Robust Control in a Nonlinear Context for Large Operating Domains

Spilios Theodoulis

Faculté des sciences d'Orsay

Supélec

UNIVERSITÉ PARIS-SUD 11

N° D'ORDRE : 9261

THÈSE DE DOCTORAT

SPÉCIALITÉ : PHYSIQUE

École Doctorale 'Sciences et Technologies de l'Information des Télécommunications et des Systèmes'

Présentée par :

Spilios THEODOULIS

Sujet :

'Robust Control in a Nonlinear Context for Large Operating Domains'

Soutenue le 1 Décembre 2008 devant les membres du jury :

Mme.	F. LAMNABHI-LAGARRIGUE	Présidente
MM.	J. BERNUSSOU	Rapporteur
	P. CHEVREL	Rapporteur
	G. DUC	Directeur de Thèse
Mme.	E. PANTELEY	Examinatrice
MM.	J.-P. HARCAUT	Examinateur
	C. RENAULT	Invité

Acknowledgments

I would like, first of all, to express my deepest thanks to my *thesis supervisor* at SUPELEC, Prof. Gilles DUC, who has never ceased to encourage and support me during these last three fruitful years. His unique ability to combine scientific exactness with unprecedented educational skill make him a real stand-out in his domain and environment; I consider myself greatly honored to have been his student and colleague. I shall really miss the short but witty discussions, the faultless way of doing things but most of all, a concrete and honorable person.

It is no easy task to describe in a few lines the gratitude I owe to Prof. Jacques BERNUSSOU not only for having examined all my work literally line by line, but mostly for his most rewarding trust when a young engineer knocked at his door (almost out of the blue) some four years ago. It has been a unique privilege being initiated to robust control theory by a specialist on the domain; I hope this work has given him at least as much pleasure in reading, as the one it gave me in writing it.

A really big recognition to the help I received from Dr. Jean-Philippe HARCAUT who followed my thesis during this past time. His broad knowledge on flight mechanics and automatic control was a precious asset for me without which, this work would have been a lot more difficult to conduct. His ability to discern the possible benefits of the approach proposed in this thesis has been a real encouragement! I cannot of course leave out his solemn but robust character, his pleasant attitude towards me and the exciting discussions we've had.

Let Mr. Cédric RENAULT find in these lines a small token of my appreciation for his support and useful comments regarding mostly the second benchmark example used in this thesis. It has been a pleasure working with such an excellent engineer and somebody capable of giving quick and intelligent answers to difficult questions; his demands also permitted me to further validate and feel more confident with my work.

My warmest thanks to Dr. Elena PANTELEY for having kept this project as a high priority in her interests; her valuable and most appreciated knowledge on the field of nonlinear control gave to this work a sense of completeness and rigor. Most of the material in the conclusions section and ideas springing up at various places of this report are mostly due to interesting (and long!) conversations with her. Furthermore, the personal advice and motivation from a warm personality have been important to me and I hold them with the greatest respect.

My most sincere appreciation for his comments I have for Prof. Philippe CHEVREL who has graciously accepted to be my second thesis examiner. An alternative vision of this work has been possible due to the consistent remarks from a true 'guru' of gain scheduling control in all its extensions.

Last but not least I would like to thank Dr. Françoise LAMNAHBI for being the president of my thesis examining committee and for her support mostly during the first part of this thesis, right when it has been most important.

I feel the need to thank also the Head of the Automatic Control Department of SUPELEC, Mr. Patrick BOUCHER for having accepted me as a PhD student at his laboratory. I would also like to thank the personnel of the Department who has contributed in all possible ways to my work and also for having treated me with the greatest respect. I, most of all, have appreciated the help I received from Mr. Martial DEMERLE concerning some software tools needed for my thesis (along with a fine bottle of old Hellenic wine), a warm everyday handshake by Mr. Emmanuel GODOY and a broad smile from our secretary Mme. Josiane DARTRON.

I would also like to thank three very important persons to me from the Polytechnic School of the University of Patras, for having played a key role to my development both as a personality and as an engineer: Prof. Antony TZES, Prof. George BITSORIS and Prof. Antonios ALEXANDRIDIS who have transferred to me their keen interest on the field but mainly, for their trust that I may go all the way. Finally I could not possibly omit to send my acknowledgements to Prof. Tryphon GEORGIOU for an avalanche of original ideas and for his interest on this thesis.

Let also all my friends, especially dear to me, find here my truest and warmest thoughts; even if it may not be always evident, this material is also their child and has been greatly facilitated due to their existence, jokes, advice and (why not) teasing! To Eirini for waking up the chaotic side, to Giuliana for being there all the time, to Zakaria for our innovative theory on aerodynamics and vegetation, Stratis for the ouzo, to Frédéric for being always correct, to Dorin for his dancing, Guillaume for the blues, Cristiane for her kind reception, Simona and Dragos for the big heart, and also to Nikos, Guillermo, Kalliopi, Konstantina, Sébastien, Lilian, Olivier, Omar, Hichem, Bilal, Lucie, Farag, Bastien, Haitham, Karim, Warodi, Konstantina, Fotis, Nikos, Panos and Spiros.

Last but not least I would like to ask from one of my favorite persons, my little sister Alexandra, to forgive me for been away from her for so long; her thoughts and care have always been most precious to me. Some words are finally, and above all, devoted to my loving parents Aggeliki and Dimos for all the endless sacrifices they have submitted themselves to for us all these years; as a return, let this work finally be the smallest of recompenses I owe them...

Saint-Louis, France, the 27^{th} of November 2008.

To the flower of my life, *Sophie*.

Table of Contents

Publications & Projects

Journal Articles

1. **Gain-scheduled Autopilot Synthesis for an Atmosphere Re-entry Vehicle**, S. Theodoulis and G. Duc, *AIAA Journal of Spacecraft and Rockets*, (submitted).

2. **Missile Autopilot Design: Gain Scheduling and the Gap Metric**, S. Theodoulis and G. Duc, *AIAA Journal of Guidance, Control, and Dynamics*, (to appear).

Conference Articles

1. **Gain-scheduled Autopilot Synthesis for an Atmosphere Re-entry Vehicle**, S. Theodoulis and G. Duc, *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2008, Honolulu, USA.

2. **Static Interpolated $\mathscr{H}_\infty$ Controllers for Missile Autopilot Synthesis**, S. Theodoulis and G. Duc, *Proceedings of the* 46^{th} *IEEE Conference on Decision & Control*, 2007, New Orleans, USA.

3. **Gain-Scheduling Techniques for Missile Autopilot Synthesis**, S. Theodoulis and G. Duc, *Proceedings of the 2007 European Control Conference*, 2007, Kos, Greece.

4. **Missile Autopilot Design Using Observer-Based Gain Scheduling**, S. Theodoulis and G. Duc, *Proceedings of the* 17^{th} *IFAC Symposium on Automatic Control in Aerospace*, 2007, Toulouse, France.

Communications & Workshops

1. **Pilotage d'un Véhicule de Rentrée par des Techniques de Séquencement de Gain**, S. Theodoulis and G. Duc, *Groupe de Travail GT MOSAR*, February 2008, ENSIEG, Grenoble, France.

2. **Séquencement de Gains et Gap-métrique: Application à la Synthèse d'un Pilote de Missile**, S. Theodoulis and G. Duc, *Journée Envol Recherche EADS*, December 2007, Maison de la Chimie, Paris, France.

3. **Missile Autopilot Design : Gain-Scheduling and the Gap Metric**, S. Theodoulis and G. Duc, *Groupe de Travail GT MOSAR*, September 2007, Ecole des Mines, Nantes, France.

4. **Interpolation de Gains et Application sur le Missile de 'Reichert'**, S. Theodoulis and G. Duc, *Journée Envol Recherche EADS*, December 2006, Institut Pasteur, Paris, France.

Projects

1. **Commande par Séquencement de Gains pour un Système de Suspension Magnétique**, R. Carstoiu, G. Duc and S. Theodoulis, *Engineering Diploma Thesis*, 2008, Automatic Control Dpt. - SUPELEC, France.

2. **Interpolation de Contrôleurs: Application à une Machine Asynchrone**, N. Janiaud, V. Bisgambiglia, G. Duc and S. Theodoulis, *Engineering Diploma Thesis*, 2007, Automatic Control Dpt. - SUPELEC, France.

Résumé

Introduction

De nombreux systèmes industriels sont caractérisés par des variations sur des paramètres qui affectent leur dynamique. Ce phénomène apparaît notamment dans le domaine aéronautique (missiles, avions, lanceurs, etc...) où les paramètres variants sont l'altitude, la vitesse ou l'angle d'incidence. Calculer un correcteur pour ce type de systèmes n'est pas une tâche facile car ces variations doivent être prises en compte afin d'obtenir des lois de commande performantes.

Ce travail utilise la méthode de commande par séquencement de gains afin de construire une commande adaptative qui tienne compte de ces variations. Cette méthode présente beaucoup d'avantages comme la possibilité d'utiliser les outils de la théorie moderne de la commande robuste et aussi la capacité de mettre à jour les paramètres du correcteur séquencé d'une façon simple et facilement mise en œuvre.

En revanche, malgré la grande richesse de résultats concernant cette méthode, il existe encore de nombreux problèmes comme par exemple l'absence d'une méthode systématique pour définir les points de synthèse et la structure des correcteurs linéaires et aussi la façon de les interpoler. Cette thèse essaie de donner une réponse à ces problèmes en proposant une stratégie de commande performante, robuste mais aussi simple pour la commande des systèmes non-linéaires à paramètres variants.

Partie I: Contexte Théorique

Chapitre 1

Séquencement de Gains Classique.

Le premier chapitre présente la première classe de méthodes de commande par séquencement de gains, souvent appelée approche par linéarisation et résumée par la formule 'diviser pour conquérir'. La commande par séquencement de gains s'inscrit dans le cadre des méthodes de commande dites adaptatives qui comprennent les commandes adaptatives directe et indirecte et la commande par séquencement de gains. Les deux premiers types de commande se basent sur une optimisation en temps réel afin de mettre à jour les coefficients du correcteur.

Par contre la commande par séquencement de gains se base sur un calcul hors ligne de ces coefficients, pour plusieurs conditions de fonctionnement du système, et adapte ensuite ces coefficients en utilisant la famille de correcteurs et des méthodes d'interpolation. L'interpolation/adaptation est faite à partir d'un ensemble de paramètres variants dans le temps qui capturent le changement du régime de fonctionnement du système, lequel est souvent appelé *vecteur de séquencement* $\varrho = \varrho(t)$. Cette méthode est encore divisée en deux sous-méthodes caractérisées principalement par la façon de modéliser le système à commander pour le rendre exploitable par une méthode de commande.

La première classe de méthodes est la méthode par linéarisation LBGS[1] qui se base sur l'approximation du système non-linéaire à paramètres variants par des modèles locaux linéaires calculés pour une famille de valeurs fixes du vecteur de séquencement. Pour cette famille de modèles, un ensemble de correcteurs linéaires invariants (LTI) est calculé, en utilisant par exemple les techniques modernes puissantes de la théorie de la commande robuste $\mathscr{H}_2$ ou $\mathscr{H}_\infty$, qui satisfait les performances exigées par l'utilisateur localement. A la fin, un correcteur global non-linéaire séquencé est calculé en combinant, en quelque sorte, ces correcteurs locaux grâce à une méthode d'interpolation à partir du vecteur de séquencement mesuré en temps réel. C'est cette méthode qui est considérée dans le travail de thèse présenté dans ce rapport.

La deuxième classe de méthodes est la méthode dite LPV[2] (ou q-LPV) qui est basée soit sur une famille continue de systèmes linéaires autour d'un ensemble de points ou de trajectoires d'équilibre du système non-linéaire initial, soit sur un système *quasi*-linéaire obtenu en re-formulant la dynamique non-linéaire.

La première classe de méthodes est préférée quand une approche systématique et non conservative est demandée, et la deuxième classe quand une méthode offrant de meilleures conditions de stabilité pour le système en boucle fermé séquencé est exigée.

Avant de commencer à décrire la méthode de séquencement par linéarisation, quelques concepts sur la modélisation des systèmes sont analysés. En premier lieu, des concepts sur les notions de points et de trajectoires d'équilibre sont détaillés pour les systèmes non-linéaires autonomes, non-autonomes et dépendants de paramètres. Un tel système a la forme suivante[3]:

$$\mathcal{S}_{\text{pd}}: \quad \begin{aligned} \dot{x} &= \mathbf{f}(x, u, \varrho) \\ y &= \mathbf{h}(x, u, \varrho). \end{aligned}$$

Ce système admet une famille de points d'équilibre pour chaque valeur du vecteur de séquencement dans un domaine d'opération $\mathbf{\Gamma}$. Il faut ajouter que la grande majorité des systèmes aéronautiques sont des systèmes de ce type, paramétrés par des grandeurs telles que l'incidence, l'altitude ou la vitesse.

[1] Pour 'Linearization-Based Gain Scheduling'.
[2] Pour 'Linear Parameter-Varying'.
[3] La notation 'pd' signifie 'parameter-dependent'.

A partir d'un système non-linéaire dépendant de paramètres, on pourrait considérer d'autres types de modélisation, afin d'approximer son comportement, avec une précision différente pour chaque catégorie, ou tout simplement de le re-configurer. Une approche de ce type est par exemple la modélisation q-LPV qui essaie de cacher la non-linéarité en faisant des manipulations algébriques sur le système $\mathcal{S}_{\text{pd}}$. Une autre modélisation dite LPV peut être faite en considérant l'ensemble des systèmes linéaires induit par la linéarisation du système autour d'un ensemble de points ou de trajectoires d'équilibre. En outre, la dépendance paramétrique du modèle linéaire par rapport au vecteur de séquencement pourrait être cachée en utilisant une autre modélisation appelée LFT (pour 'Linear Fractional Transformation'), dans laquelle cette dépendence paramétrique est considérée comme une incertitude mesurée.

D'autres types de modélisation d'un processus existent comme les systèmes LTV qui sont purement linéaires mais dont les matrices d'état dépendent explicitement du temps (et pas par l'intermédiaire d'un vecteur de séquencement ϱ). Ce type de systèmes peut être issu d'une linéarisation d'un système non-linéaire autour d'une trajectoire d'équilibre. Finalement, le cas le plus simple est le système LTI (pour linéaire invariant dans le temps) dont la modélisation est faite en utilisant des matrices d'état constantes ; ce type de systèmes peut aussi être le résultat de la linéarisation du système non-linéaire autour d'un point d'équilibre.

Cette dernière est faite en utilisant un développement au premier ordre du système non-linéaire de sorte que le modèle obtenu est une approximation locale qui peut sous quelques conditions offrir une bonne précision. D'autres types de linéarisation existent dans la littérature comme la linéarisation 'basée-vitesse' (c.f. Chapitre 2) ou la linéarisation entrée $\mapsto$ état ou entrée $\mapsto$ sortie.

La méthode de séquencement de gains par linéarisation comporte principalement cinq étapes : la paramétrisation des points d'équilibre et le calcul de la commande nominale ; l'obtention du modèle LPV en fonction du vecteur de séquencement ; la synthèse de correcteurs linéaires pour une famille de valeurs du vecteur de séquencement ; l'interpolation des correcteurs ; et finalement la construction du correcteur global non-linéaire séquencé et sa validation à travers des simulations.

En ce qui concerne les méthodes d'interpolation, elles sont divisées en deux familles principales : les méthodes qui ne garantissent pas la stabilité du système séquencé et les méthodes qui offrent un certain degré de stabilité. Les premières sont la commutation de correcteurs, le mélange de correcteurs, l'interpolation ZPK, l'interpolation des coefficients des fonctions de transfert, l'interpolation des matrices d'état, l'interpolation type retour d'état/observateur, l'interpolation des gains, l'interpolation des solutions des équations de Riccati, etc...

La première de ces méthodes commute les correcteurs linéaires dans des régions de séquencement, la deuxième combine les sorties des correcteurs, la troisième interpole leurs zéros, pôles et gains, la quatrième les coefficients des nominateurs et dénominateurs de leurs fonctions de transfert, la cinquième les

matrices de leurs représentations d'état, la sixième les matrices de l'observateur et du retour d'état, la septième interpole des gains (par ex. d'un correcteur PID) et la huitième les matrices des équations de Riccati associées à la synthèse $\mathscr{H}_\infty$.

Les méthodes qui offrent des garanties de stabilité sont aussi nombreuses et le travail dans le domaine date du début des années '90 (c.f. Section 1.3.3 pour plus de détails).

Chapitre 2

Séquencement de Gains Moderne.

Le deuxième chapitre présente brièvement la deuxième famille de méthodes de séquencement de gains dites LPV (ou q-LPV) ainsi que quelques approches alternatives (linéarisation 'basée-vitesse' et approche floue ou par réseaux de neurones).

La méthode LPV est aussi divisée en deux grandes catégories pour obtenir un correcteur séquencé, l'approche polytopique et l'approche LFT. La première utilise la formulation LMI du calcul d'une fonction de Lyapunov et des correcteurs robustes, pour une famille de systèmes obtenus par une modélisation polytopique ayant une dépendence affine par rapport aux paramètres de séquencement. Le séquencement est fait en mesurant les paramètres variants et en les exprimant comme la combinaison convexe des valeurs aux sommets du polytope considéré.

La méthode LFT à son tour modélise la dépendence paramétrique comme une incertitude bornée en norme et calcule un correcteur séquencé ayant une forme spécifique, afin de stabiliser le système pour toute variation possible des paramètres considérés.

La méthode dite 'linéarisation basée-vitesse' présente une modélisation du système non-linéaire à paramètres variants en linéarisant autour de n'importe quel point de fonctionnement du système et non nécessairement autour d'un point d'équilibre. La méthodologie de calcul du correcteur séquencé suit ensuite les mêmes bases que la méthode 'diviser pour conquérir'. Enfin, l'approche floue ou par réseaux de neurones utilise les idées spécifiques de cette communauté, bien différentes des autres méthodes.

Chapitre 3

Lois de Commande pour le Séquencement de Gains.

La théorie de la commande utilisée dans le cadre du calcul de la famille des correcteurs linéaires est présentée dans ce chapitre. Trois techniques différentes sont analysées : la commande $\mathscr{H}_\infty$ dans des régions LMI, la transformation de correcteurs sous forme estimateur/retour d'état et la commande $\mathscr{H}_\infty$ par loop shaping, dynamique et statique. Chaque méthode est associée à une méthode d'interpolation utilisée dans ce travail : mélange des correcteurs, interpolation des matrices d'estimateur et de retour d'état et interpolation des gains. De plus, un outil d'analyse de la distance entre des systèmes, la gap-métrique, est présenté.

En ce qui concerne la première méthode, il est clair qu'il serait intéressant de combiner les avantages d'un correcteur $\mathscr{H}_\infty$ quant à la robustesse en face de perturbations et d'incertitudes de modèle, avec les avantages du placement de pôles pour le réglage de performances (temps d'établissement, temps de réponse, dépassement, etc...). Cette combinaison peut être facilement traduite par un ensemble de contraintes matricielles affines si le domaine du placement des pôles considéré est affine.

Un correcteur dynamique d'ordre plein calculé avec la méthode précédente peut être transformé en un correcteur sous la forme d'un estimateur de Kalman et d'un gain de retour d'état combiné à un système toujours stable appelé le paramètre de Youla. En général, sous quelques hypothèses pendant la phase de synthèse, on peut forcer ce paramètre de Youla à zéro (en considérant un correcteur d'ordre égal à celui du système et strictement propre) et donc faciliter plus tard la tâche d'interpolation. Cependant cette transformation n'est pas du tout évidente car elle nécessite une analyse combinatoire des valeurs propres de la boucle fermée.

La méthode de commande $\mathscr{H}_\infty$ par loop shaping est un cas spécial de la synthèse $\mathscr{H}_\infty$ classique. Cette méthode conçue par McFarlane&Glover au début des années '90 construit un correcteur robuste afin de maximiser la marge de robustesse en face d'incertitudes bornées en norme sur les facteurs premiers de la fonction de transfert en boucle ouverte. Ce correcteur robuste minimise la norme $\mathscr{H}_\infty$ des différentes fonctions de transfert en boucle fermée dans des bandes de fréquences différentes et son calcul est divisé en trois étapes qui font partie d'une procédure appelée LSDP (pour Loop Shaping Design Procedure).

La première étape (loop shaping) consiste à calculer des pré/post compensateurs pour le système en boucle ouverte afin de régler ses performances d'une façon satisfaisante. Typiquement cela consiste à assurer un gain élevé/faible en basses/hautes fréquences et une pente de 20dB/dec dans la zone des fréquences intermédiaires. La deuxième étape est le calcul de la marge de robustesse qui peut être atteinte par un correcteur $\mathscr{H}_\infty$ robuste ; si cette marge est satisfaisante on peut passer à la prochaine étape, sinon il faudra recalculer les pré/post-compensateurs. Enfin, la troisième étape est de calculer un correcteur $\mathscr{H}_\infty$ robuste, pour la boucle ouverte comprenant les pré/post-compensateurs et le système, afin de le rendre stable en face d'incertitudes de modèle. Le correcteur final est la combinaison des pré/post-compensateurs avec le correcteur robuste. Le correcteur robuste étant d'ordre plein, il peut être calculé soit en utilisant l'approche classique de Doyle&Glover soit la formulation plus attirante par LMI.

En outre, la méthode par loop shaping peut être considérée en utilisant un correcteur statique, qui est évidemment plus facile à interpoler qu'un correcteur d'ordre plein. La marge de robustesse atteinte avec cette approche est toujours inférieure au cas dynamique mais souvent très satisfaisante. En revanche, la formulation LMI est issue de conditions seulement suffisantes car le problème, initialement sous forme BMI, est rendu convexe en supprimant les termes bilinéaires.

La gap-métrique est un outil très intéressant introduit au début des années '80 et exploité dans les années '90 comme une méthode pour caractériser la stabilisabilité d'un système perturbé par le correcteur calculé pour un système nominal. Sa propriété la plus étonnante est qu'elle définit une certaine distance entre ces deux systèmes liée à la marge de robustesse atteinte par le correcteur nominal. Cet outil permettra de créer un algorithme de choix des points de fonctionnement ou de synthèse pour un système non-linéaire à paramètres variants au Chapitre 6 et ainsi d'améliorer le comportement du correcteur séquencé.

Partie II: Contexte Applicatif

Chapitre 4

Modélisation et Analyse.

Les deux exemples d'application considérés dans ce travail sont les suivants : un autopilote de poursuite pour l'axe de tangage d'un missile fortement manœuvrant et un autopilote de régulation pour l'axe de tangage d'un véhicule de rentrée atmosphérique. L'autopilote du missile doit fournir des commandes de braquage afin d'assurer la poursuite de consignes en accélération normale de différentes amplitudes. Le pilote doit assurer une performance rapide et robuste pour toute l'enveloppe de vol du missile formée par ses paramètres variants : l'angle d'incidence et le nombre de Mach. Concernant le véhicule de rentrée, l'autopilote doit maintenir un angle d'incidence constant malgré la variation du Mach. D'autre part, il doit aussi être robuste en face de perturbations paramétriques et avoir des marges de stabilité satisfaisantes.

Dans ce chapitre sont présentés, d'une façon parallèle, les résultats concernant les deux premières étapes de la procédure LBGS, qui est commune aux trois méthodes d'interpolation utilisées : la paramétrisation des points d'équilibre et le calcul de la commande nominale, et l'obtention du système linéarisé en fonction du vecteur de séquencement.

Concernant le missile, un modèle non-linéaire analytique est disponible sous forme d'équations différentielles. L'état x se compose de l'angle d'incidence α et de la vitesse de tangage q et les sorties mesurées sont l'accélération normale η et la vitesse de tangage. Vu que l'angle d'incidence n'est pas mesuré, le vecteur de séquencement ne peut comprendre que le Mach et l'accélération. Une analyse algébrique est faite pour exprimer les points d'équilibre en fonction de ces variables et la commande nominale $\delta(\varrho_{\text{eq}})$ (et la surface correspondante) est finalement calculée symboliquement.

Après avoir calculé la commande nominale, le modèle linéaire pour chaque point d'équilibre, paramétré par le vecteur de séquencement, est aussi obtenu symboliquement. L'analyse de stabilité de ce modèle montre que le système est stable dans la majorité de l'enveloppe de vol en présentant des pôles mal amortis. Cependant, il y a une petite partie où le modèle est instable, notamment pour des faibles valeurs d'accélération et un Mach élevé. De plus les zéros du

système, étant dans le demi-plan complexe droit, introduisent un déphasage non minimal.

Concernant le véhicule de rentrée, la modélisation est faite en utilisant des tableaux de données pour les fonctions aérodynamiques. Pourtant dans ce cas l'enveloppe de vol est paramétrée en utilisant l'angle d'incidence et le Mach ; de plus la pression dynamique varie en fonction du temps car le scénario de rentrée suppose que l'altitude varie aussi d'une façon significative. Enfin, pour les deux exemples, la commande de pilotage est délivrée par un actionneur modélisé par un filtre du deuxième ordre.

La famille des modèles linéaires est obtenue en considérant le développement au premier ordre du modèle non-linéaire à paramètres variants du départ. Il s'avère que le modèle linéarisé est dans la limite de stabilité car il présente des valeurs propres sur l'axe imaginaire du plan complexe. La fréquence propre ω_0 des pôles complexes varie beaucoup avec le Mach et l'angle d'incidence et pose un problème de commande assez stimulant car le correcteur séquencé doit à la fois garantir un amortissement suffisant pour les pôles du système mais aussi limiter la commande exigée. De plus, il est clair (vue la forme de la surface obtenue pour la fréquence propre - c.f. Fig. 4.15c) que le choix des points de synthèse n'est pas évident et qu'une stratégie simpliste ne considérant que les sommets de l'enveloppe de vol ne suffira pas.

Les deux prochains chapitres sont consacrés à l'analyse et à la validation des méthodes de séquencement conçues dans cette thèse, afin d'affronter le problème du calcul d'une commande séquencée performante mais simple : le premier présente deux méthodes 'ad-hoc' tandis que le deuxième détaille la méthode systématique appelée *e*-LSDP.

Chapitre 5

Stratégies de Commande Ad-hoc.

Ce chapitre présente deux méthodes d'interpolation pour la commande du missile, basées sur la méthode de mélange des correcteurs et sur la méthode qui utilise une structure estimateur et retour d'état. Au début du chapitre sont présentés à la fois une recherche bibliographique assez extensive sur le calcul des autopilotes pour ce type de systèmes et les objectifs pour l'autopilote non-linéaire. Concernant les autopilotes pour chaque méthode, il est clair que l'analyse de ce chapitre concerne les trois dernières étapes de la LSDP notamment : la synthèse des correcteurs linéaires, l'interpolation de ces correcteurs et enfin la construction de la loi de commande globale séquencée.

En ce qui concerne la méthode de mélange de correcteurs, la synthèse linéaire est faite autour de neuf points de fonctionnement au total (les quatres sommets de l'enveloppe de vol plus des points additionnels). La méthode de synthèse choisie est du type sensibilité mixte S/KS, la boucle ouverte comprenant un intégrateur sur l'erreur de suivi de consigne et le modèle de l'actionneur. Le modèle standard $P(s)$ obtenu possède une commande (l'angle de braquage commandée δ_c), trois mesures (l'erreur de suivi de la consigne e_δ, l'intégrale de cette

erreur v_δ et la vitesse de tangage q_δ), une entrée de perturbation (la référence η_r) et deux signaux de performance (la dérivée de la commande $\zeta_{\dot{\delta}}$ et l'erreur de suivi de consigne ζ_e, les deux conditionnées avec des gains de pondération). Pour ce modèle linéaire, un correcteur $\mathscr{H}_\infty$ dynamique d'ordre cinq (le même que la boucle ouverte) avec des contraintes de placement des pôles dans des régions LMI, à été calculé. Ses performances dans le cas linéaire sont très satisfaisantes mais l'implantation du correcteur non-linéaire séquencé pose un grand nombre de problèmes.

Le premier problème porte sur la ré-initialisation des correcteurs linéaires au cours du passage d'une région d'interpolation vers la prochaine. Vu que la représentation d'état des différents correcteurs et leurs gains statiques ne sont pas les mêmes durant le passage, il y aura des incompatibilités d'amplitude sur le signal de commande ; ce souci peut être réglé en utilisant un conditionnement spécifique des états initiaux des correcteurs. De plus, quand un nouvel ensemble de correcteurs est introduit, il y aura des transitoires sur la commande. Ce souci peut être aussi atténué en utilisant une dynamique rapide pour les correcteurs linéaires, en espérant que ce transitoire disparaîtra assez rapidement ; de plus cette atténuation devient plus grande si le vecteur de séquencement varie lentement.

Le deuxième problème de cette méthode est qu'afin de règler ces derniers soucis, le correcteur global séquencé devient plus compliqué car un mécanisme de supervision est nécessaire pour ré-initialiser les états. De plus, en calculant des correcteurs linéaires très rapides, on risque d'exciter des modes hautes fréquences mal modélisés.

En revanche, cette méthode possède quelques avantages comme par exemple une faible quantité de calculs pour le correcteur non-linéaire et le fait que des correcteurs linéaires de différentes tailles ou structures peuvent être considérés.

En ce qui concerne la méthode d'interpolation type observateur/retour d'état, les correcteurs linéaires sont les mêmes que ceux utilisés par la méthode précédente ; cependant la paramétrisation de Youla a été utilisée afin de les exprimer sous cette forme. L'interpolation est faite sur les matrices de l'observateur et du retour d'état mais aussi sur les matrices de la représentation d'état. Cette méthode offre de bonnes performances de poursuite pour le scénario utilisé ; pourtant elle aussi présente des inconvénients assez importants.

Le premier inconvénient est que cette méthode est très gourmande en calculs car toutes les matrices de l'estimateur doivent être interpolées ; ce dernier point n'est pas évident car la façon de calculer les matrices de la dynamique du système doit être éclaircie (c.f. Section 5.5.2). Le deuxième inconvénient de cette méthode porte sur le fait que la répartition des pôles de la boucle fermée entre les pôles de l'observateur et du correcteur n'est pas unique et elle ne peut pas être faite automatiquement d'une façon facile, quand il y a par exemple des pôles complexes conjugués multiples.

En plus des inconvénients de chaque méthode, il y a d'autres soucis concernant ces deux méthodes d'interpolation comme l'ordre élevé des correcteurs

linéaires considérés, l'absence d'une méthode systématique pour choisir les points de synthèse ou bien la complexité de l'interpolation. Ces problèmes sont réglés pour les deux exemples considérés en utilisant la *e*-LSDP appliquée au chapitre suivant.

Chapitre 6

Stratégies de Commande Systématiques.

Le dernier chapitre de la thèse détaille une méthodologie innovante, systématique et performante pour la construction d'un correcteur global non-linéaire. Cette méthodologie (appelée la *e*-LSDP) correspond aussi aux trois dernières étapes de la LBGS, elle utilise la connection de la théorie de la gap-métrique avec la commande $\mathscr{H}_\infty$ par loop shaping et elle est appliquée à la fois à la commande du missile et du véhicule de rentrée.

La première étape est d'augmenter le modèle linéaire du missile ou du véhicule avec des pré/post-compensateurs afin de régler leurs réponses fréquentielles. Ceci est fait pour le missile en utilisant un correcteur P/PI sur ses boucles interne et externe autour de neuf points de fonctionnement, et pour le véhicule en utilisant un correcteur PID filtré autour de cinq points de fonctionnement. Les performances atteintes dans le cas linéaire sont inférieures à celles obtenues avec les correcteurs $\mathscr{H}_\infty$ du chapitre précédent mais elles restent tout à fait satisfaisantes vu l'ordre très faible des correcteurs utilisés. Cependant, un correcteur séquencé utilisant seulement ces correcteurs type-PID n'est pas suffisamment performant.

Pour cette raison, un algorithme de choix des points de fonctionnement additionnels est mis en œuvre en utilisant la théorie de la gap-métrique. Plus précisément, en tenant compte de la connexion entre la marge de robustesse ϵ atteinte par un correcteur robuste $\mathscr{H}_\infty$ pour un système linéaire et la distance δ_{g} entre ce système linéaire et un système perturbé, on peut trouver un ensemble de points qui captent au mieux la variation de la dynamique du système non-linéaire à paramètres variants. Les correcteurs robustes utilisés sont de type retour de sortie statique et possèdent une complexité minimale par rapport à un correcteur d'ordre plein. Pour le cas du missile il y a deux gains robustes pour chaque boucle de commande, pour un total de douze points considérés dans l'enveloppe de vol. Pour le cas du véhicule il y a trois gains pour chaque chaîne du correcteur PID calculés pour six points le long d'une ligne d'angle d'incidence constant.

Tous les gains sont interpolés en utilisant une combinaison linéaire dans des régions d'interpolation pour les deux systèmes, en minimisant ainsi la complexité d'implantation. Le correcteur non-linéaire est obtenu en combinant l'action en boucle ouverte, les correcteurs séquencés et quelques filtres du type 'feedforward' afin de mieux calibrer la réponse du système.

La stratégie de commande proposée pour les scénarios choisis est testée sur le modèle non-linéaire du véhicule et du missile et les résultats sont très satisfaisants. De plus, des analyses de type Monte Carlo, linéaires à temps figé, etc... sont faites et la robustesse de cette méthode est ainsi démontrée.

Conclusion

Cette thèse a montré, en s'appuyant sur quelques cas d'application concrets, qu'une structure très simple de correcteurs linéaires pouvait amener à un correcteur global séquencé de très haute performance, sous l'hypothèse que les correcteurs locaux soient calculés autour de points de synthèse choisis d'une façon intelligente.

Dans ce but, une méthode innovante appelée e-LSDP à été conçue et testée sur deux exemples d'application différents. Cette méthode est basée sur une connexion spécifique entre la théorie de la gap-métrique et de la commande $\mathscr{H}_\infty$ par loop shaping permettant de prendre en compte la variation de la dynamique du système non-linéaire.

Ce travail peut être étendu en considérant d'autres types d 'algorithmes qui pourraient prendre en compte la variation de cette dynamique d'une façon plus intelligente ou aussi considérer d'autres structures pour les correcteurs robustes. De plus, cette méthode pourrait être testée afin d'affronter le problème 3-axes pour le missile comme pour le véhicule, utiliser des modèles non-linéaires qui prennent en compte des modes souples, etc...

Foreword

Many industrial systems are characterized by fast variations on their parameters characterizing their (in most cases) nonlinear dynamics. This is particularly the case of aeronautical systems such as missiles, aircrafts, launchers etc. whose dynamics depend on a number of parameters such as altitude, speed and angle of attack. Designing a controller for such systems (consider for example a missile autopilot that realizes desired vertical acceleration commands issued from a guidance loop) is not an easy task, since these parameter variations need to be taken into account in some way if the designer desires to achieve highly performing and also robust control laws.

A widely recognized and highly successful control strategy for these nonlinear, parameter-dependent systems is *gain scheduling control* and more specifically *linearization-based* gain scheduling control or 'divide and conquer'[4].

This method uses a set of linear time-invariant controllers, with each member of the set being designed for a local linear model of the nonlinear system around an equilibrium-operating point. An equilibrium point is parameterized by the so-called *scheduling vector* including some (or all) of the nonlinear system's time-varying parameters. These parameters take values inside a specified range, thus forming the system's *operating domain* (or flight envelope for aeronautical systems). The aforementioned set of linear controllers is finally combined/interpolated in some way, according to the variation of these time-varying parameters (or scheduling vector), in order to compute a control law that continuously adapts itself to these changing dynamics, thus ensuring stability, performance and robustness for the whole system's operating domain.

The advantages of this method are numerous: for example, the wealth of LTI modern robust/optimal control theory (such as $\mathscr{H}_2, \mathscr{H}_\infty$) and their convex LMI formulations for the design of the local controllers may be used, thus providing all these nice features such as robustness to parametric variations, noise attenuation, control effort reduction etc. This theory provides controllers that are highly preferred and have prevailed, as far as industrial real-time implementation is concerned, over other known linear (e.g. predictive control, fuzzy/neural control etc.) or nonlinear (backstepping, feedback linearization etc.) control methods that result being either too complex to compute or ad-hoc.

[4]Other important alternatives are LPV, velocity or neural/fuzzy-based scheduling control.

Concerning implementation, gain scheduling remains always the simplest possible adaptive control strategy, since in most cases, it only needs a limited memory space in order to stock the linear controllers designed off-line and some relatively small computational power so as to interpolate them and thus calculate the final control signal.

The linearization-based gain scheduling control, as compared mainly to LPV control, presents also a number of advantages: it is less conservative since a solution to the controller design-interpolation problem may be (almost always) found for each operating point separately and not as a whole as in the LPV case and it is also more intuitive since the designer's experience with the system in various operating points may be better exploited; thus the controller tuning may be also performed more easily using classical control theory concepts (e.g. $\mathscr{H}_\infty$ loop shaping control).

Except for all these advantages, gain scheduling control (and as a result its linearization-based version explored in this work) presents still a number of inconveniences despite the intense research effort on the domain, mainly during the last twenty years. Some of these inconveniences (being treated in this thesis) are the following and are closely connected to each other:

Synthesis point selection: The number of equilibrium-operating points considered for the linear controller synthesis is very important since too many points may result in significant memory storing space demands, whereas too few may result in lack of performance/robustness since at an intermediate operating point, the interpolated controller may be too far from one designed specifically for this point. The current bibliography really lacks of systematic methods, that can for example capture the system's nonlinear dynamics variation as a function of the changing operating conditions, in order to choose an appropriate number of synthesis points.

Linear controller order/structure: The linear controllers' order and/or structure is another significant issue of this method since a higher order controller may be of high performance but may also need more space to be stored and more computational power to be interpolated than a lower order one (e.g. say a PID one). As a result, sometimes it would be preferable to design lower-order controllers at carefully chosen operating points, than for example a single full-order controller at the center or the corners only of the flight envelope.

Interpolation method: The interpolation method chosen plays a vital role in gain-scheduled control schemes since many problems may be caused by a not-carefully selected interpolation strategy. Some strategies may for example result in control signal discontinuities or transients during scheduling region transitions or in high computational power demands from the on-board plant computer.

In the context of this thesis these inconveniences have been treated in an original manner using interpolated static, output feedback $\mathscr{H}_\infty$ controllers in addition to interpolated fixed order PID control structures and an operating point selection algorithm, being the result of the connection between $\mathscr{H}_\infty$ loop shaping control theory and a robust control theory tool called the gap metric. The strategy proposed is an extension to the standard McFarlane&Glover loop shaping design procedure (LSDP) and has thus been named the *e*-LSPD[5].

This work has been the collaboration of three parties: the automatic control department of SUPELEC at Gif-sur-Yvette, the Astrium Space Transportation (EADS) and MBDA research teams and finally the LSS/CNRS (Laboratoire des Signaux et Systèmes/Centre National de la Recherche Scientifique) nonlinear systems group. The collaboration of SUPELEC's robust control team with the EADS and MBDA research teams has a long tradition; this thesis continues exactly this tradition. It starts in the early 90's with the development of $\mathscr{H}_\infty$ design methods to control missiles around fixed operating points (1993-1996, see also [44]) and then to extend these approaches to a wider domain using an analytical model (1996-1999, see also [61]). Following these results, works have been performed to design fixed-order controllers on a wide domain while taking into account multi-rate sampling (1999-2002, see also [41]). Meanwhile a similar collaboration including also LSS teams was conducted to deal with robustness issues in a nonlinear context. It began by assessing the incremental norm used for robustness issues, which leads to results about stability and robustness of scheduled controllers (1991-1995, see also [46]), and by studying how to take into account robustness requirements when using nonlinear control design based on feedback linearization-like techniques (1996-1999, see also [32]). After these studies, the next step was an assessment of other techniques for robustness analysis such as methods based on IQC's (2000-2003, see also [2]).

The thesis started in 2005 with the scope of exploring/developing tools in order to facilitate/systematize some issues of the gain scheduling practice, such as the ones in the previous page. The first six months were devoted to an extensive bibliographic research on existing gain scheduling control strategies in order to unveil their major inconveniences (see Chapters 1, 2). After this phase, three promising techniques were chosen (controller blending, observer/state feedback and gain interpolation) and during the following twelve months the first two were explored on a realistic benchmark example (R'm'B). The results were not judged totally sufficient (see Chapter 5) and thus the novel *e*-LSDP was developed using the third proposed technique (gain interpolation) and tested also on the same benchmark example (see Chapter 6), during the six months to follow. The same technique was validated on a second benchmark example (ARV) during the first six months of the thesis' last year (see also Chapter 6). Meanwhile, some additional projects in the same context, performed by final year SUPELEC undergraduate students were supervised (see Publication & Project list).

[5]The letter 'e' stands for 'extended'.

The thesis report is divided into two main parts with three chapters each:

The *first part* presents the extensive bibliographical survey on the subject of gain scheduling with some additional references on system modeling and $\mathscr{H}_\infty$ control tools, thus giving the theoretical background needed for the work that follows in the second part.

The *second part* details the control strategies proposed in this work and the corresponding application examples tested; the first one being the famous Reichert missile benchmark model (or *R'm'B*) and the second an atmosphere re-entry vehicle model (or *ARV*) provided by Astrium Space Transportation .

The *first chapter* of this report starts with an introduction to adaptive control schemes such as direct & indirect adaptive control and gain scheduling. The first part of the chapter deals with system modeling issues such as equilibrium and linearization notions, whereas the second part presents an extensive review of *linearization-based gain scheduling (LBGS)* (or divide-and-conquer) and relative issues such as interpolation methods, stability-preserving approaches etc.

The *second chapter* presents a brief overview of alternative to the LBGS method gain scheduling strategies such as LPV control, velocity-based, neural and fuzzy-based implementations. Given that these methods were not being considered in this work, this chapter is disconnected from the rest and is primarily given for the sake of completeness.

The *third chapter* completes the theoretical background and details some control tools used in the context of linearization-based gain scheduling such as $\mathscr{H}_\infty$ control in LMI regions, estimator-controller forms of compensators and $\mathscr{H}_\infty$ loop shaping control. The latter is especially interesting and is presented in more detail, along with some elements of the gap metric theory and the McFarlane&Glover loop shaping design procedure (LSDP), since they play a key role in the control strategies proposed in this work - mainly in Chapter 6.

The *fourth chapter* details the first two steps of the linearization-based gain scheduling method, namely the *trim analysis* and *plant linearization* steps, used in this work for both benchmark systems. The mathematical models with all necessary constants and specifications are also given.

The *fifth chapter* presents the gain-scheduling control strategies, namely the *controller blending* and *observer/state feedback interpolation* methods, tested on the Reichert missile benchmark model during the first year of the thesis. Their advantages (but mostly their disadvantages) are stressed out and the discussion leads to the methods proposed in the final chapter.

The *sixth chapter* introduces a novel gain-scheduling procedure proposed in this work: the *e-LSDP*, correspond to the last three steps of the linearization-based gain scheduling method, namely the *controller synthesis*, *controller interpolation* and *controller implementation & validation* steps. The proposed method is detailed and validated on both benchmarks examples considered in a parallel manner in the two parts of this chapter.

A block diagram of the thesis structure is illustrated in Fig. 1; the shaded boxes corresponding to the application/validation chapters, whereas the unshaded to the theoretical/bibliographic ones[6]. The material presented in Chapter 1, being the most used chapter of this work, is extensively used in Chapters 4, 5, 6 whereas Chapters 5, 6 use all the material of Chapters 3 and 4. Finally Chapter 2 is somewhat independent and comes as a natural follow-up of Chapter 1.

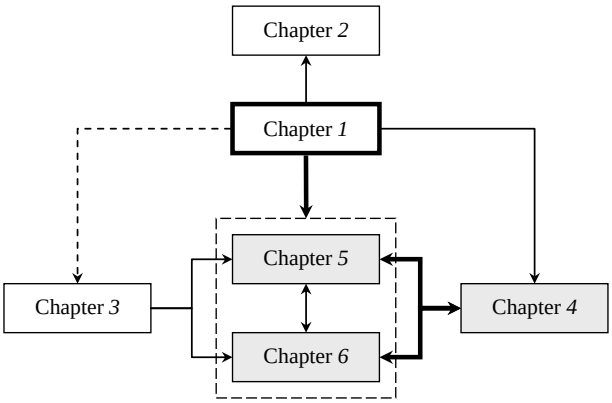


Figure 1: Thesis structure.

[6]The arrows show the connections between chapters; dashed lines show loose connection whereas thick lines stronger ones.

Part I

Theoretical background.

Chapter 1

Classic Gain Scheduling

Overview

Gain scheduling has been one of the dominant control strategies for the design of high performance controllers in the industry for the last fifty years at least. The gain scheduling practice can be roughly divided into two basic categories: *linearization-based* (or classic) and *LPV* (or modern) gain scheduling. In this chapter the first category is detailed whereas in Chapter 2 some classic results on the second category are mentioned. The *classic* method can be also divided into two subcategories: approaches that do offer some stability guaranties for the gain-scheduled system and others that do not. This chapter begins with a general introduction to the adaptive control framework (encompassing also gain scheduling) and then proceeds to a detailed citation of all classic gain scheduling methods existing in the bibliography. At the end of the chapter some additional tools used in the gain scheduling context are also detailed.

Chapter contents

1.1 Adaptive Control Schemes

Introduction

Adaptive control has risen due to the need of changing/updating afeedback controller K in order to conform to the changing parameters of a process $\mathcal{S}$. As a simple example consider the dynamics of an aircraft: this type of systems operate in different altitudes and with different speeds and thus due to several physical reasons their dynamics change drastically as a function of time. A *robust* controller designed to cope with the different operating conditions cannot always guarantee, or at least offer some good indications, that the aircraft will behave in a good way for all altitudes and speeds of its flight envelope.

To solve this problem an adaptive control system may be used in order to update the controller parameters for changing operating conditions. Three basic types of adaptive control systems exist (see Figure 1.1):

Indirect Adaptive Control

Indirect Adaptive Control (IAC): In this control scheme (see Fig. 1.1a) the controller parameters (or gains) ϑ_c are updated in real-time by an auto-tuner. This auto-tuner is based on an identified process model $\hat{\mathcal{S}}$ provided by an estimator that uses I/O plant information. This auto-tuner then calculates ϑ_c as if $\hat{\mathcal{S}} = \mathcal{S}$. The control scheme has two feedback loops: an internal loop that is fast enough to control the plant and an external one that is slower and detects any potential changes in the system's model through an estimator. An example of an IAC scheme is adaptive pole placement control: the poles of the closed loop plant are assigned in real-time to a specified location on the complex plane based on the estimate of $\mathcal{S}$ and on a given controller structure (e.g. PID).

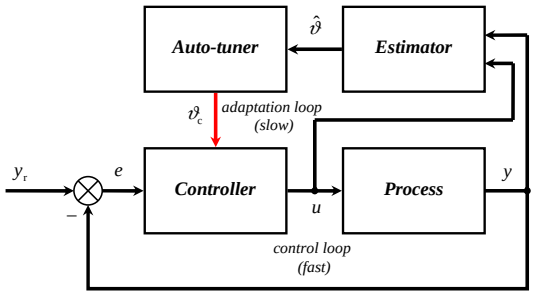
Direct Adaptive Control

Direct Adaptive Control (DAC): In this control scheme the controller parameters ϑ_c are estimated directly and the use of a plant parameter estimator is not needed. Take for example a frequently used topology of DAC: the direct Model Reference Adaptive Control (d-MRAC) configuration of Fig. 1.1b. The auto-tuner here computes the difference e_d between the outputs of the real plant $\mathcal{S}$ and of a *target* plant model $\bar{\mathcal{S}}$ and tries to find a value for ϑ_c so that this difference goes to zero. A way to do that is the famous *MIT* rule [15, 66].

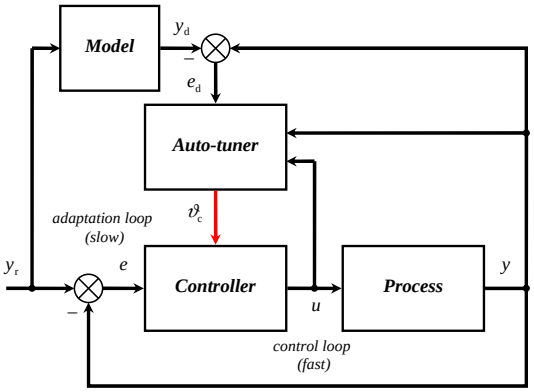
Gain Scheduling Control

Gain Scheduling Control (GSC): In this control scheme (see Fig. 1.1c) no complex algorithm is demanded for updating the controller parameters but only a parameter (or scheduling) vector ϱ (that can sufficiently capture the plant's change of dynamics) and an interpolation method. The controller parameters ϑ_c are then updated by combining/interpolating different controllers K^i designed for the plant $\mathcal{S}$, for some family of *critical* values of ϱ[1]. The simplest form of GSC is *controller switching* where no smooth controller parameter update is performed and a single controller is used, being valid for a pre-defined operating region over ϱ.

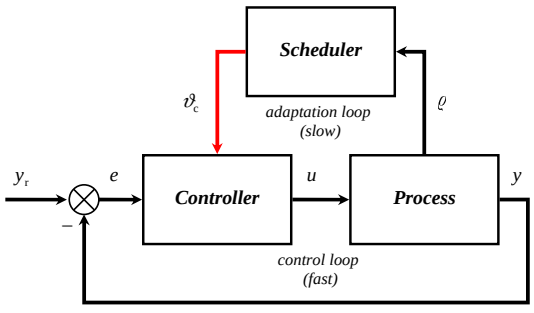
[1] For the aircraft example considered before, this vector ϱ could be the Mach and the altitude.



(a) *Indirect adaptive control configuration*



(b) *Direct adaptive control configuration*



(c) *Gain scheduling control configuration*

Figure 1.1: Various adaptive control schemes.

