Titre: Title:	Longitudinal Flight Control Methodologies for Commercial Aircraft with Handling Quality Requirements
Auteur: Author:	Cédric Dionne
Date:	2022
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Dionne, C. (2022). Longitudinal Flight Control Methodologies for Commercial Aircraft with Handling Quality Requirements [Master's thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/10568/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Longitudinal flight control methodologies for commercial aircraft with handling quality requirements

CÉDRIC DIONNE
Département de génie électrique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie aérospatial

Août 2022

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Longitudinal flight control methodologies for commercial aircraft with handling quality requirements

présenté par **Cédric DIONNE**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Lahcen SAYDY, président
David SAUSSIÉ, membre et directeur de recherche
Richard GOURDEAU, membre

DEDICATION

À mes parents biologistes,
qui m'ont toujours encouragé à faire ce qui me passionne,
même si les avions ne font pas partie du règne animal...

ACKNOWLEDGEMENTS

Merci David Saussié pour toutes les occasions que tu m'as offertes. Je me compte chanceux d'avoir pu faire ma maîtrise dans le sujet qui me passionne. De plus, l'expérience de chargé de laboratoire a elle aussi été très enrichissante. Je te remercie aussi pour ton suivi et tes conseils tout au long de ces deux dernières années.

Merci Bastien Fulchiron pour ta patience et ton support pour la majorité du projet. Malgré toutes les tâches que tu avais à gérer, ton intérêt pour ma recherche et les efforts que tu as fait pour suivre mon travail sont grandement appréciés. Je te suis aussi extrêmement reconnaissant pour le covoiturage jusqu'à Mirabel, surtout en temps de pandémie.

Thank you flight characteristics team from Airbus Canada for your warm welcome despite the (then) raging pandemic. The wisdom and tips from experienced designers were extremely helpful to my research. Furthermore, our Daily Morning Meetings made the lock-out much less lonely.

Thank you Mitacs Acceleration program from Mitacs Canada and NSERC (CRSNG) for financing this project. With the pandemic looming over the aeronautical industry, the collaboration with Airbus Canada would most likely have been impossible without your help.

RÉSUMÉ

L'introduction des technologies *Fly-By-Wire* dans l'aviation commerciale a permis aux avionneurs d'améliorer le comportement de leurs aéronefs par l'ajout de lois de pilotage. Néanmoins, le réglage de telles lois est difficile étant donné la grande variation du comportement d'un avion au sein de son enveloppe de vol. La solution industrielle à ce problème a été de séquencer les gains des correcteurs en fonction de paramètres facilement mesurables, telle la pression dynamique. Bien que ce processus soit simple et éprouvé, son application demande un temps important et occasionne donc des coûts significatifs.

Dans ce mémoire, deux approches sont considérées pour pallier aux problèmes associés au séquencement de gains. Premièrement, la synthèse H_∞ structurée est utilisée pour réduire le temps nécessaire à l'obtention d'un correcteur séquencé. Cette méthode d'optimisation robuste synthétise un correcteur rencontrant les contraintes prescrites avec une architecture prédéfinie. La méthodologie conçue permet de traduire des spécifications de performance ou de stabilité en contraintes compatibles avec la synthèse H_∞ en présence d'incertitudes importantes, par exemple, l'absence de mesure de masse et de centrage.

La deuxième approche vise à définir des lois de pilotage *par équations* (ou G*), permettant d'éviter le séquencement ainsi que de limiter les effets d'une modification du système sur le réglage des gains. Les dynamiques principales de l'avion sont placées grâce à des approximations d'ordre réduit et d'un modèle de l'avion simplifié, de sorte à obtenir un comportement satisfaisant. Cette modélisation simplifiée de l'avion, approximant l'aéronef adéquatement dans la majorité de son enveloppe, est obtenue depuis des hypothèses simplificatrices (ex. comportement linéaire). Puisque les avions modernes comportent plus de non linéarités qu'auparavant, étant donné leurs profils aérodynamiques hautement optimisés pour la croisière, ces hypothèses ne sont pas valides dans l'intégralité de l'enveloppe de vol. L'ajout de lois non linéaires permettra de s'assurer que l'avion se conforme au modèle simplifié, garantissant une bonne performance non linéaire.

Finalement, les deux approches sont appliquées et comparées sur un modèle avion fourni par Airbus Canada. Cette validation permet de s'assurer de la conformité des lois conçues aux requis de designs utilisés, principalement issus de la littérature.

ABSTRACT

The introduction of Fly-By-Wire technologies in commercial aviation has allowed manufacturers to augment modern aircraft with flight control laws, improving aircraft handling and pilot satisfaction. Designing such control laws is challenging due to the large changes aircraft dynamics undergo throughout the flight envelope. Manufacturers have dealt with this problem through the definition of controller gains in function of easily measurable parameters (e.g. dynamic pressure), a process called gain scheduling. Although conceptually simple and extensively proven, this process is time-consuming and costly.

This thesis explores two approaches to avoid common gain scheduling problems. First, long design times are addressed by numerical optimization through structured H_∞ synthesis. This robust optimization method tunes a controller to meet design constraints for a pre-defined controller architecture. The conceived methodology enables the designer to translate design objectives into constraints compatible with the H_∞ framework, even in the presence of large uncertainties such as the absence of information on the aircraft's mass and centre of gravity.

The second approach addresses multiple weaknesses of gain scheduling approaches, such as the design time and the need to update gains after aircraft model changes (e.g. in reaction to flight tests). The control law is defined from a simplified model, which is a good approximation of the aircraft in a large portion of its envelope. Doing so results in a control law "by equations" (G*), where primary aircraft dynamics are placed according to simple low-order approximations, avoiding scheduling whilst keeping the possibility to fine-tune the aircraft as desired. Practical longitudinal low-order approximations of the aircraft are developed for this methodology. As modern wing profiles are highly optimized, they are generally prone to strong nonlinearities, limiting the validity of the simplified model used. This is why a nonlinear control law is added to classical linear gains to ensure the aircraft behaves according to the simplified model, ensuring satisfactory dynamics.

Finally, both approaches are tested and compared on an aircraft model provided by Airbus Canada. The conformity of both approaches to design requirements from literature will be demonstrated on this model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF SYMBOLS

APR	Average Phase Rate
$\mathbf{A}$	Matrix
$\mathbf{a}$	Vector
a	Scalar
$\hat{A}$	Hats are used to highlight estimated values
A_c	The c suffix is used for commanded values
A_{CL}	The CL suffix is used for closed-loop values
A_e	The e suffix is used for equilibrium values, unless noted otherwise
A_f	The f suffix is used for filtered values
A_m	The m suffix is used for measured values
A_{OL}	The OL suffix is used for open-loop values
α	Angle of Attack (AoA)
β	C* mixing ratio
BW	Bandwidth
Drb	Dropback handling quality ($\frac{Drb}{q_{ss}}$)
$\bar{c}$	Mean aerodynamic chord
C_i	Lift (L) or moment (m) coefficient. Additional suffixes correspond to the derivative of C_i w.r.t. another variable
cg	centre of gravity
δ_e	Elevator position
Δ	State deviation from its equilibrium value
$\bar{\Delta}_x$	Ratio of the cg position w.r.t. $\bar{c}$ from the aerodynamic centre, see eq. [3.19]
γ	Flight path angle. In the context of a literal transfer function, it is used for the integrator pole to SP real part ratio
g	Gravitational constant
h	Altitude
K	Gain
m	Aircraft mass
M	Mach number
M_{Mo}	Maximum operating Mach
M_D	Design Mach
n_z	Load factor (vertical acceleration in the body axis normalized by g)

N_1	Low pressure turbine speed
OS	Overshoot
PRO	Pitch Rate Overshoot
PRS	Pitch Rate Sensitivity
q	Pitch rate
$\bar{q}$	Dynamic pressure
s	Laplace variable
S	Wing surface
τ	Time constant of a real pole
θ	Pitch angle
T	Time constant of a real zero
T_{θ_2}	Time constant of the short period zero
V_D	Design speed
V_{Mo}	Maximum operating speed
V_{sr}	Stall reference speed
V_T	True Airspeed
ω	Natural frequency
$\bar{x}_{cg}$	Normalized (% of $\bar{c}$) cg position in the x body axis
$\bar{z}_{cg}$	Normalized (% of $\bar{c}$) cg position in the z body axis
ζ	Damping ratio

LIST OF ABBREVIATIONS

AC	Aircraft
ASE	Aeroservoelasticity or Aeroservoelastic
ISA	International Standard Atmosphere model
HQ	Handling Quality
LTI	Linear Time-Invariant
LTV	Linear Time-Variant
MRAC	Model Reference Adaptive Control
NDI	Nonlinear Dynamic Inversion
PH	Phugoid mode
SAC	Simple Adaptive Control
SP	Short Period mode
S.M.	Stability Margins
w.r.t.	with regards to

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 Context

The introduction of Fly-By-Wire (FBW) technologies has been a significant improvement for aviation in the last 50 years. The replacement of mechanical links between pilot input and control surfaces by a digital interface has allowed for large weight reductions, which first materialized in military aviation [3]. Some noteworthy early FBW implementations include the CF-105 in 1958, with the first fly-by-wire controls, and the Concorde in 1969 with the first analog FBW system. The (then) General Dynamics F-16 in 1976 and Airbus A320 in 1987 were the first digital installments of FBW in military and commercial aviation, respectively, to enter production.

The importance of flight control has grown significantly in commercial aviation since its introduction, roughly 35 years ago. Modern control laws now provide a multitude of benefits from ease of control for the pilot to envelope protection, ensuring aircraft cannot exit their operational envelope. Although this results in competitive safety benefits, flight controls also provide ease of trimming and other functions that can increase the efficiency of aircraft, reducing fuel consumption. Furthermore, the reduction of pilot workload and the possibility of designing an aircraft family with similar characteristics open the door to shorter pilot training programs, resulting in operational cost reduction. All in all, flight control systems and FBW are now well integrated in commercial aviation.

Design of flight control laws has historically been done through LTI (Linear Time-Invariant) methods, because of the simplicity and flexibility that they allow for both tuning and validation. Furthermore, design is usually done through a process called gain scheduling, in which local controllers are conceived for a large number of flight points that span the operational envelope of the aircraft [4]. Once this is done, the local controllers are combined into discrete tables of gain values in function of quantities of interest called scheduling variables. This process has the advantage of splitting the complex flight control problem into a large number of relatively simple LTI problems. Although some conservatism is to be expected in commercial aviation practices due to extensive certification processes, fighter aircraft were also limited to such classical techniques until quite recently, including the F-22 in the United States [5].

Although classical design methods have been extensively proven to work throughout the years, they also incur significant costs for the designer due to their iterative and time-consuming nature [6]. For aircraft manufacturers, this is one of the most significant problems of these methods. Nonetheless, gain scheduling also suffers from other drawbacks, such as the need to re-define gain tables upon changes to identified aircraft dynamics or feedback loop delays, often being seen as a roadblock for future aircraft improvements. Finally, it is important to mention that classical gain-scheduling has long been described as ad-hoc in academic research due to the lack of theoretical stability guarantee [7].

1.2 Industrial context and Airbus Canada

The research activities descried in this thesis were done in collaboration with Airbus Canada, allowing the author to perform validation of control laws designed with state-of-the-art industrial data and models. Beyond further refining of the developed methodologies, this also increases the trust in their performance and validity. Nonetheless, industrial models are valuable because of the significant resources their development requires and because they describe aircraft in great detail. This is why details that can be given on such models are limited. Therefore, aircraft data on which research activities were performed will be referred to as "target aircraft". This is also why some information, e.g. some dynamics in the feedback loop, cannot be given in full detail.

1.3 Research objectives

The principal research objective is the conception of longitudinal C* flight control laws subject to handling quality requirements, in the context of commercial aviation. This objective comprises two sub-objectives:

- The conception of a methodology based on numerical optimization for gain scheduled designs that need robustness in mass and centre of gravity (cg) position. The intent is to improve and streamline gain tuning processes while remaining compatible with currently certified architectures. With classical methods, meeting this kind of robustness results in very iterative procedures and long development periods. This is why the use of a robust method like structured H_∞ synthesis will be key to the fulfillment of this objective.

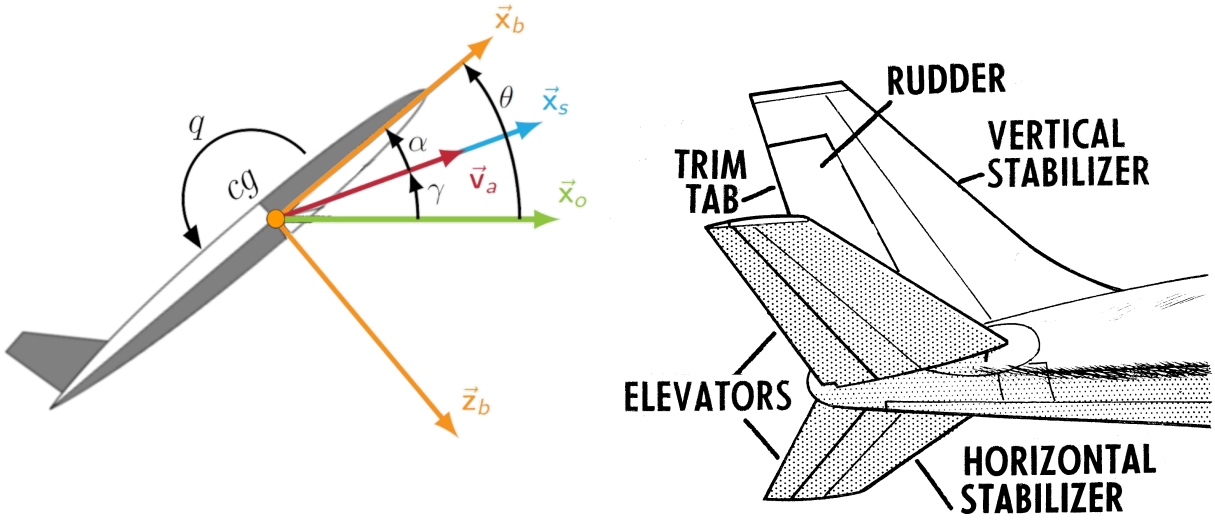
- The conception of a methodology to express longitudinal gains explicitly in function of aircraft aerodynamics, along with the compensation of aircraft nonlinearities. The definition of a control law directly in function of aircraft aerodynamics is appealing, as this has the potential to avoid multiple drawbacks of gain scheduling approaches. The G* concept (control law by equations) will be at the heart of this methodology.

1.4 Thesis outline

This thesis begins with a literature review in chapter 2, allowing to discuss the current state of longitudinal design methods and recent developments. Main handling qualities, which will be essential to evaluate the results from the methodologies conceived, will also be introduced. In chapter 3, aircraft dynamics and control loop architectures are then defined and analyzed to develop an understanding of the aircraft. Then, in chapter 4, a first methodology aligned with the first research sub-objective is detailed, based on structured H_∞ synthesis. The second sub-objective is addressed in chapter 5 through the introduction of the G* concept, and its application to longitudinal design. In chapter 6, results obtained with both methodologies are shown and compared. The thesis ends with chapter 7, which summarizes the work done and the main achievements.

CHAPTER 2 LITERATURE REVIEW

In this chapter, flight controls literature is summarized. Following research objectives, emphasis is put on longitudinal control. The longitudinal axis, shown in figure 2.1a, is the axis created by the aircraft's axis of symmetry. Longitudinal control of the aircraft is possible through the elevators and the horizontal stabilizer, which are illustrated in figure 2.1b. In practice, the stabilizer is used by pilots for aircraft trimming while the elevators are used for aircraft manœuvring. This convention remains unchanged with modern control laws, which is why longitudinal control laws will act on the elevators.



(a) Longitudinal axis

(b) Typical aircraft control surfaces

Figure 2.1 Aircraft description

2.1 C* control law

To introduce the C* control law, a short summary of flight control is required. Early stability augmentation systems (SAS) were primarily concerned with providing improved short period damping through simple feedbacks. Nonetheless, it was realized in the 1960s that the pitch rate transfer presents a zero which can vary significantly and cause large overshoots, resulting in vastly different aircraft responses even for the same poles [8]. As the understanding of desirable aircraft response improved, more complex design requirements were developed,

beyond simple modal criteria. These new requirements resulted in the creation of more complex control augmentation systems (CAS), which enabled aircraft to track a reference signal (figure 2.2).

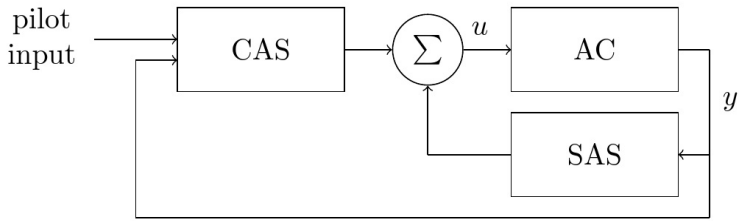


Figure 2.2 Illustration of CAS and SAS systems

As the intent of these systems is to ensure the aircraft behaves adequately in response to pilot inputs, they command aircraft elevators in the longitudinal case. Control augmentation systems created the need to translate pilot inputs into control objectives. Conceptually, a large number of outputs can be considered for longitudinal piloting, such as the pitch rate q, angle of attack α, load factor n_z, flight path γ, etc. It was theorized that at high speeds, the principal piloting cue was n_z, while control of q was more practical at low speeds. This hypothesis resulted in the development of the C* handling quality criteria, along with bounds on the C* values that "yielded adequate handling". Although the C* criteria has long been rejected, piloting of the C* value has remained popular, with a large number of applications in aviation [9].

Nowadays, the C* control law is now one of the main feedback quantities used in aircraft control. It is defined as :

$$C^* = \beta q + n_z \tag{2.1}$$

where β is set to $\frac{V_{co}}{g}$, the ratio of the crossover speed V_{co} and gravitational constant g. Typically, V_{co} is set around 240 Kts, giving $\beta \approx 12.6$ (although an additional factor of $\frac{\pi}{180}$ is needed if q is in $^\circ/s$). At this airspeed, the weight of the pitch rate and vertical acceleration is equal. The C* architecture therefore provides a natural blend between both q and n_z control laws, while keeping the tendency to stabilize the aircraft at 1 g without pilot input. It can be noted that this law has been further developed to include speed stability (C*U law), although this variant will not be used in this thesis [10].

2.2 Pilot-induced oscillations

Pilot-Induced Oscillations (PIO) have been one of the primary problems that control engineers had to address when developing FBW and flight control systems. Although some authors date the first PIO instances as early as the Wright brothers flights, the phenomenon did not receive proper attention until early fly-by-wire experiments, such as the NASA F-8 Crusader prototypes [11]. The difficulty of modifying aircraft dynamics before the use of fly-by-wire has in most cases led to the reduction of the operational envelope instead of truly addressing the source of PIO problems. Furthermore, the introduction of computational delays in fly-by-wire aircraft also has contributed to an increase in PIO occurrences. Despite having multiple possible causes, they take root in the sudden degradation of aircraft flight characteristics, ultimately leading to difficulty (or incapacity) of the pilot to accommodate to the new dynamics. The authors of [12] distinguishes between linear (type 1), quasi-linear (due to saturation, type 2) and fundamentally nonlinear PIOs (type 3). PIOs result in diverging oscillations, which in most cases may be stopped by the pilot letting go of the controls for a short time, hence the human nature of the phenomenon. Nonetheless, literature is very critical that PIOs are the result of a conception problem, with failed attempts to rename the phenomenon to a less accusatory designation. As most PIOs have been recorded in flight phases where precise attitude or flight path tracking is necessary (e.g., landing), diverging oscillations or letting go of the controls both put the mission at risk and need to be avoided.

2.3 Handling qualities

In response to these difficulties, a large number of handling quality (HQ) metrics were developed. Handling qualities are defined as "those qualities or characteristics of an aircraft that govern the ease and precision with which a pilot is able to perform the tasks required in support of an aircraft role." [13]. In essence, they characterize the "predictability" of an aircraft and the effort required to perform piloting tasks. Handling qualities are therefore at the heart of the definition of design requirements for control laws. The vast majority of handling qualities were developed for military aircraft (especially fighters) during the cold war, meaning that a preliminary study must be performed for commercial aircraft application. As there exists a multitude of HQ metrics, aircraft manufacturers generally have a subset of metrics they use, selected through such studies and past experiences. Conformity to handling qualities is generally described by multiple levels, as shown in table 2.1. It is important to note that despite the numerous criteria used to predict PIO-prone designs or pilot discomfort, pilot validation remains essential.

Table 2.1 HQ levels description (Level 1-3 descriptions are from on [1])

Level	Qualitative Summary	Description
1*	Excellent	Beyond design objectives.
1	Satisfactory	*Flying qualities clearly adequate for the mission. Desired performance is achievable with no more than minimal pilot compensation.*
2	Acceptable	*Flying qualities adequate to accomplish the mission, but some increase in pilot workload or degradation in mission effectiveness, or both, exists.*
3	Controllable	*Flying qualities such that the aircraft can be controlled in the context of the mission, even though pilot workload is excessive or mission effectiveness is inadequate, or both.*

As studying the multitude of metrics available in literature and gaining experience with them takes resources, aircraft manufacturers are not keen on divulging their design requirements. Another explanation is that design targets contain information about aircraft performance. This is problematic, as HQs are also what allows to compare a control law to another. In the context of this thesis, HQ limits (separation between the levels) will be based on commonly available data from literature. Nonetheless, as doing a comprehensive review of HQs is out of the scope of the research objectives, these requirements will be simplified. The intent is to give the inexperienced reader a sufficient understanding of common handling qualities, allowing him to evaluate and compare the control laws that will be obtained. Doing so will also permit to introduce key design HQs that will shape the methodologies developed later. Design requirements are given in table 2.2 and are introduced in the following subsections, along with their source. For reference, the aircraft is considered class III ("Large, heavy, low-to-medium maneuverability aircraft") and category B or C ("non-aggressive tasks") [14].

Table 2.2 HQ requirements

	Level	Minimum	Maximum	Reference	Model
Gain margin (dB)	-	6dB	-	[15]	Full
Phase margin (°)	-	45°	-	[15]	Full
ζ_{SP}	1	0.35	1.3	[14]	Full
	2	0.25	2	[14]	
τ_p (s)	1*	-	0.12	[11]	Full
	1	-	0.15	[14]	
	2	-	0.18	[14]	
BW_θ (rad/s)	1*	2	-	[11]	Full
	1	1.30	-	[14]	
	2	0.75	-	[14]	
BW_γ (rad/s)	1	0.55	$f(BW_\theta)$	[14]	Full
	2	0.35	$f(BW_\theta)$	[14]	
APR (°/Hz)	1	-	85	[16]	Full
	2	-	145	[16]	
	3	-	195	[16]	
f_{180} (Hz)	1	0.5	-	[16]	Full
	2	0.38	-	[16]	
PRS (dB)	1	-6	+1	[16]	Full
Drb	1*	0	0.25	[17]	SP
PRO (Ratio)	1	1	$f(Drb)$	[14]	SP
n_z OS (%)	-	-	To minimize	-	SP

Some of these requirements are to be validated on the complete (full) closed-loop aircraft model, while most time domain requirements are validated on the short period approximation of the aircraft rigid modes (to be defined in chapter 3).

Stability margins

The stability margins given must be met within the operational envelope, between 0.06 Hz and the frequency of the first aeroelastic modes. They are measured from a perturbation of the elevator command to the corresponding elevator command. Below 0.06 Hz, margins can be relaxed to 30° and 4.5 dB.

Modal requirements

The damping ratio requirement is valid for the short period mode. Although this reference also contains a requirement for the phugoid mode, it will not be considered in this thesis.

Attitude bandwidth (BW_θ)

Along with the phase delay (described below), the attitude bandwidth is a strong criterion for the prevention of PIOs. The attitude bandwidth corresponds to the minimum frequency in the attitude (θ) response at which either the phase is equal to $-135°$ or the gain is 6 dB higher than the gain at $-180°$ of phase (see figure 2.3). The level 1* is interesting, as it ensures that no PIO should occur even in the presence of large dropbacks, another HQ to be defined below.

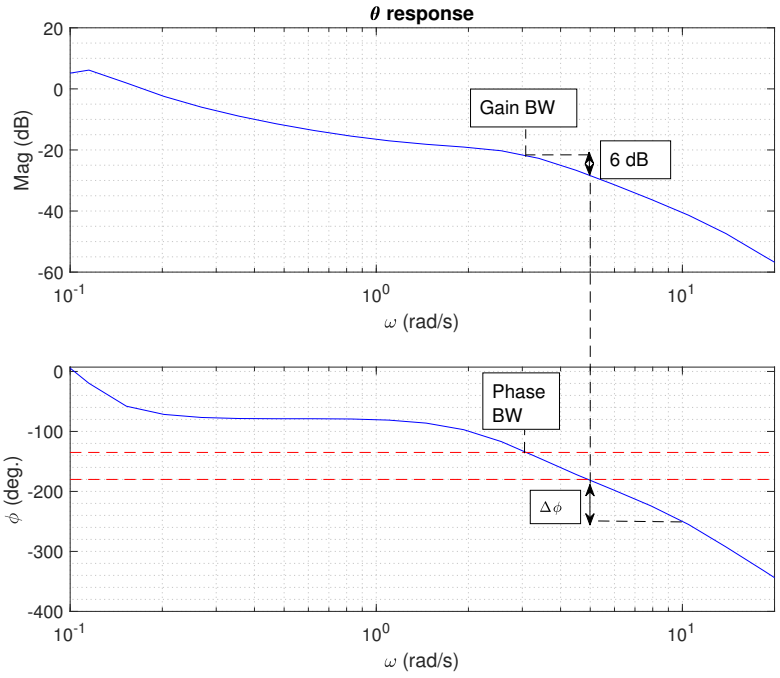


Figure 2.3 Illustrations of the attitude bandwidth BW_θ and phase delay τ_p

Phase Delay (τ_p)

The phase delay corresponds to the phase degradation rate in the pitch response (θ) between the frequency where pitch phase reaches $-180°$ (ω_{180}) and twice this frequency :

$$\tau_p = \frac{\pi}{180} \times \frac{-\phi(2\omega_{180}) - 180°}{2\omega_{180}} \tag{2.2}$$

This is illustrated in figure 2.3, where $\Delta\phi = -\phi(2\omega_{180}) - 180°$.

Flight path bandwidth (BW_γ)

The flight path bandwidth corresponds to the frequency at which the phase of the flight path (γ) response reaches $-135°$. Although the given reference value has a maximum limit in function of the BW_θ, this limit becomes quite high when BW_θ is level 1*. Nonetheless, the maximum is given by $1.2BW_\theta - 1$ for the level 1 limit or $1.2BW_\theta - 0.5$ for level 2.

Pitch Rate Sensitivity (*PRS*)

The Pitch Rate Sensitivity (PRS) corresponds to the gain of the pitch rate response, measured at the phase bandwidth (ω_{135} in θ). Limits in table 2.2 are intended for military aircraft, limiting their appeal here. Nonetheless, this metric remains interesting so long as these limits are re-evaluated.

Gibson PIO criteria

Gibson [16] introduces the Average Phase Rate (APR) and the frequency at $-180°$ of phase f_{180}. The APR is equal to the phase delay presented previously (but expressed in °/Hz), while the f_{180} is a requirement similar to the phase portion of the BW_θ, but expressed in Hz and measured at $-180°$ of phase. Once converted in the same units, the 1* level of APR is stricter than τ_p. This is because Gibson developed his criteria for fighter aircraft. As 1* f_{180} (1 Hz) is much higher than the BW_θ 1* requirements, the level 1* will be dropped for Gibson criteria. In this reference, the stick is described as a static gain. As such, Gibson criteria will be measured on the closed-loop, excluding the physical dynamics of the stick. Although Gibson also gives a maximum gain value at $-180°$ of phase in the pitch response, this criterion is too relaxed to be of use. Indeed, for all gains designed (including some that largely failed other requirements), this requirement was always met unless one of the other Gibson criteria was extremely degraded, hence why it is omitted.

Dropback (Drb)

The dropback is one of the primary handling qualities and describes the behaviour of the aircraft's nose upon letting go of the stick. Figure 2.4 shows the dropback criterion for a 10 s step. Note that although the dropback refers to the amount by which the pitch angle drops when the stick is released, this quantity is normalized by the steady-state pitch rate ($\frac{Drb}{q_{ss}}$). In this document, the normalization by the q_{ss} is implied when using the term dropback, or the abbreviation Drb, as it is always referring to the dropback design requirement. Positive dropbacks are appreciated since pilots can release the stick after reaching the desired pitch. Aircraft with negative dropbacks are usually described as unpredictable since pilots must let go of the stick before reaching the desired pitch. Although only a level 1* is given, the level 1 limit must be evaluated in conjunction with the pitch rate overshoot (PRO) (defined below). If the dropback and PRO are within the region delimited by $PRO = 3 - 0.6Drb$, the dropback is satisfactory. High dropback may cause "bobbing" tendencies, but are not at risk of causing PIOs as long as BW_θ is high enough and τ_p is low enough. No strict negative limit was found in literature, although strong negative values should be avoided as much as possible.

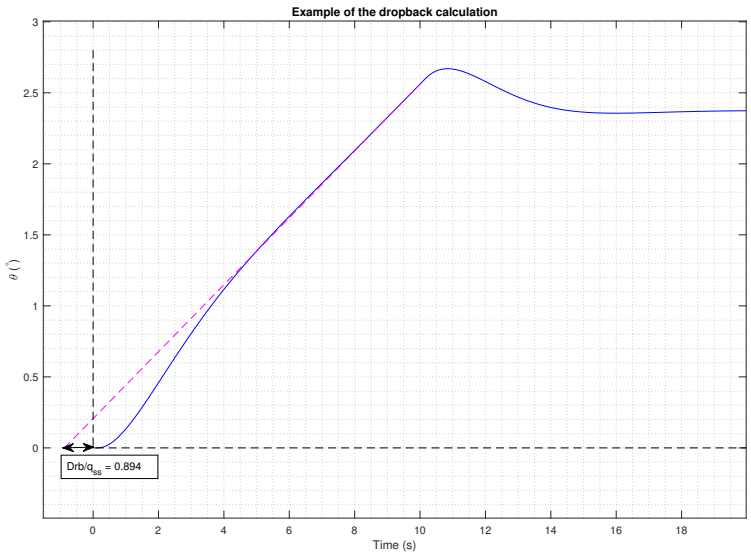


Figure 2.4 Dropback example

The Drb (w.r.t. steady-state pitch rate) may be evaluated analytically from a transfer function in two distinct ways [18]. Let a transfer function be described by:

$$\frac{q}{\delta_{ref}} = K\frac{\prod_{i=1}^{k}(1+T_i s)\prod_{i=1}^{l}(s^2+2\xi_i\phi_i s+{\phi_i}^2)}{\prod_{i=1}^{m}(1+\tau_i s)\prod_{i=1}^{n}(s^2+2\zeta_i\omega_i s+{\omega_i}^2)}; \phi_i \neq 0, k+2l < m+2n \tag{2.3a}$$

$$= \frac{a_{k+2l}s^{k+2l}+...+a_1 s+a_0}{b_{m+2n}s^{m+2n}+...+b_1 s+b_0} \tag{2.3b}$$

The corresponding dropback is :

$$Drb = \left(\sum_{j=1}^{k} T_j + \sum_{j=1}^{l} 2\frac{\xi_j}{\phi_j}\right) - \left(\sum_{j=1}^{m}\tau_j + \sum_{j=1}^{n} 2\frac{\zeta_j}{\omega_j}\right) \tag{2.4a}$$

$$= \frac{a_1}{a_0} - \frac{b_1}{b_0} \tag{2.4b}$$

From this, it is clear that the dropback of distinct components in series is simply the sum of their respective dropback and that K has no effect on this HQ.

Overshoot (*PRO* and n_z *OS*)

The *PRO* (Pitch Rate Overshoot) has a similar role to the dropback; pilots prefer significant overshoots in q for the same reasons. The upper limit is directly linked to the maximum dropback value. Control of the load factor (n_z) is essential, as exceeding the maximum design limit will result in significant damage to the aircraft. Therefore, nonlinear n_z overshoot must be limited according to design limits. Although this does not directly translate into a linear model requirement, it is likely that large linear overshoots will result in large nonlinear overshoots. This is why this metric must be limited as much as possible, although the limit is ultimately dependent on the target aircraft's design. It is interesting to note that the large difference in overshoot expectations between the q and n_z responses corresponds to natural aircraft dynamics. Indeed, there is usually a slow zero ($\frac{-1}{T_{\theta_2}}$) in q which significantly increases overshoots, while the zeros in n_z are negligible w.r.t. their effect on overshoots. This will be described in more detail in chapter 3.

OLOP

While the majority of the previous PIO criteria intend to verify the absence of linear (Type 1) PIOs, the OLOP criterion is intended to validate the absence of PIO due to command saturation (Type 2). Instead of predicting whether rate saturation occurs, the OLOP criterion predicts whether the aircraft in presence of rate saturation is PIO-prone. Rate limiting elements in the pilot-augmented aircraft closed-loop have been shown to cause sudden phase

increases ("phase jumps"), leading to reduced phase and amplitude margins, prone to cause instability [19]. The OLOP criterion is based on the computation of the open-loop (pilot-aircraft loop opened at the saturating element) response corresponding to the onset frequency, defined as the frequency where the first closed-loop saturation occurs. This frequency (called Onset Point) corresponds to the intersection of an integrator (-20 dB/dec.), whose cross-over frequency is set to the rate saturation value, with the (linear) closed-loop response from stick input (for maximum input amplitude) to the input of the saturation element. Figure 2.6a shows how to compute the Onset Point. This linear closed-loop is obtained by replacing the saturation component by a gain. Figure 2.5 illustrates the pilot-in-the-loop for OLOP computation. Once the OLOP phase and magnitude are obtained, they are displayed in a Nichols chart and validated against an upper boundary as in figure 2.6b. This summarizes the OLOP as described in [19], while the limits used are from [20].

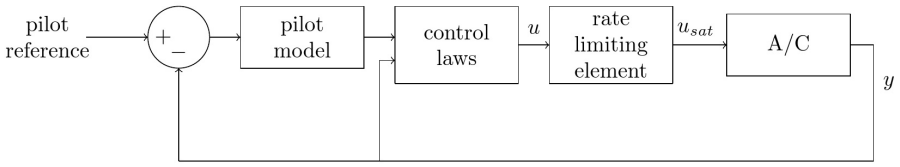
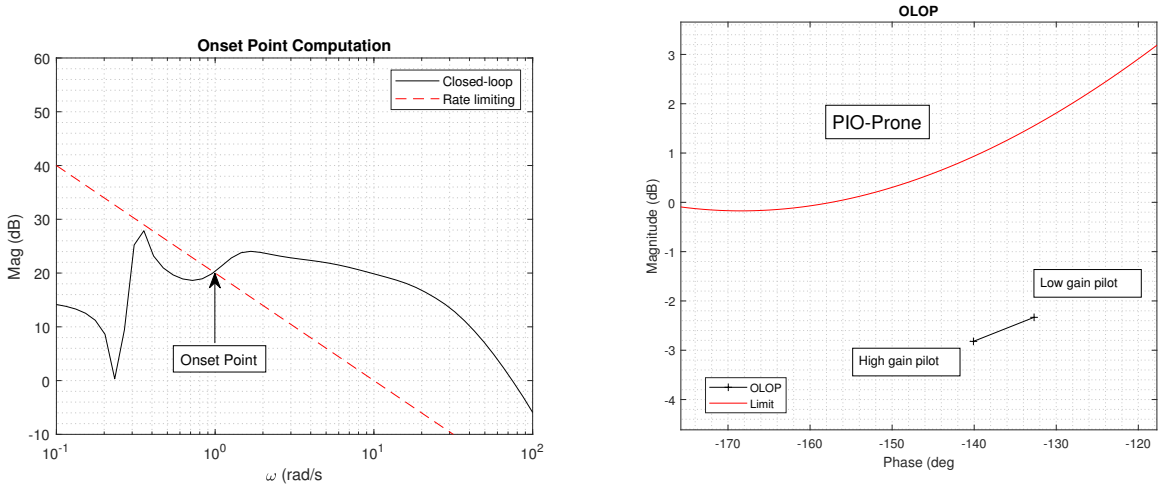


Figure 2.5 Pilot-in-the-loop diagram for the OLOP computation

All that is left is to define a pilot model. It has been shown that pilot behaviour can be assimilated to a gain in presence of fully developed PIOs [21]. Reference [19] suggests pilot gains that bring the phase at the crossover of the closed-loop system from -110 to -160 degrees. This requirement will be validated against a "low-gain" and a "high-gain" pilot. Failure of the high gain pilot suggests the design may be PIO-prone for aggressive piloting, but not critical, while failure for the low gain pilot is critical. These gains may be obtained from the closed-loop frequency response from the stick force to pitch angle transfer. By finding the gain corresponding to the desired phase for the crossover ($A(\phi_c)$), the pilot gain may be set to $1/A(\phi_c)$.

2.4 Linear control methods

Historically, linear control methods have been adopted by aircraft manufacturers out of a need for simplicity. The ability to use linear design tools at a time when computational power was limited (real-time or not) was a necessity. Nonetheless, linear design remains a crucial element of aircraft control to this day [22]. This is because most handling qualities were



(a) Example of Onset Point computation

(b) Example of OLOP validation

Figure 2.6 Illustration of the OLOP validation process

developed in the LTI framework, making linear tools essential for design. Furthermore, gain scheduling comes as a logical consequence of aerodynamic models, which generally consist of multiple tables that are recombined and interpolated in simulators. Finally, the "intuitive" nature of classical architectures is extremely important to certification authorities, leading to designers having reservations about using unproven and less intuitive architectures, partially due to their inherent validation challenges (e.g. difficulty of identifying a worst case).

As mentioned, classical architectures are generally gain-scheduled. This process allows designers to study linear approximations of the aircraft for multiple points, spanning the envelope, and choose gains locally through a linear design process. These gains are then combined into tables, which are interpolated in real-time in function of relevant variables, named scheduling variables. Reference [4] gives an extensive review of gain-scheduling practices in multiple fields. The use of exogenous signals in the controller leads to additional feedbacks when linearizing the whole closed-loop plant, called hidden coupling terms, which can affect the system's stability. A key assumption in the application of gain scheduling is that the scheduling variables are slowly varying [22], which is the case in aviation. Indeed, common scheduling variables such as M or $\bar{q}$ will have much slower dynamics than short period state variables. It is interesting to note that even if longitudinal dynamics are very much dependent on α near stall, this state varies quickly, bringing into question whether it can be used as a scheduling variable. Although "clean sheet" designs can require extensive designer efforts [6], gain scheduling allows simple, yet time-demanding, fine-tuning of gains (e.g. in

reaction to flight tests), another reason explaining the method's popularity. Because design and validation are done on a grid of scheduling parameters, there is no guarantee that worst cases are captured in the validation process [7]. These elements constitute some of the main critiques of gain scheduling, along with its ad-hoc nature.

Although many of these critiques are related to the nature of gain scheduling, the main problem for industrial application remains the time required to obtain a tuned schedule. This justifies the need of accelerating the design process through numerical optimization. Multiple in-house tools have been developed, including at Airbus Canada for the initial design of gains for the target aircraft. In literature, `CONDUIT` is a notable example of such a tool, developed by NASA [23], with some detailed application examples [24].

Beyond proprietary optimization tools, H_∞ synthesis solvers have gained in practicality in the last decade. Before this, classical H_∞ synthesized a "black box" controller of a very high order, generally requiring order reduction for implementation. Despite this, numerous applications of classical H_∞ synthesis to aircraft control may be found in literature [25], [26]. The introduction of structured H_∞ synthesis in the last decade has allowed the use of the H_∞ framework for design objectives, along with the optimization of a predefined control architecture. This process removes the need to perform controller order reduction entirely while simplifying problems related to scheduling as it is possible to define polynomial gain surfaces to optimize. Structured H_∞ has seen an increased number of directly relevant applications in literature [27], [28], making this approach very promising.

A different approach, which is not based on numerical optimization, is to define aircraft gains directly from parameters that define the aircraft's dynamics, approach named control law by equations. Nonetheless, it is more natural to introduce this methodology through a discussion on nonlinear control. Refer to section 2.5.1 for this introduction.

2.4.1 H_∞ theory and background

To introduce H_∞ synthesis, it is interesting to introduce the standard form of a system as in figure 2.7, where $\mathbf{u}$ represents systems inputs, $\mathbf{y}$ measured outputs, $\mathbf{w}$ exogenous inputs, while $\mathbf{z}$ are regulated outputs. A partitioning of $\mathbf{P}(s)$ is introduced to express each output in function of the corresponding input:

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11}(s) & \mathbf{P}_{12}(s) \\ \mathbf{P}_{21}(s) & \mathbf{P}_{22}(s) \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} \tag{2.5}$$

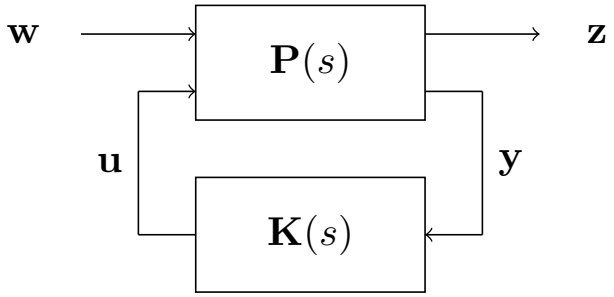


Figure 2.7 Standard form with controller

This allows to write the transfer from $\mathbf{w}$ to $\mathbf{z}$ as an LFT (Linear Fractional Transformation):

$$\mathbf{T_{zw}}(s) = \mathcal{F}_{\mathbf{l}}(\mathbf{P}(s), \mathbf{K}(s)) = \mathbf{P}_{11}(s) + \mathbf{P}_{12}(s)\mathbf{K}(s)\left(\mathbf{I} - \mathbf{P}_{22}(s)\mathbf{K}(s)\right)^{-1}\mathbf{P}_{21}(s) \tag{2.6}$$

H_∞ synthesis attempts to minimize the H_∞ norm of $\mathbf{T_{zw}}(s)$ through the stabilizing controller $\mathbf{K}(s)$, where the H_∞ norm is given by:

$$||\mathbf{T_{zw}}(s)||_\infty = \max_{\omega\in\mathbb{R}} \bar{\sigma}\left(\mathbf{T}_{zw}(j\omega)\right) \tag{2.7}$$

with $\bar{\sigma}(\mathbf{T})$ corresponding to the maximum singular value of the complex matrix $\mathbf{T}$. This is used to formulate design objectives on $\mathbf{T}(s)$, a sub-transfer matrix of $\mathbf{T_{zw}}(s)$, through weighting functions $\mathbf{W}(s)$ such that:

$$||\mathbf{W}_L(s)\mathbf{T}(s)\mathbf{W}_R(s)||_\infty < 1 \tag{2.8}$$

These weighting functions can be used for multiple purposes, such as disturbance rejection, reference tracking or stability margins. Ultimately, all weighting functions used are combined as $\mathbf{H}(s) = \text{diag}\left(\mathbf{W}_{L,1}(s)\mathbf{T_1}(s)\mathbf{W}_{R,1}(s), ..., \mathbf{W}_{L,i}(s)\mathbf{T_i}(s)\mathbf{W}_{R,i}(s), ..., \mathbf{W}_{L,n}(s)\mathbf{T_n}(s)\mathbf{W}_{R,n}(s)\right)$ and are applied by enforcing $||\mathbf{H}(s)||_\infty < 1$.

As mentioned, the main drawback of classical H_∞ synthesis is that the controller synthesized will be of an order equal to that of the augmented $\mathbf{P}(s)$ system, resulting in complex controllers in general. Structured H_∞ synthesis uses the same framework and general goal as the classical H_∞ synthesis, with the major difference that $\mathbf{K}(s)$ now has an imposed structure. Therefore, $\mathbf{K}(\boldsymbol{\lambda}, s)$ is written as a diagonal matrix of $\mathbf{K}_i(\boldsymbol{\lambda}, s)$, where $\mathbf{K}_i$ corresponds to transfer functions and $\boldsymbol{\lambda}$ is the set of gains or parameters to optimize. While classical synthesis leads to a global optimum, structured H_∞ synthesis results in non-convex problems which can have multiple optimums, leading to a dependency on initial conditions [2].

2.5 Nonlinear control

Relevant nonlinear control approaches found can be separated into three broad categories, with common methods in parenthesis. Note that the separation into categories is used to introduce the following discussions, as some methods can be argued to belong to multiple categories at once. These categories will be introduced briefly here to give a general understanding and then discussed in greater details in the following pages.

- Adaptive methods
  - Direct methods (Model Reference Adaptive Control (MRAC) (and variants), Simple Adaptive Control (SAC), L1-AC, etc.)
  - Indirect methods
- Nonlinear dynamics inversion (NDI)
  - Direct term cancellation
  - NDI-based control laws
  - Incremental NDI (INDI)
- Robust control methods

Adaptive control concerns control architectures that adapt as the controlled system is changing [29]. In that sense, a gain-scheduled design could be argued to be a basic application of adaptive control. To fully differentiate between feedback control and adaptive control, the latter may be seen as two interconnected control loops. The first one is in charge of computing the command, similar to a normal feedback controller, while the second "higher-level" loop updates the control parameters used in the first loop. There are two broad categories of adaptive control architecture. In direct adaptive control, the controller will attempt to match a set of controlled quantities in the system with a reference model. In doing so, control parameters are updated "directly", without intermediary results. Although direct methods have strict requirements on the controlled system, they generally guarantee asymptotic tracking when these requirements are met. In contrast, indirect adaptive methods will first estimate a set of quantities of interest from the plant. Control parameters are then updated in function of estimated quantities, e.g. by solving a feedback problem. Unlike their counterpart, indirect methods usually have no strong theoretical requirements, but they offer no performance guarantee either.

Nonlinear Dynamic Inversion (NDI) is, according to [30], a type of feedback linearization where the controller is intended to cancel known (modelled) nonlinearities. One application type of NDI is explicit nonlinearity cancellation of known dynamics. Another approach, in the context of aeronautical applications, is to conceive architectures such as to match angular body rates predicted by a reference model, using known aircraft dynamics to limit tracking errors. It is interesting to note that the F-35 primarily uses this kind of control scheme for its control laws [31]. The technique has also been generalized to incremental nonlinear dynamic inversion, allowing to tolerate larger errors in modelization, although precise command effectiveness knowledge is generally still required. In this NDI variant, derivatives of the commanded signals (or their estimates) are compared to those predicted by a reference model. Given the error on the commanded signal derivatives, a command is computed such as to cancel these errors. One flight test of a law using this approach was found in [32] with moderate implementation success. The principal weakness of NDI is the dependency on modelled dynamics, while INDI design require the estimation of angular accelerations.

Robust Control is defined as a control approach in which the controller guarantees some level of performance when the plant varies within bounded uncertainties [33]. As such, robust control should be expected to yield lower performance than other approaches mentioned. Robust control has been added to allow to fit classical design methods like the one used on the target aircraft, which are neither adaptive nor inversion-based and have to be robust w.r.t. centre of gravity position.

2.5.1 Adaptive Control

Adaptive control methods (and specifically direct methods) have received a lot of attention since the early stages of aircraft control, as they seem to directly address problems that lead to gain scheduling, namely, the significant variation of the system throughout the operational envelope. Literature is very extensive on these methods, with an early implementation and flight test on the X-15 program [34] and multiple implementations relevant to commercial aviation found (the American AirSTAR [35], [36] and Calspan [37] demonstrators being highly promising). Nonetheless, some methods (L1-AC and some MRAC modifications like C-MRAC) are usually implemented such as to have high gain update frequencies [38], which seem unrealistic for commercial aviation. On the other hand, SAC and MRAC seem promising and literature contains implementations and flight tests on computationally-challenged architectures, suggesting realistic requirements [39], [40]. Nonetheless, direct adaptive meth-

ods have strict theoretical requirements to ensure asymptotic convergence to the reference model (MRAC [41], SAC [42]). In general, one of the requirements can be simplified to minimum-phase behaviour in SISO cases. This is problematic, as some aircraft outputs of interest, such as the load factor (and C*), have unstable zeros. Furthermore, asymptotic convergence on an Linear Time-Variant (LTV) reference model is more difficult to show, limiting the appeal for aircraft that requires a variant reference model for the SP depending on flight conditions.

SAC has been tested briefly in this project for the pitch rate control on a simplified (constant adimensional stability derivatives[1]) nonlinear aircraft model, with moderate success. This experiment will not be detailed as it was conducted very briefly, but the tracking achieved was very sensible to the command amplitude. Furthermore, the control law attempted to cancel out the phugoid mode because it was not included in the reference model. This is problematic, as precise models of the phugoid are difficult to obtain, as later mentioned in 3.3.1, hence the difficulty of establishing a relevant reference model. In any case, PH control should not be achieved through the C* law as mentioned in section 3.3.2. Finally, validation of direct method architectures also remains challenging, as the dependency on initial conditions and transient measurements make the search for worst cases quite complex. These problems were deemed sufficient enough to reject direct methods, despite the short time spent on them.

The strong requirements associated with direct methods are no longer a problem with indirect methods, although this is at the cost of performance guarantee. The search for worst cases is therefore simplified because of the possibility to have a more "physical" or "intuitive" architecture, but more critical. Articles on real-time stability derivatives estimation with real-time control schemes using these quantities were published recently [43] (which was flight-tested in [40]). Nonetheless, the inherent validation problems of online parameter estimation seem difficult to overcome in the context of this project. Still, real-time estimation of aircraft dynamics seems interesting for failure cases and was extensively developed in [44].

Towards the G* methodology

The control law by equations concept, which will be formally defined in chapter 5, is a direct consequence of research on indirect adaptive control developments mentioned above. Reference [45] suggests a real-time estimation of aircraft stability derivatives, such as to feed a pole placement control law and ensure good handling qualities in the presence of a failure.

[1] Defined in section 3.2.

In the absence of such failures, it is straightforward to use known aerodynamics (which can be simplified and tabulated) and perform the same pole placement (online) [46]. This should yield better performance due to the reduced uncertainties, while also avoiding gain scheduling. This constitutes the heart of the linear G* control law, which has been formalized in [47]. Furthermore, given expected aircraft dynamics and the assumption that linear gain design for these dynamics is adequate, it is possible to augment the aircraft through NDI or INDI, such as to cancel nonlinearities that do not correspond to the expected dynamics. Such a law is detailed in [44], with emphasis on the pitch-up phenomenon.

2.5.2 Nonlinear Dynamic Inversion

NDI methods are also quite common in flight control literature. While adaptive control is a promising approach to reach good performance without the need for extensive identification, NDI promises increasing nonlinear performance as aircraft models are refined. Unlike adaptive control (especially direct methods), NDI control laws often retain an "intuitive" architecture, which is quite important for the comprehension and "trust" in the control law. This is further reinforced by the fact that at least one production (military) aircraft already uses this approach [31], in addition to flight-tested prototype applications (e.g. a test program on NASA F/A-18 [48]). It is interesting to note that in the case of the prototype mentioned, the controller's update rate is only 80Hz, which seems on the upper hand of the reasonable range for commercial aviation. These factors explain the large appeal of these methods, and why the G* control law will include some NDI control loops based on work from [44]. On the other hand, explicit term cancellation can only be as precise as the model used, which may become problematic for a commercial aircraft. Although large identification campaigns are a part of commercial aircraft development, they incur significant costs. This makes excessive model identification (compared to current accuracy) counter-productive, potentially rendering the conception of an exclusively NDI-based control law for commercial aviation difficult. Indeed, it is mentioned in the article detailing the F-35 application that the program has benefited from one of the largest wind-tunnel testing campaigns ever conducted, even for fighter aircraft.

