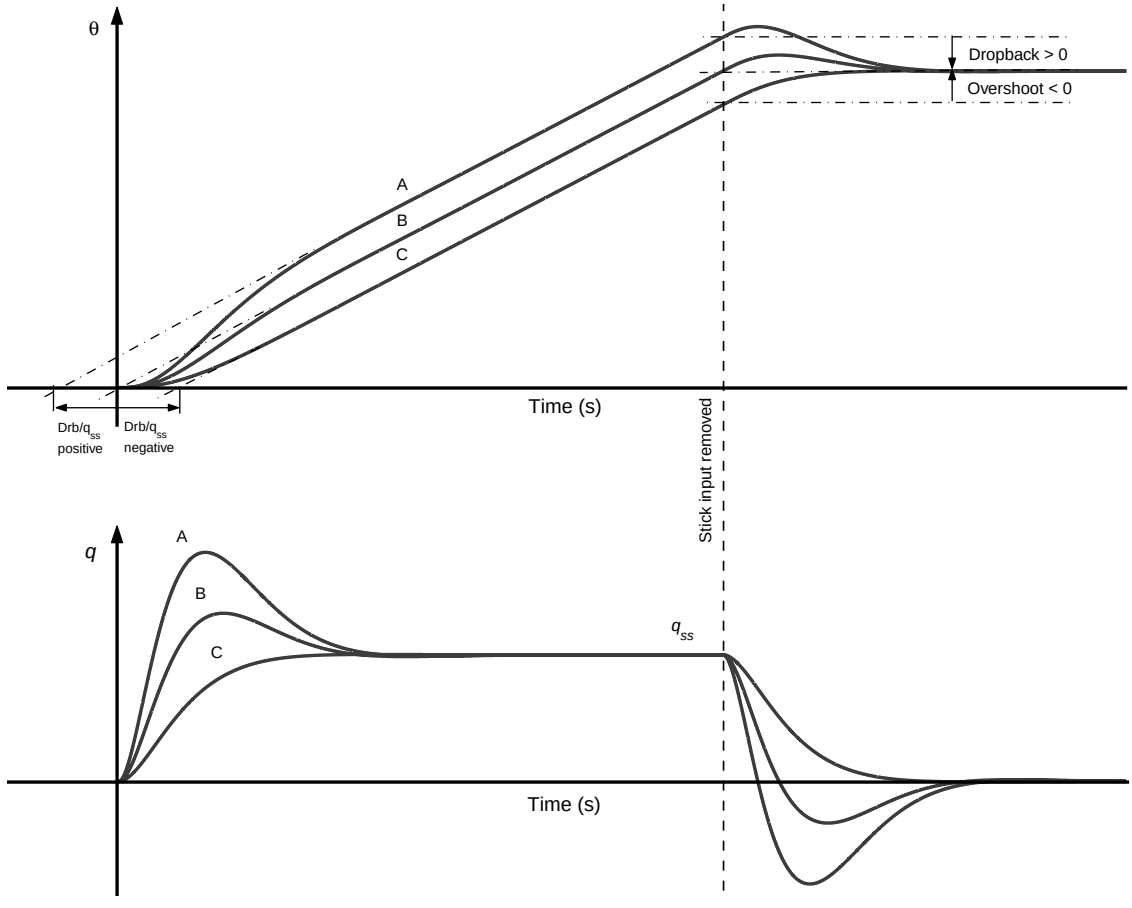


Fig. D2 Gibson's dropback definition [37].

[11] Packard, A., "Gain Scheduling via Linear Fractional Transformations," *Systems and Control Letters*, Vol. 22, No. 2, 1994, pp. 79–92. doi:10.1016/0167-6911(94)90102-3

[12] Apkarian, P., and Gahinet, P., "A Convex Characterization of Gain Scheduling H_∞ Controllers," *IEEE Transactions on Automatic Control*, Vol. 40, No. 5, 1995, pp. 853–864. doi:10.1109/9.384219

[13] Boyd, S., Ghaoui, L. E., Feron, E., and Balakrishan, V., "Linear Matrix Inequalities in System and Control Theory," *SIAM Studies in Applied Mathematics*, Vol. 15, Soc. for Industrial and Applied Mathematics, Philadelphia, 1994, pp. 99–118.

Q19 [14] Gahinet, P., Nemirovski, A. J. L., and Chilali, M., *LMI Control Toolbox*, MathWorks, Natick, MA, 1994.

[15] Prempain, E., and Postlethwaite, I., "$\mathcal{L}_2$ and $\mathcal{H}_2$ Performance Analysis and Gain-Scheduling Synthesis for Parameter-Dependent Systems," *Automatica*, Vol. 44, No. 8, Aug. 2008, pp. 2081–2089. doi:10.1016/j.automatica.2007.12.008

Q20 [16] Saydy, L., Tits, A., and Abed, E., "Guardian Maps and the Generalized Stability of Parametrized Families of Matrices and Polynomials," *Mathematics of Control, Signals, and Systems*, Vol. 3, No. 4, 1990, pp. 345–371. doi:10.1007/BF02551375

[17] Saussié, D., Saydy, L., and Akhrif, O., "Gain Scheduling Control Design for a Pitch-Axis Missile Autopilot," AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, AIAA Paper 2008-7000, 2008.