In this monograph the latter method will be considered for the control of generic nonlinear parameter/time dependent systems. The gain scheduling control practice can be further divided into two major categories: the *linearization-based* and *LPV/q-LPV* gain scheduling procedures. The major distinction between these two has to do on the one hand with the approach taken in order to obtain the final nonlinear gain-scheduled controller, and on the other hand on the way that the system nonlinearity is treated. These two methods sometimes overlap and there exist a considerable disagreement over the scientific community on which one is the best suited for a particular problem. LPV vs. LBGS

The linearization-based gain scheduling procedure (LBGS)[2] is mostly based on linearized plants of the initial nonlinear system, calculates a number of controllers of possibly, not the same structure, and finally interpolates them in order to obtain the gain-scheduled controller. The existence of a controller is (almost) always guaranteed but stability issues arise due to the ad-hoc linearization-interpolation. There exist however some notable exceptions that they do consider stability for the linearized scheduled system, but obviously not on the initial nonlinear one. In this chapter both types of methods will be discussed and some key results as well as references to real world applications will be given.

The LPV procedure tries to camouflage the nonlinear dynamics and obtain thus a linear system with time varying state space matrices. These time varying matrices can be treated either as time varying uncertainty thus leading to the so-called *LFT* formulation, or as parameter-dependent matrices that may form convex hyper-cubes for frozen values of the parameter leading to the *Polytopic* formulation. In both cases there exist stability guarantees for the overall scheduled system. However, the fact that is not clear enough (see [88], pp. 1012) is for which system the stability guarantees are offered[3].

Briefly it can be said that the first class of methods offers a systematic and unconservative design methodology that provides always a controller whereas the second offers a more theoretically sound, yet sometimes conservative in terms of system operation & controller existence, procedure that guarantees global stability of the gain-scheduled plant. In this work the first class of methods will be used and several of its problems addressed.

In this chapter the first class of methods (*classic*) is extensively detailed whereas in the next one the second class ones (*modern*) are briefly reviewed. The following section (Section 1.2) considers some general results in system modeling whereas the next one (Section 1.3.1) details the LBGS following the famous five (2+3) step procedure (see [88]). Contents

Finally, subsections 1.3.2, 1.3.3 consider both the ad-hoc and stability preserving methods existing in the bibliography whereas Section 1.4 presents some related to the gain scheduling practice results concerning interpolation and operating domain triangulation.

[2] Also called classic or divide and conquer method (see [88], pp. 1005-1008).

[3] For more details see Chapter 2 being also based on the analysis of the next section.

1.2 System Modeling

In this section general modeling issues in the context of gain scheduling are reviewed. Some system equilibrium notions are initially introduced before passing to a citation of various ways to model a physical process. Finally, some material on Jacobian linearization is covered.

1.2.1 Equilibrium Notions

`System modeling`

Consider a generic non-autonomous[4] forced nonlinear dynamic system $\mathcal{S}$ whose state and output dynamics are described by a number of coupled first-order differential equations (see Fig. 1.2):

$$\mathcal{S}: \quad \begin{aligned} \dot{x}(t) &= \mathbf{f}[x(t), u(t), t] \\ y(t) &= \mathbf{h}[x(t), u(t), t]. \end{aligned} \tag{1.1}$$

The vectors x, u, y represent the *states*, *inputs* and *outputs* of the system with $x \in \mathbb{R}^n, u \in \mathbb{R}^{n_u}, y \in \mathbb{R}^{n_y}$ respectively[5]. The vector-valued functions $\mathbf{f}, \mathbf{h}$ where $\mathbf{f} := \big[f_1(x,u), \ldots, f_n(x,u)\big]^T$ and $\mathbf{h} := \big[h_1(x,u), \ldots, h_{n_y}(x,u)\big]^T$ perform the following nonlinear mappings:

$$\mathbf{f}: \quad \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R} \mapsto \mathbb{R}^n \tag{1.2}$$

$$\mathbf{h}: \quad \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R} \mapsto \mathbb{R}^{n_y}. \tag{1.3}$$

The nonlinear system $\mathcal{S}$ in fact is a mathematical representation of a physical process and thus for $\mathcal{S}$ to provide a valid reproduction of its behavior, several additional hypotheses need to be made. These hypotheses are mostly related to the existence and uniqueness of a solution $x\big(t; t_0, x(t_0)\big)$ given a set of initial state conditions $x(t_0)$ and the differentiability of the functions $\mathbf{f}, \mathbf{h}$ with respect to an equilibrium point or trajectory (see [75], Ch. 3 for more details).

The analysis concerning equilibrium notions in the next two subsections considers both autonomous nonlinear and linear systems and their non-autonomous extensions. Another extension is also given for parameter-varying systems used mostly to model processes controlled by gain-scheduled control schemes.

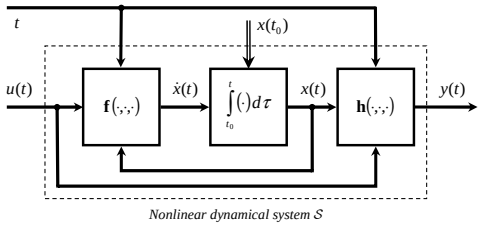


Figure 1.2: System block diagram.

[4] *Time-invariant / autonomous* are equivalent as are *time-varying / non-autonomous*.

[5] Dependence on t will be dropped when needed and a derivative is taken with respect to t.

1.2.1.1 Autonomous Systems

To introduce the notion of an equilibrium point it would be simpler to consider first an *unforced* nonlinear autonomous system $\mathcal{S}_a$ whose state dynamics are described by: Unforced case

$$\mathcal{S}_a : \quad \dot{x} = \mathbf{f}(x). \tag{1.4}$$

An equilibrium state x_{eq} for this system $\mathcal{S}_a$ is defined as the point in the state space from which when the state starts it never leaves, for every $t \geq t_0, t_0 > 0$. The condition that defines such a state is written as:

$$x_{\text{eq}} : \quad \left.\frac{dx(t)}{dt}\right|_{\text{eq}} = \dot{x}_{\text{eq}} \triangleq 0. \tag{1.5}$$

The above condition means briefly that $x\big(t; t_0, x(t_0)\big) = x(t_0) = x_{\text{eq}}$ and as a consequence from Eqs. 1.4, 1.5: Equilibrium condition

$$\mathbf{f}(x_{\text{eq}}) \triangleq 0. \tag{1.6}$$

Now in order to find the equilibrium points of such a system, a collection of n coupled algebraic equations (given by Eq. 1.6) should be solved. These equations may yield a finite or even an infinite number of equilibrium points, depending on their structure. When studying the stability of the system $\mathcal{S}_a$, using for example the Lyapunov's stability theory, it may be useful to study the stability of the state vector at the origin. This can be done by translating all its equilibrium points via a change of variables; indeed, suppose a $x_{\text{eq}} \neq 0$ and the change of variables $z = x - x_{\text{eq}}$. Then:

$$\begin{aligned} \dot{z} &= \dot{x} - \dot{x}_{\text{eq}} \\ &= \mathbf{f}(x) - 0 \\ &= \mathbf{f}(z + x_{\text{eq}}) \triangleq \mathbf{g}(z). \end{aligned} \tag{1.7}$$

The function $\mathbf{g}(\cdot)$ does not depend *explicitly* on time and thus the unforced nonlinear equivalent system $\dot{z} = \mathbf{g}(z)$ is also autonomous with an equilibrium point at the origin $z = 0$.

Consider now the case where the autonomous nonlinear system is *forced*, i.e. it's dynamics are described by: Forced case

$$\mathcal{S}_{a,f} : \quad \dot{x} = \mathbf{f}(x, u). \tag{1.8}$$

Then non-zero equilibrium states x_{eq} could be now *imposed* by using a constant corresponding equilibrium control input vector $u_{\text{eq}} = \mathbf{u}(x_{\text{eq}})$, that will maintain the state to its equilibrium value for all $t \geq t_0, t > 0$. The system's *equilibrium manifold* $\mathcal{E}_{a,f}$ is defined as the set of all the admissible states/inputs for which the right-hand side of Eq. 1.8 may go to zero: Equilibrium manifold

$$\mathcal{E}_{a,f} : \quad \big\{(x_{\text{eq}}, u_{\text{eq}}) | \mathbf{f}(x_{\text{eq}}, u_{\text{eq}}) = 0\big\}. \tag{1.9}$$

Output condition

To the equilibrium manifold $\mathcal{E}_{a,f}$, corresponds also an equilibrium value y_{eq} for the output of the nonlinear autonomous system[6]:

$$y_{\text{eq}} = \mathbf{h}(x_{\text{eq}}, u_{\text{eq}}). \tag{1.10}$$

Once again, the equilibrium points of this *forced* system $\mathcal{S}_{a,f}$ may be translated to the origin for an *unforced* equivalent system. To illustrate this, consider the change of variables $z = x - x_{\text{eq}}, v = u - u_{\text{eq}}$. Then:

$$\begin{aligned} \dot{z} &= \dot{x} - \dot{x}_{\text{eq}} \\ &= \mathbf{f}(x, u) - 0 \\ &= \mathbf{f}(z + x_{\text{eq}}, v + u_{\text{eq}}) \triangleq \mathbf{g}(z, v). \end{aligned} \tag{1.11}$$

The right hand side of Eq. 1.10 does not depend explicitly on time and thus the system described by the transformed equation $\mathbf{g}(z, v)$ admits an equilibrium point at its origin since by definition $\mathbf{f}(x_{\text{eq}}, u_{\text{eq}}) = 0$. So because of the fact that in this case $v = 0$, the transformed plant is now *unforced* and once again the analysis ends up to the study of a system like the one in Eq. 1.7 around $z = 0$.

LTI case

In the case of a *linear time-invariant* (LTI) dynamical system having the following state space representation (with $\mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{B} \in \mathbb{R}^{n \times n_u}, \mathbf{C} \in \mathbb{R}^{n_y \times n}, \mathbf{D} \in \mathbb{R}^{n_y \times n_u}$):

$$\mathcal{S}_{\text{LTI}} : \quad \begin{aligned} \dot{x} &= \mathbf{A}x + \mathbf{B}u \\ y &= \mathbf{C}x + \mathbf{D}u \end{aligned} \tag{1.12}$$

the things are simple; the origin is *always* an equilibrium state for the unforced system whereas for the *forced* one, under certain controllability conditions, one may be able to maintain the state to a given equilibrium value x_{eq} using a suitable equilibrium (or *open loop*) control u_{eq} that satisfies:

$$\mathbf{A}x_{\text{eq}} + \mathbf{B}u_{\text{eq}} = 0 \tag{1.13}$$

with a corresponding equilibrium output:

$$y_{\text{eq}} = \mathbf{C}x_{\text{eq}} + \mathbf{D}u_{\text{eq}}. \tag{1.14}$$

Summarizing, the equilibrium points of any linear or nonlinear, forced or unforced autonomous system may be translated to the origin with the resulting system being also autonomous. However, as it will be shown in the next section, the resulting equivalent system $\mathbf{g}(\cdot)$ for an equilibrium *trajectory* $x_{\text{eq}}(t)$ that is a solution to the initial *autonomous* nonlinear system $\mathcal{S}_a$ and satisfying:

$$\dot{x}_{\text{eq}}(t) = \mathbf{f}[x_{\text{eq}}(t)] \tag{1.15}$$

will be *non-autonomous* even when the initial system is.

[6] For the autonomous *unforced* system the output equilibrium value is given by $y_{\text{eq}} = \mathbf{h}(x_{\text{eq}})$.

1.2.1.2 Non-autonomous Systems

The state dynamics of the *unforced* nonlinear non-autonomous system $\mathcal{S}_{na}$ are: Unforced case

$$\mathcal{S}_{na}: \quad \dot{x} = \mathbf{f}(x,t). \tag{1.16}$$

This system has an equilibrium point at $t = 0$, if $\mathbf{f}(0,t) = 0, \forall t \geq 0$ (any equilibrium point at $t = 0$ is also an equilibrium point at all times; see for example [142], pp. 5). In addition, any constant non-zero equilibrium point can be translated to the origin (for $t = 0$) with the same procedure as in Section 1.2.1.1, both for the forced and unforced cases. In fact this may be done not only for a constant equilibrium point but also on a nonzero system equilibrium trajectory $x_{\text{eq}}(t)$. Indeed, consider the change of variables $z = z(t) = x - x_{\text{eq}}(t)$; then the time derivative of z for $\mathcal{S}_{na}$ will be:

$$\begin{aligned} \dot{z} &= \dot{x} - \dot{x}_{\text{eq}}(t) \\ &= \mathbf{f}(x,t) - \dot{x}_{\text{eq}}(t) \\ &= \mathbf{f}[z + x_{\text{eq}}(t), t] - \dot{x}_{\text{eq}}(t) \triangleq \mathbf{g}(z,t). \end{aligned} \tag{1.17}$$

Thus the equivalent system described by $\mathbf{g}(\cdot)$ has an equilibrium at the origin for $t = 0$ like the one in Eq. 1.16. From the above analysis it is also evident that even if the initial state dynamics are autonomous, the equivalent system $\mathbf{g}(\cdot)$ is non-autonomous in the case of an equilibrium trajectory since the transformed variable z depends also *explicitly* on time due to $x_{\text{eq}}(t)$.

The equilibrium analysis for the *forced* non-autonomous system $\mathcal{S}$ of Eq. 1.1 is related to its control as in the analysis of the previous section concerning this case (see Eq. 1.11). The control to maintain the system $\mathcal{S}$ to a *given* equilibrium state x_{eq} is time-varying and is composed by a steady-state value u_{eq} (translating the state to its steady-state value) plus a time-varying one u_δ that regulates the time-varying system around the origin (see [75], pp. 469-471). The analysis for the linear time-varying (LTV) version (see Eq. 1.18) is also similar. Forced case

$$\mathcal{S}_{\text{LTV}}: \quad \begin{aligned} \dot{x} &= \mathbf{A}(t)x + \mathbf{B}(t)u \\ y &= \mathbf{C}(t)x + \mathbf{D}(t)u \end{aligned} \tag{1.18}$$

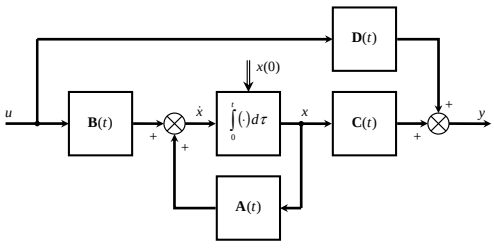


Figure 1.3: LTV system block diagram.

1.2.1.3 Parameter-dependent Systems

A certain class of systems is characterized by a dependence of their dynamics on a time-varying vector of parameters $\varrho = \varrho(t)$. This parameter vector is often called the scheduling vector and it is assumed that *it can be measured in real time* for gain-scheduled systems.

NLPD system

These systems are also called *nonlinear parameter-dependent systems* (NLPD) and their dynamics are given by:

$$\mathcal{S}_{\text{pd}}: \quad \begin{aligned} \dot{x} &= \mathbf{f}(x, u, \varrho) \\ y &= \mathbf{h}(x, u, \varrho). \end{aligned} \tag{1.19}$$

Equilibrium manifold

Leaving other modeling details for the next section, such a system representation has a meaning if an equilibrium manifold $\mathcal{E}_{\text{pd}}$ is defined and *smoothly* parameterized as a function of the scheduling vector ϱ:

$$\mathcal{E}_{\text{pd}}: \quad \Big\{ \big[x(\varrho_{\text{eq}}), u(\varrho_{\text{eq}})\big] \,\big|\, \mathbf{f}\big[x(\varrho_{\text{eq}}), u(\varrho_{\text{eq}}), \varrho_{\text{eq}}\big] = 0 \Big\}. \tag{1.20}$$

To illustrate this consider for example the trivial case of a first order SISO system with a single time-varying parameter. The equilibrium point locus may look like the one visualized in Fig. 1.4; an equilibrium state x_{eq} and a corresponding equilibrium input u_{eq} are assigned for any admissible value of the scheduling parameter ϱ.

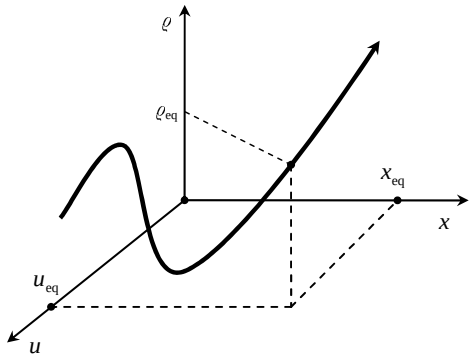


Figure 1.4: Equilibrium point locus.

Output manifold

For each value of the scheduling vector ϱ_{eq} the corresponding equilibrium output may also be defined as:

$$y_{\text{eq}} = \mathbf{h}\big[x(\varrho_{\text{eq}}), u(\varrho_{\text{eq}}), \varrho_{\text{eq}}\big]. \tag{1.21}$$

Technical Note. The computation of the equilibrium manifold of a parameter-dependent nonlinear system is by no means a trivial problem and is mainly done either by solving directly the algebraic equation $\mathbf{f}(x, u, \varrho) = 0$ for every admissible ϱ (if possible), or by numerical iterative optimization techniques (see for example the function '`findop`' of the MATLAB® Simulink Control Design toolbox).

1.2.2 System Descriptions

In this section the discussion will involve around various types of systems involved in gain-scheduled control schemes. The discussion necessitates the analysis of Section 1.2.1 on equilibrium points since for the scope of gain-scheduled control a process is often studied around such points. Details on linearization are given in the next subsection.

NLPD system

A physical process is usually modeled using a collection of nonlinear first order differential equations representing its *state dynamics* along with a second set of nonlinear algebraic equations describing its *output dynamics*. These modeling equations are often dependent to a number of external or internal variables that are regarded as parameters of the process. This parameter vector is called the *scheduling vector* $\varrho = \varrho(t)$ and gives a time-varying sense to such systems. This modeling results to the following equations for this *nonlinear parameter-dependent* system (or NLPD):[7]

$$\mathcal{S}_{\text{pd}}: \quad \begin{aligned} \dot{x} &= \mathbf{f}(x, u, \varrho) \\ y &= \mathbf{h}(x, u, \varrho). \end{aligned} \tag{1.22}$$

Besides the usual properties for the state $x(t)$, input $u(t)$, output $y(t)$ and for the functions $\mathbf{f}, \mathbf{h}$ discussed in the beginning of Section 1.2.1, additional hypotheses are made for the scheduling vector ϱ. Specifically it is assumed that $\varrho \in \mathbf{\Gamma}$, where $\mathbf{\Gamma}$ is a connected compact set with $\mathbf{\Gamma} \subset \mathbb{R}^{n_\varrho}$.

Operating domain discussion

Remark. A set is said to be *connected* if it is impossible to express it as the union of two or more disjoint open subsets. For example the set $[0, 1]$ is connected whereas the union of the sets $[0, 0.5), (0.5, 1]$ is disconnected. In fact any convex set is connected. Moreover a set is said to be *compact* if it is closed and bounded. For example the set $[0, 1]$ is closed but the sets $(0, 1), (0, 1]$ are not closed. In addition all these sets are bounded since they have finite sizes (see [19] for more details on set theory).
The above assumptions on ϱ are quite logical since in physical systems most variables take values on closed, finite and sometimes convex intervals. For example the Mach number of a missile takes values between a minimum and a maximum value; the same holds for the control inputs of a system which are bounded or a varying resistor in an electrical network.

Alternative formulation

An extended method to model a parameter-dependent system used mainly for output tracking adopts a linear robust control-type notation (see [79, 80, 128, 129, 131, 133] or even [75], pp. 474-475):

$$\mathcal{S}^*_{\text{pd}}: \quad \begin{aligned} \dot{x} &= \mathbf{f}(x, u, w) \\ \zeta &= \mathbf{h}_\zeta(x, u, w) \\ y &= \mathbf{h}_y(x, u, w). \end{aligned} \tag{1.23}$$

[7] Note that the explicit dependence on time is omitted but is assumed because $\varrho = \varrho(t)$. Many authors prefer to omit also the explicit dependence on the scheduling vector.

In this notation the signals $w(t)$ are external perturbations whereas $\zeta(t)$ are errors to be minimized (or signals to be treated) and $\mathbf{h}_\zeta, \mathbf{h}_y$ are the corresponding nonlinear algebraic functions with suitable dimensions. The scheduling vector ϱ may or may not appear directly[8] but it is always assumed to exist and defines a smooth equilibrium manifold as detailed in Section 1.2.1.3.

q-LPV system

Return now to the initial parameter-dependent system of Eq. 1.22. Performing state space transformations it is sometimes possible to transform the state/output dynamics so that an equivalent *quasi-linear* parameter-varying system $S_{q-\mathrm{LPV}}$ may appear (with σ being now the measured parameter vector):

$$\mathcal{S}_{\mathrm{q-LPV}}: \quad \begin{aligned} \dot{x} &= \mathbf{A}(x,u,\sigma)x + \mathbf{B}(x,u,\sigma)u \\ y &= \mathbf{C}(x,u,\sigma)x + \mathbf{D}(x,u,\sigma)u \end{aligned} \tag{1.24}$$

A similar modeling in q-LPV form is when the state x is divided into two parts: the part that is regarded as a parameter x_ϱ and the part that keeps its state variable notion $x^\star$. Then the final scheduling variable ϱ is consisted of the parameter-varying variable σ and of $x^\star$ (see [114], §3.2, pp. 1407).

LPV system

The (nonlinear) dependence of the system matrices on the state, input & measured parameter may also be regarded as a general time-varying parameter vector $\varrho = \varrho(x,u,\sigma)$. The trajectories of the scheduling vector ϱ are considered to be measured in real time. In this case a *linear* parameter-varying system is obtained:

$$\mathcal{S}_{\mathrm{LPV}}: \quad \begin{aligned} \dot{x} &= \mathbf{A}(\varrho)x + \mathbf{B}(\varrho)u \\ y &= \mathbf{C}(\varrho)x + \mathbf{D}(\varrho)u. \end{aligned} \tag{1.25}$$

A delicate issue arises here however: if the scheduling vector is considered to be a function of the state also, then the gain-scheduled controller is assumed to be a state feedback one [9]. As a result, it is preferable to use the *output* rather than the *state* to parameterize the system. Hence, the scheduling vector and the LPV dynamics are dependent directly on the output, the parameter vector σ, and possibly on the input (see [114], §3.2, pp. 1408-09, [120]).

Discussion on LPV systems

The solutions of the nonlinear system (and hence of the q-LPV one) are also solutions of the LPV formulation in Eq. 1.25 and thus the former is over-bounded by the latter (see [88], pp. 1012). This modeling adds some conservativeness but for gain scheduling control it may taken as a basis for controller design. Both classic/modern gain scheduling tools consider these types of models (i.e. see the fundamental work of [13, 18])[10] but for the former there exists a major difference: the LPV models used with classic (or linearization-based) gain scheduling are only valid close to equilibrium points and do not directly describe the behavior of the initial parameter-dependent nonlinear system $\mathcal{S}_{\mathrm{pd}}$ of Eq. 1.22.

[8] In some cases (see [79, 80]) the scheduling vector appears indirectly but is assumed to be a nonlinear function of the controller input and of measured external or signals. In other cases (see [128, 129, 131, 133]) it is considered as a parameter and does not appear directly.

[9] The same holds for the equilibrium manifold being also a function of the state.

[10] Modern gain scheduling design tools will be considered in the next chapter.

More precisely, such linearization-based LPV models are of the form[11]:

$$\mathcal{S}_{\text{LPV}}: \quad \begin{aligned} \dot{x}_\delta &= \mathbf{A}(\varrho)x_\delta + \mathbf{B}(\varrho)u_\delta \\ y_\delta &= \mathbf{C}(\varrho)x_\delta + \mathbf{D}(\varrho)u_\delta \end{aligned} \tag{1.26}$$

where:

$$x_\delta = x - x_{\text{eq}}(\varrho) \tag{1.27}$$
$$u_\delta = u - u_{\text{eq}}(\varrho) \tag{1.28}$$
$$y_\delta = y - y_{\text{eq}}(\varrho). \tag{1.29}$$

The distinction between the two LPV models of Eqs. 1.25, 1.26 is now clear: the first is a 'superset' of the initial nonlinear system with x, u, y being its actual variables whereas the second is a family of linear(ized) systems permitting only local description (around equilibrium points) of the nonlinear system at best.

A special case of the LPV modeling of Eq. 1.25 is the so-called LFT-based approach. This approach treats the LPV model as an LTI one with all time-varying parameters ϱ being regrouped as (measurable) uncertainties. This approach is also used in modern gain scheduling methods and it assumes that the plant $\mathcal{S}_{\text{LPV}}$ of Eq. 1.25 may be rewritten as the upper LFT (u-LFT) of an LTI standard plant $\mathbb{P}(s)$ (following a robust control-type modeling of the parameter-dependent plant) and a time-varying block operator Θ, specifying how the scheduling vector components ϱ_i enter the LPV plant dynamics: LFT case

$$\begin{bmatrix} \zeta \\ y \end{bmatrix} = \mathcal{F}_u\big(\mathbb{P}(s), \Theta\big) \begin{bmatrix} w \\ u \end{bmatrix} \tag{1.30}$$

with variables ζ, w being error/external perturbations signals respectively and:

$$\Theta = \text{blockdiag}(\varrho_1 \mathbb{I}_{\varrho_1}, \ldots, \varrho_1 \mathbb{I}_{\varrho_{n_\varrho}}) \tag{1.31}$$

and also:

$$\zeta_\theta = \Theta w_\theta \tag{1.32}$$

representing the I/O's of the uncertainty block (see Fig. 1.3).

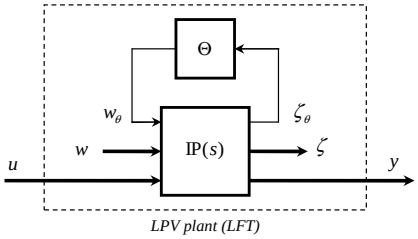


Figure 1.5: LFT description of an LPV system.

[11] More details on this formulation and linearization are given are given in the next section.

LTV systems

Another important class of systems are the linear time-varying (LTV) systems of the form (see Fig. 1.3):

$$\mathcal{S}_{\text{LTV}}: \quad \begin{aligned} \dot{x} &= \mathbf{A}(t)x + \mathbf{B}(t)u \\ y &= \mathbf{C}(t)x + \mathbf{D}(t)u \end{aligned} \tag{1.33}$$

These types of systems represent either a physical process directly, or (more often) they stem from the linearization of a nonlinear system (or also parameter-dependent one) around an *equilibrium trajectory* $x_{\text{eq}}(t), u_{\text{eq}}(t), y_{\text{eq}}(t)$ (for more details see the next section). In this case the LTV model is rather written as:

$$\mathcal{S}_{\text{LTV}}: \quad \begin{aligned} \dot{x}_\delta &= \mathbf{A}(t)x_\delta + \mathbf{B}(t)u_\delta \\ y_\delta &= \mathbf{C}(t)x_\delta + \mathbf{D}(t)u_\delta \end{aligned} \tag{1.34}$$

with:

$$x_\delta = x - x_{\text{eq}}(t) \tag{1.35}$$
$$u_\delta = u - u_{\text{eq}}(t) \tag{1.36}$$
$$y_\delta = y - y_{\text{eq}}(t). \tag{1.37}$$

LTI system

The previous model is valid in the vicinity of the equilibrium trajectory as is the following LTI system a valid approximation model of a nonlinear parameter-dependent system around an *equilibrium point*.

$$\mathcal{S}_{\text{LTI}}: \quad \begin{aligned} \dot{x}_\delta &= \mathbf{A}x_\delta + \mathbf{B}u_\delta \\ y_\delta &= \mathbf{C}x_\delta + \mathbf{D}u_\delta \end{aligned} \tag{1.38}$$

Discussion

The following figure shows an illustration of all the different ways to model a physical process. Starting from a nonlinear parameter-dependent (NLPD) model and going inwards to lesser degrees of complexity, one gets the simplest possible model which is a linear time invariant (LTI) one. An LPV model can be seen either as a conservative way to approximate a q-LPV or NLPD model or a more complex way to model a possibly time-varying LTI system. The last holds also for the LTV one that can be either seen as a linear approximation of a NLPD system around an equilibrium trajectory or a more 'realistic' way to model an LTI system. Finally, an LTI system is an approximation of a NLPD model around an equilibrium point.

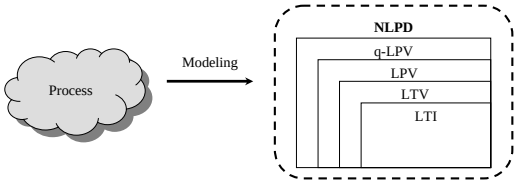


Figure 1.6: Process modeling.

1.2.3 Linearization Notions

In this section some results are presented concerning the approximation of nonlinear parameter-dependent systems by linearizing their dynamics around equilibrium points or trajectories. This section is also linked to the analysis of the two previous ones and offers the necessary material for the next section concerning linearization-based gain scheduling.

NLPD system

Suppose a given forced nonlinear parameter-dependent system $\mathcal{S}_{\text{pd}}$ is described by the following first-order differential equations:

$$\mathcal{S}_{\text{pd}}: \quad \begin{aligned} \dot{x} &= \mathbf{f}(x, u, \varrho) \\ y &= \mathbf{h}(x, u, \varrho). \end{aligned} \tag{1.39}$$

The functions $\mathbf{f}, \mathbf{h}$ with $\mathbf{f} := \big[f_1, f_2, \ldots, f_n\big]^T$ and $\mathbf{h} := \big[h_1, h_2, \ldots, h_{n_y}\big]^T$ perform the following nonlinear mappings on the state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^{n_u}$, output $y \in \mathbb{R}^{n_y}$ and scheduling vector $\varrho \in \mathbb{R}^{n_\varrho}$:

$$\mathbf{f}: \quad \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\varrho} \mapsto \mathbb{R}^n \tag{1.40}$$

$$\mathbf{h}: \quad \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\varrho} \mapsto \mathbb{R}^{n_y}. \tag{1.41}$$

As it has been detailed in Section 1.2.1.3, the scheduling vector $\varrho(t)$ defines an equilibrium manifold $\mathcal{E}_{\text{pd}}$ (see Eq. 1.20). This means that it spans the equilibrium points of the system inside its domain of operation $\mathbf{\Gamma}$, with $\mathbf{\Gamma}$ being a connected compact set (see the corresponding remark in Section 1.2.2).

Reformulation

The trajectory $x\big(t; t_0, x(t_0)\big)$ of $\mathcal{S}_{\text{pd}}$ may be *approximated* via the solution $\tilde{x}\big(t; t_0, x(t_0)\big)$ of a linearized model $\mathcal{S}_{\text{LTI}}$ in the close vicinity of an equilibrium point, defined for a constant (or frozen) value of the scheduling vector ϱ_{eq}. To obtain this approximation, reformulate first the dynamics of Eq. 1.39 (see [85], §2.1, pp. 291) as:

$$\dot{x}_\delta = \mathbf{A}(\varrho_{\text{eq}})x_\delta + \mathbf{B}(\varrho_{\text{eq}})u_\delta + \varepsilon_{\mathbf{f}} \tag{1.42}$$

$$y_\delta = \mathbf{C}(\varrho_{\text{eq}})x_\delta + \mathbf{D}(\varrho_{\text{eq}})u_\delta + \varepsilon_{\mathbf{h}} \tag{1.43}$$

Deviation quantities

where the errors $x_\delta, u_\delta, y_\delta$ are defined as:

$$x_\delta = x - x(\varrho_{\text{eq}}) \tag{1.44}$$

$$u_\delta = u - u(\varrho_{\text{eq}}) \tag{1.45}$$

$$y_\delta = y - y(\varrho_{\text{eq}}). \tag{1.46}$$

The matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are obtained by linearization (or first-order Taylor expansion) of the functions $\mathbf{f}, \mathbf{h}$ around the equilibrium point ϱ_{eq}[12] and having assumed that they have the appropriate differentiability properties.

[12] Even though the dependence of the matrices is shown to be only on ϱ_{eq} (see Eqs. 1.42, 1.43), it is assumed that there exist also a dependence on $x_{\text{eq}} = x(\varrho_{\text{eq}}), u_{\text{eq}} = u(\varrho_{\text{eq}})$. However it is omitted for notational simplicity.

System matrix computation

These matrices are computed as:

$$\mathbf{A}(\varrho_{\text{eq}}) = \nabla_x \mathbf{f}\big|_{\varrho_{\text{eq}}} \tag{1.47}$$

$$\mathbf{B}(\varrho_{\text{eq}}) = \nabla_u \mathbf{f}\big|_{\varrho_{\text{eq}}} \tag{1.48}$$

$$\mathbf{C}(\varrho_{\text{eq}}) = \nabla_x \mathbf{h}\big|_{\varrho_{\text{eq}}} \tag{1.49}$$

$$\mathbf{D}(\varrho_{\text{eq}}) = \nabla_u \mathbf{h}\big|_{\varrho_{\text{eq}}} \tag{1.50}$$

where:

$$\nabla_x \mathbf{f} = \begin{bmatrix} \dfrac{\partial f_1}{x_1} & \cdots & \dfrac{\partial f_1}{x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_n}{x_1} & \cdots & \dfrac{\partial f_n}{x_n} \end{bmatrix} \tag{1.51}$$

$$\nabla_u \mathbf{f} = \begin{bmatrix} \dfrac{\partial f_1}{u_1} & \cdots & \dfrac{\partial f_1}{u_{n_u}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_n}{u_1} & \cdots & \dfrac{\partial f_n}{u_{n_u}} \end{bmatrix} \tag{1.52}$$

$$\nabla_x \mathbf{h} = \begin{bmatrix} \dfrac{\partial h_1}{x_1} & \cdots & \dfrac{\partial h_1}{x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial h_{n_y}}{x_1} & \cdots & \dfrac{\partial h_{n_y}}{x_n} \end{bmatrix} \tag{1.53}$$

$$\nabla_u \mathbf{h} = \begin{bmatrix} \dfrac{\partial h_1}{u_1} & \cdots & \dfrac{\partial h_1}{u_{n_u}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial h_{n_y}}{u_1} & \cdots & \dfrac{\partial h_{n_y}}{u_{n_u}}. \end{bmatrix} \tag{1.54}$$

H.O.T.

The quantities $\varepsilon_{\mathbf{f}}, \varepsilon_{\mathbf{h}}$ are in fact the higher order terms (H.O.T.) in the Taylor series expansion of Eqs. 1.42, 1.43 and may be written as:

$$\varepsilon_{\mathbf{f}} = \mathbf{f}(x, u, \varrho) - \mathbf{f}(x_{\text{eq}}, u_{\text{eq}}, \varrho_{\text{eq}}) - \mathbf{A}(\varrho_{\text{eq}})x_\delta - \mathbf{B}(\varrho_{\text{eq}})u_\delta \tag{1.55}$$

$$\varepsilon_{\mathbf{h}} = \mathbf{h}(x, u, \varrho) - \mathbf{h}(x_{\text{eq}}, u_{\text{eq}}, \varrho_{\text{eq}}) - \mathbf{C}(\varrho_{\text{eq}})x_\delta - \mathbf{D}(\varrho_{\text{eq}})u_\delta. \tag{1.56}$$

Linearized system

The dynamics of the initial nonlinear parameter-dependent system may now be *approximated* by the following LTI system by *truncating* the higher-order terms $\varepsilon_{\mathbf{f}}, \varepsilon_{\mathbf{h}}$:

$$\mathcal{S}_{\text{LTI}} : \quad \begin{aligned} \dot{\tilde{x}}_\delta &= \mathbf{A}(\varrho_{\text{eq}})\tilde{x}_\delta + \mathbf{B}(\varrho_{\text{eq}})u_\delta \\ y_\delta &= \mathbf{C}(\varrho_{\text{eq}})\tilde{x}_\delta + \mathbf{D}(\varrho_{\text{eq}})u_\delta. \end{aligned} \tag{1.57}$$

A solution $x\big(t; t_0, x(t_0)\big)$ to the nonlinear system $\mathcal{S}_{\text{pd}}$ with $x(t_0)$ being 'close enough' to x_{eq} may now be written as:

$$x\big(t; t_0, x(t_0)\big) \simeq x_{\text{eq}} + \tilde{x}_\delta(t; t_0, 0) \tag{1.58}$$

or $x_\delta \simeq \tilde{x}_\delta$, with $x_\delta = x - x_{\text{eq}}$[13]. The question that rises now *is to what extent the approximation $\tilde{x}_\delta$ remains close to x_δ*. The answer to this important question is that the peak absolute difference between the two is bounded provided that the LTI system $\mathcal{S}_{\text{LTI}}$ is stable (i.e. the eigenvalues of $\mathbf{A}$ have negative real parts) and the excitation u_δ is *sufficiently* small (see [31], Ch. 5, §9 or [88], §2.1).

Further results

The same results hold when an approximation of the nonlinear system about an equilibrium *trajectory* $x_{\text{eq}}(t)$ is needed for some constant value of the scheduling vector ϱ. However this time, the state space matrices of the approximate model are *time-varying* and as a result the resulting system is LTV (see also the discussion in Section 1.2.2):

$$\mathcal{S}_{\text{LTV}}: \quad \begin{aligned} \dot{\tilde{x}}_\delta &= \mathbf{A}(\varrho_{\text{eq}}, t)\tilde{x}_\delta + \mathbf{B}(\varrho_{\text{eq}}, t)u_\delta \\ y_\delta &= \mathbf{C}(\varrho_{\text{eq}}, t)\tilde{x}_\delta + \mathbf{D}(\varrho_{\text{eq}}, t)u_\delta \end{aligned} \tag{1.59}$$

with:

$$\tilde{x}_\delta \simeq x - x_{\text{eq}}(t) \tag{1.60}$$
$$u_\delta \simeq u - u_{\text{eq}}(t) \tag{1.61}$$
$$\tilde{y}_\delta \simeq y - y_{\text{eq}}(t). \tag{1.62}$$

In the gain scheduling context however, the designer is interested to approximate the behavior of a nonlinear parameter-dependent system for a *family* of equilibrium points rather than a *single* equilibrium point. In this context, the approximation results to an LPV system, being a very different object from a nonlinear system *disguised* in LPV form via state transformations (see discussion of the previous section), parameterized by the scheduling variable ϱ as:

$$\mathcal{S}_{\text{LPV}}: \quad \begin{aligned} \dot{\tilde{x}}_\delta &= \mathbf{A}(\varrho)\tilde{x}_\delta + \mathbf{B}(\varrho)u_\delta \\ y_\delta &= \mathbf{C}(\varrho)\tilde{x}_\delta + \mathbf{D}(\varrho)u_\delta. \end{aligned} \tag{1.63}$$

with:

$$\tilde{x}_\delta = x - x(\varrho) \tag{1.64}$$
$$u_\delta = u - u(\varrho) \tag{1.65}$$
$$\tilde{y}_\delta = y - y(\varrho). \tag{1.66}$$

For any frozen value $\varrho_{\text{eq}} \in \mathbf{\Gamma}$ of the scheduling vector, the LTV plant becomes an LTI and describes the dynamics of the nonlinear parameter-dependent system locally around the corresponding equilibrium point[14].

[13] Dependence on time and initial conditions are omitted.

[14] With a small abuse in notation, the approximated state is almost always noted as x_δ instead of $\tilde{x}_\delta$ to underline our self-satisfaction in the case that $\tilde{x}_\delta \mapsto x_\delta$.

Technical Note: As a technical note concerning Section 1.2.3, the extensive computational capabilities of commercial software for linearization should be outlined. In MATLAB® Simulink Control Design Toolbox for example there exists a full suite of specialized functions (`linearize`, `linmod`) that permit linearization of a nonlinear model around user-specified equilibrium points. This may be done for any portions of the nonlinear model by specifying input/output points in open or closed loop operation. This linearization can also be performed in a *frozen-time* context during a system simulation, providing thus the opportunity to analyze the stability of a gain-scheduled control system for a specific trajectory of the scheduling vector.

Linearizing a nonlinear model is not of course a trivial procedure and the algorithms used are either symbolic (block-by block analytic linearization) or numeric (numerical-perturbation linearization). For either case special attention should be made for discontinuous blocks, delays, saturations and also on the properties of each method (e.g. perturbation levels, open or closed loop linearization etc.) since many subsequent errors are due to a black-box conception of this process.

`Other methods`

Except for the traditional linearization methods based on Taylor-series expansion, there exist also other ways to linearize a nonlinear model. The ones briefly outlined here are *velocity-based linearization* and *feedback linearization*.

Velocity-based linearization is a method that approximates the dynamics of a nonlinear system around *any* given solution-trajectory instead of considering only equilibrium operation like the Jacobian-based approach. This method has in fact received great attention in the last twenty years because it has given birth to a new class of gain-scheduled control systems (namely for autopilots, power systems etc.) and has met significant success, even though there exist controversial opinions on its capabilities (see for example the rather amusing discussion appearing in [84]). A resume of the key points of this methodology are detailed in the next chapter.

Feedback linearization in its turn is a pure nonlinear method very popular in the 70's and 80's that tries to transform the state dynamics of a nonlinear system of the form:

$$\dot{x} = \mathbf{f}(x) + \mathbf{g}(x)u \tag{1.67}$$

$$y = \mathbf{h}(x) \tag{1.68}$$

to an LTI system, using a state feedback control law:

$$u = \boldsymbol{\phi}(x) + \boldsymbol{\psi}(x)v \tag{1.69}$$

and a state transformation $z = \mathbf{z}(x)$. Except for the aforementioned input $\mapsto$ state linearization, a full input $\mapsto$ output linearization is possible. This method is outside the scope of this work, however more details can be found in standard nonlinear control textbooks (see for example [75], Ch. 13 and references therein).

1.3 Linearization-based Gain Scheduling

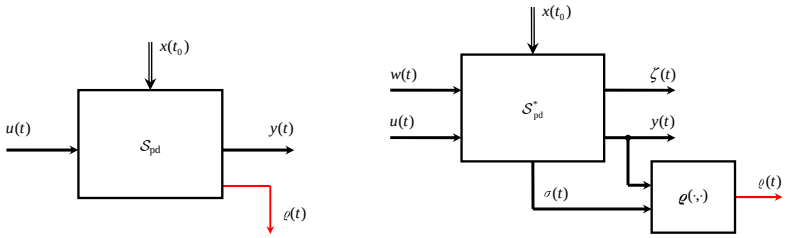
In this section a detailed review of the *Linearization-based Gain Scheduling - (LBGS)* method is presented. This general class of methods is considered for this monograph for the control of nonlinear parameter-dependent systems and is one of the most used in the control literature (see [88, 114] and references therein). The section starts with a detailed description of the corresponding procedure for the design of a nonlinear gain-scheduled controller, whereas the following sections present the methods that do not or do guarantee certain stability properties of the gain-scheduled loop.

1.3.1 Gain Scheduling Procedure

Start by considering the nonlinear state/output dynamics of a nonlinear parameter-dependent system $\mathcal{S}_{\text{pd}}$ (see Fig. 1.7a): NLPD system

$$\mathcal{S}_{\text{pd}}: \quad \begin{aligned} \dot{x} &= \mathbf{f}(x, u, \varrho) \\ y &= \mathbf{h}(x, u, \varrho) \end{aligned} \tag{1.70}$$

where $x \in \mathbb{R}^n, u \in \mathbb{R}^{n_u}, y \in \mathbb{R}^{n_y}$ are its state, input and output vectors correspondingly, $\varrho \in \mathbf{\Gamma} \subset \mathbb{R}^{n_\varrho}$ the *measured in real time* scheduling vector with $\mathbf{\Gamma}$ being a connected compact set and $\mathbf{f}, \mathbf{h}$ are nonlinear functions satisfying standard continuity & differentiability conditions.



(a) *Standard modeling* **(b)** *Alternative modeling*

Figure 1.7: *Nonlinear parameter-dependent system.*

Remark. Note that the modeling-notation existing in the survey of [88] is used (the scheduling vector however appears explicitly here) for reasons of simplicity. An alternative one is the one appearing in [114] (see Fig. 1.7b), which is more robust control/tracking-oriented: Alternative formulation

$$\mathcal{S}^*_{\text{pd}}: \quad \begin{aligned} \dot{x} &= \mathbf{f}(x, u, w, \sigma) \\ \zeta &= \mathbf{h}_\zeta(x, u, w, \sigma) \\ y &= \mathbf{h}_y(x, u, \sigma). \end{aligned} \tag{1.71}$$

The scheduling variable ϱ here is a function of σ that is a vector capturing parametric dependence of the plant and of the output y. Other slightly different formulations are also possible, e.g. see [79].

LBGS The linearization-based gain-scheduling procedure (LBGS) can be divided in five distinct steps:

Step 1 *Trim Analysis.* First the equilibrium states and the corresponding equilibrium inputs (or trim controls) are computed for every value ϱ_{eq} of the scheduling variable inside the domain of operation $\mathbf{\Gamma}$. This can be done either analytically or numerically as detailed in Section 1.2.1.3 and corresponds to finding the equilibrium manifold $\mathcal{E}_{\text{pd}}$ (see Eq. 1.20) of the system. The trim/equilibrium control hyper-surface $u_{\text{eq}} = u(\varrho_{\text{eq}})$ of the system that maintains the state to a corresponding equilibrium value $x_{\text{eq}} = x(\varrho_{\text{eq}})$ is thus obtained. In addition, the equilibrium outputs $y_{\text{eq}} = y(\varrho_{\text{eq}})$ may also be computed. In a noiseless environment, if the system is fed with an initial state $x(t_0) = x_{\text{eq}}$, then $x(t; t_0, x_{\text{eq}}) = x_{\text{eq}}, \forall t > t_0, t_0 \geq 0$ if the input is always $u(t; t_0) = u_{\text{eq}}$ for any value $\varrho_{\text{eq}} \in \mathbf{\Gamma}$.

Step 2 *System Linearization.* In this phase, the nonlinear system dynamics are approximated using Jacobian linearization for any member of the equilibrium manifold and thus, the following LPV system is obtained[15]:

$$\mathcal{S}_{\text{LPV}} : \quad \begin{aligned} \dot{x}_\delta &= \mathbf{A}(\varrho)x_\delta + \mathbf{B}(\varrho)u_\delta \\ y_\delta &= \mathbf{C}(\varrho)x_\delta + \mathbf{D}(\varrho)u_\delta \end{aligned} \tag{1.72}$$

with:

$$x_\delta = x - x(\varrho) \tag{1.73}$$
$$u_\delta = u - u(\varrho) \tag{1.74}$$
$$y_\delta = y - y(\varrho). \tag{1.75}$$

For every frozen value ϱ_{eq} of the scheduling vector, the above linear dynamics describe the initial nonlinear dynamics of Eq. 1.70 in the vicinity of the corresponding equilibrium state $x_{\text{eq}} = x(\varrho_{\text{eq}})$. Here two remarks should be made: first it is sometimes impractical to obtain symbolic expressions for the state matrices of Eq. 1.72 for *every* value of the scheduling vector; a designer may be happy with only a *tabulated* linear model around a significant number of operating points. If a linear model around an intermediate operating point is needed, then interpolation between tabulated points may be performed (this is very common with aeronautical systems where the nonlinear aerodynamic functions are computed in wind tunnels for a family of flight-operating conditions). Second, after the linear model is computed, certain open loop properties may be studied performing eigenvalue or Bode analysis iteratively for every member of the LPV plant; this is a major guideline for the next step: *Local Controller Synthesis*.

[15]The approximation $\tilde{x}_\delta$ of the plant's state is supposed to be near enough to the real value x_δ so that the 'tilde' sign may be omitted. The same also holds for the output y.

Local Controller Synthesis. This step involves the synthesis of a family of LTI controllers $\Sigma\big(K_{\text{LTI}}\big)$ for a set of linearized systems (being frozen instances of the plant $\mathcal{S}_{\text{LPV}}$) being computed for constant values ϱ^i_{eq} of the scheduling vector. These controllers are of the generic form: Step 3

$$K^i_{\text{LTI}}: \quad \begin{aligned} \dot{x}_k &= \mathbf{A}_k(\varrho^i_{\text{eq}})x_k + \mathbf{B}_k(\varrho^i_{\text{eq}})y_\delta \\ u_\delta &= \mathbf{C}_k(\varrho^i_{\text{eq}})x_k + \mathbf{D}_k(\varrho^i_{\text{eq}})y_\delta \end{aligned} \tag{1.76}$$

with $x_k \in \mathbb{R}^{n_k}$ being the controller state vector and u_δ, y_δ defined as:

$$u_\delta = u - u(\varrho^i_{\text{eq}}) \tag{1.77}$$

$$y_\delta = y - y(\varrho^i_{\text{eq}}). \tag{1.78}$$

The matrices $\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k$ and $\mathbf{D}_k$ are each time designed in such a way so that stability, performance and robustness properties are met for every member $\mathcal{S}^i_{\text{LTI}}$ of the family of linearized plants obtained for a family of values of the scheduling vector. This is in fact the strong point of the gain scheduling method: use the powerful synthesis methods of linear (mostly robust) control theory (such as $\mathscr{H}_2, \mathscr{H}_\infty$), in order to control an initially nonlinear system.

Local Controller Interpolation. In this important step lies the essence of gain scheduling design: *interpolation*. As it has been already mentioned, the scope of a gain-scheduled controller is to provide a control law *for any value* of the scheduling vector or else *for any point* of the operating domain $\boldsymbol{\Gamma}$ of the plant; be it a synthesis point or not, and not only for a family of synthesis points. Thus, when coming to on-line implementation, the designer will need only a small memory space for stocking the LTI controllers and an interpolation algorithm able to provide global operation. This may simply be restated as replacing in fact the variable ϱ_{eq} (this implies equilibrium operation) with ϱ and the constant equilibrium quantities $y_{\text{eq}} = y(\varrho_{\text{eq}}), u_{\text{eq}} = u(\varrho_{\text{eq}})$ by $y(\varrho), u(\varrho)$. For more details on this important subject, refer to the following section. Step 4

Controller Implementation & Validation. The last step of the gain-scheduling procedure concerns the final controller implementation. The main problem here is to construct the gain-scheduled controller in such a way that it provides an appropriate trim control input for every value of the scheduling vector. This in fact may be done in many ways (see for example [88], §3.1 or [114], §4.2 or even [149]) but the easiest one is to design the linear controllers so that they contain integral action; as a result $y_\delta \mapsto 0$ and the state of the controller x_k tends to an equilibrium value that corresponds to the u_δ needed for the system's state x to go to the equilibrium value dictated by the scheduling vector's equilibrium value ϱ_{eq}[16]. Step 5

[16] For more details see Section 1.3.3.