On the other hand, INDI schemes present in literature are limited by loop delays and gain update frequencies. Indeed, to compare estimated angular accelerations to the target values, it is necessary to compensate the required signals by the relevant delays, such as to ensure that they are synchronized [49]. Failing to do so will result in systematic errors being measured, degrading stability. Furthermore, in the case of an angular rate law, the aircraft's angular

acceleration must be estimated as angular acceleration sensors are generally not present on aircraft. This is done through filtering, meaning that the time constant used for the derivation will be limited by the "speed" of the feedback loop. This has been confirmed through G* development, hence the limited appeal of a control law that would be only based on INDI for commercial aviation, with hardware currently available. Finally, [32] proposes an interesting solution to the use of INDI laws in the presence of actuator saturation. As actuator efficiency is necessary for INDI schemes, saturation of these components may put the aircraft at risk.

CHAPTER 3 SYSTEM DESCRIPTION AND ANALYSIS

In this chapter, aircraft dynamics will be described and studied. First, the classical nonlinear aircraft model will be defined. The analytical linearized longitudinal model will be established, based on literature, in function of aerodynamic coefficients. This will allow to explore the structure of relevant aerodynamic quantities, which will be approximated and compared to data from numerical linearization of the longitudinal dynamics. Then, useful short period approximations will be defined, along with the highlighting of quantities of interest. Flight and loading envelopes will be specified, allowing to study the evolution of highlighted elements throughout these envelopes. After studying open-loop dynamics, the control loop will be introduced. Low-order approximations of the closed-loop aircraft will be developed, along with analytical expressions for the dropback and load factor overshoot. These approximations constitute key results for the G* methodology. Finally, the closed-loop pole configuration of the complete aircraft dynamics will be studied for a single flight condition.

3.1 Nonlinear model

Nonlinear models are essential for aircraft manufacturers. They allow to perform a wide range of tasks including failure cases studies, control law design and validation. The development and upkeep of such a simulation model is therefore a vital task and it is logical to expect any aircraft manufacturer to have a large set of simulation tools. As the base equations that govern rigid six degrees of freedom dynamics are the same from an aircraft to another, the details leading to these equations are not presented here for brevity's sake. Instead, emphasis will be put on the aerodynamic modeling relevant to longitudinal dynamics. The classical 6 degrees of freedom flat earth aircraft model is fully developed in [30]. The resulting nine equations of motion are separated into three groups:

Equations of forces:

$$\dot{u} = rv - qw - g\sin\theta + \frac{X_A + X_T}{m} \tag{3.1a}$$

$$\dot{v} = -ru + pw + g\sin\phi\cos\theta + \frac{Y_A + Y_T}{m} \tag{3.1b}$$

$$\dot{w} = qu - pv + g\cos\phi\cos\theta + \frac{Z_A + Z_T}{m} \tag{3.1c}$$

Equations of moments:

$$\Gamma\dot{p} = J_{xz}\left(J_x - J_y + J_z\right)pq - \left(J_z^2 - J_z J_y + J_{xz}^2\right)qr + J_z\left(L_A + L_T\right) + J_{xz}\left(N_A + N_T\right) \quad (3.2a)$$

$$J_y\dot{q} = \left(J_z - J_x\right)pr - J_{xz}\left(p^2 - r^2\right) + \left(M_A + M_T\right) \quad (3.2b)$$

$$\Gamma\dot{r} = \left(J_x^2 - J_y J_x + J_{xz}^2\right)pq - J_{xz}\left(J_x - J_y + J_z\right)qr + J_{xz}\left(L_A + L_T\right) + J_x\left(N_A + N_T\right) \quad (3.2c)$$

with $\Gamma = (J_x J_z - J_{xz}^2)$.

Rotational kinematics:

$$\dot{\phi} = p + \tan\theta\left(q\sin\phi + r\cos\phi\right) \quad (3.3a)$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \quad (3.3b)$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta} \quad (3.3c)$$

where:

- ϕ, θ, ψ are the roll, pitch and yaw Euler angles
- u, v, w are the components of the true airspeed of the aircraft, in the body axis
- p, q, r are the body angular rates
- X, Y, Z are the forces in the body axis
- L, M, N are the moments in the body axis
- The A subscript refers to aerodynamic efforts, while the T subscript refers to propulsive efforts
- m is the mass and J_{ij} are the moments and products of inertia.

Figure 3.1 illustrates quantities of interest. In this figure, p corresponds to an angular velocity around the x_b axis, q is around y_b and r around z_b. If expressed in body coordinates, $\mathbf{V}_A = [u, v, w]^\top$. It is assumed that the aircraft has a traditional configuration with the corresponding symmetry axis, leading to $J_{xy} = J_{yz} = 0$.

As aerodynamic efforts are generally defined in the wind axis rather than body axis, the

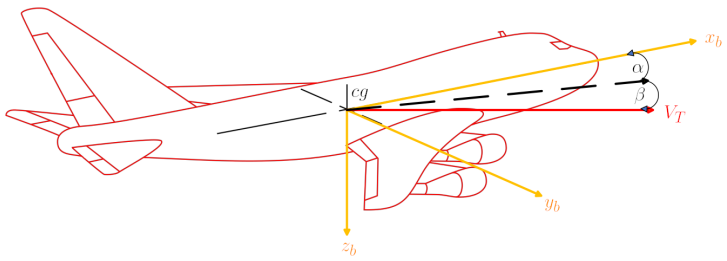


Figure 3.1 Aircraft axis

following quantities and transformation are defined:

$$V_T = \sqrt{u^2 + v^2 + w^2} \tag{3.4a}$$

$$\alpha = \text{atan2}\frac{w}{u} \tag{3.4b}$$

$$\beta = \text{asin}\frac{v}{V_T} \tag{3.4c}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -D \\ -C \\ -L \end{bmatrix} \tag{3.5}$$

where D is the drag, C is the lateral force, L is the lift force, α the angle of attack, β the angle of sideslip and V_T the true airspeed. In general, industrial models reconstruct the total drag and lift forces. Depending on the model, the lateral force can be given as C or Y directly. The structure of the moment and lift force will be explored in greater detail in the following sections. Nonetheless, it is important to remark that to perform dynamical simulations, the aerodynamic forces are needed in the body frame (X, Y, Z), as they are applied at geometrical reference points (aerodynamic centre, refer to [50]) which usually do not correspond to the cg position. Therefore, the computation of the moments at the cg requires a translation of forces by some distance, which is much easier to express in body coordinates.

3.2 Analytical linearization

Reference [30] develops the longitudinal linearization of aircraft dynamics for pseudo-equilibrium ($\dot{p} = \dot{q} = \dot{r} = \dot{V}_t = \dot{\beta} = \dot{\alpha} = 0$) in an implicit state-space form:

$$\mathbf{E}\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{b}\Delta\delta_e \tag{3.6}$$

For the longitudinal model, it is also assumed that $\beta = \phi = p = q = r = 0$ at equilibrium. Linearization results will only be detailed for the elevator input, as there is no intent to use the engine in control loops. In the equations below, $\Delta\mathbf{x}$ is a state deviation from equilibrium while $\Delta\delta_e$ is a deviation of the elevator w.r.t. the equilibrium value. The subscript e denotes equilibrium quantities, while the subscript T refers to quantities related to the propulsion system (excepted for V_{T_e}, which is the equilibrium value of V_T). In the following matrices, X_i, Z_i, M_i refer to dimensional stability derivatives and indicate the force component involved. Assuming that the engine only creates a moment around y_b (no power effects or torque effect), the longitudinal linearized dynamics are given by:

$$\Delta\mathbf{x} = \begin{bmatrix} \Delta\alpha \\ \Delta q \\ \Delta V_T \\ \Delta\theta \end{bmatrix} \tag{3.7a}$$

$$\mathbf{E} = \begin{bmatrix} V_{T_e} - Z_{\dot{\alpha}} & 0 & 0 & 0 \\ -M_{\dot{\alpha}} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7b}$$

$$\mathbf{A} = \begin{bmatrix} Z_\alpha & V_{T_e} + Z_q & Z_V - X_{T_v}\sin(\alpha_e + i_T) & -g\sin(\gamma_e) \\ M_\alpha + M_{T_\alpha} & M_q & M_V + M_{T_V} & 0 \\ X_\alpha & 0 & X_V + X_{T_V}\cos(\alpha_e + i_T) & -g\cos(\gamma_e) \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.7c}$$

$$\mathbf{b} = \begin{bmatrix} Z_{\delta_e} \\ M_{\delta_e} \\ X_{\delta_e} \\ 0 \end{bmatrix} \tag{3.7d}$$

where γ is the flight path angle given by $\gamma = \theta - \alpha$ and i_T is the engine angle w.r.t. the x_b axis. It is also possible to add the altitude state if desired, as the vertical speed is given by $\dot{h} = V_T \sin\gamma$:

$$\Delta\dot{h} = \sin\gamma\Delta V_T + V_{T_e}\cos\gamma\Delta\theta - V_{T_e}\cos\gamma\Delta\alpha \tag{3.8}$$

3.2.1 Longitudinal modes

Longitudinal open-loop rigid aircraft dynamics generally comprise three modes:

- The Short Period (SP) mode
- The Phugoid (PH) mode
- Altitude mode

Short Period: The SP mode is a rapid complex mode that is dominant in the short-term q, n_z and α responses. It is generally well damped and in the order of 1 rad/s. Good short period characteristics are critical, as the SP overlaps with the piloting frequency range. This is why this single mode is the focus of this thesis. Although this mode can degenerate in two real poles, one of which may be unstable, this configuration is generally limited to fighter aircraft which are designed to be statically unstable. Centre of gravity envelopes of commercial aircraft avoid this degeneration and, in practice, are defined from stricter criteria. As the short period is of primary interest, it is common to simplify longitudinal dynamics by truncating all non-SP modes, resulting in the short period approximation. This can be done by simply removing lines and columns in state-space matrices that belong to non-SP states, although more complex approaches may be of interest in specific cases. SP variables correspond to those that have fast dynamics ($\Delta\alpha$ and Δq with the variables used).

Phugoid: The PH mode is a slow complex mode that is dominant in the long-term speed and altitude responses, generally having a negligible effect on primary SP states. It is poorly damped or unstable, generally around 0.01 rad/s. The phugoid can also degenerate into two real poles. While controlling an unstable short period is a difficult and exhausting task (if possible) for a pilot, controlling an unstable phugoid mode poses no threat to the piloting task, beyond an increased workload. This is because the piloting frequency range is much faster than the one where this mode acts. As the period of the phugoid oscillation is generally in the hundreds of seconds, this leaves a lot of time for pilots to react and dampen this mode.

Remark about the altitude mode: In English literature, the altitude state is generally dropped for simplicity, neglecting this mode. Nonetheless, the real pole that is obtained by explicitly keeping this state during the linearization is almost always stable and describes the slow return of the flight path to its equilibrium value. It is generally around the same frequency range as the phugoid, leading to similar conclusions about its importance for piloting tasks. In French literature, this mode is called "Rappel de Propulsion". Although

the ISA atmosphere model was not detailed, increases of altitude will result in small dynamics changes. This means that a negligible coupling exists between this state and the others in numerical models.

3.2.2 SP mode approximation

It is clear from the **E** matrix that there only exists a coupling from the lift equation (α line) to the moment equation (q line). Aiming to control only the SP mode, the explicit SP linearized equation is given by $\dot{\mathbf{x}}_{SP} = \mathbf{E}_{SP}^{-1}\mathbf{A}_{SP}\mathbf{x}_{SP} + \mathbf{E}_{SP}^{-1}\mathbf{b}_{SP}\Delta\delta_e$ (the **E** matrix is always full rank in non-hovering flight).

$$\mathbf{x}_{SP} = \begin{bmatrix} \Delta\alpha \\ \Delta q \end{bmatrix} \tag{3.9a}$$

$$\mathbf{b}_{SP} = \begin{bmatrix} \frac{Z_{\delta_e}}{V_{T_e} - Z_{\dot{\alpha}}} \\ \frac{Z_{\delta_e} M_{\dot{\alpha}}}{V_{T_e} - Z_{\dot{\alpha}}} + M_{\delta_e} \end{bmatrix} \tag{3.9b}$$

$$\mathbf{A}_{SP} = \begin{bmatrix} \frac{Z_\alpha}{V_{T_e} - Z_{\dot{\alpha}}} & \frac{V_{T_e} + Z_q}{V_{T_e} - Z_{\dot{\alpha}}} \\ \frac{Z_\alpha M_{\dot{\alpha}}}{V_{T_e} - Z_{\dot{\alpha}}} + M_\alpha + M_{T_\alpha} & \frac{(V_{T_e} + Z_q) M_{\dot{\alpha}}}{V_{T_e} - Z_{\dot{\alpha}}} + M_q \end{bmatrix} \tag{3.9c}$$

$$\mathbf{E}_{SP}^{-1} = \begin{bmatrix} \frac{1}{V_{T_e} - Z_{\dot{\alpha}}} & 0 \\ \frac{M_{\dot{\alpha}}}{V_{T_e} - Z_{\dot{\alpha}}} & 1 \end{bmatrix} \tag{3.9d}$$

In most references, these equations are simplified with $Z_{\dot{\alpha}} \ll V_{T_e}$, $M_{\dot{\alpha}} \approx 0$. The first approximation is used because most stability derivatives are in the order of 10^0, while V_{T_e} should be in the order of 10^2 for commercial aviation. The second approximation seems less realistic, as $M_{\dot{\alpha}}$ is in the order of 10^0, leading to significant contributions in q and α columns. Nonetheless, these simplifications give:

$$\mathbf{A}_{SP} = \begin{bmatrix} \frac{Z_\alpha}{V_{T_e}} & \frac{V_{T_e} + Z_q}{V_{T_e}} \\ M_\alpha + M_{T_\alpha} & M_q \end{bmatrix}, \mathbf{b}_{SP} = \begin{bmatrix} \frac{Z_{\delta_e}}{V_{T_e}} \\ M_{\delta_e} \end{bmatrix} \tag{3.10}$$

As it is sometimes practical to refer to linear dynamics obtained numerically, a simplified notation that makes abstraction of stability derivatives expressions will be used in some portions of this document. This notation simply refers to the content of the **A** and **b** matrices, leaving the choice of the relevant approximations to the designer.

$$\mathbf{A}_{SP} = \begin{bmatrix} a_{11} & 1 \\ a_{21} & a_{22} \end{bmatrix}, \mathbf{b}_{SP} = \begin{bmatrix} 0 \\ b_2 \end{bmatrix} \tag{3.11}$$

This model is obtained from the $\frac{Z_q}{V_{T_e}} \ll 1$ and $Z_{\delta_e} \approx 0$ approximations, which will be shown to be valid in section 3.2.6. From a qualitative point of view, this is based on the approximations of weak C_{L_q} and negligible $C_{L_{\delta_e}}$. Outputs of interest with these approximations are given by:

$$\mathbf{y} = \begin{bmatrix} \Delta q \\ \Delta n_z \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_{11}\frac{V_{T_e}}{g} & 0 \end{bmatrix} \mathbf{x}_{SP} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Delta\delta_e \tag{3.12}$$

This results in the open loop aircraft transfer functions:

$$\frac{\Delta q}{\Delta \delta_e} = \frac{b_2(s - a_{11})}{s^2 - (a_{11} + a_{22})s + a_{11}a_{22} - a_{21}} \equiv K\frac{T_{\theta_2}s + 1}{s^2 + 2\zeta\omega s + \omega^2} \tag{3.13a}$$

$$\frac{\Delta n_z}{\Delta \delta_e} = \frac{-b_2 a_{11}\frac{V_{T_e}}{g}}{s^2 - (a_{11} + a_{22})s + a_{11}a_{22} - a_{21}} \tag{3.13b}$$

If stability derivatives are computed in degrees, an additional factor of $\frac{\pi}{180}$ is needed in the n_z transfer.

3.2.3 Comparison with numerical models of the target aircraft

Numerical models of the target aircraft provided by Airbus contain the state variables $\mathbf{x}_1 = [\Delta\theta, \Delta u, \Delta w, q, \Delta h, \dot{\alpha}_T]^\top$ where $\dot{\alpha}_T$ is a state related to the tail angle of attack due to a travel delay (defined in equation 3.21). This can be changed to $\mathbf{x}_2 = [\Delta\alpha, \Delta q, \Delta V_T, \Delta\theta, \Delta h, \dot{\alpha}_T]^\top$ through a coordinate change.

A transformation matrix ($\mathbf{T}$) can be built to have the desired states as outputs, such that $\mathbf{x}_2 = \mathbf{y} = \mathbf{T}\mathbf{x}_1$. Most of the desired states are already available as outputs in the linearized models, except for $\dot{\alpha}_T$. Although this state can be obtained by a row in $\mathbf{T}$ $[0, 0, 0, 0, 0, 1]$, this could give $K\dot{\alpha}_T$ or any linear combination of this state with others depending on the linearization performed and the units used. The matrix $\mathbf{T}$ is a transformation matrix and is non-singular so long as the outputs chosen are all independent, which is the case here, therefore:

$$\mathbf{A}_2 = \mathbf{T}\mathbf{A}_1\mathbf{T}^{-1} \tag{3.14a}$$

$$\mathbf{B}_2 = \mathbf{T}\mathbf{B}_1 \tag{3.14b}$$

$$\mathbf{C}_2 = \mathbf{C}_1\mathbf{T}^{-1} \tag{3.14c}$$

$$\mathbf{D}_2 = \mathbf{D}_1 \tag{3.14d}$$

In section 3.3.2, the $\dot{\alpha}_T$ pole is always near -600 rad/s, suggesting that this state is negligible. Furthermore, in $\mathbf{E}_{SP}$ given above, the $\dot{\alpha}_T$ dynamic is not even considered and therefore

neglected aside from the coupling from α to q. Nonetheless, direct truncation of this state results in large errors. An approach to obtain a two-pole SP model is presented below based on balanced realization. The coupling expressed through the $\mathbf{E}$ matrix could potentially lead to a "more intuitive" process of reduction if $\dot{\alpha}_T$ was readily available as an output.

3.2.4 SP model reduction through balanced realization

The effect of the $\dot{\alpha}_T$ state can be minimized by performing a balanced realization of the 3-state SP with $\mathbf{x}_{SP,3} = [\Delta\alpha, \Delta q, \dot{\alpha}_T]^\top$. The balanced realization operates a coordinate change on the system where some of the new states will have a more dominant effect on the system's dynamic than others. This allows to truncate the least dominant state obtained. Since the balanced realization process gives $\mathbf{x}_{bal} = \mathbf{T}_{bal}\mathbf{x}_{SP,3}$, the coordinate change used, the whole longitudinal system can be put into the balanced SP states by a coordinate change if desired. This coordinate change corresponds to an identity matrix where the lines and columns related to the SP are replaced by $\mathbf{T}_{bal}$. After the truncation of the least important state, the system can be returned to $\mathbf{x} = [\Delta\alpha, \Delta q, \Delta v_T, \Delta\theta, \Delta h]^\top$ with eq. 3.14a, allowing to have a classical 2-state SP model with minimal effect on its damping and frequency.

3.2.5 Analytical stability derivatives equations

Reference [30] expresses the stability derivatives in function of dimensionless stability derivatives, which are aerodynamic coefficients. Note that all moments are computed at the centre of mass, not the aerodynamic centre. By North American convention, dynamic coefficients (in q and $\dot{\alpha}$) are multiplied by a factor of 2. Numerically, (3.15a) gives more accurate results without C_{D_e}.

$$Z_\alpha = \frac{-\bar{q}S}{m}\left(C_{D_e} + C_{L_\alpha}\right) \qquad C_{L_\alpha} = \frac{\partial C_L}{\partial \alpha} \tag{3.15a}$$

$$Z_{\dot{\alpha}} = \frac{-\bar{q}S\bar{c}}{2mV_{T_e}} C_{L_{\dot{\alpha}}} \qquad C_{L_{\dot{\alpha}}} = \frac{2V_{T_e}}{\bar{c}} \frac{\partial C_L}{\partial \dot{\alpha}} \tag{3.15b}$$

$$Z_q = \frac{-\bar{q}S\bar{c}}{2mV_{T_e}} C_{L_q} \qquad C_{L_q} = \frac{2V_{T_e}}{\bar{c}} \frac{\partial C_L}{\partial q} \tag{3.15c}$$

$$Z_{\delta_e} = \frac{-\bar{q}S}{m}\left(C_{L_{\delta_e}}\right) \qquad C_{L_{\delta_e}} = \frac{\partial C_L}{\partial \delta_e} \tag{3.15d}$$

$$M_\alpha = \frac{\bar{q}S\bar{c}}{J_Y} C_{m_\alpha} \qquad C_{m_\alpha} = \frac{\partial C_m}{\partial \alpha} \tag{3.15e}$$

$$M_{\dot{\alpha}} = \frac{\bar{q}S\bar{c}}{J_Y} \frac{\bar{c}}{2V_{T_e}} C_{m_{\dot{\alpha}}} \qquad C_{m_{\dot{\alpha}}} = \frac{2V_{T_e}}{\bar{c}} \frac{\partial C_m}{\partial \dot{\alpha}} \tag{3.15f}$$

$$M_q = \frac{\bar{q}S\bar{c}}{J_Y} \frac{\bar{c}}{2V_{T_e}} C_{m_q} \qquad C_{m_q} = \frac{2V_{T_e}}{\bar{c}} \frac{\partial C_m}{\partial q} \tag{3.15g}$$

$$M_{\delta_e} = \frac{\bar{q}S\bar{c}}{J_Y} C_{m_{\delta_e}} \qquad C_{m_{\delta_e}} = \frac{\partial C_m}{\partial \delta_e} \tag{3.15h}$$

Although it is not defined in the reference given, M_{T_α} appears in the matrices of section 3.2. It may be calculated by:

$$M_{T_\alpha} = \frac{1}{J_Y} \frac{\partial M_T}{\partial \alpha} \tag{3.16}$$

where M_T is a moment resulting from the engine. For completeness, non-SP stability derivatives are also given:

$$X_{\delta_e} = \frac{-\bar{q}S}{m} C_{D_{\delta_e}} \qquad C_{D_{\delta_e}} = \frac{\partial C_D}{\partial \delta_e} \tag{3.17a}$$

$$X_\alpha = \frac{\bar{q}S}{m}\left(C_{L_e} - C_{D_\alpha}\right) \qquad C_{D_\alpha} = \frac{\partial C_D}{\partial \alpha} \tag{3.17b}$$

$$X_V = \frac{-\bar{q}S}{mV_{T_e}}\left(2C_{D_e} + C_{D_V}\right) \qquad C_{D_V} = V_{T_e} \frac{\partial C_D}{\partial V_T} \tag{3.17c}$$

$$X_{T_V} = \frac{\bar{q}S}{mV_{T_e}}\left(2C_{T_e} + C_{T_V}\right) \qquad C_{T_V} = V_{T_e} \frac{\partial C_T}{\partial V_T} \tag{3.17d}$$

$$Z_V = \frac{-\bar{q}S}{mV_{T_e}}\left(2C_{L_e} + C_{L_V}\right) \qquad C_{L_V} = V_{T_e} \frac{\partial C_L}{\partial V_T} \tag{3.17e}$$

$$M_V = \frac{\bar{q}S\bar{c}}{J_Y V_{T_e}}\left(2C_{m_e} + C_{m_V}\right) \qquad C_{m_V} = V_{T_e} \frac{\partial C_m}{\partial V_T} \tag{3.17f}$$

$$M_{T_V} = \frac{\bar{q}S\bar{c}}{J_Y V_{T_e}}\left(2C_{m_T} + C_{m_{T_V}}\right) \qquad C_{m_{T_V}} = V_{T_e} \frac{\partial C_{m_T}}{\partial V_T} \tag{3.17g}$$

3.2.6 Numerical evaluation of adimensional stability derivatives

Equations (3.15) show that dimensionless stability derivatives may be obtained from partial differentiation of the lift or moment with regards to the quantity of interest. Except for the wing-body C_{m_q}, the aerodynamic model of the target aircraft allows to reconstruct the total C_L or C_m instead of the needed partial derivatives. Two approaches were considered and tested to some extent to compute stability derivatives:

1. Given a function $[C_L, C_m] = f(M, \alpha, \delta_e, ...)$ that computes the total C_m and C_L, this function may be analytically differentiated until the coefficient of interest is expressed in function of the partial derivative of a set of tables of aerodynamic coefficients. The partial derivative of needed tables can be computed in advance, allowing simple look-up.

2. Given the same function, partial differentiation is easily obtained by computing $f(..., x+\epsilon, ...)$ and $f(..., x-\epsilon, ...)$ and performing a centre finite difference.

The first approach is expected to be more efficient when the function $f(...)$ does not depend on the differentiating variable x on many levels. If this is not the case (e.g. in α), the second approach may be of comparable efficiency, on top of allowing much easier troubleshooting. This second approach was used for all computations.

Total lift and drag coefficient of a reconstructed model were validated against numerical data (obtained through SIMULINK) for some linearized conditions with nearly exact matches. Although not available in linearized data, the total moment coefficient (including the engine) was computed and is generally in the order of 10^{-7}, suggesting a valid equilibrium. Note that failure to consider power effects from the engine on the lift and downwash leads to roughly a 1% error on the C_L and C_m.

The error between the linearized data and analytical computation of dimensionless stability derivatives was evaluated for both balanced and direct SP truncation methods. In general, errors on $C_{m_{\delta_e}}$ and C_{m_q} are within a few percents, while errors on C_{m_α} and C_{L_α} can be more significant around stall and at high Mach. This is partially explained by the fact that near stall, both of these coefficients reach small absolute magnitudes and will undergo abrupt evolution. This results in a high sensitivity w.r.t. the step used for the differentiation process. The error on C_{m_α} is given in absolute magnitude for this reason, as this term can be quite close to 0 for high α values, leading to large relative errors. Errors are shown in figure 3.2 for

the direct truncation of the SP model from numerical data[1]. As $C_{m_{\dot{\alpha}}}$ is not directly available from linearized data, this allows for a better comparison. As it is not possible to distinguish between M_α and M_{T_α} in numerical data, coefficients are referred as ca_{ij} or cb_i following the a_{ij} notation defined earlier. To obtain these values, dimensional stability derivatives are normalized by the factor that would be used to dimensionalize the main corresponding aerodynamic coefficient.

As some error patterns can be seen in ca_{12}, ca_{22} and cb_1, this suggests that some terms are neglected in the reconstructed aerodynamic model, although it is not clear where these errors are. Errors on ca_{11} and ca_{21} are much more difficult to analyze beyond what was mentioned earlier, as no clear pattern can be seen, even when plotting the errors in 3D with variables that define the equilibrium (e.g. $\bar{q}$, δ_{stab}, N_1, etc.). Note that viewing errors w.r.t. α suggests the sensibility near stalls already mentioned. To give context to the absolute results, ca_{21} varies between -4 and small positive values in linearized data.

[1]Note that a random bias in Mach was added in these figures following Airbus IP requirements. Nonetheless, this does not affect the analysis performed here.

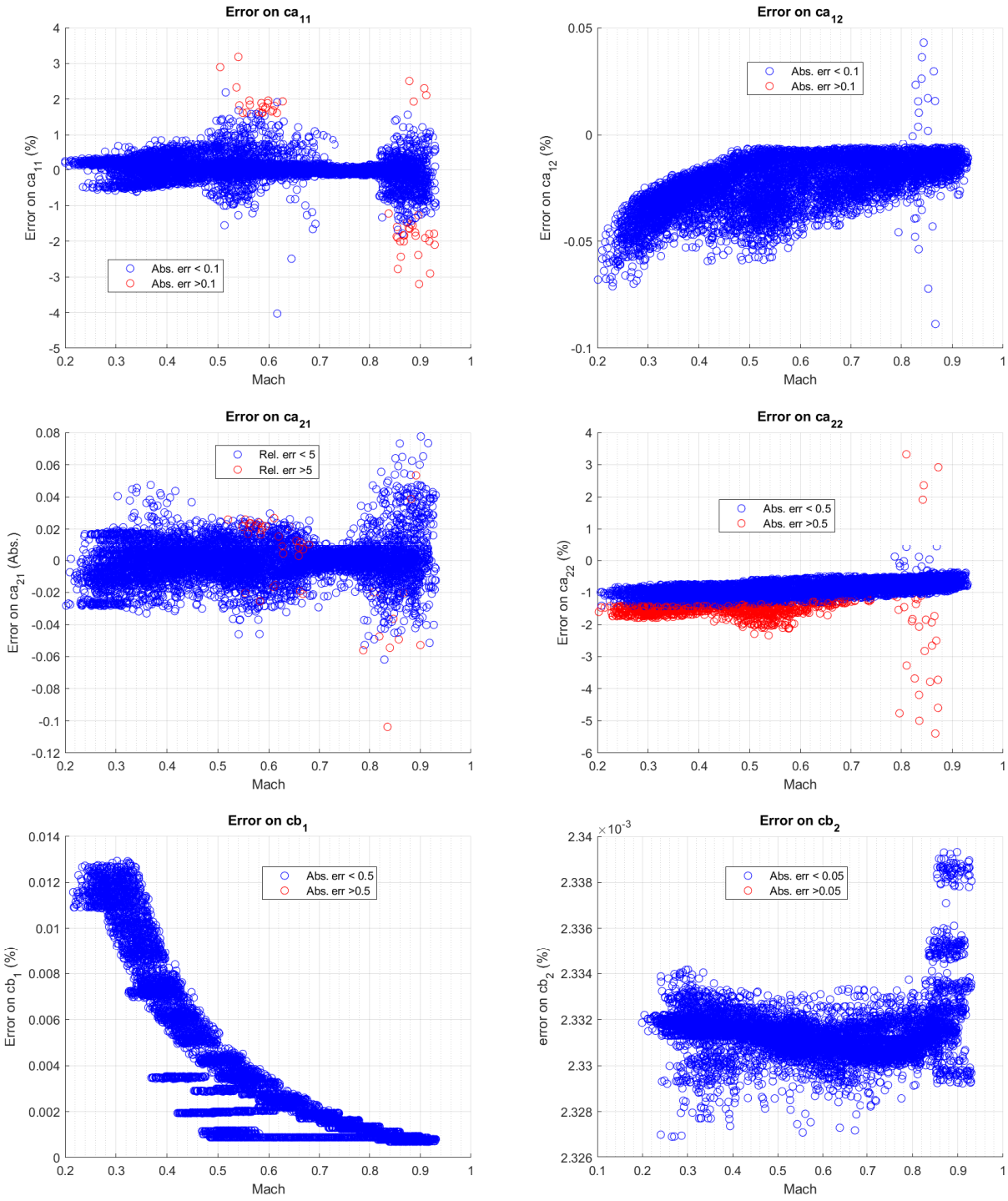


Figure 3.2 Error on dimensionless stability derivatives

3.2.7 Tabulation of adimensional stability derivatives

Using the second approach mentioned above, aerodynamic coefficients (the engine will be discussed later) may be obtained from a function in the form :

$$C_i = f(\bar{x}_{cg}, \bar{z}_{cg}, \bar{q}, V_T, \alpha, M, \delta_{flap}, \delta_{slat}, \delta_{stab}, \delta_e, C_T) \tag{3.18}$$

where the projection of the moment from the aerodynamic centre to the centre of mass is given by:

$$C_m^{cg} = C_m^{AC} + C_X \bar{\Delta}_z - C_Z \bar{\Delta}_x \tag{3.19}$$

$$\bar{\Delta}_x = \frac{\bar{x}_{cg} - 25}{100}, \qquad \bar{\Delta}_z = \frac{\bar{z}_{cg} - \bar{z}_{AC}}{100} \tag{3.20}$$

It is interesting to note that:

- Only moment coefficients are dependent in $\bar{x}_{cg}$. Furthermore, this dependency is linear and given by eq. 3.19. The only exceptions are for derivatives w.r.t. $\dot{\alpha}$ and q, where this value is also used for a dynamic correction of the tail angle of attack due to travel delay (see eq. 3.21).
- Dependency in $\bar{z}_{cg}$ is small. As $|C_X| \ll |C_Z|$, precise estimation of $\bar{z}_{cg}$ is less critical and taking an average value gives good results.
- Dynamic pressure is only used for aeroelastic corrections. Such corrections were found to be mostly negligible at low Mach, allowing to compute stability derivatives for a "high" value (e.g. 300 psf) and avoid tabulation in function of this variable if needed. Another solution could be to take an average for the Mach number range.
- True airspeed is only used for derivatives w.r.t. $\dot{\alpha}$ and q. In both cases, it is required for the dynamic correction of the tail angle of attack due to travel delay. Taking a constant value for table computation yielded interesting results, although this dynamic effect is expected to be inaccurately represented to some extent.
- Elevator input may be set to 0 at equilibrium (trimming is performed by the stabilizer) and is mostly negligible compared to other dependencies, allowing its omission from tables. This will result in errors for off-design points where the elevator deflection is very large.

- Tabulation in (α,M,$\bar{q}$) is interesting, as the couple (M,$\bar{q}$) fully defines the flight point (the corresponding h, V_T are easily found for ISA conditions). Furthermore, interpolation of the remaining parameters in function of these variables and the cg (e.g. $\delta_{stab}(\alpha, \bar{x}_{cg})$, $N_1(\alpha, M)$) over some envelope points allows to approximate equilibrium values for all required parameters. The dependency in δ_{stab} is weak, while power effects related to N_1 account for approximately 1% of the C_L, meaning limited precision for these approximations gives good results. Another solution is to solve the required N_1 and δ_{stab} for an "average" loading. This can be done numerically, as the gradient w.r.t. these parameters is easily computed from the function that computes the coefficients.

Due to the significant distance between the tail and the cg, a pitch rate will induce a local increase in the tail angle of attack following [51]:

$$\Delta\alpha_T = \frac{qL_T}{V_T} \tag{3.21}$$

where L_T is the tail aerodynamic centre to cg distance. As $\dot{\alpha}$ creates a similar effect, this explains the appearance of this state in previous equations. Using equation 3.19 to correct moment coefficients in $\bar{x}_{cg}$, coefficients are expected to be dependent in $(\alpha, M, \delta_{flap}, \delta_{slat}, \delta_{stab})$. For a given $(\delta_{flap}, \delta_{slat})$ condition, numerical inspection of the coefficients allows to identify further simplifications, leading to :

$$C_{L_\alpha} = f(\alpha, M, \bar{q}) \tag{3.22a}$$
$$C^{cg}_{m_\alpha} = f(\alpha, M, \bar{q}, \bar{x}_{cg}) \tag{3.22b}$$
$$C^{cg}_{m_q} = f(M, \bar{x}_{cg}) \tag{3.22c}$$
$$C^{cg}_{m_{\delta_e}} = f(M, \bar{x}_{cg}) \tag{3.22d}$$
$$C^{cg}_{m_{\dot{\alpha}}} = f(M, \bar{x}_{cg}) \tag{3.22e}$$

To simplify the moment tables to three variables or less, the "base" moment given by $C^{AC}_{m_\alpha} + C_{X_\alpha}\bar{\Delta}_z$ is tabulated in a first table for the mean $\bar{\Delta}_z$ value, while C_{Z_α} is also tabulated. This allows reconstruction of M^{cg}_α online from both sets of tables. This results in interpolation from six tables instead of the whole aerodynamic model, significantly reducing the computing power required while maintaining good accuracy. The use of $C^{cg}_{m_{\dot{\alpha}}}$ allows to get results comparable to the balanced realization. For computation of the tables, equilibrium thrust was computed for a grid in $(\alpha, M, \bar{q})$ for an average weight, increasing their accuracy.

Engine ramdrag effect

Although the modelization of engine thrust is independent from α, the engine ramdrag is modeled as :

$$RD = f(N_1, M, h) \tag{3.23a}$$

$$M_y = RD\,(\Delta_x \sin\alpha - \Delta_z \cos\alpha) \tag{3.23b}$$

The norm of the ramdrag is therefore independent from α, leading to :

$$\frac{\partial M_y}{\partial \alpha} = RD\,(\Delta_x \cos\alpha + \Delta_z \sin\alpha) \tag{3.24}$$

which is sufficient for M_{T_α} computation. Note that although the engine thrust is constant with regards to α, it is defined in the body reference frame (C_X and C_Z). This means that it has a small effect ($\approx$1%) on the C_{L_α}.

Adimensional stability derivatives tabulation errors

Figure 3.3 shows errors between the numerical data directly truncated and the computed adimensional stability derivatives from tables[2]. Note that rather than having a table for ca_{12} and cb_1, it is assumed that $a_{12} = 1$ and $b_1 = 0$. Errors on cb_1 are not shown, as they are always 100%.

Coefficients that were well behaved with the complete model are now mostly within $\pm$2% of error, which seems satisfactory. Error on ca_{11} and ca_{21} have increased. Since the main cause for error is believed to be the sensitivity of these coefficients near stall (this can be viewed more easily in 3D), it stands to reason that tabulation would result in larger errors. In the case of ca_{21}, some points have reached significant errors due to the high sensitivity of this coefficient to other parameters near stall. Many coefficients are poorly estimated at high Mach. Visual inspection of the tables for these conditions suggests that transonic effects cause additional sensitivity of the coefficients to parameters that are otherwise negligible. Although this is true for C_{m_α}, this is more apparent for 2D tables ($C_{m_{\dot\alpha}}$, C_{m_q} and $C_{m_{\delta_e}}$), which can undergo large changes depending on the conditions used (e.g. δ_{stab}) to compute the tables.

[2] A random bias in Mach was added in these figures following Airbus IP requirements. Nonetheless, this does not affect the analysis performed here.

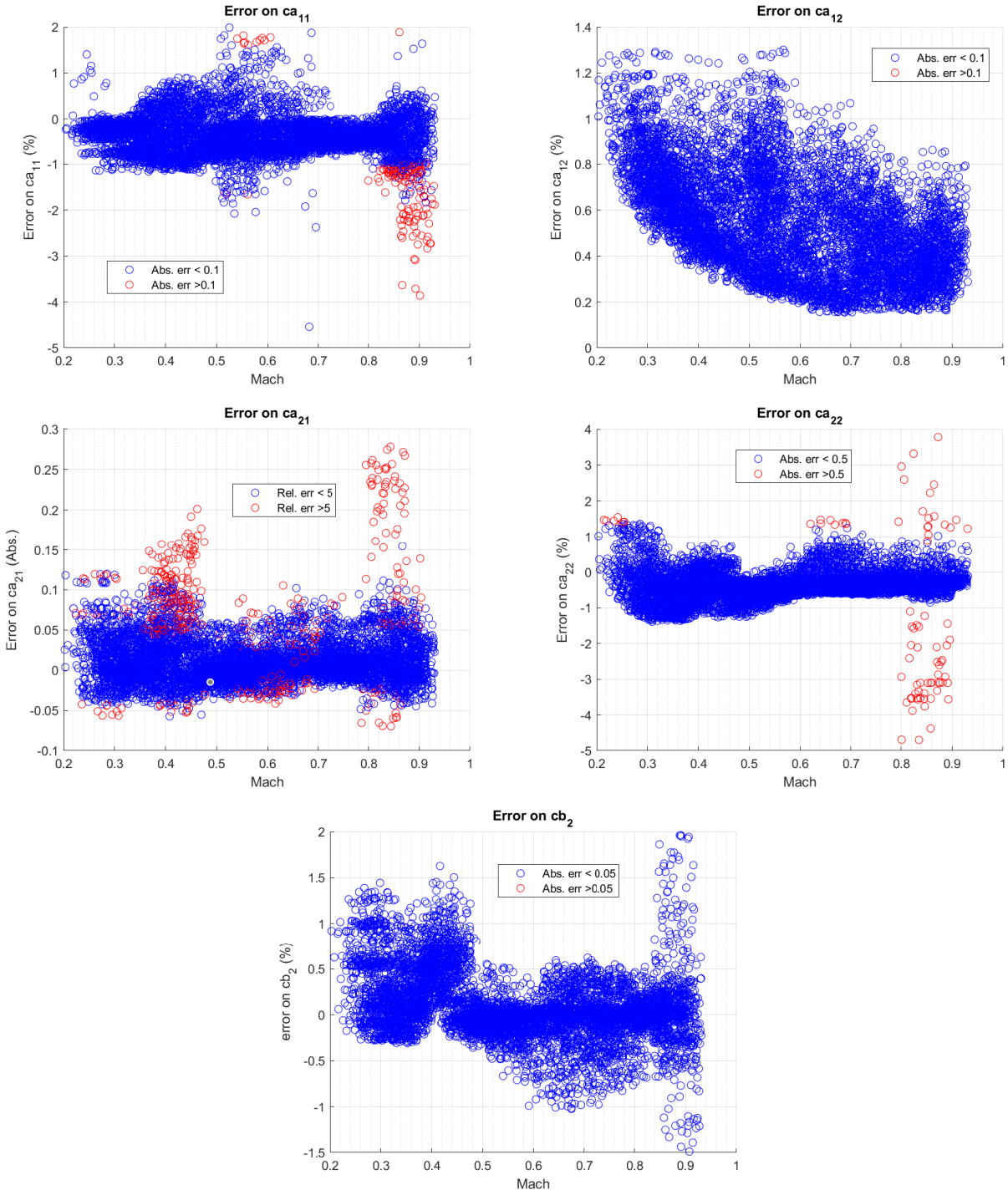


Figure 3.3 Error on tabulated adimensional stability derivatives

3.3 Open-loop analysis

3.3.1 Useful open-loop aircraft expressions

Open-loop transfer functions of the SP developed previously are a good basis for understanding the aircraft. From eq. 3.13, it is possible to express quantities of interest that will be relevant in the following sections.

The time constant of the zero present in the pitch rate response can be evaluated with:

$$T_{\theta_2} = -\frac{1}{a_{11}} \tag{3.25}$$

This quantity will be a primary driver for the dropback. An increase in dynamic pressure or C_{L_α} will decrease the time constant, while mass increments will increase it. As C_{L_α} is independent from the cg position, this remains true for T_{θ_2}.

The gain of the load factor from the $\Delta\alpha$ state will be abbreviated by:

$$N_{n_z} = -\frac{a_{11}V_T}{g} \qquad \text{(and a factor of } \frac{\pi}{180} \text{ if working in degrees)} \tag{3.26}$$

Note that with these approximations, there are no zeros in the load factor response.

The natural frequency of the SP is given by:

$$\omega_{SP} = \sqrt{a_{11}a_{22} - a_{21}} \tag{3.27}$$

In practice, a_{21} is the primary driver for this term, and references often ignore other dependencies [52], [30] for analytical analysis. As such, the open-loop natural frequency is expected to be increased by $|C_{m_\alpha}|$ and dynamic pressure, while it is decreased by the inertia. Note that $|C_{m_\alpha}|$ decreases as the cg moves aft (assuming $C_{m_\alpha} < 0$).

The damping of the SP is given by:

$$\zeta_{SP} = -\frac{a_{11} + a_{22}}{2\omega_{SP}} \tag{3.28}$$

The analysis of this quantity is more delicate, but it can be expected to increase with dynamic pressure, C_{L_α} and $|C_{m_q}|$.

A concise notation of the SP denominator coefficients is obtained with transfer function $\frac{\Delta\alpha}{\Delta\delta_e}$:

$$\frac{\Delta\alpha}{\Delta\delta_e} = \frac{b_2}{s^2 - (a_{11} + a_{22})\, s + a_{11}a_{22} - a_{21}} \equiv \frac{1}{K_2 s^2 + K_1 s + K_0} \tag{3.29a}$$

$$K_0 = \frac{a_{11}a_{22} - a_{21}}{b_2} \tag{3.29b}$$

$$K_1 = \frac{-(a_{11} + a_{22})}{b_2} \tag{3.29c}$$

$$K_2 = \frac{1}{b_2} \tag{3.29d}$$

Writing it in this form is practical for the equivalent output (E) that is used by the G* methodology. Other outputs may be obtained from it by a simple product:

$$\frac{\Delta q}{\Delta\alpha} = s - a_{11} \tag{3.30}$$

$$\frac{\Delta n_z}{\Delta\alpha} = N_{n_z} \tag{3.31}$$

A simple way to obtain $\Delta\gamma$, given by [52] is then:

$$\frac{\Delta\gamma}{\Delta\alpha} = \frac{K_\gamma}{s} \tag{3.32}$$

because $\Delta\gamma = \Delta\theta - \Delta\alpha$:

$$\frac{\Delta\gamma}{\Delta\alpha} = \frac{s - a_{11}}{s} - 1 = -\frac{a_{11}}{s} \tag{3.33}$$

Nonetheless, this approximation has severe limitations, as the SP approximation fails to capture large zeros and slower dynamics which have significant contributions to the flight path response.

Finally, we introduce the term a_0 that appears often with the C* architecture:

$$a_0 = \left(\frac{V_T \pi}{g180} + \beta\right) \tag{3.34}$$

where β is the C* ratio.

Phugoid reduced order models

Although phugoid control is not a primary objective, it remains interesting to evaluate the potential of the methods developed for the control of this mode, hence the need for analytical

models. Although multiple low order models exist in literature, they are generally of low accuracy. The most common phugoid model is the Lanchester approximation [52]:

$$\zeta_{PH} \approx 0, \qquad \omega_{PH} \approx \frac{\sqrt{2}g}{V_{T_e}} \tag{3.35}$$

This approximation can be further improved with [52]:

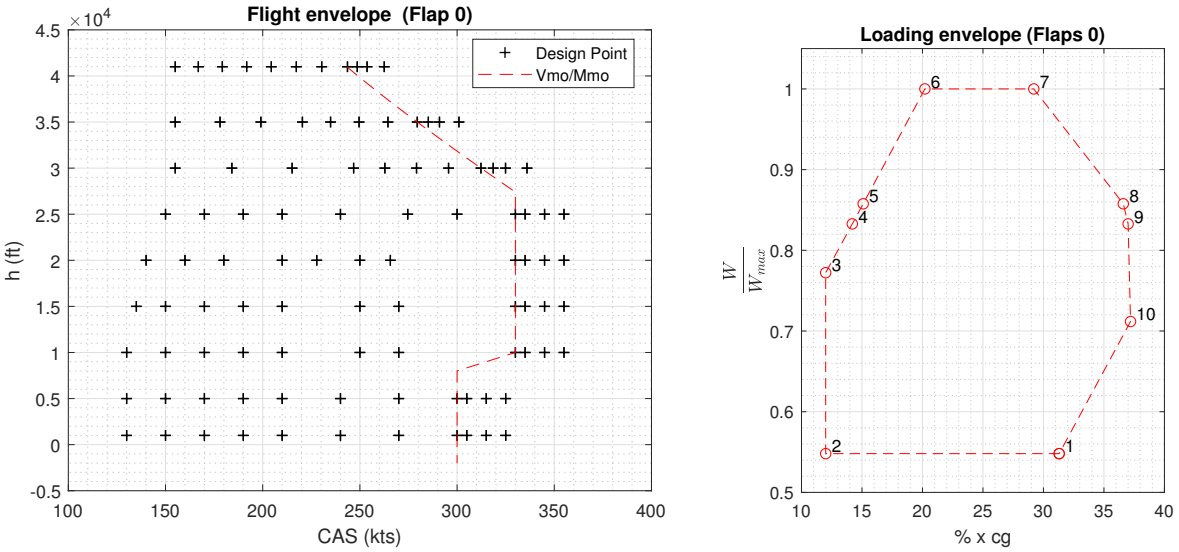
$$\zeta_{PH} \approx \frac{C_D}{\sqrt{2}C_L} \tag{3.36}$$

Nonetheless, this approximation is clearly limited, as this suggests that this mode is never unstable (in the operational envelope). Better approximations exist, including ones based on setting $\dot{\alpha} = \dot{q} = 0$ and isolating the corresponding dynamics in V_T and θ. Still, [30] mentions limitations for the damping prediction.

3.3.2 Open-loop pole evolution

Operational envelope

The flight envelope defined in figure 3.4a defines the points that will be used while studying the aircraft for flap 0. Points used for future synthesis will be a critical subset of these points and synthesis results will be validated on the whole envelope presented here. It should be noted that classical design of the target aircraft was done using a set of points defined in V_{sr} (stall reference) speeds that are not in this figure. Although they were initially used for the synthesis instead of fixed low-speed points, very low-speed points (especially at high altitudes) sometimes fail the linearisation (due to local instability of the C_{m_α}). As V_{sr} speeds vary with aircraft loading, heavy aircraft will have a higher V_{sr} speed than lighter ones. This makes analysis of the results more complicated, especially as V_{sr} speeds cannot be given. As using both types of low-speed points was found to be redundant, only fixed speeds were kept. Between 10,000 and 20,000', V_{Mo} is increased to 330 Kts. Although points could have been added at 300 Kts, this did not seem necessary in early validation, allowing to keep the same number of points as was used for lower altitudes.



(a) Flight envelope

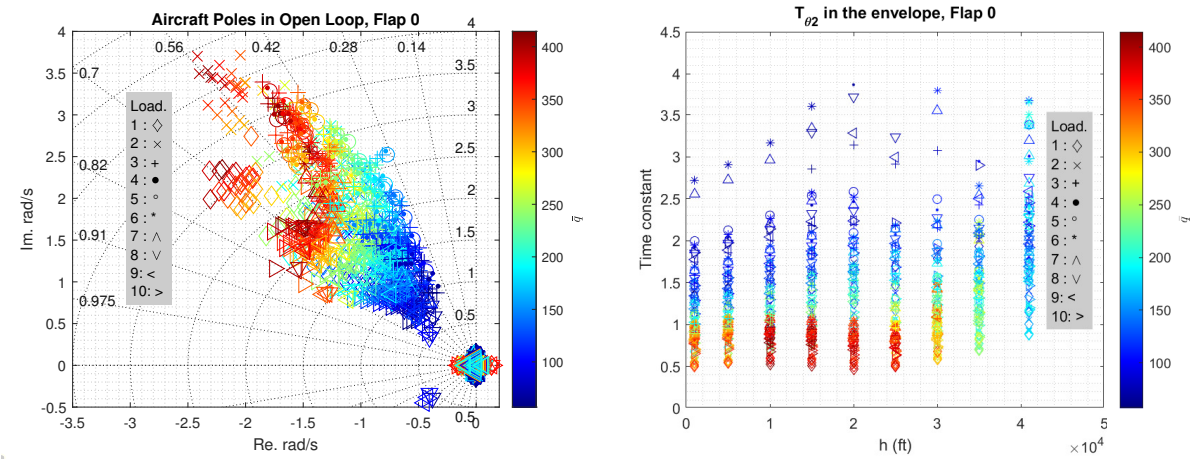
(b) Loading envelope

Figure 3.4 Operational envelope

In the context of the first sub-objective, it will be assumed that the control law designed does not have access to mass or cg estimates. This is why good robustness w.r.t. the loading of the aircraft will be necessary. The loading envelope considered is defined in figure 3.4b. The points presented here are the points that define the outer envelope, which corresponds to worst cases.

Short period evolution

According to the equations developed in the last section, it is expected that SP poles will move towards the left side of the complex plane as the weight of the aircraft decreases and as the dynamic pressure increases. SP poles for the defined envelope are displayed in figure 3.5a, where the colour corresponds to the dynamic pressure (in psf). The analytical equations presented seem to describe well the behaviour of SP poles and T_{θ_2} (see figure 3.5b).