[18] Reichert, R. T., "Dynamic Scheduling of Modern-Robust-Control Autopilot Designs for Missiles," *IEEE Control Systems Magazine*, Vol. 12, No. 5, 1992, pp. 35–42. doi:10.1109/37.158896

[19] Shamma, J. S., and Cloutier, J. R., "A Linear Parameter-Varying Approach to Gain Scheduled Missile Autopilot Design," *Proceedings of the American Control Conference*, Chicago, IL, Vol. 2, IEEE Publ., Piscataway, NJ, 1992, pp. 1317–1321.

[20] Shamma, J. S., and Cloutier, J. R., "Gain-Scheduled Missile Autopilot Design Using Linear Parameter Varying Transformations," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, 1993, pp. 256–263. doi:10.2514/3.20997

[21] Nichols, R. A., Reichert, R. T., and Rugh, W. J., "Gain Scheduling for H_∞ Controllers: A Flight Control Example," *IEEE Transactions on Control Systems Technology*, Vol. 1, No. 2, 1993, pp. 69–79. doi:10.1109/87.238400

[22] White, D. P., Wozniak, J. G., and Lawrence, D. A., "Missile Autopilot Design Using a Gain Scheduling Technique," *Proceedings of the Annual Southeastern Symposium on System Theory*, Athens, OH, IEEE Publ., Piscataway, NJ, March 1994, pp. 606–610.

[23] Wu, F., Packard, A., and Balas, G. F., "LPV Control Design for Pitch-Axis Missile Autopilots," *Proceedings of the 34th Conference on Decision and Control*, New Orleans, LA, Vol. 1, IEEE Publ., Piscataway, NJ, Dec. 1995, pp. 188–193.

[24] Schumacher, C., and Khargonekar, P. P., "Missile Autopilot Designs Using H_∞ Control with Gain Scheduling and Dynamic Inversion," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 234–243. doi:10.2514/2.4248

[25] Biannic, J. M., and Apkarian, P., "Missile Autopilot via a Modified LPV Synthesis Technique," *Aerospace Science and Technologies*, Vol. 3, No. 3, 1999, pp. 153–160. doi:10.1016/S1270-9638(99)80039-X

[26] Tan, W., Packard, A., and Balas, G., "Quasi-LPV Modeling and LPV Control of a Generic Missile,"*Proceedings of the American Control Conference*, Chicago, IL, Vol. 5, IEEE Publ., Piscataway, NJ, June 2000, pp. 3692–3696.

[27] Döll, C., Le Gorrec, Y., Ferreres, G., and Magni, J. F., "A Robust Self-Scheduled Missile Autopilot: Design by Multi-Model Eigenstructure Assignment," *Control Engineering Practice*, Vol. 9, No. 10, Oct. 2001, pp. 1067–1078. doi:10.1016/S0967-0661(01)00083-1

[28] Wu, F., Packard, A., and Balas, G. J., "Systematic Gain-Scheduling Control Design: A Missile Autopilot Example," *Asian Journal of Control*, Vol. 4, No. 3, Sept. 2002, pp. 341–347.doi:10.1111/j.1934-6093.2002.tb00362.x

[29] Bruyere, L., Tsourdos, A., and White, B. A., "Polynomial Approach for Design and Robust Analysis of Lateral Missile Control," *International Journal of Systems Science*, Vol. 37, No. 8, June 2006, pp. 585– 597. doi:10.1080/00207720600784742

[30] Stevens, B. L., and Lewis, F. L., *Aircraft Control and Simulation*, 2nd ed., Wiley, Hoboken, NJ 2003, pp. 254–381.

[31] Saydy, L., Akhrif, O., and Zhu, G., "Handling Quality Characterization of Flight System Controller Gains," *ICECS 2000: 7th IEEE International Conference on Electronics, Circuits and Systems*, Vol. 2, IEEE Publ., Piscataway, NJ, 2000, pp. 721–724.

[32] Saussié, D., Saydy, L., and Akhrif, O., "Flight Control Design with Robustness and Handling Qualities Requirements," *CCECE 2003: Canadian Conference on Electrical and Computer Engineering.*, Vol. 3, IEEE Publ., Piscataway, NJ, 2003, pp. 1749–1752.

[33] Saussié, D., Saydy, L., and Akhrif, O., "Longitudinal Flight Control Design with Handling Quality Requirements," *The Aeronautical Journal*, Vol. 110, No. 1111, 2006, pp. 627–637.

[34] Mitchell, D. G., Doman, D. B., Key, D. L., Klyde, D. H., Leggett, D. B., Moorhouse, D. J., Mason, D. H., Raney, D. L., and Schmidt, D. K., "Evolution, Revolution, and Challenges of Handling Qualities," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 1, 2004, pp. 12–28. doi:10.2514/1.3252

[35] "Flying Qualities of Piloted Aircraft," U.S. Dept. of Defense, MIL-HDBK-1797A, Washington, D.C., 1997.