1.3.2 Ad-hoc Interpolation Methods

In this section the so-called ad-hoc controller interpolation methods are presented. It is reminded that an interpolation method is needed when the scheduling control law is computed at operating points that do not belong to the set of synthesis points. An interpolation strategy permits calculation of such a control law by combining controllers computed at a small number of synthesis points.

1.3.2.1 Controller Switching

Controller set

The *controller switching* method is the simplest one of all interpolation methods; to be more precise it does not involve any interpolation at all. A set $\Sigma(K_{\text{LTI}})$ of LTI controllers is computed:

$$\Sigma(K_{\text{LTI}}) := \left\{K^1, K^2, \ldots, K^k\right\} \tag{1.79}$$

Operating domain

where each controller $K^i = K(\varrho^i)$ of the set is calculated for fixed-equilibrium values of the scheduling vector $\varrho^i = \varrho^i_{\text{eq}}$ belonging to the system's operating domain $\mathbf{\Gamma}$[17]. Each controller is designed to be *robust* for a given subset Γ^i of the operating domain around the corresponding value ϱ^i (see Fig. 1.8), with:

$$\mathbf{\Gamma} = \bigcup_{i=1}^{k} \Gamma^i \tag{1.80}$$

and:

$$\Gamma^i \bigcap \Gamma^j = \emptyset, \quad \text{with} \quad \{i, j\} = 1, \ldots, k \quad \text{and} \quad i \neq j. \tag{1.81}$$

Switching discussion

The last condition means that there exists no overlapping in the switching regions Γ^i and the controllers are simply switched according to the scheduling vector trajectory $\varrho(t)$. Now this may be the source of control signal discontinuities and chattering behavior when passing from one scheduling-switching region to the next. This may easily seen by considering the control signal produced from a controller K^1 up to a critical switching time $t = t_{\text{sw}}$. Suppose this controller to be of the standard form:

$$K^1 : \quad \begin{aligned} \dot{x}^1_k &= \mathbf{A}^1_k x^1_k + \mathbf{B}^1_k y_\delta \\ u_\delta &= \mathbf{C}^1_k x^1_k + \mathbf{D}^1_k y_\delta. \end{aligned} \tag{1.82}$$

Then the control signal is:

$$u_\delta(t) = \mathbf{C}^1_k \left[e^{\mathbf{A}^1_k (t-t_0)} x^1_k(t_0) + \int_{t_0}^{t} e^{\mathbf{A}^1_k (t-\tau)} \mathbf{B}^1_k y_\delta(\tau) \right] + \mathbf{D}^1_k y_\delta(t). \tag{1.83}$$

From Eq. 1.83 it is evident that if at $t = t_{\text{sw}}$, the controller matrices change their values, then the control signal will be discontinuous. The solution to this problem is fairly simple but it is not very often implemented in real systems.

[17] The controllers are of the form as in Eq. 1.76.

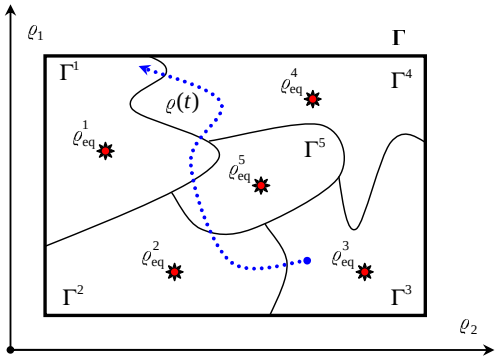


Figure 1.8: Controller switching.

To ensure that $u_\delta(t_{\text{sw}}^+) = u_\delta(t_{\text{sw}}^-)$, it suffices to initialize the new controller K^2 entering on line to a state that ensures bumpless transfer. Indeed:

$$u_\delta(t_{\text{sw}}^+) = \mathbf{C}_k^2 x_k^2(t_{\text{sw}}^+) + \mathbf{D}_k^2 y_\delta(t_{\text{sw}}^+) \triangleq u_\delta(t_{\text{sw}}^-) \tag{1.84}$$

and one has to solve for the initialization state of the second controller $x_k^2(t_{\text{sw}}^+)$[18]. For digitally implemented systems where the difference $\delta t = t_{\text{sw}}^+ - t_{\text{sw}}^- \to 0$, the bump in the control signal may be done arbitrarily small (see [64] or [59]).

`Features`

The major advantage of this interpolation method is that it is fairly simple to implement and has been used extensively in real systems. The operating domain is divided in rectangular regions and each controller K^i is valid for given ranges on each component of the scheduling vector. The major disadvantage was already stressed: control signal continuity and stability during transitions.

`Applications`

Applications of switching-based gain scheduling are: in [106] a switched gain-scheduled controller for a two link wafer transfer robot system is designed with the scheduling variable being the rotational angle difference of the two links. In [125], a technique for aircraft control is used, generating smooth control signals using LPV control and LMI's. In a similar context, a switching control scheme for the control of magnetic bearings is used in [152]. A not pure switching strategy (involves interpolation in the union of the switching regions) is used in [6] for the control of the water level of a steam generator. An aircraft control example with the speed as a scheduling variable is considered in [64].

Finally, some more theoretical work on the subject, with extensions to nonlinear control and hybrid systems, can be also found in [22, 91, 95]. The first considers a hierarchical switching controller architecture over a set of moving equilibria and uses equilibria-based Lyapunov functions to guarantee stability. The second considers also Lyapunov-based control and regions of stability for a certain class of nonlinear systems. The third one finally considers performance of switched LPV systems and extensions to hybrid systems.

[18] Notice that in general there is more than one solution to Eq. 1.84 and in order to obtain $x_k^2(t_{\text{sw}}^+)$ the pseudo-inverse of $\mathbf{C}_k^2$ may be used (see eq. 1.91).

1.3.2.2 Controller Blending

The *controller blending* method can be seen as a generalization to *controller switching*. Instead of switching controllers when passing from the one operating region to the next, the output control signals of adjacent controllers are blended in order to provide the final control command.

LTI Controller

Consider (for simplicity) a planar operating region $\mathbf{\Gamma}$ and the corresponding two-dimensional scheduling vector $\varrho = [\varrho_1 \quad \varrho_2]^T$ (see Fig. 1.9a). A set $\Sigma_{\mathcal{K}}$ of linear controllers is computed for fixed equilibrium values $\varrho_{\text{eq}}^{i,j}$ of the scheduling vector (red stars). The controllers are distributed evenly in the horizontal and vertical directions and produce rectangular scheduling regions Γ^i. To each rectangular scheduling region Γ^i correspond four controllers $K^{i,j}$, with 'i' being the region index and 'j' (with $j = 1, \ldots, 4$) the controller index of the i'th region, taken in an anti-clockwise manner[19]. An LTI controller $K^{i,j}$ of the form:

$$K^{i,j}: \quad \begin{aligned} \dot{x}_k^{i,j} &= \mathbf{A}_k^{i,j} x_k^{i,j} + \mathbf{B}_k^{i,j} y_\delta \\ u_\delta^{i,j} &= \mathbf{C}_k^{i,j} x_k^{i,j} + \mathbf{D}_k^{i,j} y_\delta \end{aligned} \tag{1.85}$$

is calculated for every synthesis point corresponding to an equilibrium value $\varrho^{i,j} = \varrho_{\text{eq}}^{i,j}$ of the scheduling vector. The total interpolated control output u_δ, for any value of the scheduling vector inside an operating region Γ^i, is calculated by *blending* the four control signals $u_\delta^{i,j}, j = 1 \ldots 4$.

This is done as a function of the distances of the current operating point (cyan star) to the four synthesis points at the edges of the corresponding operating region (see Fig. 1.9a). These distances a_l (with $l = 1 \ldots n_\varrho$ and n_ϱ being the dimension of the scheduling vector) are normalized quantities with $0 \leq a_l \leq 1, \forall l$.

Normalized distances

For a two-dimensional scheduling vector and any rectangular scheduling region Γ^i they are defined as:

$$a_1(t) = \frac{\varrho_1(t) - \varrho^{i,1}}{\varrho^{i,4} - \varrho^{i,1}} \equiv \frac{\varrho_1(t) - \varrho^{i,2}}{\varrho^{i,3} - \varrho^{i,2}} \tag{1.86}$$

$$a_2(t) = \frac{\varrho_2(t) - \varrho^{i,1}}{\varrho^{i,2} - \varrho^{i,1}} \equiv \frac{\varrho_2(t) - \varrho^{i,4}}{\varrho^{i,3} - \varrho^{i,4}}. \tag{1.87}$$

Control input

The total blended control input u_δ^i, being a function of the normalized distances and the control signals of each controller $u_\delta^{i,j}$, is computed as:

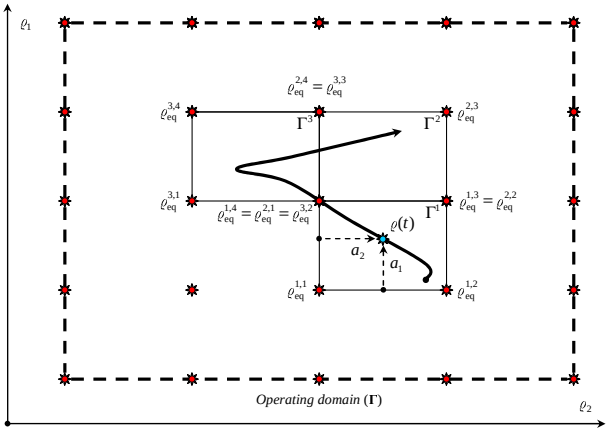
$$u_\delta^i = \big[1 - a_1(t)\big] u_\delta^{i,\{1,2\}} + a_1(t) u_\delta^{i,\{3,4\}} \equiv u_\delta^{i,\{1,2,3,4\}} \tag{1.88}$$

with:

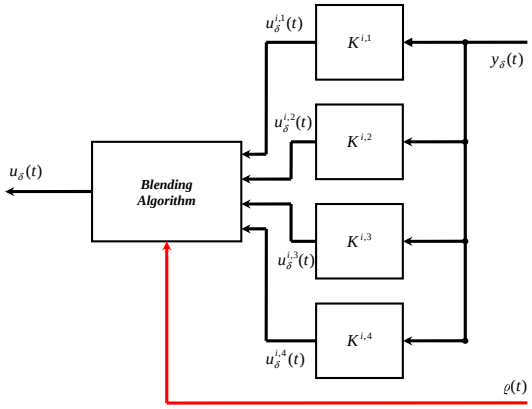
$$u_\delta^{i,\{1,2\}} = \big[1 - a_2(t)\big] u_\delta^{i,1} + a_2(t) u_\delta^{i,2} \tag{1.89}$$

$$u_\delta^{i,\{3,4\}} = \big[1 - a_2(t)\big] u_\delta^{i,4} + a_2(t) u_\delta^{i,3}. \tag{1.90}$$

[19] Obviously some controllers may be used for up to four neighbor regions, depending their position on the scheduling region $\mathbf{\Gamma}$; as a result this numbering is non-unique.



(a) *Operating region visualization*



(b) *Controller realization*

Figure 1.9: Controller blending technique.

A possible interpolation scenario is depicted in Fig. 1.9a, with the trajectory $\varrho(t)$ passing through three scheduling regions ($\Gamma^1 \nrightarrow \Gamma^3 \nrightarrow \Gamma^2$). During the first transition (which is a little exaggerated since the trajectory passes exactly from the synthesis point $\varrho_{\text{eq}}^{1,4}$) all three controllers $K^{1,1}, K^{1,2}, K^{1,3}$ turn off and controllers $K^{3,1}, K^{3,3}, K^{3,4}$ go on-line to replace them. However, controller $K^{1,4} \equiv K^{3,2}$ remains on-line for both regions Γ^1, Γ^3. Similarly, during the second transition, controller $K^{1,4}$ remains always on-line as well as controller $K^{3,3}$; however controllers $K^{3,1}, K^{3,4}$ give their place to $K^{2,2}, K^{2,3}$ respectively.

Features

A simplified structure of the interpolator is visualized in Fig. 1.9b. The scheme is rather simplified but it shows the essence of the method: only the *outputs* of the controller are processed and not the controller themselves as with other interpolation strategies (see following sections). However, a hierarchical mechanism should be added so as to decide when and how to switch on and off the controllers. This is a major advantage of this controller interpolation method: it is not obligatory to use controllers of the same structure or of the same complexity for each synthesis point since it is only each controller's output that is processed. This is not the case with other controller interpolation methods such as *gain blending* where the controller structure/order remains the same and the interpolation procedure is done on the controller parameters.

Another advantage of the method is in terms of the numerical computations needed to obtain the control law; in [73, 74] it is argued that this method is significantly faster in terms of multiplications & additions needed to compute the interpolated control signal in comparison for example with *state-space matrix* or *zero-pole-gain* interpolation.

This method however presents some important disadvantages: an important one is *controller initialization*. Consider once again the scenario of Fig. 1.9a where the scheduling vector crosses the boundary of the regions Γ^3 and Γ^2. At the exact moment t_{sw}, where the scheduling vector is on the border of the two regions, the control signal is affected by the outputs of only two controllers and two new controllers should be put on-line and *initialized to some state*, in order to be able to perform interpolation in region Γ^2 for $t > t_{\text{sw}}$ (during of course the time that $\varrho(t) \in \Gamma^2$).

However, this initialization process is not a trivial matter since if these controllers are switched on with zero initial state conditions there will probably be an initial transient on the total control output due to the inconsistency of the newly entered controllers' states added and the operating situation of the system before the switching. This transient may be rendered smaller if these states are initialized in a smarter way. A possible solution is to re-initialize *all four* controllers of the new region Γ^3 to a state dictated by the actual control signal $u_\delta(t_{\text{sw}})$, where t_{sw} is the switching time (see [137] or Chapter 5):

$$x_k^{i,j}(t_{\text{sw}}) = \left(\mathbf{C}_k^{i,j}\right)^+ \left[u_\delta(t_{\text{sw}}) - \mathbf{D}_k^{i,j} y_\delta(t_{\text{sw}})\right].^{20} \tag{1.91}$$

[20] The '+' sign in the exponent denotes the Moore-Penrose pseudo-inverse of a matrix.

These inevitable transients become more annoying when the average period that the scheduling vector stays inside any operating region is small. This may happen either due to the fact that $\varrho(t)$ varies relatively fast with respect to the plant's settling time or if the scheduling regions are too small; i.e. the gridding of the operating domain is overly dense. In this case, even though the designer could expect an increase in performance when the synthesis points augment, the initial transients may render the closed loop system slow. This problem may be corrected if 'sufficiently fast' dynamics are assigned to the LTI controllers that will internally compensate the state inconsistency.

This initial transient problem may yet be amplified if the scheduling vector demonstrates big step changes from one value to the next, making thus the interpolator jump to interpolation regions that are not neighbor. In this case the state initialization may not be useful at all and the transients heavier. A possible solution to this problem is to filter the scheduling vector with a *low-pass* filter in order to force the scheduling vector pass from all regions in between and spend a finite time at each one. However much attention should be paid on the filter's bandwidth so as not to augment the closed-loop rise & settling times.

Another disadvantage of this method is the fact that the scheduler needs four controllers (if there are two parameters, triangles could be considered and thus three controllers are enough) to be implemented, apart from the unit that performs the state initialization. This strategy is more complex than say, a gain interpolation one where only a single controller (e.g. a PID) is implemented and solely its gains interpolated, depending on the location of the scheduling vector.

Some solid work on the subject appears on two nearly identical papers (see [73, 74]), where this method is compared to state space & zero-pole-gain (ZPK) interpolation methods[21] and several of its details are discussed. However some of the disadvantages that this method possesses are not stressed out. Bibliography

Notable work on aeronautical systems (namely missile autopilots) controlled with this type of interpolation are presented in [35, 81] and [137] respectively. In [81], a missile autopilot is designed using the controller blending method but the simulation results are not so thorough, even though local linear equivalence properties for the gain-scheduled controller (see Section 1.3.3) are exploited. In [35] a μ-analysis method is used for the LTI controllers for the design of a 3DOF missile autopilot. The most complete treatment on the subject can be found in [137] (or equivalently in Chapter 5 of this monograph), where extensive simulations are used to validate and compare the approach with an alternative observer based blending strategy, detailed in Section 1.3.2.6. A controller blending approach is also used in [58] for the control of a power plant boiler. The scheduling variables used are the steam temperature and pressure and the effectiveness of the gain-scheduled over a robust control scheme is demonstrated. Finally in [60] an interesting LPV-based method for the control of a vehicle powertrain using static $\mathscr{H}_\infty$ controllers and controller blending is proposed.

[21] These methods are analyzed in the sections to follow.

1.3.2.3 ZPK Interpolation

The *zero-pole-gain interpolation* (ZPK) method is one of the standard techniques used for controller interpolation. For simplicity SISO systems will be considered, however the analysis could be extended for MIMO ones, even though the method is not adapted for truly multivariable setups.

`LTI controller`

Consider once again a set $\Sigma(K_{\text{LTI}})$ of LTI-SISO controllers designed for fixed values of the scheduling vector inside a scheduling region $\mathbf{\Gamma}$[22]. Each controller $K^{i,j}$ may be represented in the s-domain in a ZPK form as:

$$K(s)^{i,j} = K^{i,j} \frac{\prod_{k=1}^{m}\big(s - z_k^{i,j}\big)}{\prod_{l=1}^{n}\big(s - p_l^{i,j}\big)} \tag{1.92}$$

with $z_k^{i,j}, p_l^{i,j}$ being the k-*th* (respectively l-*th*) zero (respectively pole) and $K^{i,j}$ the dc-gain of the j-*th* controller[23] at the corresponding i-*th* scheduling region Γ^i.

`Global controller`

Each zero, pole and gain is interpolated in the same way as with the controller blending method. For each value $\varrho(t)$ of the scheduling vector, the normalized distances are given by Eqs. 1.86, 1.87 and thus the final interpolated compensator has the following form:

$$K(s,\varrho) = K(\varrho) \frac{\prod_{k=1}^{m}\big(s - z_k(\varrho)\big)}{\prod_{l=1}^{n}\big(s - p_l(\varrho)\big)}. \tag{1.93}$$

The zeros, poles and gain of the compensator are now dependent on time since $\varrho(t)$ draws a trajectory inside the operating region of the system. Consider for example the k-*th* zero of the interpolated compensator when the scheduling vector is inside the i-*th* scheduling region; its (time-dependent) value is given by:

$$z_k(\varrho) = \big[1 - a_1(t)\big] z_k^{i,\{1,2\}} + a_1(t) z_k^{i,\{3,4\}} \equiv z_k(\varrho)^{i,\{1,2,3,4\}} \tag{1.94}$$

with:

$$z_k^{i,\{1,2\}} = \big[1 - a_2(t)\big] z_k^{i,1} + a_2(t) z_k^{i,2} \tag{1.95}$$

$$z_k^{i,\{3,4\}} = \big[1 - a_2(t)\big] z_k^{i,4} + a_2(t) z_k^{i,3}. \tag{1.96}$$

`Features`

The major advantage of this method is that it maintains 'a good engineering feeling' in the interpolation process. Indeed, it is more natural and straightforward to interpolate the zeros-poles-gains of a controller than say, the coefficients of a transfer function since the effect of a changing pole (or a zero or also a gain) could be easily linked to the step or frequency response of the closed loop system. However this method becomes rather complicated in the case of complex poles/zeros and for multivariable systems.

[22] Consider a two-dimensional scheduling variable ϱ and the same setup and notation as in Section 1.3.2.2.

[23] Once again $j = 1, \ldots, 4$, since rectangular scheduling regions Γ^i are considered.

There is however a significant issue for this type of interpolation, concerning implementation. Modern gain-scheduled controllers are implemented using digital components thus a fundamental question arises: should a controller be first discretized and then interpolated in the z-domain or first interpolated in the s-domain and then discretized?

Example

To answer this question consider a single pole (real for simplicity) s_p and the mapped to the z-domain equivalent one $z_p = e^{s_p T}$, where T is the sampling period. Suppose both poles are *equally* perturbed to a new value s_p^* and z_p^* with $s_p^* = s_p + \delta, z_p^* = z_p + \delta$ and $\delta > 0$ a small real number. The quotient $q(s_p, T)$[24] of the difference between the z-domain mapped perturbed pole $z_p^* = \mathscr{Z}(s_p^*)$ (perturbation in the s-domain) and the nominal pole z_p, and the difference between the s-domain mapped perturbed pole $s_p^* = \mathscr{S}(z_p^*)$ (perturbation in the z-domain) and the nominal pole s_p is:

$$q(s_p, T) = \frac{z_p^* - z_p}{s_p^* - s_p} (\equiv \frac{z_\delta}{s_\delta}) = \frac{e^{(s_p+\delta)T} - e^{s_p T}}{\frac{\ln\left(e^{s_p T} + \delta\right)}{T} - s_p}. \tag{1.97}$$

From the following figure it can be seen that this difference quotient is rather small; this means that a perturbation on the s-domain pole results to a much smaller perturbation to the corresponding pole on the z-domain than the inverse. As a result, it is preferable to perform the interpolation first to the s-domain and then discretize the controller since numerical sensitivity is bigger in the z-domain (a similar but approximative analysis may be found in [74], pp. 178).

Applications

A recent paper addresses the control of a pick and place machine and performs interpolation using the length of the beam used for transportation [108]. Also in the reference paper [103], a missile autopilot using robust $\mathscr{H}_\infty$ controllers scheduled on the vertical acceleration and Mach number but the control scheme is complicated and more performing controllers are proposed in later works.

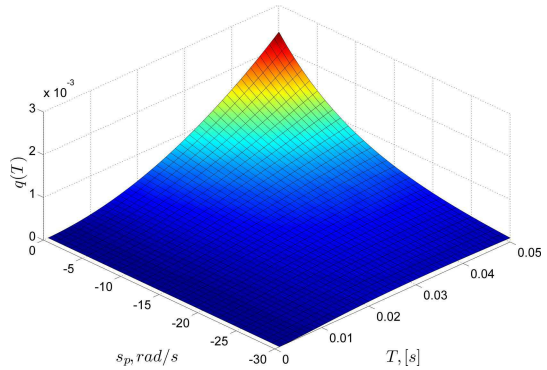


Figure 1.10: Quotient concerning ZPK mapping.

[24] The dependence on the perturbation δ is considered constant; here a value $\delta = 0.1$ rad/s is taken for both mappings.

1.3.2.4 Transfer Function Coefficient Interpolation

The *transfer function coefficient* is similar to the ZPK interpolation technique. Once again SISO systems will be considered for simplicity; in any case this method also is not particularly suited for MIMO setups.

`LTI controller`

The transfer function of an LTI SISO controller may be written in the following (alternative to Eq. 1.92 ZPK) form[25]:

$$\begin{aligned} K(s)^{i,j} &= \frac{\sum_{k=1}^{m} \beta_k^{i,j} s^k}{\sum_{l=1}^{n} \alpha_l^{i,j} s^l} \\ &= \frac{\beta_k^{i,j} s^k + \beta_{k-1}^{i,j} s^{k-1} + \ldots + \beta_1^{i,j} s + \beta_0^{i,j}}{\alpha_l^{i,j} s^l + \alpha_{l-1}^{i,j} s^{l-1} + \ldots + \alpha_1^{i,j} s + \alpha_0^{i,j}}. \end{aligned} \tag{1.98}$$

`Global controller`

The numerator and denominator have m and n coefficients respectively that define their dynamics. In the gain-scheduling context, a set of controllers is again designed at a number of synthesis points and the corresponding controller coefficients for the i-th scheduling region and the j-th controller of this region are denoted by $\beta_k^{i,j}$ and $\alpha_l^{i,j}$. To obtain a global interpolated controller:

$$\begin{aligned} K(s,\varrho) &= \frac{\sum_{k=1}^{m} \beta_k(\varrho) s^k}{\sum_{l=1}^{n} \alpha_l(\varrho) s^l} \\ &= \frac{\beta_k(\varrho) s^k + \beta_{k-1}(\varrho) s^{k-1} + \ldots + \beta_1(\varrho) s + \beta_0(\varrho)}{\alpha_l(\varrho) s^l + \alpha_{l-1}(\varrho) s^{l-1} + \ldots + \alpha_1(\varrho) s + \alpha_0(\varrho)} \end{aligned} \tag{1.99}$$

the transfer function coefficients of adjacent controllers are interpolated using the same formulas as in the previous section.

`Features`

Now this method seems well suited for SISO systems and relatively low order controllers (lead, lag, PID) since if the controller's order rises the effect of interpolation on its stability becomes less clear. This means that there is no guarantee that the interpolated controller will have linearly varying dynamics even if the coefficients are updated linearly (this is not however the case with ZPK interpolation which is more direct).

`Applications`

Relative work on the subject may be found in [150], where a missile autopilot is obtained using interpolation of controller coefficients. In [62] a similar system is considered but this time $\mathscr{H}_\infty$ loop shaping controllers are designed and a least-squares analytic approach to obtain the global controller coefficients is adopted. In [92] a robust controller for aircraft is designed whereas finally in [26] a MIMO controller for a frigate ship is computed using sea state data and the ship's velocity. In this final work, an LPV gain-scheduled controller is compared with the linearization-based with coefficient interpolation and with a robust LTI; in all cases the gain-scheduled schemes perform better than the robust one.

[25] Very often the coefficients of the transfer function are normalized so that the *highest power* coefficients of both the numerator and the denominator become unitary.

1.3.2.5 State Space Matrix Interpolation

The *state-space matrix interpolation* is a method that offers a nonlinear gain-scheduled controller by blending the coefficients of the state space representation of local LTI controllers. Consider for example the following state space representation of a controller $K^{i,j}$ designed at the j-*th* point of the i-*th* scheduling region of the operating domain $\mathbf{\Gamma}$ of a nonlinear parameter-dependent system:

LTI controller

$$K^{i,j}: \quad \begin{aligned} \dot{x}_k^{i,j} &= \mathbf{A}_k^{i,j} x_k^{i,j} + \mathbf{B}_k^{i,j} y_\delta \\ u_\delta^{i,j} &= \mathbf{C}_k^{i,j} x_k^{i,j} + \mathbf{D}_k^{i,j} y_\delta \end{aligned} \tag{1.100}$$

In contrast to the *controller blending* method which interpolates the outputs of the controllers $u_\delta^{i,j}$, the *state-space matrix interpolation* method interpolates directly their internal structure. As a result, the structure of the gain-scheduled controller $K(\varrho)$ will be:

Global controller

$$K(\varrho): \quad \begin{aligned} \dot{x}_k &= \mathbf{A}_k(\varrho) x_k + \mathbf{B}_k(\varrho) y_\delta \\ u_\delta &= \mathbf{C}_k(\varrho) x_k + \mathbf{D}_k(\varrho) y_\delta. \end{aligned} \tag{1.101}$$

Now each element of each matrix $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ is obtained linearly by interpolating the four adjacent controller matrices using the normalized distances a_1, a_2 (when scheduling on the plane) that are a function of the scheduling vector as in the previous sections. Consider for example the matrix $\mathbf{A}$:

$$\mathbf{A}_k(\varrho) = \big[1 - a_1(t)\big]\mathbf{A}_k^{i,\{1,2\}} + a_1(t)\mathbf{A}_k^{i,\{3,4\}} \equiv \mathbf{A}_k(\varrho)^{i,\{1,2,3,4\}} \tag{1.102}$$

with:

$$\mathbf{A}_k^{i,\{1,2\}} = \big[1 - a_2(t)\big]\mathbf{A}_k^{i,1} + a_2(t)\mathbf{A}_k^{i,2} \tag{1.103}$$

$$\mathbf{A}_k^{i,\{3,4\}} = \big[1 - a_2(t)\big]\mathbf{A}_k^{i,4} + a_2(t)\mathbf{A}_k^{i,3}. \tag{1.104}$$

Features

This method, even though it seems rather straightforward, it presents several disadvantages compared to other methods. First, it is rather demanding on calculations since all elements of a state space realization need to be interpolated; this may be very conservative. For example the state space representation of a SISO, 2^{nd} order controller may have up to nine coefficients whereas a ZPK or transfer function realization up to five (for a 3^{rd} order controller it is even worse: sixteen and seven respectively). Second, the effect of interpolation to the zeros and poles and thus to the final controller dynamics is not straightforward and may also lead to numerical problems for ill-conditioned realizations. Third, a similar controller structure/realization is assumed for all synthesis points in order for the interpolation to have a meaning since interpolating between different states is not a sound strategy (see [73] for more details).

For these reasons, there does not exist significant applications using this type of interpolation, even though some work on stability preserving interpolation can be found in [133], as it will be presented in the next part of this chapter.

1.3.2.6 Observer/State Feedback Interpolation

The *observer/state feedback interpolation* method is somewhat close to the *state-space matrix interpolation* one in the sense that controller matrices are interpolated in order to obtain a gain-scheduled controller. Consider once again the setup used in the previous sections concerning the synthesis points and the scheduling vector.

A well-known control strategy for MIMO LTI systems is the observer/state feedback compensator. For the initial nonlinear parameter-dependent system in Eq. 1.70, consider a family of (strictly proper for simplicity) linearized plants computed at a number of operating points[26]:

$$\mathcal{S}_{\text{LTI}}^{i,j}: \quad \begin{aligned} \dot{x}_\delta &= \mathbf{A}^{i,j}x_\delta + \mathbf{B}^{i,j}u_\delta \\ y_\delta &= \mathbf{C}^{i,j}x_\delta. \end{aligned} \tag{1.105}$$

LTI Controller

An observer/state feedback controller for each of the above linearized systems is written as:

$$K^{i,j}: \quad \begin{aligned} \dot{\hat{x}}_\delta &= \mathbf{A}^{i,j}\hat{x}_\delta + \mathbf{B}^{i,j}y_\delta + \mathbf{K}_{\text{o}}^{i,j}(y_\delta - \mathbf{C}^{i,j}\hat{x}_\delta) \\ u_\delta &= \mathbf{K}_{\text{c}}^{i,j}\hat{x}_\delta \end{aligned} \tag{1.106}$$

with $\mathbf{K}_{\text{c}}^{i,j}, \mathbf{K}_{\text{o}}^{i,j}$ being respectively the observer and controller matrices for each design point. Now the observer is estimating the state error close to the equilibrium point whereas the controller uses this estimation to perform a pole placement. The interpolation procedure here updates *all* matrices as a function of the scheduling vector (the plants' included) in order to provide the following gain-scheduled controller:

Global controller

$$K(\varrho): \quad \begin{aligned} \dot{\hat{x}}_\delta &= \mathbf{A}(\varrho)\hat{x}_\delta + \mathbf{B}(\varrho)y_\delta + \mathbf{K}_{\text{o}}(\varrho)\big(y_\delta - \mathbf{C}(\varrho)\hat{x}_\delta\big) \\ u_\delta &= \mathbf{K}_{\text{c}}(\varrho)x_\delta \end{aligned} \tag{1.107}$$

Features

Now here arise several issues: first, it is clear that this interpolation method is very demanding in calculations since both the controller structure (matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$) *and* the controller dynamics (matrices $\mathbf{K}_{\text{c}}, \mathbf{K}_{\text{o}}$) need to be updated as a function of the scheduling vector $\varrho(t)$. Second, an important issue is the update of the controller structure itself; normally, the observer should reconstruct the state of the linearized system at any possible operating point. This is done by considering the system matrices at the specific operating point; however given that these matrices are in general computed at a small number of operating points (along with the controller/observer gain matrices), it is clear that an interpolation used for any other operating point may yield a different linearized plant than the one *explicitly or symbolically* computed at this operating point.

[26] Once again 'i' denotes the scheduling region and 'j' the index of the controller as in the previous sections. Also the indexes i,j are omitted for the state, input and output of each linearized plant for simplicity. However it remains evident that the errors $x_\delta, u_\delta, y_\delta$ are taken with respect to the equilibrium values of *each* synthesis point.

To illustrate this fact, consider the following (oversimplified) example: suppose that a nonlinear parameter-dependent system has been linearized around the origin and the following LPV plant describes its dynamics around this point: Example

$$\dot{x}_\delta = 2\varrho^2 x_\delta = \mathbf{A}(\varrho)x_\delta. \tag{1.108}$$

Suppose also that the scheduling vector ϱ may vary arbitrarily between 1 and 2 and the state deviation is defined as:

$$x_\delta = x - x_{\text{eq}} \equiv x. \tag{1.109}$$

Suppose that an observer for the system's state is to be constructed using the two extremal values $\mathbf{A}(\varrho_{\min}) = 2 \cdot 1^2 = 2$ and $\mathbf{A}(\varrho_{\max}) = 2 \cdot 2^2 = 8$ of the system dynamics, for the operating point corresponding to $\varrho = 1.5$. If a linear interpolation between the two values is taken, then the interpolated matrix used in the observer will be $\tilde{\mathbf{A}} = 2 + \frac{8-2}{2} = 5$.

However the true value, if exact calculations were to be used, is computed as $\mathbf{A}(\varrho)\big|_{1.5} = 2 \cdot 1.5^2 = 4.5$. Now, if the synthesis point were four (for $\varrho = 1, 1.33, 1.66$ and 2 respectively) then the interpolated values would have been 4.55; much closer to the explicitly calculated one.

This simplistic example shows the danger of estimating the dynamics of a 'wrong' plant which may result in poor performance, if the operating domain gridding is not done correctly. A solution to this problem is to perform a denser gridding when linearizing the plant but calculate the controller/observer matrices at a smaller number of points. A possible scenario is depicted in Fig. 1.11 where the blue stars denote linearization points and the red both linearization & controller synthesis-interpolation ones[27].

A third issue has to do with appropriate realization of the controller and observer so that there is a meaning in interpolation. In addition, given that the interpolation is on the controller matrices, it is not very clear if the interpolated quantities will behave well as far as stability is concerned (this in fact is the same problem with state-space matrix interpolation).

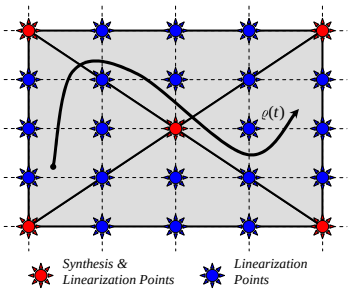


Figure 1.11: A potential gridding scenario.

[27] In this example four triangular interpolation regions are defined that reduce conservatism at it will be discussed in Section 1.4.

The major advantage of this control configuration is that it is based on an estimation of the nonlinear plant's state itself; this feature may be *highly desirable* for the validation of a gain-scheduled control scheme. In addition, the synthesis of observer-based state feedback schemes is rather simple and may easily treat MIMO systems, in contrast to say, *ZPK* or *transfer function coefficient* interpolation. In addition, discretization-implementation is by far more straightforward in state-space setups. Finally, there exist significant work in the field of *stability preserving interpolation* schemes using this type of control loops (see later sections).

`Applications` As far as applications of this interpolation method are concerned, in [7, 137] the observed-based interpolation technique was compared to the controller blending one for an integrated flight and propulsion control system and a missile autopilot respectively. In [23], a gain-scheduled controller was used to attenuate disturbances due to engine-induced vibrations whereas in [147], arbitrary $\mathscr{H}_\infty$ compensators are converted to state feedback/observer form for the control of a launcher. In [100] a discrete time controller is interpolated for the stabilization of an electrostatic levitator whereas in [29] a nice application in a multi-motor web transport system is presented. Finally [65], is a rather good reference on the subject.

`Youla param/tion` An interesting extension to the state feedback/observer interpolation method is the *Youla parameter-based interpolation* method. It is known that any stabilizing LTI controller K for an LTI plant $\mathcal{S}_{\text{LTI}}$ may be written as the l-LFT of a stabilizing observer-based state feedback controller $\mathcal{J}$ *plus* a free, stable system $\mathcal{Q}$ which is called the *Youla* parameter, being driven by the *innovations signal*.

If desired stabilizing (but of arbitrary structure) compensators $K^{i,j}$ are designed for some specific operating points of the plant, it is possible to retain the same observer-state feedback controller for every operating point and change only the corresponding Youla parameter $\mathcal{Q}^{i,j}$ so as to obtain:

$$K^{i,j} = \mathcal{F}_l(\mathcal{J}, \mathcal{Q}^{i,j}). \tag{1.110}$$

The Youla parameter may then be scheduled in order to obtain a scheduler transfer function $\mathcal{Q}(\varrho)$ and therefore a scheduled global controller $K(\varrho)$. Possible stability preserving extensions to this method are considered in the next section.

`Applications` An interesting application of this method is found in [102] where the scheduled parameter $\mathcal{Q}$ is used to achieve rejection of vibrations in magnetic bearing systems. In [134] a SIMO servo controller is scheduled using two extremal LTI controllers (that correspond to different Youla parameters): the first one for performance and the second one for good robustness and error tracking suppression. In [104] a scheme for gain scheduling control is devised when the scheduling parameter ϱ is not known and need to be estimated; this is done using the Youla parameterization procedure mentioned above. Finally in [110] some theoretical work is done in the context of continuation of observer-based structures for issues arising from interpolation in gain scheduling control.

1.3.2.7 Other Interpolation Schemes

In this section some additional interpolation techniques will be cited that are either less used in the bibliography, or they may be regarded as transformations to the existing methods detailed in the previous sections.

`Gain blending`

The first technique is called *Gain Blending* and is maybe the more standard interpolation technique of all, since due to this method the *Gain Scheduling* terminology rises. A very often used industrial controller is the 'PID' type, existing both in a simple SISO form or in more complex inner-outer loop or MIMO forms. It is not needed to cite the benefits from PID control since they are widely accepted: ease of use and implementation, optimality, engineering intuition preserved etc[28]. The PID controller is nicely tailored for a great variety of systems (automotive, aeronautical etc.) and tuning a PID is the most frequent task a systems engineer may be asked to perform on the field. In addition, adaptive PID controllers are an excellent (and preferable) choice for the control of parameter-dependent systems.

The *Gain Blending* technique is exactly that: for a set of operating points, compute a family of PID controllers of the form:

$$K(s)^{i,j} = K_{\mathrm{p}}^{i,j} + K_{\mathrm{i}}^{i,j}\frac{1}{s} + K_{\mathrm{d}}^{i,j}s. \tag{1.111}$$

Then interpolate the gains $K_{\mathrm{p}}, K_{\mathrm{i}}, K_{\mathrm{d}}$ at each operating region following the scheduling vector evolution in order to obtain a gain-scheduled controller[29].

`Applications`

Some nice applications of gain blending can be found for example in [138] where a missile autopilot was calculated by scheduling PID controllers of a special type or in [139] where a re-entry vehicle autopilot is considered[30]. Another good practical example can be found in [69] where a PID controller is scheduled for the control of a diesel engine.

`Riccati interpolation`

Another interpolation method used in the context of robust gain-scheduled control is based on the interpolation of the solutions $\mathbf{X}_\infty, \mathbf{Y}_\infty$ of Riccati equations relevant to $\mathscr{H}_\infty$ control synthesis. For further details on applications of this method see [8, 112].

`Other methods`

Finally, other methods could be considered such as coprime factor scheduling, fuzzy interpolation schemes or even mixed strategies. In the following table, the most important interpolation schemes described in the previous sections are compared with each other using various criteria (industrial use, implementation complexity, possible use for MIMO systems etc.).

[28] Another interesting feature of the PID controller is that it includes integral action that ensures proper reference tracking; feature that is very important and very often a problem with other interpolation structures that do not necessarily provide a correct trim input for non-synthesis operating points (see Section 1.3.3 for more details)

[29] The PID controller may be seen as a controller with transfer function $K(s) = \frac{K_{\mathrm{d}}s^2 + K_{\mathrm{p}}s + K_{\mathrm{i}}}{s}$ or even in ZPK form. That is why a *Gain Blending* terminology does not really define a separate interpolation strategy.

[30] Both articles being part of this thesis are considered in Part II of this manuscript.

Table 1.1: Comparison of interpolation methods.

Features → *Methods* ↓	Industrial Spread	Computational Complexity	MIMO Use	Stability-preserving Extensions	Limitations on Controller Order - Structure	Signal Continuity Interpolation Coherence
Controller Switching	☺	☺	☺	☺☹	☺	☹
Controller Blending	☹	☺	☺	☹	☺	☹
ZPK Interpolation	☺☹	☺☹	☺☹	☹	☹	☺
TF Coefficient Interpolation	☺☹	☺☹	☺☹	☹	☹	☹
SS Matrix Interpolation	☹	☹	☺	☺	☹	☹
Observer-based Interpolation	☺☹	☹	☺	☺	☹	☺☹
Gain Blending	☺	☺	☺☹	☹	☺☹	☺

1.3.3 Stability-preserving Methods

The interpolation procedure used for the construction of the global gain-scheduled controller is crucial since it may cause instability to the closed loop system, even when the LTI controllers are designed to assure stability around the synthesis points. In this section some results concerning the analysis of methods that assure a degree of stability for the gain-scheduled plant are presented.

1.3.3.1 The Origins

A thorough analysis of gain-scheduled control systems lacked in the bibliography for many years, even though many real-world systems used this attractive control tool since the 1950's. Most theoretic work focused on general stability theory for feedback time-varying systems and the connection between theory and practice was not clear. The first systematic work on this subject appears in the bibliography with the PhD thesis of J. S. Shamma (see [119]) in the late 80's. The author considers three major types of gain-scheduled systems: a parameter-dependent linear plant (LPV) scheduling on its time-varying parameters, a nonlinear plant (rendered LPV by linearization) scheduled on a reference trajectory or on its output. In the first case, the author considers LPV systems of the following form[31]:

$$\mathcal{S}_{\text{LPV}}: \quad \begin{aligned} \dot{x} &= \mathbf{A}(\varrho)x + \mathbf{B}(\varrho)u \\ y &= \mathbf{C}(\varrho)x + \mathbf{D}(\varrho)u \end{aligned} \tag{1.112}$$

and it is argued that closed loop stability and the good properties of the feedback loop, around the family of operating points for which LTI controllers have been designed, may be retained provided that the parameter vector ϱ *varies slowly* in some operating domain $\mathbf{\Gamma}$. The overall analysis is rather complicated and conservative but gives for the first time sufficient conditions for the good behavior of a gain-scheduled control system.

In the second case when scheduling on a reference trajectory $y_{\text{ref}}(t)$ (that may or may not be known beforehand), the plant's output y should follow $y_{\text{ref}}(t)$. The author shows that the closed loop system is stable if the controller designed *for all* frozen-values of time (representing distinct values of the reference trajectory and thus distinct LTI snapshots of the LPV plant in Eq. 1.112) provides robust stability/performance *and* the reference trajectory *varies slowly*.

Finally in the third case, a more realistic case is considered where the gain-scheduled controller is updated (using state space matrix interpolation) with the output of the plant in order to ensure following of a reference output value. Again slowness conditions are imposed as well as the notion of 'capturing the plant's nonlinearities' with the scheduling variable, meaning that unmodeled dynamics of the plant should be relatively small. This work has led to two important, yet not so easily exploited, reference papers on the subject (see [120] and [121]).

[31] This system may in fact represent linearized dynamics of a nonlinear system with x being in fact $x_\delta = x - x_{\text{eq}}(\varrho)$ and the same for the input and output vector.

Several important points of the gain scheduling practice relevant to the two aforementioned publications are developed in more detail in [122]. These points include loss of stability and non-minimum phase properties of a gain-scheduled plant for rapid parameter variations; in addition real-world examples are given.

Another very interesting approach for the stability analysis of gain-scheduled systems is found in [118]. In this important paper that was published in the early nineties, right after the pioneering work of J.S. Shamma, the important problem of state feedback gain-scheduled regulation is considered. More precisely, for the LPV dynamics of Eq. 1.112[32], the following state feedback control law is considered (see Fig. 1.12):

$$u = -\mathbf{K}(\varrho)x + v \tag{1.113}$$

The state feedback matrix $\mathbf{K}$ is computed at a finite number of points $[\varrho_1, \ldots, \varrho_k]$ so that the eigenvalues of the corresponding frozen closed loop systems $\mathbf{A}(\varrho_j) - \mathbf{B}(\varrho_j)\mathbf{K}(\varrho_j)$ induce exponential stability. The choice of the synthesis points is done in such a way that the closed loop *scheduled* systems $\mathbf{A}(\varrho) - \mathbf{B}(\varrho)\mathbf{K}(\varrho)$ are also exponential stable and their eigenvalues are *close enough* to the frozen designs *if* the gain matrices are interpolated linearly between the synthesis points. In addition, the gain-scheduled loop remains stable *only if* the scheduling vector derivative is smaller to a certain amount. So once again, the notion of slowness in the time-varying parameter is inferred. Finally, conditions and ways of computing the state feedback matrix, so that the bounds on the parameter variation rate may be made arbitrarily high without causing instability, are devised.

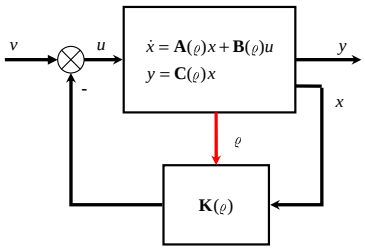


Figure 1.12: State feedback gain-scheduling.

The initial period on gain scheduling was also marked by a paper (see [113]) considering the problem of appropriate realization of a gain scheduling controller. This is conformable to the discussion of Section 1.3.1 where the five steps of gain scheduling control were detailed and is relevant with the final step of controller implementation. This premature paper was the beginning of a series of papers concerning this important issue as it will be detailed in the next subsection and triggered a controversy in the scientific community concerning classical and velocity-based gain scheduling (details for the latter are given in the next chapter).

[32] In the paper strictly proper dynamics are in fact assumed.

1.3.3.2 Mature Era

The mature era starts with the famous paper of R.A. Nichols, R.T. Reichert and W.J. Rugh (see [103]) for the gain-scheduled autopilot of an Air-to-air missile and is considered as a benchmark paper. It develops in fact ideas in the controller realization found in the previous cited paper [113]. However the full theoretical results are given in the important paper of D.A. Lawrence and W.J. Rugh in 1995 (see [79]).

`Generic nonlinear controller`

In this paper, the authors establish some very useful conditions for the gain scheduling controller realization to provide correct trim control *and* avoid the famous *coupling terms* that stem from the gain-scheduling practice. To illustrate this fact, suppose a nonlinear gain-scheduled controller to be written in the following *generic* form:

$$\dot{x}_k(t) = \mathbf{f}_k[x_k(t), x_i(t), \zeta(t), y(t), w(t), r(t)] \tag{1.114}$$
$$\dot{x}_i(t) = \mathbf{f}_i[x_k(t), \zeta(t), y(t), w(t), r(t)] \tag{1.115}$$
$$u(t) = \mathbf{f}_u[x_k(t), x_i(t), \zeta(t), y(t), w(t), r(t)] \tag{1.116}$$

where $\mathbf{f}_k, \mathbf{f}_i, \mathbf{f}_u$ are the nonlinear functions of the dynamic control, integral-error and output portions of the controller respectively[33] and $x_k, x_i, u, y, w, r, \zeta$ are the states of the dynamic control & integral error, control input, measured output, measured external parameter (used for scheduling), reference input and tracking error vectors. The setup is a servo problem one and the goal is to minimize (ideally nullify) the tracking error when the time-varying scheduling vector ϱ, being a nonlinear function of w, r and y, is taking values in a set $\mathbf{\Gamma}$[34].

`Controller nonlinear functions`

The realization of the nonlinear gain-scheduled controller functions that is proposed by the authors has the following form:

$$\begin{aligned}\mathbf{f}_k =& \mathbf{A}_{kk}(\varrho)[x_k - x_{k,\mathrm{eq}}(\varrho)] + \mathbf{A}_{ki}(\varrho)[x_i - x_{i,\mathrm{eq}}(\varrho)] + \\ & \mathbf{B}_{k\zeta}(\varrho)[\zeta - \zeta_{\mathrm{eq}}(\varrho)] + \mathbf{B}_{ky}(\varrho)[y - y_{\mathrm{eq}}(\varrho)] + \\ & \mathbf{B}_{kw}(\varrho)[w - w_{\mathrm{eq}}(\varrho)] + \mathbf{B}_{kr}(\varrho)[r - r_{\mathrm{eq}}(\varrho)]\end{aligned} \tag{1.117}$$

$$\begin{aligned}\mathbf{f}_i =& \mathbf{A}_{ik}(\varrho)[x_k - x_{k,\mathrm{eq}}(\varrho)] + \\ & \mathbf{B}_{i\zeta}(\varrho)[\zeta - \zeta_{\mathrm{eq}}(\varrho)] + \mathbf{B}_{iy}(\varrho)[y - y_{\mathrm{eq}}(\varrho)] + \\ & \mathbf{B}_{iw}(\varrho)[w - w_{\mathrm{eq}}(\varrho)] + \mathbf{B}_{ir}(\varrho)[r - r_{\mathrm{eq}}(\varrho)]\end{aligned} \tag{1.118}$$

$$\begin{aligned}\mathbf{f}_u =& \mathbf{C}_{uk}(\varrho)[x_k - x_{k,\mathrm{eq}}(\varrho)] + \mathbf{C}_{ui}(\varrho)[x_i - x_{i,\mathrm{eq}}(\varrho)] \\ & \mathbf{D}_{u\zeta}(\varrho)[\zeta - \zeta_{\mathrm{eq}}(\varrho)] + \mathbf{D}_{uy}(\varrho)[y - y_{\mathrm{eq}}(\varrho)] + \\ & \mathbf{D}_{uw}(\varrho)[w - w_{\mathrm{eq}}(\varrho)] + \mathbf{D}_{ur}(\varrho)[r - r_{\mathrm{eq}}(\varrho)].\end{aligned} \tag{1.119}$$

[33] Note that the integral part is particularly important as it ensures appropriate error tracking when the scheduling vector is varying.

[34] This setup is rather complicated and may be simplified for particular applications but encompasses most cases.