(a) Short period poles

(b) Short period q zero time constant

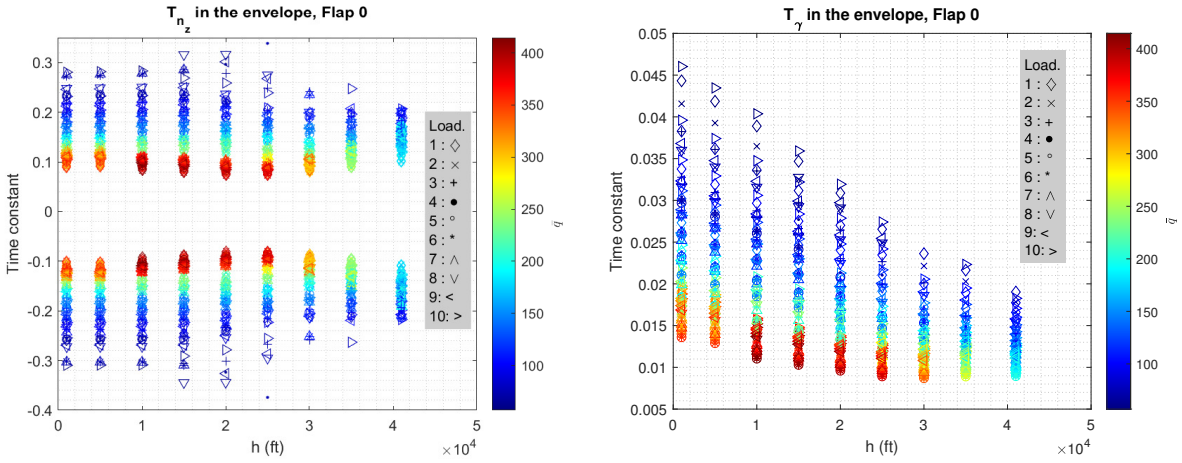
Figure 3.5 Evolution of primary short period dynamics

The simplicity of the approximations in γ and n_z will make them useful design tools later on, but they do not correspond to the results of the numerical evaluation of these transfer functions. Indeed, for γ, figure 3.6 shows a very fast zero. Given the speed of this zero, the approximation that ignores it seems plausible. On the other hand, the n_z transfer function has two moderately slow zeros with roughly the same real part, but opposite signs. To ensure that neglecting these zeros is reasonable, the step error between a transfer function without and with two real zeros of opposite real signs has been developed for an under-damped second-order system (see annex B) and may be bound by:

$$e(t) = \left| L^{-1}\left\{ \frac{{\omega_n}^2}{s}\left(\frac{(\tau s+1)(\tau s-1)}{s^2+2\zeta\omega_n+{\omega_n}^2} + \frac{1}{s^2+2\zeta\omega_n+{\omega_n}^2} \right) \right\} \right| \tag{3.37a}$$

$$\leq \sqrt{1.5}{\omega_n}^2\tau^2 e^{-\zeta\omega_n t} \text{ if } 0<\zeta<1 \tag{3.37b}$$

Given the range of values that ω_{SP} and τ may take, the time response error quickly becomes negligible. Indeed, the main use of the n_z approximation will be to estimate the overshoot and settling time of the n_z transfer. By evaluating the error at the peak time (π/ω_d), the exponential term is below 0.1 for damping ratios greater than 0.6. For an absolute worst case of $T_{n_z} = 0.3$ and $\omega_n = 3$ (which are obtained at low and high speeds, respectively), the error is inferior to 10%.



(a) n_z (b) γ

Figure 3.6 Evolution of other zeros time constants in the envelope

Comments on other aircraft poles

Phugoid poles are much harder to characterize. Aside from being qualitatively slower with weight increases, increasing the speed tends to increase the real parts of the pole for a fixed altitude. Nevertheless, it is difficult to make observations that are not vaguely qualitative. Simple approximations present in literature do not seem to hold. The same applies to the altitude pole. As the C* feedback is not intended to improve the behaviour of these modes (and designer experience suggests that it has little effect on them), characterization to only a qualitative level is deemed enough. Finally, it should be noted that some of these slower modes are unstable in figure 3.5a. The target aircraft has a separate control law to stabilize these modes (not presented), which is designed on a nonlinear model, hence why this part is not incorporated in the controller design.

The pole related to the $\dot{\alpha}$ state does not move significantly ([-606,-600] rad/s) throughout all of the envelope. Since there is also a zero near this pole, it seems possible to remove it if needed.

3.4 Control law architecture and description

The aircraft system considered has 7 main components, shown in figure 3.7. Pilot inputs are done on a side stick that has a given dynamic. The side stick position is sampled and sent to the control laws, which generate an elevator command. The elevator has its dynamic and its movement affects the aircraft. Measurements from sensors are used to estimate the quantities needed for feedback, which are filtered, closing the control loop.

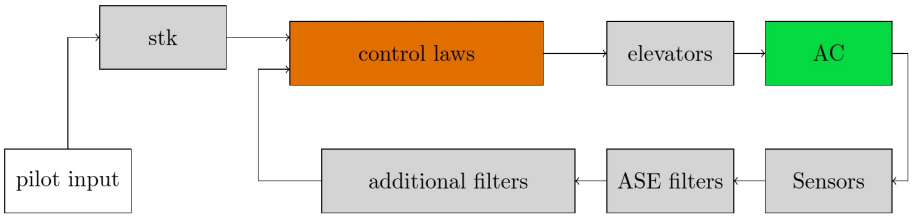


Figure 3.7 Main components of the closed loop system

Figure 3.8 shows the control law that is considered in the present study (longitudinal C* law). First, the stick command is filtered by two first-order lead-lag filters. Envelope protection functions generate a corresponding n_z command. Since these limiting functions vary over the envelope, they are approximated as a constant gain for small amplitude deflections for each design point. The remaining components of the control loop correspond to a more classical C* architecture: a pitch rate feedback to improve the SP damping and a PID with feed-forward for the C* control. It should be noted that there is another lead-lag filter in the pitch rate path on top of a washout filter. When tuning the control law through "classical means", this filter was added to improve the stability margins, especially with flaps. Finally, the pseudo-derivator's time constant (τ_d) is set to 0.1 s.

3.4.1 Fixed components

Components presented in this section are fixed, meaning that their dynamic does not change with the flight condition, loading, etc. Space-state representations or transfer functions presented in this section were obtained from a linearization tool based on the `linmod` `MATLAB` function. Information on these components is summarized in table 3.1, where U.A. refers to an information that cannot be disclosed (unavailable).

Table 3.1 Fixed dynamics summary

Component	Delay (ms)	Dynamic
Flight Control System	U.A.	-
Stick	≈1	Eq. 3.38
Elevator	U.A.	Eq. 3.39
Sensors	≈20	Eq. 3.40
ASE filters	U.A.	U.A.
Additional filtering	-	Eq. 3.41a and 3.41a

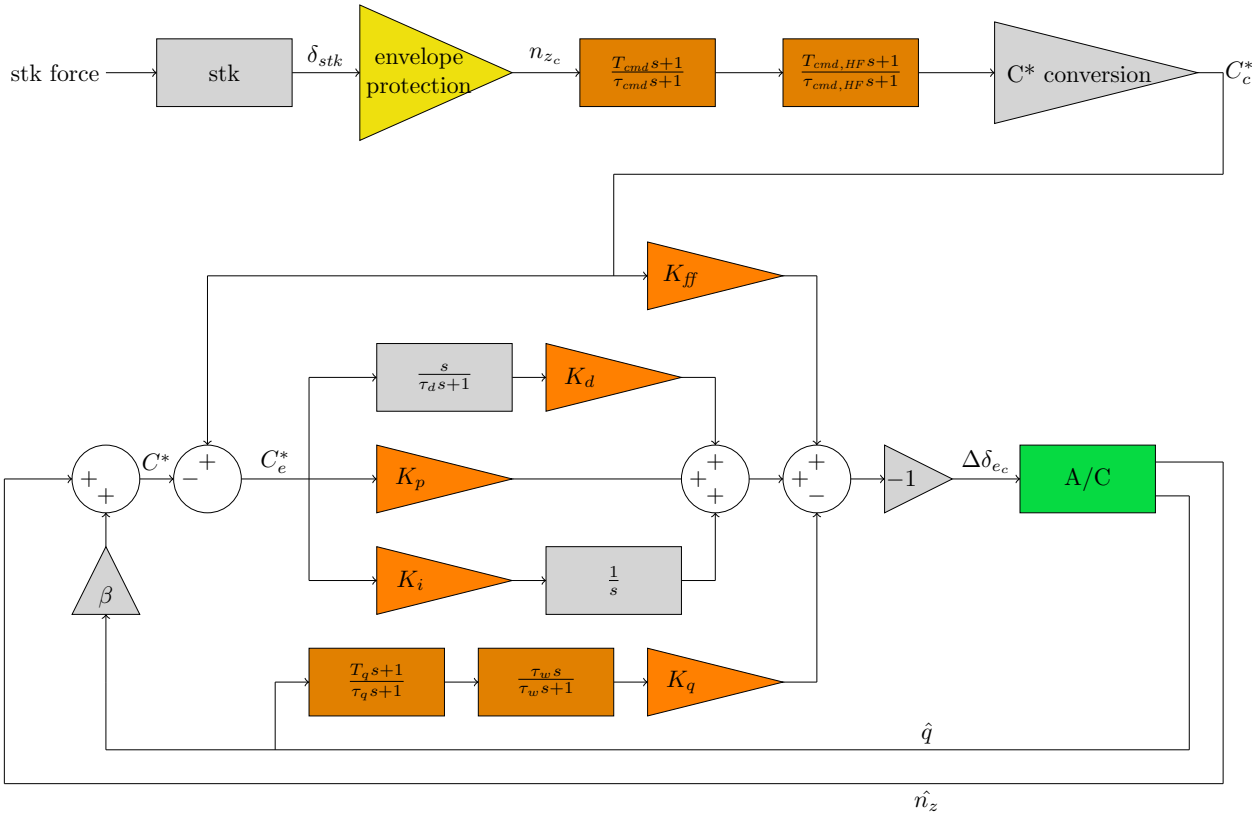


Figure 3.8 Control architecture in C^*

Stick transfer function:

$$\frac{\delta_{stk}(^\circ)}{Force(LBS)} = \frac{7.0884 \cdot 10^9(s^2 - 640s + 1.28 \cdot 10^5)}{(s+291)(s+367.4)(s+101.8)(s^2+53.05s+868.5)(s^2+429s+1.056 \cdot 10^5)} \tag{3.38}$$

Elevator transfer function:

$$\frac{\delta_e}{\delta_{e_c}} = \frac{5220}{s^2 + 103.2s + 5220} \tag{3.39}$$

Sensor dynamics:

$$\frac{n_{z,m}}{n_z} = \frac{q_m}{q} = \frac{7.9423 \cdot 10^5(s^2 - 343.3s + 3.684 \cdot 10^4)}{(s+156.1)(s^2+111.1s+6169)(s^2+230.1s+3.038 \cdot 10^4)} \tag{3.40}$$

Additional acceleration filtering:

$$\frac{n_{z,f_2}}{n_{z,f}} = \frac{10}{(s+10)} \tag{3.41a}$$

$$\frac{n_{z,f_2}}{q_f} = \frac{-3.489s}{(s+10)(s+20)} \tag{3.41b}$$

where the m suffix is used for measured quantities, and the f suffix is used for filtered quantities. Numeric suffixes on f, e.g. $n_{z_{f_2}}$ are used for additional filtering steps after ASE filters.

Aeroservoelastic (ASE) filters ensure that flexible modes do not affect signals used by the controller, avoiding excitation of these modes. No information can be given on this component or on the aircraft's flexible modes. For reference, a NASA flutter analysis of the DC-10 places the slowest mode at 3 Hz ($\approx$ 19 rad/s) [53]. As aircraft structural optimization has improved since and that many manufacturers turn towards composite materials, modern flexible modes can be expected to be slower than in this study. A modern numerical analysis of a generic model made to represent aircraft similar to Boeing 737-200/300 or Embraer ERJ 190/195 obtained flexible modes ranging between 4 and 40 rad/s, depending on the flexibility level used [54]. Although the article does not deal with an industrial aircraft model, it still allows to evaluate the order of magnitude in which flexible modes can be expected.

Finally, it is interesting to note that the elevator has a maximum deflection rate, which can be used for OLOP computation. Nonetheless, this value cannot be disclosed.

3.5 Closed-loop analysis

3.5.1 Useful closed-loop aircraft description

From the longitudinal control architecture shown in section 3.4, the control loop can be separated into two distinct sections. The stick, the envelope protection functions and the two lead-lag filters can be seen as command shaping components, as they act on the C* command. The remaining components (feedback and feedforward gains) will be referred to as the C* tracking loop, as they should ensure that the aircraft follows the desired C* command. These two sections are in series, simplifying their analysis. Before studying how the poles evolve in closed-loop, it is interesting to study the proprieties of the closed-loop aircraft from the point of view of some handling qualities through analytical tools. As will be explained in section 5.3, the dropback and load factor overshoot will be at the centre of the design with the G* methodology. To study them, the command shaping portion

will be ignored to focus on the C* tracking loop, reducing the order of transfer functions. All feedback dynamics and delays will also be ignored for the same reason, along with the derivative gain. Although the role of K_d will only be explained in detail in section 5.2.4, for now, it is sufficient to say that this gain's main use is to limit the degradation of high-order modes, making it irrelevant for the low-order approximation desired here. Furthermore, if a pure derivator is considered, this gain adds no degree of freedom for SP pole placement, justifying the previous comment. All of these simplifications lead to:

$$\frac{\Delta\alpha}{C_c^*} = -\frac{\left(\left(K_{ff}+K_p\right)s+K_i\right)}{\left(K_2s^2+K_1s+K_0\right)s-\left(K_q\left(s-a_{11}\right)s+\left(K_ps+K_i\right)\left(\beta s+N_{n_z}-\beta a_{11}\right)\right)} \tag{3.42}$$

By analogy, an alternative representation for this transfer function is:

$$\frac{\Delta\alpha}{C_c^*} = K\frac{\omega_{CL}^3\gamma\zeta_{CL}\left(T_{K_{ff}}s+1\right)}{\left(s^2+2\zeta_{CL}\omega_{CL}+\omega_{CL}^2\right)\left(s+\gamma\zeta_{CL}\omega_{CL}\right)} \tag{3.43}$$

where $T_{K_{ff}}$ is the feedforward zero and γ represents the ratio of the integrator pole over the real part of the closed-loop short period.

Closed-loop dropback

From (2.4a) and by multiplying (3.43) by $(T_{\theta_2}s+1)$, computation of the dropback is straightforward:

$$Drb = T_{K_{ff}} + T_{\theta_2} - \frac{2\zeta_{CL}+\frac{1}{\gamma\zeta_{CL}}}{\omega_{CL}} \tag{3.44}$$

Closed-loop dropback from stability derivatives

Although the previous equation is practical, it does not show how the stability derivatives affect the dropback. Developing the denominator of eq. 3.42 in powers of s leads to:

$$\frac{q}{C_c^*} = \tag{3.45}$$
$$-\frac{s^2(K_{ff}+K_p)+s(K_i-a_{11}\{K_{ff}+K_p\})-K_ia_{11}}{s^3K_2+s^2(K_1-K_q+K_p\beta)+s(K_0+K_qa_{11}-K_p[N_{n_z}-\beta a_{11}]-K_i\beta)-K_i(N_{n_z}-\beta a_{11})}$$

From (2.4a), the dropback may be expressed as:

$$Drb = \frac{N_1}{N_0} - \frac{D_1}{D_0} = \frac{K_i + a_{11}(K_p + K_{ff})}{K_i a_{11}} + \frac{K_0 + K_q a_{11} - K_p(N_{n_z} - \beta a_{11}) - K_i\beta}{K_i(N_{n_z} - \beta a_{11})} \tag{3.46a}$$

$$= \frac{-1}{a_{11}} + \frac{K_{ff} + K_p}{K_i} + \frac{K_0 + K_q a_{11}}{K_i(N_{n_z} - \beta a_{11})} - \frac{K_p}{K_i} - \frac{\beta}{N_{n_z} - \beta a_{11}} \tag{3.46b}$$

Substituting the expressions of N_{n_z} and K_0 :

$$\begin{aligned} Drb &= \frac{-1}{a_{11}} + \frac{K_{ff} + K_p}{K_i} - \frac{\frac{a_{11}a_{22} - a_{21}}{b_2} + K_q a_{11}}{K_i a_{11}(\frac{V_T\pi}{g180} + \beta)} + \frac{\beta}{a_{11}(\frac{V_T\pi}{g180} + \beta)} - \frac{K_p}{K_i} \\ &= \frac{-a_{22}}{b_2(\frac{V_T\pi}{g180} + \beta)K_i} + \frac{1}{a_{11}}\left(\frac{\beta}{(\frac{V_T\pi}{g180} + \beta)} - 1 + \frac{a_{21}}{b_2(\frac{V_T\pi}{g180} + \beta)K_i}\right) + \\ &\frac{K_{ff} + K_p}{K_i} - \frac{K_q}{K_i(\frac{V_T\pi}{g180} + \beta)} - \frac{K_p}{K_i} \end{aligned} \tag{3.47a}$$

Substituting a_0 (see eq. 3.34), which is constant for a given flight point :

$$Drb = \frac{-a_{22}}{b_2 a_0 K_i} + \frac{1}{a_{11}}\left(\frac{\beta}{a_0} - 1 + \frac{a_{21}}{b_2 a_0 K_i}\right) + \frac{K_{ff} + K_p}{K_i} - \frac{K_q}{K_i a_0} - \frac{K_p}{K_i} \tag{3.48}$$

It is clear that the loading conditions that result in the worst cases of Drb will be the same for all gains if K_i is constant. Substituting only the part defined from aerodynamic quantities by the adimensional stability derivatives :

$$\begin{aligned} Drb &= \frac{-\frac{\bar{q}S\bar{c}^2}{2J_y V_T}C_{m_q}}{\frac{\bar{q}S\bar{c}}{J_y}C_{m_{\delta_e}} a_0 K_i} + \frac{mV_T}{-\bar{q}SC_{L_\alpha}}\left(\frac{\beta}{a_0} - 1 + \frac{\frac{\bar{q}S\bar{c}}{J_y}C_{m_\alpha}}{\frac{\bar{q}S\bar{c}}{J_y}C_{m_{\delta_e}} a_0 K_i}\right) + ... \\ &= -\frac{\bar{c}C_{m_q}}{2V_T C_{m_{\delta_e}} a_0 K_i} - \frac{mV_T}{\bar{q}SC_{L_\alpha}}\left(\frac{\beta}{a_0} - 1 + \frac{C_{m_\alpha}}{C_{m_{\delta_e}} a_0 K_i}\right) + ... \end{aligned} \tag{3.49}$$

This expression can be maximised or minimised for each flight condition to get worst cases for constant gains w.r.t. loading. Considering the partial derivatives w.r.t. the four stability derivatives, one obtains :

$$\frac{\partial Drb}{\partial C_{m_\alpha}} = -\frac{mV_T}{\bar{q}SC_{L_\alpha}a_0K_iC_{m_{\delta_e}}} > 0 \tag{3.50a}$$

$$\frac{\partial Drb}{\partial C_{m_q}} = -\frac{\bar{c}}{2V_Ta_0C_{m_{\delta_e}}K_i} > 0 \tag{3.50b}$$

$$\frac{\partial Drb}{\partial C_{m_{\delta_e}}} = \frac{\bar{c}C_{m_q}}{2V_Ta_0K_iC_{m_{\delta_e}}^2} + \frac{mV_TC_{m_\alpha}}{\bar{q}SC_{L_\alpha}a_0K_iC_{m_{\delta_e}}^2} < 0 \tag{3.50c}$$

$$\frac{\partial Drb}{\partial C_{L_\alpha}} = \frac{mV_T}{\bar{q}Sa_0C_{L_\alpha}^2}\left(\beta - a_0 + \frac{C_{m_\alpha}}{K_iC_{m_{\delta_e}}}\right) \tag{3.50d}$$

Within the operational envelope, $C_{L_\alpha} > 0$, $C_{m_\alpha} < 0$, $C_{m_q} < 0$, $C_{m_{\delta_e}} < 0$ and $K_i > 0$, hence the sign of the first three equations. The sign of the last equation is easily found with $\text{sign}\{\beta - a_0 + \frac{C_{m_\alpha}}{K_iC_{m_{\delta_e}}}\}$. As these signs are constant, maximum Drb always occurs at :

$$\begin{cases} \text{Max.}C_{L_\alpha} & \text{if} \quad \text{sign}\{\beta - a_0 + \frac{C_{m_\alpha}}{K_iC_{m_{\delta_e}}}\} > 0 \\ \text{Min.}C_{L_\alpha} & \text{if} \quad \text{sign}\{\beta - a_0 + \frac{C_{m_\alpha}}{K_iC_{m_{\delta_e}}}\} < 0 \end{cases} \tag{3.51a}$$

$$\text{Max.} \quad C_{m_\alpha} \tag{3.51b}$$

$$\text{Max.} \quad C_{m_q} \tag{3.51c}$$

$$\text{Min.} \quad C_{m_{\delta_e}} \tag{3.51d}$$

This has been tested through the envelope and the approximation of the reference model is valid. Although the four coefficients are treated as variables, in practice C_{m_q} and $C_{m_{\delta_e}}$ are nearly independent from the cg position. It has been mentioned that the largest component of C_{m_q} comes from the elevator, meaning that in both cases the cg position is essentially negligible compared to the elevator to aerodynamic centre distance. Furthermore, C_{L_α} is independent from the cg position. This all means that C_{m_α} is the main contributor to the dropback and that maximum values are reached for aft cg.

Closed-loop load factor overshoot equation

The n_z transfer function from eq. 3.43 may be decomposed into partial fractions for a step input :

$$\frac{\gamma\zeta\omega^3(T_{K_{ff}}s + 1)}{s(s + \gamma\zeta\omega)(s^2 + 2\zeta\omega s + \omega^2)} = \frac{A}{s} + \frac{Bs + D}{s^2 + 2\zeta\omega s + \omega^2} + \frac{E}{s + \gamma\zeta\omega} \tag{3.52}$$

The steady-state gain of the transfer is normalized to 1, as it will not affect the overshoot. Solving the resulting system of equations (development is in annex A) yields:

$$E = \frac{T_{K_{ff}}\gamma\zeta\omega - 1}{\gamma^2\zeta^2 - 2\gamma\zeta^2 + 1} \tag{3.53a}$$

$$B = -E - 1 \qquad A = 1 \tag{3.53b}$$

$$D = -\left(2\zeta\omega + \zeta\omega(2-\gamma)E\right) \tag{3.53c}$$

This gives the time response (assuming $0 < \zeta < 1$):

$$y(t) = 1 + \left(B\cos\omega_d t + \frac{D - B\zeta\omega}{\omega_d}\sin\omega_d t\right)e^{-\zeta\omega t} + Ee^{-\gamma\zeta\omega t} \tag{3.54}$$

Overshoots are usually computed by maximizing $y(t)$, which requires to solve $\dot{y}(t) = 0$. In this case, $\dot{y}(t) = 0$ seems difficult to solve analytically :

$$\dot{y}(t) = e^{-\zeta\omega t}\left(\cos\omega_d t\left[D - 2B\zeta\omega\right] + \sin\omega_d t\left[-B\omega_d - \zeta\omega\frac{D - B\zeta\omega}{\omega_d}\right]\right) - \gamma\zeta\omega E e^{-\gamma\zeta\omega t} \tag{3.55}$$

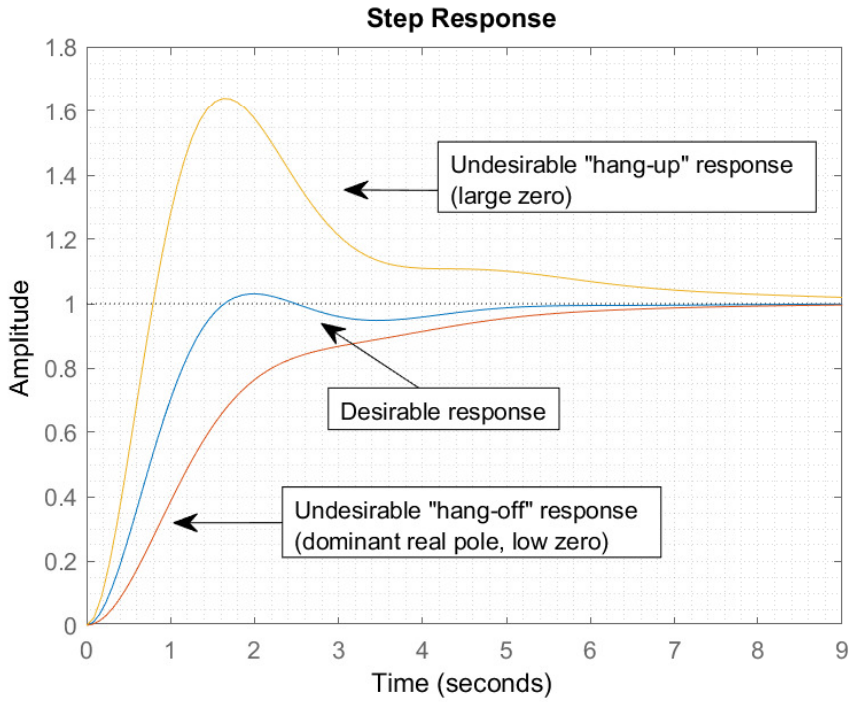


Figure 3.9 3-pole SP n_z response types

The fast exponential term is generally dominant for responses that have a "desirable" second-order behavior for n_z (see figure 3.9). This means that an approximate peak time may be computed by solving :

$$0 = \cos\omega_d t\left[D - 2B\zeta\omega\right] + \sin\omega_d t\left[-B\omega_d - \zeta\omega\frac{D - B\zeta\omega}{\omega_d}\right] \tag{3.56}$$

this leads to :

$$t_{peak} = \frac{\arctan\left(-\frac{D-2B\zeta\omega}{-B\omega_d - \zeta\omega\frac{D-B\zeta\omega}{\omega_d}}\right) + \pi}{\omega_d} \tag{3.57}$$

where the second positive solution is selected (periodic in π). In most cases, $|\arctan(...)| \ll \pi$, leading to the classical second order peak time (π/ω_d). This result is not surprising given the assumption that the system's response is "close" to a second-order system. This remains a very good approximation for systems with the desired qualitative behavior (see figure 3.10, which was generated for $\omega = 1$, $\zeta = 0.6$, $\gamma = 0.6$, $T_{K_{ff}} = 2.52$). Furthermore, this removes dependency in ω, except for B, D, E. Evaluating $y(t_{peak})$ gives :

$$y(t_{peak}) = 1 + e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}(1 + E) + Ee^{\frac{-\gamma\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{3.58}$$

Assuming $\gamma = 0.6$ and $\zeta = 0.6$, the overshoot is given by :

$$OS = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}(1 + E) + Ee^{\frac{-\gamma\zeta\pi}{\sqrt{1-\zeta^2}}} \approx 0.0948 + 0.3380E \tag{3.59}$$

For 5% overshoot (for example) :

$$0.05 = 0.0948 + 0.3380\frac{T_{K_{ff}}\gamma\zeta\omega - 1}{\gamma^2\zeta^2 - 2\gamma\zeta^2 + 1} \tag{3.60}$$

$$T_{K_{ff}}\omega = \psi = 2.521 \tag{3.61}$$

Although the full equation is quite complex, the dependency w.r.t. ω_{CL} and $T_{K_{ff}}$ can be mitigated through the parameter ψ. This will be an essential design tool in the design with the G* methodology in chapter 5. As a final remark, numerical evaluation of overshoots for multiple ω values suggests that this is a very good approximation in general, as the overshoot with ψ is always constant for a given ζ and γ. It is suspected that the general solution to the peak time is divided by ω, meaning it will always be dependent on $1/\omega$. This doubt is reinforced by the results of the next section.

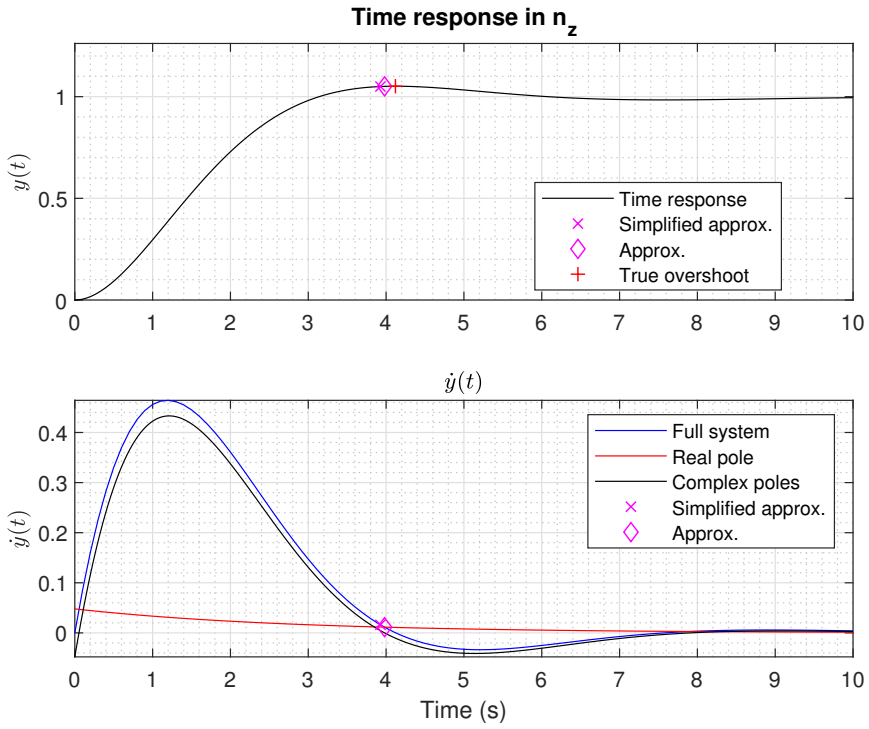


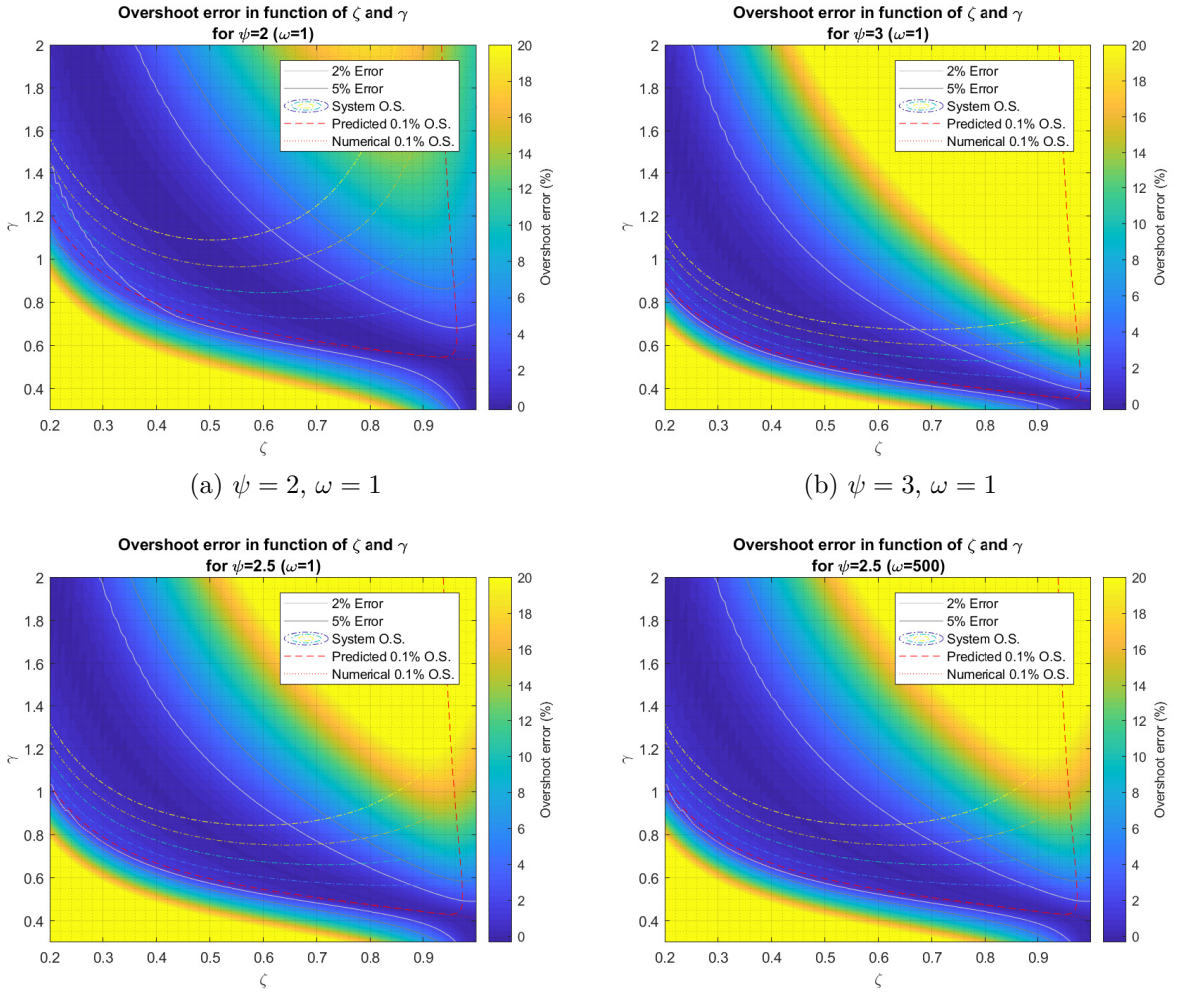
Figure 3.10 Precision of the $\dot{y} = 0$ approximation

To evaluate the precision of this approximation, figure 3.11 allows to compare the overshoot error (real-approximation) in function of ζ and γ, for values of ψ that were useful during design. This evaluation is done for $\omega = 1$ and $\omega = 500$, to show that this parameter has no effect, despite the incapacity to find the analytical solution. Note that there are large errors when the system does not overshoot because the approximation predicts negative overshoots, due to the ignored slower dynamics.

An effective way to study these figures is to look at the 10% system overshoot line and at the range of validity (ζ, γ) w.r.t. the 2% overshoot error, which is considered a "tolerable" precision. The approximation keeps a good precision for $\zeta < 0.75$. This can also be repeated for a higher system overshoot and the 5% error line. Note that significant errors are reached near $\zeta \approx 0.95$, where the assumption of an oscillatory response becomes less valid.

Closed-loop load factor overshoot worst cases without loading scheduling

Along with the dropback, the n_z overshoot is of critical importance for design. On the other hand, there is no practical equation that allows to do a similar development to study the effect of stability derivatives. Indeed, using the equation developed above would require



(a) $\psi = 2$, $\omega = 1$

(b) $\psi = 3$, $\omega = 1$

(c) $\psi = 2.5$, $\omega = 1$

(d) $\psi = 2.5$, $\omega = 500$

Figure 3.11 Precision of the overshoot approximation

the resolution of the zeros of a third-order polynomial. Although this was attempted, the resulting equations are extremely impractical, leading to no useful results.

Nonetheless, looking at HQs in function of the loading envelope with the target aircraft, worst case load factor overshoots are obtained for aft cg positions throughout the envelope (both on the linear and nonlinear model). This tendency is also used for aircraft validation on the nonlinear model. This is "intuitive" as aircraft are generally more responsive at aft cg. Nonetheless, this tendency will be used extensively for the robust G* design, hence the need for a more concrete proof.

Starting from eq. 3.42, the n_z response may be expressed as:

$$\frac{n_z}{C_c^*} = \tag{3.62}$$
$$-\frac{N_{n_z}\left(\left(K_{\not f}+K_p\right)s+K_i\right)}{s^3K_2+s^2(K_1-K_q+K_p\beta)+s(K_0+K_qa_{11}-K_p\left(N_{n_z}-\beta a_{11}\right)-K_i\beta)-K_i(N_{n_z}-\beta a_{11})}$$

which may be generalized to:

$$\frac{K_{n_z}\left(T_{K_{\not f}}s+1\right)}{s^3+c_2s^2+c_1s^1+c_0} \tag{3.63}$$

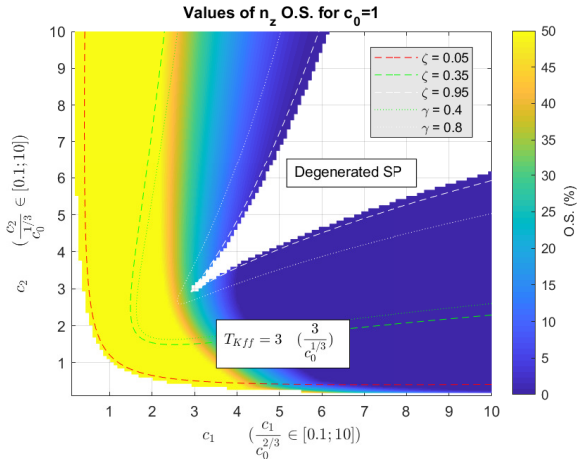
It is important to note that for a given flight condition, gain values are fixed, meaning $T_{K_{\not f}}$ and c_0 are constant. The value of M_α will be a large contributor to K_0 (multiple references consider ω_{SP} to be primarily driven by C_{m_α} as mentioned). The dependency of C_{L_α} w.r.t. the cg is null, while C_{m_q} and $C_{m_{\delta_e}}$ are weakly dependent on the cg. Aiming to study the effect of the *cg* position for a constant mass (by necessity) and assuming the cg position's contribution to the inertia is weak, it is possible to simplify the effect of a change of cg to a reduction of c_1. As $c_1 = \frac{(K_0+K_qa_{11}-K_p[N_{n_z}-\beta a_{11}]-K_i\beta)}{K_2}$, one can deduce:

$$\frac{K_0}{K_2} = a_{11}a_{22} - M_\alpha(cg) \approx d_1 - \frac{\bar{q}S\bar{c}}{J_y}\left(C_{m_\alpha}^{AC} - \bar{\Delta}_xC_{Z_\alpha}\right) \approx d_2 + \frac{\bar{q}S\bar{c}}{J_y}\bar{\Delta}_xC_{Z_\alpha} \tag{3.64}$$

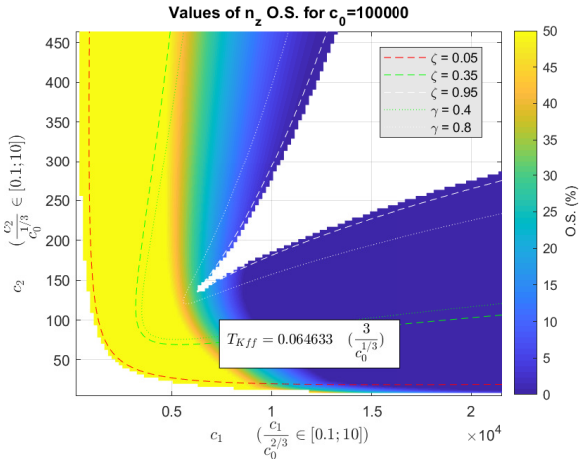
where d_i is a constant. Therefore, aft cg imply a smaller value of c_1 (larger $\bar{\Delta}_x$ and $C_{Z_\alpha} < 0$).

Studying the values of n_z overshoot in function of c_2 and c_1 for a given c_0 results in figure 3.12 (the color legend of the *OS* is limited to 50% for visibility). Although changing the value of c_0 will also change the values of c_1 and c_2 needed to get a given overshoot, the qualitative behavior of the figure does not change. Increasing $T_{K_{\not f}}$ for a given c_0 will increase the overshoot. To show this, figures were generated for both of these modifications. Displaying values of c_1 normalized by factors of $1/{c_0}^{2/3}$, c_2 by factors of $1/{c_0}^{1/3}$ and $T_{K_{\not f}}$ by factors of $1/{c_0}^{1/3}$ removes the dependency in ω (developing the denominator of eq. 3.43 and dividing the terms by the lower order one shows this for c_2 and c_1, while for $T_{K_{\not f}}$, this is based on a constant ψ value). From these figures, it seems that the behavior is not dependent on the c_0 value.

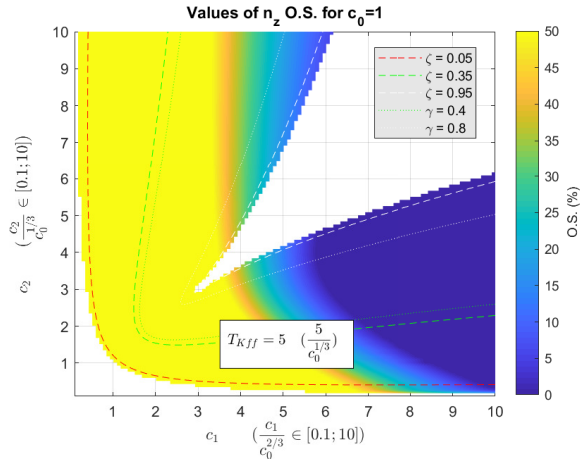
In these figures, the reduction of c_1 is the main contributor to large overshoots (especially when there is adequate damping), meaning that reducing this value always leads to larger overshoots. Therefore, maximum overshoots are achieved at aft cg positions. Furthermore, in some regions, c_2 reductions also lead to larger overshoots. This suggests that overshoots are likely to be worst for lower inertia or mass values.



(a) Reference



(b) Large c_0



(c) Large $T_{K_{ff}}$

Figure 3.12 Overshoot distribution (in %)

3.5.2 Closed-loop pole evolution

To identify how each component evolves in closed-loop, pole identification was done on a single flight condition. The intent is to gain a qualitative understanding of the system. Loading 3 was chosen to do this analysis, although a short verification was done on other points to ensure no significant changes to the root locus occurred. To identify which pole belonged to which component, a gain was added to the elevator command. Varying this gain from 0 to 1 allowed to plot a root locus, describing the open to closed-loop pole evolution. It should be noted that the gains used are those obtained through classical means for the target aircraft. Figure 3.13 shows different zooms of the complex plane.

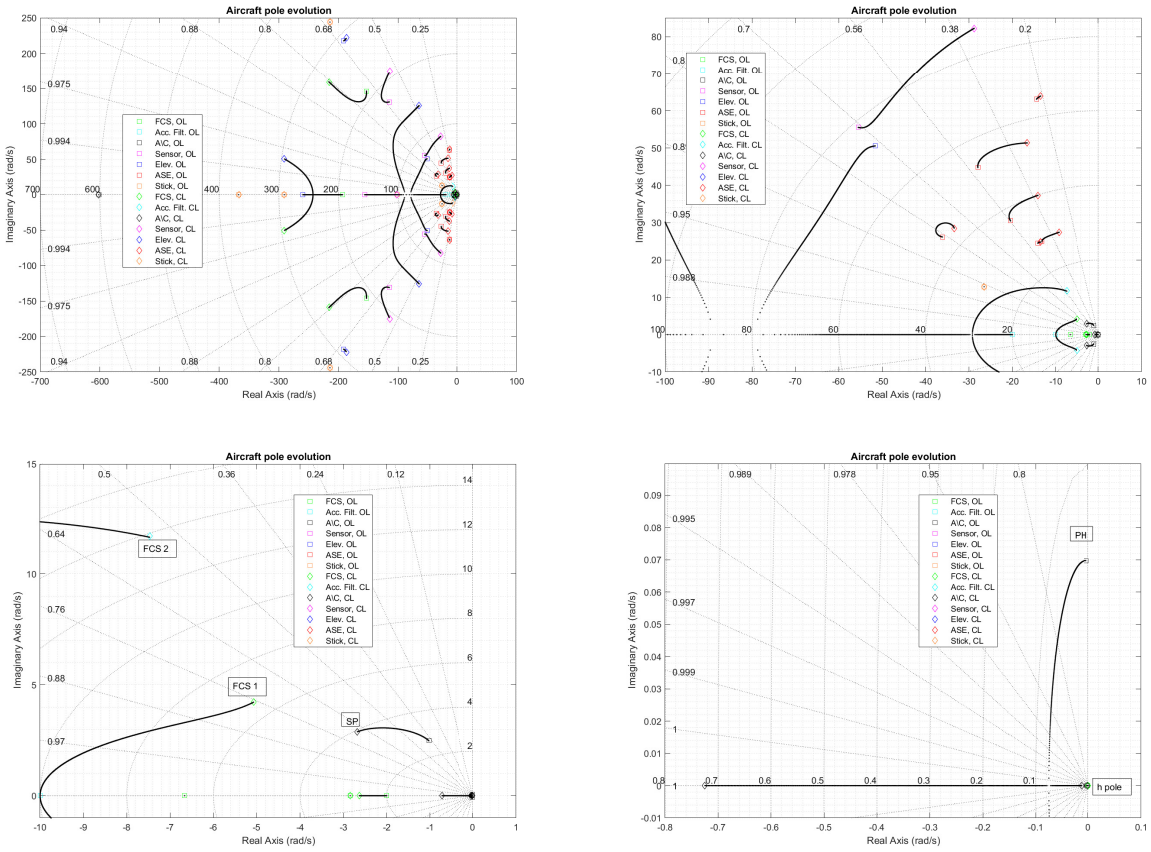


Figure 3.13 Pole evolution from closed to open loop

Most poles faster than 60 rad/s stay fast compared to the closed-loop SP and have a general tendency to lose damping. While this cannot be seen in these figures, these poles did not change much from one condition to another, which is not surprising, as these poles belong to fixed components. This statement also stays true for sensor poles and PCU poles. The fast aircraft pole belongs to the $\dot{\alpha}$ state. This pole does not move noticeably from open loop to closed loop. Poles below 60 rad/s, but faster than 5 rad/s, tend to have a decreased real part in closed-loop. This is especially true for ASE poles. Two complex modes are created from the combination of acceleration filtering poles with other poles. As could be expected, the SP mode gains in speed and damping. Finally, the PH mode degenerates into two real poles, one of which is quite slow, while the altitude pole comes closer to the complex axis. It should be noted that the aircraft has another control loop (not considered) to improve the phugoid damping and ensure it does not degenerate into unstable poles with an unacceptable real part.

3.6 Conclusion

In this chapter, the linearized aircraft model has been expressed in function of known aerodynamic quantities. The structure of these quantities has been explored throughout the flight envelope, allowing to identify key simplifications for their tabulation. Afterwards, the evolution of aircraft poles and zeros has been studied throughout the envelope, giving bounds on principal system characteristics that define the short period response. The structure of the control loop has also been defined, along with its components, allowing to study how closed-loop poles evolve for existing gains on a single flight condition. Finally, equations were developed to study the dropback and load factor overshoot, which will be key to the development of the G* methodology in chapter 5. Now that aircraft dynamics are well understood, it is time develop methods to control the longitudinal behavior of the aircraft.

CHAPTER 4 STRUCTURED H_∞ SYNTHESIS

In this chapter, the methodology based upon the structured H_∞ synthesis will be presented. As mentioned in the introduction, the highly iterative and time consuming process that comes with classical gain scheduling approaches is a large drawback for designers. The intent is therefore to reduce the design time of gain schedules for existing architectures through structured H_∞ synthesis. First, a short introduction to the H_∞ synthesis framework in MATLAB is given. Then, a simplified methodology will be defined to meet handling qualities on one flight condition and loading at a time (process abbreviated by "local" synthesis), removing scheduling and robustness problems. As stability margin and pole damping requirements exist in the MATLAB framework, the main challenge will be to meet the desired handling qualities, which will be resolved through reference model matching. Once good local HQs can be achieved, the full methodology will be introduced. Although heavily based on the local methodology, the main methodology will have additional degrees of freedom in the reference model to address the robustness problem. Finally, badly behaving gain surfaces are discussed, along with additional practical considerations such as initial conditions and convergence problems. Application of the methodology to the flap 0 configuration of the target aircraft is detailed in the chapter 6. The resulting gains and handling qualities will also be shown in this chapter along with the results of the G* methodology from chapter 5.

4.1 Introduction to structured H_∞ synthesis in MATLAB

When performing structured H_∞ synthesis in MATLAB, the synthesis follows these steps:

1. Define the model on which the synthesis will be performed. The `realp` function allows to define real tuneable variables which can be used along with traditional LTI functions (`ss`, `tf`, `connect`, etc.) to define the full closed-loop system. Alternatively, SIMULINK can also be used for this step.

2. Define synthesis objectives through the `TuningGoal` functions. The relevant constraints are described briefly in table 4.1. Existing functions allow to define H_∞ or even H_2 constraints on the desired transfers of the model.

3. Call the `systune` function with the closed-loop system and the constraints in arguments.

It is the `systune` function that performs the non-smooth optimization process required to reach local optimums of structured H_∞ problems [55]. This function can take two types of

optimization objectives: hard constraints and soft constraints (hard and soft requirements will be noted by $\mathbf{g}(\mathbf{x})$ and $\mathbf{f}(\mathbf{x})$ as in MATLAB documentation). For now, it is sufficient to say that the synthesis will attempt to strictly respect hard constraints before trying to get the best results w.r.t. soft constraints. Greater details on constraints types will be given in section 4.5.6.

Table 4.1 Useful MATLAB H_∞ constraints

Function name	**Description**
`TuningGoal.Poles`	Constraints on the closed-loop poles (ζ_{min},ω_{min},ω_{max}).
`TuningGoal.Margins`	Minimum stability margins at the specified location, enforced as an ellipse margin delimited by a minimum gain and phase margin.
`TuningGoal.WeightedGain`	Ensures $\|\|\mathbf{W}_L(s)\mathbf{H}(s)\mathbf{W}_R(s)\|\|_\infty<1$, where $\mathbf{H}(s)$ is a transfer specified.

4.2 Local synthesis

4.2.1 Definition of a reference model

Classical H_∞ problems [56], [57] are often formulated with some of the following constraints, to which secondary requirements may be added :

- Stability requirements
- Performance requirements
- High frequency penalization of the command
- Disturbance rejection

From the design requirements, it is clear that stability margins and that some form of command penalization (which should help meeting the OLOP criterion) are needed. The main question is therefore how to meet the remaining handling qualities. Some HQs could be enforced directly through H_∞ constraints with `TuningGoal` functions. Nonetheless, bandwidths and other criteria defined from the phase response seem difficult to express directly with H_∞ constraints, hence the need to use a reference model and penalize low frequency tracking errors. Type 1 PIO criteria are intended to limit high-order dynamics degradations, meaning that enforcing low-order responses should yield good results. Indeed, literature also

contains optimization routines which penalize "poor" fits to low-order models [24]. Note that there are other alternatives to meet performance specifications without using a reference model. A common approach is the shaping of the sensitivity and complementary sensitivity functions [27], [58]. The main problem with such methodologies is that few articles consider all relevant types of the design requirements, especially those that are defined by phase behaviors. From those that do, [25] explores a different approach named "matrix experiments" to define synthesis weights through an iterative process. The reference model approach was found to be more intuitive, hence why it was chosen.

The reference model should be as simple as possible, making it easy to select its parameters. On the other hand, it must be a realistic target for the system, which may impose additional complexity. This section describes the process that led to the reference model, justifying its relative complexity.

Closed-loop transfer function

Developing the full closed-loop transfer function of the aircraft will help identifying the structure of the reference model. Equations 4.1 to 4.3 show the transfer function for the architecture presented in figure 3.8:

$$\frac{C_c^*}{\delta_{stk}} = F_{stk}(s) K_{prot} \frac{(T_{HF,cmd}s + 1)}{(\tau_{HF,cmd}s + 1)} \frac{(T_{cmd}s + 1)}{(\tau_{cmd}s + 1)} \tag{4.1}$$

$$\frac{\delta_{e,c}}{C_c^*} = \frac{(K_i + (Kp + K_{ff})(0.1s^2 + s) + K_d s^2)\, D_q D_m}{(0.1s^2 + s)\,(D_q D_m + K_q \tau_w s(T_q s + 1)) + (K_i + K_p(0.1s^2 + s) + K_d s^2)\, N_{C^*} D_q} \tag{4.2}$$

where :

- $F_{stk}(s)$ is the stick transfer function
- K_{prot} is the gain resulting from the protection functions linearization
- The c subscript denotes the command
- τ_w is the washout filter time constant
- T_q and τ_q are the q lead-lag numerator and denominator time constants.

- $D_q = (\tau_w s + 1)(\tau_q s + 1)$, the q feedback denominator
- $\frac{y_f}{\delta_{e,c}} = \frac{N_y}{D_m}$, the command to measured and filtered output transfer
- $N_{C^*} = (\beta N_q + N_{n_z})$, the C* zeros

Therefore :

$$\frac{q}{\delta_{stk}} = \frac{q}{\delta_{e,c}} \frac{\delta_{e,c}}{C_c^*} \frac{C_c^*}{\delta_{stk}} \tag{4.3}$$

where $\frac{q}{\delta_{e,c}}$ is the transfer from the elevator command up to the aircraft (real) pitch rate. This result will still apply to other outputs. From these equations, two important conclusions may be inferred :

- Closed-loop zeros consist of open-loop aircraft zeros and those added by the command lead-lag filters, PID and feedforward.
- Closed-loop poles are made of the command filter poles, stick poles and the poles resulting from the C* feedback.