[36] Hodgkinson, J., *Aircraft Handling Qualities*, AIAA, Reston, VA, 1999, pp. 35–86.

[37] Gibson, J. C., *Development of a Methodology for Excellence in Handling Qualities Design for Fly by Wire Aircraft*, Delft Univ. Press, Delft, The Netherlands, 1999, pp. 93–137.

[38] Stephanos, C., "Sur une Extension du Calcul des Substitutions Linéaires," *Journal de Mathématiques Pures et Appliquées*, Vol. 5, No. 6, 1900, pp. 73–128.

[39] Saussié, D., Bérard, C., Akhrif, O., and Saydy, L., "Longitudinal Flight Control Synthesis with Guardian Maps," AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA Paper 2009-5988, Chicago, 2009.

[40] Saussié, D., Saydy, L., and Akhrif, O., "Pitch Rate Control with Guardian Map Based Algorithm," *18th Mediterranean Conference on Control and Automation*, Marrakech, Morocco, 23–25 June 2010, pp. 1473–1478. Q21

[41] Lancaster, P., and Tismenetsky, M., *The Theory of Matrices*, Academic Press, New York, 1985, pp. 454–460.

[42] Higham, J. N., "Optimization by Direct Search in Matrix Computations," *SIAM Journal on Matrix Analysis and Applications*, Vol. 14, No. 2, April 1993, pp. 317–333. doi:10.1137/0614023

[43] Kolda, T., Lewis, R., and Torczon, V., "Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods," *SIAM Review*, Vol. 45, No. 3, 2003, pp. 385–482. doi:10.1137/S003614450242889

Queries

IMPORTANT: PLEASE READ CAREFULLY.

When production of AIAA journal papers begins, the official approved PDF is considered the authoritative manuscript. Authors are asked to submit source files that match the PDF exactly, to ensure that the final published article is the version that was reviewed and accepted by the associate editor. Once a paper has been accepted, any substantial corrections or changes must be approved by the associate editor before they can be incorporated.

If you and the EIC settled on some final changes to your manuscript after it was accepted, it is possible that your page proofs do not reflect these final changes. If that is the case, please submit these changes as itemized corrections to the proofs.

If final changes were made to the figures, please check the figures appearing in the proofs carefully. While it is usual procedure to use the figures that exist in the source file, if discrepancies are found between figures (manuscript source file vs the approved PDF), the figures from the PDF are inserted in the page proofs, again deferring to the PDF as the authoritative manuscript. If you find that agreed-upon final changes to your figures are not appearing in your page proofs, please let us know immediately.

Q1. Only one affiliation can appear under each author's name. Please choose among these approaches for David Saussié: A) It is customary to list the affiliation at the time of writing at the top, and a subsequent affiliation can be listed in the footnote. For this approach, please indicate which affiliation was at the time of writing and which is current. B) If preferred, only the current affiliation can be listed at the top. For this approach, please indicate the former affiliation and it will be removed. C) If the author currently has a double affiliation, please indicate which one is the primary and should appear at the top.
Q2. Please check that the copyright (©) type is correct.
Q3. Minor syntax adjustments were made throughout; please read closely to confirm that your meaning was retained.
Q4. Numbered items were edited to be complete sentences, per journal guidelines. Please check that your meanings were retained.
Q5. AIAA has not indicated color use for your paper. Please confirm that your paper should be grayscale and that all figures are satisfactory. If any replacement figures are needed, please send them in .eps or .tiff format to aiaa-proofs@beacon.com and use code 52178 in the subject line. Formats of .jpg, .doc, or .pdf can be used with some loss of quality.
Q6. Does "Theorem" refer to Theorem II.1? If not, please clarify.
Q7. Please supply a definition for LTI.
Q8. "until possibly covering" is unclear. Please review and edit as necessary.
Q9. Please supply a definition for PD.
Q10. Please supply a definition for IRU.
Q11. Please supply a definition for PI.
Q12. Please clarify "Algorithm IIIB", as there is no Algorithm IIIB included in this paper.
Q13. "for most part of flight envelope" is unclear. Please review and edit as necessary.
Q14. The Conclusions section was edited to be in the third person, per journal requirements. Please check that your meaning was retained.
Q15. The sentence beginning, "Denote by V_i^n..." is unclear. Please review and edit as necessary.
Q16. Are Eqs. A3 and A4 parts of the same equation? If so, we will renumber them as A3a and A3b. Please advise.
Q17. For Ref. [7], our online database shows that the paper of this name was published in *Automatica* in Vol. 36, Issue 5, on pages 665–671 (see doi:10.1016/S0005-1098(99)00193-4). Please review and edit as necessary.
Q18. This query was generated by an automatic reference checking system. DOIs for Refs. [7, 33, 38] could not be located in the databases used by the system. While the references may be correct, we ask that you check them so we can provide as many links to the referenced articles as possible.
Q19. Please provide the pages used for Ref. [14].
Q20. The issue has been added for Ref. [16]. Please confirm this is correct.
Q21. Please provide the name and location of the publisher (NOT the conference host) for Ref. [40].
Q22. Please supply a definition for HQ.