Now the above equations[35] have a linear form with respect to deviations from equilibrium values and should provide the appropriate control input (trim control) for equilibrium operation. In addition they linearize to an LTI controller with matrices $\mathbf{A}_{kk}(\varrho_{\text{eq}}), \ldots, \mathbf{D}_{ur}(\varrho_{\text{eq}})$ for *fixed* values of the scheduling vector that may be designed for a collection of equilibrium points in the operating domain $\mathbf{\Gamma}$ of the system.

Coupling terms

The gain-scheduled controller however has to update the control input as a function of the scheduling vector and thus one should consider its behavior when $\varrho = \varrho(t)$ is not constant. Indeed, in this case *the scheduling vector is function of w, r, y and when linearizing Eqs. 1.117-1.119 there appear additional terms besides the ones corresponding to the linear controllers designed for constant equilibrium point operation.* These terms are the famous **hidden coupling terms**, they are time-varying and add up to the overall system dynamics causing problems to the closed loop operation. It turns out from the analysis in [79] that the inclusion of the integral-error component in the nonlinear controller (see Eq. 1.115) guarantees with appropriate use the existence of a controller *without the coupling terms*. Examples for cases that this may be achieved can be found in [79] or even in the survey of [114].

The second series of articles concerning linearization-based gain scheduling starts to appear in the late 90's with the work of D.J. Stilwell. The first paper (see [132]) addresses a similar problem initially discussed in [118]. A parameter-dependent system $\mathcal{S}_{\text{pd}}$ as the one in Eq. 1.70 is considered, as well as an LPV one as in Eq. 1.72 stemming from linearization around equilibrium points dictated by the scheduling variable.

Stability covering condition

Given now this LPV system, an observer/state feedback control setup is assumed (see Section 1.3.2.6). The issue addressed is how to compute the control matrices $\mathbf{K}_{\text{c}}, \mathbf{K}_{\text{o}}$ as a function of the scheduling vector so as to guarantee stability of the LPV system, both for fixed and varying values of ϱ. It turns out that controllers are calculated at a number of points in the operating domain $\mathbf{\Gamma}$ of the system satisfying a certain *stability covering condition* and then are interpolated in a specific way that guarantees closed loop stability up to a certain extent of variation rate of the scheduling vector. Concerning this condition[36], suppose that a set $\Sigma(K_{\text{LTI}}) = [\mathbf{K}_{\text{c}}^1, \ldots, \mathbf{K}_{\text{c}}^k]$ of state feedback gains are computed for corresponding constant values of the scheduling vector[37] $[\varrho^1, \ldots, \varrho^k] \in \mathbf{\Gamma}$ in such a way that each gain stabilizes the LTI plant (i.e. $\mathbf{A}(\varrho^i)+\mathbf{B}(\varrho^i)\mathbf{K}_{\text{c}}^i$ is stable) obtained by a frozen LPV one for $\varrho = \varrho^i$. In addition each gain also stabilizes the LPV plant inside a region (i.e. $\mathbf{A}(\varrho) + \mathbf{B}(\varrho)\mathbf{K}_{\text{c}}^i$ is stable) Γ^i with $\varrho^i \in \Gamma^i$. *If* $\mathbf{\Gamma} \subset \bigcup_{i=1}^{k} \Gamma^i$ (their intersection is not necessarily empty) then the family of gains is said to satisfy the *stability covering condition*.

[35] The 'eq' notation means equilibrium operation for fixed values of the scheduling vector.

[36] Pure state feedback is assumed since the analysis for the full observer-based state feedback proceeds in an analogous manner.

[37] Suppose that $\varrho \in \mathbb{R}^1$ for simplicity.

This condition means in brief that the LPV plant must be stabilized at (possibly mutually overlapping) regions, whose union overbounds the operating region of the plant, using a single feedback gain per region. Of course at overlapping regions the LPV system may be stabilized by more than one gain. See for example Fig. 1.13 where a two dimensional scheduling vector is considered and three overlapping scheduling regions (light shading) for each one of the three synthesis points (red stars) inside the rectangular operating domain $\mathbf{\Gamma}$ of the plant. Three regions (dark shaded) exhibit overlapping of two controllers whereas one (small central blue region) exhibits overlapping of all controllers.

Returning to the one dimensional scheduling vector discussion, the global gain-scheduled controller will apply only one stabilizing LTI controller (robust functioning) for non-overlapping regions whereas in overlapping ones a gain blending approach is used that guarantees stability and control signal continuity. For more details on this particular type of interpolation, refer to [132], pp.1227, Theorem II.2.

`SS & NCF interpolation`

Another paper from the same authors (see [133] or equally [131]) generalizes the results of the previous paper [132] for two other cases. The first is *state space matrix interpolation* (as in Section 1.3.2.5) and the second one is a particular type of interpolation using a *normalized coprime factor* development of the controllers. As it has been already noted, the first method has several computational and other disadvantages. The second one has not received practically any attention in the scientific communities and it would may be interesting in a $\mathscr{H}_\infty$ loop-shaping control context.

`J-Q method`

Finally a very interesting theoretically justified interpolation approach uses LTI controllers of arbitrary structure (but of the same order) that may all be parameterized by a *single* dynamic system $\mathcal{J}$ and different stable Youla parameters $\mathcal{Q}^i$ [38]. The method uses once again the stability covering condition and interpolates the state space matrices of the Youla parameter in order to obtain a gain-scheduled controller. This article (see [129] or equally [128]) also offers a missile autopilot example in order to justify the practicality of the approach.

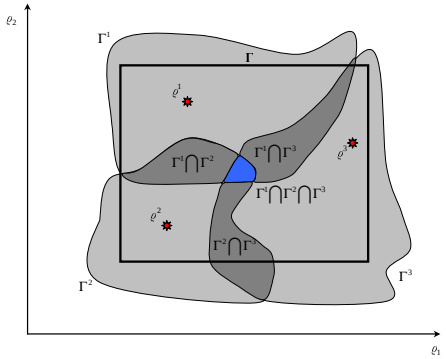


Figure 1.13: Stability covering regions.

[38] See also Section 1.3.2.6.

1.3.3.3 Modern Approaches

The modern era on gain scheduling seems to have started by the two surveys on the subject of gain scheduling in general published the same year (see [88, 114]). These surveys cover most gain-scheduled control schemes (linearization-based, LPV-based, fuzzy etc.) known to the scientific community and offer some good insights on how the gain scheduling research context should be defined in the future. However, one can distinguish a bias to velocity-based gain scheduling (see Chapter 2) in the first, and to linearization-based gain scheduling (with the appropriate controller realizations detailed in Section 1.3.3.2) in the second, mostly due to the corresponding to either methods work, of each pair of authors. Despite this fact, these survey should be taken as a landmark for future work.

A piece of work that should be noted is the extension of the one by D.J. Stilwell and D.A. Lawrence detailed in the previous section on sampled-data systems and it may be found in [130]. In the paper it is shown that certain linearization properties of a continuous time gain-scheduled plant *carry on* to the sampled data one, a fact that is important for real world systems.

Finally, a work that stands out from the rest in the gain-scheduling control community can be found in [40] where it is shown that the control objectives of a gain-scheduled controller can be expressed as the weighted incremental norm minimization of a nonlinear operator. However this method has yet to prove its effectiveness in practical situations.

1.4 Operating Domain Issues

In this section some tools used for controller interpolation will be presented. It has been already remarked that the 2^{nd} and 3^{rd} steps of the linearization-based gain scheduling procedure involve linearization of the nonlinear plant at a certain set of equilibrium points inside its operating domain $\mathbf{\Gamma}$. These points emerge from constant (or frozen-time) values of the scheduling vector $\varrho(t)$ and the LTI systems obtained are used in order to obtain a set of LTI controllers $\Sigma(K_{\text{LTI}})$. Then the LTI controllers are somehow blended to form a gain-scheduled controller. The latter is done by dividing the operating region $\mathbf{\Gamma}$ to appropriate scheduling regions Γ^i considering a number of neighbor controllers being placed at the edges of these regions. Until now, the issues of both choosing these points in a systematic way and forming the regions have been avoided.

Operating domain gridding

Given that very often the operating domain $\mathbf{\Gamma}$ of a system may be considered/rendered convex, an equidistant gridding is performed (see Fig. 1.11) with no idea on the density of the considered points. As a result, either only region corners are considered, or the gridding is performed in a trial-error manner until satisfactory performance is obtained[39].

[39] An exception is the work found in [132] but it remains rather conservative in terms of the maximum of the scheduling vector rate of variation.

Thus, the designer is not sure if he has considered too many or too few points in his study. Additionally, it is not obligatory that the points be taken equidistantly or even in a rectangular gridding[40]. This problem is addressed for the first time in this thesis using appropriate operating point selection algorithms as it will be detailed in Chapter 5.

The result of these algorithms are a number of operating points dispersed in an *non-uniform* way on the operating domain $\mathbf{\Gamma}$. Thus, for implementation purposes, a way to define scheduling regions Γ^i should be found. It turns out that the rectangular ones are nor redundant, given that a plane may be defined using only three points and not four, nor the interpolation is straightforward since the operating points will not be in general aligned. As a result, the more efficient way should be to triangulate the operating domain of the system and interpolate the controllers in triads (see Fig. 1.14).

`Triangulation`

An efficient way to triangulate a convex polygonal domain is the famous *Delaunay Triangulation*. A triangulation $\triangle$ of a convex domain $\mathbf{\Gamma}$ is a Delaunay triangulation provided that for every triangle Γ^j in $\triangle$, there is no vertex $\varrho^i \notin \Gamma^j$[41] inside the circumcircle around Γ^j. An extension for non-convex operating domains can be found in [78], pp. 109-110. A related notion are the *Voronoi Diagrams* which are polygonal domains formed around each vertex ϱ^i and define a certain zone of influence around every ϱ^i.

Technical Note. In the context of Gain Scheduling, given a set of points, one can use the MATLAB function `TRI=delaunay`(ϱ_x, ϱ_y), where ϱ_x, ϱ_y are the one to one ordered vectors of the x and y coordinates of all points, in order to obtain a Delaunay triangulation. Then, for any value $\varrho(t)$ of the scheduling vector the function `tsearch` (taking also as inputs all the triangle edges triplets `TRI`) gives the corresponding triangle (or else scheduling region Γ^j). Thus, controller interpolation may be performed by using the LTI controllers stored for every edge of the triangle chosen (see Figure 1.15 for an example of Delaunay triangulation and Voronoi tesselation).

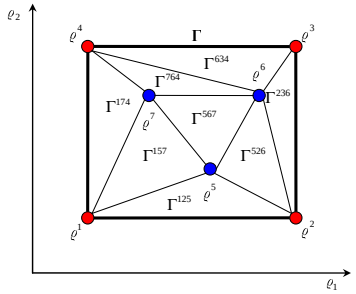
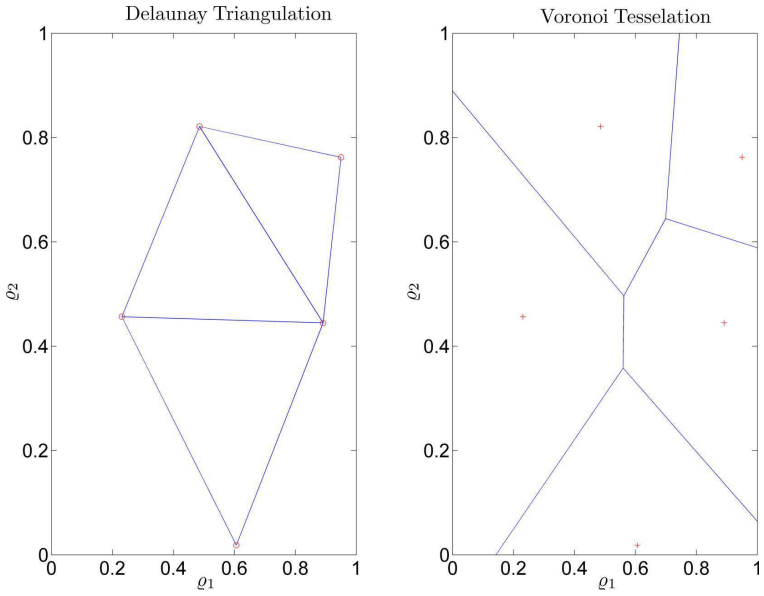


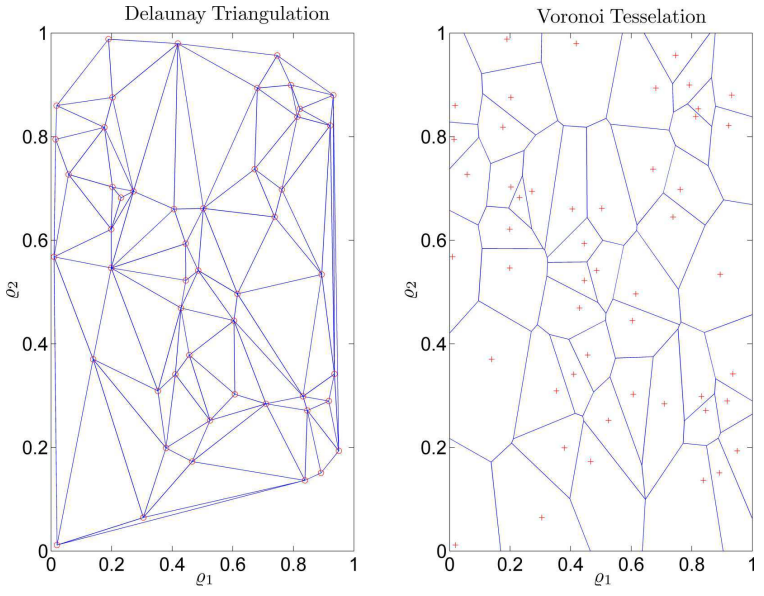
Figure 1.14: Operating domain triangulation.

[40] Planar operating regions are considered for simplicity.

[41] Here it is supposed that the vertices are dictated by constant values of the scheduling vector ϱ^i on the operating domain $\mathbf{\Gamma}$ of the system.



(a) *Five points*



(b) *Fifty points*

Figure 1.15: Delaunay triangulation and Voronoi Tesselation.

Chapter 2

Modern Gain Scheduling

Overview

This chapter offers a relatively short overview of alternative gain scheduling methods that have been proposed the last years. These methods are the LPV, velocity-based and neural/fuzzy gain scheduling approaches and they play also a significant role on the subject, being extensively used on real-world systems. Since these methods have not been given further consideration in this work, the material referenced here is not exhaustive and is presented as a bibliographic complement to Chapter 1 detailing linearization-based gain scheduling techniques.

Chapter contents

2.1 LPV Gain Scheduling

Motivation

The LPV (Linear Parameter Varying) gain scheduling approach is the major alternative to the linearization-based one presented in Chapter 1. This method has some advantages over the latter since it offers more serious stability guaranties for the gain-scheduled system. Even though it may also be used for purely LPV plants resulting from linearization of a nonlinear parameter-dependent system, it is most interesting when it is applied to an over-bounding q-LPV reformulation of the nonlinear system. This method can also incorporate bounds on the scheduling vector rates (and thus reducing conservatism) using parameter-dependent Lyapunov functions. However in some cases it may be rather conservative due to this reformulation incorporating redundant trajectories that may not belong to the nonlinear system and in addition, it does not offer feasibility guaranties for the existence of the gain-scheduled controller.

During the last fifteen years there has been a true wealth in the bibliography on these methods and several issues continue to be treated; the following analysis attempts to give only some of the basic results concerning this approach, being mainly divided in two major categories: polytopic and LFT gain scheduling.

2.1.1 Polytopic Approach

Control goal

The polytopic approach in LPV gain scheduling is very common in the scientific community and has been extensively studied; the results presented here are taken mainly from [13]. The main goal behind this approach is to calculate a gain-scheduled controller $K(s,\varrho)$ that will guarantee internal stability and in addition quadratic $\mathscr{H}_\infty$ performance γ on the performance vector ζ_∞ over external disturbances w and for all admissible values of the scheduling vector components ϱ_i (see Fig. 2.1).

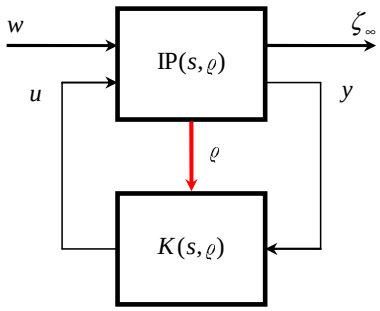


Figure 2.1: Polytopic gain-scheduling structure.

The class of systems considered have an affine dependence of their state space matrices on the scheduling vector components and in addition the scheduling vector takes values inside a convex polytope[1].

[1] As a result the state-space matrices take also values inside a convex polytope.

Searching for a (single) Lyapunov matrix $\mathbf{X} = \mathbf{X}^T > 0$ that satisfies the BRL[2] ensuring quadratic $\mathscr{H}_\infty$ performance for a generic (not polytopic) LPV (or q-LPV) plant poses an *infinite* number of constraints; however for the polytopic case, the problem is tractable and reduces to a *finite* number of constraints posed for each vertex of the polytope. This is the result of the famous *vertex property* stating that the following two arguments are equivalent:

Vertex property

- The polytopic LPV system is stable with quadratic $\mathscr{H}_\infty$ performance γ.
- There exist a single matrix $\mathbf{X} = \mathbf{X}^T > 0$ satisfying the collection of LMI's[3]:

$$\mathscr{B}_{\left[\mathbf{A}^i_{\rm cl},\mathbf{B}^i_{\rm cl},\mathbf{C}^i_{\rm cl},\mathbf{D}^i_{\rm cl}\right]}(\mathbf{X},\gamma) < 0, \quad i = 1,\ldots,r.$$

Controller computation

The gain-scheduled LPV controller sought will also be of polytopic form and the LPV synthesis problem is primarily to find a common Lyapunov matrix for all vertices; this is done considering the corresponding set of the classic LMI feasibility conditions, given also in Section 3.3.3.2 of this report. The LMI's are in fact solved for two matrices $\mathbf{R}, \mathbf{S}$ and the Lyapunov matrix is finally constructed solving some matrix equations.

Once the feasibility conditions are met and the Lyapunov matric $\mathbf{X}$ computed, all vertex controllers:

$$\mathbf{\Omega_i} = \begin{pmatrix} \mathbf{A}^i_k & \mathbf{B}^i_k \\ \mathbf{C}^i_k & \mathbf{D}^i_k \end{pmatrix} \tag{2.1}$$

may be sequentially computed either by solving the BRL's either by the same convex optimization algorithms or symbolically. The final LPV controller will be of the form

$$K(\varrho): \quad \begin{aligned} \dot{x}_k &= \mathbf{A}_k(\varrho)x_k + \mathbf{B}_k(\varrho)y \\ u &= \mathbf{C}_k(\varrho)x_k + \mathbf{D}_k(\varrho)y \end{aligned} \tag{2.2}$$

and its matrices computed as a convex combination of the vertex controllers, using the current/measured value of the scheduling vector $\varrho(t)$.

Additional work

For a similar treatment of the problem see [18]; being one of the first works on the subject. A multi-objective approach treating simultaneously $\mathscr{H}_\infty$ and $\mathscr{H}_2$ performances, passivity, asymptotic disturbance rejection, time-domain constraints and constraints on the closed loop location for different channels of the closed loop system with a common Lyapunov function is given in [115]. A good reference on advanced gain-scheduling techniques in order to reduce computational burden is proposed in [9]. Parameter-dependent controllers if the scheduling parameters are real are using a skew-symmetric technique is presented in [117] whereas some work using the elimination lemma is proposed in [126]. Finally, other more recent approaches may be found in [5, 127, 154].

[2]Bounded Real Lemma.

[3]The symbol '$\mathscr{B}$' denotes BRL and $\mathbf{A}^i_{\rm cl}, \mathbf{B}^i_{\rm cl}, \mathbf{C}^i_{\rm cl}, \mathbf{D}^i_{\rm cl}$ are the closed loop standard model matrices at each of the r vertices of the polytope (see also [13], Eq. 29).

2.1.2 LFT Approach

The LFT (Linear Fractional Transformation) gain scheduling approach is an important alternative to the standard Lyapunov-based LPV one. Perhaps the one of the first and most widely known attempts to deal with this formulation can be found in [12] and the material presented in this section is mainly drawn from there.

`LFT vs. Polytopic`

This method is different from the classical polytopic one since it does not require an affine dependence of the system's state space matrices on the time-varying parameters ϱ_i since they enter both the plant & controller dynamics in a specific way. In addition, concerning implementation, the parameter vector need not be expressed as the convex combination of its vertex values. However, this method may become cumbersome when the plant's LFT form needs to be computed since it is highly non-unique and the designer may need repeated parameter blocks, thus increasing the order of the system.

`Modeling`

The main idea behind this method is to re-cast the initial gain scheduling problem as one of robust performance in the face of structured uncertainty using small gain theory. In the beginning the designer performs the following procedure: starting from a generic nonlinear parameter-dependent plant $\mathcal{S}_{\text{pd}}$ as in Eq. 1.22 (with $\varrho(t)$ being the on-line measured parameter/scheduling vector) obtains a q-LPV re-formulated system $\mathcal{S}_{q-\text{LPV}}$ (see Eq. 1.24) or an LPV one using Jacobian linearization (see Eqs. 1.26-1.29). From this point, an equivalent u-LFT formulation of this system is calculated as a specific connection of a block-diagonal parameter block Θ, containing the measured parameters (see Eqs. 1.30-1.31 and Fig. 1.5)[4]. This u-LFT connection is written as:

$$\begin{bmatrix} \zeta_\infty \\ y \end{bmatrix} = \mathcal{F}_u\big(\mathbb{P}, \Theta\big) \begin{bmatrix} w \\ u \end{bmatrix} \tag{2.3}$$

where ζ, y, w, u are once again the performance, controller input, perturbation and controller output vectors respectively.

`Control goal`

The gain-scheduling problem now is to find a time-varying controller having a similar l-LFT formulation as the plant; thus find a control input u that satisfies:

$$u = \mathcal{F}_l(K, \Theta). \tag{2.4}$$

This specific structure of the 'time-varying' uncertainty interconnection between the system, the controller and the scheduling vector block containing the varying parameters is depicted in Fig. 2.1. Alternatively, the overall feedback connection may be expressed as

$$T(\mathbb{P}, K, \Theta) = \mathcal{F}_l\big(\mathcal{F}_u(\mathbb{P}, \Theta), \mathcal{F}_l(K, \Theta)\big) \tag{2.5}$$

and the corresponding $\mathscr{H}_\infty$ control problem may now be posed as follows:

[4]The parameters are supposed to be confined to a ball with radius γ^{-1}, with γ being the corresponding $\mathscr{H}_\infty$ performance level; however a re-scaling on the input perturbations may be performed in order to permit larger variations.

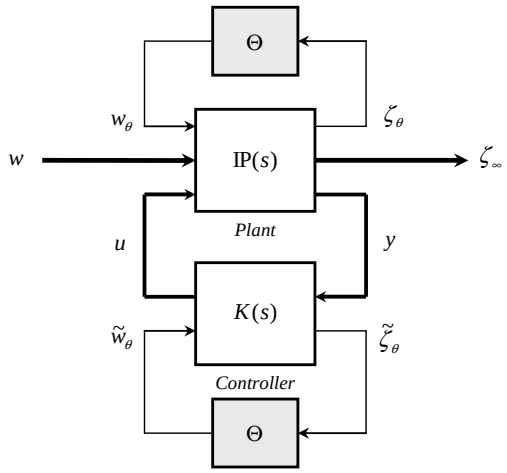


Figure 2.2: LFT gain-scheduling structure.

Find a structure (i.e. state space matrices) for the controller $K(s)$ such that the gain-scheduled LPV controller $\mathcal{F}_l(K,\Theta)$ satisfies the following properties:

- The closed loop system given by Eq. 2.5 is internally stable for all parameter trajectories $\varrho_i(t)$ that satisfy the scaling $\gamma^2\Theta^T\Theta \leq 1$.
- The induced $\mathcal{L}_2$ norm of the closed loop system satisfies:

$$\max_{\|\Theta\|_\infty \leq \gamma^{-1}} \|T(\mathbb{P},K,\Theta)\|_\infty \leq \gamma.$$

Now the closed loop system may be written as the u-LFT of the l-LFT of a specific augmented system $\mathbb{P}_{aug}(s)$ (containing $\mathbb{P}(s)$) with the unknown controller $K(s)$, and a 2×2 block diagonal matrix containing all measurable time-varying parameters:

$$T(\mathbb{P},K,\Theta) = \mathcal{F}_u\left(\mathcal{F}_l\big(\mathbb{P}_{aug},K\big),\begin{pmatrix}\Theta & 0\\ 0 & \Theta\end{pmatrix}\right). \tag{2.6}$$

`Controller computation`

From the aforementioned equation it may be deduced that the original gain scheduling problem may be viewed as a classical robust performance problem in the face of the block-repeated uncertainty and sufficient conditions for solvability are provided by the small gain theory. This results to a particular case of the general $\mathscr{H}_\infty$ synthesis problem which is rather easily transformed to a set of LMI existence conditions that may be solved using interior point algorithms. Finally, the controller matrices are computed by the corresponding to the $\mathscr{H}_\infty$ problem scaled BRL (see also Section 3.1.3).

Globally, a nice introduction to the subject may be found in the survey papers [88] and [114], whereas more modern results concerning this method are given in [14, 67, 94, 148, 151, 153].

2.2 Velocity-based Gain Scheduling

The velocity-based gain scheduling method belongs to the class of methods characterized by the 'divide and conquer' design presented in Chapter 1. It has been already remarked that (see Section 1.3.2.2) gain-scheduled controller realizations ensuring correct trim control and matching of the linearization of the gain-scheduled controller, at each equilibrium/synthesis point, with the corresponding member of the controller family designed at this point, may be designed.

`Motivation`

This property is often called *local linear equivalence* (see [88], §3.1b) but it does not treat the extent for which this equivalence is valid. A particular class of methods claiming to treat this subject was developed in the late 90's by D. J. Leith and W. E. Leithead with a series of articles and is called velocity-based implementation of the gain scheduling controllers, satisfying an *extended* local linear equivalence property. Given the fact that this method has not received the appropriate attention henceforth and due to the already mentioned exchange in [84], its potential remains yet to be proved.

This method differentiates itself from the classical first order series (or Jacobian) linearization theory used by every method mentioned so far, in order to obtain a family of linear systems computed at a corresponding family of equilibrium points of a nonlinear parameter-dependent system. The most important point here is that when considering linearized models such as the ones in Eq. 1.26, the notation $x_\delta = x - x_{\text{eq}}$ is abused since this difference quantity only *tends* to describe the true state difference and only under heavy assumptions on the corresponding linear system's range of operation, on the input rate etc.

`Modeling`

In addition, when a nonlinear system is not confined to a vicinity of an equilibrium point, the linear approximation of Eq. 1.26 does not offer an accurate approximation of the nonlinear system dynamics. To overcome this difficulty, Leith & Leithead offer an alternative linearization around a generic operating point (x_1, u_1) of the system (see [85], §3.2, 3.3). The linearized model thus is[5]:

$$\begin{aligned} \dot{\hat{x}}_\delta &= \nabla_x \mathbf{f}(x_1, u_1)\hat{x}_\delta + \nabla_u \mathbf{f}(x_1, u_1)u_\delta + \mathbf{f}(x_1, u_1) \\ \hat{y}_\delta &= \nabla_x \mathbf{h}(x_1, u_1)\hat{x}_\delta + \nabla_u \mathbf{h}(x_1, u_1)u_\delta \end{aligned} \tag{2.7}$$

with $u_\delta = u - u_1, \hat{y}_\delta = \hat{y} - y_1, \hat{x}_\delta = \hat{x} - x_1$ and $\dot{\hat{x}}_\delta = \dot{\hat{x}}$, with the neighborhoods around the operating point being sufficiently small. The main difference with the Jacobian linearization is that there is a nonzero term in the first equation[6] thus making the approximation model nonlinear. Re-arranging the previous equation using the aforementioned transformations, the following approximation of the system's state is obtained:

$$\begin{aligned} \dot{\hat{x}} &= \nabla_x \mathbf{f}(x_1, u_1)\hat{x} + \nabla_u \mathbf{f}(x_1, u_1)u + \mathbf{f}(x_1, u_1) - \nabla_x \mathbf{f}(x_1, u_1)x_1 - \nabla_u \mathbf{f}(x_1, u_1)u_1 \\ \hat{y} &= \nabla_x \mathbf{h}(x_1, u_1)\hat{x} + \nabla_u \mathbf{h}(x_1, u_1)u + \mathbf{h}(x_1, u_1) - \nabla_x \mathbf{h}(x_1, u_1)x_1 - \nabla_u \mathbf{h}(x_1, u_1)u_1 \end{aligned} \tag{2.8}$$

[5] Dependence on the scheduling vector is for now omitted for simplicity.

[6] Note that in the Jacobian case this term vanishes since this point is an equilibrium one.

Velocity form

Now by differentiating Eq. 2.8 and considering the appropriate initial conditions for the state of the system, one gets the following *velocity form* that is now totally linear:

$$\begin{aligned} \dot{\hat{x}} &= \hat{w} \\ \dot{\hat{w}} &= \nabla_x \mathbf{f}(x_1, u_1)\hat{w} + \nabla_u \mathbf{f}(x_1, u_1)\dot{u} \\ \dot{\hat{y}} &= \nabla_x \mathbf{h}(x_1, u_1)\hat{w} + \nabla_u \mathbf{h}(x_1, u_1)\dot{u}. \end{aligned} \tag{2.9}$$

From the analysis found in [85], it turns out that the aforementioned linear system yields an approximation of the initial nonlinear system's dynamics during a certain time interval for the operating point considered, accurate now to a *second* order (instead of a first one in the Jacobian case). It is evident that additional linearizations are needed for subsequent operating points, when the approximation error starts to increase.

Controller computation

Suppose now that the initial nonlinear system is in fact dependent on the scheduling vector with $\varrho = \varrho(x, u)$; then the approximation performed in Eq. 2.9 is now *scheduling vector-dependent*. Based on this modeling, encapsulating an approximation of the nonlinear system for an arbitrary operating point, a gain scheduling procedure may be devised.

This is first done by calculating a family of specific velocity-based linearization type of controllers that achieves the performance requirements for the now linear velocity-based model of the nonlinear plant. Since this linear is smoothly parameterized by the scheduling vector ϱ, one needs to calculate an infinite number of controllers for every possible value of ϱ; however, the same strategy as with conventional gain scheduling may be used where controllers are designed only at a number of operating points.

The final gain-scheduled controller is obtained from the family of linear controllers by permitting the scheduling vector to vary with the operating point. A thorough treatment of the subject is clearly out of the scope of this work but one can refer to the series of papers [82, 83, 86] and additional details may be also found in [87].

Alternative method

An interesting approach that satisfies however only the local linear equivalence property has been given in [71]. This work once again considers the demand for an appropriate gain-scheduled controller implementation that preserves the input-output properties of the closed loop systems locally about each equilibrium point. The method and the control law proposed follow the so-called $\mathscr{D}$ procedure and use a particular form in order to construct the controller. Integral action is added at its input whereas some of its inputs are differentiated before actually given to the controller.

It is claimed that this scheme does not introduce any additional noise amplification at the relevant inputs and outputs of the linearized feedback system since all closed loop functions are preserved. However the issue of noise amplification inside the controller and how it impacts on the behavior of the nonlinear feedback system is not addressed.

2.3 Neural/Fuzzy Gain Scheduling

An unconventional method to construct gain-scheduled controllers has been developed by the fuzzy/neural community and applied on numerous cases, e.g. for flight control laws; for some recent examples see [70, 107, 141]. The material presented here is mainly from the survey [88] and from the tutorial-like paper [135] reviewing separately classical, fuzzy, neural, and neuro-fuzzy gain-scheduling.

`Modeling`

The first step towards the design of a fuzzy gain-scheduled controller is a representation of a nonlinear system as a blend of i local models of the form:

$$\dot{x} = \sum_i \bar{\mathbf{f}}_i(x,u)\mu_i(\sigma) \tag{2.10}$$

$$y = \sum_i \bar{\mathbf{h}}_i(x,u)\mu_i(\sigma) \tag{2.11}$$

The functions μ_i are the so-called *membership functions* used to blend these models, with $\sum_i \mu_i(\sigma) = 1$ and the quantity $\sigma = \sigma(x,u)$ shows the dependence of this blending on the state and the input. The interesting fact in the approach is that the blended models may be considered as affine local models:

$$\bar{\mathbf{f}}_i(x,u) = \alpha_i + \bar{\mathbf{A}}_i x + \bar{\mathbf{B}}_i u \tag{2.12}$$

$$\bar{\mathbf{h}}_i(x,u) = \beta_i + \bar{\mathbf{C}}_i x + \bar{\mathbf{D}}_i u \tag{2.13}$$

`Controller computation`

This blending representation can thus directly lead to a divide and conquer gain scheduling strategy since a local controller may be designed for a local model and then blended using the weighting functions, according to the quantity σ. This representation of the nonlinear system may be also considered in another context: each member of the local models family may be used only at a certain operating region of the system, leaving the blending occur at transition regions; however, with this approach problems occur concerning coupling terms with the derivatives of the membership functions.

`Features & comments`

The primary advantage of a fuzzy gain-scheduled controller is that the plant modeling may be done exploiting human expertise on particular systems where modeling using the physics laws of physics is not possible or does not lead to reliable results. However, this procedure of determining a fuzzy model may be time consuming and demanding extensive computer simulations to reassure the designer for the closeness of the fuzzy model to the real-world system.

This leads to neural network-based gain scheduling that utilizes the learning capabilities of a neural network so that the controller parameters are 'learned' without a detailed prior knowledge of the plant. This method also has drawbacks since a neural-network does not give much insight into the plant dynamics and its structure is not an easy to task to construct. Thus, combined schemes may be used that take advantage of each approaches benefits (see [135] and references therein).

Chapter 3

Control Theory for Gain Scheduling

Overview

A major advantage of gain scheduling control is that it provides *nonlinear* parameter-dependent systems a *nonlinear* time varying controller by using *linear* time invariant ones. It has been remarked that for real world applications the elegant and powerful results of the modern $\mathscr{H}_\infty$ control theory are particularly interesting for the synthesis of LTI controllers in contrast to other methods such as predictive and/or pure nonlinear control strategies that risk being overly complex and/or difficult to implement. In this work two $\mathscr{H}_\infty$ control structures were tested in order to provide the necessary LTI controllers needed for interpolation in the gain scheduling control context. This chapter offers a solid yet not exhaustive review of two of these methods: $\mathscr{H}_\infty$ dynamic output feedback with pole placement constraints and $\mathscr{H}_\infty$ dynamic and static loop shaping. In addition some rather standard results concerning full order state observers and Youla parametrization (in use with the first synthesis method) and an important system analysis tool called the *gap metric* (in use with the second synthesis method) are presented.

Chapter contents

3.1 $\mathscr{H}_\infty$ Control in LMI Regions

In this section some theoretical results concerning $\mathscr{H}_\infty$ control with pole placement constraints in LMI regions will be presented. The section starts with a classical analysis motivating the use of this powerful synthesis method for the computation of LTI controllers at the first benchmark example of Chapter 5. The subsequent sections give all the necessary results for a systematic treatment of this control problem with most of the material drawn from [27].

3.1.1 Motivation

2^{nd} order system analysis

Consider a SISO linear time invariant system $G(s)$ and a controller $K(s)$ in a standard closed loop control configuration (see Fig. 3.1a). The primary goal of classical control systems is to design the controller K so that the time response $y(t)$ to a step reference input $y_\mathrm{r}(t)$ has *good* properties. Many of these properties are dominated mostly by the location of the poles λ of the closed loop system $H(s)$ with:

$$H(s) \triangleq \frac{Y(s)}{Y_\mathrm{r}(s)} = \frac{G(s)K(s)}{1 + G(s)K(s)} \tag{3.1}$$

To quantify the influence of the pole location to the time response of the closed loop system $H(s)$, suppose that H is or may be approximated by a second order system (see Fig 3.1b), as is the case very often in practice, with:

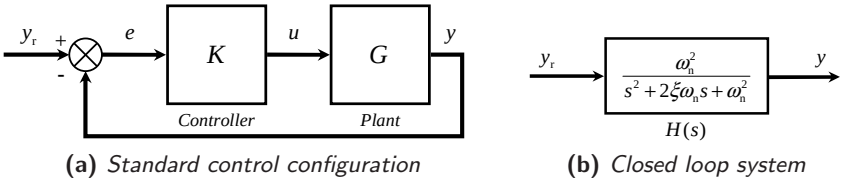
$$H(s) = \frac{\omega_\mathrm{n}^2}{s^2 + 2\xi\omega_\mathrm{n}s + \omega_\mathrm{n}^2} \tag{3.2}$$

The transfer function parameters ω_n and ξ are called *undamped natural frequency* and *damping ratio* of the poles $\lambda_{1,2}$ of H, being the roots of its denominator, with:

$$\lambda_{1,2} = -\xi\omega_\mathrm{n} \pm j\omega_\mathrm{n}\sqrt{1-\xi^2}. \tag{3.3}$$

Poles

The quantitative meaning of the two fundamental variables ω_n and ξ is related to the step response of H. The undamped natural frequency is the system's output oscillation frequency if its damping ratio is reduced to zero whereas the damping ratio is closely related to the overshoot experienced on the system's step response, given that the system is underdamped.



(a) *Standard control configuration* **(b)** *Closed loop system*

Figure 3.1: Basic analysis block diagrams.

Damping scenarios

The poles $\lambda_{1,2}$ depend on both ω_n and ξ (see Eq. 3.3), but it is the latter that characterizes the form of the step response $y(t)$. Four scenarios are considered for the damping ratio: $\xi = 0$ (non-damped), $0 < \xi < 1$ (underdamped), $\xi = 1$ (critically damped) and $\xi > 1$ (overdamped). The first and the third scenarios may be considered as limit cases of the second and the fourth ones.

In the non-damped case ($\xi = 0$) the closed loop poles are purely imaginary with $\lambda_{1,2}^{nd} = \pm j\omega_n$ and the time response is purely oscillatory whereas in the critically damped case ($\xi = 1$) the closed loop poles are purely real and negative with equal values $\lambda_{1,2}^{cd} = -\omega_n$. The system in the first case is said to be *conditionally stable* whereas in the second remains always *stable*. In the overdamped case ($\xi > 1$) the system demonstrates two distinct stable real poles with $\lambda_{1,2}^{od} = (-\xi \pm \sqrt{\xi^2 - 1})\omega_n$. For a constant undamped natural frequency, as the damping ratio increases the first stable pole goes to infinity whereas the second goes to zero. Thus, the time response of such as system becomes sluggish since it gets dominated by a slow stable eigenvalue. All three cases are not interesting for a control system for stability and/or speed reasons, so only the underdamped case is considered in the following analysis.

For an underdamped system $0 < \xi < 1$ its step response $y(t)$ and step tracking error $e(t) = y_r(t) - y(t)$ (see Fig. 3.1b) are computed using basic knowledge of ODE theory as (see [105], pp. 147-148)[1]:

$$y(t) = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin\left(\omega_d t + \arctan\frac{\sqrt{1-\xi^2}}{\xi}\right) \tag{3.4}$$

$$e(t) = e^{-\xi\omega_n t}\left(\cos\omega_d t + \frac{\xi}{\sqrt{1-\xi^2}}\sin\omega_d t\right). \tag{3.5}$$

The step response $y(t)$ of $H(s)$ for a given ω_n, presents different amounts of overshoot and oscillation around the desired reference trajectory $y_r(t)$ for different values of ξ (see Fig. 3.2a) whereas its settling speed for a given ξ is a function of ω_n (see Fig. 3.2b).

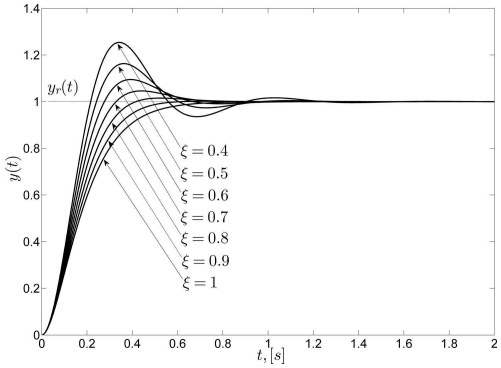
In order to characterize an LTI system in a more uniform way, several properties of its step time response $y(t)$ may be defined, depending only on the damping ratio ξ and undamped natural frequency ω_n. Some of these properties are the *rise time* t_r, *peak time* t_p, *settling time* t_s and *overshoot* M_p (see Fig. 3.2c) and may be easily calculated for a second order system as:

Step response properties

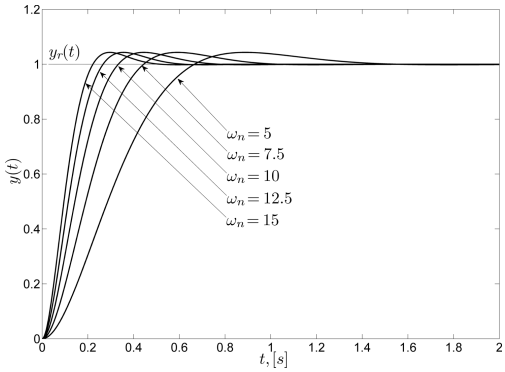
1. *Rise Time* t_r*:* It is usually defined as the time that the step response $y(t)$ takes to reach its 100% value for the first time. It may be computed from Eq. 3.4 by letting $y(t) = 1$:

$$t_r = \frac{\pi - \arctan\dfrac{\sqrt{1-\xi^2}}{\xi}}{\omega_n\sqrt{1-\xi^2}}. \tag{3.6}$$

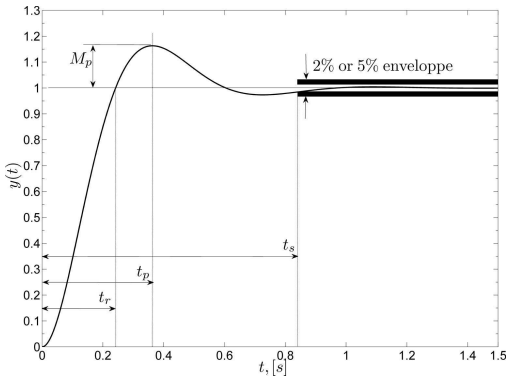
[1] The quantity $\omega_d = \omega_n\sqrt{1-\xi^2}$ is called the *damped* natural frequency.



(a) *Step responses (varying ξ)*



(b) *Step responses (varying ω_n)*



(c) *Step response characteristics*

Figure 3.2: Step response study - *underdamped case*.

2. *Peak Time* t_p*:* It is defined as the time that the step response $y(t)$ takes to reach its maximum value. It is computed by letting the derivative of $y(t)$ go to zero:

$$t_\mathrm{p} = \frac{\pi}{\omega_\mathrm{n}\sqrt{1-\xi^2}} = \frac{\pi}{\omega_\mathrm{d}}. \tag{3.7}$$

3. *Settling Time* t_s*:* It is defined as the time that the step response $y(t)$ takes to reach a 2% or 5% envelope around its steady state value $y(t_\infty)$. It is approximatively computed as:

$$t_\mathrm{s} = \frac{3}{\xi\omega_\mathrm{n}} \quad (5\% \text{ criterion}) \tag{3.8}$$

$$t_\mathrm{s} = \frac{4}{\xi\omega_\mathrm{n}} \quad (2\% \text{ criterion}) \tag{3.9}$$

4. *Maximum Overshoot* M_p*:* It is defined as the maximum positive percentage deviation (occurring at the peak time $t = t_\mathrm{p}$) of the step response $y(t)$. It is computed as:

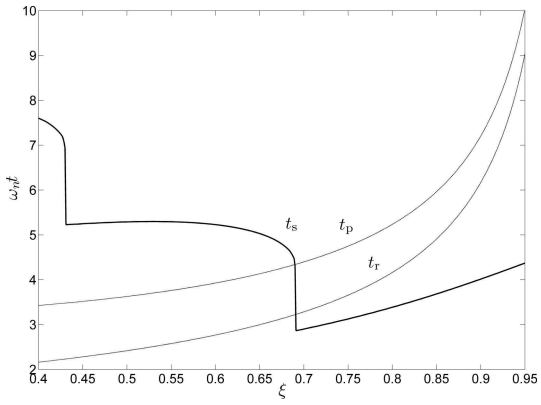
$$M_\mathrm{p} = \frac{y(t_\mathrm{p}) - y(t_\infty)}{y(t_\infty)} \cdot 100\% = e^{\frac{-\pi\xi}{\sqrt{1-\xi^2}}} \cdot 100\% \tag{3.10}$$

Pole placement discussion

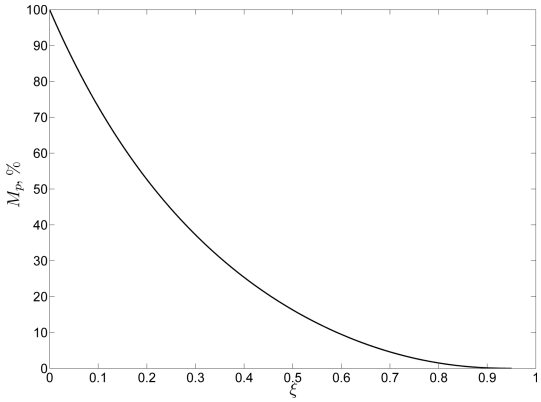
A control system should be able to provide satisfactory *response times* and *damping* for the plant under control. For a second order system with the simple form of Eq. 3.2, this is done by placing its poles $\lambda_{1,2}$ (see Eq. 3.3) to an appropriate location following two rules of thumb, as it has been implied in the preceding analysis: first the desired settling time t_s of the process is set by adjusting the undamped natural frequency ω_n and then an appropriate damping ratio ξ is chosen in order on the one hand avoid excessive overshoot, and on the other hand obtain a time response for the system that is not too sluggish.

The dependence of the rise, peak and settling times over the damping ratio and a given undamped natural frequency is shown in Fig. 3.3a[2]. Even though the rise and peak times augment monotonically with the damping ratio ξ, it does not happen the same with the settling time. It may be remarked that while the settling time is almost constant for medium values of the damping ratio $0.45 \leq \xi \leq 0.65$, it reaches a minimum for $\xi \simeq 0.69$ and then starts to rise almost linearly. The corresponding percentage overshoot M_p for this optimal value of the damping ratio is about 4.7% (see Fig. 3.3b). In practice, a damping ratio between 0.6 and 0.8 for the closed poles of a real-world system is considered satisfactory with the undamped natural frequency being chosen as a function of the specific bandwidth demanded from the control system.

[2]The figure shows the settling time for $\omega_\mathrm{n} = 1 rad/s$ and thus provides scaling for any ω_n.



(a) *Step response times*



(b) *Step response overshoot*

Figure 3.3: Step response characteristics.

3.1.2 LMI Regions

Motivation for eigenvalue clustering

From the analysis of the previous section it has been made clear that the transient behavior of a control system is dominated by the location of its closed loop poles. For a simple second order system as the one in Eq. 3.2, it is generally easy to obtain the desired closed loop dynamics by setting the damping ratio and undamped natural frequency to some desired values. For a higher order system there exist also solid methods for robust state/output feedback eigenvalue placement to an arbitrary accuracy (see for example [72] for details on the algorithm implemented in MATLAB® for state feedback eigenvalue placement).

Besides focusing on eigenvalue placement only, a control system could provide a control law that takes into account constraints over frequency domain aspects, robustness over external perturbations and parametric uncertainties. A good way to take into account all these requirements is the $\mathscr{H}_\infty$ robust control context with additional eigenvalue placement constraints. There exists an extensive literature over this general problem of *root clustering* (e.g. see [28, 56, 57]); here however the approach found in [27] will be preferred since the author believes that it gives the more general results on the subject. Having given the motivation why eigenvalue placement is so important in Section 3.1.1, this section presents some introductory material over the famous LMI regions.

3.1.2.1 Design Objectives

As pointed out in the previous analysis, an eigenvalue placement procedure could be very efficient for a control system. This procedure could be either a rather exact or one-to-one eigenvalue assignment to predefined locations, or a more general placement of the system's state space representation eigenvalues into convex sub-regions of the complex plane $P_{\mathbb{C}}$. The latter method is very appealing because it can be cast as an LMI convex optimization problem solvable by efficient algorithms.

$\mathcal{D}(\alpha, r, \vartheta)$ region

These regions may be vertical or horizontal strips, circles, parabolas or general conic sections on the complex plane. An LMI region used often in practice is the $\mathcal{D}(\alpha, r, \vartheta)$ performance-stability region of Fig. 3.4. This particular LMI region could define a useful design objective as it is the intersection of an α-stability vertical strip $\mathcal{D}_\alpha$ that provides a minimum decay rate α, a semi-circular region $\mathcal{D}_r$ imposing undamped natural frequency constraints and a triangular constraint region $\mathcal{D}_\vartheta$ that sets minimum damping on the closed loop eigenvalues. For any complex number $z = x + yj \in \mathbb{C}$ these regions are defined as:

$$\mathcal{D}_\alpha : \quad \mathcal{R}e\{z\} = x \leq -\alpha,\ \alpha > 0 \tag{3.11}$$

$$\mathcal{D}_r : \quad |z| \leq r,\ r > 0 \tag{3.12}$$

$$\mathcal{D}_\vartheta : \quad \tan\vartheta \cdot x \leq -|y|,\ 0 < \vartheta < \pi/2 \tag{3.13}$$

and

$$\mathcal{D}(\alpha, r, \vartheta) \triangleq \mathcal{D}_\alpha \cap \mathcal{D}_r \cap \mathcal{D}_\vartheta. \tag{3.14}$$

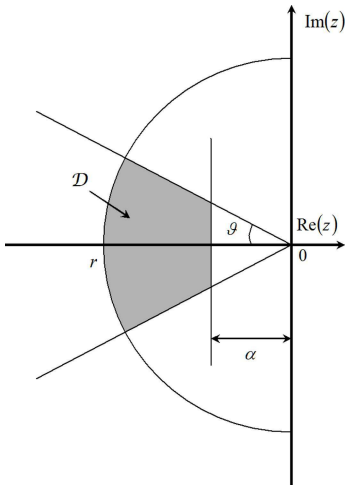


Figure 3.4: $\mathcal{D}$ performance-stability region.