These results are in agreement with the previous separation of the closed loop into command filtering ($\frac{C_c^*}{\delta_{stk}}$) and C* tracking ($\frac{q}{C_c^*}$) sections.

Remarks on the C* reference model choice

A first guess for the C* tracking reference model could be to keep a pair of complex poles (closed-loop short period) and a single zero, effectively mimicking the open-loop transfer function of the aircraft:

$$\frac{q}{C_c^*} = \frac{K{\omega_n}^2(Ts + 1)}{(s^2 + 2\zeta\omega_n s + {\omega_n}^2)} \tag{4.4}$$

While this configuration is simple, it is not optimal for the system. Compared to eq. 3.43, it is clear that the integrator pole is neglected, meaning that the synthesis is likely to converge to a very high K_i gain, making this pole negligible and degrading stability margins. Another possibility is that the integrator pole could be cancelled by the feed-forward zero, keeping T_{θ_2} and resulting in an under-constrained problem. Note that the cancellation of T_{θ_2} creates an exceedingly slow response in n_z when the zero is slow. Therefore, the reference model in q should keep T_{θ_2}, include a new zero (to use K_{ff}) and a new real pole (the integrator pole). For H_∞ synthesis, this pole is placed to the same real part as the closed-loop SP poles as this gives good results and remains simple. As a command filter is required to meet HQs, varying γ (integrator to SP real part ratio) would have added another parameter and additional complexity to the methodology.

The last consideration of interest is the gain of the C* tracking reference model, which is critical for reference matching. Looking at equation 3.42, the steady state gain is:

$$\frac{\Delta\alpha}{C_c^*}(0) = \frac{K_i}{K_i\left(N_{n_z} - \beta a_{11}\right)} = \frac{-1}{a_{11}a_0} \tag{4.5}$$

$$\frac{q}{C_c^*}(0) = \frac{-a_{11}}{-a_{11}a_0} = \frac{1}{a_0} \tag{4.6}$$

This is representative of the full system, as neglected components (sensor, filters and elevator dynamics) should have unit gains. For ease of computation, one could instead use $\frac{\Delta\alpha}{C_c^*}(0) = \frac{1}{\beta\frac{q}{\delta_{e,c}}(0)+\frac{n_z}{\delta_{e,c}}(0)}$. This gain implies that the closed-loop will have a 1 C* steady-state gain, due to the integrator. As a final note, the reference for H_∞ synthesis was developed in q, as most HQs are computed from this output. Nonetheless, other outputs could be chosen.

Reference model used

Following the previous remarks, the final q reference model is :

$$\frac{q_{ref}}{\delta_{stk}} = \frac{e^{-T_{eq}s}(T_{\theta_2}s+1)(T_{K_{ff}}s+1){\omega_n}^2(T_{cmd}s+1)}{a_0(s^2+2\zeta\omega_n s+{\omega_n}^2)(\frac{s}{\zeta\omega_n}+1)(\tau_{cmd}s+1)}F_{stk}(s)K_{prot} \tag{4.7}$$

where T_{eq} is greater than the sum of the delays in the control loop, allowing to less penalize the rise time of the system. Taking the delay into account is essential since the synthesis would otherwise be limited by high-frequency errors. The stick transfer function has been kept according to the closed-loop transfer function. Only one command filter has been kept, since having two lead-lags in series is detrimental to the convergence of the synthesis (additional detail will be given in section 4.2.3). Having one lead-lag filter is necessary since this allows to increase the "dropback margin", which will be defined in section 4.2.1, giving better bandwidths. Furthermore, fast-forwarding to G* conclusions, phase-lag is essential to avoid HQ degradations at very low or high speeds.

Alternative reference model when *PRS* limited

It is important to keep in mind that the reference model introduced in section 4.2.1 is one option within many possibilities. This reference model was used successfully for clean configurations, although the BW_γ was often low. A synthesis in a landing configuration was done to ensure this did not prove problematic in cases where this metric was crucial. The problematic HQs with this synthesis were the *PRS*, Gibson criteria and OLOP. With this first reference model, it was essentially impossible to respect the *PRS* for low damping ratios

of the reference model. While increasing the damping ratio improved the results, it also increased command use, degrading the OLOP. As the *PRS* is measured at the phase bandwidth, increasing the damping causes a smoother phase transition, increasing the bandwidth and reducing the corresponding gain. This is because the SP is generally around the same frequency range as the phase bandwidth. To solve the *PRS* problem, a 3 state model was used. The third pole is called τ_{cmd} to keep a direct analogy with the methodology that will be developed, but in practice, it corresponds to the integrator pole. The alternative reference model for conditions limited by the upper *PRS* limit is given by:

$$\frac{q_{ref}}{\delta_{stk}} = \frac{e^{-T_{eq}s}(T_{\theta_2}s+1)(T_{K_{ff}}s+1){\omega_n}^2(T_{cmd}s+1)}{a_0(s^2+2\zeta\omega_n s+{\omega_n}^2)(\tau_{cmd}s+1)}F_{stk}(s)K_{prot} \tag{4.8}$$

Manually choosing parameters for this new reference model is more challenging, although this will not be a problem for the scheduled synthesis. Furthermore, this reference model is more dependent on the damping ratio, which needs to be changed slightly from one condition to another. This is problematic for the methodology that will be introduced in section 4.4.1 if this variation is too large. Nonetheless, for slow flight conditions limited by the maximum *PRS* limit, this reference model allows to reach better results and was preferable for the landing configuration studied.

Reference model parameters tuning for local synthesis

While the presented reference models result in realistic behaviors for the aircraft, they are fairly complex. The methodology defined below allows to select good values for $T_{K_{ff}}, T_{cmd}$, $\omega_n, \zeta, \tau_{cmd}$ and obtain desirable handling qualities with relative ease. Although the selection of reference model parameters will no longer require manual tuning when the controller is scheduled, it remains necessary to discuss the local methodology and the process that led to the final approach. Because of the transfer function's complexity, this process is iterative. It is described in algorithm 1.

Algorithm 1 Selection of the reference model's parameters for local synthesis

1: Impose ζ (this parameter has a minor role in q, the overshoot being largely defined by the zeros).
2: Choose ω_n and τ_{cmd} such that transfer function 4.9 meets the desired settling time ($T_{s,des}$).
3: Choose T_{cmd} and $T_{K_{ff}}$ simultaneously such that they meet the desired *Drb* while maximizing the bandwidths.
4: If results are not as intended, return to step 1.

$$\frac{Y(s)}{U(s)} = \frac{(T_{\theta_2}s+1)}{(s^2+2\zeta\omega_n s+{\omega_n}^2)(\frac{s}{\zeta\omega_n}+1)}\frac{\omega_n^2}{(\tau_{cmd}s+1)} \tag{4.9}$$

Analytical solutions for step 2 would be difficult to obtain. Indeed, the time response for this transfer function was developed but attempts to isolate the settling time did not result in any useful results. Instead, it is possible to evaluate the output $y(t)$ of the transfer function for the desired settling time for a grid of parameters. This can be done from a numerical partial fraction decomposition and remains reasonably fast. It is then possible to choose τ_{cmd} such that $|1-y(T_{s,des})| < \epsilon$ for a unit step, while minimizing the value of ω_n (hopefully minimizing commands). In this equation, the quantity ϵ is the convergence criterion of the settling time. Although there is no settling time requirement, this metric remains very intuitive for designers and can easily be used to meet other HQs. For reference, [59] gives guidelines for satisfactory settling times (maximum of 4.4 s within 10%).

Finally, for step 3, it is possible to find the phase variation due to the two zeros. For a given dropback target :

$$Drb_{tgt} - T_{\theta_2} + \frac{2\zeta+\frac{1}{\zeta}}{\omega_n} + Drb_{stk} + \tau_{cmd} = M_{Drb} = T_{K_{ff}} + T_{cmd} \tag{4.10}$$

where M_{Drb} is the aforementioned dropback margin, since it limits how large the zero time constants may be chosen. If this quantity is negative, the time constants will be negative as Drb_{tgt} is too aggressive. The phase variation due to the zeros is :

$$\Delta\phi(\omega) = \arctan(T_{K_{ff}}\omega) + \arctan((M_{Drb}-T_{K_{ff}})\omega) \tag{4.11}$$

To maximize the phase increase (and the resulting bandwidths) in function of $T_{K_{ff}}$:

$$\frac{\partial\Delta\phi(\omega)}{\partial T_{K_{ff}}} = \frac{\omega}{1+(T_{K_{ff}}\omega)^2} - \frac{\omega}{1+((M_{Drb}-T_{K_{ff}})\omega)^2} = 0 \tag{4.12}$$

A solution of this equation is $T_{K_{ff}} = T_{cmd} = \frac{M_{Drb}}{2}$. The second derivative is given by:

$$\frac{\partial^2\Delta\phi(\omega)}{\partial T_{K_{ff}}^2} = \frac{2\omega^2 T_{K_{ff}}}{(1+(T_{K_{ff}}\omega)^2)^2} + \frac{2\omega^2 T_{K_{ff}}}{(1+((M_{Drb}-T_{K_{ff}})\omega)^2)^2} > 0 \tag{4.13}$$

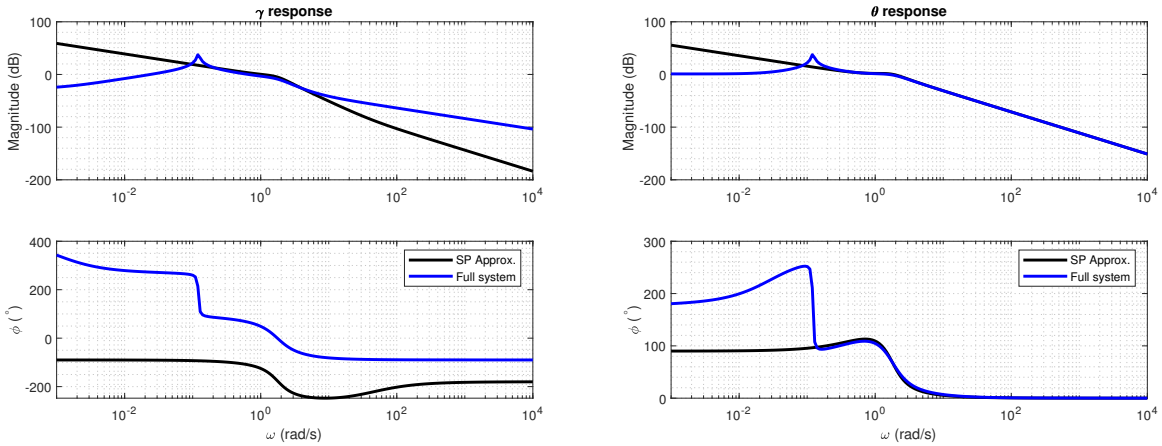
meaning that the solution found is a maximum ($T_{K_{ff}} > 0$). From 4.10, it can be seen that the pole of the lead-lag command filter adds to the Drb margin. Indeed, for very slow speed conditions, it would be difficult to satisfy the bandwidths requirements without large dropbacks

when the command filter is not used. This equation also shows that T_{θ_2} limits the available performance (bandwidths) for a given Drb.

This process allows to meet requirements relatively easily, except for BW_γ, which will be discussed in the next section. Although the reference model only considers the q output, other outputs are only different due to open-loop zeros or steady-state gains. Therefore, as long as the synthesis is successful and the q reference is adequate, handling qualities for other outputs will also be satisfactory. As the reference model has been defined for the short period model, the last question is whether this model is representative of the full order model in the frequency range of interest. Ideally, imposing good bandwidths on the reduced model should result in good bandwidths on the full model. This question is assessed in the next section and is related to difficulties with the BW_γ.

Remarks for BW_γ and validity of the SP model for bandwidths

Figure 4.1 shows an overlay of the full and short period models of the aircraft, in open-loop, for γ and θ (only transfers on which full model requirements apply).



(a) γ (b) θ

Figure 4.1 Precision of the SP approximation in pitch and flight path

In both cases, the SP approximations are of the third order, as the relevant SP model has been integrated (divided by s). This figure was generated for a single flight condition and loading, but the behaviour is similar for most design points. For θ, the short period approximation is very representative of the full model between [0.1, 100] rad/s, which is not

surprising, since this model has been conceived to study most of the faster aircraft responses. On the other hand, the phase γ response is badly captured by the SP model as this approximation neglects slow zeros. Although there is a phase offset at high frequencies, this offset is not constant as the frequency is reduced. The phase response (and gain response to a lesser extent) are therefore quite approximative within the frequency range related to BW_γ.

It should be noted that, while a synthesis only on the full model would be possible, this results in degraded SP handling qualities to an unacceptable extent. Furthermore, it is not possible to match both the SP approximation and the full system to their respective reference model at once. Indeed, while the SP model is not representative of the full order flight path response, enforcing a response in q will still enforce behaviour in γ. To achieve good BW_γ, one must choose a relatively low ζ for the reference model. This can be justified by the fact that since the BW_γ is usually slower than 1 rad/s, the abrupt phase transition for the SP poles (around 2-3 rad/s) due to low damping ratios is beneficial for this bandwidth.

4.2.2 H_∞ constraints for the local synthesis

As explained in section 4.2.1, three main types of constraints are needed :

Constraint 1: A stability requirement to enforce good stability margins. For most conditions, these are gain margins of 6 dB and phase margins of 45°.

Constraint 2: A reference tracking requirement. Good tracking is required at low frequencies. As the fastest design criterion is the phase delay, which is affected by the phase at $2\omega_{180}$, an upper bound of 20 rad/s gives good results, leading to the weighting function W_1 (4.14a). `RelGap` is a factor that allows to tighten the tracking required. The name of this parameter is kept following the MATLAB `TuningGoal.StepTracking` propriety [60]. The normalization by the reference's steady-state gain is needed to obtain a similar tracking precision at low and high speeds with the same `RelGap` value. This is due to the C* architecture, as the steady-state gain in q is higher at low speeds.

Constraint 3: A requirement to limit high-frequency command use. Since low-frequency command use will be largely determined by the tracking requirement, the command will be penalized slightly before the cut-off frequency of W_1, leading to the equation 4.14b for W_2. `Ponderation` allows to define how much the command should be penalized (a value of 1 was found to be extremely strict, while 2 or 3 is more reasonable and usually respects the OLOP criteria).

$$W_1 = \frac{20^2}{(s+20)^2}\frac{1}{Ref(0)\texttt{RelGap}} \tag{4.14a}$$

$$W_2 = \frac{5s}{(s+10)\texttt{Ponderation}} \tag{4.14b}$$

By putting both tracking and stability margins as hard requirements, very small values of `RelGap` may result in margins not being respected when the synthesis is unable to satisfy $\mathbf{g}(\mathbf{x}) < 1$, which is not intended. On the other hand, putting the tracking requirement as a soft constraint is not possible either as `systune` always aims for $\mathbf{g}(\mathbf{x}) = 1$, resulting in always being on the margins limits. This problem is avoided by a hard tracking requirement where `RelGap` is a tunable parameter that must be minimized through a soft constraint, as in figure 4.2 and table 4.2 . This results in an "artificial" transfer for the synthesis. Components of functional diagrams will be colored as follows:

- Tunable components
- Non-tunable components
- Non-tunable components that are approximated
- Known/approximated aerodynamics
- Actual/numerically linearized aerodynamics
- Transfers containing tuneable values (H_∞)
- H_∞ constraints and I/O

4.2.3 Architecture modifications to improve synthesis convergence

H_∞ synthesis, like most numerical optimization methods, is sensible with regard to problems that are under-constrained. In the case of structured synthesis, under-constrained paths are

Table 4.2 Local synthesis constraints

Objective	Constraint Type	Model	Weight
Stability Margins	Hard	Full	`TuningGoal.Margins`
Tracking	Hard	SP	W_1
RelGap Minimization	Soft	N/A	$\frac{1}{\texttt{RelGap}_{tgt}}$
Command Use	Soft	SP	W_2

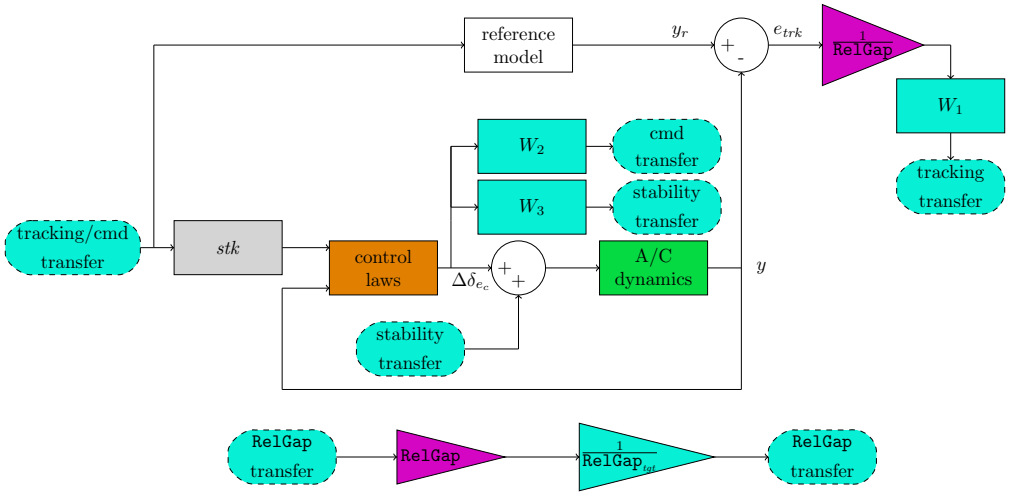


Figure 4.2 H_∞ synthesis constraints

likely to converge to values that perform well from an I/O perspective, but that behave in peculiar ways when looking at the gain values. To illustrate this, let $C(s)$ be a controller, defined as:

$$C(s) = \frac{N_1 s + 1}{D_1 s + D_0} K$$

In this example, $C(s)$ will necessarily be under-constrained, as $C(0)$ can be set by both D_0 and K. Although designers could make use of this additional degree of freedom to normalize the gains in some way to make them more intuitive, performing an H_∞ synthesis on this controller would likely yield gains that would end up in an impractical form or simply have difficulty converging. Unconstrained dynamical degrees of freedom will suffer from similar problems, although examples quickly become less intuitive.

By looking at the architecture of figure 3.8, two transfers can be identified where such problems are likely. First, having two lead-lag filters in series may be problematic. After a few tests with local synthesis, it was found that cancelling one of the filters resulted in no significant performance degradation but removed the tendency to have a very fast pole that would have led to discrete implementation issues. The second problematic transfer is the pitch rate feedback. On top of having a similar problem as there are two first-order filters at the denominator, the time constant of the washout can be used in a similar way to K_q as it is also on the numerator. After a few tests, the lead-lag filter was completely removed and the washout was re-written as $s/(\tau s + 1)$ to avoid conflict with K_q. Designer experience suggests that the lead-lag filter was to be used if better margins were needed, although it

was canceled for flap 0. Finally, the washout time constant will be set as a scalar gain (not scheduled) as no significant performance improvements were obtained by changing this value between design points. These tests have been performed again on the full envelope with the gain scheduling and have confirmed that these modifications improved the convergence while causing little to no performance loss.

4.3 Towards the synthesis of a scheduled controler

Gain scheduling through classical techniques has already been performed on the target aircraft by the industrial partner, schedules being tabulated in function of the dynamic pressure and altitude. These tables provide a reliable reference, both in terms of performance target and in terms of desired gain behaviour. Indeed, a rough estimate of the gain behaviour is required before performing a priori scheduling. As gain schedules tuned through structured H_∞ synthesis will be defined as polynomials, it is necessary to choose their order. The use of polynomial surfaces is interesting, as this imposes a degree of smoothness and continuity gains must follow. Smooth gain evolution is essential to gain scheduling, as discontinuous gain surfaces could result in discontinuous command signals.

Primary gains of the target aircraft, such as K_p, K_i and K_{ff}, behave as the inverse of the dynamic pressure ($\bar{q}$), while the dependency on altitude is relatively weak and much less pronounced. The gains K_d and K_q behave in a roughly linear way and decrease in function of $\bar{q}$. The evolution of command filters is less straightforward, although one is a lead filter with constant parameters, while the second is a lag filter that provides increasing lag at higher dynamic pressure. This lag is also provided at higher frequencies as altitude and dynamic pressure increase.

4.3.1 Scheduling variable choice and needed polynomial order

Three types of polynomials have been considered:

$$K(\bar{x},\bar{y}) = K_{00} + K_{10}\bar{x} + K_{01}\bar{y} \tag{4.15a}$$

$$K(\bar{x},\bar{y}) = K_{00} + K_{10}\bar{x} + K_{01}\bar{y} + K_{11}\bar{x}\bar{y} + K_{20}\bar{x}^2 + K_{02}\bar{y}^2 \tag{4.15b}$$

$$K(\bar{x},\bar{y}) = K_{00} + K_{10}\bar{x} + K_{01}\bar{y} + K_{11}\bar{x}\bar{y} + K_{20}\bar{x}^2 + K_{02}\bar{y}^2 + K_{21}\bar{x}^2\bar{y} + K_{12}\bar{x}\bar{y}^2 + K_{30}\bar{x}^3 + K_{03}\bar{y}^3 \tag{4.15c}$$

where $\bar{x}$ and $\bar{y}$ are scheduling variables normalized such that they range between $[-1, 1]$. The goal of this normalization is to reduce matrix conditioning. Such polynomials have been successfully used as aircraft gain schedules [28]. Alternative scheduling variables have been tested by fitting gains in function of different variables (e.g. Mach number). This was done through classical least of squares to evaluate all coeficient values, giving an R^2 value that characterizes the quality of the fit. Alternative scheduling variable choices achieved very similar fitting "goodness", w.r.t. visual inspection and R^2 values, which is logical as two scheduling variables fully define a flight point for ISA conditions (e.g. the Mach number can easily be obtained from $(\bar{q}, h)$). Best results were achieved with second-order polynomials, which successfully approximate the $1/\bar{q}$ behaviour for primary gains. Although third-order polynomials achieve better fits for secondary gains, the additional complexity was deemed too large for the little benefits they brought. Furthermore, some synthesis attempts with third-order polynomials resulted in gain surfaces having "kinks", justifying not to use them. Fitting of first-order polynomials in $1/\bar{q}$ gave good results, but created significant errors for secondary gains.

4.3.2 Reference model parameter interpolation

While choosing the reference model parameters with the iterative process described is convenient for a few flight conditions, doing so throughout the whole envelope would be time-consuming. Interpolating a set of parameters that were defined by hand in function of the only parameter that is imposed (T_{θ_2}) can simplify this process. Figure 4.3 shows this interpolation for a single loading, but for a set of flight conditions. Most of the parameters have a linear relationship with the zero, aside from ω_n which may be fitted in function of $1/T_{\theta_2}$. As ζ is imposed, this parameter has a very small correlation with T_{θ_2}, which is why the R^2 of the fit was not computed.

4.3.3 Robustness problems and limitations of reference model parameter interpolation

A few syntheses were performed for a single loading and multiple flight conditions, with the same constraints as the local synthesis, to test the interpolated reference model. To do so, the tracking requirement was modified to have a unique reference model for each flight condition, along with the corresponding weight. Doing so yielded good handling qualities. However, when introducing a range of masses and/or cgs, the tracking achieved was extremely limited, resulting in poor handling qualities. To eliminate potential scheduling problems, the same process was repeated for a single flight condition, but multiple loadings, with the same

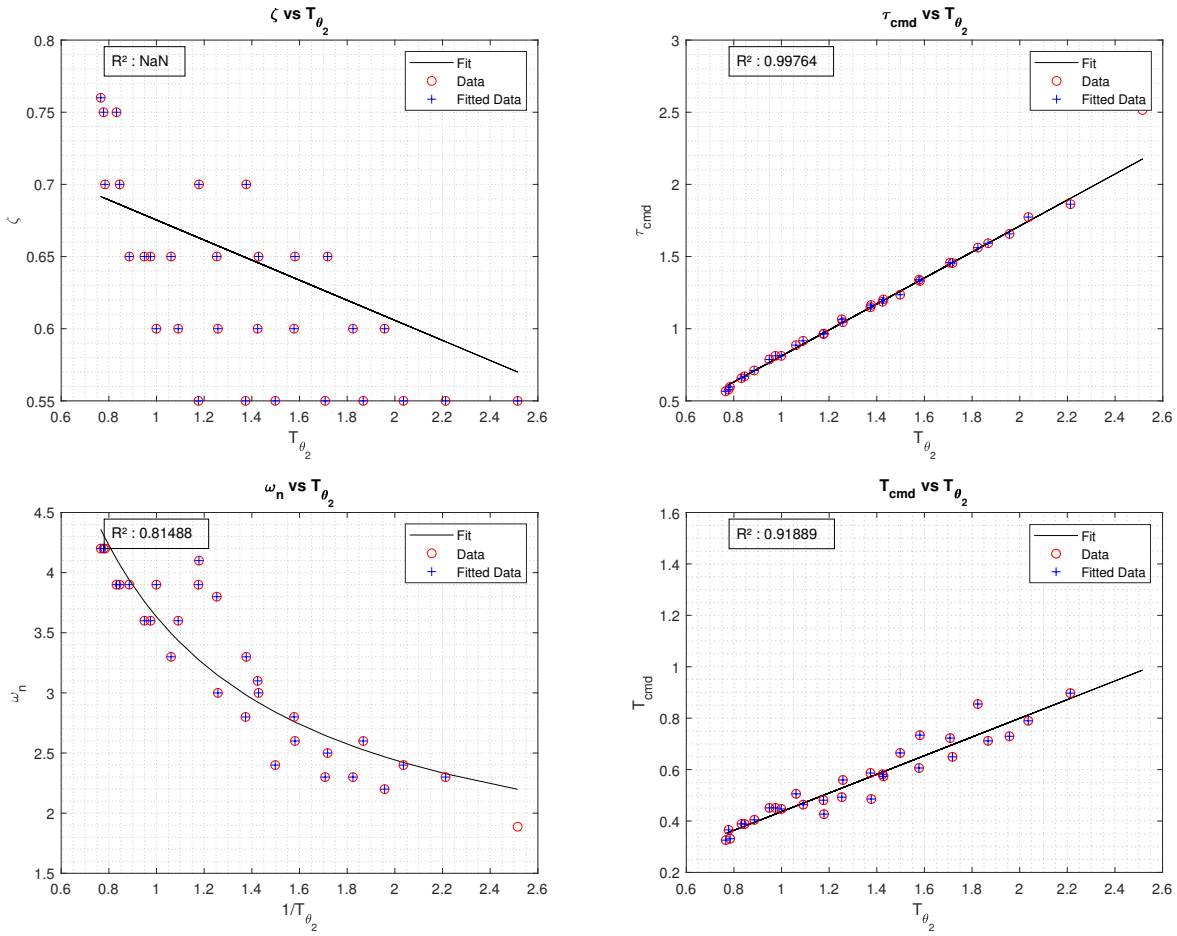


Figure 4.3 Reference model parameter interpolation

results. As T_{θ_2} is independent of the cg position, the same reference model ends up being used for multiple cg values. As gains are not scheduled w.r.t. the weight nor the cg, the synthesis is unable to match the (unique) reference model to the vastly different aircraft dynamics. Although T_{θ_2} is dependent on the weight, the small reference model variations are not sufficient to give adequate closed-loop targets. Given the absence of relevant scheduling variables, imposing a behaviour for a given loading implicitly imposes other behaviours for the rest of the loading envelope. Although defining a reference model per loading would have been a solution, this would have led to long design times without any guarantee of being optimal.

In reaction to this problem, it was decided to let the synthesis tune the optimal reference model for each synthesis point, allowing the tracking requirement to adapt to aircraft dynamics. Handling qualities will instead be met through control of the range each parameter

of the reference model can take. Assuming a good enough matching, if it is known that all reference models within a given subspace of parameters are satisfactory references, the synthesis results should also be satisfactory.

4.4 A priori scheduled controller

In this section, the synthesis methodology for the whole envelope is detailed. Following the previous discussion, the process that allows to limit the variable reference model to acceptable parameters is described. To do so, handling qualities are first evaluated on a 4D parameter grid in $(T_{cmd}, \omega_n, \tau_{cmd}, T_{\theta_2})$. The combinations resulting in unsatisfactory HQs are then filtered out, giving an "acceptable space" of parameters. To limit parameters within this space in the synthesis, this discretized 4D space is approximated by polynomials obtained through a special fitting process. Reference model parameters can then be defined in the synthesis through changes of variables that allow to limit them within these polynomial surfaces. Once the variable reference model process has been discussed, updated synthesis constraints are given. Finally, a process to define the K_{ff} surface prior to the synthesis to avoid nonlinear problems is established. This process is based on a criterion that ensures the absence of windup upon command reversal.

4.4.1 Variable reference model

The reference model as defined in section 4.2.1 has six parameters which may change from one synthesis points to another $(T_{K_{ff}}, T_{cmd}, \omega_n, \zeta, \tau_{cmd}, T_{\theta_2})$, although T_{θ_2} is imposed by the synthesis point. Furthermore, $T_{K_{ff}}$ and T_{cmd} should be set to the same value to maximize the bandwidths. From the experience gained performing local synthesis, the damping ratio has a very small effect on handling qualities (so long as it is taken low enough to meet good BW_γ requirements). Good results may be achieved with a fixed ζ, potentially leaving some small variations accessible to the synthesis for very high speeds ($\geq V_{Mo}$) if needed. Therefore, to define a space of acceptable parameters that will meet the requirements, the handling qualities of a 3D grid of combinations of $(T_{cmd}, \omega_n, \tau_{cmd})$ can be evaluated numerically for a given T_{θ_2}. The assumption that ζ is relatively constant allows displaying of this space in 3D, which is useful (arguably essential) for design purposes.

Evaluating handling qualities from reference models

Since the reference model has been defined for the pitch rate transfer, the evaluation of handling qualities related to this output can be done directly (Drb, PRO, BW_θ, τ_p, Gibson criteria). Indeed, even for HQs that should be measured on the complete model, it has been shown in section 4.2.1 that the SP approximation is adequate for BW_θ measurement. It might be possible to save time by evaluating only the BW_θ and not the f_{180} (as the requirements are similar) or not evaluating either the APR or τ_p (as the low order of the reference model yields good characteristics). The PRS will be addressed in the following pages, as this HQ is dependent on the reference model gain, which is affected by parameters other than T_{θ_2}.

For HQs that apply to other transfers, approximations from section 3.3.1 may be used to evaluate the corresponding HQs (in n_z, γ or α). Note that the BW_γ cannot be explicitly enforced with this methodology, since the SP reference model is not representative of the full model for the flight path response. Nonetheless, from the results shown in section 4.4.1, this HQ seems controllable indirectly through the reference model when performing a synthesis on the whole envelope, despite the poor correlation that was observed for local synthesis.

Special modifications for the pitch rate sensitivity In the previous section on local synthesis, no guidelines for reference model parameters were given for the PRS. This is because this HQ was only verified after each local synthesis. For the few problematic cases, changing the damping ratio generally solved the problem. The majority of these syntheses were performed on a loading located around the centre of the envelope, which might have increased the chances of this happening. Furthermore, attempts to significantly change the PRS by changing the reference model parameters usually required the degradation of some other HQ outside of the desired bounds. This became problematic for the synthesis on the whole envelope, as the PRS of some flight conditions degraded to an unacceptable extent when left unchecked. The main difficulty of limiting the PRS with the variable reference model is that this HQ is dependent on the system's steady-state gain, which varies in function of the scheduled and unscheduled variables. While open-loop aircraft gains can be accounted for with dynamic pressure and altitude, the gain used to approximate the envelope limiting functions for small amplitudes is not negligible. This gain is dependent on the loading of the aircraft, as heavier aircraft will fly at higher α, usually resulting in more interactions with envelope limiting functions. It is important to note that no tunable gain can affect the reference model steady-state gain (as can be inferred from eq. 4.1). For each new variable, the design of the reference model is significantly more complex as the number of design points increases exponentially, even if local limit design may be done in a 3D space.

The first solution tested was to simply fit the upper and lower system gain boundaries throughout the whole envelope in function of T_{θ_2}. The reference model parameters could then be limited by ensuring the *PRS* was adequate for both boundaries. While extremely conservative, this approach has the benefit of keeping the same number of design variables. To do so, the steady-state reference model gain was computed for all available design points, shown in figure 4.4. Note that the peak variation obtained this way is larger than the difference between the maximum and minimum *PRS* requirements. This means that it is impossible to meet the requirement through this approach.

A similar solution that yielded satisfactory results was to define similar boundaries in function of T_{θ_2} for each design altitude. Although this adds a design variable, the number of design altitude is relatively small and design in function of the altitude is already necessary. Indeed, the current control law achieves very small linear overshoots at low altitudes, while at medium and high altitudes they are much larger. This behaviour change is due to strong nonlinearities in the aircraft C_m (pitch-up effect) in transonic regime. This degradation is addressed by control laws that are intended to deal with nonlinearities, which are omitted in linear HQ evaluation. Looking at each altitude slice separately, the worst-case variations are smaller than the design requirements, which allows to achieve satisfactory performance. A 3D view of the reference gain in function of altitude is given in figure 4.5.

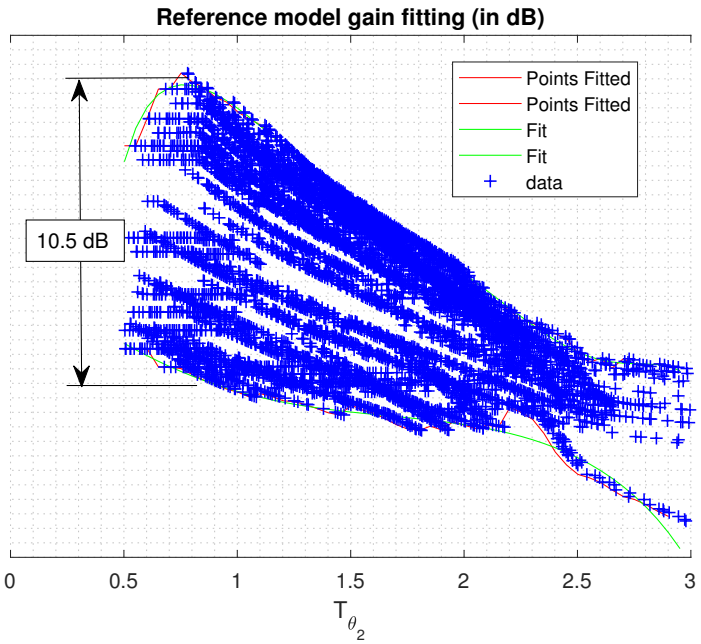


Figure 4.4 First attempt to limit the *PRS*

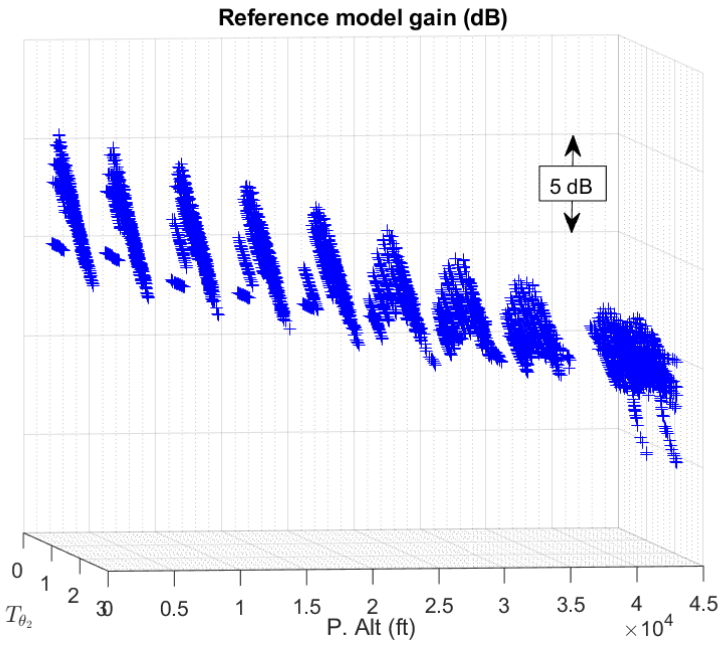


Figure 4.5 Reference model gain in function of altitude

Correlation between the reference model and system flight path bandwidth Figure 4.6 shows the error between the reference model BW_γ and the corresponding value on the closed-loop system for a synthesis (a negative error means the aircraft has better bandwidths than the reference model). Note that this synthesis was performed with a different flight and loading envelope. The standard deviation of this error is approximately 0.06. Although this is not expected to allow precise control of this HQ, it is still small enough to limit significant degradations and was sufficient to achieve good results.

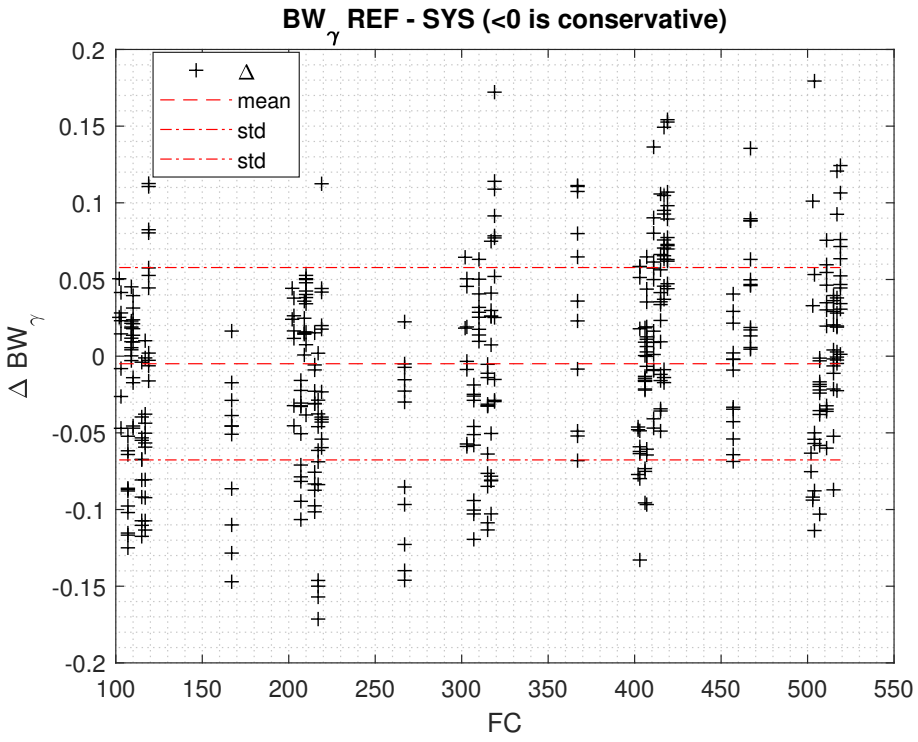


Figure 4.6 Error between the reference model and the real BW_γ

Defining reference model parameters limits for the synthesis

Figure 4.7a shows the boundaries of the upper and lower ω_n limits of the acceptable parameter space for $T_{\theta_2} = 1.93$, $\zeta = 0.6$ and level 1 HQ limits. To obtain these parameter limits, handling qualities were computed over a grid and parameter combinations that resulted in HQs outside the desired ones (e.g. $Drb < 0$) were filtered out. The range of parameters in the grid was heavily based on the range of values used by the interpolated reference model. After a first HQ evaluation, it was adjusted when the portion giving acceptable HQs was significantly truncated. While this whole acceptable space would ideally be accessible to the synthesis, implementing this in the MATLAB H_∞ framework is problematic. Indeed, tunable parameters declared with the function `realp` may have fixed numerical upper and lower bounds, but not functions dependent on other parameters. Inscribing a rectangular prism within the allowable limits would be extremely conservative, despite improving results compared to a fixed reference model.

Before presenting the final approach used for parameter limitation, it is interesting to mention that the constraint presented in (4.16) was first used to limit ω_n in function of τ_{cmd} and

T_{cmd}. This constraint is formulated as a gain limit for an artificial transfer. This gain is the distance between ω_n and a median plane $\mu(\tau_{cmd,i}, T_{cmd,i})$, normalized by the maximum allowable distance $\sigma(\tau_{cmd,i}, T_{cmd,i})$. Using this constraint and limiting the other two parameters explicitly (resulting in a rectangle) yields significant improvements over a simple cuboid. Nonetheless, the constraint in ω_n degrades the convergence of the synthesis.

$$G_{2,i} = \left|\left| \frac{\mu(\tau_{cmd,i}, T_{cmd,i}) - \omega_{n,i}}{\sigma(\tau_{cmd,i}, T_{cmd,i})} \right|\right| \leq 1 \tag{4.16}$$

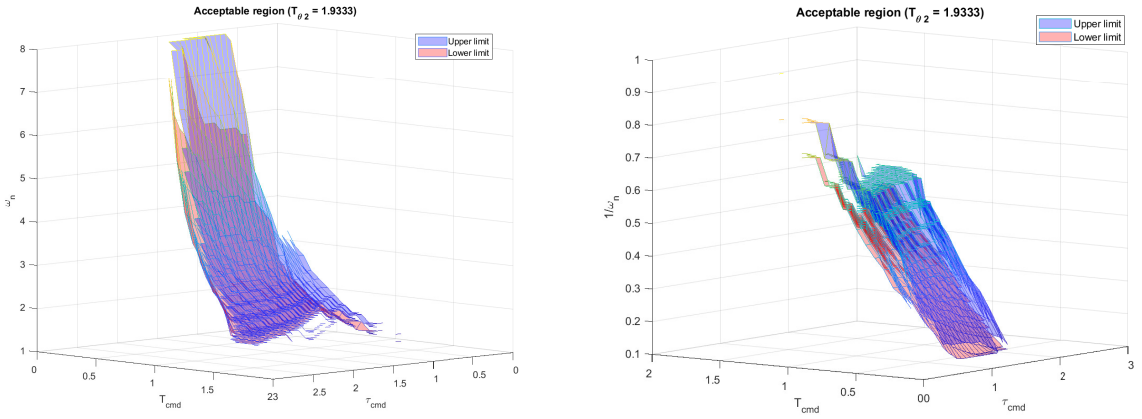
The second tested methodology was to add a similar constraint in T_{cmd}, allowing to cover most of the acceptable space. This resulted in unacceptable convergence degradation, which allowed to identify the weakness of this type of constraint. When this type of formulation is used, the dependency of the limited quantity w.r.t. other parameters is highly implicit. For example, the partial derivative for τ or T would not take into account the performance change from a potential increase of the ω_n limit. This is the best explanation found for the convergence degradation, although it remains very high level and qualitative.

The final approach is based on two changes of variables that make the dependency of parameter limits explicit :

$$T_{cmd} = \mu_T(\tau_{cmd}) + \sigma_T(\tau_{cmd})\rho_T, \quad \rho_T \in [-1, 1] \tag{4.17a}$$

$$\omega_n = \mu_\omega(\tau_{cmd}, T_{cmd}) + \sigma_\omega(\tau_{cmd}, T_{cmd})\rho_\omega, \quad \rho_\omega \in [-1, 1] \tag{4.17b}$$

where τ_{cmd}, ρ_T and ρ_ω are tunable variables with explicit bounds, while $\mu_i(...)$ and $\sigma_i(...)$ are polynomial surfaces that correspond to the mean value and maximum deviation from the mean value. This change of variable allows to limit both values according to a maximum distance from a nominal value. When implementing these limits in MATLAB, it is important to minimize the number of times each tunable parameter appears by factorizing expressions, as these parameters are unique to every synthesis point. Although MATLAB does not detail how tunable parameters are handled internally, it seems that every instance is stored independently, including duplicated variables. This is why `b = 2*a` or `b = a*(1 + a*(1+a))` will take roughly half the memory that unfactorized expressions would have used (`b = a + a` or `b = a + a^2 + a^3`). Although this is negligible for small systems, the variable change creates a significant amount of parameter duplication (due to μ and σ). This resulted in large tunable arrays (up to ≈ 3 Gb) leading to large RAM use and long `.mat` file saving times. Factorizing expressions allowed to reduce the size of synthesis data by a factor of 2. Finally, ω_n limits have a $1/x$ behaviour making the surface concave. Fitting $1/\omega_n$ (see figure 4.7b) gives limits that are more convex, generally improving the quality of fits.



(a) Acceptable parameter space for ω_n (b) Acceptable parameter space for $1/\omega_n$

Figure 4.7 Acceptable parameter space

Definition of limiting surfaces As the synthesis is expected to be limited by stability margins and by reference model parameter limits, a larger accessible parameter space should yield better reference matching. The definition of μ and σ limits should therefore maximize the region accessible to the synthesis, while staying within the acceptable space. As the ω_n limits are valid for a single value of T_{θ_2}, interpolation of μ and σ in function of T_{θ_2} is needed to map all values of T_{θ_2} within design limits, allowing to avoid limit computation for each zero used for design. It is assumed that a fine enough grid will allow to correctly limit the reference model values. This results in two interdependent interpolation layers. Due to the discrete nature of the fitted data, performing locally optimal (w.r.t. T_{θ_2}) fits and interpolating them gives oscillating and/or sub-optimal results. The methodology described here will be optimal with regard to the cost function:

$$J = \frac{1}{2}\sum_{i}^{N} f(e_{1,i})^2 + f(e_{2,i})^2 \tag{4.18}$$

with:

$$e_{1,i} = Q_{max,i} - \mu(\mathbf{x}_i, T_{\theta_2,i}) - \sigma(\mathbf{x}_i, T_{\theta_2,i}) \tag{4.19a}$$

$$e_{2,i} = \mu(\mathbf{x}_i, T_{\theta_2,i}) - \sigma(\mathbf{x}_i, T_{\theta_2,i}) - Q_{min,i} \tag{4.19b}$$

where $f(e)$ characterizes the "goodness" of the fit, Q_{max} is an upper bound on the limited value, while Q_{min} is a lower bound. The vector $\mathbf{x}$ corresponds to the polynomial basis of

independent variables, aside from T_{θ_2} (e.g. $[1; \tau_{cmd}; T_{cmd}, ...]$). This formulation gives positive errors when the accessible space is within $[Q_{min}, Q_{max}]$, allowing the use of an asymmetric cost function to penalize fits that are not conservative enough. Although a classical least of squares (e.g. $f(e) = e$) is a good first choice, $f(e) = \ln(c + e)$ allows to specify a degree of conservatism (through c) which is interesting for design. With this cost function, it is essential to monitor the iteration gain and initial conditions to ensure the gradient is bounded. As $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ (surfaces evaluated for a given T_{θ_2}) must be implemented in the H_∞ framework, it is difficult to define these surfaces with other functions than low order polynomials. On the other hand, more freedom exists for the interpolation in T_{θ_2}. Nonetheless, attempts to optimize numerically piece-wise functions (linear interpolation or Akima interpolation [61]) were not successful due to excessive dependence on initial conditions. This is why this second interpolation layer will also be done with polynomials. The main drawback of polynomials is that the use of a high-order (compared to the number of points) will result in oscillations. As a relatively high order is needed to obtain good fits, one must ensure to have a fine enough grid in T_{θ_2}. Variables to optimize can therefore be written as :

$$\mu(\mathbf{x}, T_{\theta_2}) \text{ or } \sigma(\mathbf{x}, T_{\theta_2}) \equiv \left[\begin{array}{cccc} \mathbf{p_0}\mathbf{T}_{\boldsymbol{\theta_2}} & \mathbf{p_1}\mathbf{T}_{\boldsymbol{\theta_2}} & ... & \mathbf{p_n}\mathbf{T}_{\boldsymbol{\theta_2}} \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \\ ... \\ x_n \end{array} \right] \tag{4.20}$$

where $\mathbf{T}_{\boldsymbol{\theta_2}}$ is a set of powers of T_{θ_2} arranged in a column, $\mathbf{p_i}$ is a set of coefficients (p_{ij}, in a row), such that $\mathbf{p_i}\mathbf{T}_{\boldsymbol{\theta_2}}$ gives a coefficient (p_i) of μ that is evaluated for a T_{θ_2} value. These evaluated coefficients may then be multiplied by their corresponding x_i (some power and combination of the independent(s) variables, i.e., τ and/or T) to obtain μ. It is possible to define σ in a similar way. Note that variables may be normalized within $[-1, 1]$ to avoid numerical problems. Therefore, the derivative of $\mu(\mathbf{x}_i, T_{\theta_2,i})$, where i is the data point, with regards to each coefficient in μ ($\mathbf{p}_\mu$)is :

$$\frac{\partial \mu(\mathbf{x}_i, T_{\theta_2,i})}{\partial \mathbf{p}_\mu} = \tag{4.21}$$
$$\left[\left[\begin{array}{cccc} 1 & T_{\theta_2,i} & ... & T_{\theta_2,i}^m \end{array} \right] x_{1,i} \quad \left[\begin{array}{cccc} 1 & T_{\theta_2,i} & ... & T_{\theta_2,i}^m \end{array} \right] x_{2,i} \quad ... \quad \left[\begin{array}{cccc} 1 & T_{\theta_2,i} & ... & T_{\theta_2,i}^m \end{array} \right] x_{n,i} \right]$$

where m is the order of the T_{θ_2} interpolation polynomial. It is clear that the derivative of μ with regard to σ coefficients (or the opposite) is $\mathbf{0}$ and that the derivative of σ with regard to its coefficients results in the same expression. It is interesting to remark that this derivative is not dependent on the fit coefficients, allowing to compute this only once.

The gradient to use for the numerical optimization can therefore be written as (making abstraction of $f(e)$) :

$$\frac{\partial J}{\partial \mathbf{p}} = \sum_{i}^{N} f(e_{1,i}) \frac{\partial f(e_{1,i})}{\partial e_{1,i}} \left(-\frac{\partial \mu}{\partial \mathbf{p}}(\mathbf{x}_i, T_{\theta_2,i}) - \frac{\partial \sigma}{\partial \mathbf{p}}(\mathbf{x}_i, T_{\theta_2,i}) \right) + f(e_{2,i}) \frac{\partial f(e_{2,i})}{\partial e_{2,i}} \left(\frac{\partial \mu}{\partial \mathbf{p}}(\mathbf{x}_i, T_{\theta_2,i}) - \frac{\partial \sigma}{\partial \mathbf{p}}(\mathbf{x}_i, T_{\theta_2,i}) \right) \tag{4.22}$$

This allows to use a first order solver described in algorithm 2.

Algorithm 2 First order solver for the definition of limiting surfaces

1: Compute locally optimal surfaces for all discrete T_{θ_2} values.
2: Interpolate these parameters and initialize the coefficients of μ to the interpolated values.
3: Initialize σ as a small positive constant surface (increases odds of initial conditions within $[Q_{min}, Q_{max}]$). Compute $\frac{\partial J}{\partial \mathbf{p}}$ and $J(\mathbf{p})$
4: Update $\mathbf{p}_{tmp} = \mathbf{p} - k_{iter} \frac{\partial J}{\partial \mathbf{p}}^{\top}$
5: Compute $\frac{\partial J}{\partial \mathbf{p}_{tmp}}$ and $J(\mathbf{p}_{tmp})$. If they are bounded and real ($\ln(< 0)$ returns a complex number in MATLAB), $\mathbf{p} = \mathbf{p}_{tmp}$ and loop over step 4, else reduce k_{iter} and discard $\mathbf{p}_{tmp}$. Continue until $||\mathbf{p} - \mathbf{p}_{tmp}|| < \epsilon$

Bounds on the considered values of τ_{cmd} in function of T_{θ_2} may be found using a similar fit (but here the upper and lower limits may be independent, as these bounds are constant for a single T_{θ_2}). The limits for T_{cmd} and $1/\omega_n$ may be evaluated sequentially. As surfaces are expected to cover the majority of the acceptable space, this should not impact their optimality significantly. The limit in T_{cmd} is therefore defined in function of (τ_{cmd}, T_{θ_2}), while $1/\omega_n$ is defined in function of (T_{cmd}, τ_{cmd}, T_{θ_2}).