3.1.2.2 D-Stability

In order to use the powerful machinery of LMI solvers to confine the eigenvalues of a plant inside a given region $\mathcal{D}$ of the complex plane $P_{\mathbb{C}}$, a formal definition of such a region is needed and it is given by the following statement [27]:

Definition 3.1. A subset $\mathcal{D}$ of the complex plane $P_{\mathbb{C}}$ is called an *LMI region* if there exists a symmetric matrix $\mathbf{\Lambda}$ with $\mathbf{\Lambda} = \mathbf{\Lambda}^T \in \mathbb{R}^{m\times m}$ and a matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$ so that: LMI region

$$\mathcal{D} =: \{f_{\mathcal{D}}(z) < 0,\ z \in \mathbb{C}\} \tag{3.15}$$

with:

$$f_{\mathcal{D}}(z) = \mathbf{\Lambda} + z\mathbf{M} + \bar{z}\mathbf{M}^T. \tag{3.16}$$

□

Given the negative definitiveness of Eq. 3.15 the LMI regions are always convex and symmetric with respect to the negative real axis of $P_{\mathbb{C}}$ since $f_{\mathcal{D}}(\bar{z}) = \bar{f}_{\mathcal{D}}(z)$. In addition, more complex LMI regions may be constructed by simpler ones since they are in general invariant under set intersection[3]. This result was used for example in the previous section in order to construct the $\mathcal{D}(\alpha, r, \vartheta)$ performance-stability region of Fig. 3.4 and will be further exploited when it comes to the placement of the eigenvalues of a LTI system inside this region.

Consider the following LTI and finite dimensional unforced system with $x \in \mathbb{R}^{n\times 1}$ and $\mathbf{A} \in \mathbb{R}^{n\times n}$:

$$\dot{x} = \mathbf{A}x \tag{3.17}$$

[3]This means that the intersection $f_{\mathcal{D}_1} \cap f_{\mathcal{D}_2}$ of two LMI regions is also an LMI region with $f_{\mathcal{D}_1\cap\mathcal{D}_2} = \mathrm{Diag}(f_{\mathcal{D}_1}, f_{\mathcal{D}_2})$.

The necessary and sufficient condition for the plant to be *quadratically asymptotically stable* is the following well-known Lyapunov inequality condition:

$$\exists \mathbf{X} = \mathbf{X}^T > 0 : \quad \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T < 0. \tag{3.18}$$

The aforementioned condition may be extended for general *stable* subregions $\mathcal{D}$ of the complex plane (LMI regions) as in Definition 3.1; if the spectrum of $\mathbf{A}$ belongs to $\mathcal{D}$, then the system in Eq. 3.17 is called $\mathcal{D}$-stable. The following theorem gives necessary and sufficient conditions for $\mathcal{D}$-stability of such a system:

Stability condition for eigenvalue placement

Theorem 3.1. Consider the system of Eq. 3.17 and a convex LMI region $\mathcal{D}$, characterized by the matrices $\mathbf{\Lambda}, \mathbf{M}$ and described by the complex function $f_{\mathcal{D}}(z)$ as in Definition 3.1. Consider also the $m \times m$ block matrix $\mathbf{F}_{\mathcal{D}}(\mathbf{A}, \mathbf{X})$ with:

$$\begin{aligned} \mathbf{F}_{\mathcal{D}}(\mathbf{A}, \mathbf{X}) &= \mathbf{\Lambda} \otimes \mathbf{X} + \mathbf{M} \otimes (\mathbf{A}\mathbf{X}) + \mathbf{M}^T \otimes (\mathbf{A}\mathbf{X})^T \\ &= \left[\Lambda_{kl}\mathbf{X} + \mathrm{M}_{kl}\mathbf{A}\mathbf{X} + \mathrm{M}_{lk}(\mathbf{A}\mathbf{X})^T \right]_{1 \le k,l \le m} \end{aligned} \tag{3.19}$$

The system in Eq. 3.17 is then called $\mathcal{D}$-stable if and only if there exists a matrix $\mathbf{X} = \mathbf{X}^T > 0$ so that the following LMI condition holds:

$$\mathbf{F}_{\mathcal{D}}(\mathbf{A}, \mathbf{X}) < 0. \tag{3.20}$$

□

Proof. See [27], Appendix.

■

From the preceding analysis it is obvious that one could concatenate more than one LMI's of the form $\mathbf{F}_{\mathcal{D}_i}(\mathbf{A}, \mathbf{X}) < 0$ for each i'th LMI region; their intersection then forms the desired eigenvalue placement region of Eq. 3.14. This is exactly the power of the method since complex, performance-tailored LMI regions may be easily described in this way.

$\mathcal{D}(\alpha, r, \vartheta)$ stability conditions

The corresponding LMI conditions for each of the $\mathcal{D}(\alpha, r, \vartheta)$ subregions are given by the following expressions:

$$\mathcal{D}_\alpha : \quad \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T + 2\alpha\mathbf{X} < 0 \tag{3.21}$$

$$\mathcal{D}_r : \quad \begin{bmatrix} -r\mathbf{X} & \mathbf{A}\mathbf{X} \\ \mathbf{X}\mathbf{A}^T & -r\mathbf{X} \end{bmatrix} < 0. \tag{3.22}$$

$$\mathcal{D}_\vartheta : \quad \begin{bmatrix} \sin\vartheta \left(\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T\right) & \cos\vartheta \left(\mathbf{A}\mathbf{X} - \mathbf{X}\mathbf{A}^T\right) \\ \cos\vartheta \left(\mathbf{X}\mathbf{A}^T - \mathbf{A}\mathbf{X}\right) & \sin\vartheta \left(\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T\right) \end{bmatrix} < 0. \tag{3.23}$$

This concludes the analysis concerning the conditions for eigenvalue placement inside LMI regions. In the following section the synthesis equations for the calculation of an output feedback $\mathscr{H}_\infty$ controller with additional eigenvalue placement constraints[4] for the closed loop eigenvalues will be given.

[4]The regional constraints will be of the form as in Eqs. 3.21-3.23.

3.1.3 Controller Synthesis

Consider a finite dimensional LTI standard plant $\mathbb{P}(s)$ where $x \in \mathbb{R}^{n\times 1}$ is the state vector, $u \in \mathbb{R}^{n_u\times 1}$ the control vector, $y \in \mathbb{R}^{n_p\times 1}$ the measurement vector, $\zeta_\infty \in \mathbb{R}^{n_\zeta\times 1}$ a generalized performance vector and $w \in \mathbb{R}^{n_w\times 1}$ an external perturbation vector[5]:

$$\mathbb{P}: \quad \begin{aligned} \dot{x} &= \mathbf{A}x + \mathbf{B}_w w + \mathbf{B}_u u \\ \zeta_\infty &= \mathbf{C}_\zeta x + \mathbf{D}_{w\zeta} w + \mathbf{D}_{u\zeta} u \\ y &= \mathbf{C}_y x + \mathbf{D}_{wy} w + \mathbf{D}_{uy} u \end{aligned} \tag{3.24}$$

Standard plant P

Consider also an LMI $\mathcal{D}$-stability region $\mathcal{D}(\alpha, r, \vartheta)$ (see Fig. 3.4) and some $\mathscr{H}_\infty$ performance level $\gamma > 0$. The goal is to calculate an output feedback dynamic controller $K(s)$ in a standard l-LFT form (see Fig. 3.5) so that:

Synthesis constraints

- The eigenvalues of the closed loop system interconnection are placed inside the LMI region $\mathcal{D}(\alpha, r, \vartheta)$.
- The $\mathscr{H}_\infty$ performance level for the transfer function from the disturbance to the performance vector is satisfied; i.e. $\|T_{w\zeta_\infty}(s)\|_\infty < \gamma$ with $T_{w\zeta_\infty}(s) = \mathcal{F}_l(\mathbb{P}, K)$.

The output feedback controller $K(s)$ having as a task to achieve the aforementioned goals has the following standard form (with $x_k \in \mathbb{R}^{n_k\times 1}$ the controller state):

$$K: \quad \begin{aligned} \dot{x}_k &= \mathbf{A}_k x_k + \mathbf{B}_k y \\ u &= \mathbf{C}_k x_k + \mathbf{D}_k y. \end{aligned} \tag{3.25}$$

Feedback controller

The closed loop system transfer function $T_{w\zeta_\infty}(s)$ from the perturbation to the performance vector is written as:

$$T_{w\zeta_\infty}(s) \triangleq \mathbf{C}_{\text{cl}}(s\mathbb{I} - \mathbf{A}_{\text{cl}})^{-1}\mathbf{B}_{\text{cl}} + \mathbf{D}_{\text{cl}}. \tag{3.26}$$

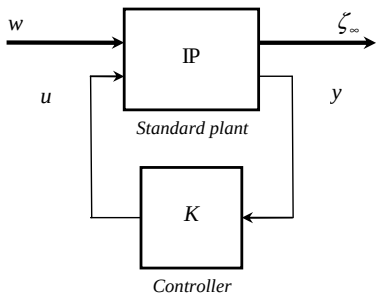


Figure 3.5: Standard l-LFT interconnection.

[5]The plant may be considered strictly proper ($\mathbf{D}_{uy} = 0$) without any loss of generality.

with:

$$\mathbf{A}_{\text{cl}} = \begin{bmatrix} \mathbf{A} + \mathbf{B}_u \mathbf{D}_k \mathbf{C}_y & \mathbf{B}_u \mathbf{C}_k \\ \mathbf{B}_k \mathbf{C}_y & \mathbf{A}_k \end{bmatrix} \tag{3.27}$$

$$\mathbf{B}_{\text{cl}} = \begin{bmatrix} \mathbf{B}_w + \mathbf{B}_u \mathbf{D}_k \mathbf{D}_{wy} \\ \mathbf{B}_k \mathbf{D}_{wy} \end{bmatrix} \tag{3.28}$$

$$\mathbf{C}_{\text{cl}} = \begin{bmatrix} \mathbf{C}_\zeta + \mathbf{D}_{u\zeta} \mathbf{D}_k \mathbf{C}_y & \mathbf{D}_{u\zeta} \mathbf{C}_k \end{bmatrix} \tag{3.29}$$

$$\mathbf{D}_{\text{cl}} = \mathbf{D}_{w\zeta} + \mathbf{D}_{u\zeta} \mathbf{D}_k \mathbf{D}_{wy}. \tag{3.30}$$

Before giving the formulas for controller synthesis, it should be outlined that these will involve the satisfaction of the BRL for the $\mathscr{H}_\infty$ part (with a Lyapunov matrix $\mathbf{X}_\infty$) and the eigenvalue placement LMI's of Eqs. 3.19-3.20 (with a Lyapunov matrix $\mathbf{X}$). However the problem is not tractable when considering two Lyapunov matrices so with some conservativeness it will be assumed that $\mathbf{X}_\infty = \mathbf{X}$. In addition, the BRL is not initially an LMI when substituting inside it the matrices of Eqs. 3.27-3.30 so a change of variables is needed (see [48]). The following theorem gives the final necessary and sufficient conditions for the problem:

Conditions for $\mathscr{H}_\infty$ control with eigenvalue placement constraints

Theorem 3.2. Let $\mathcal{D}$ be a desired LMI region of the complex plane $P_{\mathbb{C}}$ described by a characteristic function $f_{\mathcal{D}}(z)$ as in Eq. 3.16, with corresponding matrices $\mathbf{\Lambda}, \mathbf{M}$. There exist a Lyapunov matrix $\mathbf{X}$ and an output feedback controller $K(s) \triangleq \left[\begin{smallmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{C}_k & \mathbf{D}_k \end{smallmatrix}\right]$ that assure $\|T_{w\zeta_\infty}\|_\infty < \gamma$ and $\lambda(\mathbf{A}_{\text{cl}}) \in \mathcal{D}$ if and only if the following LMI's are satisfied:
Find $\mathbf{R} = \mathbf{R}^T, \mathbf{S} = \mathbf{S}^T \in \mathbb{R}^{n \times n}$ and matrices $\mathcal{A}_k, \mathcal{B}_k, \mathcal{C}_k, \mathbf{D}_k$ such that:

$$\begin{bmatrix} \mathbf{R} & \mathbb{I} \\ \mathbb{I} & \mathbf{S} \end{bmatrix} < 0 \tag{3.31}$$

$$\underset{1 \le k,l \le m}{} \left[\Lambda_{kl} \begin{bmatrix} \mathbf{R} & \mathbb{I} \\ \mathbb{I} & \mathbf{S} \end{bmatrix} + \mathrm{M}_{kl} \mathbf{\Phi} + \mathrm{M}_{lk} \mathbf{\Phi}^T \right] < 0 \tag{3.32}$$

$$\begin{bmatrix} \mathbf{\Psi}_{11} & \mathbf{\Psi}_{21}^T \\ \mathbf{\Psi}_{21} & \mathbf{\Psi}_{22} \end{bmatrix} < 0 \tag{3.33}$$

with the matrices $\mathbf{\Phi}, \mathbf{\Psi}_{11}, \mathbf{\Psi}_{21}, \mathbf{\Psi}_{22}$ being defined as:

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{AR} + \mathbf{B}_u \mathbf{C}_k & \mathbf{A} + \mathbf{B}_u \mathbf{D}_k \mathbf{C}_y \\ \mathcal{A}_k & \mathbf{SA} + \mathcal{B}_k \mathbf{C}_y \end{bmatrix} \tag{3.34}$$

$$\mathbf{\Psi}_{11} = \begin{bmatrix} \mathbf{AR} + (\mathbf{RA})^T + \mathbf{B}_u \mathcal{C}_k + (\mathbf{B}_u \mathcal{C}_k)^T & \mathbf{B}_w + \mathbf{B}_u \mathbf{D}_k \mathbf{D}_{wy} \\ (\mathbf{B}_w + \mathbf{B}_u \mathbf{D}_k \mathbf{D}_{wy})^T & -\gamma \mathbb{I} \end{bmatrix} \tag{3.35}$$

$$\mathbf{\Psi}_{21} = \begin{bmatrix} \mathcal{A}_k + (\mathbf{A} + \mathbf{B}_u \mathbf{D}_k \mathbf{C}_y)^T & \mathbf{SB}_w + \mathcal{B}_k \mathbf{D}_{wy} \\ \mathbf{C}_\zeta \mathbf{R} + \mathbf{D}_{u\zeta} \mathcal{C}_k & \mathbf{D}_{w\zeta} + \mathbf{D}_{u\zeta} \mathbf{D}_k \mathbf{D}_{wy} \end{bmatrix} \tag{3.36}$$

$$\mathbf{\Psi}_{22} = \begin{bmatrix} \mathbf{SA} + (\mathbf{SA})^T + \mathcal{B}_k \mathbf{C}_y + (\mathcal{B}_k \mathbf{C}_y)^T & (\mathbf{C}_\zeta + \mathbf{D}_{u\zeta} \mathbf{D}_k \mathbf{C}_y)^T \\ \mathbf{C}_\zeta + \mathbf{D}_{u\zeta} \mathbf{D}_k \mathbf{C}_y & -\gamma \mathbb{I} \end{bmatrix}. \tag{3.37}$$

□

Proof. See [27], pp. 365.

■

Controller reconstruction

Once the feasibility LMI's are solved obtaining $\mathbf{R}, \mathbf{S}, \mathcal{A}_k, \mathcal{B}_k, \mathcal{C}_k, \mathbf{D}_k$ and the minimal value of γ, the matrices $\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k$ of a full-order controller (i.e. $n_k = n$) can be then computed[6]. To do this, first a full rank factorization $\mathbf{N}_1\mathbf{N}_2^T = \mathbb{I} - \mathbf{RS}$ (with $\mathbf{N}_1, \mathbf{N}_2$ being square and invertible) is performed using SVD. The matrices $\mathbf{N}_1, \mathbf{N}_2$ are in fact parts of the Lyapunov matrix $\mathbf{X}$ and its inverse partitioned as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{R} & \mathbf{N}_1 \\ \mathbf{N}_1^T & \mathbf{U} \end{bmatrix} \tag{3.38}$$

$$\mathbf{X}^{-1} = \begin{bmatrix} \mathbf{S} & \mathbf{N}_2 \\ \mathbf{N}_2^T & \mathbf{V} \end{bmatrix}. \tag{3.39}$$

The controller matrices are finally calculated by inverting the transformations done to render the BRL convex[7]:

$$\mathcal{B}_k = \mathbf{N}_2\mathbf{B}_k + \mathbf{S}\mathbf{B}_u\mathbf{D}_k \tag{3.40}$$

$$\mathcal{C}_k = \mathbf{C}_k\mathbf{N}_1^T + \mathbf{D}_k\mathbf{C}_y\mathbf{R} \tag{3.41}$$

$$\mathcal{A}_k = \mathbf{N}_2\mathbf{A}_k\mathbf{N}_1^T + \mathbf{N}_2\mathbf{B}_k\mathbf{C}_y\mathbf{R} + \mathbf{S}\mathbf{B}_u\mathbf{C}_k\mathbf{N}_1 + \mathbf{S}(\mathbf{A} + \mathbf{B}_u\mathbf{D}_k\mathbf{C}_y)\mathbf{R}. \tag{3.42}$$

Obviously, if the eigenvalues of the system have to be confined inside a generic LMI region being the intersection of several basic LMI regions $\mathcal{D}_i$, as is the case with the $\mathcal{D}(\alpha, r, \vartheta)$ performance-stability region of Fig. 3.4, then several inequality conditions of the type as in Eq. 3.32 should be concatenated and satisfied simultaneously.

Technical Note: As far as practical implementation is concerned, one could either code the LMI feasibility conditions of Theorem 3.2 using available software for LMI solution (MATLAB® LMI Toolbox, YALMIP etc.) or use the available macro `hinfmix` of MATLAB® Robust Control Toolbox which performs multi-objective $\mathscr{H}_2/\mathscr{H}_\infty$ synthesis with eigenvalue placement constraints. This macro gives the dynamic controller $K(s)$ satisfying all the constraints as well as the minimum γ attained and the feasibility matrices $\mathbf{R}, \mathbf{S}$ of Theorem 3.2.
The macro `hinfmix` takes the argument `'region'` that is consisted in fact of the matrices $\mathbf{\Lambda}, \mathbf{M}$ characterizing the LMI region $\mathcal{D}$ used for the eigenvalue placement. The argument `'region'` is obtained from an interactive dialog with the user by the macro `lmireg` of the same toolbox. A predefined inventory with various LMI regions (horizontal and vertical strips, disks, conic sections etc.) is available by default.

[6] More precisely the order n_k is equal to rank($\mathbf{RS} - \mathbb{I}$), with $\mathbf{R}, \mathbf{S}$ given by Theorem 3.2.
[7] The final implemented controller is given then by Eq. 3.25.

3.2 Compensator Estimator-Controller Form

In this section some results will be presented on the subject of converting a generic compensator $K(s)$ into an equivalent observer/state feedback controller form. A very good treatment of the subject can be found in [4] with most of the material drawn from the original work found in [20, 21].

3.2.1 Motivation

The conversion of a generic compensator $K(s)$ to an observer/state feedback controller form may be useful for two reasons. The first has to do with the advantage of observer/state feedback controllers to preserve the same state representation as the plant for which they have been computed, since the controller states can be viewed as estimates of the plant's states. The second reason is relevant with gain scheduling; if the compensator's parameters are to be changed/updated, it is better to update only the two matrices $\mathbf{K}_{\mathrm{c}}, \mathbf{K}_{\mathrm{o}}$ of the observer/state feedback form of the compensator instead of updating the compensator matrices $\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k, \mathbf{D}_k$.

The material of this section will be used in conjunction with the analysis of Section 3.1, where dynamic compensators with eigenvalue placement constraints are designed, in order to conceive gain-scheduled control laws based on the observer/state feedback controller interpolation technique of Chapter 5.

3.2.2 Controller Transformation

The general idea is based on the fact that an output feedback dynamic compensator $K(s)$ of the form (with $x_k \in \mathbb{R}^{n_k \times 1}$ being the compensator state vector and $n_k \geq n$):

$$K: \quad \begin{aligned} \dot{x}_k &= \mathbf{A}_k x_k + \mathbf{B}_k y \\ u &= \mathbf{C}_k x_k + \mathbf{D}_k y. \end{aligned} \tag{3.43}$$

may be transformed to an equivalent observer/state feedback controller being of the same order as the plant for which it has been computed (order n) plus the famous Youla parameter $Q(s)$ that is an always stable system of order $n_k - n$.

Discussion on the Youla parameter

In order to avoid having an additional dynamical system $Q(s)$ for which interpolation should be also used when coming to gain scheduling, the compensators designed must be forced to have *the same order* as the plant. In this case, after the transformation that will be detailed in the following lines, the Youla parameter becomes only a static gain:

$$Q(s) \equiv \mathbf{D}_k. \tag{3.44}$$

Furthermore, if the compensators designed are restricted to be strictly proper, then the Youla parameter is zero and the compensator $K(s)$ may be represented by a standard Kalman observer plus a state feedback gain (see Fig. 3.6).

The conversion formulas will now be given according to [4]. Consider without loss of generality a finite dimensional strictly proper LTI system $G(s)$ with the following state space representation (with $x \in \mathbb{R}^{n\times 1}$ the state vector, $u \in \mathbb{R}^{n_u\times 1}$ the control vector and $y \in \mathbb{R}^{n_y\times 1}$ the measurement vector):

$$G: \quad \begin{aligned} \dot{x} &= \mathbf{A}x + \mathbf{B}u \\ y &= \mathbf{C}x. \end{aligned} \tag{3.45}$$

Consider also a LTI dynamic output feedback compensator as in Eq. 3.43[8] that stabilizes the plant and may also provide additional useful properties such as robustness to external perturbations, eigenvalue placement etc.

Conversion procedure

The idea here is to transform the dynamic controller $K(s)$ in such a way so that it would be a Kalman observer of a linear transformation of the plant state x or:

$$x_k = \mathbf{T}\hat{x}. \tag{3.46}$$

A Kalman observer of the system's state plus the state feedback controller are written as:

$$\begin{aligned} \dot{\hat{x}} &= \mathbf{A}\hat{x} + \mathbf{B}u + \mathbf{K}_\mathrm{o}(y - \mathbf{C}\hat{x}) \\ u &= \mathbf{K}_\mathrm{c}\hat{x}. \end{aligned} \tag{3.47}$$

If the state feedback controller equation is substituted to the observer one then the following dynamic controller is obtained:

$$\begin{aligned} \dot{\hat{x}} &= (\mathbf{A} + \mathbf{B}\mathbf{K}_\mathrm{c} - \mathbf{K}_\mathrm{o}\mathbf{C})\hat{x} + \mathbf{K}_\mathrm{o}y \\ u &= \mathbf{K}_\mathrm{c}\hat{x}. \end{aligned} \tag{3.48}$$

Now if the state transformation of Eq. 3.46 is performed to Eq. 3.48, then the following compensator form is calculated:

$$\begin{aligned} \dot{x}_k &= \mathbf{T}(\mathbf{A} + \mathbf{B}\mathbf{K}_\mathrm{c} - \mathbf{K}_\mathrm{o}\mathbf{C})\mathbf{T}^{-1}x_k + \mathbf{T}\mathbf{K}_\mathrm{o}y \\ u &= \mathbf{K}_\mathrm{c}\mathbf{T}^{-1}x_k. \end{aligned} \tag{3.49}$$

Finally, by performing an one to one identification of the controller matrices of Eq. 3.49 using Eq. 3.47, the following conditions are found:

$$\mathbf{A}_k\mathbf{T} - \mathbf{T}\mathbf{A} - \mathbf{T}\mathbf{B}\mathbf{C}_k\mathbf{T} + \mathbf{B}_k\mathbf{C} = 0 \tag{3.50}$$

and:

$$\mathbf{K}_\mathrm{o} = \mathbf{T}^{-1}\mathbf{B}_k \tag{3.51}$$

$$\mathbf{K}_\mathrm{c} = \mathbf{C}_k\mathbf{T}. \tag{3.52}$$

Equation 3.50 is a generalized non-symmetric rectangular Riccati equation that can be solved using the following technical note. After it has been solved, the observer/controller matrices of Eqs. 3.51-3.52 may be finally computed.

[8] The compensator $K(s)$ is considered strictly proper ($\mathbf{D}_k = 0$).

Technical Note: To solve the Riccati equation [9] (Eq. 3.50), it suffices to observe that it may be written as:

$$\begin{bmatrix} -\mathbf{T} & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B}\mathbf{C}_k \\ \mathbf{B}_k\mathbf{C} & \mathbf{A}_k \end{bmatrix} \begin{bmatrix} \mathbb{I} \\ \mathbf{T} \end{bmatrix} = 0. \tag{3.53}$$

The matrix in the middle of Eq. 3.53 is nothing else than the closed loop matrix of the system $\mathbf{A}_{\text{cl}}$ (see Eq. 3.27 with $\mathbf{B}_u = \mathbf{B}, \mathbf{C}_y = \mathbf{C}$ and $\mathbf{D}_k = 0$). Use then eigenvector decomposition in order to find a matrix $\mathbf{U}$ and a matrix $\mathbf{\Lambda}$ so that:

$$\mathbf{U}^{-1}\mathbf{A}_{\text{cl}}\mathbf{U} = \mathbf{\Lambda}. \tag{3.54}$$

Then compute the solution to the generalized non-symmetric rectangular Riccati equation of Eq. 3.50 as:

$$\mathbf{T} = \mathbf{U}_{21}\mathbf{U}_{11}^{-1} \tag{3.55}$$

where the matrix $\mathbf{U}$ is partitioned as $\mathbf{U} = \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ \mathbf{U}_{21} & \mathbf{U}_{22} \end{bmatrix}$.

Eigenvalue ordering

Now if the eigenvalues of the closed loop system *are not repeated*, then the columns of $\mathbf{U}$ are simply the eigenvectors of the closed loop system matrix $\mathbf{A}_{\text{cl}}$ and $\mathbf{\Lambda}$ is a diagonal matrix whose elements are the eigenvalues of $\mathbf{A}_{\text{cl}}$. The solutions of the Riccati equations are not unique since each solution correspond to a different ordering of the columns of $\mathbf{U}$. In general, the eigenvectors should be ordered in such a way that the first n ones correspond to the closed loop controller eigenvalues whereas the rest to the n estimator eigenvalues. This ordering should be done according to the rapidity of the eigenvalues and is not always trivial (for further discussion see [21]). The possible orderings are in fact the combinations of n eigenvalues out of totally $2n$ or $\binom{2n}{n}$ and can be as high as 20 for a third order compensator.

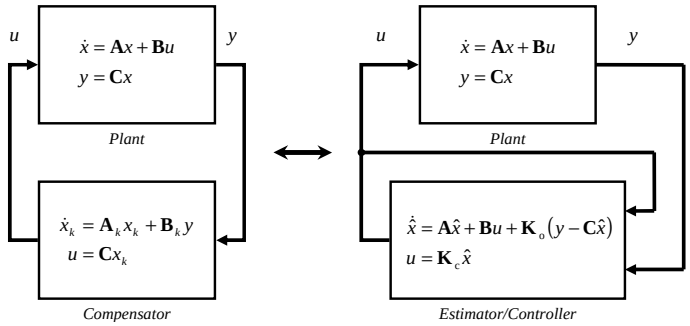


Figure 3.6: Compensator and estimator/controller equivalence.

[9] For alternative methods to solve the Riccati equation see [20], pp. 1576 and references therein.

3.3 $\mathscr{H}_\infty$ Loop Shaping

In this section, an alternative method to the classic $\mathscr{H}_\infty$ formulation for the design of robust output feedback controllers of Section 3.1 will be presented. This method is based on the famous *loop shaping* approach for the design of MIMO output feedback controllers first introduced in [98].

In Section 3.3.1 the principal ideas motivating the method are presented whereas in Section 3.3.2 the method itself is outlined. The solution to the problem of designing full order controllers, satisfying the analysis of Sections 3.3.1, 3.3.2 using the Riccati and LMI-based formulation, is presented in 3.3.3. Finally in Section 3.3.4 the problem of designing a static instead of a full order controller is described.

3.3.1 Motivation

Standard $\mathscr{H}_\infty$ control vs. LSDP

The methodology of designing a classic $\mathscr{H}_\infty$ robust output feedback controller $K(s)$ for a LTI system $G(s)$ is mainly based on shaping the singular values of certain transfer functions of the closed loop via the use of weighting filters. These closed loop transfer functions are defined using an input vector w representing external disturbances and/or reference signals and an output vector ζ_∞ representing critical signals on the closed loop system needing particular treatment as for example error and/or control signals (see Fig. 3.5). Then a stabilizing controller is computed, ensuring that the $\mathscr{H}_\infty$ norm of the transfer function $T_{w\zeta_\infty}(s) = \mathcal{F}_l(\mathbb{P}, K)$ from w to ζ_∞ is minimized. Additional constraints (such as closed loop eigenvalue placement ones) could be subsequently added as in Section 3.1.

A major disadvantage of this method is that the choice of the weighting functions used is either highly empirical or involves a great number of trial and error experiments until a good compromise between performance and robustness is found. At the end, one cannot be sure that the best choice of filters has been done and as a result, it cannot be argued that the controller computed is the optimal one (even though it is optimal (or sub-optimal) for the selected filters).

An alternative to this method is the so-called *loop shaping design procedure* (LSDP) first introduced in [98]. It is based on the fact that performance and robustness optimization could be separated in two phases (retaining always a trade-off between the two) and therefore obtain more flexibility in controller design. This procedure uses this time the *open loop* singular values $\sigma(GK)$ over specified frequency ranges in order to characterize the closed loop performance and robustness of the plant instead of using directly various weighted *closed loop* transfer functions as with standard $\mathscr{H}_\infty$ control.

This in fact is in accordance with the classic SISO control practice where controllers are designed in order to achieve high/low open loop gain at low/high frequencies for good performance/robustness and a correct roll off rate at the gain crossover frequency to maintain stability avoiding excessive phase lag.

Generalized feedback scheme

To prove that in fact it is possible to describe *closed loop* performance and robustness requirements by the use of *open loop* information, consider the following generalized feedback interconnection scheme where d, n may model various disturbances, sensor noises or reference signals acting on the plant G, whereas y, u are the inputs/outputs to the controller K.

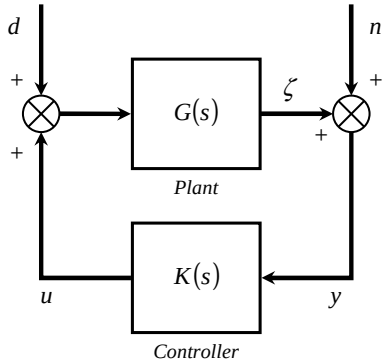


Figure 3.7: Generalized feedback interconnection.

The reason for which this feedback structure is chosen is twofold. First it incorporates all possible external signals acting on G in a general manner. Second it permits to define an uncertain system G_Δ with additive perturbations to its coprime factors in a straightforward way as it will be presented at the end of this section. The controller K is then computed to stabilize the plant G in the face of such uncertainties in the context of the LSDP. As an extension to this stabilization property, this feedback structure permits also the introduction of another significant system notion: the *gap metric*. The interconnection of the gap metric and the robust stabilization of an uncertain system G_Δ using a controller K is very important as it will be shown in Section 3.4 and it will be used for the development of gain-scheduled control laws in Chapter 6.

The analysis presented further down will show that many performance and requirements for a closed loop system may be incorporated using the feedback scheme shown in Fig. 3.7. The transfer functions considered are the ones from the external disturbances n, d to the controller output/inputs y, u. After simple matrix manipulation one gets:

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} KS & T' \\ S & SG \end{bmatrix} \begin{bmatrix} n \\ d \end{bmatrix}. \tag{3.56}$$

In the previous equation S is called the *output sensitivity matrix* and T' the *input complementary sensitivity matrix* defined as:

$$S \triangleq (\mathbb{I} - GK)^{-1} \tag{3.57}$$

$$T' \triangleq K(\mathbb{I} - GK)^{-1}G = KSG. \tag{3.58}$$

A very good control strategy, following a classic $\mathscr{H}_\infty$ formulation, is in fact to design a stabilizing controller K in such a way that the effect of the input/output disturbances d, n on the open loop plant G is also minimized over all frequencies. This means that K is designed to satisfy:

$$\left\| \begin{bmatrix} KS & T' \\ S & SG \end{bmatrix} \right\|_\infty \equiv \left\| \begin{bmatrix} SG & T \\ S' & KS \end{bmatrix} \right\|_\infty < \gamma, \quad \gamma > 0 \tag{3.59}$$

with S' being the *input sensitivity matrix* and T the *output complementary sensitivity matrix* defined as[10]:

$$S' \triangleq (\mathbb{I} - KG)^{-1} \tag{3.60}$$

$$T \triangleq GK(\mathbb{I} - GK)^{-1} = GKS. \tag{3.61}$$

From *open* to *closed*-loop objectives

The minimization of the $\mathscr{H}_\infty$-norm of each one of the six closed loop transfer functions-objectives S, S', T, T', SG, KS of Eq. 3.59 over all frequencies is not generally possible because there is always a conflict between them due to the fact that for example $S + T = \mathbb{I}$. However this minimization could be performed for different zones of frequencies and also considering *open loop* instead of these *closed loop* objectives.

Take for instance the $\mathscr{H}_\infty$-norm of the output sensitivity matrix S which can be studied by studying its maximum singular values $\overline{\sigma}(S)$ for all frequencies. For frequencies where the minimum singular values of the open loop transfer function are big (i.e. $\underline{\sigma}(GK) \gg 1$), then the following approximation holds (see [155], pp. 131-133, 486 or [96], pp. 102):

$$\overline{\sigma}(S) = \overline{\sigma}\big((\mathbb{I} - GK)^{-1}\big) \simeq \frac{1}{\underline{\sigma}(GK)} \tag{3.62}$$

The sensitivity function gives the influence of output disturbances n to the plant output y. If these disturbances n are reference signals (typically low frequency ones) then $\|S\|_\infty$ should be minimized for low frequencies in order to ensure good reference tracking and hence from Eq. 3.62, $\underline{\sigma}(GK)$ should be maximized for a *low* frequency band $\omega < \omega_l$.

In conflict with the above requirement, suppose the output disturbance n is a high frequency noise. The complementary sensitivity function $T = GKS$ gives the *closed loop* noise influence on the plant output ζ. The maximum singular values of T can be approximated for some frequencies where the maximum of the *open loop* singular values are small ($\overline{\sigma}(GK) \ll 1$) with:

$$\overline{\sigma}(T) = \overline{\sigma}\big(GK(\mathbb{I} - GK)^{-1}\big) \simeq \overline{\sigma}(GK). \tag{3.63}$$

In this case (and given that n appears in high frequencies) $\overline{\sigma}(GK)$ should be minimized at a *high* frequency band $\omega > \omega_h$. As a final remark it could be added that the frequency zone (ω_l, ω_h), defining the *roll-off* rate, cannot be made very small since this could lead to excessive phase lag and hence instability[11].

[10] The classic relations $S + T = \mathbb{I}$ and $S' + T' = \mathbb{I}$ always hold.

[11] A theoretical maximum is -40db/dec but in practice a 20db/dec rate is preferred.

On loop shaping

The two aforementioned closed loop objectives (reference signal tracking, output measurement noise rejection) are typically *performance* and *robustness* ones and can be adjusted by making the minimum/maximum *open loop* system singular values big/small at low/high frequencies ($\omega < \omega_l, \omega > \omega_h$) respectively while retaining a roll of rate of around $20db/dec$ for intermediate frequencies. Typically this leads to a singular value shaping as in Fig. 3.8[12].

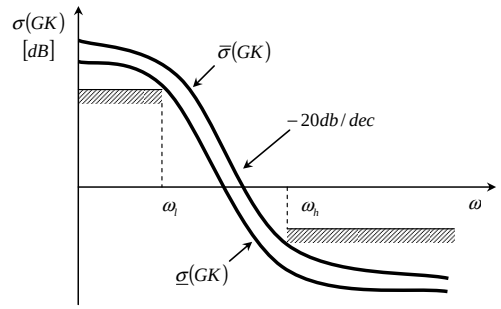


Figure 3.8: Shaping of the open loop singular values.

In the context of the LSDP (see Section 3.3.2), these *open loop* singular values could be 'shaped' using appropriate pre/post filters W_1, W_2 before and after the system G so that the shaped open loop plant's $G_s = W_2 G W_1$ singular values have a desired shape (see Fig. 3.8). However, this shaping is done *without regarding the plant's phase* (except for a rough demand on the roll-off rate) and hence loop stability. The solution to this problem is addressed by adding an additional controller acting on G_s in such a way that guarantees loop stability. In the LSDP procedure, this controller is of a special type: it stabilizes but additionally it renders the loop robust over left unstructured NCF uncertainties.

NCF robust stab/tion

In the remainder of this section, this problem of robust stabilization over left unstructured NCF uncertainties is posed. Its solution is presented in Section 3.3.3 using both the standard Glover/Doyle and the equivalent LMI formulations. In addition further down it will be shown that the solution to this four-block shaping problem of Eq. 3.59 detailed before, and which has a standard $\mathscr{H}_\infty$ formulation, is equivalent to this robust stabilization problem.

The question however arising is why consider such a specific type of uncertainties. It has been argued in [143, 144] that this way of representing process uncertainty is very attractive since no particular information in its form in demanded and hence general perturbed plants G_Δ that are situated 'around' a nominal plant G could be considered. This could be of use for example for linearized plants of a nonlinear system being computed on equilibrium points close enough to a reference equilibrium point; these linearized plants could be considered as perturbed plants G_Δ over the reference plant G.

[12] Additionally one could also treat the closed loop objectives SG, KS that boil down to specifications on the controller's minimum/maximum singular values $\underline{\sigma}(K), \overline{\sigma}(K)$ for low/high frequencies (performance/robustness specifications respectively) by further adjusting K.

NCF perturbed system

Start by considering the left NCF representation of a nominal plant $G(s)$ as (with $\tilde{N}\tilde{N}^* + \tilde{M}\tilde{M}^* = \mathbb{I}$)[13]:

$$G \triangleq \tilde{M}^{-1}\tilde{N}. \tag{3.64}$$

A perturbed plant G_Δ with the perturbations acting on its left NCF's is considered as (see Fig. 3.9)[14]:

$$G_\Delta = (\tilde{M} + \Delta_M)^{-1}(\tilde{N} + \Delta_N) \tag{3.65}$$

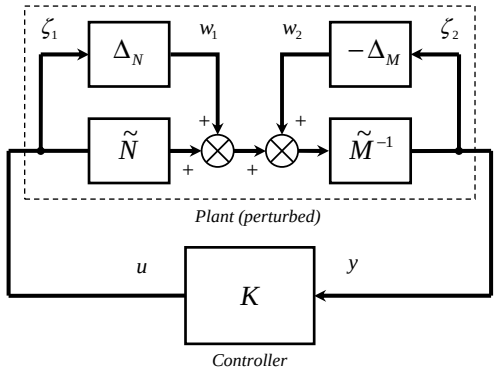


Figure 3.9: NCF perturbed plant.

If $w = w_1 + w_2$ is the external perturbation, $\zeta_\infty = [\zeta_1\ \zeta_2]^T = [u\ y]^T$ is the performance vector and the uncertainty is regrouped as $\Delta = [\Delta_N\ -\Delta_M]$, then the perturbed plant can be written as the upper linear fractional transformation (u-LFT) of the standard plant $\mathbb{P} = \left[\begin{smallmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{smallmatrix}\right]$ (corresponding to the nominal plant G) and the unstructured uncertainty Δ as:

$$\begin{aligned} G_\Delta &= \mathcal{F}_U(\mathbb{P}, \Delta) \\ &\triangleq P_{22} + P_{21}\Delta(\mathbb{I} - P_{11}\Delta)^{-1}P_{12}. \end{aligned} \tag{3.66}$$

where the following standard I/O signal expression:

$$\begin{bmatrix} \zeta_\infty \\ y \end{bmatrix} = \mathbb{P} \begin{bmatrix} w \\ u \end{bmatrix} \tag{3.67}$$

defines $\mathbb{P}$ as:

$$\mathbb{P} = \begin{bmatrix} 0 & \mathbb{I} \\ \tilde{M}^{-1} & G \\ \tilde{M}^{-1} & G \end{bmatrix}. \tag{3.68}$$

[13] In the context of the LSDP this open loop plant will be in fact an already shaped one $G_s = W_2 G W_1$.

[14] The perturbations considered are unstructured & unknown but stable & bounded transfer functions with $\left\|[\Delta_M\ \Delta_N]\right\|_\infty < \epsilon$ and $\epsilon > 0$.

The NCF robust stab/tion problem

The NCF *robust stabilization problem* of G (equivalently G_s in the LSDP) with a controller K is taken for their l-LFT $\mathcal{F}_L(\mathbb{P}, K)$ with:

$$\mathcal{F}_L(\mathbb{P}, K) \equiv \begin{bmatrix} KS\tilde{M}^{-1} \\ S\tilde{M}^{-1} \end{bmatrix}. \tag{3.69}$$

A formal definition of the NCF optimal robust stabilization problem is given in the following theorem [96, 97]:[15]

Theorem 3.3. Consider a plant G and stable norm-bounded left NCF uncertainties $\Delta = [\Delta_N \ -\Delta_M]$ with $\|\Delta\|_\infty < \epsilon$ and $\epsilon > 0$. A robust controller K stabilizes the perturbed plant $G_\Delta = \mathcal{F}_U(\mathbb{P}, \Delta)$ for all such Δ with $\mathbb{P}$ being the related to G standard plant if and only if:

a. K stabilizes G.

b. $\|\mathcal{F}_L(\mathbb{P}, K)\|_\infty = \left\| \begin{bmatrix} K(\mathbb{I} - GK)^{-1}\tilde{M}^{-1} \\ (\mathbb{I} - GK)^{-1}\tilde{M}^{-1} \end{bmatrix} \right\|_\infty \leq \epsilon^{-1}$.

□

Proof. See [96], pp. 33-36.

■

From *two* to *four*-block problem

The above *two-block* robust optimization problem is proven immediately to be equivalent to the initial *four-block* disturbance rejection problem since the factorization of G is normalized and:

$$\begin{bmatrix} \mathbb{I} & G \end{bmatrix} = \tilde{M}^{-1} \begin{bmatrix} \tilde{M} & \tilde{N} \end{bmatrix}. \tag{3.70}$$

So finally Eqs. 3.59, 3.69 are equivalent:

$$\begin{aligned} \left\| \begin{bmatrix} K(\mathbb{I} - GK)^{-1} & K(\mathbb{I} - GK)^{-1}G \\ (\mathbb{I} - GK)^{-1} & (\mathbb{I} - GK)^{-1}G \end{bmatrix} \right\|_\infty &\equiv \left\| \begin{bmatrix} K(\mathbb{I} - GK)^{-1}\tilde{M}^{-1} \\ (\mathbb{I} - GK)^{-1}\tilde{M}^{-1} \end{bmatrix} \right\|_\infty \\ \Leftrightarrow \left\| \begin{bmatrix} K \\ \mathbb{I} \end{bmatrix} (\mathbb{I} - GK)^{-1} \begin{bmatrix} \mathbb{I} & G \end{bmatrix} \right\|_\infty &\equiv \left\| \begin{bmatrix} K \\ \mathbb{I} \end{bmatrix} (\mathbb{I} - GK)^{-1}\tilde{M}^{-1} \right\|_\infty. \end{aligned} \tag{3.71}$$

The motivation for the use of robust stabilizing $\mathscr{H}_\infty$ controllers is now clear: since the feedback loop could obtain some good initial performance and maybe robustness properties by the use for example of some pre-compensators shaping its open loop singular values, such an $\mathscr{H}_\infty$ stabilizing controller could be subsequently added in the loop rendering it also robust over generalized unstructured coprime factor uncertainties. This procedure outlined here will be presented in detail in the following section.

[15] It is of course attempted to maximize this robustness margin ϵ to an optimum value $\epsilon_{\max}$.

3.3.2 The Loop Shaping Design Procedure (LSDP)

The *loop shaping design procedure* (LSDP) of McFarlane & Glover presented in detail in [97], [96] - Ch. 6, [155] - Ch. 18, [93] - Ch. 12 will be outlined in the first part of this section. Some results concerning this method will equally be presented in the second part[16]. The LSDP can be divided in three distinct steps and is visualized in Figs. 3.10a, 3.10b. These steps are:

The LSDP

Step 1 - Loop Shaping. The initial open loop plant $G(s)$ is augmented using pre/post compensators W_1, W_2 in order to shape the singular values $\sigma(G_\mathrm{s})$ of the new augmented *open loop* plant $G_\mathrm{s} = W_2 G W_1$, following the analysis of the previous section (see Fig. 3.8). Typically the designer chooses W_1 as a low pass filter in order to have a sufficiently small loop gain in high frequencies ($\omega > \omega_h$) and W_2 in order to assure a good tracking performance with high loop gain in low frequencies ($\omega < \omega_l$). In addition the compensators should also be chosen in such a way so as to avoid excessive roll-off rates (and hence instability) in intermediate frequencies (typically around the gain crossover frequency ω_g); a good value being about $-20dB/dec$[17].

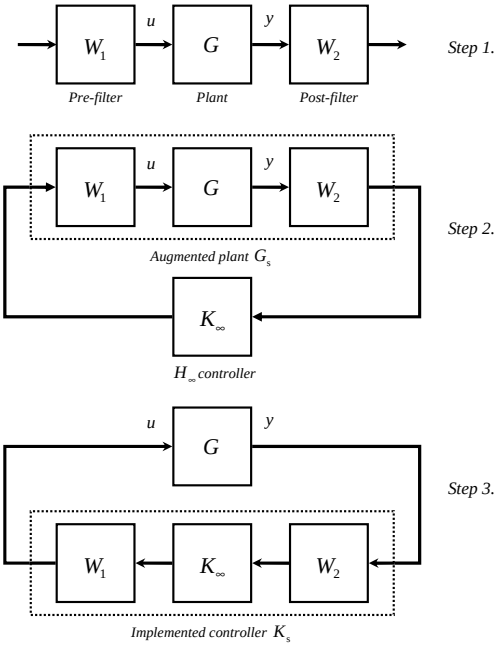
Step 2 - Controller Calculation. The maximum robustness margin ϵ_max achieved by a robust stabilizing controller K_∞, designed for the open loop system G_s, can be calculated before actually K_∞ is computed (see Section 3.3.3). This maximum stability margin ϵ_max is also a measure of the success of the loop shaping performed in *Step 1*. If $\epsilon_\mathrm{max} \ll 1$ then there is an incompatibility between the chosen loop shape, the shaped plant's phase and robust closed loop stability; thus W_1, W_2 should be readjusted. If ϵ_max is satisfactory (typically around 0.3), then select a sub-optimal $\epsilon < \epsilon_\mathrm{max}$ and calculate the $\mathscr{H}_\infty$ controller K_∞ that robustly stabilizes the loop.

Step 3 - Controller Implementation. The final controller $K_\mathrm{s} = W_1 K_\infty W_2$ being the series interconnection of the pre/post compensators and the robust controller may now be implemented. The order of K_s is equal to the order of the plant n plus the sum of the orders of the pre/post compensators $n_w = n_{w_1} + n_{w_2}$. As a result, the order n_s of the final controller is $n + 2n_w$, fact that could be conservative for implementation. To solve this problem, model reduction techniques could be applied on K_∞ while using compensators W_1, W_2 of the simplest possible structure. Another solution could be the design of a reduced order or even a static robust controller K_∞ from the beginning, a problem which is in general difficult to solve[18].

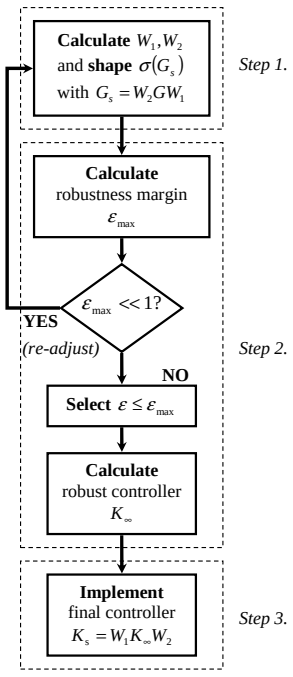
[16] This method gives a solid alternative to classic $\mathscr{H}_\infty$ control strategy. It has been applied to a big number of study cases (see Section 5.2 for a comprehensive review).

[17] Note that the compensators may also be unstable and/or having poles/zeros on the imaginary axis without this fact posing any problems since they are on the feedback loop in contrast to the classic $\mathscr{H}_\infty$ practice.

[18] In the context of gain-scheduled control where reduced order controllers are of particular interest, a solution to this problem is given in Section 3.3.4.



(a) *Block diagram form*



(b) *Algorithm form*

Figure 3.10: The loop shaping design procedure (LSDP).

The robust controller $K_\mathrm{s} = W_1 K_\infty W_2$ calculated using the LSDP will try to stabilize the plant while also rendering it robust to coprime factor uncertainties. However it may be possible that it will degrade the singular value shape of the initial open loop $G_s = W_2 G W_1$ adjusted with the pre-post compensators. The following analysis shows that it is in fact possible to compute maximum limits for this degradation that depend not on the controller K_∞ but only on the singular value shape of the initial open loop and also on the robustness margin ϵ that may be achieved by K_∞. Given however that this robustness margin depends by its turn *only* on W_1, W_2 (and hence on G_s), it follows that the degradation is dependent only on the initial loop shape.