Flight conditions with high T_{θ_2}

As explained when the dropback margin was introduced in section 4.10, T_{θ_2} imposes a performance limit on the system in the bandwidths versus *Drb* trade-off. This may also be interpreted as a compromise between keeping an acceptable dropback (by placing a pole near T_{θ_2}), while also meeting good settling times in n_z (affected by the new pole). The acceptable parameter space is relatively small for values of T_{θ_2} around ≈ 0.5 for level 1 HQs, then gets bigger as the zero increases. Above values of ≈ 2.0, the space gets smaller and becomes extremely small for values near 3.0, after which the space gets too small to be captured without a very fine discretization. This means that for exceedingly large zero values, it becomes impossible to satisfy the requirements. As seen in section 3.5b, this happens for some synthesis points. Such points with $T_{\theta_2} > 3$ are ignored in the synthesis.

4.4.2 H_∞ constraints for the scheduled synthesis

Overall, synthesis constraints are very similar to those in section 4.2.2:

Constraint 1: A stability requirement to enforce good stability margins. For most conditions, these are gain margins of 6 dB and phase margins of 45°.

Constraint 2: A minimum damping requirement for poles in the SP frequency range to ensure adequate damping. This requirement was added as some iterations for low reference damping degraded further than intended.

Constraint 3: A modified reference tracking requirement to enforce HQs. As the reference model is unique for each point, the same is true for the weight (W_1). The function $W_{1,1}$ corresponds to the RMS norm of the reference model band filtered in [0.1,20] rad/s, which corresponds to the frequency range where HQs are measured. When performing a synthesis on multiple points at once, points where the reference model gain is large would be disproportionately penalized. As this is a dynamic norm, it is dependent on the reference model parameters. For scheduled synthesis, it should be updated in-between synthesis iterations (e.g. `RelGap` values) when multiple points have HQs beyond those imposed. Although this results in additional complexity, this performs better than simply using the steady-state gain. For large synthesis envelopes where only a few points degrade, manually increasing their W1 value (only for the loading and flight condition in question) generally solves the problem.

Constraint 4: The same requirement to limit high-frequency command use as the one used previously (W_2).

Table 4.3 Scheduled synthesis constraints

Objective	Constraint Type	Model	Weight
Stability Margins	Hard	Full	`TuningGoal.Margins`
Minimum SP damping	Hard	Full	`TuningGoal.Poles`
Tracking	Hard	SP	W_1
RelGap Minimization	Soft	N/A	$1/\texttt{RelGap}_{tgt}$
Command Use	Soft	SP	W_2

$$W_1 = \frac{20^2}{(s+20)^2}\frac{1}{W_{1,1}\texttt{RelGap}} \tag{4.23a}$$

$$W_{1,1} = \left\| \left(\frac{s}{s+0.1}\right)^2 Ref(s) \left(\frac{20}{s+20}\right)^2 \right\|_2 \tag{4.23b}$$

Synthesis constraints are summarized in table 4.3. A systematic method to adjust `RelGap` and `Ponderation` may be used. First, a synthesis is performed with a very high `Ponderation` (effectively ignoring the requirement) to see the best achievable `RelGap`. Then, a second synthesis is run with `RelGap` slightly below the achieved value and `Ponderation` $\in [1.5, 2.5]$. While these steps may not be necessary for local synthesis, they streamline the design process with the scheduled synthesis. This process also allows to see the best handling qualities achievable, eliminating the OLOP from the limiting factors.

4.4.3 Pre-fitted Kff surface

During the design process, the K_{ff} gain had a strong tendency to increase with dynamic pressure, which does not correspond to the use intended for this gain. From a nonlinear point of view, the "ideal" feed-forward would correspond to the steady-state deflection of the elevator, although this is dependent on the mass and cg of the aircraft, which are unavailable for scheduling. Figure 4.8 shows the behaviour of this deflection in function of the dynamic pressure. There are multiple z points for a given x, y couple due to different loadings being displayed. Still, this gain should decrease with the dynamic pressure. It is important to note that although this gain behavior is undesirable according to designer experience, HQs obtained were excellent. Upon validation of the peculiar K_{ff} on the non-linear model, overshoot problems (w.r.t. envelope limiting functions) were present for stick reversals in α and θ. These problems were attributed to integrator windup. Indeed, if for some mass and cg conditions the feed-forward is greater than the steady-state elevator deflection, the integrator must converge to a negative input (overcorrect in the opposite direction), resulting in heavy commands upon input reversal (high K_{ff} and high initial integrator condition). This problem was solved by ensuring the steady-state integrator value was positive. Constraints for the gain surfaces that were developed to solve this problem are developed in annex D, as they may be useful for different problems related to gain behaviour. Although these constraints allowed to solve α overshoots, the low speed behaviour of the gains obtained remained unsatisfactory as the resulting gain surfaces were nearly constant. This problem was solved by fitting the K_{ff} surface to an upper boundary that ensures no windup. Note that limiting

the K_{ff} surface resulted in the synthesis being unfeasible, even for hard constraints alone (stability constraints) when keeping the PH mode for all flight conditions as discussed in section 4.5.3.

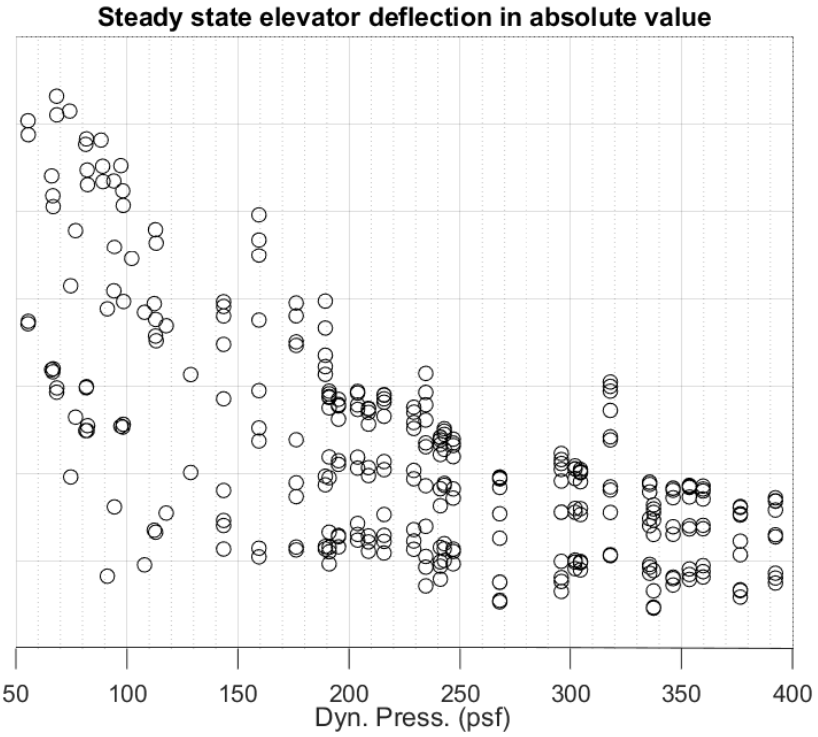


Figure 4.8 Steady state elevator deflection in function of $\bar{q}$

In an attempt to verify whether the main cause of nonlinear overshoots was truly integrator windup, a maximum K_{ff} surface has been defined and tested on the nonlinear model. To do so, the minimum (over the loadings) steady-state gain of the C* command to elevator deflection transfer was computed for each flight point. Following the reasoning given previously, ensuring that the K_{ff} surface stays below these points should eliminate windup problems. Therefore, using a surface that approximates these points should constitute an adequate feedforward surface. Although such a surface could be obtained with a least of squares, this approach would not penalize points above the fitted data. The fit should be strictly below these points, which calls for a peculiar cost function.

Let $e_i = y_i - f(\mathbf{x}_i)$, the fit error at point i in function of scheduling variables $\mathbf{x}$, a cost function that achieves the desired behavior is :

$$J = \sum_i \ln^2(c + e_i) \tag{4.24}$$

where c is a constant that allows to adjust the degree of conservatism with regards to being strictly below the fitted data as seen in figure 4.9.

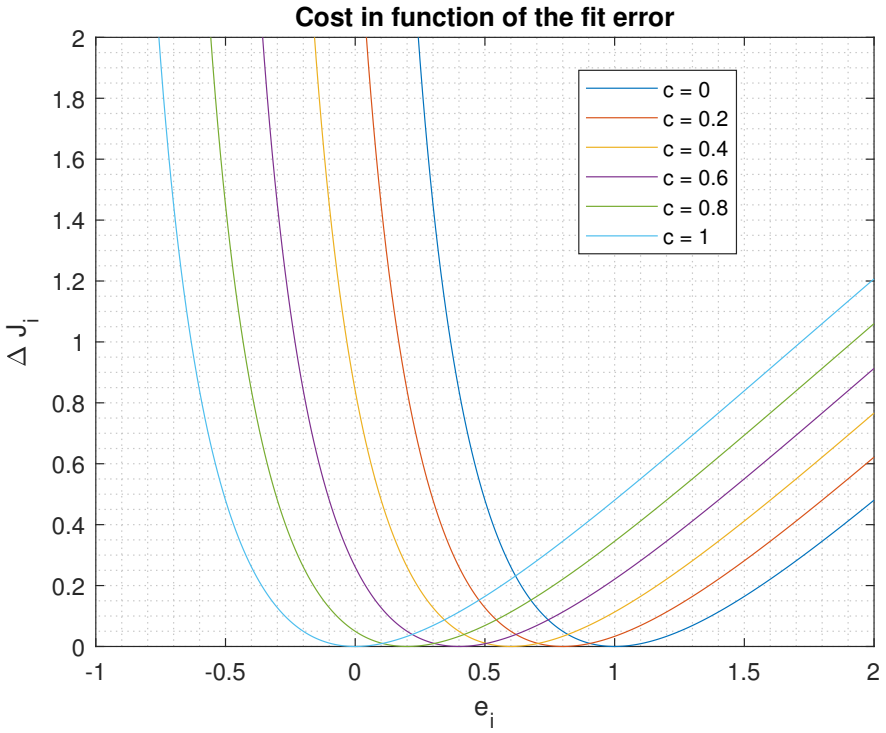


Figure 4.9 Cost in function of the fitting error and value of c

The gradient to use (with an algorithm similar to 2) is:

$$\frac{\partial J}{\partial \mathbf{p}} = \sum_i \frac{2\ln(c + e_i)}{c + e_i} \frac{\partial e_i}{\partial \mathbf{p}} \tag{4.25}$$

where $\mathbf{p}$ are the coefficients of the polynomial being fitted (e.g. $f = p_0 + p_1\bar{x} + p_2\bar{y} + ...$).

Equation 4.25 is not bounded when the surface is above the data by more than c for one point. Nonetheless, ensuring the initial surface is below the points to fit (e.g. $K_{ff}(\bar{q}, h) = 0$) and reducing the iteration gain whenever an iteration overshoots to an unbounded gradient allows to obtain good fits, as shown in figure 4.10.

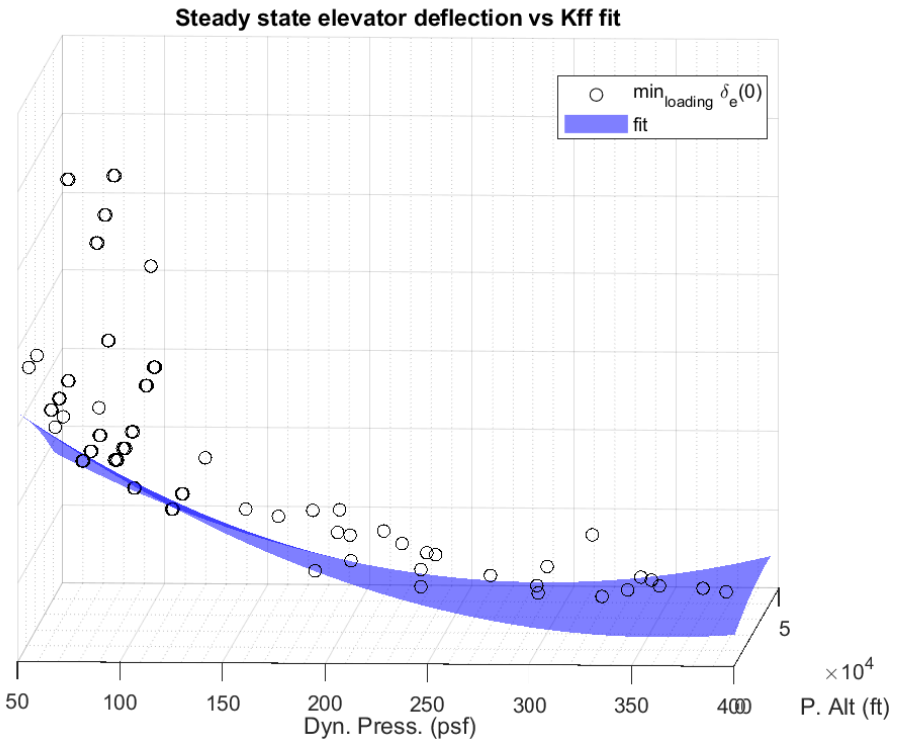


Figure 4.10 Pre-fitted K_{ff}

Although initially intended for validation of the criteria for windup, synthesis with the pre-fitted feedforward performed better in nonlinear tests. Furthermore, syntheses with the K_{ff} fixed to the surface obtained with this methodology do not seem to be significantly penalized compared to constrained surfaces when it comes to reference model matching. This is why this methodology was retained, as it was able to achieve better nonlinear behaviour, with little to no HQ degradation compared to other approaches.

4.5 Other considerations for the synthesis

When introducing the methodology with the variable reference model, high-level details were given. This is because a designer using it has a large amount of freedom when time comes for implementation, making it difficult to go into more details (e.g. limits polynomial order) without giving instructions that would be irrelevant for a different aircraft. This section intends to identify where such design choices must be taken and give general guidelines for future application of the methodology. Convergence problem observed and initial conditions are then discussed

4.5.1 HQ limits for the reference model

The primary factor that determines whether it is possible to achieve good reference model matching is the size of the acceptable space. Small spaces are problematic because there may not be enough room to account for the dynamics of different loadings. Conversely, needlessly large spaces may result in under-constraining the problem from the point of view of handling qualities. H_∞ synthesis relies on the minimization of the worst-case design objectives, meaning that all "non-limiting" constraints are essentially ignored. If the acceptable space is exceedingly large, the synthesis will converge towards an optimum that either minimizes `RelGap` or the command use. There is no guarantee that such an optimum will be in the centre of the space, which is often closer to the designer's intent. The best procedure for defining HQ limits is to first run a synthesis for "minimalist" design requirements to get an idea of what is achievable. HQ limits may then be tightened progressively as desired. It is often difficult to see what has limited a synthesis that is too stringent (as performance is degraded for multiple design points), while it is easy to identify where improvements are needed. It is expected that a completely new design (where no prior estimate of what HQ can be achieved) will require some iterations of the HQ limits.

4.5.2 Use of different ζ values for the reference model

As previously mentioned, a fixed value of ζ is required to allow the design of parameter limits in 3D space. A value of 0.6 is satisfactory for the considered target aircraft, as values below $\approx$0.55 were found to lead to long settling times (due to large undershoots that start exiting the settling criteria). Values above 0.8 result in problems for the bandwidths and OLOP. Still, it is expected that (if it was possible) letting the damping ratio as a free tunable parameter should help to achieve better reference matching. By leaving the damping ratio tunable within a small range where the corresponding HQ degradation is small (e.g. [0.55,0.65]), it was found that better reference matching could sometimes be achieved. Nonetheless, it must be noted that this approach has the potential to converge to unacceptable HQs due to deviations from HQ evaluation, potentially making the additional matching ineffective, especially if the range given is too large.

4.5.3 Need for applying stability margins on the SP

The `TuningGoal.Margin` requirement has an implicit pole stability constraint, meaning that the transfers on which this requirement is applied must be strictly stable. This is problematic for the synthesis, as the phugoid may be unstable. As the C* architecture does not have a

great effect on this mode, synthesis with a stability requirement on the full model for such conditions will either result in the incapacity to meet the stability margins or difficulties significant enough to heavily penalize the reference matching. This is why such flight conditions had stability margins enforced on the SP model.

4.5.4 Notes on the linearization of envelope protection functions

Although envelope protection functions have an effect on the stick position to n_z command slope (hence the gain linearized for small amplitudes), they also have dynamical effects. The linearization of this component is separated from the longitudinal gain path and consists of an isolated block that biases the $n_{z,cmd}$ depending on sensor measurements. Figure 4.11 illustrates this. Handling quality assessment should be performed on the full closed-loop system for maximum accuracy. Nonetheless, synthesis performed with this component often fail because the reference model is no longer representative of the closed-loop. Indeed, this component often introduces a bias even at steady-state, effectively deviating from 1 C*, which causes problems. While the reference model could be modified to take this into account, the effect of these dynamics on most handling qualities was found to be negligible and remained qualitatively small when the aircraft was close to an envelope limit (e.g. high weights at low speeds). The synthesis is therefore done without this component and performance is validated afterwards. No problematic performance degradation was found with this process. One very important note is that these functions have a significant effect on stability margins. Gains that seem to respect stability margins in absence of the protection linearization may fail the requirements when evaluated on the full linear model. This is why two sets of linear models should be used for synthesis: a reduced SP linear model without the envelope protection for reference model matching and a second model (full order or reduced SP) that includes the command generation for stability requirements. See section 4.5.3 for details on why use the SP approximation even for the stability margins.

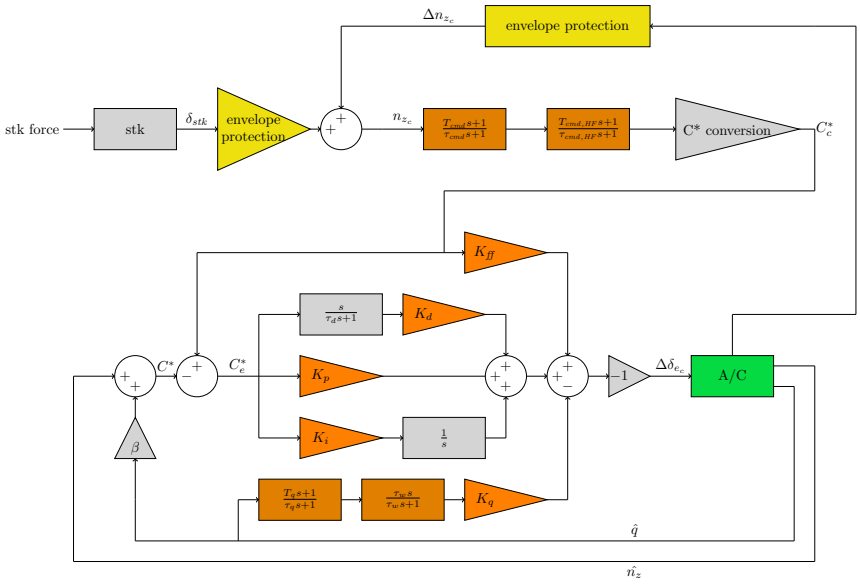


Figure 4.11 Linearization of the envelope protection functions dynamics

4.5.5 Initial conditions for the synthesis

Throughout literature, the common trend is to initialize structured H_∞ synthesis for relatively small problems with random values [2]. This approach is based on the fact that, since structured H_∞ may only reach a local optimum, trying multiple values that hopefully span multiple regions of convergence should help in finding better optimums. While this approach is convenient for small problems and allows to get results even without prior estimates of the gains, it becomes impractical as problem size increases.

On top of significant computational costs related to running multiple times the same (large) synthesis, one may expect an increase in the number of local optimums. These problems are especially relevant to the use case described in previous sections, as the reference model itself is tunable, resulting in a large number of degrees of freedom. To alleviate such problems, initial conditions for gains were instead set to polynomial interpolations of gains obtained by "classical" means for different aircraft variants. Similar approaches were effective with various gain estimates, whether from classical design [27], or from interpolation of local synthesis (effectively creating a crudely scheduled controller) [57]. In addition to providing a realistic guess that would be available for new aircraft, these gains also exhibit the desired behaviours (i.e. higher gains for low dynamic pressure). This increases the odds of keeping these desirable characteristics, instead of converging to gains that perform well on the LTI model, but that go against "design experience" and are likely to suffer from difficulties on non-linear models.

For the reference model, two initial condition options were tested:
Option 1: Using the interpolated reference model as initial conditions.
Option 2: Using the interpolated reference model as initial conditions for τ_{cmd} and initializing ρ_T and ρ_ω to 0.
In general, the second option seems to result in the best convergence proprieties, although it is difficult to compare these options without doing a large survey on many problems, which was not deemed important enough.

4.5.6 Modifications to `systune` to improve convergence proprieties

Description of the `systune` algorithm for multi-objective synthesis

From MATLAB documentation [62], the `systune` algorithm solves:

$$\min_{\mathbf{x}} \max\{\alpha f(\mathbf{x}), g(\mathbf{x})\}, \quad \text{s.t.} \quad g(\mathbf{x}) \leq 1 \tag{4.26}$$

where $\mathbf{x}$ are tunable variables, $g(\mathbf{x})$ are hard constraints, $f(\mathbf{x})$ soft constraints and α a variable (scalar) weight. While the quantity of information on this algorithm is relatively limited, [2] gives a high level description for solving an equivalent problem:

$$\min_{\mathbf{x}} \max\{f(\mathbf{x}), \mu g(\mathbf{x})\} \tag{4.27}$$

This article was written Pierre Apkarian, one of the main contributors to the `systune` function. The core of the function is a bisection algorithm on μ to solve $g(\mathbf{x}) = 1$. Each evaluation of the constraints for a given μ corresponds to a single H_∞ problem. To achieve better computational times and better overall results, the result from the last sub-problem is always used as initial condition for the next sub-problem. The optimization logic is shown in algorithm 3. Note that step 3 comes from the fact that in a "worst case" were both $f(\mathbf{x}_f)$ and $\mu g(\mathbf{x}_f)$ are active constraints, $f(\mathbf{x}_f) \approx \mu g(\mathbf{x}_f)$, hence the upper bound of $\mu \approx f(\mathbf{x}_f)/g(\mathbf{x}_f)$. For the presented constraints, it is simpler to implement the α variant since there are less soft constraints. Setting $\alpha = 1/\mu$ allows to do so. Indeed, for $J = \min_{\mathbf{x}} \max\{f(\mathbf{x}), \mu g(\mathbf{x})\}$, this substitution gives $J = \min_{\mathbf{x}} \max\{f(\mathbf{x}), \frac{1}{\alpha} g(\mathbf{x})\}$. Multiplying both constraints by α does not change the problem (making abstraction of the solver and its implementation), hence the equivalence of both methods. This variant also has the advantage of always respecting hard constraints (when the synthesis is successful).

Algorithm 3 Multi-objective optimization with soft and hard constraints, from [2]

1: Initialize $\mu\_ = 0$
2: Find a strictly feasible point such that $g(\mathbf{x}_f) < 1$
3: Initialize an upper bound on μ : $\mu^- = \frac{f(\mathbf{x}_f)}{g(\mathbf{x}_f)}$. Set $\mu = (\mu^- + \mu\_)/2$
4: Stop if $|\mu^- - \mu\_| < \epsilon$, else solve the problem for μ
5: If $g(\mathbf{x}) > 1$, set $\mu\_ = \mu$, else set $\mu^- = \mu$.
6: Update μ and loop over 4

Convergence problems observed

The first convergence problem that was identified with `systune` is shown in figure 4.12. This figure corresponds to the output of the `systune` function with the `display` option set to `sub`, which displays information on each sub-problem solved. At the first sub-problem (initial conditions already satisfied the hard constraints), the algorithm breaches the hard constraints and is unable to go back to the initial "valid" solution. In this example, the synthesis wasted a significant amount of time without improving results.

The most likely cause for this problem is that the synthesis overshot to a different point of convergence, which is not acceptable as $g(\mathbf{x}) > 1$, and that this point is more attractive than the initial one. If this hypothesis is true, using the last synthesis where $g(\mathbf{x}) \leq 1$ as initial conditions will ensure that it is impossible to re-use a synthesis that converged to an unacceptable point. This has been able to solve this type of convergence difficulty for all syntheses attempted since this modification. Furthermore, this convergence problem was related to the first limits used for the reference parameter limiting. Although this was not re-evaluated, synthesis with (4.17a) are not expected to suffer from this problem, even without modification.

```
Synthesis Starting...
alpha=0.1: Soft = 1.62, Hard = 0.9986, Iterations = 0
alpha=1.233: Soft = 0.854, Hard = 1.0546, Iterations = 213
alpha=0.9246: Soft = 0.854, Hard = 1.0545, Iterations = 11
alpha=0.7705: Soft = 0.855, Hard = 1.0543, Iterations = 11
alpha=0.6934: Soft = 0.858, Hard = 1.0539, Iterations = 11
alpha=0.6549: Soft = 0.858, Hard = 1.0537, Iterations = 11
alpha=0.6357: Soft = 0.86, Hard = 1.053, Iterations = 11
alpha=0.626: Soft = 0.861, Hard = 1.0523, Iterations = 11
alpha=0.6212: Soft = 0.865, Hard = 1.0514, Iterations = 11
alpha=0.6188: Soft = 0.866, Hard = 1.0511, Iterations = 11
alpha=0.6176: Soft = 0.867, Hard = 1.0507, Iterations = 11
alpha=0.617: Soft = 0.871, Hard = 1.0498, Iterations = 11
Final: Soft = 1.62, Hard = 0.9986, Iterations = 323
Saving Data
RelGap achieved : 0.110
Data Saved. Computing time : 52613.8 s
```

Figure 4.12 Typical convergence problem with `systune`

A second difficulty observed on large problems is the existence of multiple solutions for the parameter α. As `systune` uses the α version of the algorithm, a strictly feasible solution is found using $\alpha = 0.1$. This parameter is then increased progressively until an upper bound is found. While this is valid for problems with a unique solution, when multiple optimums exist this often results in finding a solution with a low α. This is problematic since solutions with higher values will achieve better $f(\mathbf{x})$. Indeed, since $\alpha f(\mathbf{x}) \approx g(\mathbf{x}) \approx 1$, $1/\alpha$ gives a very good approximation of the soft constraint that will be achieved. In figure 4.13, the same constraints were used on the same problem, but different initial conditions, termination criteria and initial values of α for the feasibility synthesis were used. It should be noted that the synthesis that achieves the best results converged in the most sub-problem calls and run time (51 hours). If these problems were to be run sequentially, one can easily understand how this would significantly limit design iterations (e.g. on reference model parameters).

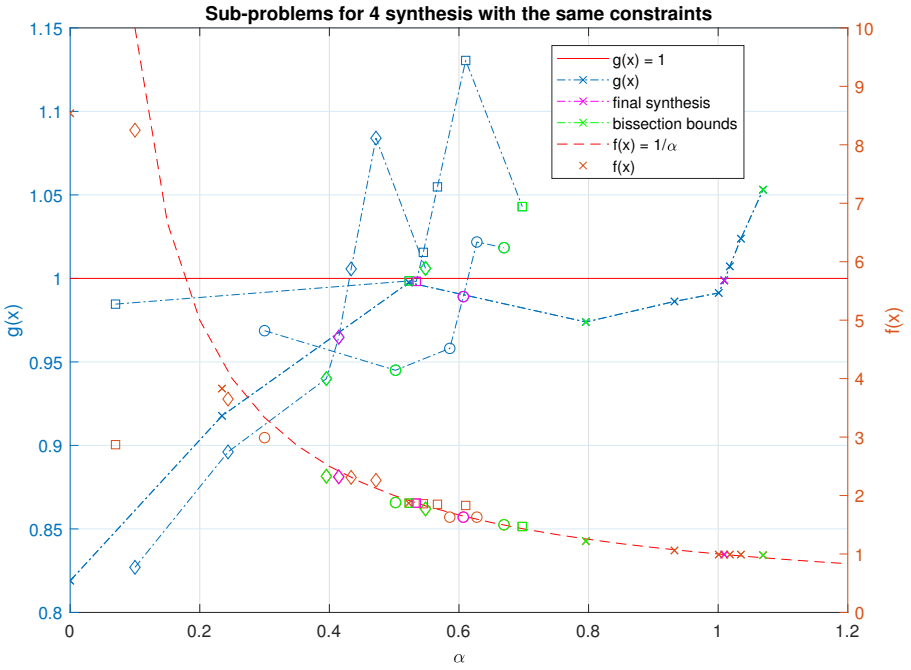


Figure 4.13 Multiple optimums problem with `systune`

In this figure, it can be seen that the optimums ($g(\mathbf{x}) = 1$) behave quite erratically from one synthesis to another, even if the problem is essentially the same and only initial conditions or solver parameters are changed. Furthermore, the assumption that lower values of α will yield lower values of $g(\mathbf{x})$ does not hold globally. Given the erratic behaviour observed and the fact that each sub-problem takes multiple (if not tens of) hours to run, it is more interesting to let the choice of α to the designer (possibly using a similar plot to choose which

α try). The sequential process of `systune` is beneficial when $g(\mathbf{x}) < 1$, as running a new synthesis from the best initial conditions generally results in better values of α being achievable. Indeed, running anew a synthesis for a value of α that was already attempted and failed can work for initial conditions resulting from a higher α than what was previously attempted.

On the other hand, this sequential process results in significant time losses when $g(\mathbf{x}) > 1$, as these results may not be used as initial conditions due to the first problem presented. The best compromise found is to run parallelized batches of sub-problems simultaneously, greatly reducing the probability that they all yield $g(\mathbf{x}) > 1$, and always using the best initial conditions available at the start of each batch. The values of α to try are ultimately dependent on designer experience and the knowledge of what values of `RelGap` gives good HQ. When such knowledge is not available, using an exponential spread (similar to `systune`) can be a last resort. This effectively solves the problem observed in 4.13. Furthermore, as reference matching can be inferred from $1/\alpha$, one can know what to expect from each synthesis assuming it succeeds, allowing to terminate the process early when additional sub-problems would offer virtually no improvements over already achieved results.

This process is summarized in algorithm 4, where n_c is the number of cores available for the synthesis. Note that running a parallelized bisection method is always possible near the end of the process, or using a method such as [63] that offers an order of convergence proportional to the number of threads with no derivative and reasonable assumptions for root finding. Nonetheless, the evolution and unpredictability of the bounds seem too severe to justify the use of root-finding algorithms.

Algorithm 4 Multi-objective optimization with soft and hard constraints guidelines

1: Choose n_c values of "low" α (0.1 is a good reference point) and launch said n_c synthesis, achieving $g(\mathbf{x}) < 1$ for at least one.
2: Choose new values if this is not the case (running a synthesis with $\alpha = 0$ allows to verify if the problem is feasible).
3: Choose the synthesis that yielded the lowest $f(\mathbf{x})$, subject to $g(\mathbf{x}) < 1$, and use it as initial condition.
4: View previous synthesis results in a plot similar to figure 4.13 and choose n_c values of α such that all of the tested values result in significant improvements of $f(\mathbf{x})$ over already achieved synthesis.
5: Loop over 2 while significant progress is judged achievable. The use of gradient-less root-finding algorithms may be used for final iterations.

4.6 Conclusion

In this chapter, a methodology to perform gain scheduling through structured H_∞ has been developed. This methodology is based on the respect of stability margins, limiting the high-frequency command gain and reference model matching. To allow sufficient flexibility w.r.t. the wide range of dynamics that result from the lack of loading information, this reference model is different for each synthesis point and is tuned by the synthesis. The space accessible to reference model parameters is limited to a sub-space with the desired handling qualities, ensuring good HQs so long as the reference matching is adequate.

Given the number of parameters required to define the reference model throughout the envelope, it is clear that some fine-tuning is required to obtain optimal results with this methodology. As the synthesis process is fairly time-consuming ($\approx$12 hours with the envelope used), the need for these iterations is the main drawback. Nonetheless, the overall design effort remains significantly reduced compared to classical methods, as the design can be done by a single designer who is free to perform other tasks while the synthesis is running.

CHAPTER 5 CONTROL LAW BY EQUATIONS (G*)

The concept of control law by equations (called G*) was developed in recent years by Airbus [46], [47] to address some conceptual problems of gain scheduling:

- Flight tests often allow to refine aircraft models, leading to gain re-design.
- Gain scheduling leads to dissimilarities of control laws between aircraft, due to architecture change risks.
- Changes to hardware components (e.g. control loop delays) may lead to gain re-design.
- When using a large number of scheduling variables, table interpolation may lead to significant computational loads.

Furthermore, from experience gained through structured H_∞ synthesis, additional weaknesses may be identified with regard to modern gain optimization process:

- Most gain optimization methods (including many approaches in literature, e.g. [24]) rely on the capacity of designers to give HQ targets to meet during the optimization process. In many cases, the fine-tuning of handling qualities is complex and iterative, due to the necessity of exploring what the aircraft can achieve through the optimization process. This is justified by the fact that some handling qualities become mutually exclusive near envelope limits. In other words, the designer must perform a first synthesis with an initial HQ guess to then adjust the target handling qualities from previous synthesis results. This increases design time, as the optimization process is generally time-consuming.
- The gain design problem seems qualitatively under-constrained when only considering linear HQs. This can be seen either by the need to pre-define the feedforward in H_∞ design or by the need to remove lead-lag filters in series. Obtaining gains that perform well in the linear model does not guarantee good nonlinear results when designer experience is not respected.

These weaknesses are addressed by the G* methodology that will be developed in this chapter. While the first set of limitations is handled by the G* philosophy in itself, which will be described in the following paragraphs, the second set is addressed through the methodology developed in section 5.3. The nonlinear design will then be discussed to address some of the strong hypotheses used in linear design.

5.1 Description of the G* philosophy

Paraphrasing the description given in [47], a G* control law is composed of a cascade of equations, which computes gains to achieve lower-level objectives up to higher-level ones, in real-time. This computation starts with a simplified aerodynamic model and feedback (sensor) dynamics, ensuring good lower-level performance (e.g. C* tracking) through pole placement. The corresponding closed-loop transfer function is obtained as part of the pole placement process, allowing to repeat it for higher-level control loops (e.g. autopilot functions), using the same generic functions. The only functions which may not be designed with this methodology are aircraft structure-specific functions (e.g. ASE filters), although gain dependency with regards to these elements is limited through the feedback dynamics approximation used for gain computation. This approximation can easily be changed from one aircraft to another.

In the case of the longitudinal control law, the architecture includes a linear loop (classical longitudinal gains and autopilot) on top of a nonlinear control loop. The simplified aerodynamic model is developed under the assumption of linear aerodynamics; for a given Mach (and other relevant quantities), aerodynamic coefficients are assumed to be constant, neglecting nonlinearities due to the angle of attack. Recalling the gain-scheduling discussion from chapter 2, the use of scheduling variables that change "quickly", such as α, is likely to give discontinuous gain surfaces and lead to implementation problems. Furthermore, as coefficients can evolve in discontinuous or abrupt ways near stall, tabulation of these phenomenons is difficult (as was shown), resulting in gain discontinuities. Instead, such nonlinearities are addressed by the nonlinear control law. In other words, linear gain design is done for low angle of attack behaviour, while the nonlinear loop will attempt to match the expected linear behaviour. As gain computation is done in real-time, it is necessary to use simple equations, hence the need for a good analytical understanding of the aircraft. The appeal of real-time computation is double; this avoids tabulation of gains and nonlinear laws will make use of aircraft aerodynamics in real-time. Nonetheless, the control law remains deterministic, in the sense that classical gain schedules could be computed if desired, a key feature for validation.

5.2 Pole placement algorithm

Effective and computationally simple pole placement is at the heart of the G* methodology. Although the example shown in [47] is for n_z tracking, it is easily generalized to other architectures. The main difference will be how to use the additional degrees of freedom of the

controller. This section will describe the application of the G* concept to the "main gains" of the C* architecture presented, while closed-loop objectives will be addressed in the next section. For implementation, matrix inversion should be done analytically to avoid iterative inversion, limiting the maximum size of the matrix that can be inverted to 3×3 or maybe 4×4[1]. Real-time iterative matrix inversion has been successfully implemented in the military aviation context [31]. Nonetheless, commercial aircraft validation processes are much more stringent which is likely to result in conservative hardware specifications. As cost reduction is a much more important aspect than in military applications, iterative inversion is seen as problematic.

5.2.1 Equivalent feedback filter

Pole placement is done considering an approximation of the system dynamics that are not from the rigid modes of the aircraft. The transfer function to approximate should include all non-aircraft dynamics, that is, actuators, ASE filters, sensors, etc. Doing so has a double advantage, as this both removes dependency w.r.t. aircraft-specific architecture, while also reducing the order of the system and allowing the use of analytical tools. These dynamics are approximated by a second-order filter as suggested by [47]. To do so, the magnitude response of the feedback transfer is matched at a given frequency and an octave lower. The solution to this problem is developed in annex C for completeness, although it is taken from [47]. If this failed to meet the desired performance, this task would be straightforward through structured H_∞ synthesis. As the slowest dynamic in the transfer is the acceleration filter pole at 10 rad/s, this frequency can be chosen as ω_1 and will approximate the feedback path well in the piloting frequency band.

The other portion of the equivalent feedback is the approximation of the loop delay (T_{eq}), the sum of the delays of all components, as a second-order pade which is given by :

$$F_{pade}(s) = \frac{\frac{{T_{eq}}^2}{12}s^2 - \frac{T_{eq}}{2}s + 1}{\frac{{T_{eq}}^2}{12}s^2 + \frac{T_{eq}}{2}s + 1} \tag{5.1}$$

5.2.2 Pole placement

Pole placement is done considering a generic architecture that is regulating the output E, an equivalent feedback quantity, as shown in figure 5.1. Note that K_{Ec} is a feed-forward gain, meaning it will have no effect on pole placement. The appeal of using E is that we

[1] Although analytical inversion of 4×4 matrices is possible, the resulting equations are significantly complexified.

can re-use the same generic computation functions for different objectives if desired. As all outputs have been defined from $\Delta\alpha$ in chapter 3, for short period control $E = \Delta\alpha$.

$$F_{eq.delay}(s) = \frac{\theta_2 s^2 - \theta_1 s + \theta_0}{\theta_2 s^2 + \theta_1 s + \theta_0} = \frac{N_\theta}{D_\theta} \tag{5.2a}$$

$$F_{eq.filter}(s) = \frac{1}{as^2 + bs + d} = \frac{1}{D_f} \tag{5.2b}$$

$$F_{A/C}(s) = \frac{1}{K_2 s^2 + K_1 s + K_0} = \frac{1}{D_{A/C}} \tag{5.2c}$$

The aircraft does not have zeros with this notation, but it will be shown that this is not problematic. With this architecture, the closed loop transfer function is given by:

$$\frac{E}{E_c} = \frac{\left(K_{\dot{E}} s^2 + \left(K_{Ec} + K_E\right) s + K_{Ei}\right) N_\theta}{D_\theta D_{A/C} D_f s - N_\theta \left(K_{\dot{E}} s^2 + K_E s - K_{Ei}\right)} \tag{5.3}$$

Considering a simplified architecture in C* with a PI, feedforward and q feedback as in figure 5.2, the aircraft closed-loop transfer function is developed in (5.4).

$$\frac{E}{C_c^*} = \frac{-\left(\left(K_{ff} + K_p\right) s + K_i\right) N_\theta}{D_\theta D_{A/C} D_f s - N_\theta \left(K_q \left(s - a_{11}\right) s + \left(K_p s + K_i\right) \left(N_{n_z} + \beta \left(s - a_{11}\right)\right)\right)} \tag{5.4}$$

with N_{n_z} the numerator of the n_z transfer given by (3.26). The K_d gain is neglected here to introduce the methodology. Note that the washout filter in the pitch rate path has also been neglected. Including this filter increases the order of the system, significantly complexifying equations and putting into question their practicality for real-time pole placement. Nonetheless, the corresponding equations are developed in annex E.

By identification of the feedback terms coefficients in powers of s:

$$K_i = \frac{-K_{Ei}}{N_{n_z} - \beta a_{11}} \tag{5.5a}$$

$$K_p = \frac{K_E + K_{\dot{E}} a_{11} - K_i \beta}{N_{n_z}} \tag{5.5b}$$

$$K_q = K_{\dot{E}} - \beta K_p \tag{5.5c}$$

which defines the pole placement process in function of the generic gains.

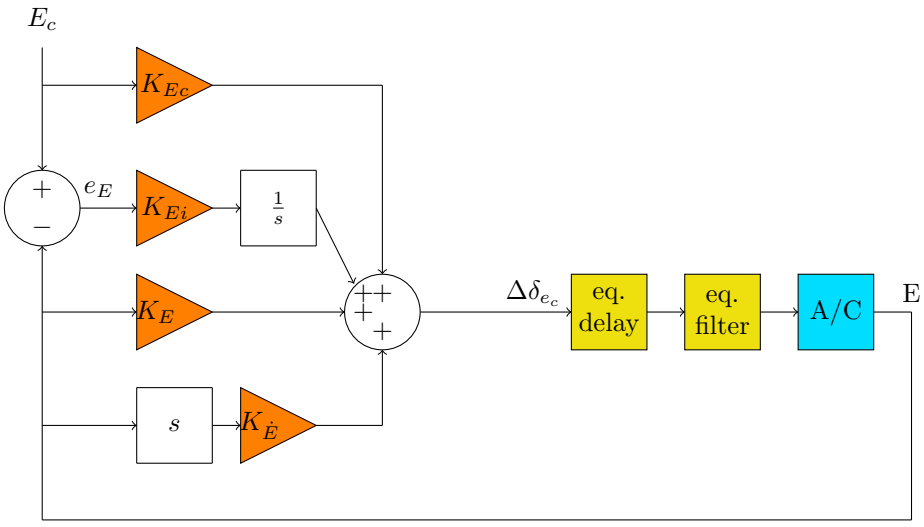


Figure 5.1 Generic G* architecture

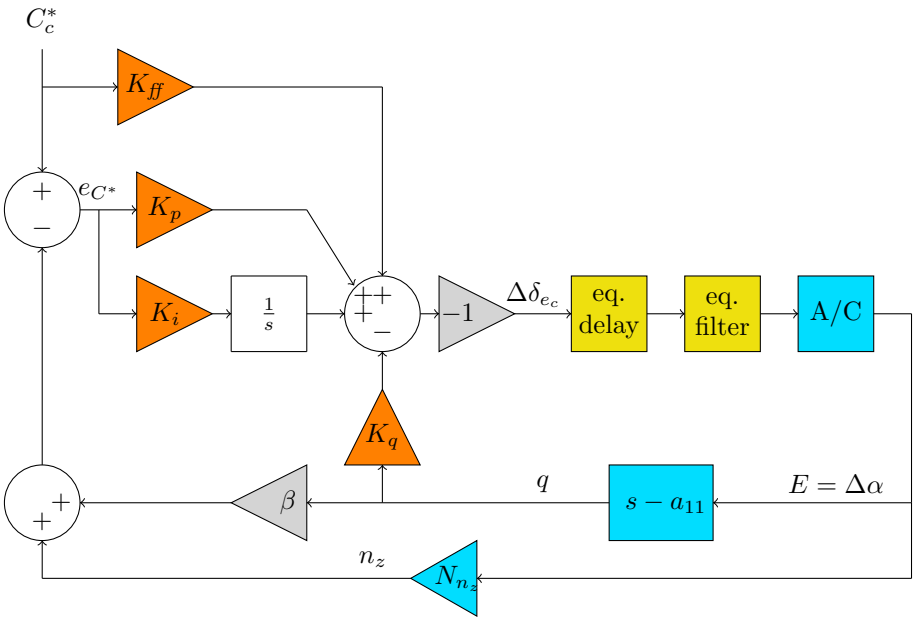


Figure 5.2 Simplified C* architecture for pole placement

The computation of the generic gain values is fully described in [47]. Developments are given here for the completeness of this thesis. Developing the denominator of the generic system from (5.3), it may be re-written as:

$$\begin{aligned} &T_7 s^7 + T_6 s^5 + T_5 s^5 + (T_4 - \theta_2 K_{\dot{E}}) s^4 + (T_3 - \theta_2 K_E + \theta_1 K_{\dot{E}}) s^3 \\ &+(T_2 + \theta_2 K_{Ei} + \theta_1 K_E - \theta_0 K_{\dot{E}}) s^2 + (T_1 - \theta_1 K_{Ei} - \theta_0 K_E) s + \theta_0 K_{Ei} + T_0 \end{aligned} \tag{5.6}$$

where :

$$T_7 = a\theta_2 K_2 \tag{5.7a}$$
$$T_6 = a(\theta_1 K_2 + \theta_2 K_1) + b\theta_2 K_2 \tag{5.7b}$$
$$T_5 = a(\theta_0 K_2 + \theta_1 K_1 + \theta_2 K_0) + b(\theta_1 K_2 + \theta_2 K_1) + d\theta_2 K_2 \tag{5.7c}$$
$$T_4 = a(\theta_0 K_1 + \theta_1 K_0) + b(\theta_0 K_2 + \theta_1 K_1 + \theta_2 K_0) + d(\theta_1 K_2 + \theta_2 K_1) \tag{5.7d}$$
$$T_3 = a\theta_0 K_0 + b(\theta_0 K_1 + \theta_1 K_0) + d(\theta_0 K_2 + \theta_1 K_1 + \theta_2 K_0) \tag{5.7e}$$
$$T_2 = b\theta_0 K_0 + d(\theta_0 K_1 + \theta_1 K_0) \tag{5.7f}$$
$$T_1 = d\theta_0 K_0 \tag{5.7g}$$
$$T_0 = 0 \tag{5.7h}$$

Once all T_i have been evaluated numerically for the current flight conditions, it is possible to match the desired closed-loop equation :

$$\left(x_4 s^4 + x_3 s^3 + x_2 s^2 + x_1 s^1 + x_0\right)\left(\mu_3 s^3 + \mu_2 s^2 + \mu_1 s^1 + \mu_0\right) \tag{5.8}$$

where x_i's are coefficients corresponding to the closed-loop dynamics that are not placed (equivalent delay or filter) and the μ_i's are coefficients of the three closed-loop poles that are placed. In other words, $\mu_3 s^3 + \mu_2 s^2 + \mu_1 s^1 + \mu_0$ are the imposed poles. Developing eq. 5.8 and comparing it to (5.2.2) gives a linear system in function of x_i and the generic gains.

$$\left[\begin{array}{c} T_7 \\ T_6 \\ T_5 \\ \hdashline T_4 \\ T_3 \\ T_2 \\ T_1 \\ T_0 \end{array}\right] = \left[\begin{array}{ccc:ccccc} \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_2 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_1 & \mu_2 & \mu_3 & 0 & 0 & 0 & 0 & 0 \\ \hdashline \mu_0 & \mu_1 & \mu_2 & \mu_3 & 0 & \theta_2 & 0 & 0 \\ 0 & \mu_0 & \mu_1 & \mu_2 & \mu_3 & -\theta_1 & \theta_2 & 0 \\ 0 & 0 & \mu_0 & \mu_1 & \mu_2 & \theta_0 & -\theta_1 & -\theta_2 \\ 0 & 0 & 0 & \mu_0 & \mu_1 & 0 & \theta_0 & \theta_1 \\ 0 & 0 & 0 & 0 & \mu_0 & 0 & 0 & -\theta_0 \end{array}\right] \left[\begin{array}{c} x_4 \\ x_3 \\ x_2 \\ \hdashline x_1 \\ x_0 \\ K_{\dot{E}} \\ K_E \\ K_{Ei} \end{array}\right] \tag{5.9}$$

As the matrix is lower block-triangular, it is straightforward to evaluate the values of x_2 through x_4 with the first three lines:

$$x_4 = \frac{T_7}{\mu_3} \tag{5.10a}$$

$$x_3 = \frac{T_6 - \mu_2 x_4}{\mu_3} \tag{5.10b}$$

$$x_2 = \frac{T_5 - (\mu_1 x_4 + \mu_2 x_3)}{\mu_3} \tag{5.10c}$$

It is therefore possible to reduce the system by sending these values on the left-hand side of (5.9). This results in a 5×5 linear system for x_0, x_1 and the gains, with the structure shown in (5.11).

$$\left[\begin{array}{c} T_4 - \mu_0 x_4 - \mu_1 x_3 - \mu_2 x_2 \\ T_3 - \mu_0 x_3 - \mu_1 x_2 \\ \hdashline T_2 - \mu_0 x_2 \\ T_1 \\ T_0 \end{array}\right] = \left[\begin{array}{cc:ccc} \mu_3 & 0 & \theta_2 & 0 & 0 \\ \mu_2 & \mu_3 & -\theta_1 & \theta_2 & 0 \\ \hdashline \mu_1 & \mu_2 & \theta_0 & -\theta_1 & -\theta_2 \\ \mu_0 & \mu_1 & 0 & \theta_0 & \theta_1 \\ 0 & \mu_0 & 0 & 0 & -\theta_0 \end{array}\right] \left[\begin{array}{c} x_1 \\ x_0 \\ \hdashline K_{\dot{E}} \\ K_E \\ K_{Ei} \end{array}\right] \tag{5.11}$$

From the two first lines, the variables x_1 and x_0 can be isolated as:

$$\begin{aligned} \left[\begin{array}{c} x_1 \\ x_0 \end{array}\right] &= \left[\begin{array}{cc} \mu_3 & 0 \\ \mu_2 & \mu_3 \end{array}\right]^{-1} \left[\begin{array}{c} T_4 - \mu_0 x_4 - \mu_1 x_3 - \mu_2 x_2 \\ T_3 - \mu_0 x_3 - \mu_1 x_2 \end{array}\right] - \left[\begin{array}{cc} \mu_3 & 0 \\ \mu_2 & \mu_3 \end{array}\right]^{-1} \left[\begin{array}{ccc} \theta_2 & 0 & 0 \\ -\theta_1 & \theta_2 & 0 \end{array}\right] \left[\begin{array}{c} K_{\dot{E}} \\ K_E \\ K_{Ei} \end{array}\right] \\ &= \left[\begin{array}{c} C_1 \\ C_2 \end{array}\right] - \left[\begin{array}{ccc} \frac{\theta_2}{\mu_3} & 0 & 0 \\ -\frac{\theta_2\mu_2 + \theta_1\mu_3}{\mu_3^2} & \frac{\theta_2}{\mu_3} & 0 \end{array}\right] \left[\begin{array}{c} K_{\dot{E}} \\ K_E \\ K_{Ei} \end{array}\right] \end{aligned} \tag{5.12}$$

with:

$$C_1 = \frac{T_4 - \mu_0 x_4 - \mu_1 x_3 - \mu_2 x_2}{\mu_3} \tag{5.13a}$$

$$C_2 = \frac{T_3 - \mu_0 x_3 - \mu_1 x_2}{\mu_3} - \frac{\mu_2 C_1}{\mu_3} \tag{5.13b}$$

By injecting (5.12) in the last three lines of (5.11), the following 3×3 system is obtained to find the gains:

$$\begin{bmatrix} T_2 - \mu_0 x_2 - \mu_1 C_1 - \mu_2 C_2 \\ T_1 - \mu_0 C_1 - \mu_1 C_2 \\ T_0 - \mu_0 C_2 \end{bmatrix} = \begin{bmatrix} \theta_0 - \mu_1 \frac{\theta_2}{\mu_3} + \mu_2 \frac{\theta_2\mu_2 + \theta_1\mu_3}{\mu_3^2} & -\theta_1 - \frac{\mu_2\theta_2}{\mu_3} & -\theta_2 \\ -\mu_0 \frac{\theta_2}{\mu_3} + \mu_1 \frac{\theta_2\mu_2 + \theta_1\mu_3}{\mu_3^2} & \theta_0 - \frac{\mu_1\theta_2}{\mu_3} & \theta_1 \\ \mu_0 \frac{\theta_2\mu_2 + \theta_1\mu_3}{\mu_3^2} & -\frac{\mu_0\theta_2}{\mu_3} & -\theta_0 \end{bmatrix} \begin{bmatrix} K_{\dot{E}} \\ K_E \\ K_{Ei} \end{bmatrix} \tag{5.14}$$

This system can be solved analytically through existing 3×3 matrix inverse solutions. Although autopilots are not considered in this document, their gains could be computed with the same [2] or a similar generic function, starting from the closed-loop system denominator, which is known from x_i and μ_i.

5.2.3 Inclusion of a pre-defined K_d

It seems difficult to use the K_d gain for pole placement, as it is not clear what should be done with this additional degree of freedom. Furthermore, it is preferable to avoid 4×4 matrix inversion if possible to avoid needless increases in hardware costs. For these reasons, a constant value of K_d will be considered for pole placement. Adding the pseudo-derivator and the K_d gain in figure 5.2 leads to the transfer function :

$$\frac{E}{{C^*}_c} = \frac{-\left(\left(K_{ff} + K_p\right) D_d s + K_d s^2 + K_i D_d\right) N_\theta}{D_\theta D_{A/C} D_f D_d s - N_\theta \left(K_q \left(s - a_{11}\right) D_d s + \left(K_d s^2 + \left(K_p s + K_i\right) D_d\right) \left(N_{n_z} + \beta \left(s - a_{11}\right)\right)\right)} \tag{5.15}$$

where $D_d = 0.1s + 1$.

From this equation, K_{ff} can be identified in function of $T_{K_{ff}}$, K_i, K_p and K_d, where $T_{K_{ff}}$ is the desired time constant for one of the zeros created. Setting the numerator equal to 0, we get:

$$\left(0.1s^2 + s\right)\left(K_{ff} + K_p\right) + K_d s^2 + K_i \left(0.1s + 1\right) = 0 \tag{5.16}$$

Defining $\bar{K_p} = K_{ff} + K_p$ results in:

$$(K_d + 0.1\bar{K_p})s^2 + (\bar{K_p} + 0.1K_i)s + K_i = 0 \tag{5.17a}$$

$$z_i = \frac{-\bar{K_p} - 0.1K_i \pm \sqrt{\left(\bar{K_p} + 0.1K_i\right)^2 - 4K_i\left(K_d + 0.1\bar{K_p}\right)}}{2(K_d + 0.1\bar{K_p})} \tag{5.17b}$$

[2] If they do not add poles to the system.

This yields the zero time constants:

$$T_{K_{ff},i} = \frac{2\left(K_d + 0.1\bar{K}_p\right)}{\bar{K}_p + 0.1K_i \mp \sqrt{\left(\bar{K}_p - 0.1K_i\right)^2 - 4K_dK_i}} \tag{5.18}$$

From this equation, it is clear that two zeros will be created. Furthermore, the second zero will be implicitly defined from the first one. The intent is therefore to control the slowest zero, assuming the other one will be negligible. Note that if $K_d = 0$, one zero can be placed with K_{ff} while the second one has a time constant of 0.1 s, cancelling the pseudo-derivator pole. Manipulating the zero time constants to obtain K_{ff} in function of $T_{K_{ff}}$, we get :

$$\pm\sqrt{\left(\bar{K}_p - 0.1K_i\right)^2 - 4K_dK_i} = \frac{2\left(K_d + 0.1\bar{K}_p\right)}{T_{K_{ff}}} - (\bar{K}_p + 0.1K_i) \tag{5.19a}$$

$$\left(\bar{K}_p - 0.1K_i\right)^2 - 4K_dK_i = \tag{5.19b}$$

$$\frac{4\left(K_d + 0.1\bar{K}_p\right)^2}{T_{K_{ff}}^2} - \frac{4}{T_{K_{ff}}}\left(K_d + 0.1\bar{K}_p\right)\left(\bar{K}_p + 0.1K_i\right) + \left(\bar{K}_p + 0.1K_i\right)^2 \tag{5.19c}$$

$$0 = \frac{\left(K_d + 0.1\bar{K}_p\right)^2}{T_{K_{ff}}^2} - \frac{\left(K_d + 0.1\bar{K}_p\right)\left(\bar{K}_p + 0.1K_i\right)}{T_{K_{ff}}} + K_i\left(K_d + 0.1\bar{K}_p\right) \tag{5.19d}$$

Assuming $K_d + 0.1\bar{K}_p \neq 0$, which is always true for positive gains, this equation is simplified to :

$$0 = \frac{\left(K_d + 0.1\bar{K}_p\right)}{T_{K_{ff}}^2} - \frac{\left(\bar{K}_p + 0.1K_i\right)}{T_{K_{ff}}} + K_i \tag{5.20a}$$

$$\bar{K}_p\left(0.1 - T_{K_{ff}}\right) = 0.1K_iT_{K_{ff}} - K_iT_{K_{ff}}^2 - K_d \tag{5.20b}$$

If we presume that $T_{K_{ff}} \neq 0.1$, we finally get:

$$K_{ff} = \frac{10K_d + 10K_i{T_{K_{ff}}}^2 - K_iT_{K_{ff}}}{10T_{K_{ff}} - 1} - K_p \tag{5.21}$$

The assumption that $T_{K_{ff}} \neq 0.1$ is valid for computation of K_{ff}. Fast forwarding to application of the methodology, one has $\psi > 1.5$ (see eq. 3.61) for all designs iterations made, leading to $T_{K_{ff}} > 0.3$ assuming that $\omega_{CL} < 5$ rad/s. While equation 5.15 can be used to find the zeros added to the system precisely, it is not practical for placing the three desired poles. Indeed, the multiplication by $0.1s + 1$ significantly increases the complexity of equations.

Therefore, a pure derivator is considered for simplicity:

$$\frac{E}{{C^*}_c} = \frac{-\left(K_d s^2 + \left(K_{ff} + K_p\right) s + K_i\right) N_\theta}{D_\theta D_{A/C} D_f s - N_\theta \left(K_q \left(s - a_{11}\right) s + \left(K_d s^2 + K_p s + K_i\right) \left(N_{n_z} + \beta \left(s - a_{11}\right)\right)\right)} \tag{5.22}$$

As K_d is pre-defined, its contribution can be included in the generic system. After developing $D_\theta D_{A/C} D_f s - N_\theta \left(N_{n_z} + \beta \left(s - a_{11}\right)\right) K_d s^2$ into a polynomial of seventh order:

$$T_7 s^7 + T_6 s^6 + T_5 s^5 + T_4 s^4 + T_3 s^3 + T_2 s^2 + T_1 s + T_0 \tag{5.23}$$

one obtains by identification:

$$T_7 = a\theta_2 K_2 \tag{5.24a}$$

$$T_6 = a(\theta_1 K_2 + \theta_2 K_1) + b\theta_2 K_2 \tag{5.24b}$$

$$T_5 = a(\theta_0 K_2 \theta_1 K_1 + \theta_2 K_0) + b(\theta_1 K_2 + \theta_2 K_1) + d\theta_2 K_2 - K_d \beta \theta_2 \tag{5.24c}$$

$$T_4 = \tag{5.24d}$$

$$a(\theta_0 K_1 + \theta_1 K_0) + b(\theta_0 K_2 \theta_1 K_1 + \theta_2 K_0) + d(\theta_1 K_2 + \theta_2 K_1) - K_d(\theta_2[N_{n_z} - \beta a_{11}] - \theta_1 \beta)$$

$$T_3 = a\theta_0 K_0 + b(\theta_0 K_1 + \theta_1 K_0) + d(\theta_0 K_2 \theta_1 K_1 + \theta_2 K_0) - K_d(\theta_0 \beta - \theta_1[N_{n_z} - \beta a_{11}]) \tag{5.24e}$$

$$T_2 = b\theta_0 K_0 + d(\theta_0 K_1 + \theta_1 K_0) - K_d(\theta_0[N_{n_z} - \beta a_{11}]) \tag{5.24f}$$

$$T_1 = d\theta_0 K_0 \tag{5.24g}$$

$$T_0 = 0 \tag{5.24h}$$

This allows to simply replace the equations for T_i in (5.14) without other modifications.

5.2.4 Tuning of Kd

The derivative gain cannot easily be used to place a pole and it is not strictly needed to achieve satisfactory performance (with mass and cg estimates, setting $K_d = 0$ gives good results). Nonetheless, it seems plausible that this gain could be used to improve aircraft performance. Furthermore, existing gains of the target aircraft need the K_d gain and some H_∞ synthesis attempts without this gain have failed to meet the required loading robustness. Expressions developed in the previous section allow to express the needed K_{ff} by taking K_d (and the pseudo-derivator) into account, meaning this gain will not impact the zeros. Following previous assumptions, a pure derivator will be considered. As previous equations place the 3 SP poles by taking K_d into account, these poles will not be affected by K_d. In other words, only the equivalent filter and delay poles will move in the complex plane. The corresponding root locus is shown in figure 5.3. From this figure, it can be seen that (with the pure derivative approximation):

- K_d does not affect the three SP poles
- Two out of the four equivalent poles are negligible (they correspond to the pade).
- Low K_d values result in slow equivalent filter poles, while high values will recombine the equivalent filter poles, slowing them down and reducing their damping.

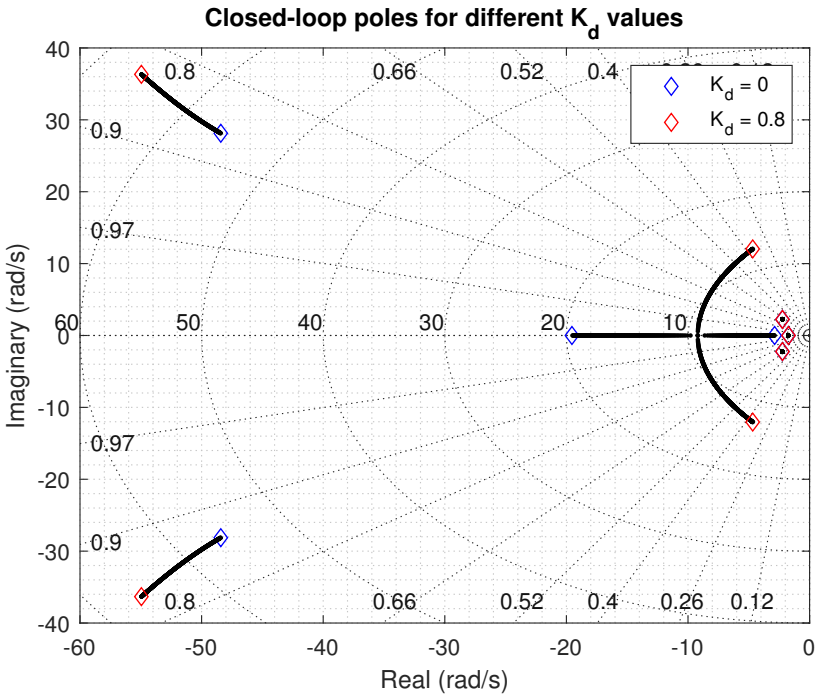


Figure 5.3 Typical root locus for K_d

The slow real pole can be non-negligible w.r.t. the SP. This results in very low dropbacks (potentially < -1) for high speeds (where the SP is fast). This is because this pole is neglected by the third-order approximation that will be used extensively. The idea is therefore to use K_d to bring the filter poles to an optimal position in terms of speed, considered to be where a double pole is obtained, ensuring that they are negligible.

Given the closed-loop poles of the equivalent feedback (fourth-order polynomial) as a function of the derivative gain, the optimal value of K_d could potentially be selected to achieve the desired double pole. Nonetheless, this requires finding the zeros of a fourth-order polynomial in real-time. Although this could be possible through the Ferrari method, a simpler alternative is considered. The optimal value of K_d will be computed by neglecting the equivalent delay, resulting in a second-order polynomial. In the case of aircraft where the equivalent delay is not negligible, potential solutions could be to find which component is more problematic (delay or filter) or to compute a second-order equivalent filter that would also approximate the delay to some extent. Setting $\theta_2 = \theta_1 = 0$ and θ_0=1 in (5.24a) and using the same methodology, we get:

$$\left(x_2 s^2 + x_1 s^1 + x_0\right)\left(\mu_3 s^3 + \mu_2 s^2 + \mu_1 s^1 + \mu_0\right) = \tag{5.25a}$$

$$T_5 s^5 + T_4 s^4 + T_3 s^3 + \left(T_2 - K_{\dot{E}}\right) s^2 + \left(T_1 - K_E\right) s + K_{Ei} \tag{5.25b}$$

Developing the terms needed to compute $x_2 s^2 + x_1 s^1 + x_0$ yields:

$$T_5 = x_2 \mu_3 = K_2 a \tag{5.26a}$$

$$T_4 = x_2 \mu_2 + x_1 \mu_3 = K_1 a + K_2 b \tag{5.26b}$$

$$T_3 = x_2 \mu_1 + x_1 \mu_2 + x_0 \mu_3 = K_0 a + K_1 b + K_2 d - K_d \beta \tag{5.26c}$$

As filter coefficients are decoupled from other gain values, the coefficients x_i's are given by:

$$x_2 = \frac{K_2 a}{\mu_3} \tag{5.27a}$$

$$x_1 = \frac{K_1 a + K_2 b - x_2 \mu_2}{\mu_3} \tag{5.27b}$$

$$x_0 = \underbrace{\frac{K_0 a + K_1 b + K_2 d - x_2 \mu_1 - x_1 \mu_2}{\mu_3}}_{x_{0,0}} - \frac{\beta}{\mu_3} K_d \tag{5.27c}$$

Solving this equation for a null discriminant (double poles):

$$0 = {x_1}^2 - 4x_2\left(x_{0,0} - \frac{\beta}{\mu_3}K_d\right) \tag{5.28}$$

leads to:

$$K_d = \frac{\mu_3}{\beta}\left(x_{0,0} - \frac{{x_1}^2}{4x_2}\right) \tag{5.29}$$

This approximation is of sufficient precision to improve performance significantly both with and without mass/cg estimates while remaining reasonably simple. Furthermore, the obtained values also happen to be close to the schedules defined with classical methods at high speed (where K_d is relevant).

5.3 Meeting linear performance requirements by pole placement

Equations developed in the last section enable designers to control dominant poles and zeros as desired. This leads to the problem of using this capability to meet good handling qualities. The linear performance will be tuned using the third-order model and equations introduced in section 3.5.1, neglecting the effect of lead-lag filters in series with the C* tracking loop. These filters, if they are needed, will be defined afterwards to address specific problems of the C* tracking loop. It is easy to justify the use of the third-order reference model, given that K_d can be used to limit the effect of higher-order dynamics, as shown in the last section. Therefore, for each flight point, the designer has four parameters at his disposition to meet linear HQs: $T_{K_{ff}}$, ζ_{CL}, ω_{CL} and γ. The effect of these parameters can be studied qualitatively.

5.3.1 Effect of the four parameters

- ω_{CL} can be well understood through the analogy of a second-order system. Higher values degrade stability margins and OLOP, in exchange for higher bandwidths, lower settling times and other characteristics of a faster system. This parameter should be increased until limited by the OLOP or stability margins.

- $T_{K_{ff}}$ is extremely important, as its value will affect nearly all HQs, the main exception being stability margins. Increasing this parameter increases the dropback, overshoots, bandwidths, *PRS* and degrades the OLOP. Higher values also result in a higher feed-forward gain (This is easily understood for $K_d = 0$).

- γ corresponds to the ratio of the integrator pole to the real part of the SP poles. As such, higher values increase K_i, leading to faster responses and lower phase margins. In

general, increasing γ for a constant value of $T_{K_{ff}}$ results in qualitatively similar results to increases of $T_{K_{ff}}$ for a constant γ. In general, designs with high γ values (e.g. ≈ 1) were found to have high *PRS* gains.

- ζ_{CL} has a limited effect on handling qualities so long as its value is "adequate". Too low values result in large settling times due to large undershoots. Furthermore, taking $\zeta_{CL} < \zeta_{OL}$ can result in negative K_q values. As mentioned in H_∞ synthesis, lower values tend to increase the γ bandwidth and reduce OLOP. Overall, this parameter should be set as low as settling times and K_q permit while respecting a minimum damping.

From structured H_∞ synthesis experience and preliminary attempts made to satisfy HQs by "trial and error" with the G* methodology, a large portion of HQs are expected to be met as long as the aircraft meets requirements in *Drb*, BW_θ, n_z overshoot and stability margins.

For a large portion of the flight envelope, the pitch rate response benefits from large phase advances (high $T_{K_{ff}}$). This stops being the case when T_{θ_2} is large, as additional phase advance increases the *Drb* (and *PRO*) beyond acceptable limits, while some phase advance is still required to get good bandwidths. On the other hand, a slow zero leads to higher n_z overshoots while increasing the *Drb* which can be low at high speeds, hence the clear trade-off between these objectives.

Finally, the closed-loop frequency should be limited by the OLOP and stability margins. It was mentioned by experimented designers during the H_∞ synthesis development that responses that pass margins requirements generally pass the OLOP (especially if better margins than required are kept). This seemed to remain true for most results with flap 0 for H_∞ synthesis and preliminary G* trials (especially if K_d is low). As margins are much easier to compute than OLOP, this quantity will be used, justified by the fact that if the OLOP is not passed, S.M. could be increased to fix the problem or that a lead-lag filter can be added to address the OLOP.

This potentially leaves the *PRS* free, which is more difficult to address due to its dependency on the system's steady-state gain, although it was never problematic in G* design. With these four HQ requirements, the problem seems to be fully defined and there are four parameters to impose on the closed-loop system ($T_{K_{ff}}$, ω, γ, ζ). In section 3.5.1, equations were developed to evaluate the dropback and n_z overshoot. Studying the stability margins or bandwidths analytically is impractical, as the combination of arctangent functions leaves little room for analytical results that can be used online. Furthermore, stability margins will

be highly dependent on the equivalent feedback dynamics, demanding at least a seventh-order approximation to have meaningful results.

Because of the real-time nature of the G* architecture, the methodology should be sequential, in the sense that the parameters should be selected one after the other. Although finding optimal values for each parameter simultaneously through the resolution of some nonlinear equation system would be optimal, this would result in a high computational load. Therefore, the proposed design process is to let the designer choose ζ_{CL}, due to this parameter's small effect on HQs. Then, ω_{CL} must be selected such as to meet the desired stability margins. This will be done by a numerical optimization offline as shown in section 5.3.2, allowing to find a simplified approximate solution that can easily be used online. Given the complexity of the overshoot approximation from (3.59), selecting $T_{K_{ff}}$ in function of ω_{CL} for "nominal" values of γ and ζ_{CL} seems like the approach with the least complexity. To do so, the designer must choose a value of ψ, limiting the maximum overshoot. Figure 5.4 shows that, within an "acceptable" range of values of γ and ζ, the overshoot may be bounded by considering the maximum value of γ. This figure was generated for a 5% overshoot, but this behaviour remains true for the range of interest. Finally, the *Drb* will be enforced through either ψ or γ as described in section 5.3.3. This leaves BW_θ free, but the minimum bandwidths will be controlled through the range of values that ψ or γ can take.

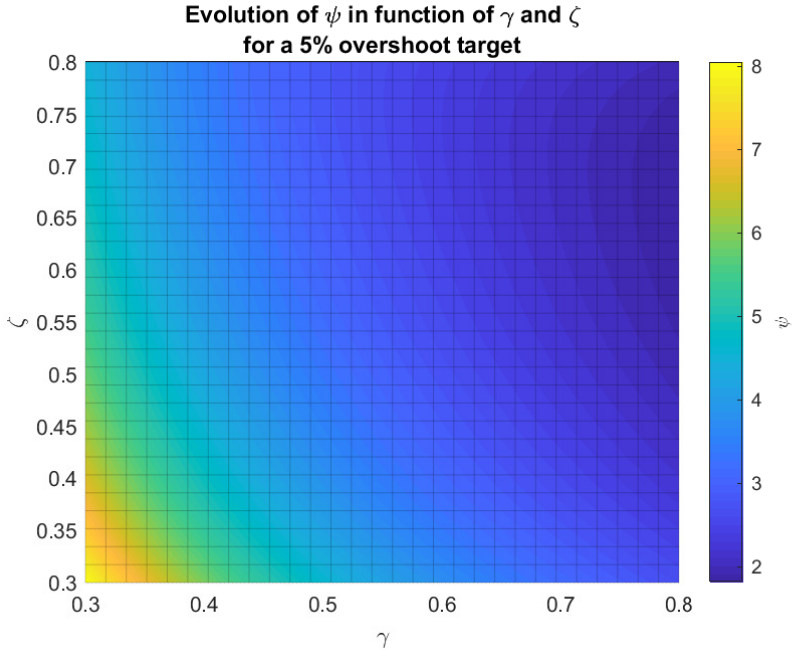


Figure 5.4 Evolution of ψ for a 5% overshoot target

5.3.2 Meeting a stability margin target

As mentioned, finding a solution for the maximum value of ω_{CL} seems unlikely analytically, given that it is not possible to neglect feedback dynamics. Therefore, the idea is to solve this problem numerically offline and approximate the solution throughout the envelope to evaluate it in real-time. Conveniently, the H_∞ framework has a constraint for disk stability margins, allowing to use this optimization method. Furthermore, finding a solution valid within a range of γ values may be necessary, hence the appeal of a robust technique.

To evaluate stability margins, a few hypotheses will be used to simplify the design process and reduce the dependency of the limit with regard to other parameters used:

- The feedforward path is assumed to have no effect on stability margins, allowing to remove it from the synthesis. This assumption remains true when considering only the C* tracking loop.
- The closed-loop damping ratio is set by the designer. If this is not the case, bounds on this parameter should be known.
- Bounds on γ are assumed to be known.
- The proportional and derivative gains are assumed to be feedback gains rather than on the C* error. This has no effect on poles but removes zeros that would be dependent on the gains.

These assumptions simplify the closed-loop to figure 5.5. In this figure, K_d is neglected, but it can be included as a feedback gain without other changes. Fast-forwarding to nonlinear design, the ANL law that will be defined (and less importantly the APU law) should be included in this process, as it is linear for a given flight condition and can significantly affect stability margins.

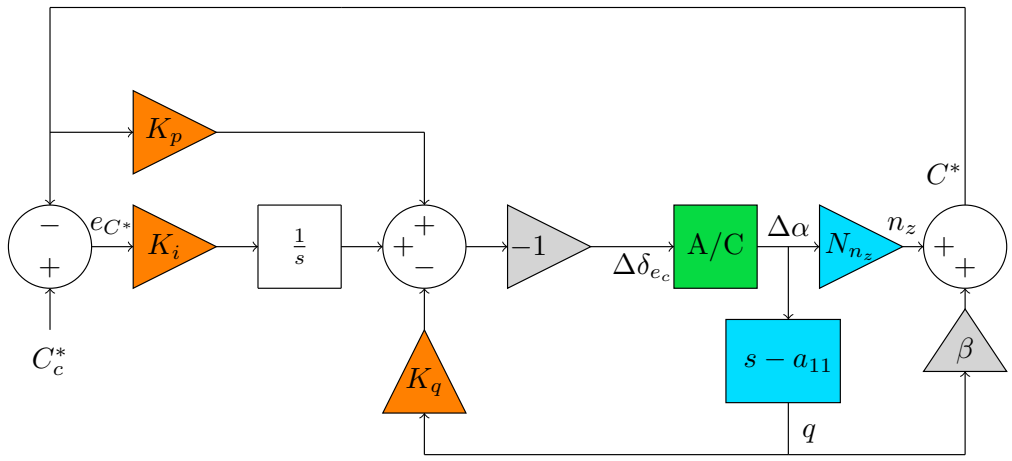


Figure 5.5 Modified architecture for H_∞

Even with these simplifications, a zero remains in closed-loop. If the synthesis is formulated for the C* output (with a unit gain), the transfer function is given by:

$$\frac{C^*}{C_c^*} = \frac{\gamma\zeta\omega^3 \left(\frac{\beta s}{-a_{11}\left(\beta + \frac{V_T \pi}{g180}\right)} + 1 \right)}{\left(s^2 + 2\zeta\omega s + \omega^2\right)\left(s + \gamma\zeta\omega\right)} \tag{5.30}$$

This defines all that is needed for the reference model, as only ω remains unknown. The synthesis should perform three objectives:

1. Limit stability margins above the design requirement (hard constraint).
2. Match the reference model within the piloting frequency band to some extent (hard constraint). This can be done using the first W_1 defined in the H_∞ chapter for local synthesis. Here, `RelGap` can be a constant as the intent is to put the system on stability margin limits. Furthermore, if some higher-order dynamics are to be allowed, a real pole may be added to the reference model as a tuneable parameter (σ_4). It is then possible to ensure that the real pole is negligible compared to the short period by $\sigma_4/\sigma_{SP} > K$, where K is a positive constant (e.g. 5). Adding a hard constraint such as $K\zeta\omega/\sigma_4 < 1$ ensures this is respected.
3. Maximise ω (soft constraint)

This H_∞ synthesis problem can be formulated as in figure 5.6. Note that a damping requirement could be added to enforce a minimum damping, but that γ is only enforced through reference model matching.

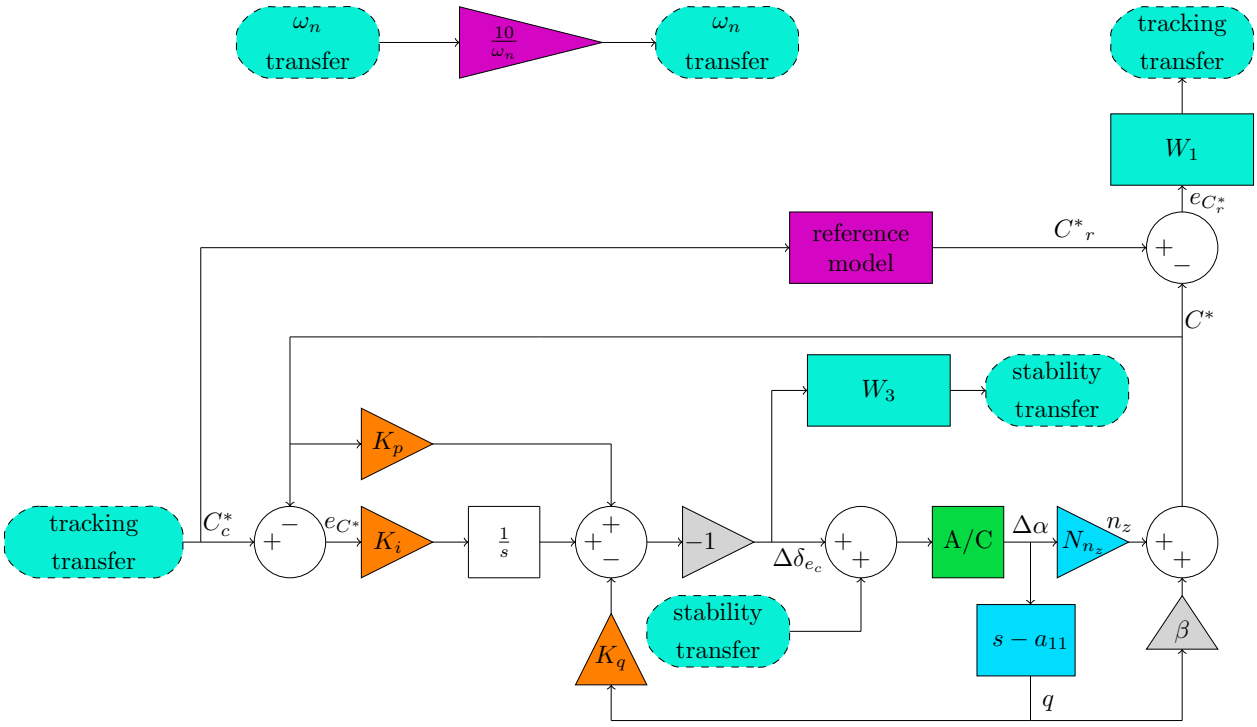


Figure 5.6 H_∞ problem definition

From the transfer function given in equations 3.13a and 3.13b, it is clear that to adequately span the envelope, five variables must be considered, namely : a_{11}, a_{21}, a_{22}, b_2 and V_T. As a_{ij} are not independent of each other, nor of V_T, they are either obtained from linearized data or the aerodynamic model. Individual syntheses were performed for a set of flight points and loadings that accurately represents the envelope, although scheduling of the maximum ω_{CL} could be attempted directly. By displaying the resulting ω_{CL} in function of these parameters (two at a time), the individual effect of most of these parameters did not seem strong. Instead, it is more interesting to display the resulting frequency in function of the open-loop frequency and the open-loop damping ratio. This is done in figure 5.7 for $\gamma_{CL} = 0.6$ and $\zeta_{CL} = 0.6$, with the addition of a second-order fit in ω_{OL} and ζ_{OL}. Note that a few points are negative. The synthesis failed to meet the required margins (due to a tracking requirement that was too strict) in some cases, which were flagged this way. Such a polynomial is easily evaluated in real-time. Another option could be to create a table to interpolate in real-time. Although less appealing, this would allow easier fine-tuning than a polynomial. When there is no scheduling with regards to mass and cg, a polynomial in classical scheduling variables (e.g. dynamics pressure and altitude) could also be used.

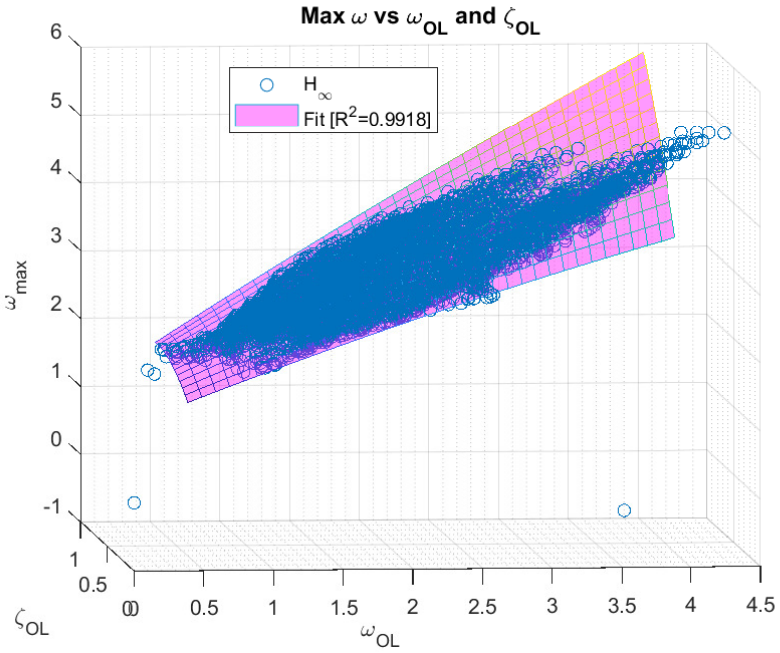


Figure 5.7 New fit of the maximum ω

5.3.3 Meeting a dropback target

For a fixed ψ value, the dropback is given by eq. 3.44:

$$Drb_{tgt} = \frac{\psi}{\omega_{CL}} + T_{\theta_2} - \frac{2\zeta + \frac{1}{\gamma\zeta_{CL}}}{\omega_{CL}} \tag{5.31}$$

Conditions where $T_{\theta_2} > 3$ are especially problematic as they will often be outside level 1 *Drb* as mentioned for the acceptable space from H_∞ synthesis. As a maximum value for ψ is already defined by the overshoot, it is interesting to use ζ and/or γ to meet the desired *Drb*. A solution based on imposing $\gamma = \zeta$ was developed, but rejected as it is not straightforward to study the effect of γ and ζ on overshoots while doing so for a constant ζ and variable γ is relatively simple as was shown in 5.4. Although the partial differentiation of the *OS* w.r.t. γ can be evaluated, the result is of limited practicality. The reduction in γ required to meet a dropback target is given by isolating γ from eq. 5.31:

$$\gamma = -\frac{1}{\zeta\left(\left[Drb_{tgt} - T_{\theta_2}\right]\omega_{CL} - \psi + 2\zeta\right)} \tag{5.32}$$

In practice, γ may be bounded within a maximum and minimum value to ensure adequate handling qualities. Alternatively, ψ can be varied for a constant value of γ to meet a target:

$$\psi = \omega_{CL}\left(Drb_{tgt} - T_{\theta_2}\right) + 2\zeta_{CL} + \frac{1}{\gamma\zeta_{CL}} \tag{5.33}$$

With this equation, ψ can be limited below the value computed from the overshoot target and above a minimum value (e.g. 0% overshoot) to meet minimum bandwidths. Overall, both approaches to dropback limiting are equivalent, although the ψ approach was found to be more intuitive and avoids the iterative process necessary for Δ_{Drb} (later defined in (5.35)). Furthermore, as no robustness is needed in ω_{CL}, it is likely that the limit will be more aggressive. It is interesting to note that some designers [5], [44] suggest the cancellation of the integrator pole (e.g. with $T_{K_{ff}}$), resulting in an under-constrained problem from a linear point of view. With the methodology defined, the under-constrained character expresses itself through the capacity to control the dropback and n_z overshoot through both ψ and γ_{CL}. Furthermore, when evaluating the integrator time constant for the values given in chapter 6, $T_{\int} = \frac{1}{\gamma\zeta_{CL}\omega_{CL}} \approx \frac{2.53}{\omega_{CL}}$ which is very close to $T_{K_{ff}} \in \frac{[2.25,2.55]}{\omega_{CL}}$. It is reassuring that, although different design objectives were used, the same classical result is obtained. Nonetheless, full pole-zero cancellation was found to be less practical as meeting low overshoots requires a high ζ_{CL}, which can lead to poor OLOP and other limitations, on top of the inability to explicitly control the dropback. Furthermore, as the pole-zero cancellation is not exact on the full system, adjusting γ becomes necessary to address pole placement inaccuracies' effects on the overshoot, limiting the values that this parameter can take.

Finally, having a discontinuous dropback target can create odd behaviours in gain surfaces, especially K_{ff} and K_q. An effective way to avoid such discontinuities is to define the Drb in function of T_{θ_2}, which describes the capacity of the aircraft for a centered (25%) cg position (dependency on the dynamic pressure, C_{L_α}, mass, etc.). As mentioned, values of $T_{\theta_2} > 3$ are extremely difficult to work with, which is why aiming for the level 1 limit is adequate. On the other hand, aiming for a value in the middle of the 1^* envelope is adequate when $T_{\theta_2} \approx 0.5$ (minimum value reached by the target aircraft). As it is preferable to stay closer to the 1^* limit (and based on H_∞ experience with the size of the acceptable zone in function of T_{θ_2}), aiming for 0.5 around $T_{\theta_2} = 2$ yields good results. Finally, to ensure the Drb is always increasing with the value of T_{θ_2}, it is possible to enforce a null derivative at the minimum value. This gives four linear constraints, allowing to use the third-order polynomial:

$$Drb_{25} = 0.0764T_{\theta_2}^3 - 0.0738T_{\theta_2}^2 + 0.0164T_{\theta_2} + 0.1507 \tag{5.34}$$

Although the exact polynomial may be changed between designs, the process remains simple and the polynomial is easily evaluated.

5.3.4 Design process without knowledge of the mass and cg of the aircraft

The problem of design without mass and cg is addressed first, as it will require some additional developments which can then be used even when this information is known. For conciseness, the design problem without loading information will be referred to as robust, while nominal design case will be used when such information is available. The loss (or lack) of information on the loading of the aircraft puts into question the relevance of the G* methodology, as this implies uncertainties on stability derivatives that are much larger than the ones due to aircraft model inaccuracy. On the other hand, it is interesting to verify how much performance the methodology retains, as the loss of mass and/or cg estimates remains a failure case even for aircraft where they are normally available. Assuming the performance is too poor to be of use, this would require the design of an additional robust law (either through H_∞ or classical means) on top of the G* law, ultimately negating many methodological advantages w.r.t. design effort. Conversely, if performance is adequate, it may offer an alternative to H_∞ and classical methods that would be directly applicable to current aircraft.

As can be expected, robust design is much more difficult due to the inherent performance trade-offs required. On the target aircraft, high-order mode degradation is much more pronounced than in the nominal case, leading to excessively unsatisfactory dropback values at high speeds. Furthermore, the n_z overshoot approximation developed is only valid for specific values of ζ_{CL}, γ and, more importantly, ω_{CL}. This is problematic, as these values cannot be known (without significant computational efforts for online applications) for all loading values, leading to an excessively large degradation of the overshoot limiting. Considering the tabulation of aerodynamic variables done in earlier sections, removing the mass, cg and α (linear aerodynamics approximation) dependencies results in gains that are only function of classical scheduling variables, such as $\bar{q}$, h or M for ISA conditions. Note that these approximations render the engine effect weak.

Selection of the design loading

The fundamental problem of robust design is one of choosing the design loading (inertia, mass and cg). While defining a completely new methodology (e.g. a robust pole placement algorithm) would be possible, this would be at the cost of dissimilarity with the nominal law

(limiting many methodological advantages of G*) and it is a reasonable assumption that this would be done at the cost of additional complexity. To avoid such problems, it is decided to keep the same pole placement algorithm by feeding the control law a "synthetic" or "design" loading, which is chosen offline. Validation will show that doing so yields excellent results, although theoretical proofs of stability throughout the loading envelope remain to be developed.

The n_z overshoot approximation relies on accurate knowledge of other closed-loop parameters. Computing such parameters online for the design loading requires the solutions (poles) of the characteristic equation of the closed-loop transfer function. Even neglecting feedback dynamics, this requires the zeros of a third-order polynomial (eq. 3.62), solutions that exist analytically through Cardano's method, but are excessively unwieldy. Instead, if the loading that results in the worst-case overshoot is known, this loading may be used for design. It was shown in section 3.5.1 that this corresponds to the aft cg limit, most likely at light weight. Low inertia conditions also generally have worst stability margins for constant gains, as higher gains will affect them more than conditions at higher inertia. On the other hand, for the nonlinear performance of the ANL law (to be defined), considering higher inertia improves load factor limiting, as elevator commands will be larger (smaller b_2 coefficient at high inertia values). In other words, if the nonlinear law must use the same stability derivatives as the linear law, a trade-off must be made. Otherwise, linear design should be done for light aircraft weight.

Meeting the *Drb* requirement

Using the loading that results in the largest load factor overshoots will ensure that this HQ is met throughout the envelope. On the other hand, the worst-case dropback at high speeds (where the lowest values are achieved) is unlikely to be within the design limits. Looking at eq. 3.51a, the main driver for the degradation of the *Drb* w.r.t. cg position is C_{m_α}, which is tabulated as $C_{m_{\alpha,AC}} - \bar{\Delta}_x C_{Z_\alpha}$. As C_{Z_α} is negative, the minimum *Drb* (minimum C_{m_α}) will occur at the minimum $\bar{\Delta}_x$, therefore at forward cg positions. This is problematic, as the designer will only have a tight HQ control on aft cg positions. Nonetheless, it was mentioned throughout H_∞ synthesis that enforcing a behaviour for a given loading would implicitly enforce a different behaviour for other loadings due to the absence of the related scheduling variables. Therefore, if the *Drb* degradation ($\Delta_{Drb} = Drb_{AFT} - Drb_{FWD}$) can be computed, meeting $Drb_{AFT} = Drb_{min} + \Delta_{Drb}$, where Drb_{min} is a design criteria, will ensure adequate *Drb* values for all loadings. Although the worst case could be computed precisely through eq. 3.49, a simpler alternative is considered to avoid additional interpolations. Considering

only cg effects (neglecting the mass effect on Δ_{Drb}, which gives good results for the target aircraft) and the previous approximations on the aerodynamic coefficients (constant C_{L_α}, C_{m_q} and $C_{m_{\delta_e}}$) :

$$\Delta_{Drb} = Drb_{AFT} - Drb_{FWD} = \frac{mV_T}{\bar{q}SC_{L_\alpha}} \frac{C_{m_{\alpha,FWD}} - C_{m_{\alpha,AFT}}}{C_{m_{\delta_e}} a_0 K_i} = \frac{mV_T}{\bar{q}SC_{L_\alpha}} \frac{C_{Z_\alpha} (\bar{x}_{FWD} - \bar{x}_{AFT})}{C_{m_{\delta_e}} a_0 K_i} \tag{5.35}$$

In practice, slightly increasing Drb_{min} will allow to accommodate the small inaccuracies of this approximation. The two dropback-limiting approaches defined earlier may be used with this correction. If using the γ version (see eq. 5.32), a large enough range of values for γ should be considered when computing the ω_{CL} limits to ensure stability. Furthermore, ψ should be defined for the maximum value of γ that the system can use to ensure the overshoot criterion is respected. This leaves a single problem for the use of this criterion: the knowledge of K_i. Indeed, γ defines the position of the integrator pole, but its value is now dependent on the integrator gain. Three options were considered to solve this problem :

- Using the value of K_i computed at the previous iteration (in real-time gain computation) for the *Drb* correction. This algorithm converges very quickly (within 2 or 3 iterations from the minimum K_i value in the envelope), suggesting no real-time problems. Numerous nonlinear simulations were performed with this option without any issues. Nonetheless, validation of this architecture may be more difficult.
- Pre-computing the values of K_i and using this schedule (which can be coarse) as an estimate for the *Drb* correction. As the robust G* is equivalent to classical gain schedules in 2 variables, this can easily be done at a small increase in complexity.
- Finding a modified version of the algorithm that allows to solve for K_i explicitly. This has been attempted, but taking K_i into account greatly increases the complexity of equations, which is why this approach was rejected. Furthermore, if the first two options are seen as problematic, eq. 5.33 can be used.

For nominal design, it is interesting to also introduce a different quantity, which is the dropback difference compared to the aircraft with the cg on the aerodynamic centre:

$$\Delta_{Drb_{25}} = Drb - Drb_{25} = \frac{mV_T}{\bar{q}SC_{L_\alpha}} \frac{C_{Z_\alpha} \bar{x}}{C_{m_{\delta_e}} a_0 K_i} \tag{5.36}$$

As a final note on the *Drb* correction, the surface of ω_{CL} should be maximized such as to be limited by the stability margins. The derivative gain should be used if possible, ignoring the OLOP criteria. To increase the *Drb* (at high speeds), there are three options :

- Increase the feed-forward zero, but this degrades the n_z OS.
- Increase the value of γ (if using eq. 5.32), but this will give high K_i values which may degrade stability margins.
- Increase ω_{CL}, this will decrease the magnitude of the term divided by ω, which is normally negative, in eq. 5.31.

If ω_{CL} is not aggressive enough, it may not be possible to meet both the overshoot and Drb requirements.

Command filter tuning

Now that the inner-loop of the control law has been defined (K_i, K_p, K_q, K_{ff} and K_d, if available) such as to meet the Drb and n_z overshoot requirements, the vast majority of the envelope should be adequate in terms of HQs. If this is not the case, re-tune the values of ψ, ζ, γ, and the ω_{CL} limit. Nonetheless, two HQs are susceptible of being inadequate in some envelope regions: the OLOP at low speed and the BW_θ at high speeds.

OLOP improvement

Worst cases of OLOP were found to always be at low speed (whatever the method used for design, including H_∞ and classical methods), when the elevator inputs are of high amplitude. The aggressive ω_{CL} limit will likely have degraded the OLOP. Aiming to solve this problem without re-tuning the inner-loop, adding a gain reduction through a command filter centered on the onset frequency will increase the onset frequency, reducing the OLOP gain. Although aiming for a gain reduction equal to the initial OLOP gain gives good results, this is a qualitative rule of thumb. For a given denominator time constant, the required numerator to offset the lag filter's magnitude response at ω_{onset} by Δ_{dB} is given by:

$$M(\omega_{onset}) = 10\left(\log\left[1+{\omega_{onset}}^2T^2\right] - \log\left[1+{\omega_{onset}}^2\tau^2\right]\right) \tag{5.37}$$

Solving for the value of T, we get:

$$10^{(\Delta_{dB}/10)}\left(1+{\omega_{onset}}^2\tau^2\right) = 1+{\omega_{onset}}^2T^2 \tag{5.38a}$$

$$T = \sqrt{\frac{10^{(\Delta_{dB}/10)}\left(1+\omega_{onset}^2\tau^2\right)-1}{\omega_{onset}^2}} \tag{5.38b}$$

Gain BW improvement

Throughout H_∞ and nominal G* improvement, the BW_θ was always considered as phase-limited, as it was generally low-speed cases that were unsatisfactory due to insufficient phase advance. Nonetheless, with the robust G* tuning and the use of the K_d gain, the BW_θ systematically degraded at or above V_{Mo}/M_{Mo}, a surprising behaviour as the phase bandwidth should have been great due to the high speed of the system. This behaviour had also been observed in many H_∞ syntheses, although the cause was not found during this part of the project. After further investigation, it appears that the gain response for certain conditions exhibits a flat portion below ω_{180}, sometimes lowering the gain bandwidth below 1.5 rad/s (see figure 5.8).

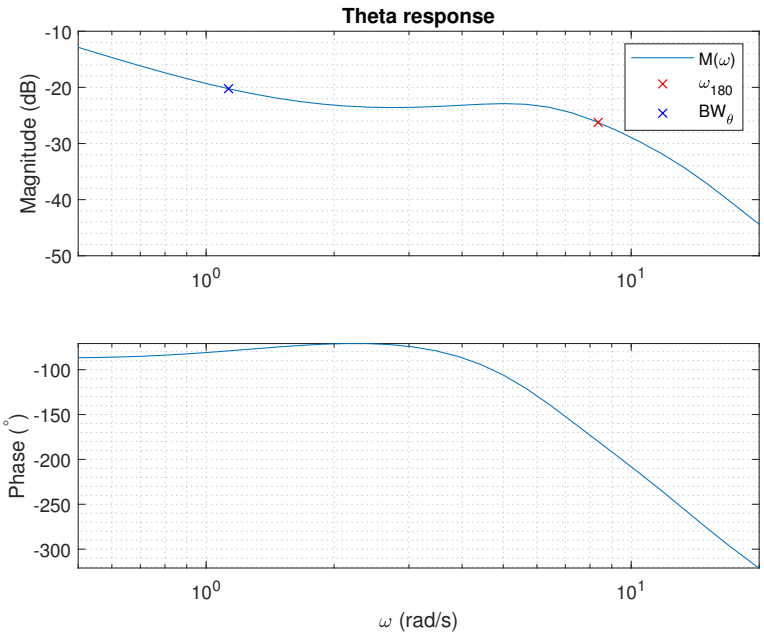


Figure 5.8 Typical high speed bandwidth degradation

The only solution found to this problem was the addition of a lag filter in this frequency range, allowing to increase the roll-off, at the cost of degradation in other high-frequency PIO criteria. Putting this gain roll-off between the desired bandwidth (e.g. 2 rad/s) and the initial ω_{180} gives good results at fairly low complexity. Nonetheless, even if enough gain reduction is added to give a 6dB reduction between these 2 frequencies, the lag filter will also decrease ω_{180}, making this method approximative. For a given denominator time constant, the required numerator to give a Δ_{dB} gain variation between ω_{BW} and ω_{180} is :

$$\begin{aligned} &M(\omega_{BW}) - M(\omega_{180}) = \\ &10\left(\log\left[1+{\omega_{BW}}^2T^2\right] - \log\left[1+{\omega_{BW}}^2\tau^2\right] - \log\left[1+{\omega_{180}}^2T^2\right] + \log\left[1+{\omega_{180}}^2\tau^2\right]\right) \end{aligned} \tag{5.39}$$

Isolating the value of T yields:

$$10^{(\Delta_{dB}/10)}\frac{1+{\omega_{BW}}^2\tau^2}{1+{\omega_{180}}^2\tau^2} = c_0 = \frac{1+{\omega_{BW}}^2T^2}{1+{\omega_{180}}^2T^2} \tag{5.40a}$$

$$c_0(1+{\omega_{180}}^2T^2) = 1+{\omega_{BW}}^2T^2 \tag{5.40b}$$

$$T = \sqrt{\frac{1-c_0}{c_0{\omega_{180}}^2 - {\omega_{BW}}^2}} \tag{5.40c}$$

Computation of the command filtering

Both HQ improvements require the addition of a lag filter. Although each method is not intended to be used at the same time, there is no guarantee that the amount of lag that will be computed for both methods is compatible. This is especially true in terms of frequency range where the magnitude reduction is required. Furthermore, even if both methods were optimal w.r.t. solving their respective problem and did not overlap, the lag added may not be optimal w.r.t. other HQs. For example, if the OLOP onset frequency occurs between ω_{180} and $2\omega_{180}$, the lag added will significantly degrade the phase delay. For these reasons, it was chosen to have a lag filter with a fixed denominator time constant, which allows the designer to choose himself (approximately) where the phase lag should occur. Furthermore, this solves the problem of conflicting lag, as it is possible to simply choose the most dephasing filter to meet both objectives. Good results were achieved with the same denominator time constant over the whole envelope. The design process of the lead-lag filter is described in algorithm 5.

Although the command filter design process is much more qualitative than the first steps, it remains simple. More effort could be put into optimizing this process, but the tuning parameters chosen this way have been able to achieve adequate linear and nonlinear performance. Furthermore, the rules of thumb defined give a great starting point if further manual opti-

Algorithm 5 G* lead-lag design process

1: Compute ω_{onset}, M_{OLOP} and ω_{180}, $M(\omega_{180})$, $M(\omega_{BW})$ in the θ response, where ω_{BW} is the desired bandwidth, throughout the whole envelope and select the worst cases for each design flight point.
2: Compute the value of T_{cmd} from equations 5.38b and 5.40c for a pre-selected value of τ_{cmd}. Looking at the absolute worst cases of OLOP and BW_θ will allow to find a good value. Some iterations must then be made to evaluate the best choice w.r.t. the phase delay degradations ($\tau_{lag} \in [0.3, 0.5]$s gave good results).
3: Define a schedule of T_{cmd}, by taking the filter with the minimum numerator time constant.
4: Use the second command filter to cancel the *Drb* reduction of the lag filter ($T_{lead} - \tau_{lead} = \tau_{lag} - T_{lag}$). The value of τ_{lead} should be selected such as to provide phase advance outside the frequency range relevant to both HQs from this section ($\tau_{lead} = 1$ s gave good results). Although the lead filter will degrade the overshoots, the maximum phase offset of the lead filter will be lower than the lag's due to the higher time constant (lower ratio).

mization is desired. Using H_∞ synthesis would easily allow to tune the command filter such as to meet the OLOP with W_2 used in the previous chapter. On the other hand, meeting BW_θ may require more effort, as this problem was also present in H_∞ synthesis.

Summarized robust design process

Given the flexibility of the G* methodology, it is always possible to define special objectives for problematic parts of the envelope, similar to how a gain schedule can be modified locally. Although this option is interesting for industrial application, this section intends to show that such fine-tuning is not necessary. Indeed, for the target aircraft, a reduced set of parameters was sufficient to perform adequate tuning without such ad-hoc modifications. The design process used is summarized in algorithms 6 and 7.

Algorithm 6 G* robust design process with variable γ

1: Choose ζ_{CL}.
2: Choose a range of γ values ([0.5,0.7] is a good first guess).
3: Compute the ω_{CL} surface.
4: Define a maximum n_z overshoot. Compute the corresponding value of ψ for the maximum γ used.
5: Define a minimum dropback value (Drb_{min}). Use eq. 5.32 to choose γ in real-time to meet the target dropback. This target dropback should be equal to $Drb_{min} + \Delta_{Drb}$, where Δ_{Drb} is given by eq. 5.35.
6: Compute handling qualities over the envelope and change parameters accordingly. The addition of a lead-lag filter through the process described is likely to be necessary.

Algorithm 7 G* robust design process with variable ψ

1: Choose ζ_{CL}.
2: Choose γ (0.6 is a good first guess).
3: Compute the ω_{CL} surface.
4: Define a maximum n_z overshoot. Compute the corresponding value of ψ for the maximum γ used. Define a minimum ψ value (0% overshoot is a good first guess).
5: Define a minimum dropback value (Drb_{min}). Use eq. 5.33 to choose ψ in real-time to meet the target dropback. This target dropback should be equal to $Drb_{min} + \Delta_{Drb}$, where Δ_{Drb} is given by eq. 5.35.
6: Compute handling qualities over the envelope and change parameters accordingly. The addition of a lead-lag filter through the process described is likely to be necessary.