Degradation of the initial loop shape

To compute this degradation it is imperative to compare the initial loop shape of $G_s = W_2 G W_1$ with the final ones $W_1 K_\infty W_2 G = K_\mathrm{s} G$ at the plant input and $G W_1 K_\infty W_2 = G K_\mathrm{s}$ at the plant output. For low (respectively high) frequencies the minimum (respectively maximum) singular values of the initial and final open loop shapes are compared.
For *low frequencies* the following relations hold[19]:

$$\textit{Plant output:} \qquad \underline{\sigma}(GK_\mathrm{s}) \geq \frac{\underline{\sigma}(G_\mathrm{s})\underline{\sigma}(K_\infty)}{c(W_2)} \tag{3.72}$$

$$\textit{Plant input:} \qquad \underline{\sigma}(K_\mathrm{s}G) \geq \frac{\underline{\sigma}(G_\mathrm{s})\underline{\sigma}(K_\infty)}{c(W_1)} \tag{3.73}$$

whereas for *high frequencies*:

$$\textit{Plant output:} \qquad \overline{\sigma}(GK_\mathrm{s}) \leq \overline{\sigma}(G_\mathrm{s})\overline{\sigma}(K_\infty)c(W_2) \tag{3.74}$$

$$\textit{Plant input:} \qquad \overline{\sigma}(K_\mathrm{s}G) \leq \overline{\sigma}(G_\mathrm{s})\overline{\sigma}(K_\infty)c(W_1). \tag{3.75}$$

The only obstacle to find a lower (respectively upper) bound for the low (respectively high) frequency loop shape deterioration at the input/output of the plant is to obtain bounds on the minimum (respectively maximum) singular values of the controller K_∞. This bound is obtained in [96], pp. 110-116:

$$\underline{\sigma}(K_\infty) \geq \frac{\underline{\sigma}(G_\mathrm{s}) + \sqrt{\gamma^2 - 1}}{1 + \underline{\sigma}(G_\mathrm{s})\sqrt{\gamma^2 - 1}} \quad \text{for } \forall\omega \text{ where} \quad \underline{\sigma}(G_\mathrm{s}) > \sqrt{\gamma^2 - 1} \tag{3.76}$$

$$\overline{\sigma}(K_\infty) \leq \frac{\overline{\sigma}(G_\mathrm{s}) - \sqrt{\gamma^2 - 1}}{1 - \overline{\sigma}(G_\mathrm{s})\sqrt{\gamma^2 - 1}} \quad \text{for } \forall\omega \text{ where} \quad \overline{\sigma}(G_\mathrm{s}) < \frac{1}{\sqrt{\gamma^2 - 1}}. \tag{3.77}$$

In addition these two bounds $\underline{\sigma}(K_\infty), \overline{\sigma}(K_\infty)$ tend asymptotically to $\dfrac{1}{\sqrt{\gamma^2 - 1}}$ and to $\sqrt{\gamma^2 - 1}$ respectively if the values of $\underline{\sigma}(G_\mathrm{s})$ and $\overline{\sigma}(G_\mathrm{s})$ are much greater than $\sqrt{\gamma^2 - 1}$ and much smaller than $\dfrac{1}{\sqrt{\gamma^2 - 1}}$ respectively, with $\gamma \triangleq \epsilon^{-1}$.

[19] $c(G)$ denotes the frequency dependent condition number of G with $c(G) = \dfrac{\overline{\sigma}(G)}{\underline{\sigma}(G)}$.

In addition to the previous results linking the initial loop shape specified by the use of the pre/post compensators and the final loop shape when the $\mathscr{H}_\infty$ controller has been added there are also results concerning the behavior of the closed loop performance & robustness objectives $S, SG, K_\mathrm{s}S, K_\mathrm{s}SG$. Briefly it can be said that bounds on these closed loop objectives can also be computed as a function of γ, G, W_1 and W_2 only[20].

In this section the standard McFarlane & Glover loop shaping design procedure (LSDP) has been detailed and some theoretical results justifying this procedure presented. In the context of gain-scheduled control considered in this work, this procedure will be used for the design of LTI controllers for LTI approximations of nonlinear systems around a set of equilibrium points in Chapter 6.

3.3.3 Full Order Case

In this section the solution to the sub-optimal robust stabilization problem of Theorem 3.3 linked to the LSDP of Section 3.3.2 will be given. For completeness, both the solution based on the classic Glover & Doyle formulation [53] and an LMI formulation will be detailed in the following two subsections.

Briefly the problem is to find a dynamic output feedback stabilizing controller K_∞ to the already shaped open loop plant $G_\mathrm{s} = W_2GW_1$ of Fig. 3.10a[21] that also satisfies the requirement:

$$\left\| \begin{bmatrix} K_\infty \\ \mathbb{I} \end{bmatrix} (\mathbb{I} - G_\mathrm{s}K_\infty)^{-1}\tilde{M}^{-1} \right\|_\infty \leq \gamma, \quad \gamma > 0. \tag{3.78}$$

Robustness margin

It is clear that if $\epsilon_{\max}$ is the maximum robustness margin that can be achieved, an $\epsilon \lesssim \epsilon_{\max}$ is chosen in order to construct the robust controller with $\gamma = 1/\epsilon$ and typically $2 < \gamma < 10$. The controller K_∞ will robustify the open loop plant G_s in the face of additive unstructured uncertainties acting on its left NCF's as presented in Section 3.3.1 (see Fig. 3.9, with K_∞ here being the robust controller K and $G_\Delta = \mathcal{F}_U(G_\mathrm{s}, \Delta)$ the perturbed plant).

As a final remark before actually giving the solution to Eq. 3.78, the maximum robustness margin $\epsilon_{\max}$ (or equivalently $\gamma_{\min}$) can be computed beforehand and before actually K_∞ and in an one step procedure for the *normalized* coprime factorization of G_s. In addition it depends *only* on the coprime factorization $[\tilde{M}, \tilde{N}]$ of G_s as[22]:

$$\epsilon_{\max} = \sqrt{1 - \left\| [\tilde{M}, \tilde{N}] \right\|_H^2}. \tag{3.79}$$

[20] For additional details see [97], Section IV or [96], Sections 6.5, 6.6.

[21] Note that now the *final* open loop plant of Theorem 3.3 is the one obtained by augmenting the *initial* open loop plant with the pre/post compensators of the LSDP of Section 3.3.2.

[22] The subscript 'H' denotes the Hankel norm of a system associated to its Hankel singular values.

3.3.3.1 Standard Solution

Consider the strictly proper[23] state space representation of the open loop shaped plant G_s (see Fig. 3.10a) satisfying the classic $\mathscr{H}_\infty$ assumptions needed (see [155], Ch. 16): Shaped system

$$G_\mathrm{s}: \quad \begin{aligned} \dot{x} &= \mathbf{A}_\mathrm{s} x + \mathbf{B}_\mathrm{s} u \\ y &= \mathbf{C}_\mathrm{s} x \end{aligned} \tag{3.80}$$

Then out of all stabilizing controllers achieving also the tolerance level γ, the so-called central or maximum entropy one K_∞ with: Feedback controller

$$K_\infty: \quad \begin{aligned} \dot{x}_k &= \mathbf{A}_k x_k + \mathbf{B}_k y \\ u &= \mathbf{C}_k x_k + \mathbf{D}_k y \end{aligned} \tag{3.81}$$

is given by:

$$\begin{aligned} K_\infty &\overset{\mathrm{ss}}{=} \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{C}_k & \mathbf{D}_k \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_\mathrm{s} - \mathbf{B}_\mathrm{s}\mathbf{B}_\mathrm{s}^T\mathbf{X} + \gamma^2\mathbf{W}^{-1}\mathbf{Z}\mathbf{C}_\mathrm{s}^T\mathbf{C}_\mathrm{s} & \gamma^2\mathbf{W}^{-1}\mathbf{Z}\mathbf{C}_\mathrm{s}^T \\ {\mathbf{B}_\mathrm{s}}^T\mathbf{X} & 0 \end{bmatrix}. \end{aligned} \tag{3.82}$$

The matrices $\mathbf{X}, \mathbf{Z}$ solve the *control* (respectively *filtering*) algebraic Riccati equations (CARE, FARE):

$$\text{CARE}: \quad \mathbf{A}_\mathrm{s}^T\mathbf{X} + \mathbf{X}\mathbf{A}_\mathrm{s} - \mathbf{X}\mathbf{B}_\mathrm{s}\mathbf{B}_\mathrm{s}^T\mathbf{X} + \mathbf{C}_\mathrm{s}^T\mathbf{C}_\mathrm{s} = 0 \tag{3.83}$$

$$\text{FARE}: \quad \mathbf{Z}\mathbf{A}_\mathrm{s}^T + \mathbf{A}_\mathrm{s}\mathbf{Z} - \mathbf{Z}\mathbf{C}_\mathrm{s}^T\mathbf{C}_\mathrm{s}\mathbf{Z} + \mathbf{B}_\mathrm{s}\mathbf{B}_\mathrm{s}^T = 0 \tag{3.84}$$

with the matrix $\mathbf{W}$ (entering $\mathbf{A}_k, \mathbf{B}_k$) given by:

$$\mathbf{W} = \left(\mathbb{I} + \mathbf{X}\mathbf{Z} - \gamma^2\mathbb{I}\right)^T. \tag{3.85}$$

The maximum robustness margin ϵ_max obtained can be also calculated using the solutions $\mathbf{X}, \mathbf{Z}$ of the Riccati equations as (with $\epsilon_\mathrm{max} < 1$ always): Optimal robustness margin

$$\epsilon_\mathrm{max} = \frac{1}{\sqrt{1 + \lambda_\mathrm{max}(\mathbf{X}\mathbf{Z})}}. \tag{3.86}$$

Technical Note: The problem of designing a sub-optimal robust output feedback controller following the McFarlane & Glover loop shaping design procedure (LSDP) is solved by MATLAB® with the macro '`ncfsyn`' of the Robust Control Toolbox. This macro designs a positive feedback controller K_∞ for an open loop plant G with corresponding pre/post compensators W_1, W_2. Additionally it computes the corresponding maximum stability margin ϵ_max, the stability margin ϵ achieved and the corresponding augmented open loop plant G_s and stabilized closed loop system transfer functions.

[23] The *strictly proper* case is considered here in order to simplify the formulas.

3.3.3.2 LMI Solution

In this section an alternative method for the solution of the sub-optimal robust stabilization corresponding to the LSDP of Section 3.3.2 is presented. This method is based on a LMI formulation of the problem of finding the stabilizing robust dynamic controller K_∞ that also satisfies Eq. 3.78.

The idea is simple enough: given that the problem of designing this stabilizing controller corresponds to the minimization of the $\mathscr{H}_\infty$ norm from the input disturbance w (representing the additive uncertainty over the normalized left coprime factors of the shaped open loop plant G_s) to the performance vector ζ_∞ consisting of the input u and the output y of G_s (see Fig. 3.9); then it suffices to design the controller K_∞ for the corresponding standard plant P_s of Fig. 3.11 in such a way that $\|T_{w\zeta_\infty}\|_\infty = \|\mathcal{F}_L(\mathbb{P}_\mathrm{s}, K_\infty)\|_\infty < \gamma$ with K_∞ stabilizing and γ minimized. The solution to this problem via LMI's could be done in more than one ways; here the basic approach found in [11] is presented.

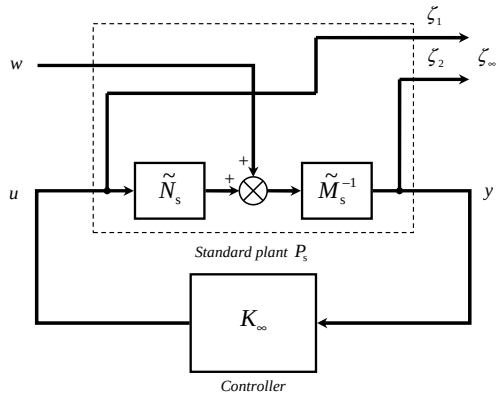


Figure 3.11: LFT of the LSDP.

To find the standard plant representation $\mathbb{P}_\mathrm{s}$ of the open loop shaped plant G_s, with a general standard plant representation defined as:

$$\begin{bmatrix} \dot{x} \\ \zeta_\infty \\ y \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_w & \mathbf{B}_u \\ \mathbf{C}_\zeta & \mathbf{D}_{w\zeta} & \mathbf{D}_{u\zeta} \\ \mathbf{C}_y & \mathbf{D}_{wy} & \mathbf{D}_{uy} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix} \tag{3.87}$$

NCF's

it suffices to obtain a minimal normalized LCF of the shaped open loop plant $G_\mathrm{s} = \left[\begin{smallmatrix} \mathbf{A}_\mathrm{s} & \mathbf{B}_\mathrm{s} \\ \mathbf{C}_\mathrm{s} & \mathbf{D}_\mathrm{s} \end{smallmatrix}\right]$[24]. These coprime factors of are given by [155] (with $G_\mathrm{s} = \tilde{M}^{-1}\tilde{N}$):

$$\tilde{N} \overset{\text{ss}}{=} \begin{bmatrix} \mathbf{A}_\mathrm{s} + \mathbf{L}\mathbf{C}_\mathrm{s} & \mathbf{L} \\ \mathbf{R}^{-1/2}\mathbf{C}_\mathrm{s} & \mathbf{R}^{-1/2} \end{bmatrix} \tag{3.88}$$

$$\tilde{M} \overset{\text{ss}}{=} \begin{bmatrix} \mathbf{B}_\mathrm{s} + \mathbf{L}\mathbf{D}_\mathrm{s} & \mathbf{L} \\ \mathbf{R}^{-1/2}\mathbf{D}_\mathrm{s} & \mathbf{R}^{-1/2} \end{bmatrix}. \tag{3.89}$$

[24] The more general case where the plant is not necessarily strictly proper is considered here with $\mathbf{A}_\mathrm{s} \in \mathbb{R}^{n\times n}, \mathbf{B}_\mathrm{s} \in \mathbb{R}^{n\times n_u}, \mathbf{C}_\mathrm{s} \in \mathbb{R}^{n_y\times n}, \mathbf{D}_\mathrm{s} \in \mathbb{R}^{n_y\times n_u}$.

After the necessary matrix manipulations the standard plant $\mathbb{P}_\mathrm{s}$ is found to be (one-to-one connection using Eq. 3.87): LSDP standard plant

$$\mathbb{P}_\mathrm{s} \overset{\mathrm{ss}}{=} \begin{bmatrix} \mathbf{A}_\mathrm{s} & -\mathbf{L}\mathbf{R}^{1/2} & \mathbf{B}_\mathrm{s} \\ \begin{bmatrix} 0 \\ \mathbf{C}_\mathrm{s} \end{bmatrix} & \begin{bmatrix} 0 \\ \mathbf{R}^{1/2} \end{bmatrix} & \begin{bmatrix} \mathbb{I}_{n_u} \\ \mathbf{D}_\mathrm{s} \end{bmatrix} \\ \mathbf{C}_\mathrm{s} & \mathbf{R}^{1/2} & \mathbf{D}_\mathrm{s} \end{bmatrix}. \tag{3.90}$$

The coprime factorization matrices $\mathbf{R}, \mathbf{L}$ in Eq. 3.90 are defined as:

$$\mathbf{R} = \mathbb{I} + \mathbf{D}_\mathrm{s}\mathbf{D}_\mathrm{s}^T \tag{3.91}$$

$$\mathbf{L} = -\big(\mathbf{B}_\mathrm{s}\mathbf{D}_\mathrm{s}^T + \mathbf{Z}\mathbf{C}_\mathrm{s}^T\big)\mathbf{R}^{-1} \tag{3.92}$$

where $\mathbf{Z} = \mathbf{Z}^T, \mathbf{Z} > 0$ is the solution to the (generalized) *filtering* algebraic equation (GFARE):

$$\begin{aligned} \text{GFARE}: \quad \big(\mathbf{A}_\mathrm{s} - \mathbf{B}_\mathrm{s}\mathbf{S}^{-1}\mathbf{D}_\mathrm{s}^T\mathbf{C}_\mathrm{s}\big)\mathbf{Z} + \mathbf{Z}\big(\mathbf{A}_\mathrm{s} - \mathbf{B}_\mathrm{s}\mathbf{S}^{-1}\mathbf{D}_\mathrm{s}^T\mathbf{C}_\mathrm{s}\big)^T \\ -\mathbf{Z}\mathbf{C}_\mathrm{s}^T\mathbf{R}^{-1}\mathbf{C}_\mathrm{s}\mathbf{Z} + \mathbf{B}_\mathrm{s}\mathbf{S}^{-1}\mathbf{B}_\mathrm{s}^T = 0 \end{aligned} \tag{3.93}$$

and

$$\mathbf{S} = \mathbb{I} + \mathbf{D}_\mathrm{s}^T\mathbf{D}_\mathrm{s}. \tag{3.94}$$

Given the standard plant $\mathbb{P}_\mathrm{s}$ the solution to the problem of finding the dynamic controller K_∞ is divided into two phases: the *feasibility problem* and *controller reconstruction*. The first phase involves the solution of a number of LMI's, which is the result of the effort of rendering the corresponding Bounded Real Lemma (BRL) convex, in order to obtain the problem's Lyapunov matrices[25]. If these exist, then the *initial* BRL becomes convex, the second phase has a meaning and the controller matrices may be computed by solving the *initial* BRL either as an LMI problem or symbolically (for the latter method see [68]).

Feasibility Problem: The following theorem gives the necessary and sufficient conditions for the existence of a sub-optimal full order output feedback $\mathscr{H}_\infty$ controller K for a general standard plant $\mathbb{P}$ like the one of Eq. 3.87[26]: Feasibility problem

Theorem 3.4. Under the hypothesis that the pairs $(\mathbf{A}, \mathbf{B}_u)$ and $(\mathbf{A}, \mathbf{C}_\mathbf{y})$ are stabilizable and detectable then there exists a full order dynamic controller K (as the one in Eq. 3.43) ensuring that the transfer function $T_{w\zeta_\infty}(s) = \mathcal{F}_L(\mathbb{P}, K)$ from the vector of disturbances w to a performance vector ζ_∞ will be stable and also the $\mathscr{H}_\infty$-norm of this transfer function will be less than a performance level γ ($\|T_{w\zeta_\infty}\|_\infty < \gamma$), if and only if there exist symmetric matrices $\mathbf{X}, \mathbf{Y}$ and a performance level $\gamma > 0$ satisfying the following three LMI's:

[25] This is equivalent to the solution of the two Riccati equations when using the standard problem formulation of the previous section.

[26] In the following analysis it will be assumed without loss of generality that $D_{uy} = 0$ in order to simplify the calculations.

$$\begin{bmatrix} \mathcal{N}_{\mathbf{X}} & 0 \\ 0 & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T & \mathbf{X}\mathbf{C}_\zeta^T & \mathbf{B}_w \\ \mathbf{C}_\zeta \mathbf{X} & -\gamma\mathbb{I} & \mathbf{D}_{w\zeta} \\ \mathbf{B}_w^T & \mathbf{D}_{w\zeta}^T & -\gamma\mathbb{I} \end{bmatrix} \begin{bmatrix} \mathcal{N}_{\mathbf{X}} & 0 \\ 0 & \mathbb{I} \end{bmatrix} < 0 \tag{3.95}$$

$$\begin{bmatrix} \mathcal{N}_{\mathbf{Y}} & 0 \\ 0 & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}^T\mathbf{Y} + \mathbf{Y}\mathbf{A} & \mathbf{Y}\mathbf{B}_w^T & \mathbf{C}_\zeta^T \\ \mathbf{B}_w^T \mathbf{Y} & -\gamma\mathbb{I} & \mathbf{D}_{w\zeta}^T \\ \mathbf{C}_\zeta & \mathbf{D}_{w\zeta} & -\gamma\mathbb{I} \end{bmatrix} \begin{bmatrix} \mathcal{N}_{\mathbf{Y}} & 0 \\ 0 & \mathbb{I} \end{bmatrix} < 0 \tag{3.96}$$

$$\begin{bmatrix} \mathbf{X} & \mathbb{I} \\ \mathbb{I} & \mathbf{Y} \end{bmatrix} \geq 0. \tag{3.97}$$

□

Proof. See [11], pp. 10-11.

■

The matrices $\mathcal{N}_{\mathbf{X}}, \mathcal{N}_{\mathbf{Y}}$ are (preferably orthonormal) bases to the null spaces of $\begin{bmatrix} \mathbf{B}_u^T & \mathbf{D}_{u\zeta}^T \end{bmatrix}$ and $\begin{bmatrix} \mathbf{C}_y & \mathbf{D}_{wy} \end{bmatrix}$ respectively. Now in practice, one tries to solve a minimization problem by trying to solve for the matrices $\mathbf{X}, \mathbf{Y}$ for the smallest γ possible; once this is done the controller matrices are obtained following the analysis below.

`Controller recon/tion`

Controller Reconstruction: The problem of obtaining the feasibility solutions involves in fact the convexifying of an initially BMI problem. This problem emerges by imposing satisfaction of the BRL for the closed loop system given by the l-LFT of the general standard plant $\mathbb{P}$ and the controller K. This problem can be translated into finding the matrix $\mathbf{\Theta} = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{C}_k & \mathbf{D}_k \end{bmatrix}$, regrouping the controller matrices, that satisfies the following inequality:

$$\mathbf{\Psi} + \mathbf{Q}^T \mathbf{\Theta}^T \mathbf{P} + \mathbf{P}^T \mathbf{\Theta} \mathbf{Q} < 0. \tag{3.98}$$

The aforementioned equation contains the matrices $\mathbf{\Psi}, \mathbf{P}, \mathbf{Q}$ that are generally dependent[27] on the known standard plant matrices *and* on $\mathbf{X}, \mathbf{Y}, \gamma$ obtained from the feasibility problem[28].

The initial problem of computing a robust dynamic controller K_∞ for the shaped open loop plant G_s can be in fact viewed as a standard $\mathscr{H}_\infty$ problem with a special structure on the corresponding standard plant $\mathbb{P}$. Given that the standard plant referring to this special problem $\mathbb{P}_\mathrm{s}$ may be computed using Eqs. 3.88-3.90, then the problem of designing K_∞ may be solved by directly applying the aforementioned LMI formulation of Theorem 3.4.

[27] For the exact dependence and how to solve for these matrices refer to [11].

[28] As a small technical note it may be added that the LMI of Eq. 3.98 can be readily solved using the macro '`basiclmi`' of the MATLAB® Robust Control Toolbox.

3.3.4 Static Case

In this section the problem of computing static instead of dynamic robust $\mathscr{H}_\infty$ controllers for the LSDP of Section 3.3.2 will be addressed. The problem in fact is to find a static output feedback control law $u = -K_\infty y$ for the shaped plant G_s of Fig. 3.10a that will stabilize the plant and also render it robust to left NCF perturbations; this is translated in the *two-block* $\mathscr{H}_\infty$ problem of Eq. 3.78 that is repeated here:

$$\left\| \begin{bmatrix} K_\infty \\ \mathbb{I} \end{bmatrix} (\mathbb{I} - G_\mathrm{s} K_\infty)^{-1} \tilde{M}^{-1} \right\|_\infty \leq \gamma, \quad \gamma > 0. \tag{3.99}$$

Static vs. dynamic $\mathscr{H}_\infty$ control

This approach of designing simpler controllers is very important in the context of gain scheduling, as it will be detailed in Chapter 6, since the controller updating procedure has a very reduced complexity. Indeed, from the discussion of Section 3.3.2 it has been made clear that the order of a dynamic controller K_s is equal to the sum of the orders of the plant plus the pre/post compensators'. Thus for example for a second order system G shaped by a PID post-compensator W_2 and a low pass filter W_1 as pre-compensator, the final controller would be of order five making things maybe more complicated than they should have been.

Two are the major solutions proposed for this problem of high controller order: the first is try to reduce the order of K_∞ (using for example techniques such as Hankel norm approximation or balanced truncation) whereas the second is attempting to compute reduced order controllers directly from the beginning; here this second method will be outlined.

The basic difficulty in designing reduced order controllers comes from the famous rank minimization condition for the existence of a controller of order $k < n$ (where n is the plant order):

$$\mathrm{Rank}(\mathbb{I} - \mathbf{XY}) \leq k. \tag{3.100}$$

Several heuristic methods have been proposed in order to take out this condition such as the alternative projection method, the XY-centering algorithm, the cone complementarity linearization method and others (see for example [52, 54] and references therein). However these approaches (along with the one trying to solve the initial problem, which is of course a BMI, with appropriate solvers) do not guarantee convergence even though they have been successfully applied to real world applications.

Other methods focus their effort to obtain only sufficient conditions for the existence of lower order controllers, conditions which however involve the formulation of the problem into LMI's. One of these methods has appeared in [111] and will be exploited in this work. Briefly, it involves the aforementioned *two-block* loop shaping problem itself and demands only the solution of two LMI's to guarantee the existence of a static controller K_∞. Another method that addresses the equivalent *four-block* problem (see Eq. 3.71) and claims to achieve better results than the previous one has been very recently proposed in [109].

Following the analysis in [111] and always under the LSDP context of the previous sections, consider the open shaped plant $G_s = \left[\begin{smallmatrix} \mathbf{A}_s & \mathbf{B}_s \\ \mathbf{C}_s & \mathbf{D}_s \end{smallmatrix}\right]$ of Figs. 3.10a-3.11. The problem this time is finding a *static* controller K_∞ and the following theorem gives sufficient conditions for the existence of such a controller.

LSDP: static $\mathscr{H}_\infty$ control feasibility

Theorem 3.5. There exist a static stabilizing controller K_∞ that also satisfies the *two-block* LSDP $\mathscr{H}_\infty$ robustness problem:

$$\left\| \begin{bmatrix} K_\infty \\ \mathbb{I} \end{bmatrix} (\mathbb{I} - G_s K_\infty)^{-1} \tilde{M}^{-1} \right\|_\infty \leq \gamma, \quad \gamma > 0.$$

if there exist $\gamma < 1$ and a matrix $\mathbf{X} = \mathbf{X}^T > 0$ that satisfies the LMI's:

$$(\mathbf{A}_s + \mathbf{L}\mathbf{C}_s)\mathbf{X} + \mathbf{X}(\mathbf{A}_s + \mathbf{L}\mathbf{C}_s)^T < 0 \tag{3.101}$$

$$\begin{bmatrix} \mathbf{A}_s\mathbf{X} + \mathbf{X}\mathbf{A}_s^T - \gamma\mathbf{B}_s\mathbf{B}_s^T & \mathbf{X}\mathbf{C}_s^T - \gamma\mathbf{B}_s\mathbf{D}_s^T & -\mathbf{L}\mathbf{E}^{1/2} \\ \mathbf{C}\mathbf{X} - \gamma\mathbf{D}_s\mathbf{B}_s^T & -\gamma\mathbf{E} & \mathbf{E}^{1/2} \\ -\mathbf{E}^{1/2}\mathbf{L}^T & \mathbf{E}^{1/2} & -\gamma\mathbb{I}_{n_y} \end{bmatrix} < 0 \tag{3.102}$$

with $\mathbf{L} = -\big(\mathbf{B}_s\mathbf{D}_s^T + \mathbf{Z}\mathbf{C}_s^T\big)\mathbf{R}^{-1}$, $\mathbf{Z} = \mathbf{Z}^T \geq 0$ being the solution to the GFARE of Eq. 3.93 and $\mathbf{R} = \mathbb{I} + \mathbf{D}_s\mathbf{D}_s^T, \mathbf{S} = \mathbb{I} + \mathbf{D}_s^T\mathbf{D}_s$.

□

Proof. See [111], pp. 1519.

■

Comments

As a first comment it should be stressed that the difficulty in computing a static controller comes from the fact that Eq. 3.101 is a BMI transformed to an LMI by neglecting an additive factor $\mathfrak{F}$ which is quadratic on $\mathbf{R}$, with $\mathfrak{F}$ being equal to[29]:

$$\mathfrak{F} = -\gamma\mathbf{R}\mathbf{C}_s^T\mathbf{E}^{-1}\mathbf{C}_s\mathbf{R}. \tag{3.103}$$

As a second comment it may be added that the procedure of arriving to the conditions of Eqs. 3.101-3.102 is nothing more than applying the standard LMI conditions of Theorem 3.4 to this particular static output feedback problem. As a result, the analysis of this section may be considered as an extension to the discussion of Section 3.3.3.2.

As a third comment it should be made clear that because of the fact that a much simpler structure controller has been designed in contrast with the full order case of the previous section, the performance level γ (respectively robustness margin ϵ) will be bigger (respectively smaller). In practice, and as it will be observed in Chapter 6, this trade-off between controller complexity and robust stability leans heavily towards the first; this meaning that the robustness margin deterioration in the static case is only about 30% for a seventh order open loop plant G_s, thus justifying the static approach.

[29] The simplification of this quadratic factor $\mathfrak{F}$ renders the conditions sufficient instead of necessary and sufficient in the beginning.

Static controller recon/tion

Now once the performance level γ and the Lyapunov matrix $\mathbf{X}$ have been found, the static output feedback gain can be computed following the *Controller Reconstruction* phase of Theorem 3.4 that involves the solution of the initial Bounded Real Lemma relevant to the problem. If $K_\infty = \tilde{K}_\infty(\mathbb{I} - \mathbf{D}_\mathrm{s}\tilde{K}_\infty)^{-1}$ then K_∞ may be computed by solving the LMI of Eq. 3.98 for $\tilde{K}_\infty$ with:

$$\mathbf{\Psi} = \begin{bmatrix} \mathbf{A}_\mathrm{s}\mathbf{X} + \mathbf{X}\mathbf{A}_\mathrm{s}^T & 0 & \mathbf{X}\mathbf{C}_\mathrm{s}^T & -\mathbf{L}\mathbf{E}^{1/2} \\ 0 & -\gamma\mathbb{I}_{n_u} & 0 & 0 \\ \mathbf{C}_\mathrm{s}\mathbf{X} & 0 & -\gamma\mathbb{I}_{n_y} & \mathbf{E}^{1/2} \\ -\mathbf{E}^{1/2}\mathbf{L}^T & 0 & \mathbf{E}^{1/2} & -\gamma\mathbb{I}_{n_y} \end{bmatrix} \tag{3.104}$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{B}_\mathrm{s} \\ \mathbb{I}_{n_u} \\ \mathbf{D}_\mathrm{s} \\ 0 \end{bmatrix}^T \tag{3.105}$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{C}_\mathrm{s}\mathbf{X} & 0 & 0 & \mathbf{E}^{1/2} \end{bmatrix}. \tag{3.106}$$

This concludes the discussion on the classic LSDP of McFarlane&Glover and on how to obtain full order or static controllers using the standard Glover&Doyle formulation or an alternative LMI approach. As a last comment it may be added that this LSDP analyzed in the last sections offers indeed a sound and theoretically justified approach but which on the same time preserves its intuitive nature concerning the synthesis of MIMO controllers for LTI plants. From the one side it can encompass many interesting features that could be useful for a control system (low/high open loop gains at high/low frequencies, treatment for the general I/O disturbance rejection problem, stabilization over unstructured uncertainties etc.) but on the other hand it can be seen, in the context of singular value shaping, as a generalization to well known loop shaping practices. In addition this synthesis procedure can be further simplified by considering *simpler* control structures for the robust closed loop controller; this fact is easily done due to the fact that the LSDP can be recast as a standard $\mathscr{H}_\infty$ synthesis problem.

3.4 The Gap Metric

In this section a powerful system analysis tool will be presented that will permit the development of advanced gain scheduling controllers in Chapter 6. This tool is called *the gap metric* and it has been primarily introduced in the control community with the work of El-Sakkary in the 80's [38, 39]. It has been mostly exploited by Georgiou&Smith both concerning computational aspects and its connection with $\mathscr{H}_\infty$ control [49, 50]. Other interesting metrics were conceived by Vidyasagar [143] and more recently by Vinnicombe [146]. Additional work in a nonlinear context has been done for example in [51] and in [34] for the LTV case.

3.4.1 Motivation & Definitions

Distance between systems

A major motivation for the development of the gap metric and other metrics was the desire to obtain a measure of the distance between two systems in a *closed loop* setting and the possible extension of this idea to study the stability of a perturbed system $G_\Delta(s)$ given that it is controlled by the same compensator $K(s)$ computed for a nominal system $G(s)$. Moreover this metric should be able to compare different types of systems: stable or unstable, of the same or different orders etc.; the only restriction is that they should have the same number of inputs or outputs since they are compared in the same closed loop setting (i.e. using the same compensator).

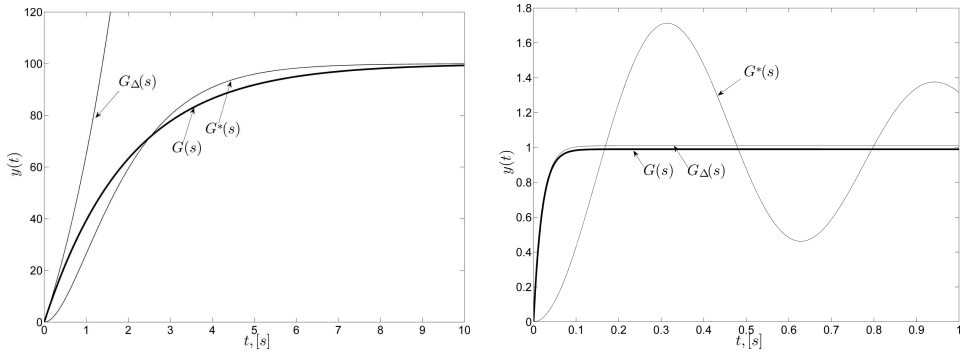
The gap metric captures differences between the closed loop behaviors of two plants $G(s), G_\Delta(s)$ when the same compensator $K(s)$ is applied. If another system $G^*(s)$ is considered, that may have similar *open loop* but very different *closed loop* when compared to G (always with the same compensator K), then this metric should be able to distinguish G^* from G_Δ which behaves well/similarly as G in *closed loop*, though it may be totally different in *open loop*[30]. As an illustrative example suppose that G is a stable first-order system, G_Δ an unstable one, whereas G^* is similar to G in terms of open loop step response:

$$G(s) = \frac{100}{2s+1},\ G_\Delta(s) = \frac{100}{2s-1},\ G^*(s) = \frac{100}{(s+1)^2}. \tag{3.107}$$

Suppose also that all plants are given a unity negative feedback ($K(s) = -1$); then their closed loop transfer functions $H(s), H_\Delta(s), H^*(s)$ are:

$$H(s) = \frac{100}{2s+101},\ H_\Delta(s) = \frac{100}{2s+99},\ H^*(s) = \frac{100}{s^2+2s+101} \tag{3.108}$$

and all open and closed loop step responses are shown in Figs. 3.12a, 3.12b.



(a) *Open loop step responses* **(b)** *Closed loop step responses*

Figure 3.12: System responses comparison.

[30]For example G may be stable and G_Δ unstable in open loop as it is the case in real life situations when a nominal but uncertain plant may become unstable under heavy perturbations.

The graph topology

The topology inducing the gap metric is depicted in Fig. 3.13; this topology takes into account the special structure of the feedback problem seen in Section 3.3. In this structure, a nominal plant G that is stabilized by a compensator K, receives input/output perturbations d, n whereas another plant G_Δ, receives the same perturbations while being in feedback interconnection with the same compensator K. The goal then is to find a distance measure for the inputs/output behaviors of each system G, G_Δ under this specific feedback form. This distance measure, that is defined for these two systems G, G_Δ, is called *the gap metric* $\delta_{\mathrm{g}}(G, G_\Delta)$ and it is the maximum of the two *directed gaps* $\vec{\delta}_{\mathrm{g}}(G, G_\delta), \vec{\delta}_{\mathrm{g}}(G_\Delta, G)$:

$$\delta_{\mathrm{g}}(G, G_\Delta) \triangleq \max\left\{\vec{\delta}_g(G, G_\Delta), \vec{\delta}_g(G_\Delta, G)\right\}. \tag{3.109}$$

In this closed loop structure of Fig. 3.13, the two systems G, G_Δ may be considered close if for all *appropriate* inputs u to the plant G there exists an *appropriate* input u_Δ that makes a certain norm small. This norm, being of key importance, can be found in the actual definition of the *directed gap* between two systems which is the following:

$$\vec{\delta}_{\mathrm{g}}(G, G_\Delta) \triangleq \sup_{\begin{bmatrix} u \\ y \end{bmatrix} \in \mathcal{G}_G} \inf_{\begin{bmatrix} u_\Delta \\ y_\Delta \end{bmatrix} \in \mathcal{G}_{G_\Delta}} \frac{\left\| \begin{bmatrix} y \\ u \end{bmatrix} - \begin{bmatrix} y_\Delta \\ u_\Delta \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} y \\ u \end{bmatrix} \right\|_2}. \tag{3.110}$$

These *appropriate* inputs u, u_Δ mentioned before, along with the corresponding outputs $y = Gu, y_\Delta = G_\Delta u_\Delta$ are said to belong to the *domain* $\mathcal{D}(\mathbf{M}_G)$ of a *multiplication operator* $\mathbf{M}_G$. This *domain* is defined as:

$$\mathcal{D}(\mathbf{M}_G) \triangleq \{u \in \mathscr{H}_2 : Gu \in \mathscr{H}_2\}. \tag{3.111}$$

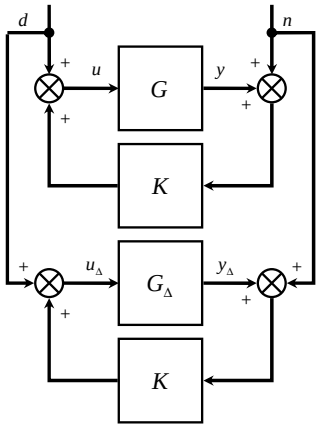


Figure 3.13: Gap metric related topology

The multiplication operator $\mathbf{M}_G$ is an operator on the input signal space $\mathcal{H}_2$, defining the signals being zero for $t < 0$ and with bounded energy for $t > 0$, and is described as:

$$\mathbf{M}_G : \mathscr{H}_2 \to \mathscr{H}_2,\ u \mapsto Gy. \tag{3.112}$$

where the system G belongs to the class of rational transfer functions with n_u inputs and n_y outputs respectively.

Returning to the definition of the directed gap of Eq. 3.110, it may be added that the gap metric can be also viewed as the distance between the so-called *graphs* of the two systems $\mathcal{G}_G, \mathcal{G}_{G_\Delta}$; this is the reason because it is a metric induced by the *graph topology*. Given that a system G may be seen as a multiplication operator $\mathbf{M}_G$, the graph of a system is defined as all the possible $\mathscr{H}_2$-bounded input/output pairs of the corresponding to G multiplication operator $\mathbf{M}_G$:

$$\mathcal{G}_G \equiv \mathcal{G}(\mathbf{M}_G) \triangleq \left\{ \begin{bmatrix} y \\ u \end{bmatrix} : u \in \mathcal{D}(\mathbf{M}_G) \right\}. \tag{3.113}$$

Gap metric properties

A very useful property of the gap metric δ_g, defined in Eq. 3.109 by means of the directed gap, is that given it is a metric the following hold[31]:

$$0 \leq \delta_g(G, G_\Delta) \leq 1 \tag{3.114}$$

and also:

$$\delta_g(G, G_\Delta) \equiv \delta_g(G_\Delta, G). \tag{3.115}$$

Returning to the initial motivation example, given that the gap metric shows the *closed loop closeness* between two systems under the same feedback structure, the gaps between all the systems are seen in Table 4.1. On the one hand it is verified that even though the systems G, G^* have similar open loop step responses (see Fig. 3.12a) their gap is large, showing exactly the incompatibility of the controller K for G^* in closed loop (see Fig. 3.12b). On the other hand, even though the open loop step response of G_Δ is unstable, therefore very different to the one corresponding to G, their gap is small since the compensator K is very good for both systems under the graph topology. In addition, because of the fact that G, G_Δ are very close by means of the gap whereas G, G^* are not, it is expected that the gap between G_Δ, G^* will be also large.

Table 3.1: Gaps concerning the motivation example.

System	$G(s)$	$G_\Delta(s)$	$G^*(s)$
$G(s)$	0	0.0205	0.8995
$G_\Delta(s)$	0.0205	0	0.8946
$G^*(s)$	0.8995	0.8946	0

[31] A large gap between two systems means that they are not close in the graph topology whereas a gap close to zero shows closed loop compatibility for an appropriate compensator K.

3.4.2 Connection to the $\mathscr{H}_\infty$ Theory

In this section the connection of the gap metric and the $\mathscr{H}_\infty$ loop shaping theory of Section 3.3 will be outlined. This important result obtained by Georgiou&Smith in [50] is of key interest since two systems G, G_Δ may be compared in a closed loop setting as in Fig. 3.13, with the second system G_Δ considered as a coprime factor perturbed of the initial one G. Quantitative results are then obtained on the maximum amount of uncertainty Δ that can be tolerated over G until the behavior of its closed loop starts to deteriorate rapidly.

The theory presented assumes *normalized* coprime factorizations of an open loop plant G and a stabilizing controller K which stabilizes and renders G robust over unstructured additive uncertainties Δ on its normalized coprime factors, exactly as in the setting of Fig. 3.9. It is argued by Georgiou&Smith that iff the $\mathscr{H}_\infty$-norm of the uncertainty introduced is less than the gap between the systems G, G_Δ then K stabilizes the perturbed system G_Δ. This result is formally given by the following theorem:

Gap metric & robust stab/tion

Theorem 3.6. Consider a system G with a *right* normalized coprime factorization $G = NM^{-1}$ and a controller K that stabilizes it. Take a real number ϵ so that $0 \leq \epsilon \leq 1$. Then these two statements are equivalent:

a. The closed loop pair $[G_\Delta, K]$ is stable for every plant G_Δ with $G_\Delta \triangleq (N + \Delta_N)(M + \Delta_M)^{-1}$ being a *right* NCF perturbed plant G, where $\Delta_N, \Delta_N \in \mathbb{R}\mathcal{H}_\infty$ and $\left\| \left[\begin{smallmatrix} \Delta_M \\ \Delta_N \end{smallmatrix} \right] \right\|_\infty < \epsilon$.

b. The closed loop pair $[G_\Delta, K]$ is stable for every plant G_Δ for which $\delta_{\mathrm{g}}(G, G_\Delta) < \epsilon$.

□

Proof. See [50], pp. 679.

■

Right vs. *left* NCF's

Now it is interesting to note that the theorem above is given in terms of *right* NCF's whereas the formulation of the robust stabilization problem stated in Theorem 3.3 is done in terms of *left* NCF's. It follows from the analysis in [50] that a controller K, being optimal for coprime factor perturbations, is optimal for both left *and* right factorizations of the plant G, even though these two types of uncertainty generate in general different classes of perturbed plants.

Given now that this controller K is optimal for both types of uncertainties, the connection with the $\mathscr{H}_\infty$ loop shaping theory is clear: a (sub)optimal controller K robustly stabilizes a plant G if the conditions of Theorem 3.3 hold, roughly if it satisfies (recall that $\mathbb{P}$ is the corresponding to G standard plant):

$$\|\mathcal{F}_L(\mathbb{P}, K)\|_\infty = \left\| \begin{bmatrix} K(\mathbb{I} - GK)^{-1}\tilde{M}^{-1} \\ (\mathbb{I} - GK)^{-1}\tilde{M}^{-1} \end{bmatrix} \right\|_\infty \leq \epsilon^{-1}. \tag{3.116}$$

Discussion on the gap

Then a simple and qualitative condition to verify if this controller K stabilizes a NCF perturbed plant G_Δ is to check whether the gap between G, G_Δ is less than the stability margin ϵ. Thus a compensator stabilizes a ball of uncertainty in the gap metric of given radius iff it stabilizes a ball of uncertainty of the same radius defined by perturbations on a normalized left coprime fraction.

It should be made clear that a perturbed plant at a distance *greater* than the stability margin will not necessarily be *destabilized* by some compensator that stabilizes the nominal plant with a stability margin equal to the gap between the two systems. This latter feature is the property of another metric: the ν-gap metric proposed by Vinnicombe in [145, 146].

As a last comment it should be added that this analysis can be extended in the context of the LSDP of Section 3.3.2. This means that the plants considered for the gap metric calculation are G_s (initial open loop plant) and $G_{\mathrm{s},\Delta}$ (the left NCF perturbed plant) with the robust controller K_∞ calculated for G_s.

3.4.3 Computation of the Gap Metric

In this final section the problem of actually computing the gap metric between two systems will be briefly addressed. For more details on the mathematics and on the method used to actually calculate the gap, the reader is encouraged to refer to [49, 50].

Computation of the gap

Even though the gap metric was introduced in [39], its actual computation for any given plants G, G_Δ was obtained by Georgiou&Smith in [49]. This is done first by noting that the gap metric $\delta_\mathrm{g}(G, G_\Delta)$ is computed as the maximum of the two directed gaps as in Eq. 3.109. Then, a quantitative expression for computing the directed gap stems from Eq. 3.110 by using the Commutant Lifting Theorem (see [50], pp. 674 and references therein):

$$\vec{\delta}_\mathrm{g}(G, G_\Delta) = \inf_{Q \in \mathscr{H}_\infty} \left\| \begin{bmatrix} M_G \\ N_G \end{bmatrix} - \begin{bmatrix} M_{G_\Delta} \\ N_{G_\Delta} \end{bmatrix} Q \right\|_\infty . \tag{3.117}$$

In the above equation M_i, N_i are right NCF of any plant G_i[32] and Q is related to the Youla parametrization procedure of any stabilizing controller K acting on G. Even though this equation does not give on its own any clear way on how to compute the gap, it may be further manipulated in order to conceive a traceable algorithm for this computation and it was a milestone getting a quantitative method to transform the graph topology-induced metric (see Eq. 3.110) into the exploitable expression in Eq. 3.117.

Technical Note: The gap metric and the ν-gap metric can be calculated to any desired accuracy by using MATLAB® Robust Control Toolbox and the command '`gapmetric`'. The calculation time needed is small enough (typically less than 500ms) to permit its use inside a robustness verification algorithm as it will be seen in Chapter 6.

[32] Recall that $G_i(s) \triangleq N_i(s) M_i^{-1}(s)$ and $M_i^* M_i + N_i^* N_i = \mathbb{I}$, with $M_i, N_i \in \mathbb{R}\mathcal{H}_\infty$.

3.5 Conclusions

In this chapter some topics of modern $\mathscr{H}_\infty$ theory were presented in order to give insights and to provide completeness in the technical part of this work.

In Section 3.1 the general framework of output feedback $\mathscr{H}_\infty$ control with pole placement in LMI regions was presented. It has been argued that this control synthesis approach is attractive since it permits the tailoring of the plant's closed loop dynamics in terms of eigenvalue placement, along with $\mathscr{H}_\infty$-based optimization on several closed loop objectives. The disadvantage of this method is the complexity of the controller obtained when full order convex synthesis is assumed.

In Section 3.2 some standard results on how to transform an arbitrary compensator into an equivalent observer-based state feedback controller were reviewed. It has been argued that in order to simplify the control structure in the context of linearization-based gain-scheduled control, it is preferable to consider strictly proper compensators with order equal to that of the plant in order to obtain a null Youla parameter Q. A drawback of this method is that the partition of the closed loop eigenvalues to the state feedback or the observer part of the controller is sometimes not so trivial.

In Section 3.3 some material on the classic McFarlane&Glover loop shaping design procedure (LSDP) was detailed. The problem of coprime factor robustness stabilization was also linked to the LSDP and the solution with a robust, either full order or static, controller was presented. The discussion involved the result that this procedure is particularly attractive in the context of gain scheduling control for two reasons: first it offers an intuitive but theoretically justified procedure of computing high performance controllers based on frequency domain analysis and second because it gives the means to link stability of a plant as a function of the uncertainty introduced on its coprime factors.

In Section 3.4 finally, a tool (the gap metric) linking stability and uncertainty was presented. The gap metric is in fact a norm under a closed loop feedback setting and it can give information on the stabilizability of a NCF perturbed plant by a controller computed for a nominal plant. With this tool the classic LSDP can be linked with gain scheduling as it will be seen in Chapter 6: a LTI plant calculated around an equilibrium point near (or not too near) another 'nominal' one may be seen as a 'perturbed' plant. Thus the gap between these two plants could give valuable information on the stabilizability of the second plant using a controller calculated for the nominal one.

Part II

Application examples.

Chapter 4

Modeling & Analysis

Overview

The gain scheduling methods of Chapter 1 that were judged to be the more promising and also offering the best background for further development were tested using two different benchmark examples. The first one is an analytic nonlinear model of the pitch axis dynamics of a highly manoeuvrable missile called the *Reichert Missile Benchmark (R'm'B)*. The second one is a tabulated nonlinear example of an atmospheric reentry vehicle *(ARV)* provided by the EADS Astrium Space Transportation corporation. This chapter gives the results obtained from the application of the first two steps of the Linearization-based Gain Scheduling Procedure (LBGS) of Section 1.3.1 (trimming and linearization) on these systems. Given that these two steps are common for any candidate for gain scheduling nonlinear system, the same analysis techniques were used and the results are presented in a similar way.

Chapter contents

4.1 The Reichert Benchmark Missile Model

The Reichert Missile Benchmark (R'm'B) was first presented in the control literature in the early 90's (see [112]) and has been the benchmark system for many works since, mostly due to the fact that it incorporates analytic formulas for the aerodynamic functions of the system. In this monograph, a similar but more recent version of the model appearing in [103] will be preferred.

4.1.1 Airframe Modeling

`System dynamics`

The nonlinear model of the R'm'B describes the longitudinal (or pitch) dynamics of a highly manoeuvrable missile airframe around its center of mass. The state vector x of the missile (see Fig. 4.1)[1] is its angle of attack α (in rad) and pitch rotational rate q (in rad$\cdot$s^{-1}). The command is the elevator deflection angle δ (in rad), the output is the vertical acceleration η (in g's) and its Mach number M is considered as an internal time varying parameter[2]. The state dynamics of the missile are given by:

$$\frac{d\alpha}{dt} = K_\alpha M C_{\mathrm{n}}(\alpha, M, \delta)\cos\alpha + q \tag{4.1}$$

$$\frac{dq}{dt} = K_{\mathrm{pr}} M^2 C_{\mathrm{m}}(\alpha, M, \delta) \tag{4.2}$$

whereas the output dynamics are:

$$\eta = K_\eta M^2 C_{\mathrm{n}}(\alpha, M, \delta)\,. \tag{4.3}$$

`Aerodynamic functions`

The lift force and pitching moment aerodynamic functions C_{n}, C_{m} are described by the following equations in standard notation:

$$C_{\mathrm{n}}(\alpha, M, \delta) = C_{\mathrm{n}\alpha}(\alpha, M)\,\alpha + C_{\mathrm{n}\delta}\delta \tag{4.4}$$

$$C_{\mathrm{m}}(\alpha, M, \delta) = C_{\mathrm{m}\alpha}(\alpha, M)\,\alpha + C_{\mathrm{m}\delta}\delta \tag{4.5}$$

with

$$C_{\mathrm{n}\alpha}(\alpha, M) = \left(\frac{180}{\pi}\right)^3 a_{\mathrm{n}}\alpha^2 + \left(\frac{180}{\pi}\right)^2 b_{\mathrm{n}}|\alpha| + \frac{180}{\pi} c_{\mathrm{n}}\left(2 - \frac{M}{3}\right) \tag{4.6}$$

$$C_{\mathrm{m}\alpha}(\alpha, M) = \left(\frac{180}{\pi}\right)^3 a_{\mathrm{m}}\alpha^2 + \left(\frac{180}{\pi}\right)^2 b_{\mathrm{m}}|\alpha| + \frac{180}{\pi} c_{\mathrm{m}}\left(-7 + \frac{8M}{3}\right) \tag{4.7}$$

and

$$C_{\mathrm{n}\delta} = \frac{180}{\pi} d_{\mathrm{n}} \tag{4.8}$$

$$C_{\mathrm{m}\delta} = \frac{180}{\pi} d_{\mathrm{m}}. \tag{4.9}$$

[1] The symbols $G_{\mathrm{m}}, G_{\mathrm{p}}$ correspond to the missile's center of mass and center of pressure.