Note that if the ANL law is used (later defined in section 5.4.2), it should be included in the linear design process, as it will have a significant effect on the dropback, n_z overshoot and stability margins.

5.3.5 Design process with knowledge of the mass and cg of the aircraft

The design process for the nominal case is very similar to the previous one. The main difference is that it is much easier to meet design requirements, usually allowing not to use lead-lag filters. Another difference is that the dropback is relaxed compared to the nominal 25% target. This is done to avoid strange gain shapes resulting from unnatural dropback requirements. This relaxation assumes that the K_i gain is constant between cg positions, which is not true. Nonetheless, as the intent is simply to get good gain behaviours, this is not a problem. This modified design process is presented for the γ relaxation in algorithm 8 (the corresponding ψ relaxation process can easily be inferred from the last section).

5.3.6 Applying the G* concept to other objectives

Improvements of the flight path bandwidth

Precise control of BW_γ has remained difficult throughout H_∞ synthesis and G* development, especially without mass and cg. Although providing phase advance is beneficial to bandwidths, at low speeds (worst bandwidths), phase advance is limited by the maximum dropback and OLOP. It was mentioned in section 4.2.1 that lowering the short period damping is beneficial to the bandwidth. Although this is generally true, imposing a damping lower than the open-loop damping ratio often results in negative K_q gains, which could create non-linear problems. Therefore, the main design trade-off is to meet these requirements without creating a K_q surface that seems problematic.

Algorithm 8 G* nominal design process with variable γ

1: Choose ζ_{CL}. Although a constant value will give good HQs, this may result in peculiar K_q gains due to the wide range of cg values. Instead, it is recommended to use the open-loop damping, subject to $\zeta_{min} < \zeta_{CL} < \zeta_{max}$, where the minimum and maximum values are to be defined.
2: Choose a range of γ values ([0.5,0.7] is a good first guess)
3: Compute the ω_{CL} surface.
4: Define a maximum n_z overshoot. Compute the corresponding value of ψ for the maximum γ used. If the range ζ_{CL} can take is significant, a polynomial can be defined to fit the profile of ψ in function of ζ_{CL}.
5: Define a minimum and maximum dropback. Use eq. 5.32 to choose γ in real-time to meet a target dropback. This target dropback, which should be bounded in minimum and maximum Drb, should be $Drb_{tgt} = Drb_{25}(T_{\theta_2}) + \Delta_{Drb_{25}}$.
6: Compute handling qualities over the envelope and change parameters accordingly. A lead-lag filter can be added if desired.

If this trade-off is to be avoided, it is possible to consider a washout in the pitch rate feedback path. The idea is to keep a good damping for the short period (of natural frequency around ≈ 2 rad/s), while limiting the effect of K_q in the BW_γ frequency range (≈ 0.55 rad/s). On the other hand, simply adding a washout degrades the SP damping and the n_z overshoots beyond acceptable values when BW_γ is improved. It is therefore necessary to include this washout in the pole placement algorithm. Doing so significantly increases the complexity of equations and results in a 4×4 matrix, bringing into question the compatibility with online gain computation. Nonetheless, it remains an interesting approach if gains are tabulated with the G* tuning method. As equations obtained are more complex, they are developed in annex E. Using these equations allows to slightly improve the BW_γ without loading information (control laws with mass and cg already having good bandwidths). Although the time spent experimenting with the washout equations was short, it was noticed that the washout cut-off frequency must be set lower than half of the SP real-part. Failure to do so results in badly behaving gains, this phenomenon was not explored further, although it is logical that the washout reduces the pitch rate feedback's effectiveness if it is too fast, resulting in difficulties to meet the required damping.

Application to the phugoid damping

Given the difficulties to characterize the PH mentioned in section 3.3.1, the application of the G* concept to the PH seems difficult. Due to the absence of an accurate reduced-order model, it would be necessary to work with a fourth-order open-loop transfer function. This

could be acceptable in terms of complexity, given that feedback dynamics are essentially negligible in the frequency range of the phugoid. Nonetheless, an accurate linear description of the phugoid seems difficult due to the number of stability derivatives, especially given the need for exotic stability derivatives (derivatives with regards to the true airspeed). This is why the application of the G* concept to phugoid damping was not developed beyond this thought experiment.

5.4 Nonlinear control law design

Despite the good handling qualities achieved with the previous developments w.r.t. the linear model, considering linear aircraft dynamics is unlikely to be effective for the whole aircraft envelope. Indeed, it was mentioned in section 3.2.7 that significant errors in M_α were reached close to stall even when tabulating with α. Although some errors were still present for other coefficients, C_{m_α} was by far the most problematic as it became dependent on many factors that could normally be neglected at high or low (negative) α. It is interesting to display C_m in function of the angle of attack and Mach for an aft cg (least stability), allowing to highlight locally unstable regions ($C_{m_\alpha} > 0$) prone to "pitch-up" problems. In these pitch-up regions, the effectiveness of envelope protection functions is likely to be reduced, because C_{m_α} nonlinearities may results in high overshoots. Running a nonlinear benchmark for overshoots w.r.t. the protection functions on the target aircraft allows to confirm this, as problematic points where large overshoots are achieved are within pitch-up regions. Therefore, it is necessary to address such nonlinearities. This section is divided into a short explanation of classical methods used to handle pitch-up problems, an in-depth description of the method selected and notes on validation of nonlinear control laws.

5.4.1 Classical nonlinear longitudinal design

Conceptually, if the aircraft modelization is very precise, a "simple" solution to the pitch-up problem is to tabulate the error (w.r.t. linear aerodynamics) in C_{m_α} throughout the envelope and explicitly cancel known deviations in the moment equation. In the case of the target aircraft, the nonlinear performance is limited by the absence of information w.r.t. mass and cg. The stability derivative C_{m_α} is largely affected by the cg position, making explicit cancellation unlikely. Still, overshoot worst cases were shown to be at aft cg for linear aircraft and this behaviour stays true for the nonlinear models according to simulation results. This means that explicit cancellation for an aft cg should be effective. On the other hand, the additional damping could slow down the response of forward-loaded aircraft. In the control law designed through classical means, the AoA augmentation terms are only active above a

minimum α threshold, roughly corresponding to the start of the nonlinearities. When active, a feedback in α provides the additional damping, while K_i is reduced by a factor to improve stability margins. Although very effective, this architecture results in a degradation of HQs within AoA augmentation.

5.4.2 Chosen nonlinear longitudinal design methods

If all loading information is known, some NDI approaches mentioned in the literature review seem promising. Values of the $\Delta\alpha$ and $\Delta\delta_e$ states will be useful in the following subsections, despite being impossible to measure physically on the aircraft. In simulations, trimming is performed by the horizontal stabilizer ($\Delta\delta_{e_e} = 0$), meaning that the initial value of α is the equilibrium value. In practice, the pilot may have to stabilize the aircraft transiently, e.g. at 1.2g in a turn, at a given angle of attack with the elevator, making the definition of these equilibrium values more complex. A washout filter ($\frac{\tau_w s}{\tau_w s+1}$) may be used to compute these states with time constants in the order of seconds, which gives good results in simulations. Rather than using a washout filter in α, the corresponding state can be isolated from the lift equation as suggested by [44], leading to:

$$\Delta\dot{\alpha} = a_{11}\Delta\alpha + q \tag{5.41}$$

This results in a first-order filter that can easily be implemented using solely the pitch rate measurement, which is interesting for failure cases. Nonetheless, this alternative has drawbacks at high AoA where a_{11} may be difficult to estimate (nonlinear C_{L_α}). Furthermore, during prolonged pulls to the α limit, the pitch rate may be negative even though $\dot{\alpha} \approx 0$ as γ decreases. With this alternative equation, the computed value of $\Delta\alpha$ for such maneuvers is negative, which may introduce undesirable effects. Nonetheless, both alternatives performed similarly in simulations.

Cancelation of known nonlinearities

Although many sources of nonlinearity exist at high angles of attack or Mach, pitch-up effects are a common problem in commercial aviation. This phenomenon is especially problematic, as it causes a pitch rate increase when α is high, pushing the aircraft closer to stalls. Although precise modelling of pitch-up effects is difficult for the same reasons mentioned in section 3.2.6, industrial aircraft models are expected to capture this phenomenon. If the modelization is accurate enough, [44] describes a method to explicitly cancel the pitch-up. Given the known pitch-up represented as $C_{m_\alpha}^{PU}(\alpha)$, this undesirable term can be cancelled with the elevator correction:

$$\Delta {\delta_e}^{APU} = -\frac{C_{m_\alpha}^{PU}(\alpha)\Delta\alpha}{C_{m_{\delta_e}}} \tag{5.42}$$

This is because the moment equation may be simplified to:

$$\frac{\dot{q}J_Y}{\bar{q}S\bar{c}} = \left(C_{m_\alpha}^{Lin} + C_{m_\alpha}^{PU}(\alpha)\right)\Delta\alpha + \frac{\bar{c}}{2V_T}C_{m_q}q + C_{m_{\delta_e}}\left(\Delta {\delta_e}^{Lin} + \Delta {\delta_e}^{APU}\right) \tag{5.43}$$

Although this correction seems valid for a finely tuned table of $C_{m_\alpha}^{PU}(\alpha)$, the table obtained in section 3.2.7 varies quite abruptly in pitch-up regions. This is why simulations have revealed sharp $\Delta {\delta_e}^{APU}$ evolutions with this correction. Furthermore, the tabulation of C_m is usually done as a function of α, making the conversion to C_{m_α} time-consuming and precarious as tables may need to be smoothed to ensure continuous corrections. Finally, this correction assumes that the pitch-up term is constant over $\Delta\alpha$, which may not be true. This is illustrated in figure 5.9, where an aircraft trimmed at $\alpha_e = 5$ is brought to $\alpha = 7$. The moment error computed with eq. 5.42 is compared to the actual error.

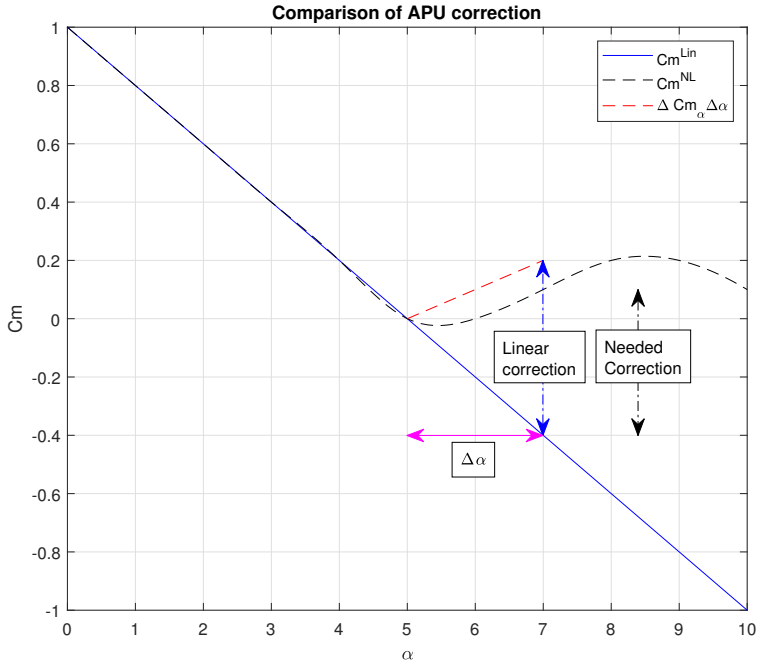


Figure 5.9 Pitch-up cancellation effectiveness

To alleviate these problems, a different correction is introduced, based on the assumption that the moment coefficient of the aircraft can be expressed through the equation:

$$C_m = C_m^\alpha(\alpha, ...) + C_m^o(\delta_{stab}, \delta_e, ...) \tag{5.44}$$

In eq. 5.44, $C_m^{\alpha}(\alpha, ..)$ fully describes the effect of α, while $C_m^{o}(\delta_{stab}, \delta_e, ...)$ contains effects fully independent from α. In other words, the moment change due to α is only a function of the angle of attack and a reduced set of parameters (e.g. Mach), while it is independent from most other parameters. In practice, complex couplings at high α can occur in transonic regime, resulting in a large number of variables being required in the α table to perfectly match the aircraft model. Nonetheless, it is generally possible to simplify such tables to a reduced set of parameters (e.g. $\alpha, M, \bar{q}$) that capture the "main" pitch-up effect, allowing interpolation in real-time with reasonable computational loads. For readability, only the explicit dependency in α will be kept, i.e. $C_m^{\alpha}(\alpha)$. Assuming that the aircraft is operating around an equilibrium α_e, such that $\alpha = \Delta\alpha + \alpha_e$, the C_m around the equilibrium point corresponding to the linear aerodynamics approximation is given by:

$$C_m^{Lin}(\alpha) = C_{m_\alpha}\Delta\alpha + C_m^{\alpha}(\alpha_e) + C_m^{o}(...) \tag{5.45}$$

while, by definition, the C_m from the model is:

$$C_m^{m}(\alpha) = C_m^{\alpha}(\alpha) + C_m^{o}(...) \tag{5.46}$$

Hence, the correction can be changed to:

$$\Delta\delta_e{}^{APU} = -\frac{C_m^{\alpha}(\alpha) - C_{m_\alpha}\Delta\alpha - C_m^{\alpha}(\alpha_e)}{C_{m_{\delta_e}}} \tag{5.47}$$

Although this correction requires two table interpolations instead of one, it is much easier to implement and less susceptible of creating unwanted behaviour due to numerical differentiation errors. Furthermore, $C_m^{o}(...)$ is canceled, meaning that additive offsets in the C_m^{α} table are not problematic. Nonlinear simulations with this control law have resulted in greatly improved performance compared to linear gains only. Although this improvement is to be expected, it is satisfying to have good results for the low time required to compute the C_m table. Furthermore, this law performs well without loading information when an aft cg is taken into account. Finally, note that using ca_{21} rather than C_{m_α} results in performance loss, as this will not fully compensate the pitch-up effect.

Cancelation of unknown nonlinearities

Multiple nonlinearities are essentially impossible to precisely characterize due to their chaotic nature, such as ice shapes, transient high α effects, modelling inaccuracies, etc. As direct cancellation is not realistic for such cases, a different approach is required. Furthermore, it

is likely that the C_m tables used for $\Delta\delta_e^{APU}$ computation will have to be limited to a few variables to allow rapid interpolation. Because the G* law should provide excellent behaviour so long as the aircraft's aerodynamics are linear, the intent is once again to cancel out undesirable aircraft nonlinearities. Direct adaptive methods do not seem like a good solution to this problem due to the different difficulties mentioned in section 2.5. Furthermore, within articles found that seem to have "realistic" computational needs for civilian aviation, the lack of recent experimental testing in nonlinear regions is also an obstacle w.r.t. validation. (Note that without this computational restraint, there are many examples of experimental projects that have succeeded [64], [65]) Instead, the INDI correction from [44] is considered. Given the linear aircraft model in the moment equation, the expected angular acceleration is given by:

$$\dot{q}_{tgt} = a_{21}\Delta\alpha + a_{22}\Delta q + b_2\Delta\delta_e^{Lin} \tag{5.48}$$

In practice, the real aircraft angular acceleration will be given by:

$$\dot{q} = a_{21}\Delta\alpha + a_{22}\Delta q + b_2\Delta\delta_e^{Lin} + \Delta\dot{q} \tag{5.49}$$

where $\Delta\dot{q}$ is an error term, which is time varying. This equation can be re-written as:

$$\dot{q} = \dot{q}_{tgt} + \Delta\dot{q} \tag{5.50a}$$

$$\Delta\dot{q} = \dot{q} - \dot{q}_{tgt} \tag{5.50b}$$

If $\hat{\Delta\dot{q}}$ is available, the correction that cancels this term is:

$$\Delta\delta_e^{ANL} = -\frac{\hat{\Delta\dot{q}}}{b_2} = -\frac{\hat{\dot{q}} - \dot{q}_{tgt}}{b_2} \tag{5.51}$$

Replacing this term into eq. 5.49 ensures the aircraft follows $\dot{q}_{tgt}$, assuming $\hat{\dot{q}} = \dot{q}$ and that $\Delta\delta_e^{ANL} = \Delta\delta_{e_c}^{ANL}$:

$$\dot{q} = a_{21}\Delta\alpha + a_{22}\Delta q + b_2\Delta\delta_e^{Lin} - \hat{\dot{q}} + \dot{q}_{tgt} + \dot{q} - \dot{q}_{tgt}$$
$$\hat{\dot{q}} = \dot{q} = \dot{q}_{tgt}$$

From these assumptions, it is clear that the control law's speed (update rate, loop delays, filtering, etc.) limits its effectiveness. Note that this correction was developed for dimensional stability derivatives because this allows considering $M_{\dot{\alpha}}$. Reference [44] develops these equations in terms of dimensionless stability derivatives and gives recommendations for the estimation of $\dot{q}$. In practice, $\hat{\dot{q}}$ must be estimated by a pseudo-derivator with a unit gain and

a cut-off frequency on the upper boundary of the piloting bandwidth, while remaining below flexible modes' lower frequencies. Using a pseudo-derivator that is "too fast" will degrade stability margins. Furthermore, q must be filtered to ensure there is no interaction with the aircraft's flexible modes. Finally, it is essential that all signals used be phased and corrected w.r.t. timing and delays to avoid introducing large transient errors.

The elevator position resulting from the linear aircraft command is also needed. This is problematic, as this measurement is unavailable. One way to estimate this quantity is to approximate the $\frac{\Delta\delta_e}{\Delta\delta_{e_c}}$ transfer (e.g. by a first-order dynamic and delay), allowing to reconstruct $\Delta\delta_e^{Lin}$ from previous linear commands. The delays related to sensor measurements can then be added to the actuator delay to ensure this signal is in phase with q and α measurements. The resulting architecture is shown in figure 5.10. Note that washout filters have been added following the previous remark on equilibrium values. In practice, the time constant of these washouts can be adjusted in reaction to nonlinear simulations (e.g. worst case n_z overshoots).

Nonlinear simulations with only this correction yield a significant improvement compared to the linear gains. Nonetheless, the best performance is achieved by combining both pitch-up and unknown dynamics corrections, giving excellent results so long as the pseudo-estimator is "fast enough".

5.4.3 Control law validation

The current industrial state of the art for validation of new control laws comprises two main aspects. First, all components are linearized over a grid that covers the flight and loading envelopes. These local linearizations are then used to study linear handling qualities and ensure that stability margins are respected, even with nonlinear control laws. Once linear design requirements are met, nonlinear simulations are run over the envelope to ensure conformity concerning nonlinear design requirements. To do so, the aircraft is submitted to a wide range of inputs that correspond to the worst cases for the evaluated metric. For example, worst-case load factor overshoots are reached for maximum amplitude push-pull manœuvers. The knowledge of such worst cases is critical, as this allows to greatly accelerate the validation process and avoid time-consuming simulations that are irrelevant.

Once the G* control law has been implemented on a nonlinear simulation model, running these simulations is straightforward and requires little changes from what is done industrially. Furthermore, the ANL law can (and should) be included in linear HQ and stability margin

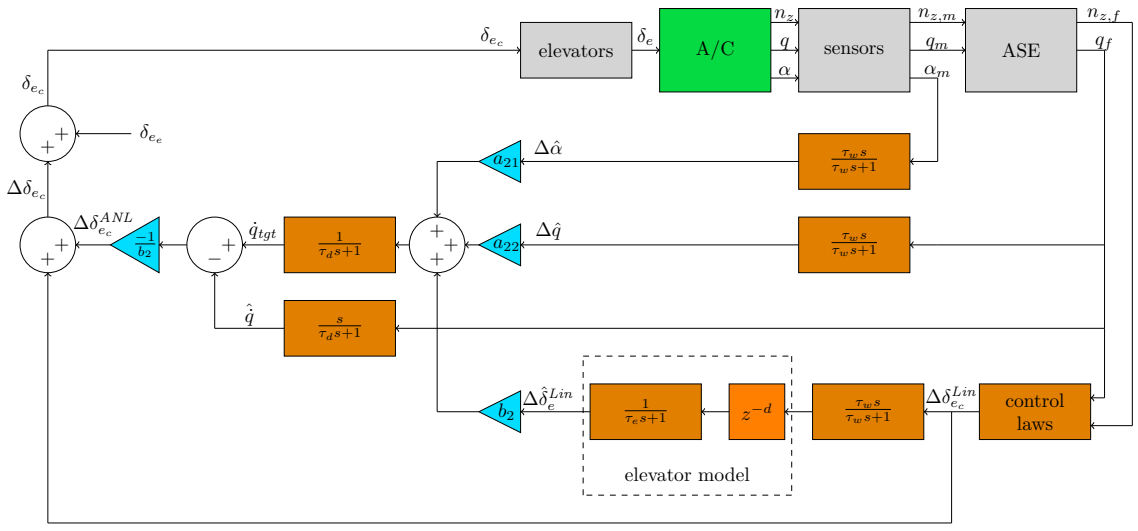


Figure 5.10 ANL law architecture

evaluations, since this law is linear for a given flight point (see figure 5.10). Although the APU law is fundamentally nonlinear, it is possible to evaluate its effect by computing the local linearization around α_e. This results in a correction following eq. 5.42. Although stability margins provide an intuitive measure of the system's tolerance to some uncertainties, this approach is unlikely to capture the absolute worst case in the envelope as these analyses are done on a grid. Work from [44] further develops robustness analysis on grids through μ (LTI uncertainties) and IQC (LTV uncertainties) analysis. Nonetheless, validation attempts on a continuum (a finite and continuous envelope subsection) resulted in excessive conservatism. Validation in [32], the only example found of an INDI law that was flight tested, is based on "classical" nonlinear grid analysis. Clearly, work on nonlinear validation remains a subject of interest.

Going back to the decision of feeding the pole-placement algorithm a fictive loading for robust designs, no theoretical result has been developed to show that this does not destabilize the aircraft for some loading conditions. Nonetheless, validation on a grid of mass/cg for all designs at aft conditions never resulted in weakly damped SP poles (below 0.35). There seems to be no discontinuity in the evolution of the damping, suggesting the validity of the grid validation. This justifies not using a dedicated robust pole placement method, although additional research on pole evolution through the loading envelope would be interesting.

Trade-offs between linear and nonlinear performance

As mentioned, the ANL law tends to degrade stability margins. When cg and mass estimates are available, this degradation is relatively mild, as the ANL correction should remain small for most conditions. This makes it relatively simple to get excellent linear performance while having a fast ANL law, leading to great nonlinear behaviour. In practice, the tuning of τ_d, the pseudo-derivator's time constant, should be done based on nonlinear simulations. Identification of a set of worst-case overshoots allows to evaluate the aircraft behaviour for different values and define the highest value that yields the desired performance. The higher the time constant is, the smaller the stability margins degradations will be, at the cost of degraded ANL performance. It is interesting to note that when stability margins are not met, it is possible to improve phase margins in exchange for a gain margin degradation (or the opposite) by a lead (or lag) filter on $\Delta\delta_{e_c}^{ANL}$. Although this is likely to also affect linear HQs, this can be an interesting design tool.

For the robust case, the use of the ANL law is much less trivial. Indeed, to limit worst-case overshoots and avoid computing other stability derivatives, an aft cg should be used for the ANL law. This generally improves linear HQs, as the aircraft is more responsive for aft cg, meaning that matching this behaviour is likely to improve some HQs for forward cg positions beyond what is possible with constant gains. This is especially apparent on the dropback. On the other hand, the maximization of ω_{CL} is even more critical for robust design to meet both the n_z overshoot and the dropback at high-speed, while the tradeoff is between the bandwidths and dropback at lower speeds. This means that the aircraft will be limited in terms of stability margin, as maximum linear performance is obtained for a maximum ω_{CL}. This results in the need to compromise on the stability margins that are "reserved" for nonlinear design, lowering ω_{CL} and the corresponding linear performance.

5.5 Conclusion

In this chapter, a methodology for longitudinal aircraft control has been defined. This methodology is based on real-time pole placement, from known aircraft (SP) dynamics. This process avoids the need to perform a re-design when some loop components change (e.g. the aerodynamic model or loop delays). Critical linear handling qualities are guaranteed by the computation in real-time of 4 parameters that define the closed-loop SP transfer, while secondary HQs are met through a reduced set of higher-level tuning parameters. This tuning method is of comparable effectiveness to the H_∞ methodology for linear HQs.

To ensure gain smoothness, linear design is performed with the linear aerodynamics approximation (low α). As modern aircraft are not linear throughout their envelope, additional control laws are required to use such an approximation. Two control laws have been defined for this purpose, the first one using known aircraft nonlinearities to cancel them, while the second estimates the pitch acceleration error w.r.t. the linear aircraft and cancels this offset. These corrections are effective on the target aircraft when used together, even without mass and cg information.

CHAPTER 6 Comparison of structured H_∞ synthesis and G* on the flap 0 configuration

In this short chapter, both methodologies developed in this thesis are applied to the flap 0 configuration of the target aircraft. First, structured H_∞ synthesis parameters are given along with the synthesis points used. Then, G* tuning parameters are given. The resulting gains of both methodologies are then shown and discussed, before doing the same for time responses, handling qualities and a typical nonlinear response.

6.1 Structured H_∞ synthesis parameters

Parameters used for the structured H_∞ synthesis for flap 0 are given in tables 6.1 to 6.4. Most of these parameters are related to the reference model HQ limits. It should be noted that the HQs that will be obtained are not necessarily within the limits of the reference model. Since the HQs are validated on points other than those used in the synthesis, some degradation is to be expected. Furthermore, the `RelGap` value quantifies the error between the reference model and the closed-loop aircraft. Therefore, higher values are expected to lead to more degraded HQs. These values were obtained by starting from HQ requirements and performing design iterations until reaching satisfactory results.

6.1.1 Choosing the synthesis points

Figures 6.1 and 6.2 show the envelope used for the flap 0 synthesis. The flight envelope has initially been defined with an educated guess, followed by synthesis iterations to identify potential degradations (e.g. points added at high speeds at 5,000 feet and some very high-speed points). In general, if the handling qualities on the synthesis points are satisfying, it is safe to assume that degradations on the full envelope are due to a poor selection of the synthesis points (from the flight and loading envelopes). Although having a small synthesis flight envelope is desirable, minimizing the number of loadings used is critical (as it is expected that more flight conditions are required than loading conditions) to have reasonable synthesis times and RAM use. Iterations to identify which loading conditions should be used are less straightforward than for flight conditions.

Table 6.1 Synthesis Parameters (flap 0)

Parameter	Value
$\texttt{RelGap}_{tgt}$	0.053
Ponderation	2.65
ζ_{min}	0.37

Table 6.2 Reference model HQ limits (flap 0)

Parameter	Value	
ζ	0.7	
Parameter	min	max
$T_{\theta 2}$	0.4	3
PD	0	0.145
BW_{γ}	0.6	∞
BW_{θ}	1.8	∞

Table 6.3 Altitude-dependent reference model HQ limits (flap 0)

Altitude (Kft)	1		5		10		15		20	
Parameter	min	max	min	max	min	max	min	max	min	max
Drb	-0.18	1.6	-0.18	1.6	-0.18	1.6	-0.18	1.6	-0.18	1.6
OS n_z (%)	0	8	0	8	0	8	0	8	0	8

Altitude (Kft)	25		30		35		41	
Parameter	min	max	min	max	min	max	min	max
Drb	-0.18	1.6	-0.30	1.6	-0.18	1.6	-0.18	1.6
OS n_z (%)	0	23	0	23	0	23	0	23

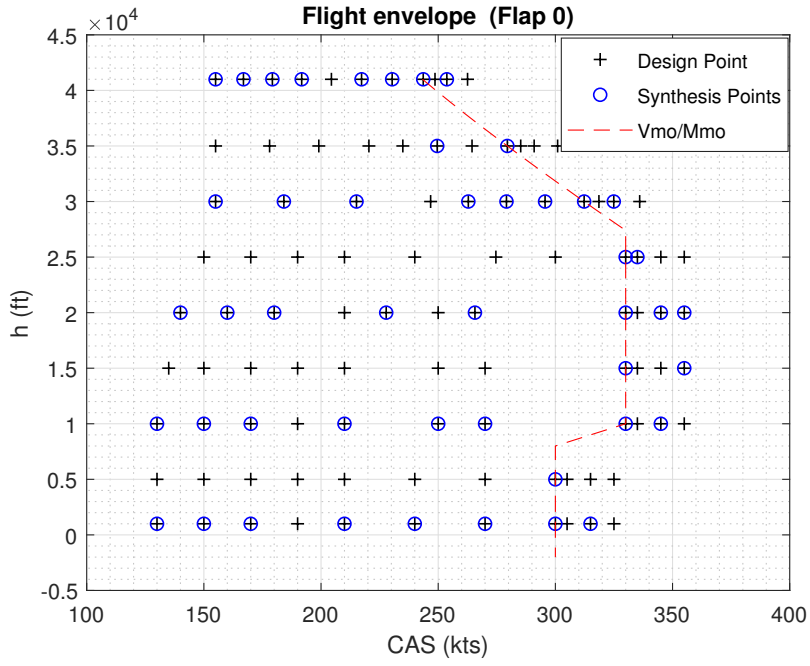


Figure 6.1 Flap 0 synthesis flight envelope

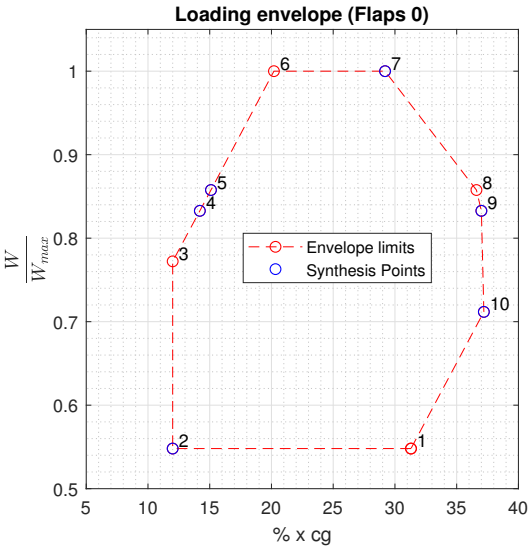
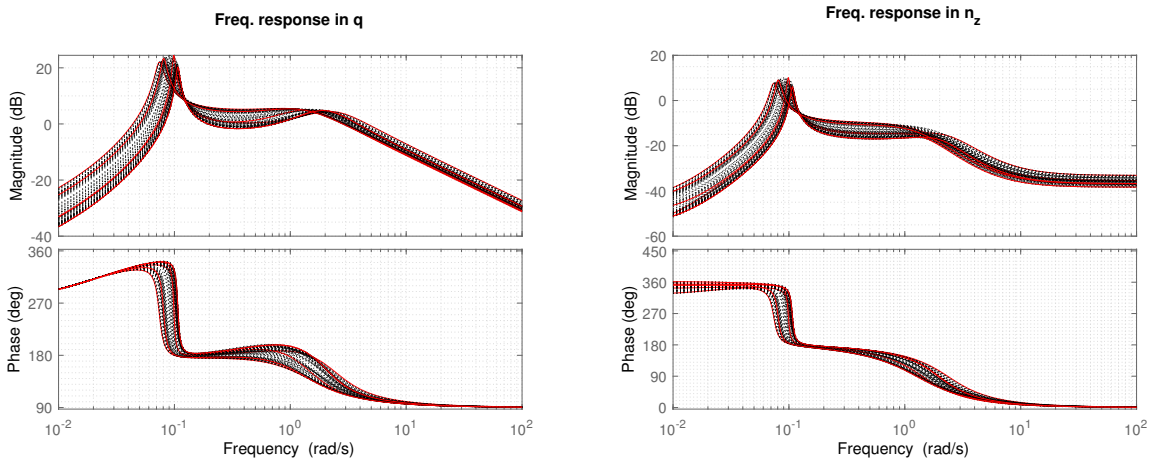


Figure 6.2 Flap 0 synthesis loading envelope

In [24], the open-loop aircraft response for a single flight condition but multiple loadings was overlayed for outputs of interest. Responses that bounded the aircraft behaviour (in the frequency range of interest) were selected for the synthesis. This process is very interesting for structured H_∞ synthesis, as gain optimization is done in the frequency domain. This process may then be repeated for a few flight conditions to ensure the selected points truly bound the possible responses. In the case of the target aircraft, six loadings were needed to bound the pitch rate and load factor responses, as shown in figure 6.3. Responses in red correspond to the loadings selected for the synthesis, while black ones are other mass and weight combinations.

Table 6.4 Acceptable space fit parameters

Fit	**Parameter**	**Value**
τ_{cmd}	Order in T_{θ_2}	3
T_{cmd}	Order in (τ_{cmd})	2
	Order in T_{θ_2}	15
$\frac{1}{\omega_n}$	Order in (τ_{cmd}, T_{cmd})	2
	Order in T_{θ_2}	10



(a) q (b) n_z

Figure 6.3 Typical open-loop bode response of the aircraft

6.2 G* tuning parameters

Four variations of a G* law were tested in the context of this project:

- Nominal G* law
- Robust G* law, optimized for linear HQ
- Robust G* law designed to offer a trade-off between linear HQ and ANL performance
- Robust G* law, optimized for nonlinear performance

The nominal G* law highlights the "best" achievable performance, although there is no reference to which it could be compared. The robust G* law that offers a trade-off between linear and nonlinear performance will be detailed, as it is thought to be the most realistic use-case of the G* methodology without mass and cg. The other 2 robust G* iterations will not be studied in this thesis, as these laws remain close to other robust G* iterations. Indeed, the main HQ difference was the improved *Drb* in exchange for bandwidth degradation. To avoid saturating HQ plots, they are not included. The parameters that define the robust control laws are given below. It is interesting to compare the small number of parameters that fully define the gains compared to those required to obtain H_∞ surfaces (especially when acceptable space fit parameters are considered), making the G* methodology appealing. Although aerodynamic tables are also required, these tables were not tuned by hand at all.

G* - Robust design, trade-off between linear and nonlinear performance

Tuning parameters for the G* methodology are given below. These values were obtained through the tuning process detailed in algorithm 7.

- $\zeta_{CL} = 0.66$
- $\omega_{CL} = -0.8213 + 0.8517\omega_{OL} + 5.2624\zeta_{OL} + 1.2196\omega_{OL}\zeta_{OL} - 0.1331\omega_{OL}^2 - 4.8067\zeta_{OL}^2$
- $\gamma_{CL} = 0.6$
- $Drb_{min} = -0.15$
- $\psi \in [2.25, 2.55]$
- $\tau_{cmd} = 1.2, \quad \tau_{cmd,HF} = 0.4$, $T_{cmd,HF}$ is defined in table 6.5, while T_{cmd} is used to cancel the dropback added by the HF lead-lag filter.

6.3 Comparison of the robust gain surfaces

Gains surfaces from G* have been truncated to only keep data points below M_D in figures 6.4 and 6.5. Aerodynamic coefficients above this speed are prone to rapid evolution and tend to misbehave. This is to be expected, as these points are outside of the operational envelope. Furthermore, the tabulation done is not valid at such speeds. This is not a problem for H_∞, as the polynomials ensure discontinuous evolution does not occur. Nonetheless, within the envelope, the G* gains behave in continuous and "intuitive" ways so long as the design parameters are realistic for the system. The only exception to this is that sharp changes in aerodynamic coefficient can affect these surfaces, which causes the dents above M0.75 and M_{Mo} on most surfaces. These dents are not representative of the target aircraft's dynamics,

Table 6.5 G* - Robust lag filter

$\bar{q}$(psf)\h(ft)	1,000	5,000	10,000	15,000	20,000	25,000	30,000	35,000	41,000
75	0.25	0.24	0.25	0.25	0.27	0.30	0.33	0.35	0.37
100	0.25	0.24	0.25	0.25	0.27	0.30	0.33	0.35	0.37
125	0.25	0.25	0.25	0.25	0.27	0.30	0.30	0.35	0.32
150	0.25	0.25	0.25	0.25	0.27	0.27	0.27	0.30	0.27
175	0.25	0.25	0.25	0.25	0.27	0.27	0.24	0.25	0.22
200	0.25	0.25	0.25	0.25	0.27	0.27	0.21	0.20	0.17
225	0.25	0.25	0.25	0.25	0.25	0.24	0.18	0.15	0.15
250	0.25	0.25	0.25	0.25	0.23	0.21	0.15	0.15	0.15
275	0.25	0.25	0.25	0.24	0.21	0.18	0.15	0.15	0.15
300	0.25	0.25	0.25	0.23	0.19	0.15	0.13	0.15	0.15
325	0.25	0.25	0.25	0.22	0.15	0.13	0.13	0.15	0.15
350	0.25	0.25	0.25	0.20	0.13	0.13	0.13	0.15	0.15
375	0.25	0.25	0.24	0.18	0.13	0.13	0.13	0.15	0.15
400	0.25	0.25	0.22	0.16	0.13	0.13	0.13	0.15	0.15
425	0.25	0.25	0.20	0.16	0.13	0.13	0.13	0.15	0.15

as they are the result of poor tabulation precision. For HQ evaluation, G* gains were computed for a Mach number limited below 0.75. If these dents are to be avoided, tables can be re-computed and fine-tuned to ensure they are representative.

In contrast, it is less trivial to force the H_∞ synthesis to have an "intuitive" behaviour for gains other than K_p and K_i. Although this has been mitigated for K_{ff} by pre-defining the surface, K_q takes some slightly negative values, K_d converged to maximum values at high Mach, while the command filter evolution is difficult to analyze. Furthermore, in synthesis for a high flap setting, gains had even stronger tendencies to converge to "odd" behaviours, which could not be avoided in the short time spent on this flap setting. The G* methodology does not use a pitch rate washout, while the time constant is set to 1.4587s for H_∞.

The K_i and K_p gains have similar behaviours with both methods, although they are smaller for G*. This is because ω_{CL} cannot be fully maximized if the ANL law is used to meet stability margins. Furthermore, K_p takes small values at high altitudes and low speeds, where the aircraft operates in reduced C_{m_α} regions, because of the linear aerodynamics assumption. As the aircraft is within this region at equilibrium, H_∞ converged to a higher proportional gain. To avoid this problem, aerodynamic tables could be biased, although this low gain did not cause nonlinear problems. K_q takes high values at low speeds and high altitudes for G*. This is because in this part of the envelope, the aircraft's natural damping is low while the damping requirement is kept constant throughout the envelope. K_{ff} surfaces have similar shapes at high speeds, while low-speed maximums are different. In general, G* gains tend to "peak" at low speeds, which is intuitive. In practice, it is necessary to limit gain computation to a minimum value of dynamic pressure to avoid extremely large gains if the aircraft is operated near stall.

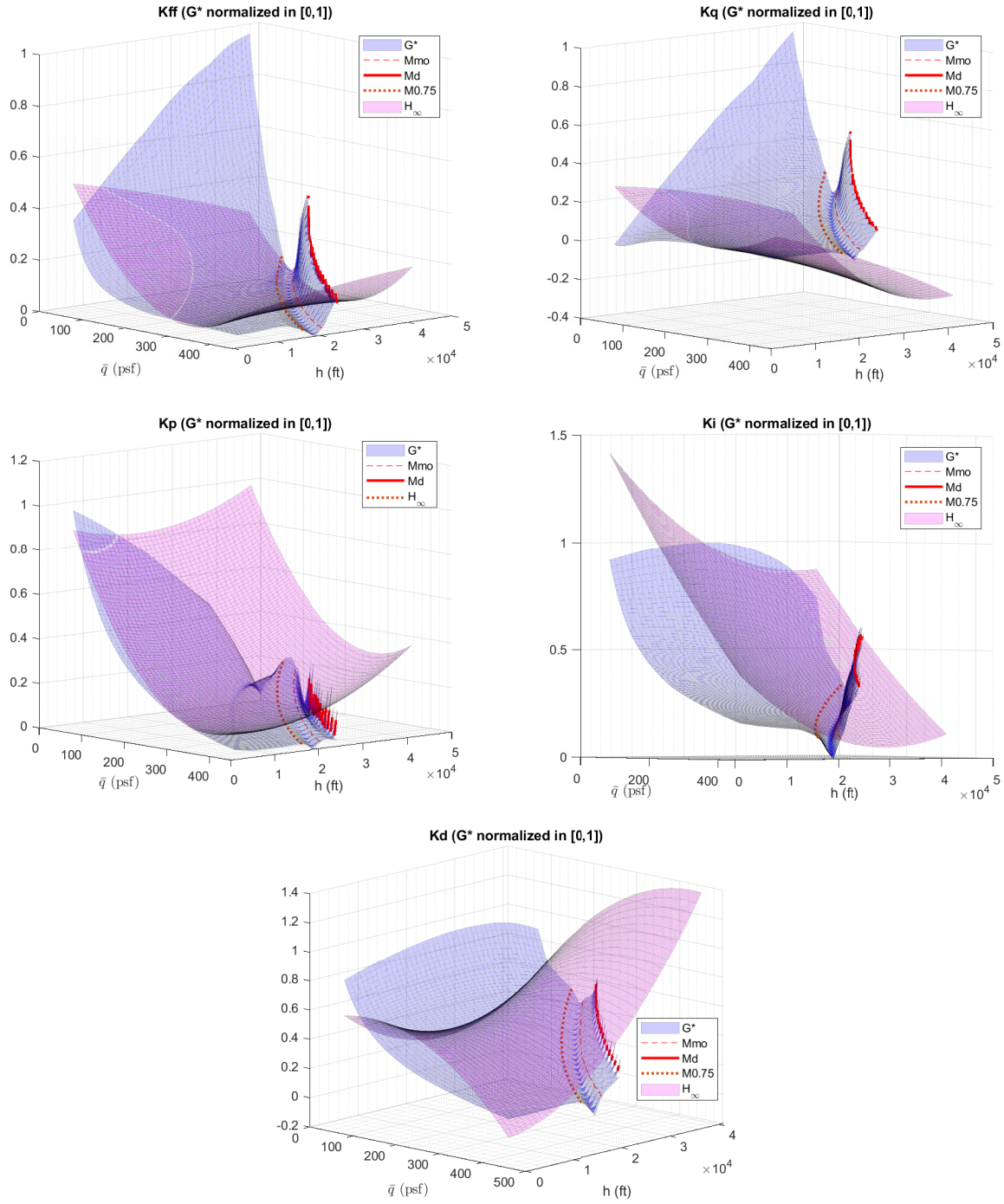


Figure 6.4 Comparison of the Robust G* and H_∞ gains

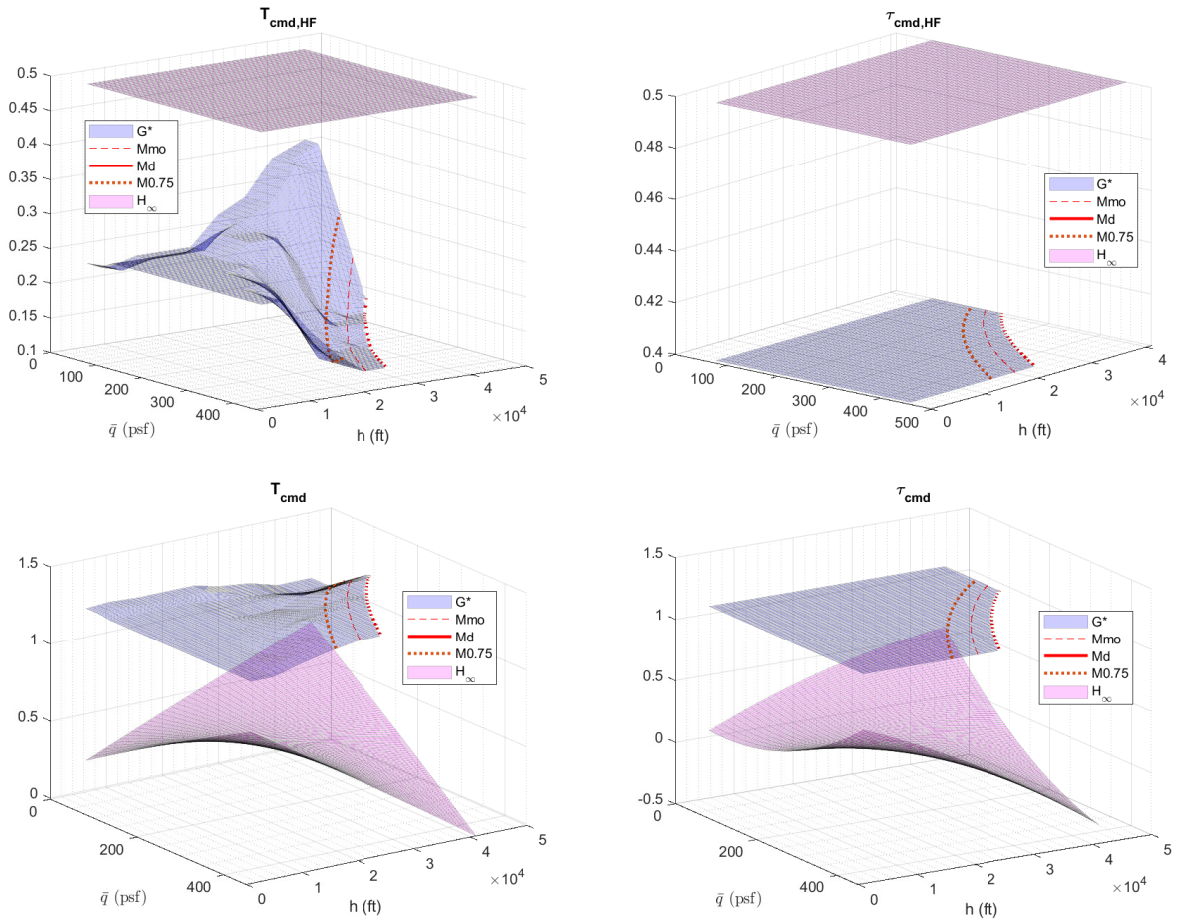


Figure 6.5 Comparison of the Robust G* and H_∞ gains (2)

6.4 Linear time responses

Figure 6.6 shows the time responses of the short period model for the load factor, pitch rate and pitch responses. To allow easier qualitative comparison of aircraft behavior, these responses were normalized to a unit steady-state gain. As no settling time design objective was defined, the 4.4 s (within 10% error) criterion from reference [59] was used to provide a guideline. The yellow line in the q response corresponds to the PRO limit for a $Drb \approx 1.5$. Precise validation of this criterion will be done with HQ figures in section 6.5 and this line is provided here for reference. Overall, aircraft responses are satisfactory and there is no "aberant" result, e.g. highly underdamped oscillations. The H_∞ gains seem to accelerate the n_z response more than the G* gains, while G* gains can be seen to have higher Drb values in the θ responses.

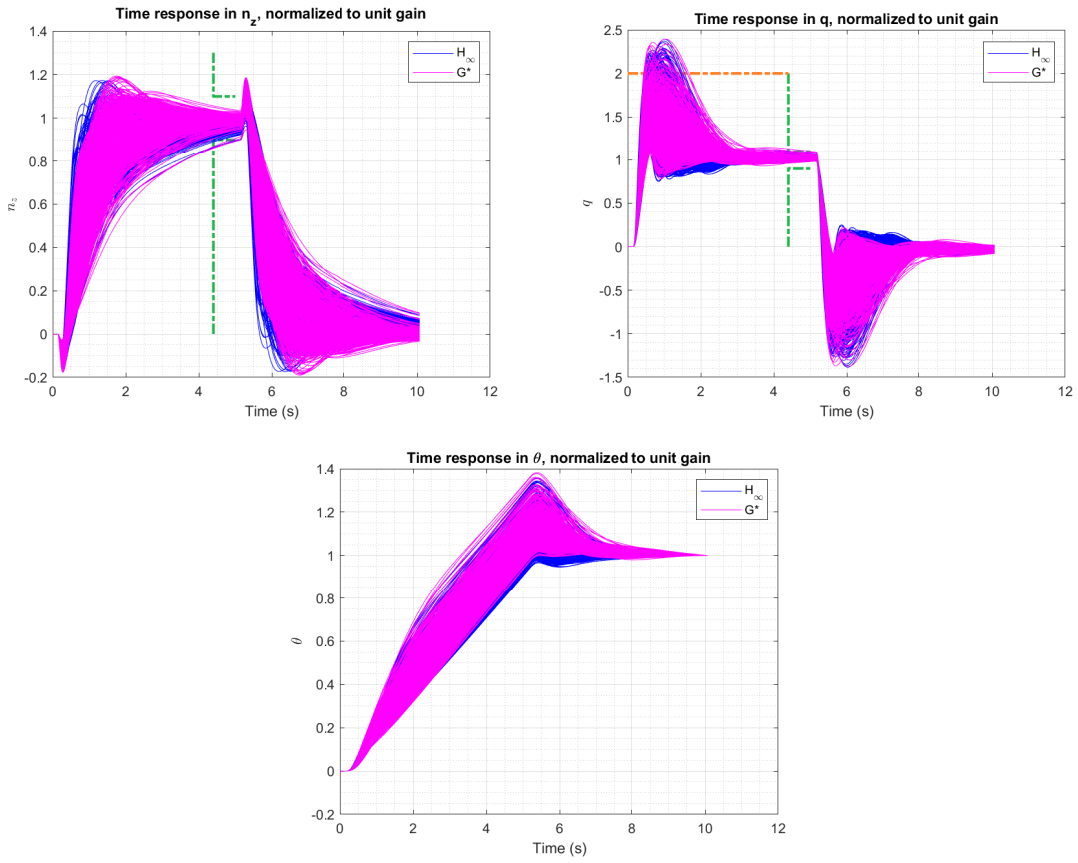


Figure 6.6 Comparison of the Robust G* and H_∞ time responses

6.5 Comparison of linear HQs

Classically, handling qualities are validated in scatter plots, where the x and y axis correspond to different HQs, along with design limits. As some HQ limits are functions of other handling qualities, using this type of figure is intuitive. Nonetheless, these plots give no information w.r.t. where poor HQs are obtained, limiting their appeal for design. For example, poor bandwidths are to be expected (to some extent) at low speeds, hence the need to contextualize results. One solution to this problem is to plot each HQ individually in function of a flight condition number, a unique number assigned to a ($\bar{q}$, h) couple. Points for HQ which depend on multiple values can be colour coded for validation of HQ levels. In this section, results are shown with the first type of plot, while the commentaries will provide some context about where the worst cases occur. This validation is done against the numerically linearized aircraft model for the flight and loading envelopes defined. Green lines correspond to level 1* HQ limits (if defined), while lines in yellow and red are level 1 and 2, respectively.

As the G* law has been tuned with the nonlinear law, it is necessary to consider all components in HQ evaluation. Indeed, the ANL law degrades stability margins, resulting in the need for additional trade-offs and the need to reduce the ω_{CL} limit. At the same time, the ANL law improves other HQs, resulting in vastly different values if it is ignored. When linearized, the APU law is a feedback on the $\Delta\alpha$ state, giving it a similar role to the q feedback with regards to BW_γ degradations. This is apparent at low speed and high altitudes, where this HQ is on the level 2 limit for G*. Without the APU law, results are aligned with H_∞. The robust G* law tuning is more aggressive than H_∞ and some non-conformal high pilot-gain OLOP points were allowed to counteract the ω_{CL} limitation due to the ANL law.

Figure 6.7 contains results w.r.t. design criteria defined in this document. Both control laws have similar breaches of the high $Drb - PRO$ limit, which is reached at low speeds. Although it is possible to fully respect the upper limit, this results in the degradation of bandwidths. At higher speeds, the ANL law increases the Drb, which is why G* minimum values are significantly higher. Indeed, few points are negative throughout the operational envelope.

Level 1 phase delay is respected for both control laws, while the majority of BW_θ values are within level 1*. At low speeds, for H_∞ this metric is better due to the higher ω_{CL}, while at high speeds H_∞ results are largely degraded by the magnitude bandwidth limit. This problem was fixed for G* by the tuning of the command filter. This can be seen in the BW_γ plot, as low H_∞ BW_θ are above the level 2 upper limit. As mentioned, in nonlinear regions, worst BW_γ values are reached with G* because of the APU law.