[2] Explicit time dependence will be dropped when needed for the sake of simplicity.

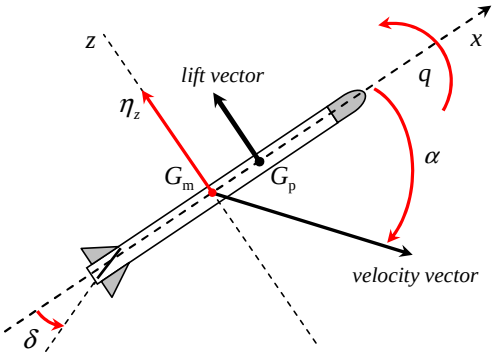


Figure 4.1: Missile pitch view

Table 4.1: Missile & actuator coefficients.

Name	Symbol	Expression	Value	Unit
Reference area	S	-	0.04088	m^2
Diameter	d	-	0.2286	m
Mass	m	-	204.02	kg
Moment of inertia	I_{yy}	-	247.44	$\mathrm{kg} \cdot \mathrm{m}^2$
Static pressure	P_0	-	46601.6	$\mathrm{N/m}^2$
Speed of sound	v_{s}	-	315.89	m/s
Drag coefficient	C_{a}	-	-0.3	-
Damping ratio	ξ	-	0.7	-
Natural frequency	ω_{a}	-	150	rad/s
-	K_α	$0.7P_0S/mv_{\mathrm{s}}$	0.02069	s^{-1}
-	K_{pr}	$0.7P_0Sd/I_{\mathrm{yy}}$	1.23194	s^{-2}
-	K_η	$0.7P_0S/mg$	0.66624	-
-	A_{x}	$0.7P_0SC_{\mathrm{a}}/I_{\mathrm{yy}}$	-1.96074	N/m
-	a_{n}	-	0.000103	deg^{-3}
-	b_{n}	-	-0.00945	deg^{-2}
-	c_{n}	-	-0.1696	deg^{-1}
-	d_{n}	-	-0.034	deg^{-1}
-	a_{m}	-	0.000215	deg^{-3}
-	b_{m}	-	-0.0195	deg^{-2}
-	c_{m}	-	0.051	deg^{-1}
-	d_{m}	-	-0.206	deg^{-1}

[(i)] The altitude is considered constant ($\simeq 6100m$).

The missile is considered to be operating during the terminal target intercepting phase with its engine thrust equal to zero and the Mach profile given by the following nonlinear differential equation: Mach trajectory

$$\frac{dM}{dt} = \frac{1}{v_s}\big(-|\eta|\sin|\alpha| + A_x M^2 \cos\alpha\big), \text{ with } M(0) = M_0. \tag{4.10}$$

The elevator fin is driven by an actuator modeled using the following second order filter (δ_c (in rad) is the control signal provided by the autopilot): Actuator dynamics

$$\frac{d^2\delta}{dt} + 2\xi\omega_a\frac{d\delta}{dt} + \omega_a^2\delta = \omega_a^2\delta_c. \tag{4.11}$$

The actuator and missile data coefficients are shown in Table 4.1. It should be noted that the latter are generally dependent on the flight altitude that is here considered as constant. The nonlinear mathematical model of the missile is valid for $-20^\circ \leq \alpha \leq 20^\circ$ and for $1.5 \leq M \leq 3$; these two variables forming its flight envelope.

The aerodynamic functions related to the angle of attack $C_{n\alpha}, C_{m\alpha}$ for $\alpha > 0$[3] are shown in Figs. 4.2a, 4.2b. It can be observed that there exists a significant variation of the functions values over α and M.

4.1.2 Trim Analysis

In this section the application of the first step of the Linearization-based Gain Scheduling Procedure (LBGS), concerning the missile trim control computation, will be detailed. The trim control δ_r is the rudder reference deflection angle needed in order to stabilize the missile around an equilibrium (or reference) point in the absence of external perturbations.

The equilibrium points can be parameterized as function of the angle of attack α or the vertical acceleration η, and the Mach number M. Each pair, α, M or η, M, forms the so-called scheduling vector ϱ used to describe the flight envelope of the missile.

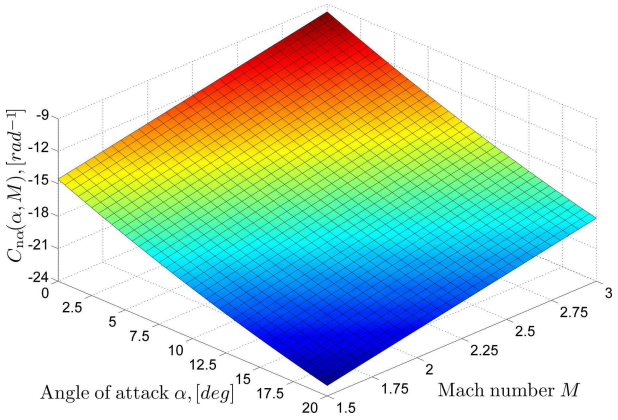
4.1.2.1 Parametrization on α

The trim control $\delta(\varrho_r) = \delta_r$ for each value of the scheduling vector $\varrho_r = [\alpha_r\ M_r]^T$ inside the flight envelope specifications ($-20^\circ \leq \alpha_r \leq 20^\circ$ and $1.5 \leq M_r \leq 3$) can be calculated easily using Eqs. 4.2, 4.5, 4.7 and 4.9. Given that the airframe is on equilibrium for a given value ϱ_r, then $\frac{dq}{dt}\big|_r = 0$[4] and so: Trim input

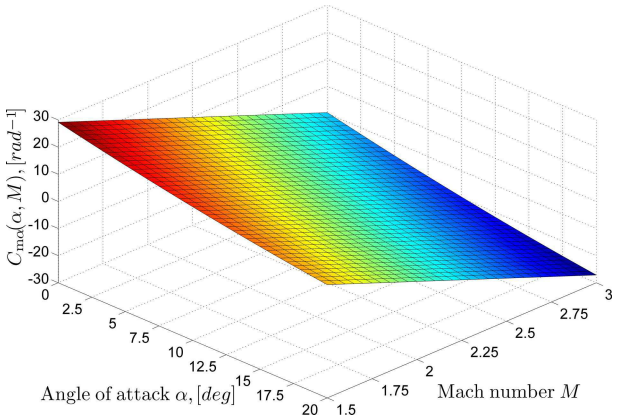
$$\delta(\varrho_r) = -\frac{C_{m\alpha}(\varrho_r)}{C_{m\delta}}\alpha_r. \tag{4.12}$$

[3] For $\alpha < 0$ the functions are symmetric due to $|\alpha|$ entering in Eqs. 4.6, 4.7.

[4] The 'r' notation means calculation on a *reference-equilibrium* point.



(a) *Lift aerodynamic function* $C_{\mathrm{n}\alpha}$



(b) *Pitching moment aerodynamic function* $C_{\mathrm{m}\alpha}$

Figure 4.2: Missile aerodynamic function surfaces.

Furthermore, the corresponding trim values $q(\varrho_\mathrm{r}) = q_\mathrm{r}$ and $\eta(\varrho_\mathrm{r}) = \eta_\mathrm{r}$ can be calculated by letting $\frac{d\alpha}{dt}\big|_\mathrm{r} = 0$ and then substituting Eq. 4.12 into Eqs. 4.1 and 4.3 respectively: Trim outputs

$$\begin{aligned} q(\varrho_\mathrm{r}) &= -K_\alpha M_\mathrm{r} C_\mathrm{n}(\alpha_\mathrm{r}, M_\mathrm{r}, \delta_\mathrm{r}) \cos\alpha_\mathrm{r} \\ &= -K_\alpha M_\mathrm{r} \Big[C_{\mathrm{n}\alpha}(\varrho_\mathrm{r}) - \frac{C_{\mathrm{n}\delta}}{C_{\mathrm{m}\delta}} C_{\mathrm{m}\alpha}(\varrho_\mathrm{r}) \Big] \alpha_\mathrm{r} \cos\alpha_\mathrm{r} \end{aligned} \tag{4.13}$$

and

$$\begin{aligned} \eta(\varrho_\mathrm{r}) &= K_\eta M_\mathrm{r}^2 C_\mathrm{n}(\alpha_\mathrm{r}, M_\mathrm{r}, \delta_\mathrm{r}) \cos\alpha_\mathrm{r} \\ &= K_\eta M_\mathrm{r}^2 \Big[C_{\mathrm{n}\alpha}(\varrho_\mathrm{r}) - \frac{C_{\mathrm{n}\delta}}{C_{\mathrm{m}\delta}} C_{\mathrm{m}\alpha}(\varrho_\mathrm{r}) \Big] \alpha_\mathrm{r}. \end{aligned} \tag{4.14}$$

The results (3D and contour maps) of the trim procedure are visualized in Figs. 4.3a-4.3f in the next page. It may be observed that for positive values of the angle of attack, the corresponding trim control is negative, the trim pitch rate positive, and the trim output negative. For negative angles of attack the results are of course symmetric.

4.1.2.2 Parametrization on η

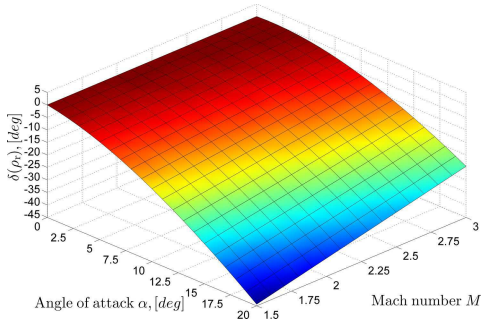
The parametrization of the trim control using the angle of attack α described previously (see Section 4.1.2.1) is not preferable since α is usually not measured. The variable that is actually measured (using accelerometers) is the output of the plant η. As a result the trim control δ_r should be re-parameterized in terms of a new scheduling vector $\varrho = [\eta\ M]^T$ for every equilibrium point. To do this, the following procedure is used: Trimming algorithm

1. *Trim Control:* Express the trim control δ_r as a function of the new scheduling vector $\varrho_\mathrm{r} = [\eta_\mathrm{r}\ M_\mathrm{r}]^T$ and the corresponding trim value for the angle of attack $\alpha(\varrho_\mathrm{r}) = \alpha_\mathrm{r}$ that is not known for the moment. To do this use Eq. 4.3 along with Eqs. 4.4, 4.6 and 4.8. Trim input

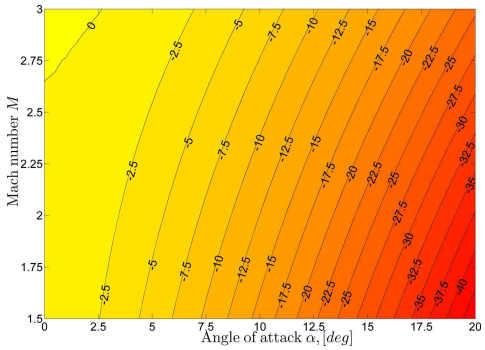
$$\delta(\eta_\mathrm{r}, M_\mathrm{r}, \alpha_\mathrm{r}) = \frac{\dfrac{\eta_\mathrm{r}}{K_\eta M_\mathrm{r}^2} - C_{\mathrm{n}\alpha}(\alpha_\mathrm{r}, M_\mathrm{r})\alpha_\mathrm{r}}{C_{\mathrm{n}\delta}} \tag{4.15}$$

2. *Angle of Attack:* Supposing that the system is on equilibrium (briefly $\dot{x}|_\mathrm{r} \triangleq 0$ and so the left hand sides of Eqs. 4.1, 4.2 go to zero), replace $\delta(\eta_\mathrm{r}, M_\mathrm{r}, \alpha_\mathrm{r})$ of Eq. 4.15 into the pitch rate equation (see Eq. 4.2) obtaining:

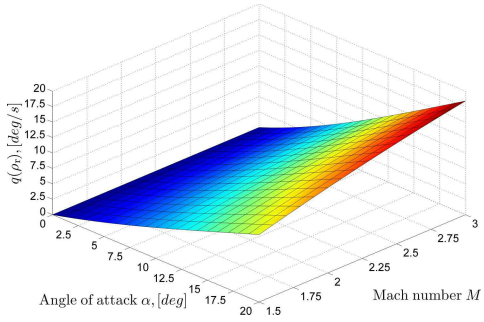
$$\begin{aligned} 0 &= C_\mathrm{m}(\alpha_\mathrm{r}, M_\mathrm{r}, \delta_\mathrm{r}) \\ &= C_{\mathrm{m}\alpha}(\alpha_\mathrm{r}, M_\mathrm{r})\alpha_\mathrm{r} + C_{\mathrm{m}\delta}\delta(\eta_\mathrm{r}, M_\mathrm{r}, \alpha_\mathrm{r}) \\ &= C_{\mathrm{m}\alpha}(\alpha_\mathrm{r}, M_\mathrm{r})\alpha_\mathrm{r} + \frac{C_{\mathrm{m}\delta}}{C_{\mathrm{n}\delta}} \Big(\frac{\eta_\mathrm{r}}{K_\eta M_\mathrm{r}^2} - C_{\mathrm{n}\alpha}\alpha_\mathrm{r} \Big). \end{aligned} \tag{4.16}$$



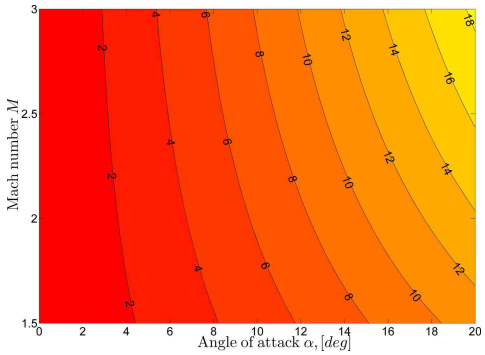
(a) *Trim control surface*



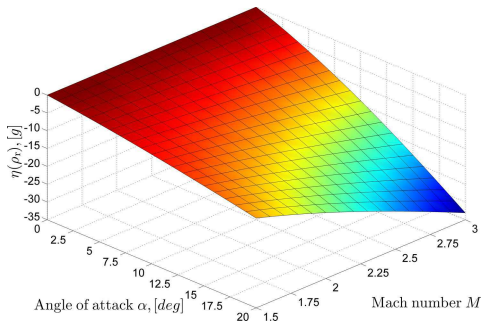
(b) *Trim control contour map*



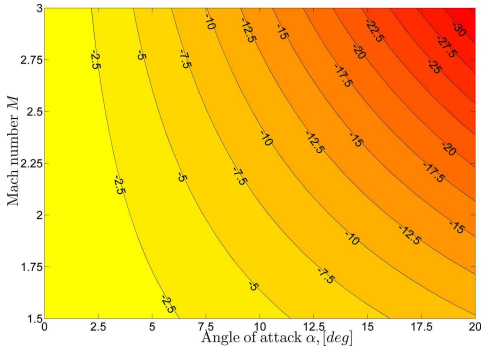
(c) *Trim pitch rate surface*



(d) *Trim pitch rate contour map*



(e) *Trim vertical acceleration surface*



(f) *Trim vertical acceleration contour map*

Figure 4.3: Missile trim results -*parametrization on* α

From the last expression, the following third order polynomial equation for $\alpha_{\rm r}$ as a function of the scheduling vector variables $\eta_{\rm r}, M_{\rm r}$ is taken:

$$\left(a_{\rm m} - \frac{d_{\rm m}}{d_{\rm n}}a_{\rm n}\right)\left(\frac{180}{\pi}\alpha_{\rm r}\right)^3 + \operatorname{sgn}(\alpha_{\rm r})\left(b_{\rm m} - \frac{d_{\rm m}}{d_{\rm n}}b_{\rm n}\right)\left(\frac{180}{\pi}\alpha_{\rm r}\right)^2 + \\ + \left[c_{\rm m}\left(-7 + \frac{8M_{\rm r}}{3}\right) - \frac{d_{\rm m}}{d_{\rm n}}c_{\rm n}\left(2 - \frac{M_{\rm r}}{3}\right)\right]\left(\frac{180}{\pi}\alpha_{\rm r}\right) + \frac{d_{\rm m}}{d_{\rm n}K_\eta}\frac{\eta_{\rm r}}{M_{\rm r}^2} = 0 \tag{4.17}$$

or in a more compact, $\varrho_{\rm r}$-dependent form:

$$k_1\alpha_{\rm r}^3 + k_2\operatorname{sgn}(\alpha_{\rm r})\alpha_{\rm r}^2 + k_3(\varrho_{\rm r})\alpha_{\rm r} + k_4(\varrho_{\rm r}) = 0. \tag{4.18}$$

Finally, because of the fact that $\operatorname{sgn}(\alpha_{\rm r}) = -\operatorname{sgn}(\eta_{\rm r})$ (see Figs. 4.3e, 4.3f), the last equation can be written: Trim AoA

$$k_1\alpha_{\rm r}^3 - k_2\operatorname{sgn}(\eta_{\rm r})\alpha_{\rm r}^2 + k_3(\varrho_{\rm r})\alpha_{\rm r} + k_4(\varrho_{\rm r}) = 0. \tag{4.19}$$

The previous polynomial equation can be solved for $\alpha_{\rm r}$, for each value of $\varrho_{\rm r}$ using either the classic method of Cardano or numerical root finding methods. In either case one will get three solutions for $\alpha_{\rm r}$; however only one has a physical sense[5]. For every $\varrho_{\rm r}$, one solution has always the opposite sign than expected whereas another one violates the flight envelope constraints taken over α (see Section 4.1.2.1). The acceptable solution is shown in Figs. 4.4a, 4.4b[6].

3. Pitch Rate: Since the trim value $\alpha_{\rm r}$ is computed by solving Eq. 4.19 for every $\varrho_{\rm r}$, the corresponding trim control $\delta_{\rm r}$ may be calculated by replacing $\alpha_{\rm r}$ into Eq. 4.15. In addition, the trim pitch rate values can be also found by replacing $\alpha_{\rm r}, \delta_{\rm r}$ into Eq. 4.1 given that $\frac{d\alpha}{dt}\big|_{\rm r} = 0$: Trim pitch rate

$$q_{\rm r} = -K_\alpha M_{\rm r} C_{\rm n}(\alpha_{\rm r}, M_{\rm r}, \delta_{\rm r})\cos\alpha_{\rm r}. \tag{4.20}$$

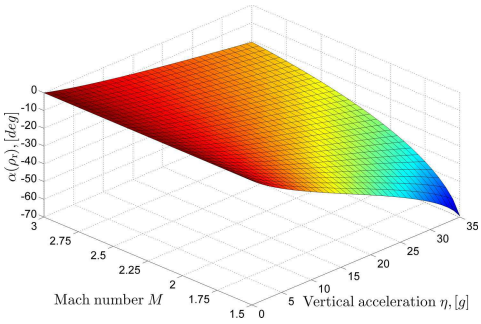
The trim control and trim pitch rate when using the η-parametrization are shown in Figs. 4.4c-4.4f.

The trim control $\delta(\varrho_{\rm r})$ is needed as a necessary part of a gain-scheduled control law in order to ensure proper reference point tracking. For implementation of such control laws, on line computation of $\delta(\varrho_{\rm r})$ is unrealistic since it involves real time solution of the aforementioned polynomial equation (see Eq. 4.19). For this reason, the trim control is calculated off-line for a sufficient number of points and the results are stored in a look-up table. Linear interpolation is then used to provide an appropriate value for every other point $\varrho_{\rm r}$ of the flight envelope[7].

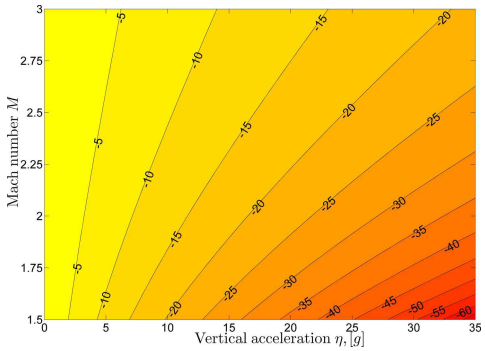
[5] For the singular case $\eta_{\rm r} = 0$, the solution considered is $\alpha_{\rm r} = 0$.

[6] Only positive values for $\eta_{\rm r}$ are considered; for negative ones the results are symmetric.

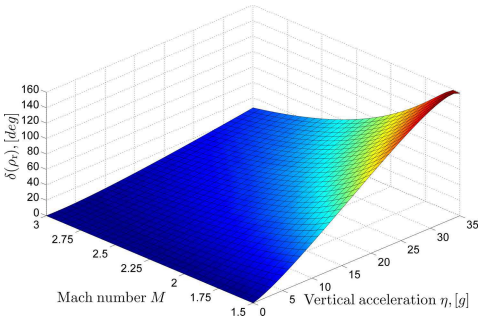
[7] Here a total number of $66 \times 66 = 4356$ points was used.



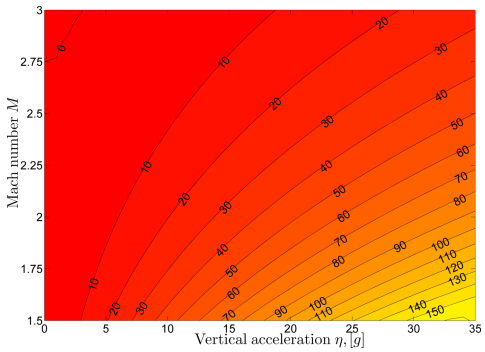
(a) *Trim angle of attack surface*



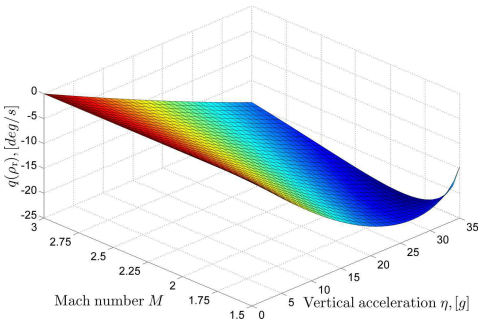
(b) *Trim angle of attack contour map*



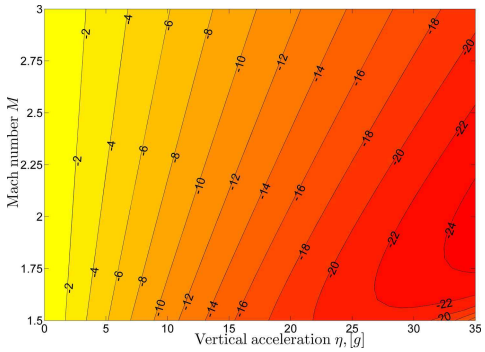
(c) *Trim control surface*



(d) *Trim control contour map*



(e) *Trim pitch rate surface*



(f) *Trim pitch rate contour map*

Figure 4.4: Missile trim results - *parametrization on* η

4.1.2.3 Flight Envelope Analysis

Most works concerning the R'm'B model lack a thorough analysis of the missile's flight envelope (see [17, 36, 55, 81, 99]). This is probably due to the fact that the flight envelope is directly parameterized using α, M and not η, M that is more realistic, since α is not available for feedback[8]. However, all the operating constraints are initially imposed on the angle of attack and the Mach, as presented in Section 4.1.1, defining the corresponding $[\alpha, M]$-dependent flight envelope $\mathbf{\Gamma}_{\text{fe}}^{[\alpha,M]}$:

$[\alpha, M]$ `flight envelope`

$$\mathbf{\Gamma}_{\text{fe}}^{[\alpha,M]} : \big[|\alpha| \leq 20^\circ, 1.5 \leq M \leq 3\big]. \tag{4.21}$$

The flight envelope can be re-parameterized in terms of η, M, using the analysis of Section 4.1.2. The result is a non convex hull as it can be seen in Fig. 4.4b (for $\eta_{\text{r}} > 0$), with the isoline $\alpha = -20^\circ$ setting the right border of the envelope. An analytic expression $\eta_{\text{fe}}(M)$ for this isoline can be easily found by setting $\alpha = -20^\circ$ in Eq. 4.17 (symmetric results are obtained for for $\eta_{\text{r}} < 0$):

$$\eta_{\text{fe}}(M) \simeq -0.454M^3 + 5.035M^2. \tag{4.22}$$

The $[\eta, M]$-dependent flight envelope $\mathbf{\Gamma}_{\text{fe}}^{[\eta,M]}$ is now given by Eq. 4.23 and is visualized in Fig. 4.5 (yellow surface). A convex linear approximation $\mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]}$ (yellow plus red surface) will be used from now on to simplify the shape of the flight envelope in order to make the task of interpolation easier and is given by Eq. 4.24:

$[\eta, M]$ `flight envelopes`

$$\mathbf{\Gamma}_{\text{fe}}^{[\eta,M]} : \big[0 \leq \eta \leq \eta_{\text{fe}}(M), 1.5 \leq M \leq 3\big] \tag{4.23}$$

$$\mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]} : \big[0 \leq \eta \leq \eta_{\text{fe,lin}}(M), 1.5 \leq M \leq 3\big] \tag{4.24}$$

with

$$\eta_{\text{fe,lin}}(M) \simeq 15.506M - 13.462. \tag{4.25}$$

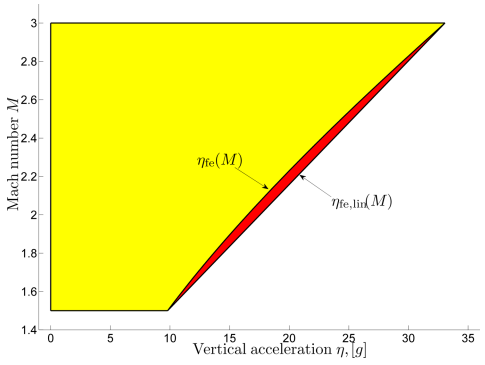


Figure 4.5: Missile flight envelope

[8] In practice, an estimator could be used to obtain α but this results to greater complexity.

The discussion of this section showed that if a careful analysis of the operating domain is not performed according to the initial nonlinear system constraints, redundancy will occur. Indeed, if a rectangular flight envelope had been used as in Section 4.1.2, the surface redundancy with respect to $\mathbf{\Gamma}_{\text{fe}}^{[\eta,M]}$ would have been around 60%, whereas with the linear approximation $\mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]}$ it is only 3.6%. This surface redundancy is particularly important for a gain-scheduled controller since it can significantly augment the number of synthesis points and hence the interpolation complexity.

4.1.3 System Linearization

After the analysis of Sections 4.1.2.1-4.1.2.3 and the parametrization of the equilibrium points of the missile in terms of the scheduling vector $\varrho = [\eta\ M]^T$, the second step of the Linearization-based Gain Scheduling procedure (LBGS) concerning linearization will now be detailed according to the standard analysis of Section 1.3.1.

4.1.3.1 LTI Models

LTI models

The goal here is to provide an LPV model of the missile's nonlinear dynamics (see Eqs. 4.1-4.9) smoothly parameterized by the scheduling vector $\varrho = [\eta\ M]^T$ with $\varrho \in \mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]}$ and the corresponding equilibrium manifold information obtained from the trim analysis. For notational simplicity, frozen instances of the LPV model will be considered (with 'r' meaning frozen equilibrium-reference operation):

$$\dot{x}_\delta = \mathbf{A}(\varrho_{\text{r}})x_\delta + \mathbf{B}(\varrho_{\text{r}})\delta_\delta \tag{4.26}$$
$$y_\delta = \mathbf{C}(\varrho_{\text{r}})x_\delta + \mathbf{D}(\varrho_{\text{r}})\delta_\delta \tag{4.27}$$

with $x = [\alpha\ q]^T, y = [\eta\ q]^T$ and

$$x_\delta = x - x(\varrho_{\text{r}}) \tag{4.28}$$
$$\delta_\delta = \delta - \delta(\varrho_{\text{r}}) \tag{4.29}$$
$$y_\delta = y - y(\varrho_{\text{r}}). \tag{4.30}$$

The linear systems' matrices are computed using Jacobian linearization of the initial nonlinear system dynamics, for any desired value ϱ_{r} of the scheduling vector inside the flight envelope:[9]

$$\left[\begin{array}{c|c} \mathbf{A}(\varrho_{\text{r}}) & \mathbf{B}(\varrho_{\text{r}}) \\ \hline \mathbf{C}(\varrho_{\text{r}}) & \mathbf{D}(\varrho_{\text{r}}) \end{array}\right] \triangleq \begin{bmatrix} \nabla_{x,\text{r}} f_x & \nabla_{\delta,\text{r}} f_x \\ \nabla_{x,\text{r}} h_y & \nabla_{\delta,\text{r}} h_y \end{bmatrix} \tag{4.31}$$

[9]For notational simplicity, $f_x = [f_\alpha, f_q]^T : \mathbb{R}^4 \to \mathbb{R}^2$ and $h_y = [h_\eta, q]^T : \mathbb{R}^3 \to \mathbb{R}$ are the nonlinear functions of the missile's state and output dynamics (Eqs. 4.1-4.3).

with: Jacobians

$$\nabla_x f_x = \begin{pmatrix} \nabla_\alpha f_\alpha & \nabla_q f_\alpha \\ \nabla_\alpha f_q & \nabla_q f_q \end{pmatrix} \tag{4.32}$$

$$\nabla_\delta f_x = \begin{pmatrix} \nabla_\delta f_\alpha \\ \nabla_\delta f_q \end{pmatrix} \tag{4.33}$$

$$\nabla_x h_y = \begin{pmatrix} \nabla_\alpha h_\eta & \nabla_q h_\eta \\ 0 & 1 \end{pmatrix} \tag{4.34}$$

$$\nabla_\delta h_y = \begin{pmatrix} \nabla_\delta h_\eta \\ 0 \end{pmatrix}. \tag{4.35}$$

The partial derivatives entering all the previous equations can be explicitly computed using the following formulas[10]:

$$\nabla_\alpha f_\alpha = K_\alpha M \Big[\cos\alpha \big(C_{\mathrm{n}\alpha} + \alpha \nabla_\alpha C_{\mathrm{n}\alpha} \big) - \sin\alpha C_{\mathrm{n}} \Big] \tag{4.36}$$

$$\nabla_q f_\alpha = 1 \tag{4.37}$$

$$\nabla_\alpha f_q = K_{\mathrm{pr}} M^2 \big(C_{\mathrm{m}\alpha} + \alpha \nabla_\alpha C_{\mathrm{m}\alpha} \big) \tag{4.38}$$

$$\nabla_q f_q = 0 \tag{4.39}$$

$$\nabla_\delta f_\alpha = K_\alpha M C_{\mathrm{n}\delta} \cos\alpha \tag{4.40}$$

$$\nabla_\delta f_q = K_{\mathrm{pr}} C_{\mathrm{m}\delta} M^2 \tag{4.41}$$

$$\nabla_\alpha h_\eta = K_\eta M^2 \big(C_{\mathrm{n}\alpha} + \alpha \nabla_\alpha C_{\mathrm{n}\alpha} \big) \tag{4.42}$$

$$\nabla_q h_\eta = 0 \tag{4.43}$$

$$\nabla_\delta h_\eta = K_\eta C_{\mathrm{m}\delta} M^2. \tag{4.44}$$

The partial derivatives (computed using Eqs. 4.36-4.44) of the LTI models (see Eqs. 4.26, 4.27) are not only dependent on M but also on α, δ; parameters that not belong to the scheduling vector ϱ. However, given that these derivatives are computed at desired operating-equilibrium points and the corresponding equilibrium values $\alpha_{\mathrm{r}}, \delta_{\mathrm{r}}$ can be parameterized as a function of the scheduling vector ϱ_{r} (according to the analysis of Sections 4.1.2.2, 4.1.2.3), it can be clearly seen that these LTI models *are fully parameterized by the scheduling vector only.*

Regrouping the above results, all linear time invariant, scheduling vector dependent (with $\varrho \in \mathbf{\Gamma}_{\mathrm{fe,lin}}^{[\eta,M]}$) models of the R'm'B can be written in the following transfer function and state space forms (see Eqs. 4.45, 4.46):

$$\mathcal{S}_{\mathrm{LPV}}(\varrho_{\mathrm{r}}) \overset{\mathrm{tf}}{:} \quad \left\{ \begin{bmatrix} \eta_\delta(s) \\ q_\delta(s) \end{bmatrix} = \begin{bmatrix} G_\eta(s) \\ G_q(s) \end{bmatrix} \delta_\delta = G(s)\delta_\delta. \right. \tag{4.45}$$

[10]The aerodynamic functions $C_{\mathrm{n}}, C_{\mathrm{n}\alpha}, C_{\mathrm{m}\alpha}$ dependency on α, M, δ is omitted for notational simplicity.

$$\mathcal{S}_{\text{LPV}}(\varrho_{\text{r}}) \overset{\text{ss}}{:} \begin{cases} \begin{pmatrix} \dot{\alpha}_\delta \\ \dot{q}_\delta \end{pmatrix} = \begin{pmatrix} \nabla_{\alpha,\text{r}} f_\alpha & 1 \\ \nabla_{\alpha,\text{r}} f_q & 0 \end{pmatrix} \begin{pmatrix} \alpha_\delta \\ q_\delta \end{pmatrix} + \begin{pmatrix} \nabla_{\delta,\text{r}} f_\alpha \\ \nabla_{\delta,\text{r}} f_q \end{pmatrix} \delta_\delta \\ \begin{pmatrix} \eta_\delta \\ q_\delta \end{pmatrix} = \begin{pmatrix} \nabla_{\alpha,\text{r}} h_\eta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_\delta \\ q_\delta \end{pmatrix} + \begin{pmatrix} \nabla_{\delta,\text{r}} h_\eta \\ 0 \end{pmatrix} \delta_\delta \end{cases} \tag{4.46}$$

Missile transfer functions

The matrix transfer function $G(s) = C(sI - A)^{-1}B + D$ presents totally two poles, two zeros for the η-channel and one zero for the q-channel I/O transfer functions as seen from the following relation:

$$G = \begin{bmatrix} G_\eta \\ G_q \end{bmatrix} = \frac{\begin{bmatrix} D_{11}s^2 + (C_{11}B_{11} - A_{11}D_{11})s + C_{11}B_{21} - A_{21}D_{11} \\ -B_{21}s + A_{21}B_{11} - B_{21}A_{11} \end{bmatrix}}{s^2 - A_{11}s - A_{21}}. \tag{4.47}$$

The elements of the state space matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$, depending on the scheduling vector $\varrho = [\eta \ M]^T$, make the values of the zeros and poles of the aforementioned I/O transfer functions varying over the flight envelope. This necessitate a comprehensive stability and dynamics analysis of the linear systems $\mathcal{S}_{\text{LPV}}(\varrho_{\text{r}})$ for every value of the scheduling vector that will shed some light on the stability of the initial nonlinear plant. These six scheduling vector-dependent elements $A_{11}, A_{21}, B_{11}, B_{21}, C_{21}, D_{11}$ are visualized in Figs 4.6a-4.6f.

4.1.3.2 Stability Analysis

The local stability properties of the missile nonlinear dynamics (see Eqs. 4.1-4.2) can be investigated using the well-known Lyapunov's indirect method. For a given reference-equilibrium state x_{r} parameterized in terms of the scheduling vector $\varrho_{\text{r}} = [\eta_{\text{r}} \ M_{\text{r}}]^T$, the eigenvalues of $\mathbf{A}(\varrho_{\text{r}})$ provide the information if the missile is locally stable around this equilibrium point. The eigenvalues and the corresponding stability condition are:

Eigenvalues

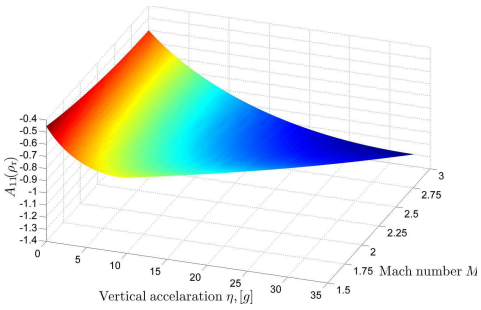
$$\lambda_{1,2}(\varrho_{\text{r}}) = \frac{\nabla_{\alpha,\text{r}} f_\alpha \pm \sqrt{\left(\nabla_{\alpha,\text{r}} f_\alpha\right)^2 + 4\nabla_{\alpha,\text{r}} f_q}}{2} \tag{4.48}$$

Stability condition: The linear missile dynamics are stable iff for $\varrho_{\text{r}} \in \mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]}$,

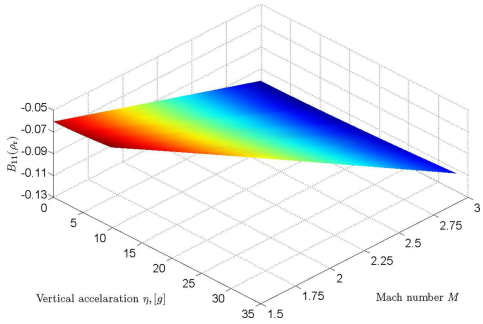
$$\begin{aligned} \nabla_{\alpha,\text{r}} f_\alpha &< 0 \\ \nabla_{\alpha,\text{r}} f_q &< 0. \end{aligned} \tag{4.49}$$

From Fig. 4.6a it can be seen that the first stability condition is always satisfied for all the flight envelope; however the second one not always (see Fig. 4.6c). Using Eq. 4.38 it can be rewritten as a condition over $C_{\text{m}\alpha}$:

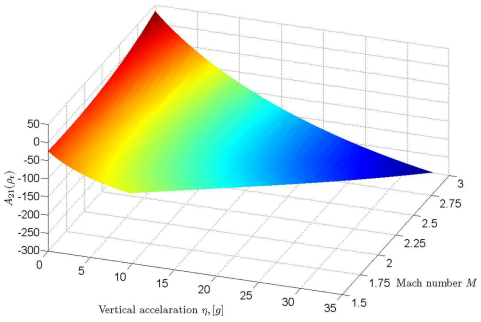
$$C_{\text{m}\alpha,\text{r}} < -\alpha_{\text{r}} \nabla_{\alpha,\text{r}} C_{\text{m}\alpha}. \tag{4.50}$$



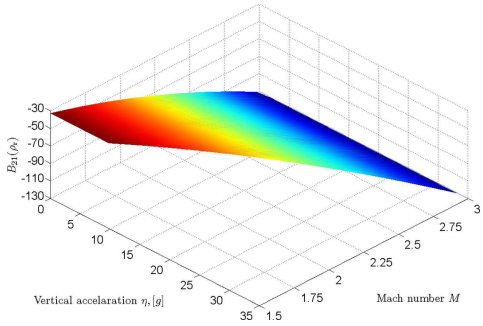
(a) *Element* A_{11}



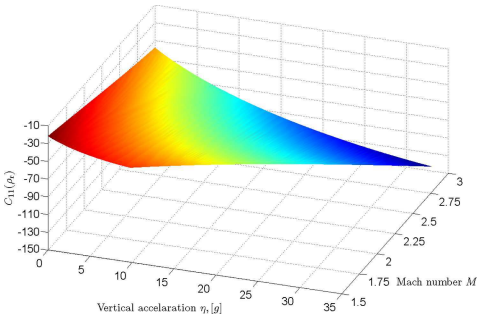
(b) *Element* B_{11}



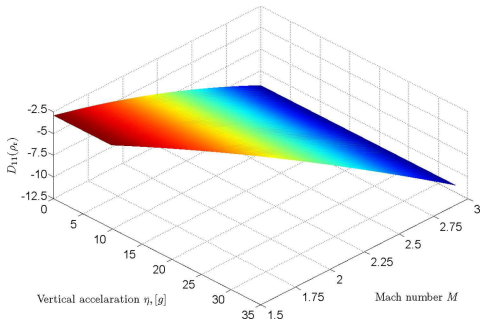
(c) *Element* A_{21}



(d) *Element* B_{21}

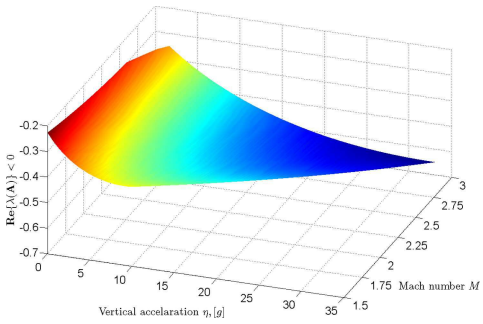


(e) *Element* C_{11}

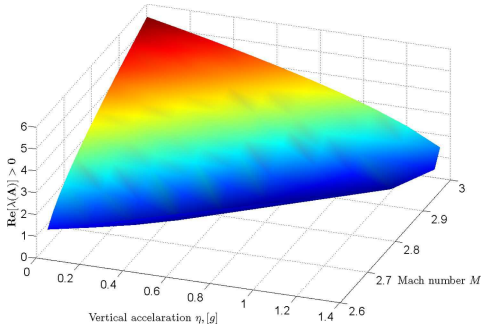


(f) *Element* D_{11}

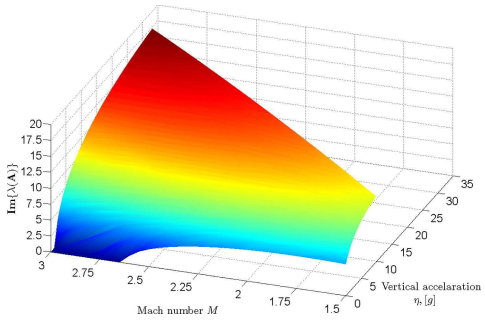
Figure 4.6: Missile LTI system matrix elements



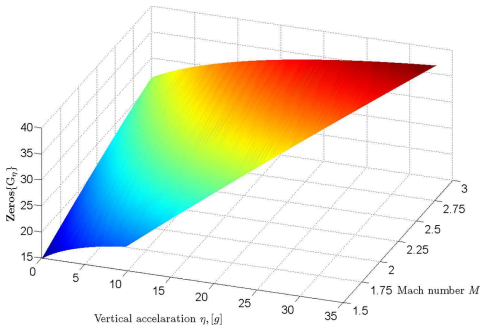
(a) *Stable eigenvalues' real part*



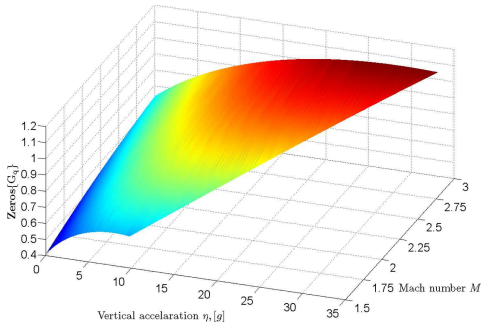
(b) *Unstable eigenvalues' real part*



(c) *Eigenvalues' absolute imaginary value*



(d) G_η *transmission zeros (absolute value)*



(e) G_q *transmission zeros*

Figure 4.7: Missile linearization results - *eigenvalues, transmission zeros*

Stability discussion

The right hand side of Eq. 4.50 is always positive since by observing Fig. 4.2a, the slope of the aerodynamic function is always negative for $\alpha > 0$ (symmetry exists always for $\alpha < 0$). Thus it can be said that roughly, the airframe is stable iff $C_{\mathrm{m}\alpha} < 0$ but this is not totally correct. This previous type of stability analysis based on the sign of $C_{\mathrm{m}\alpha}$ is rather classical (see [77]) and is based on the fact that if for a given equilibrium angle of attack α_{r} and a corresponding trim input δ_{r}, the variation on the pitching moment due to the aerodynamic forces with respect to the center of gravity, caused by an external perturbation and forcing the plant to a new $\alpha = \alpha_{\mathrm{r}} + \Delta\alpha$, tends to bring the angle of attack to its initial equilibrium value, then the airframe is stable.

The full stability conditions (see Eq. 4.49) are given as a function of α and it is difficult to translate them directly on η in order to symbolically calculate the boundaries of the unstable region. The symbolical calculations can be avoided and stability could be studied by iteratively computing the sign of the eigenvalues of the missile linearized dynamics for a fixed gridding of the flight envelope. Thus a good approximation of the unstable subregion $\mathbf{\Gamma}_{\mathrm{fe,un}}$ (with $\mathbf{\Gamma}_{\mathrm{fe,lin}}^{[\eta,M]} \subset \mathbf{\Gamma}_{\mathrm{fe,lin}}$) can be found (see Fig. 4.8). The surface percentage of $\mathbf{\Gamma}_{\mathrm{fe,un}}$ with respect to $\mathbf{\Gamma}_{\mathrm{fe,lin}}^{[\eta,M]}$ and $\mathbf{\Gamma}_{\mathrm{fe}}^{[\eta,M]}$ is 0.82% and 0.79% respectively.

LTI models properties

The linear analysis of the missile's nonlinear dynamics can also provide some very interesting insight results visualized in the following pages. In Figs. 4.7a, 4.7b, the amplitude of the real part of the LTI plants' eigenvalues for both the stable and unstable parts of $\mathbf{\Gamma}_{\mathrm{fe,lin}}^{[\eta,M]}$ is visualized whereas in Fig. 4.7c the imaginary part is displayed. The evolution of the transmission zeros of $G(s)$ (see Eq. 4.47) is also shown in Figs 4.7d, 4.7e.

Finally, in Figs. 4.9a-4.9h the Bode diagrams and the I/O pole-zero maps of $G = [G_\eta \;\; G_q]^T$ are visualized for four different values of the Mach. Two things may be observed: first, the poles of the system are stable but badly damped (except for some unstable cases for $M = 3$, corresponding to $\mathbf{\Gamma}_{\mathrm{fe,un}}$) and second, the plant has non-minimal phase transmission zeros for G_η whereas the zeros of G_q remain stable. In general it can be remarked that all these characteristics of the LTI plants are considerably varying over the flight envelope.

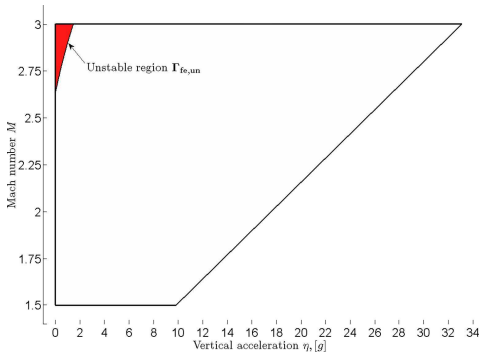
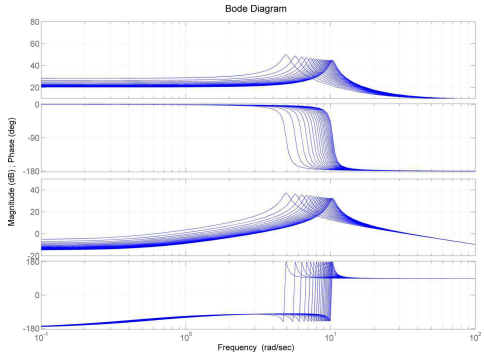
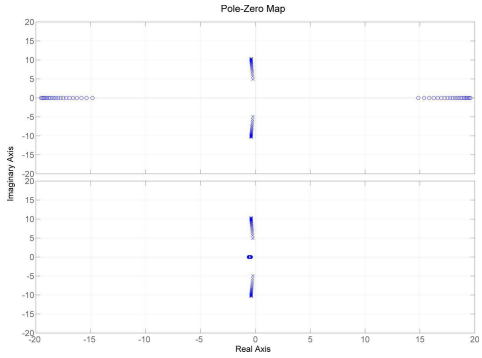


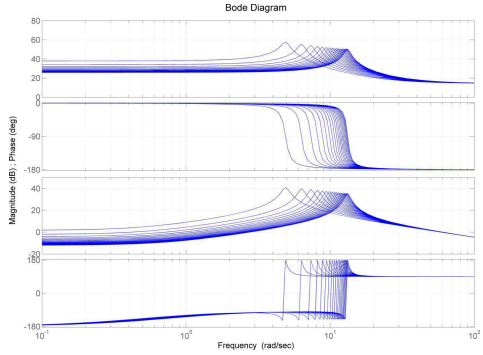
Figure 4.8: Missile flight envelope unstable part



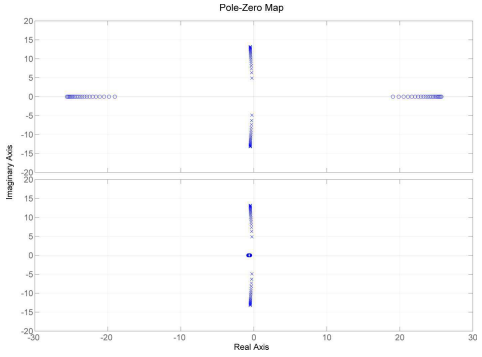
(a) $M = 1.5$



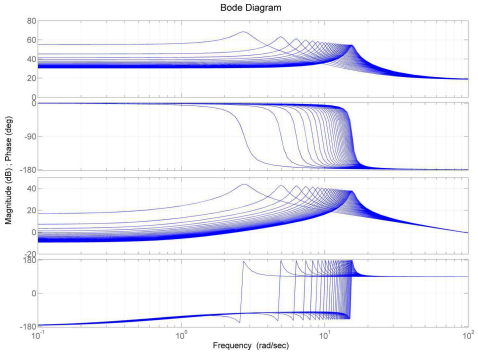
(b) $M = 1.5$



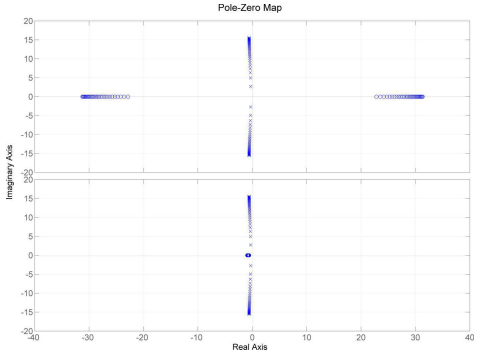
(c) $M = 2$



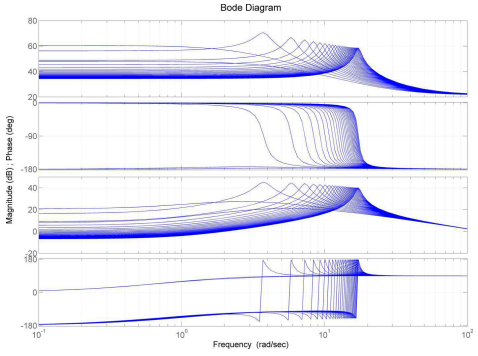
(d) $M = 2$



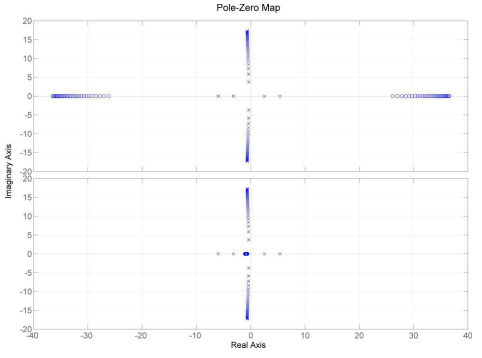
(e) $M = 2.5$



(f) $M = 2.5$



(g) $M = 3$



(h) $M = 3$

Figure 4.9: Missile Bode and Pole-zero maps of G_η, G_q

4.2 The ARV Benchmark Model

The second system considered in this work is an atmospheric re-entry vehicle example (ARV) provided by EADS ASTRIUM Space Transportation corporation. It is used to validate the techniques developed during this thesis and the results given are by no means representing real situations; however they are accurate enough to provide insight into the control methods presented in the next chapters.