Gibson criteria (APR and f_{180}) are of limited use. Indeed, the APR and f_{180} limits are respected with large margins (note that the stricter 1* limits [$APR< 50$ and $f_{180} > 1$] would be breached systematically). Bounds on these HQs should be adapted to commercial aviation for further use. Validation of the PRS is difficult without information on the flight point, as conditions above V_{Mo}/M_{Mo} undergo a rapid PRS degradation. Sensitivities were adjusted by a constant offset to allow to keep Gibson limits for the target aircraft, the intent being to study the range of variation of this metric rather than its absolute value. Overall, both tuning methods vary by about 7dB when ignoring these high-speed points, meaning that they are within the requirement. Nonetheless, as already mentioned, this requirement should be adapted to commercial aviation if the intent is have an absolute upper and lower bound. It is interesting to mention that G* gains generally achieve lower PRS values. From experience, increasing γ increases the PRS. This is in accordance to what is seen here, as H_∞ design was done for $\gamma = 1$ instead of $\gamma = 0.6$.

Finally, stability margins are similar with both methods, although H_∞ synthesis is usually limited by the phase margin, while G* is on the disk limits. A few points have failed the margin requirements for H_∞ by a small amount. Damping ratio limits are respected, although G* damping is higher.

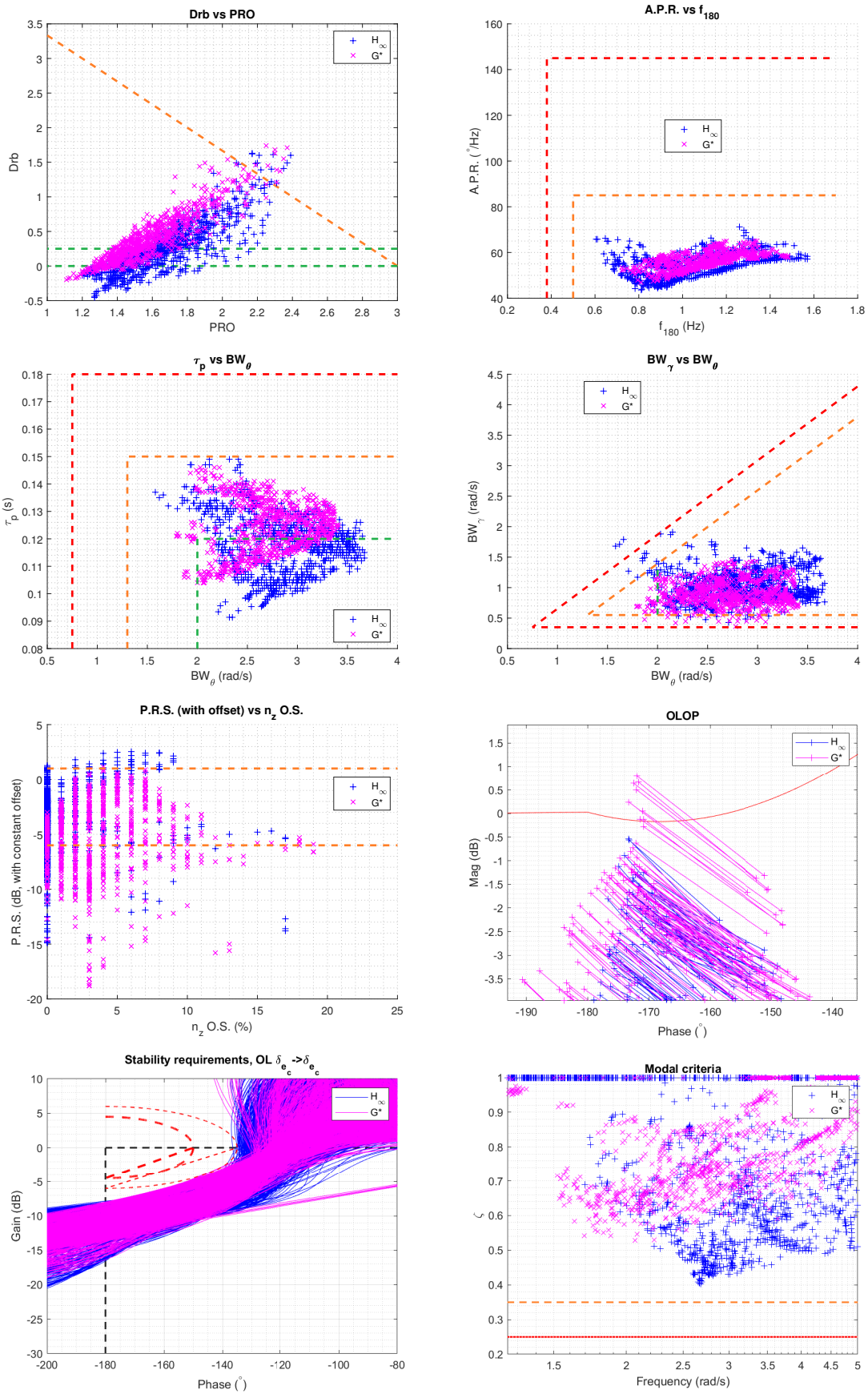


Figure 6.7 Comparison of the Robust G* and H_∞ HQs

6.5.1 G* nonlinear results

Simulations were performed throughout the envelope for critical sets of maneuvers to see how the G* law behaves. When information on mass and cg is available, the nonlinear performance is excellent. This is because the ANL law can be tuned more aggressively and it is easier to meet good linear HQs in the nominal case. Indeed, meeting low linear n_z overshoots, which are the same as α overshoots on the linear model with the approximations used, is much easier. Nonlinear simulation results are essentially exempt from overshoots with regards to limiting functions, including in nonlinear portions of the envelope. In the robust case, these results are slightly degraded. Overshoots for high α manœuvers remain quite similar to the nominal case. Figure 6.8 shows an example of the α response with and without ANL and APU laws, with the same linear G* tuning. The nonlinear laws fully prevent the overshoot, although they initially stabilize the aircraft below the α limit (around $\approx$ 6 s) before the integrator converges. This behavior is to be expected as this simulation is performed with a different aircraft mass than the one used in the APU law, resulting in a highly damped response. At higher speeds in transonic regions, a few negative overshoots of the n_z command can be found. Still, they remain small and are deemed satisfactory considering the law has no information on mass and cg. From linear and nonlinear results, it does not seem necessary to bias the C_{m_α} coefficient to increase the K_p gain, although this might accelerate the convergence to the α limit.

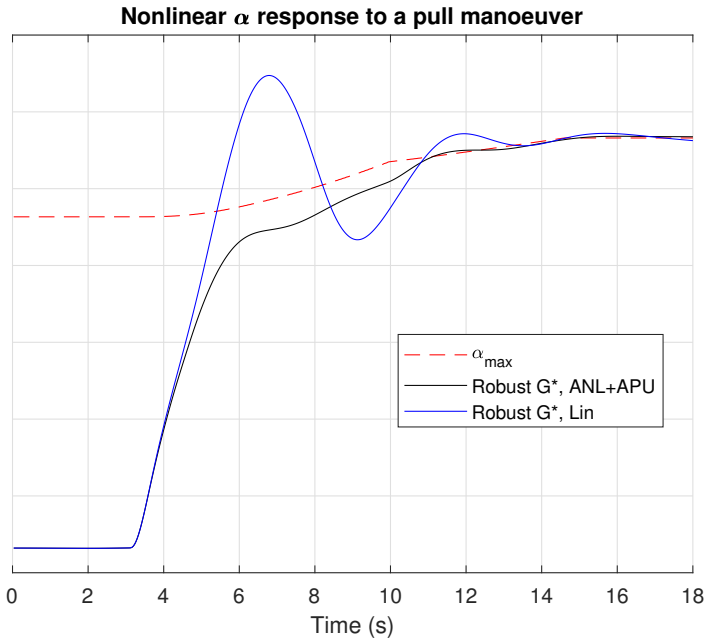


Figure 6.8 Example of nonlinear response

6.6 Conclusion

Overall, both developed methodologies were shown to yield adequate performance on the target aircraft. The principal difference in the results obtained was that the trade-offs that had to be made with both approaches were different. Because only linear behavior was considered for structured H_∞ synthesis, there was no need to compromise on the closed-loop frequencies that could be achieved. Nonetheless, this is not a limitation of the G* methodology, but rather, a choice that was made to improve the nonlinear aircraft behavior.

CHAPTER 7 CONCLUSION

7.1 Summary of Works

In this thesis, two methodologies for longitudinal gain design have been developed. The first approach, based on structured H_∞ synthesis, allows designers to compute gain surfaces defined as polynomials that meet the desired stability margins and handling qualities. The closed-loop system is matched to a reference model, within the SP frequency range, enforcing good transient characteristics. As aircraft dynamics vary significantly for different cg and mass values, the required robustness is met through a variable reference model which is limited to a space of parameters that yields good handling qualities. This space can be controlled by the designer, allowing to define HQ limits for the synthesis.

The second approach, the linear G* methodology, consists of an online pole-placement algorithm and simplified aerodynamic modelization of the aircraft. The aircraft response is controlled through the closed-loop short period and integrator poles, along with the feedforward zero, resulting in four tuning parameters for a given design point. An offline optimization method is used to compute the maximum closed-loop frequency such as to meet the required stability margins, while the damping is chosen by the designer. These values are then evaluated online through simple polynomials which approximate the offline optimization results. Analysis of the closed-loop transfer function in q and n_z has allowed to define the dropback and n_z overshoot in function of the closed-loop poles and zero. This enables the designer to explicitly meet these two handling qualities. These values are computed online from HQ requirements that are implicitly defined offline through higher-level tuning parameters. This procedure has the advantage of eliminating the need for scheduling (although G* can be used to design gain schedules) while greatly reducing the number of parameters the designer must tune.

The linear pole placement in the G* methodology is done for simplified linear aerodynamics (low α approximation), resulting in excellent behaviour when the aircraft behaves linearly. This ensures that gains evolve smoothly. The nonlinear behaviour is then matched to this expected linear response through nonlinear dynamic inversion. As industrial models are expected to capture undesirable nonlinear phenomenons such as pitch-up effects, these behaviours can be cancelled explicitly from the moment equation to some extent. The remaining nonlinearities are then cancelled through the comparison of the aircraft's angular acceleration to a target resulting from linear aircraft behaviour.

7.2 Limitations

As the H_∞ approach is used for gain scheduling, it retains most limitations associated with this category of approaches. The main exception is that the effort required to obtain gain schedules in function of HQ targets is greatly reduced. Nonetheless, the gain design problem is implicitly linked to the problem of finding the best compromise in HQs, as the design requirements used over-constrain the problem. The exploration of the possible solutions is therefore time consuming and is limited by the synthesis time.

Although the G* approach addresses multiple problems related to linear design, the nonlinear design methodology relies on strong assumptions, especially when no mass or cg information is known. The designer is faced with the need to compromise between linear and nonlinear performance, as the aircraft's angular acceleration must be estimated through filtering, degrading stability margins. Furthermore, nonlinear performance is highly dependent on the correct synchronization of the signals used.

7.3 Future Research

The nonlinear control laws conceived have been studied through common industrial validation methods, such as linear stability margins or handling qualities and nonlinear simulations. Validation of these laws through theory-based nonlinear methods could not be performed, due to the limited time remaining for the research of such methods. Therefore, nonlinear validation remains a key challenge.

As the nonlinear INDI law performance is strongly limited by the "speed" at which the pitch acceleration can be estimated, classical NDI architectures common in military aviation (e.g. [48], [31]) seem interesting to avoid such problems, as they have been tested at moderate sample rates. Furthermore, the reference model design can largely be based on the G* tuning parameters and closed-loop objectives. Nonetheless, it is not clear how these architectures could be used alongside the "traditional" C* tracking architecture. Experience gained with "traditional" architectures is precious to designers for validation purposes, complicating the transition to a completely different architecture. Furthermore, NDI architectures are even more dependent on flight mechanics identification. Additional work is required to know whether industrial models are of sufficient precision to reach adequate performance.

REFERENCES

[1] "Flying qualities of piloted aircraft," U.S. Department of Defense, Tech. Rep. MIL-HDBK-1797, 1997.

[2] P. Apkarian, "Tuning controllers against multiple design requirements," in *American Control Conference*, 2013, pp. 3888–3893.

[3] I. Nicolin and N. Adrian, "The fly-by-wire system," *INCAS BULLETIN*, vol. 11, pp. 217–222, 12 2019.

[4] D. Leith and W. Leithead, "Survey of gain-scheduling analysis and design," *International Journal of Control*, vol. 73, pp. 1001–1025, 01 2000.

[5] J. Harris and G. Black, "F-22 control law development and flying qualities," in *21st Atmospheric Flight Mechanics Conference*, 07 1996.

[6] A. Teixeira *et al.*, "2.63 - online control strategies," in *Comprehensive Biotechnology*, 2nd ed., M. Moo-Young, Ed. Burlington: Academic Press, 2011, pp. 875–882.

[7] J. Caigny *et al.*, "Gain-scheduled dynamic output feedback control for discrete-time LPV systems," *International Journal of Robust and Nonlinear Control*, vol. 22, pp. 535 – 558, 03 2012.

[8] E. Field, "The application of a C* flight control law to large civil transport aircraft," https://dspace.lib.cranfield.ac.uk/bitstream/handle/1826/186/coareport9303.pdf?sequence=2&isAllowed=y, Cranfield Institute of technology, Tech. Rep. College of Aeronautics report No.9303, 1993.

[9] D. Niedermeier and A. Lambregts, "Fly-by-wire augmented manual control - basic design considerations," *28th Congress of the International Council of the Aeronautical Sciences 2012, ICAS 2012*, vol. 4, pp. 3073–3086, 01 2012.

[10] Boeing, "Aircraft pitch-axis stability and command augmentation," U.S. Patent US5 722 620A, 1998.

[11] "Flight control design – best practices," Task Group SCI-026 on Flight Control Law Design, NATO, Tech. Rep. RTO 29, 2000.

[12] H. Klyde *et al.*, "Unified pilot-induced oscillations theory volume 1 : PIO analysis with linear and nonlinear effective vehicle characteristics, inbcluding rate limiting," Weight Laboratory, Tech. Rep. WL-TR-96-3028, 1995.

[13] G. Cooper and R. Harper, "The use of pilot ratings in evaluation of aircraft handling qualities," NASA Ames research center, Tech. Rep. Agard 567, 05 1969.

[14] D. Mitchell *et al.*, "Proposed incorporation of mission-oriented flying qualities into MIL-STD-1797A," Wright laboratory, Tech. Rep. WL-TR-94-3162, 10 1994.

[15] "Military specification: Flight control systems - design, installation and test of piloted aircraft general specification," USAF, Tech. Rep. MIL-F-9490D, 1975.

[16] J. Gibson, *Development of a Methodology for Excellence in Handling Qualities Design for Fly by Wire Aircraft*, ser. 3 - Control and simulation. Delft University Press, 1999.

[17] J. Gibson, *The Definition, Understanding and Design of Aircraft Handling Qualities*, ser. 3 - Control and simulation. Delft University Press, 1997.

[18] D. A. Saussié, "Contrôle du vol longitudinal d'un avion civil avec satisfaction de qualités de manoeuvrabilité," Ph.D. dissertation, École Polytechnique de Montréal, 8 2010.

[19] H. Duda, "Prediction of pilot-in-the-loop oscillations due to rate saturation," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 581–587, 1997. [Online]. Available: https://doi.org/10.2514/2.4080

[20] H. Duda *et al.*, "New flight simulator experiments on pilot-involved oscillations due to rate saturation," *Journal of Guidance, Control, and Dynamics*, vol. 23, pp. 312–318, 03 2000.

[21] S.-H. Park, "Handling deflection limit in open-loop-onset-point pio analysis," *Journal of The Korean Society for Aeronautical & Space Sciences*, vol. 38, pp. 135–140, 02 2010.

[22] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109800000583

[23] M. Tischler *et al.*, "Conduit: A new multidisciplinary integration environment for flight control development," NASA, Tech. Rep. NASA Technical Memorandum 112203, 07 1997.

[24] T. Berger *et al.*, "Business jet fly-by-wire control laws handling qualities flight test assessment," *Journal of Guidance, Control, and Dynamics*, vol. 44, pp. 1–19, 05 2021.

[25] C.-D. Yang, H.-S. Ju, and S.-W. Liu, "Experimental design of H-infinity weighting functions for flight control systems," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 3, pp. 544–552, 1994. [Online]. Available: https://doi.org/10.2514/3.21232

[26] R. Lind, "Gain-scheduled approximations to H-infinity controllers for the f/a-18 active aeroelastic wing," in *Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, Aug. 1999. [Online]. Available: https://doi.org/10.2514/6.1999-4205

[27] A. Marcos, "Revisiting the aircraft C-star control law: A comparison between classical and structured H-infinity designs," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 2114–2119.

[28] H. Lhachemi, D. Saussie, and G. Zhu, "A structured H-infinity-based optimization approach for integrated plant and self-scheduled flight control system design," *Aerospace Science and Technology*, vol. 45, 04 2015.

[29] K. Åström and B. Wittenmark, *Adaptive Control*, ser. Dover Books on Electrical Engineering. Dover Publications, 2008.

[30] B. Stevens, F. lewis, and E. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. John Wiley & Sons, 10 2015.

[31] J. J. Harris, "F-35 flight control law design, development and verification," in *2018 Aviation Technology, Integration, and Operations Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, Jun. 2018.

[32] F. Grondman *et al.*, "Design and flight testing of incremental nonlinear dynamic inversion-based control laws for a passenger aircraft," in *2018 AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, Jan. 2018. [Online]. Available: https://doi.org/10.2514/6.2018-0385

[33] P. Ioannou and S. Baldi, *Robust Adaptive Control*. Courier Corporation, 2012.

[34] S. of the Flight Research Center, "Experience with the X-15 adaptive flight control system," NASA, Tech. Rep. TN D-6208, 1971.

[35] I. Gregory *et al.*, "Flight test of an L-1 adaptive controller on the nasa airstar flight test vehicle," in *AIAA Guidance, Navigation, and Control Conference*, 08 2010.

[36] I. Gregory, R. Gadient, and E. Lavretsky, "Flight test of composite model reference adaptive control (CMRAC) augmentation using nasa airstar infrastructure," in *AIAA Guidance, Navigation, and Control Conference*, 08 2011.

[37] K. Ackerman *et al.*, "L1 stability augmentation system for calspan's variable-stability learjet," in *Journal of Guidance, Control, and Dynamics*, 01 2016.

[38] K. A. Ackerman *et al.*, "Evaluation of an L1 adaptive flight control law on calspan's variable-stability learjet," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 4, pp. 1051–1060, Apr. 2017. [Online]. Available: https://doi.org/10.2514/1.g001730

[39] H. Matsuki *et al.*, "Flight test of fault-tolerant flight control system using simple adaptive control with PID controller," *Aircraft Engineering and Aerospace Technology*, vol. 90, pp. 00–00, 11 2017.

[40] G. Hardier, G. Ferreres, and M. Sato, "Design and flight testing of an adaptive gain-scheduled controller using on-line model estimation," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, 2018, pp. 766–773.

[41] P. Ioannou and J. Sun, *Robust Adaptive Control*, ser. Robust Adaptive Control. PTR Prentice-Hall, 1996, no. v. 1.

[42] I. Barkana, "Gain conditions and convergence of simple adaptive control," *International Journal of Adaptive Control and Signal Processing*, vol. 19, pp. 13 – 40, 02 2005.

[43] G. Hardier, G. Ferreres, and M. Sato, "On-line parameter estimation for indirect adaptive flight control: a practical evaluation of several techniques," in *2020 IEEE Conference on Control Technology and Applications (CCTA)*, 2020, pp. 180–187.

[44] S. Oudin, "Commande adaptative pour avion de transport tolérante aux erreurs de modèle et aux pannes," Ph.D. dissertation, Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), 11 2013.

[45] S. Oudin *et al.*, "Design and worst-case validation of a longitudinal adaptive flight control law : a practical approach," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 08 2013.

[46] S. Delannoy and S. Oudin, "Longitudinal control law for modern long-range civil aircrafts," in *EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation and Control*, 2013.

[47] S. Delannoy, "G*, airbus generic control laws," in *EuroGNC 2019*, 2019.

[48] C. Miller, "Nonlinear dynamic inversion baseline control law: Architecture and performance predictions," *American Institute of Aeronautics and Astronautics*, 08 2011.

[49] Y. Li *et al.*, "Incremental nonlinear control for aircraft with sensors measurement compensation," in *2021 40th Chinese Control Conference (CCC)*, 2021, pp. 7689–7694.

[50] J. Anderson, *Fundamentals of Aerodynamics*. McGraw-Hill Education, 2010.

[51] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Controls*. Design Analysis and Research Corporation, 1995, no. p. 1.

[52] M. V. Cook, "Chapter 6 - longitudinal dynamics," in *Flight Dynamics Principles*, 3rd ed. Butterworth-Heinemann, 2013, pp. 147–181.

[53] A. B. Taylor, "DC-10 winglet flight evaluation summary report," NASA, Tech. Rep. NAS1-15327, 1983.

[54] A. L. da Silva, P. Paglione, and T. Yoneyama, "Conceptual flexible aircraft model for modeling, analysis and control studies," in *AIAA Atmospheric Flight Mechanics Conference*, 2010. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2010-7806

[55] P. Apkarian, P. Gahinet, and C. Buhr, "Multi-model, multi-objective tuning of fixed-structure controllers," in *2014 European Control Conference (ECC)*, 2014, pp. 856–861.

[56] J. E. Bibel and D. S. Malyevac, "Guidelines for the selection of weighting functions for H-Infinity control," Naval Surface Warfare Center, Dahlgren Division, Tech. Rep., 1992.

[57] H. Lhachemi, "Synthèse et validation d'un système de commandes de vol robuste et autoséquencé," Master's thesis, École Polytechnique de Montréal, 12 2013.

[58] M. Dulau and S.-E. Oltean, "The effects of weighting functions on the performances of robust control systems," in *The 14th International Conference on Interdisciplinarity in Engineering—INTER-ENG 2020*, vol. 63, no. 1, 2020. [Online]. Available: https://www.mdpi.com/2504-3900/63/1/46

[59] H. Mooji, "Criteria for low-speed longitudinal handling qualities of transport aircraft with closed-loop flight control systems," Ph.D. dissertation, Delft University of Technology, 1984.

[60] Tuninggoal.steptracking class - step response requirement for control system tuning. [Online]. Available: https://www.mathworks.com/help/control/ref/tuninggoal.steptracking-class.html

[61] H. Akima, "A new method of interpolation and smooth curve fitting based on local procedures," *Journal of the Association for Computing Machinery*, vol. 17, no. 4, p. 589–602, 1970. [Online]. Available: https://doi.org/10.1145/321607.321609

[62] systune - tune fixed-structure control systems modeled in matlab. [Online]. Available: https://www.mathworks.com/help/control/ref/lti.systune.html

[63] M. Nijmeijer, "A parallel root-finding algorithm," *LMS Journal of Computation and Mathematics*, vol. 18, pp. 713–729, 01 2015.

[64] J. Bosworth, "Flight results of the NF-15B intelligent flight control system (ifcs) aircraft with adaptation to a longitudinally destabilized plant," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2012. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2008-6985

[65] M. Sharma, E. Lavretsky, and K. Wise, "Application and flight testing of an adaptive autopilot on precision guided munitions," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 06 2006. [Online]. Available: https://doi.org/10.2514/6.2006-6568

APPENDIX A PARTIAL FRACTION DECOMPOSITION OF THE THIRD ORDER n_z RESPONSE

Starting from eq. 3.5.1:

$$\frac{\gamma\zeta\omega^3(T_{K_{ff}}s+1)}{s(s+\gamma\zeta\omega)(s^2+2\zeta\omega s+\omega^2)} = \frac{A}{s} + \frac{Bs+D}{s^2+2\zeta\omega s+\omega^2} + \frac{E}{s+\gamma\zeta\omega} \tag{A.1}$$

the equation to solve is therefore :

$$\gamma\zeta\omega^3\left(T_{K_{ff}}s+1\right) = (s+\gamma\zeta\omega)(s^2+2\zeta\omega s+\omega^2)A+(s+\gamma\zeta\omega)(Bs+D)s+(s^2+2\zeta\omega s+\omega^2)Es \tag{A.2}$$

Once developed, one gets:

$$s^3: \quad 0 = A+B+E \tag{A.3a}$$
$$s^2: \quad 0 = B\gamma\zeta\omega + A(\gamma\zeta\omega+2\zeta\omega) + 2\zeta\omega E + D \tag{A.3b}$$
$$s: \quad T_{K_{ff}}\gamma\zeta\omega^3 = A(\omega^2+2\gamma\zeta^2\omega^2) + D\gamma\zeta\omega + E\omega^2 \tag{A.3c}$$
$$s^0: \quad \gamma\zeta\omega^3 = \gamma\zeta\omega^3 A \tag{A.3d}$$

Therefore, $A = 1$ and $B = -1 - E$. Substituting this into the remaining equations yields:

$$0 = (-1-E)\gamma\zeta\omega + \gamma\zeta\omega + 2\zeta\omega + 2\zeta\omega E + D \tag{A.4a}$$
$$T_{K_{ff}}\gamma\zeta\omega^3 = \omega^2 + 2\gamma\zeta^2\omega^2 + D\gamma\zeta\omega + E\omega^2 \tag{A.4b}$$

Solving for D from A.4a and substituting gives:

$$D = -\left(-[E+1]\gamma\zeta\omega + \gamma\zeta\omega + 2\zeta\omega[1+E]\right) = -\left(E[2-\gamma]\zeta\omega + 2\zeta\omega\right) \tag{A.5a}$$
$$T_{K_{ff}}\gamma\zeta\omega^3 = \omega^2 + 2\gamma\zeta^2\omega^2 + E\omega^2 - \gamma\zeta\omega\left(E[2-\gamma]\zeta\omega + 2\zeta\omega\right) \tag{A.5b}$$
$$T_{K_{ff}}\gamma\zeta\omega^3 = E\left(\omega^2+\gamma^2\zeta^2\omega^2-2\gamma\zeta^2\omega^2\right) + \omega^2 \tag{A.5c}$$
$$E = \frac{T_{K_{ff}}\gamma\zeta\omega - 1}{\gamma^2\zeta^2 - 2\gamma\zeta^2 + 1} \quad \text{if} \quad 0 \neq \gamma^2\zeta^2 - 2\gamma\zeta^2 + 1 \tag{A.5d}$$

The assumption that $0 \neq \gamma^2\zeta^2 - 2\gamma\zeta^2 + 1$ is valid. Manipulating this equation yields:

$$-\frac{1}{\zeta^2} = \gamma^2 - 2\gamma \tag{A.6}$$

Looking for extremums leads to:

$$\frac{\partial(\gamma^2 - 2\gamma)}{\partial\gamma} = 2\gamma - 2 = 0 \implies \gamma = 1 \tag{A.7a}$$

$$\frac{\partial^2(\gamma^2 - 2\gamma)}{\partial\gamma^2} = 2 > 0 \tag{A.7b}$$

Hence, the minimum is in $\gamma = 1$, which gives -1 and A.6 can only be true if $\zeta \geq 1$, which does not respect the assumption of an oscillating response. This completes the development that lead to eq. 3.53a.

APPENDIX B INVERSE LAPLACE OF THE n_z RESPONSE WITH ZEROS

Considering a second-order transfer function with two real zeros of opposite signs for eq. 3.37a, the step time response is given by:

$$e(t) = \mathcal{L}^{-1}\left\{\frac{{\omega_n}^2}{s}\left(\frac{(\tau s+1)(\tau s-1)}{s^2+2\zeta\omega_n+{\omega_n}^2}+\frac{1}{s^2+2\zeta\omega_n+{\omega_n}^2}\right)\right\} \tag{B.1a}$$

$$= \mathcal{L}^{-1}\left\{\frac{\tau^2{\omega_n}^2 s^2}{s(s^2+2\zeta\omega_n+{\omega_n}^2)}\right\} \tag{B.1b}$$

$$= \mathcal{L}^{-1}\left\{\frac{\tau^2{\omega_n}^2 s}{(s+\zeta\omega_n)^2+{\omega_d}^2}\right\} \tag{B.1c}$$

$$= \tau^2{\omega_n}^2\sqrt{\frac{1+2\zeta^2}{1+\zeta^2}}\sin\{\omega_d+\phi\}e^{-\zeta\omega_n t} \tag{B.1d}$$

Aiming to get a bound on this time response, we get:

$$|e(t)| = \left|\tau^2{\omega_n}^2\sqrt{\frac{1+2\zeta^2}{1+\zeta^2}}\sin\{\omega_d+\phi\}e^{-\zeta\omega_n t}\right| \tag{B.2a}$$

$$\leq \sqrt{1.5}{\omega_n}^2\tau^2 e^{-\zeta\omega_n t} \tag{B.2b}$$

Assuming $\zeta < 1$. This concludes the development for 3.37a.

APPENDIX C SOLUTION TO THE EQUIVALENT FEEDBACK DYNAMICS FILTER PROBLEM

The annex is intended to lead to the equivalent filter definition from [47]. This development is included for completeness of this document. Given ω_1, $\omega_2 = 0.5\omega_1$, $G(s)$, the transfer function and $g_i = M(G(\omega_i))$, the desired filter $B(s)$, such that $M(B(\omega_i)) = M(G(\omega_i))$, is given by:

$$B(s) = \frac{{\omega_0}^2}{s^2 + 2\zeta\omega_0 + {\omega_0}^2} \tag{C.1}$$

Looking for the values of ω_0 and ζ:

$$M(B(\omega)) = \frac{1}{\sqrt{\left(1 - \frac{\omega^2}{\omega_0^2}\right)^2 + 4\zeta^2\frac{\omega^2}{\omega_0^2}}} \tag{C.2a}$$

$$\frac{1}{g_i^2} = 1 - \frac{2\omega_i^2}{\omega^2} + \frac{\omega_i^4}{\omega_0^4} + 4\zeta^2\frac{\omega_i^2}{\omega_0^2} \tag{C.2b}$$

$$\frac{4\zeta^2}{\omega_0^2} = \frac{1}{\omega_i^2}\left(\frac{1}{g_i^2} - 1 + 2\frac{\omega_i^2}{\omega_0^2 - \frac{\omega_i^4}{\omega_0^4}}\right) \tag{C.2c}$$

$$\omega_2^2\left(\frac{1}{g_1^2} - 1 + 2\frac{\omega_1^2}{\omega_0^2 - \frac{\omega_1^4}{\omega_0^4}}\right) = \omega_1^2\left(\frac{1}{g_2^2} - 1 + 2\frac{\omega_2^2}{\omega_0^2 - \frac{\omega_2^4}{\omega_0^4}}\right) \tag{C.2d}$$

The last equation can be developed into:

$$\frac{1}{g_1^2} - \frac{4}{g_2^2} = \frac{3\omega_1^4}{4\omega_0^4} - 3 \tag{C.3}$$

Defining $X = \frac{\omega_1^2}{\omega_0^2}$, the solution is given by:

$$\omega_0 = \sqrt{\frac{\omega_1^2}{X^2}} \tag{C.4}$$

$$\zeta = \sqrt{\frac{1}{4X}\left(\frac{1}{g_1^2} - (1 - X)^2\right)} \tag{C.5}$$

where X can be evaluated from :

$$\frac{1}{g_1^2} - \frac{4}{g_2^2} = 3X^2 - 3 \tag{C.6}$$

APPENDIX D CONSTRAINTS FOR SHAPING H_∞ GAIN SURFACES

Additionnal gain constraints

From the K_{ff} example detailed in section 4.4.3, it can be seen that enforcing the sign of the partial derivative of a gain or the sign of a gain can be an interesting design tool. Developments will be done for a second-order polynomial as in equation D.1, where $\bar{x}$ and $\bar{y}$ are normalized scheduling variables such that $\bar{x}$ and $\bar{y} \in [-1, 1]$

$$K = K_0 + K_{10}\bar{x} + K_{01}\bar{y} + K_{11}\bar{x}\bar{y} + K_{20}\bar{x}^2 + K_{02}\bar{y}^2 \tag{D.1}$$

Strictly decreasing gain To ensure K is always decreasing with regards to variable $\bar{x}$, $\frac{\partial K}{\partial \bar{x}} = K_{10} + K_{11}\bar{y} + 2K_{20}\bar{x} < 0$. Also, $\frac{\partial^2 K}{\partial \bar{x}^2} = 2K_{20} > 0$, as greater gains are expected for low dynamic pressure. It is therefore clear that either (or both) K_{10} or $K_{11}\bar{y}$ must be negative. Forcing $K_{20} < 0$ can be done directly with the `Minimum` or `Maximum` proprieties of `realp` objects. The constraint on the first derivative is more complex to implement, since H_∞ are concerned with norms (absolute values). One possible implementation is :

$$\left|\left|\frac{c + K_{10} + K_{11}\bar{y} + 2K_{20}\bar{x}}{c}\right|\right| < 1 \tag{D.2}$$

But since $\bar{x}$ and $\bar{y}$ are bounded, it is sufficient to enforce :

$$\left|\left|\frac{c + K_{10} + K_{11} + 2K_{20}}{c}\right|\right| < 1 \tag{D.3}$$

$$\left|\left|\frac{c + K_{10} - K_{11} + 2K_{20}}{c}\right|\right| < 1 \tag{D.4}$$

where c is a constant big enough to ensure the numerator does not change sign. Note that while c must be a bound on the limited quantity, using a needlessly high value will limit the optimization, as the constraint will always be close to 1, significantly degrading the convergence.

Limits on gain values Strict positiveness may be enforced using a similar formulation as D.2, but without additional constraints, this is less practical (especially if no global minimum

of K exists), as it must be checked for all or a large subset of synthesis points :

$$\left\| \frac{c + K(\bar{x}, \bar{y})}{c} \right\| < 1 \tag{D.5}$$

It is simple to modify D.5 to enforce negative gains or define a maximum/minimum, although this does not alleviate the practicality of this constraint. For dynamic values (i.e. the steady-state value of the integrator), this may be implemented to ensure the command due to a specific gain is always within given bounds. To do so, $K(\bar{x}, \bar{y})$ must be replaced by the transfer function of interest. This requirement can then be limited to the frequency range of interest, either with the `Focus` propriety of `TuningGoal` objects or with a filter used as a weighting function (although this approach is expected to be more computationally demanding for cases where only steady-state values are needed).

APPENDIX E POLE PLACEMENT EQUATIONS WITH WASHOUT AND PURE DERIVATOR

Considering the washout and pure derivator, the closed-loop system for pole placement is shown in figure E.1. The closed-loop transfer function becomes:

$$\frac{E}{C_c^*} = \frac{-[K_d s^2 + (K_{ff} + K_p)s + K_i] N_\theta D_{\tau_w}}{D_f D_\theta D_{A/C} D_{\tau_w} s - N_\theta \left(\tau_w s^2 \left(s - a_{11}\right) K_q + \left(\left(s - a_{11}\right)\beta + N_{n_z}\right)\left(K_d s^2 + K_p s + K_i\right) D_{\tau_w}\right)} \tag{E.1}$$

where the washout is given by $\frac{\tau_w s}{D_{\tau_w}}$ and $D_{\tau_w} = \tau_w s + 1$. Developing $D_f D_\theta D_{A/C} D_{\tau_w} s$ in powers of s gives:

$$s^8; \quad [AC_8] = a\theta_2 K_2 \tau_w \tag{E.2a}$$

$$s^7; \quad [AC_7] = \tau_w[a(\theta_1 K_2 + K_1\theta_2) + b\theta_2 K_2] + a\theta_2 K_2 \tag{E.2b}$$

$$s^6; \quad [AC_6] = \tau_w[a(\theta_0 K_2 + \theta_1 K_1 + \theta_2 k_0) + b_2(\theta_2 K_1 + \theta_1 K_2) + d\theta_2 K_2] + a(\theta_1 K_2 + K_1\theta_2) + d\theta_2 K_2 \tag{E.2c}$$

$$s^5; \quad [AC_5] = \tau_w[a(\theta_0 K_1 + \theta_1 K_0) + b(\theta_0 K_2 + \theta_1 K_1 + \theta_2 K_0) + d(\theta_2 K_1 + \theta_1 K_2)] + a(\theta_0 K_2 + \theta_1 K_1 + \theta_2 K_0) + b(\theta_2 K_1 + \theta_1 K_2) + d\theta_2 K_2 \tag{E.2d}$$

$$s^4; \quad [AC_4] = \tau_w[a\theta_0 K_0 + b(\theta_0 K_1 + \theta_1 K_0) + d(\theta_0 K_2 + \theta_1 K_1 + \theta_2 K_0)] + a(\theta_0 K_1 + \theta_1 K_0) + b(\theta_0 K_2 + \theta_1 K_1 + \theta_2 K_0) + d(\theta_2 K_1 + \theta_1 K_2) \tag{E.2e}$$

$$s^3; \quad [AC_3] = \tau_w[b\theta_0 K_0 + d(\theta_0 K_1 + \theta_1 K_0)] + a\theta_0 K_0 + b(\theta_0 K_1 + \theta_1 K_0) + d(\theta_0 K_2 + \theta_1 K_1 + \theta_2 K_0) \tag{E.2f}$$

$$s^2; \quad [AC_2] = \tau_w d\theta_0 K_0 + b\theta_0 K_0 + d(\theta_0 K_1 + \theta_1 K_0) \tag{E.2g}$$

$$s^1; \quad [AC_1] = d\theta_0 K_0 \tag{E.2h}$$

$$s^0; \quad [AC_0] = 0 \tag{E.2i}$$

Developing $-\left(\theta_{j,n}\right)\left(\tau_w s^2 \left(s - a_{11}\right) K_q + \left(\left(s - a_{11}\right)\beta + N_{n_z}\right)\left(K_d s^2 + K_p s + K_i\right) D_{\tau_w}\right)$ into pow-

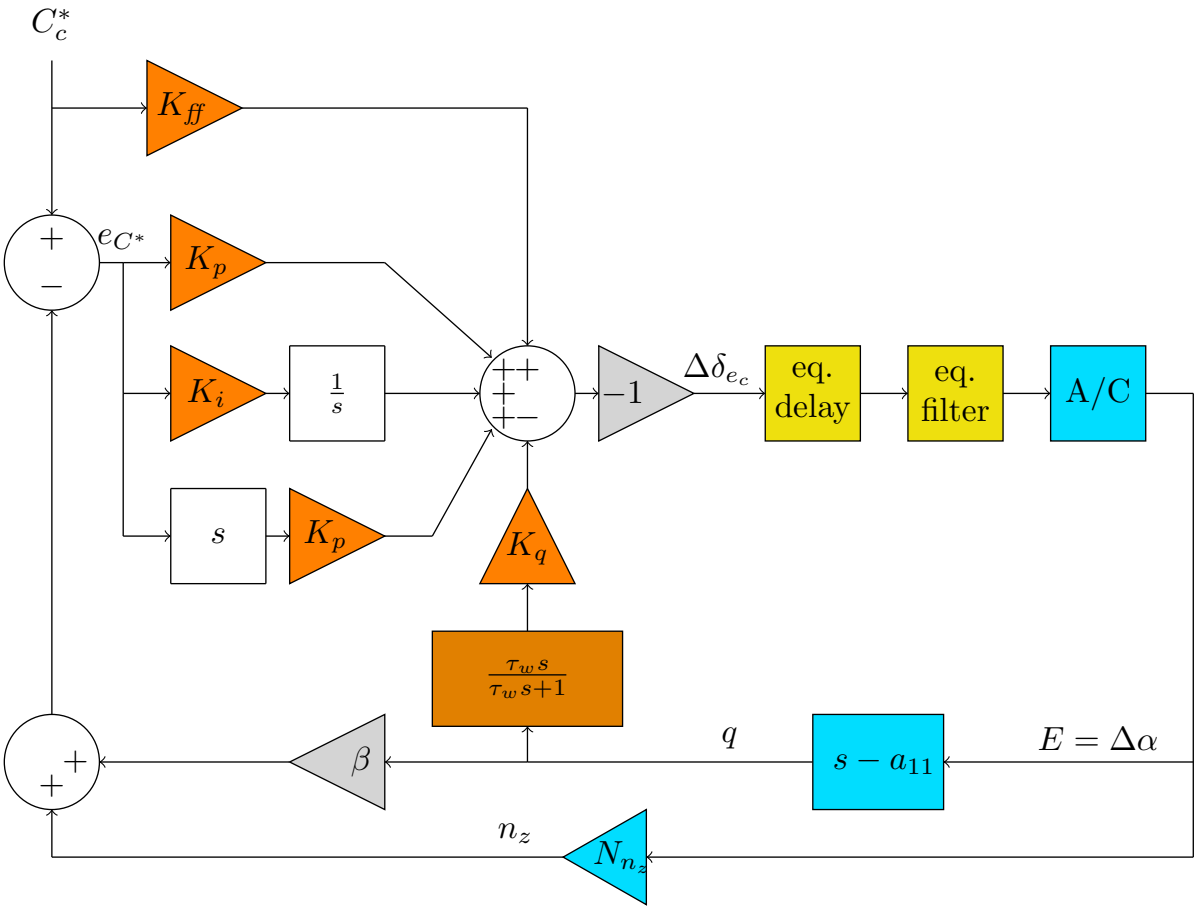


Figure E.1 C* architecture for pole placement

ers of s:

$$s^8; \quad 0 \tag{E.3a}$$
$$s^7; \quad 0 \tag{E.3b}$$
$$s^6; \quad -\theta_2[1] \tag{E.3c}$$
$$s^5; \quad \theta_1[1] - \theta_2[2] \tag{E.3d}$$
$$s^4; \quad -\theta_0[1] + \theta_1[2] - \theta_2[3] \tag{E.3e}$$
$$s^3; \quad -\theta_0[2] + \theta_1[3] - \theta_2[4] \tag{E.3f}$$
$$s^2; \quad -\theta_0[3] + \theta_1[4] - \theta_2[5] \tag{E.3g}$$
$$s^1; \quad -\theta_0[4] + \theta_1[5] \tag{E.3h}$$
$$s^0; \quad -\theta_0[5] \tag{E.3i}$$

where:

$$[1] = \mathbf{K_d}\beta\tau_w \tag{E.4a}$$
$$[2] = \tau_w K_q + \beta K_p \tau_w + \mathbf{K_d}(\beta + [N_{n_z} - a_{11}\beta]\tau_w) \tag{E.4b}$$
$$[3] = -\tau_w K_q a_{11} + \beta(K_i\tau_w + K_p) + (N_{n_z} - a_{11}\beta)\tau_w K_p + (N_{n_z} - a_{11}\beta)\mathbf{K_d} \tag{E.4c}$$
$$[4] = \beta K_i + (N_{n_z} - a_{11}\beta)(K_p + K_i\tau_w) \tag{E.4d}$$
$$[5] = (N_{n_z} - a_{11}\beta)K_i \tag{E.4e}$$

From here, it is possible to re-define the characteristic equation as:

$$\begin{aligned} &T_8 s^8 + T_7 s^7 + T_6 s^6 + (T_5 + K_q c_{5,q} + K_p c_{5,p} + K_i c_{5,i}) s^5 + (T_4 + K_q c_{4,q} + K_p c_{4,p} + K_i c_{4,i}) s^4 \\ &+ (T_3 + K_q c_{3,q} + K_p c_{3,p} + K_i c_{3,i}) s^3 + (T_2 + K_q c_{2,q} + K_p c_{2,p} + K_i c_{2,i}) s^2 \\ &+ (T_1 + K_q c_{1,q} + K_p c_{1,p} + K_i c_{1,i}) s^1 + (T_0 + K_q c_{0,q} + K_p c_{0,p} + K_i c_{0,i}) \end{aligned} \tag{E.5a}$$

where (defining $\beta_0 = \{N_{n_z} - a_{11}\beta\}$):

$$T_8 = [AC_8] \tag{E.6a}$$
$$T_7 = [AC_7] \tag{E.6b}$$
$$T_6 = [AC_6] - \theta_2\beta K_d\tau_w \tag{E.6c}$$
$$T_5 = [AC_5] + K_d[\theta_1\beta\tau_w - \theta_2(\beta + \beta_0\tau_w)] \tag{E.6d}$$
$$T_4 = [AC_4] + K_d[-\theta_0\beta\tau_w + \theta_1(\beta + \beta_0\tau_w) - \theta_2\beta_0] \tag{E.6e}$$
$$T_3 = [AC_3] + K_d[-\theta_0(\beta + \beta_0\tau_w) + \theta_1\beta_0] \tag{E.6f}$$
$$T_2 = [AC_2] + K_d[-\theta_0\beta_0] \tag{E.6g}$$
$$T_1 = [AC_1] \tag{E.6h}$$
$$T_0 = [AC_0] \tag{E.6i}$$
$$\tag{E.6j}$$

and:

$$c_{5,q} = -\theta_2\tau_w \qquad c_{5,p} = -\theta_2\beta\tau_w \tag{E.7a}$$
$$c_{4,q} = \theta_1\tau_w + \theta_2 a_{11}\tau_w \qquad c_{4,p} = \theta_1\beta\tau_w - \theta_2[\beta_0\tau_w + \beta] \tag{E.7b}$$
$$c_{3,q} = -\theta_0\tau_w - \theta_1 a_{11}\tau_w \qquad c_{3,p} = -\theta_0\beta\tau_w + \theta_1[\beta_0\tau_w + \beta] - \theta_2\beta_0 \tag{E.7c}$$
$$c_{2,q} = \theta_0 a_{11}\tau_w \qquad c_{2,p} = -\theta_0[\beta_0\tau_w + \beta] + \theta_1\beta_0 \tag{E.7d}$$
$$c_{1,q} = 0 \qquad c_{1,p} = -\theta_0\beta_0 \tag{E.7e}$$
$$c_{0,q} = 0 \qquad c_{0,p} = 0 \tag{E.7f}$$

$$c_{5,i} = 0 \tag{E.8a}$$
$$c_{4,i} = -\theta_2\beta\tau_w \tag{E.8b}$$
$$c_{3,i} = \theta_1\beta\tau_w - \theta_2[\beta + \beta_0\tau_w] \tag{E.8c}$$
$$c_{2,i} = -\theta_0\beta\tau_w + \theta_1[\beta + \beta_0\tau_w] - \theta_2\beta_0 \tag{E.8d}$$
$$c_{1,i} = -\theta_0[\beta + \beta_0\tau_w] + \theta_1\beta_0 \tag{E.8e}$$
$$c_{0,i} = -\theta_0\beta_0 \tag{E.8f}$$

Developing the desired closed-loop poles (where μ_i are imposed coefficients)

$$\left(x_5 s^5 + x_4 s^4 + x_3 s^3 + x_2 s^2 + x_1 s + x_0\right)\left(\mu_3 s^3 + \mu_2 s^2 + \mu_1 s + \mu_0\right) \tag{E.9}$$

into :

$$s^8; \quad x_5\mu_3 \tag{E.10a}$$
$$s^7; \quad x_4\mu_3 + x_5\mu_2 \tag{E.10b}$$
$$s^6; \quad x_3\mu_3 + x_4\mu_2 + x_5\mu_1 \tag{E.10c}$$
$$s^5; \quad x_2\mu_3 + x_3\mu_2 + x_4\mu_1 + x_5\mu_0 \tag{E.10d}$$
$$s^4; \quad x_1\mu_3 + x_2\mu_2 + x_3\mu_1 + x_4\mu_0 \tag{E.10e}$$
$$s^3; \quad x_0\mu_3 + x_1\mu_2 + x_2\mu_1 + x_3\mu_0 \tag{E.10f}$$
$$s^2; \quad x_0\mu_2 + x_1\mu_1 + x_2\mu_0 \tag{E.10g}$$
$$s^1; \quad x_0\mu_1 + x_1\mu_0 \tag{E.10h}$$
$$s^0; \quad x_0\mu_0 \tag{E.10i}$$

This leads to the linear equation system:

$$\begin{bmatrix} T_8 \\ T_7 \\ T_6 \\ T_5 \\ T_4 \\ T_3 \\ T_2 \\ T_1 \\ T_0 \end{bmatrix} = \begin{bmatrix} \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_2 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_1 & \mu_2 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_0 & \mu_1 & \mu_2 & \mu_3 & 0 & 0 & -c_{5,q} & -c_{5,p} & 0 \\ 0 & \mu_0 & \mu_1 & \mu_2 & \mu_3 & 0 & -c_{4,q} & -c_{4,p} & -c_{4,i} \\ 0 & 0 & \mu_0 & \mu_1 & \mu_2 & \mu_3 & -c_{3,q} & -c_{3,p} & -c_{3,i} \\ 0 & 0 & 0 & \mu_0 & \mu_1 & \mu_2 & -c_{2,q} & -c_{2,p} & -c_{2,i} \\ 0 & 0 & 0 & 0 & \mu_0 & \mu_1 & 0 & -c_{1,p} & -c_{1,i} \\ 0 & 0 & 0 & 0 & 0 & \mu_0 & 0 & 0 & -c_{0,i} \end{bmatrix} \begin{bmatrix} x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \\ x_0 \\ K_q \\ K_p \\ K_i \end{bmatrix} \tag{E.11}$$

Changing the order of variables to put the system into block-triangular form:

$$\begin{bmatrix} T_8 \\ T_7 \\ T_6 \\ T_5 \\ T_4 \\ T_3 \\ T_2 \\ T_1 \\ T_0 \end{bmatrix} = \left[\begin{array}{ccc|cccccc} \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_2 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_1 & \mu_2 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \mu_0 & \mu_1 & \mu_2 & \mu_3 & -c_{5,q} & -c_{5,p} & 0 & 0 & 0 \\ 0 & \mu_0 & \mu_1 & \mu_2 & -c_{4,q} & -c_{4,p} & \mu_3 & -c_{4,i} & 0 \\ 0 & 0 & \mu_0 & \mu_1 & -c_{3,q} & -c_{3,p} & \mu_2 & -c_{3,i} & \mu_3 \\ 0 & 0 & 0 & \mu_0 & -c_{2,q} & -c_{2,p} & \mu_1 & -c_{2,i} & \mu_2 \\ 0 & 0 & 0 & 0 & 0 & -c_{1,p} & \mu_0 & -c_{1,i} & \mu_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -c_{0,i} & \mu_0 \end{array}\right] \begin{bmatrix} x_5 \\ x_4 \\ x_3 \\ x_2 \\ K_q \\ K_p \\ x_1 \\ K_i \\ x_0 \end{bmatrix} \tag{E.12}$$

Therefore, x_5 to x_3 can easily be solved and sent to the left-hand side (process abreviated by T_i^*), reducing the order of the system:

$$\begin{bmatrix} T_5^* \\ T_4^* \\ T_3^* \\ T_2 \\ T_1 \\ T_0 \end{bmatrix} = \begin{bmatrix} \mu_3 & -c_{5,q} & -c_{5,p} & 0 & 0 & 0 \\ \mu_2 & -c_{4,q} & -c_{4,p} & \mu_3 & -c_{4,i} & 0 \\ \mu_1 & -c_{3,q} & -c_{3,p} & \mu_2 & -c_{3,i} & \mu_3 \\ \mu_0 & -c_{2,q} & -c_{2,p} & \mu_1 & -c_{2,i} & \mu_2 \\ 0 & 0 & -c_{1,p} & \mu_0 & -c_{1,i} & \mu_1 \\ 0 & 0 & 0 & 0 & -c_{0,i} & \mu_0 \end{bmatrix} \begin{bmatrix} x_2 \\ K_q \\ K_p \\ x_1 \\ K_i \\ x_0 \end{bmatrix} \tag{E.13}$$

Although the matrix structure can be exploited to eliminate x_2 and x_0 relatively easily, the system remains 4×4, limiting the appeal of the methodology for real-time use. Nonetheless, for offline gain computation, this process can be of interest.