4.2.1 Airframe Modeling

System dynamics

The nonlinear model of the vehicle[11] presented here describes its longitudinal motion during the atmospheric re-entry phase (a pitch view is shown in Fig. 4.10). The state x here is once again the angle of attack α (in rad) and the pitch rate q (in $\text{rad} \cdot \text{s}^{-1}$). Two control signals $\delta_{\text{el}}, \delta_{\text{er}}$ (in rad) representing the left and right tail elevator deflections are available to manipulate the vehicle's pitch and roll motion. The deflections are symmetric for pitch control (defining the pitch control signal δ_{e}) and antisymmetric for roll control; here only the first will be considered and is defined as[12]:

$$\delta_{\text{e}} = \frac{1}{2}(\delta_{\text{el}} + \delta_{\text{er}}). \tag{4.51}$$

The pitch rate dynamics of the vehicle are dependent on the Mach number M following a predefined time trajectory (Fig. 4.12a), on the dynamic pressure Q (in N/m^2) depending on the Mach (Fig. 4.12b) and on the physical parameters of the vehicle (Table 4.2). The state dynamics are:

$$\frac{d\alpha}{dt} = q \tag{4.52}$$

$$\frac{dq}{dt} = \frac{SlQ}{I_{\text{yy}}} C_{\text{m}}(\alpha, M, \delta_{\text{e}}) \tag{4.53}$$

where the pitching moment aerodynamic function C_{m} is defined as:

$$C_{\text{m}}(\alpha, M, \delta_{\text{e}}) = C_{\text{m0}}(\alpha, M) + C_{\text{me}}(\alpha, M)\delta_{\text{e}}. \tag{4.54}$$

Flight envelope

The highly nonlinear aerodynamic function derivatives $C_{\text{m0}}, C_{\text{me}}$ are not available in symbolic form as in the missile but are rather tabulated for various points of the vehicle flight envelope (Figs. 4.11a, 4.11b). The latter is parameterized in terms of the angle of attack and the Mach number, thus the scheduling vector taken here is $\varrho = [\alpha \; M]^T$. The flight envelope $\mathbf{\Gamma}_{\text{fe}}^{[\alpha,M]}$ is defined as:

$$\mathbf{\Gamma}_{\text{fe}}^{[\alpha,M]} : \left[30^\circ \leq \alpha \leq 50^\circ, 4 \leq M \leq 26\right]. \tag{4.55}$$

[11] Real values for several parameters are not given for confidentiality reasons.
[12] Once more time dependence is omitted to simplify the equations.

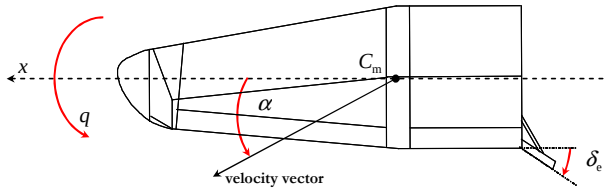
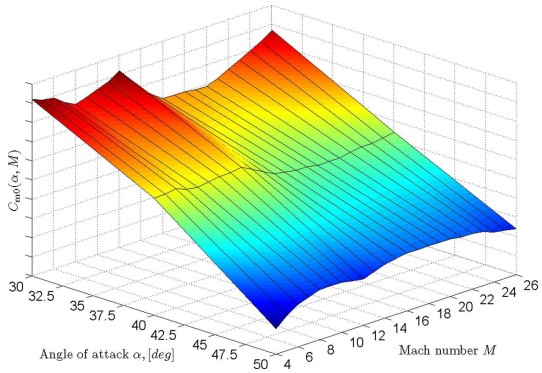


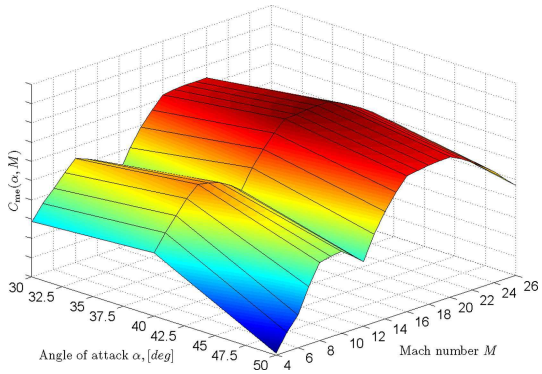
Figure 4.10: The ARV vehicle

Table 4.2: Vehicle & actuator coefficients

Name	Symbol	Unit
Damping ratio	ξ	-
Natural frequency	ω_a	rad/s
Reference area	S	m^2
Reference length	l	m
Moment of Inertia	I_{yy}	kgm^2

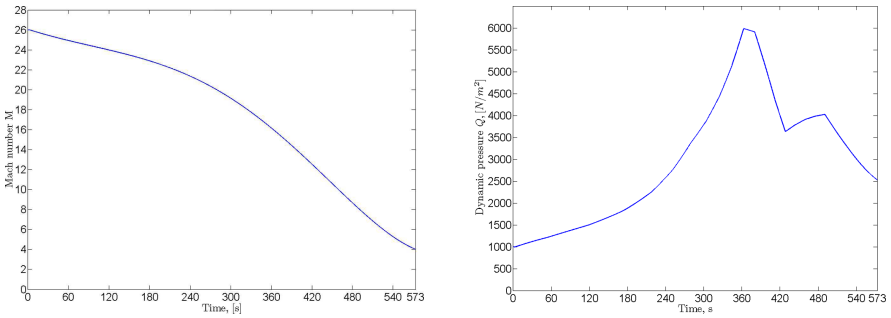


(a) C_{m0}



(b) C_{me}

Figure 4.11: Vehicle aerodynamic functions



(a) *Mach number M* **(b)** *Dynamic pressure Q*

Figure 4.12: Time profiles for *M*, *Q*

The elevator fins are driven by an actuator that can be modeled as a second order filter governed by the following I/O representation:

$$\frac{d^2\delta_e}{dt} + 2\xi\omega_a\frac{d\delta_e}{dt} + \omega_a^2\delta_e = \omega_a^2\delta_c. \tag{4.56}$$

4.2.2 Trim Analysis

`Trim control`

The first step of the *LBGS* procedure (trim control computation) is detailed in this section. The trim control $\delta_e(\varrho_r) = \delta_{e,r}$ maintains the vehicle at a desired angle of attack in the absence of external perturbations. Of course since the Mach number varies according to the profile of Fig. 4.12a this control is not sufficient to stabilize the vehicle and a feedback control should be added. The trim control can be calculated as a function of the scheduling vector ϱ by supposing that at an equilibrium or reference state is imposed and consequently $\frac{dq}{dt}|_r = 0$. To compute $\delta_{e,r}$, Eq. 4.53-4.54 are used and the trim surface is obtained (Fig 4.13):

$$\delta_e(\varrho_r) = -\frac{C_{m0}(\varrho_r)}{C_{me}(\varrho_r)}. \tag{4.57}$$

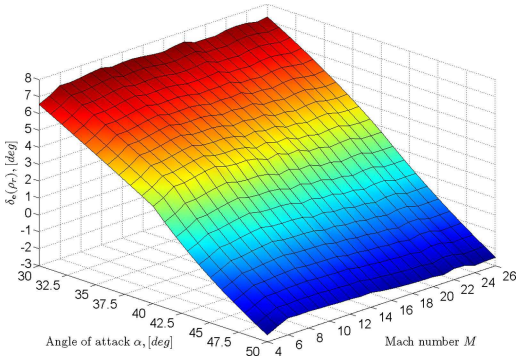


Figure 4.13: Vehicle trim control surface

4.2.3 System Linearization

4.2.3.1 LTI Models

LTI models

Having parameterized the system in terms of a scheduling vector ϱ, the second step of the (LBGS) procedure is to obtain LTI models of the vehicle for every desired operating point inside the flight envelope $\mathbf{\Gamma}_{\text{fe}}^{[\alpha,M]}$. Similarly to the procedure used for the missile, a family of linear models $\mathcal{S}_{\text{LPV}}(\varrho_{\text{r}})$ for every ϱ_{r} can be written in the following state space form:

$$\mathcal{S}_{\text{LPV}}(\varrho_{\text{r}}) \overset{\text{ss}}{:} \quad \dot{x}_\delta = \mathbf{A}(\varrho_{\text{r}})x_\delta + \mathbf{B}(\varrho_{\text{r}})\delta_{e,\delta} \tag{4.58}$$

with $x = [\alpha\ q]^T$ and:

$$x_\delta = x - x(\varrho_{\text{r}}) \tag{4.59}$$

$$\delta_{e,\delta} = \delta_{\text{e}} - \delta(\varrho_{\text{r}}). \tag{4.60}$$

The linearized matrices $\mathbf{A}, \mathbf{B}$ are given by:[13]

$$\mathbf{A}(\varrho_{\text{r}}) = \begin{pmatrix} 0 & 1 \\ \nabla_{\alpha,\text{r}} f_q & 0 \end{pmatrix} \tag{4.61}$$

$$\mathbf{B}(\varrho_{\text{r}}) = \begin{pmatrix} 0 \\ \nabla_{\delta_{\text{e}},\text{r}} f_q \end{pmatrix} \tag{4.62}$$

Jacobians

with:[14]

$$\nabla_{\alpha,\text{r}} f_q = \frac{SlQ_{\text{r}}}{I_{\text{yy}}} \left[\frac{\partial C_{\text{m0}}(\varrho_{\text{r}})}{\partial \alpha} - \frac{\partial C_{\text{me}}(\varrho_{\text{r}})}{\partial \alpha} \frac{C_{\text{m0}}(\varrho_{\text{r}})}{C_{\text{me}}(\varrho_{\text{r}})} \right] \tag{4.63}$$

$$\nabla_{\delta_{\text{e}},\text{r}} f_q = \frac{SlQ_{\text{r}}}{I_{\text{yy}}} C_{\text{me}}(\varrho_{\text{r}})\,. \tag{4.64}$$

The family of LTI systems $\mathcal{S}(\varrho_{\text{r}})$ is written in transfer function form:

$$\mathcal{S}(\varrho_{\text{r}}) \overset{\text{tf}}{:} \quad \left\{ \begin{bmatrix} \alpha_\delta(s) \\ q_\delta(s) \end{bmatrix} = \begin{bmatrix} G_\alpha(s) \\ G_q(s) \end{bmatrix} \delta_{\text{e},\delta} = G(s)\delta_{\text{e},\delta} \right. \tag{4.65}$$

where:

$$G(s) = \begin{bmatrix} G_\alpha(s) \\ G_q(s) \end{bmatrix} = \frac{1}{s^2 + \omega_0^2} \begin{bmatrix} b \\ bs \end{bmatrix}. \tag{4.66}$$

The corresponding natural frequency ω_0 and open loop gain b of the linear systems are calculated from the matrix elements A_{21}, B_{21} and vary as a function of the scheduling vector:

$$\omega_0^2(\varrho_{\text{r}}) = -\nabla_{\alpha,\text{r}} f_q \tag{4.67}$$

$$b(\varrho_{\text{r}}) = \nabla_{\delta_{\text{e}},\text{r}} f_q. \tag{4.68}$$

[13] The function $f_q : \mathbb{R}^3 \to \mathbb{R}$ is the right hand side of Eq. 4.53.

[14] The dynamic pressure Q is not considered as a scheduling parameter since it depends directly on the Mach; however the corresponding reference value Q_{r} is shown in the linearization equations.

4.2.3.2 Stability Analysis

A stability analysis of the vehicle dynamics is given here, based on the family of LTI models $\mathcal{S}_{\text{LPV}}(\varrho_{\text{r}})$ calculated for every value of the scheduling vector ϱ inside the vehicle flight envelope. It may be observed (e.g. from Eq. 4.65) that the linear models present two complex conjugate eigenvalues with zero real parts; thus the vehicle is *conditionally stable*.

`Stability discussion`

The three element surfaces $A_{12}(\varrho_{\text{r}})$, $B_{12}(\varrho_{\text{r}})$ and $\omega_0^2(\varrho_{\text{r}})$ are visualized in Figs. 4.15a-4.15c. The first two figures presenting the evolution of the LTI matrix elements do not give more information further than underlining the heavy change of the system dynamics for all values of ϱ. However, Fig. 4.15c showing the form of the LTI models natural frequency dependence on ϱ, is particularly interesting. This is because a closed loop controller (namely a gain-scheduled one) should be able to maintain appropriate damping to the imaginary closed loop poles and also sufficient stability margins despite this dependence.

This 'bell' type surface is a very good way to characterize the variation of the system's dynamics and will also give rise to the discussion of Chapter 6 concerning gain scheduling control laws and their ability to *capture the plant's nonlinearities and change of dynamics*; it will indeed be shown that the gain-scheduled control laws calculated in Chapter 6 achieve this task by means of the gap metric.

This change of the natural frequency ω_0 can be also visualized in the following figure (Fig. 4.14) representing Bode magnitude diagrams of transfer functions G_α of the vehicle's family linear systems $\mathcal{S}_{\text{LPV}}(\varrho_{\text{r}})$ for a significant number of frozen values for ϱ.

As a last comment it can be said that whereas in the missile the pitch rate q is used also as a measured output; here it is not the case and only the angle of attack α is used. This is done primarily for reasons of simplicity of the feedback loop as it will be seen in the following chapters since a gain-scheduled controller of the least possible complexity is always sought.

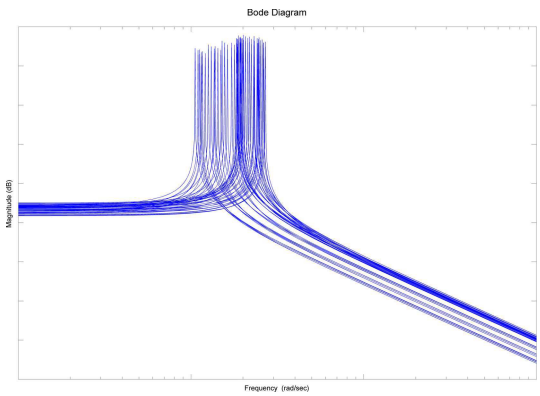
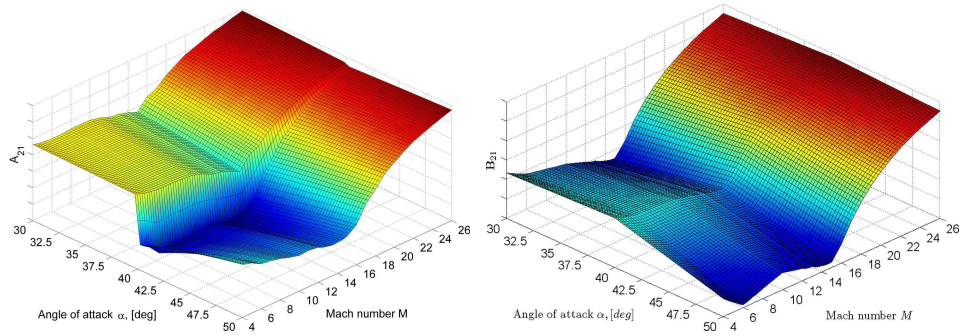
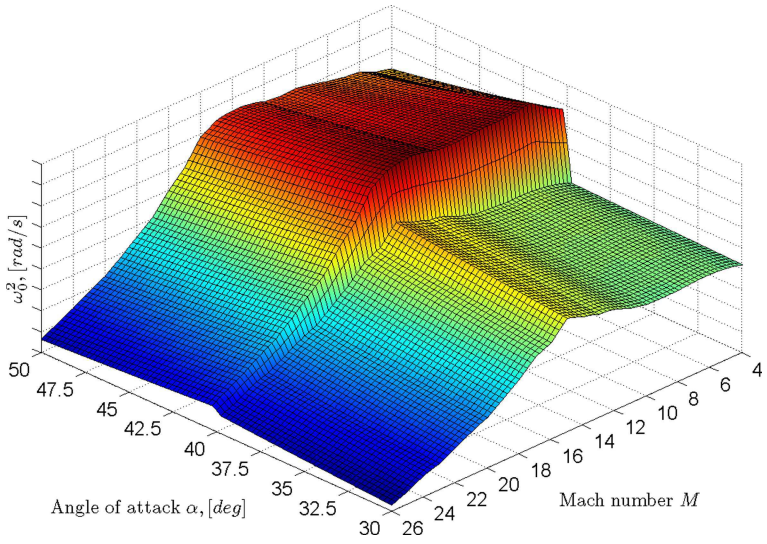


Figure 4.14: Vehicle Bode magnitude diagrams



(a) *Element* A_{21} **(b)** *Element* B_{21}



(c) *Natural frequencies*

Figure 4.15: Vehicle linearization results

4.3 Conclusions

In this chapter we have presented the preliminary work conducted concerning the modeling and analysis of the two benchmark systems used during the thesis in order to validate the proposed gain scheduling strategies of the following chapters. This phase practically corresponds to the first two steps of the LBGS procedure detailed in Section 1.3.1, namely the trim analysis (or equilibrium point parametrization) and the Jacobian linearization of the plants.

The procedure followed is similar in both cases: first choose a family of system variables (scheduling vector) to parameterize the equilibrium points of the initial nonlinear system and then use either symbolical or numerical techniques to calculate a trim control in order to equilibrate the state/output of the plant to a pre-defined desired value for all the operating domain of the system. Second, calculate LTI models of the system for a family of reference values of the scheduling vector and analyze their stability.

It has been analyzed that for the missile this parametrization is *output*-based whereas for the missile is *state*-based. The missile presents a small unstable region of its flight envelope whereas the vehicle is everywhere between the limits of stability and instability.

Chapter 5

Ad-hoc Control Strategies

Overview

In this chapter two gain scheduling control strategies are applied to the autopilot problem of an Air-to-air missile. These strategies are the *controller blending* and *state feedback/observer*-based interpolation methods detailed in Chapter 1 of this thesis and they are used for the control of the pitch-axis nonlinear model of the Reichert Air-to-air missile. These methods present some advantages in terms of using powerful tools of the modern robust control theory to control a nonlinear system; even so, in terms of implementation they are not so realistic compared to the *gain blending* method detailed in the next chapter. For low dimension and/or not much nonlinear systems however they still remain attractive and rather intuitive. The scope of this chapter is mostly to insist on their characteristics, rather than perform an exhaustive simulation procedure, remaining though inside a practical context.

Chapter contents

5.1 Introduction

Method overview

In this chapter two major interpolation methods will be applied to the pitch axis autopilot problem of an Air-to-air missile. These methods are the *controller blending* and *observer/state feedback* interpolation methods detailed in Sections 1.3.2.1 and 1.3.2.6 respectively. These methods have been selected over others (e.g. *controller switching*, *state space matrix interpolation*) due to the advantages they present. The *controller blending* method for example demands less computational effort and presents a rather small interpolation difficulty due to the fact that only output signals are interpolated whereas the *observer/state feedback* method is rather straightforward since it gives an estimate of the plant state while being a true MIMO method. In addition, it rises as a good extension to the *controller blending* method since by Youla parameterization (see Section 3.2) the same LTI controllers may be used for both methods and thus a comparison between the methods is easier to perform.

These methods are compared to each other and to the more systematic *gain blending* method (detailed in the next chapter) using a realistic scenario and extensive simulations. The advantages and disadvantages of each method are stressed out and potential improvements are proposed. This chapter comes as a natural extension of Chapter 4 where the first two steps of the linearization-based gain scheduling procedure (see Section 1.3.1) were applied to obtain the *trim control surface/operating point parametrization* and the corresponding *LTI/LPV models* for each value of the scheduling vector, both for the Reichert missile and the re-entry vehicle. In this chapter, the three remaining steps are detailed, namely: the *LTI controller computation*, *controller interpolation* and *controller implementation & validation*.

The chapter is organized in two main parts: the first one (see Section 5.3) presents the *controller blending* method whereas the second (see Section 5.4) presents the *observer/state feedback method*. Finally, the chapter ends with some conclusions.

5.2 Related Work

The origins

A full bibliographic study on the subject of missile control is beyond the scope of this work for two reasons: first, several methods have been proposed to cope with different problems and second, the missile autopilot design dates back to the 40's and a huge number (maybe hundreds) of references on the subject can be found; in this monograph only the most notable works since the 90's are cited.

Perhaps one of the first works published[1] is the one found in [16] exactly half a century ago. It concerns a simple angular rate/position feedback in order to stabilize the roll motion of an Air-to-air missile. Of course there was no question using even the simplest adaptive control scheme with the technology of the time.

[1] One can find even earlier research work on guided missiles; see for example [1].

An adaptive control approach using parameter estimation and gain scheduling was used in [76] to obtain an autopilot for a flexible Air-to-air missile. An ad-hoc control scheme is used and its gains are scheduled by identifying the fin position pitching coefficient for the entire flight envelope of the missile.

One of the first examples in gain scheduling applied on missile control is found in [112] in the early 90's. In this work modern $\mathscr{H}_\infty$-μ techniques are used to obtain LTI controllers which are put in observer/state feedback form. The solutions to the corresponding Riccati equations are then scheduled as a function of the angle of attack (AoA) and the Mach number in order to provide a time-varying gain-scheduled controller.

In [17], the pitch axis missile nonlinear model is linearized and put in LFT form with the uncertainty block Δ using the AoA. Linear controllers are designed (being robust with respect to the AoA) and scheduled for different values of the Mach number. The results are promising, even though the simulations are not too exhaustive. Another similar approach using tools of the μ-analysis to guarantee stability between the controller synthesis points is found in [37]. The controllers are optimal LQ regulators but the results were not too good.

An early attempt to use a q-LPV modeling of a missile's dynamics starting from a pure nonlinear model can be found in [123]. Then a particular type of trajectory scheduling is used in order to avoid the classical procedure of designing a finite number of LTI controllers for various design points. The controllers used are once again optimal LQ regulators, however the simulation results mostly demonstrate the feasibility of the approach.

The 'Reichert' missile

A major advance in missile autopilot design was done by the classical work found in [103]. This paper presents a benchmark pitch axis model of an air-air missile (Reichert missile Benchmark or *R'm'B*) and uses classic $\mathscr{H}_\infty$ control theory, along with a particular type of controller implementation found in more detail in [79], in order to remove the famous *hidden coupling terms*[2]. The controllers are scheduled using *directly* the output and not the state (as well as the Mach number), and ZPK interpolation (see Section 1.3.2.3) and the overall control scheme is tested using extensive simulations. Similar ideas, along with a very small reference on the missile's flight envelope can be found in [150] where instead of robust $\mathscr{H}_\infty$, reduced order controllers are considered. In the same framework as in the previous two references, an observer/state feedback form of the central $\mathscr{H}_\infty$ controller obtained by appropriate frequency weighting is used in [55] in order to construct a gain-scheduled controller.

A modern approach using modern $\mathscr{H}_\infty$ LPV control and LMI's is the famous paper [13] (see equally [10]). In this paper the controllers are scheduled using two nonlinear functions (depending on the Mach, altitude and AoA respectively) inside a polytope. The overall scheme gives all the stability guarantees of this class of control methods but suffers also of all their inherent drawbacks (conservativeness, complexity).

[2]For more details on this important notion see the discussion in 1.3.3.2.

Another application example using a q-LPV formulation (using state transformations) of a 6DoF bank-to-turn (BTT) missile is the one in [25]. Tools of the μ-synthesis robust control theory are used and the resulting controllers for the pitch and roll/yaw channels are scheduled using the AoA and the roll rate (similar approaches may be found either in [124] or in [123]).

In [116] a pure nonlinear control strategy (dynamic inversion) is compared with $\mathscr{H}_\infty$ gain scheduling using the so-called $\mathcal{D}$ implementation for the autopilot of a bank-to-turn missile. The two methods were found to be equivalent in terms of performance and robustness thus clearly favoring the gain scheduling approach due to its simplicity.

The *controller blending* interpolation method, along with the ideas concerning controller realization treating the hidden coupling terms[3], is tested using the Reichert missile benchmark in [81]. The methodology, even though it appears rather promising, is not validated by simulations in an appropriate manner.

A very good reference on the subject is the work found in [45]. A full autopilot is designed for an Aerospatiale missile with very lightly damped bending modes. The classic Glover&McFarlane $\mathscr{H}_\infty$ loop shaping design procedure (LSDP) is used and robustness is verified using the ν-tool and a complete simulation suite.

In [24] an autopilot for a 6DoF skid-to-turn missile is designed using μ-analysis. The controllers are scheduled using the dynamic pressure and a signal conditioning/blending technique similar to *controller blending*. The effectiveness of the control loop is also validated by an interesting engagement scenario with the missile pursuing an aircraft sustaining 9g normal accelerations.

A self-scheduled nonlinear pitch-axis autopilot for a missile is designed in [63] using LPV synthesis tools coupled with $\mathscr{H}_\infty$ loop shaping design criteria. The controller is scheduled using the Mach number, altitude and AoA over a wide flight envelope. The approach is systematic and stability preserving, offering good results in terms of performance and robustness.

An interesting theoretical work may be found in [47] where notions like incremental stability are used for the analysis of a PI controlled missile (namely the Reichert benchmark model). The analysis, even though it turns out to require the solution of LMI's is not put in the traditional autopilot performance/robustness framework and has not been used in other cases to date.

The ideas behind stability preserving interpolation using the stability covering condition analysis of 1.3.3.2 were applied as a benchmark to the autopilot synthesis of the Reichert benchmark model in [132] and [129]. The first paper concerns observer/state feedback interpolation whereas the second interpolation using the Youla parameter. Both papers give a theoretical flare to the application but unfortunately lack significantly on the controller validation part.

Another approach using a q-LPV formulation of a missile dynamics and modern LPV control (casts the problem as a generalized disturbance rejection one) gives some good results and is detailed in [136].

[3] See once again the classic paper [79] or Section 1.3.3.2 for further details.

A comparative paper demonstrating various control strategies for a skid-to-turn Aerospatiale missile is found in [33]. The control strategies include classic PI-like loops as well as nonlinear linearizing static and dynamic state feedbacks. The results clearly favor the nonlinear control methods; however it remains to be seen if they are on the one hand realistic for implementation and on the other hand, if being compared to more advanced gain-scheduled robust control schemes, retain their advantages.

A rather unusual missile control problem is outlined in [101]. The problem is the trajectory following of a particular 6DOF skid-to-turn missile. The controlled outputs are the AoA, sideslip and Euler roll angle and the control setup consists of two parts: first an open loop dynamic inverse of the plant puts it on the desired trajectory and second an eigenstructure assignment LTV (due to trajectory linearization) control law stabilizes the open loop dynamics. The method is promising and according to the authors presents advantages over the traditional gain scheduling design but robustness testing is yet required.

An alternative strategy for missile autopilot design based on velocity-based gain-scheduling is found in [90] and [89]. This class of methods has been analyzed in Chapter 2 and their inventors claim that they present superior features over the traditional gain scheduling ones, even though their merits are doubted by some (see for example [84]). In any case, the simulation results appearing in both seem nice and demonstrate the feasibility of the approach.

Some work on digitally implemented autopilot control laws in the $\mathscr{H}_\infty$ loop shaping context can be found in [43] or equally in [42]. In the first case, reduced order discrete time dynamic controllers are considered and interpolated using the output (vertical acceleration) and the Mach number, whereas in the second case, multi-rate design is considered.

Another method based on multi-model eigenstructure assignment and μ-iteration is presented in [36]. This work is notable since on the one hand, it presents very good simulation results on the well known Reichert benchmark missile and on the other hand, because several other classical works on gain scheduling control autopilots are compared (see for example Section 4.5 of the paper) to the proposed approach.

In [3], integral quadratic constraints (IQC's) and LPV modeling are used to analyze the stability robustness of an uncertain nonlinear missile control system based on dynamic inversion. The tuning of the controller was done using a genetic algorithm and then it was put (together with the plant) in LPV/LFT form in order to use the IQC analysis tools. Since the latter can be put into LMI form they are computationally tractable. Finally, the stability of of the autopilot was proven under the appearance of time-varying parameters.

Some very recent work may be found in [141] where a fuzzy interpolation control is used for the missile autopilot. In [99] an LPV control scheme is compared to an eigenvalue assignment technique for the autopilot of a skid-to-turn missile. Finally a paper concerning control of a square cross section missile using classical $\mathscr{H}_\infty$ control and nonlinear dynamic inversion is the one in [30].

5.3 Missile Control Objectives

Missile model

The missile autopilot control goals-objectives will be detailed in this section. Recall from Chapter 4 that the pitch axis missile nonlinear parameter-dependent model $\mathcal{S}_{\text{pd}}$, is a SIMO system with its state being comprised by the angle of attack (AoA) α (in rad) and the pitch rate q (in rad/s); its control input being the tail elevator deflection angle δ (in rad) and the measure vector being comprised by the vertical acceleration η (in g's) and the pitch rate q. The nonlinear model is parameterized also by the Mach number M considered as an internal variable (see Fig. 5.1).

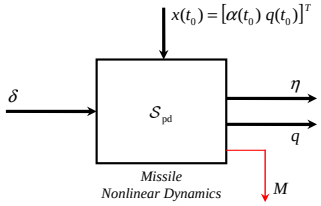


Figure 5.1: Missile block diagram.

Flight envelope

The missile's rectangular flight envelope is formed due to the restrictions on the AoA and Mach number: $-20^\circ \leq \alpha \leq 20^\circ$ and $1.5 \leq M \leq 3$ respectively and linear models may be computed for every α, M inside this envelope. Given though that the AoA *is not* available as a measurement, it *should not* be used directly when calculating trim conditions and linear models.

The vertical acceleration η should be used in its stead, and thus the flight envelope can be re-parametrized as a function of η, M. The obtained flight envelope $\mathbf{\Gamma}_{\text{fe}}^{[\eta,M]}$ has now a more complicated non-convex trapezoidal form (see Fig. 4.5) and thus, for every value of M, the maximum admissible value for η (due to the limits on α) is given by (see also Section 4.1.2.3)[4]:

$$\eta_{\text{fe}}(M) \simeq -0.454M^3 + 5.035M^2, \quad \alpha \leq 0. \tag{5.1}$$

As it has been already detailed, a linear approximation of the aforementioned equation yields a superset of $\mathbf{\Gamma}_{\text{fe}}^{[\eta,M]}$; this new, slightly redundant[5], flight envelope (denoted as $\mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]}$) has been used instead for simplicity, and it forms a trapezium whose four corners are the following pairs:

$$[\eta, M]: \quad [0, 1.5], [9.7969, 1.5], [0, 3], [0, 33.0559]. \tag{5.2}$$

The scheduling vector ϱ (being now formed by η, M) takes values inside $\mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]}$ and parameterizes all the plants equilibrium points, trim control and linear models, as it has been detailed in the previous chapter.

[4] Of course since the AoA may also take positive values, the flight envelope is symmetric with respect to the Mach number and so do the linear models, trim control etc.

[5] Since this approximation adds up about 3.6% of surface.

Control objectives

The *control objective* for the autopilot is simple enough: track step reference signals $\eta_{\mathrm{r}}(t)$ of different amplitudes inside the flight envelope while the Mach varies with certain *performance* and *robustness* constraints. The *performance* constraints concern the output tracking characteristics (P_1) and maximum control signal rates (P_2) whereas the *robustness* constraints concern stability margins/high frequency open loop attenuation (R_1) and robust stability under aerodynamic coefficient perturbations (R_2). These constraints are taken from the benchmark paper [103] and are the following:

P_1: Track step commands on $\eta_{\mathrm{r}}(t)$ of various amplitudes with a time constant[6] $\tau \leq 0.35$s, overshoot $M_{\mathrm{p}} \leq 10\%$ and steady state error $e_{\mathrm{ss}} \leq 1\%$.

P_2: The tail elevator angle deflection rate $|\dot{\delta}|$ should be inferior to 25deg/s for 1g step reference commands.

R_1: The missile should exhibit robust stability inside all its flight envelope when the pitching moment coefficients $(a_{\mathrm{m}}, b_{\mathrm{m}}, c_{\mathrm{m}})$ and d_{m} vary around their nominal values by $\pm 25\%$.

R_2: The linearized system should maintain at least 30dB attenuation for the gain amplitude of the obtained open loop transfer function, when the loop is opened just before the actuator.

Performance

The first two performance characteristics denote objectives that in the benchmark paper [103] are tested for a given simulation profile on the nonlinear system. This profile may sometimes be too favorable for the autopilot since variations of big amplitude are demanded for relatively high values of the Mach number where the controllers are in general more performing. The procedure often used by the designers is to calculate a small number of controllers and try to adapt the performance objectives only for these points, hoping that the design will carry on to the nonlinear system. There is no indication if the number of points considered is too small, too big or even if the points themselves are chosen in places where the nonlinear dynamics need treatment.

Robustness

The situation for the robustness objectives is the same: these objectives are most of the time satisfied (with a relative margin) at the synthesis points and the nonlinear gain-scheduled controller is exhaustively tested for stability when the aerodynamic coefficients are perturbed. This analysis is done using a Monte Carlo procedure for hundreds (in real world systems maybe several thousands) of operating points; thus guaranteeing in a way a posteriori the well-behaved of the controller. Other tests performed include linearization of the total open loop of the system plus the gain-scheduled controller in order to check its stability margins for frozen values of time along the desired reference trajectory of the scheduling vector.

[6] To avoid confusion the time constant is here defined as the time it takes the tracking output to reach the 63.2% of its final value.

5.4 Controller Blending

In this section the results from the first ad-hoc scheduling method, based on *controller blending*, will be presented. Recall from Section 1.3.2.2, that this method uses LTI controllers designed around a family of operating points inside the system's operating domain and then interpolates the *outputs* of adjacent controllers in order to provide the final control signal.

Synthesis points

Initially in this work, only four controllers at the corners of the flight envelope $\mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]}$ were used, but the performance of the gain-scheduled controller was not satisfactory. This was due to the fact that such a small number of controllers was inadequate to capture the variation of the plant dynamics inside the operating domain. Thus, it was decided to divide the flight envelope in four scheduling regions $\Gamma^1, \Gamma^2, \Gamma^3, \Gamma^4$ formed by nine synthesis points. These synthesis points, for each value $\varrho^i, i = 1, \ldots, 9$[7], are shown in Table 5.1.

Table 5.1: Controller synthesis points[(i),(ii)].

Point	1	2	3	4	5	6	7	8	9
η	0	4.898	9.797	0	10.713	21.426	0	16.528	33.056
M	1.5	1.5	1.5	2.25	2.25	2.25	3	3	3

(i) Each point corresponds to a value ϱ^i.
(ii) The values for η are approximated to the third digit.

The flight envelope along with the four scheduling regions are illustrated in Fig. 5.2 (the symmetrical part for negative values of the vertical acceleration is not here shown). Before proceeding to the LTI controller calculation, it should be stressed out that the same nine synthesis points are used for every type of interpolation method in the rest of this work.

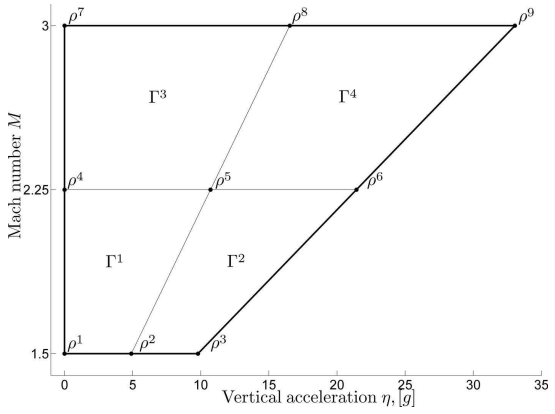


Figure 5.2: Flight envelope, synthesis points and scheduling regions.

[7]Instead of noting each synthesis point as $\varrho^{i,j}$ (where $i = 1, \ldots, 4$ is the scheduling region and $j = 1, \ldots, 4$ the number of the controller in each region) as in Section 1.3.2.2, a simpler notation is used with $i = 1, \ldots, 9$ denoting globally every controller in $\mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]}$.

5.4.1 LTI Controller Synthesis

Missile LPV model

In this section the LTI controller synthesis procedure (corresponding to the third step of the linearization-based gain scheduling procedure) will be detailed. Recall that in Section 4.1.3.1, an LPV model of the missile's nonlinear dynamics, parameterized by the scheduling vector $\varrho = [\eta\ M]^T$, was obtained. This model $\mathcal{S}_{\text{LPV}}(\varrho)$ was of the following form:

$$\mathcal{S}_{\text{LPV}}(\varrho): \quad \begin{aligned} \dot{x}_\delta &= \mathbf{A}(\varrho)x_\delta + \mathbf{B}(\varrho)\delta_\delta \\ y_\delta &= \mathbf{C}(\varrho)x_\delta + \mathbf{D}(\varrho)\delta_\delta \end{aligned} \tag{5.3}$$

with $x = [\alpha\ q]^T, y = [\eta\ q]$ and:

$$\begin{aligned} x_\delta &= x - x(\varrho) \\ \delta_\delta &= \delta - \delta(\varrho) \\ y_\delta &= y - y(\varrho). \end{aligned} \tag{5.4}$$

For the sake of correctness, it should be stressed that the above LPV model is in fact a family of LTI models, smoothly parameterized by fixed-equilibrium values ϱ_{r} of the scheduling vector. Thus, each member of this family $\mathcal{S}_{\text{LTI}}(\varrho_{\text{r}})$ of linear models describes the behavior of the initial nonlinear parameter-dependent missile model $\mathcal{S}_{\text{pd}}$, *locally* around the equilibrium point.

The corresponding constant system matrices $\mathbf{A}(\varrho_{\text{r}}), \mathbf{B}(\varrho_{\text{r}}), \mathbf{C}(\varrho_{\text{r}})$ and $\mathbf{D}(\varrho_{\text{r}})$ are of course obtained by Jacobian linearization using appropriate trim values for the state and the input. The corresponding to each frozen state space description $\mathcal{S}_{\text{LTI}}(\varrho_{\text{r}})$, I/O matrix transfer function $G(s)$ is written as:

$$\begin{bmatrix} \eta_\delta(s) \\ q_\delta(s) \end{bmatrix} = \begin{bmatrix} G_\eta(s) \\ G_q(s) \end{bmatrix} \delta_\delta = G(s)\delta_\delta. \tag{5.5}$$

LTI synthesis

In the context of controller synthesis, the plant was preceded by the actuator transfer function $G_{\text{a}}(s)$ (see Eq. 4.11) and also augmented by an integrator acting on the tracking error $e_\delta = \eta_{\text{r}} - \eta_\delta$ (in order to ensure proper reference trajectory following). A robust $\mathscr{H}_\infty$, S/KS-type mixed sensitivity control strategy was then selected in order to treat the tracking error dynamics on the one hand, but limit on the other hand the control effort rate, conformably to the performance objectives P_1, P_2. Appropriate constant (for simplicity) weights $k_e, k_{\dot{\delta}}$ were added for use with this method on each signal $e_\delta, \dot{\delta}_\delta$.

In addition to the $\mathscr{H}_\infty$ optimization, LMI pole placement constraints were imposed, to have a better control on the closed loop dynamics and avoid inherent inconveniences of standard $\mathscr{H}_\infty$ control (e.g. very fast closed loop eigenvalues). Thus, an LMI region $\mathcal{D}(\lambda_{\min}, r_{\max}, \vartheta_{\min})$ was used with $\lambda_{\min}, r_{\max}, \vartheta_{\min}$ providing minimum decay rate, maximum undamped natural frequency and minimum damping constraints for the closed loop eigenvalues (see Fig. 3.4)[8].

[8] For each of the nine synthesis points, $\lambda_{\min}$ was variable in order to fine-tune the maximum control signal rate whereas $r, \theta, k_e, k_{\dot{\delta}}$ were held constant. The values used were $r_{\max} = 150, \vartheta_{\min} = 0.707$ (actuator undamped natural frequency/damping) and $k_e = 1, k_{\dot{\delta}} = 3$.

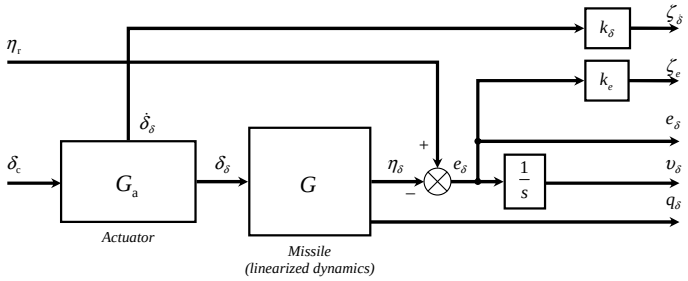


Figure 5.3: $\mathscr{H}_\infty$ synthesis standard form.

The total synthesis block diagram used is illustrated in Fig. 5.3. The external 'disturbance' signal (often denoted by w) is the output reference signal η_r, the 'performance' signals $\zeta_{\dot{\delta}}, \zeta_e$ are the weighted actuator deflection rate $\dot{\delta}_\delta$ and tracking error e_δ. The control input is the commanded actuator signal δ_c and the controller inputs are the tracking error e_δ, the integral of the tracking error v_δ, and the pitch rate q_δ[9].

Standard form

The controller synthesis standard form may be written thus in the following compact state-space form:

$$\begin{bmatrix} \dot{x}_\mathrm{aug} \\ \zeta_\infty \\ y_\mathrm{aug} \end{bmatrix} = \mathbb{P} \begin{bmatrix} x_\mathrm{aug} \\ w \\ u_\mathrm{aug} \end{bmatrix} \tag{5.6}$$

where:

$$\mathbb{P} = \begin{bmatrix} \mathbf{A}_\mathrm{aug} & \mathbf{B}_w & \mathbf{B}_u \\ \mathbf{C}_\zeta & \mathbf{D}_{w\zeta} & \mathbf{D}_{u\zeta} \\ \mathbf{C}_y & \mathbf{D}_{wy} & \mathbf{D}_{uy} \end{bmatrix}. \tag{5.7}$$

The vectors in the above equations are: $x_\mathrm{aug} = \left[\dot{\delta}_\delta, \delta_\delta, \alpha_\delta, q_\delta, v_\delta\right]^T \in \mathbb{R}^{5\times 1}$ is the augmented (without the controller) state vector, $\zeta_\infty = [\zeta_{\dot{\delta}}, \zeta_e]^T \in \mathbb{R}^{2\times 1}$ is the performance vector, $y_\mathrm{aug} = [e_\delta, v_\delta, q_\delta]^T \in \mathbb{R}^{3\times 1}$ is the controller input vector and $u_\mathrm{aug} = \delta_\mathrm{c}$ is the controller output and thus evidently $\mathbb{P} \in \mathbb{R}^{10\times 7}$. The latter may also be written as:

$$\begin{bmatrix} \zeta_\infty \\ y_\mathrm{aug} \end{bmatrix} = \begin{bmatrix} P_{w\zeta}(s) & P_{u\zeta}(s) \\ P_{wy}(s) & P_{uy}(s) \end{bmatrix} \begin{bmatrix} w \\ u_\mathrm{aug} \end{bmatrix}. \tag{5.8}$$

The standard form synthesis matrices of the above Eq. 5.7 are given by the following equations:

$$\mathbf{A}_\mathrm{aug} = \begin{bmatrix} -2\zeta\omega_\mathrm{n} & -\omega_\mathrm{n}^2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & B_{11} & A_{11} & A_{12} & 0 \\ 0 & B_{21} & A_{21} & A_{22} & 0 \\ 0 & -D_{11} & -C_{11} & -C_{12} & 0 \end{bmatrix} \tag{5.9}$$

[9]Note that the 'δ' notation is maintained to emphasize that the signals are in fact perturbation ones around equilibrium points.

$$\mathbf{B}_{\mathrm{w}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{B}_{\mathrm{u}} = \begin{bmatrix} \omega_{\mathrm{n}}^2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.10}$$

$$\mathbf{C}_{\infty} = \begin{bmatrix} k_{\dot{\delta}} & 0 & 0 & 0 & 0 \\ 0 & -k_e D_{11} & -k_e C_{11} & -k_e C_{12} & 0 \end{bmatrix}$$

$$\mathbf{C}_{\mathrm{y}} = \begin{bmatrix} 0 & -D_{11} & -C_{11} & -C_{12} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.11}$$

$$\mathbf{D}_{w\zeta} = \begin{bmatrix} 0 \\ k_e \end{bmatrix} \quad \mathbf{D}_{u\zeta} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{D}_{wy} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{D}_{uy} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.12}$$

The additional constants $A_{11}, A_{12}, A_{21}, A_{22}, B_{11}, B_{21}, C_{11}, C_{12}, D_{11}$ are in fact the (frozen) state-space matrix components of the missile linearized dynamics around an equilibrium point (see Eq. 5.3) with:

$$\begin{aligned} \mathbf{A}(\varrho_{\mathrm{r}}) &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad \mathbf{B}(\varrho_{\mathrm{r}}) = \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix} \\ \mathbf{C}(\varrho_{\mathrm{r}}) &= \begin{bmatrix} C_{11} & C_{12} \\ 0 & 1 \end{bmatrix} \quad \mathbf{D}(\varrho_{\mathrm{r}}) = \begin{bmatrix} D_{11} \\ 0 \end{bmatrix}. \end{aligned} \tag{5.13}$$

Furthermore, due to the special form of the missile nonlinear dynamics, $A_{22} = C_{12} = 0$ and $A_{12} = 1$. It should be also noted that the minus signs inside the matrices are due to the negative addition of the reference signal η_{r} to the vertical acceleration error η_{δ} (see Fig. 5.3).

The goal for the robust controller $K(s)$ now is to ensure closed stability, minimization of the $\mathscr{H}_{\infty}$ norm of the transfer function from the disturbance to the performance vector, and also ensure a correct eigenvalue placement inside the LMI region $\mathcal{D}$. Briefly this can be denoted as:

$\mathscr{H}_{\infty}$ *synthesis*: Calculate a linear MISO, dynamic output feedback controller $K(s) = \left[\begin{smallmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{C}_k & \mathbf{D}_k \end{smallmatrix}\right]$ with $u_{\mathrm{aug}}(s) = K(s) y_{\mathrm{aug}}(s)$, so that $\|T_{w\zeta_{\infty}}\|_{\infty} < \gamma$ with $T_{w\zeta_{\infty}}$ stable, and additionally $\lambda(\mathbf{A}_{\mathrm{cl}}) \in \mathcal{D}(\lambda_{\min}, r_{\max}, \vartheta_{\min})$[10].

Now for each synthesis point, it has been tried to compute a feedback controller in such a way that the step response time constant is minimized while the control effort rate is limited (according to performance objectives P_1, P_2). The parameters used are shown in Table 5.2.

[10]Recall that $T_{w\zeta_{\infty}} = \mathcal{F}_l(\mathbb{P}, K) = \mathbf{C}_{\mathrm{cl}}(s\mathbb{I} - \mathbf{A}_{\mathrm{cl}})^{-1}\mathbf{B}_{\mathrm{cl}} + \mathbf{D}_{\mathrm{cl}}$.

Table 5.2: Controller synthesis parameters[(i)].

Points	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
$\lambda_{\min}$	5.831	5.922	6.025	11.23	11.52	11.90	18.18	18.27	19.09
τ	361.9	339.3	325.6	199.2	191.3	182.2	129.2	128.9	122.0
$t_{\rm s}$	616.4	576.6	557.7	332.1	316.1	300.6	209.6	208.2	196.3
$M_{\rm p}$	0	0	0	0.041	0.054	0.051	0.068	0.076	0.108
GM	27.93	20.93	19.75	11.66	16.43	15.06	6.93	14.86	13.83
PM	72.53	70.67	68.10	63.99	70.75	68.01	56.02	67.25	67.89
$\omega_{\rm c}$	12.82	18.19	20.54	18.11	27.23	31.63	23.80	27.59	37.41
$d_{\rm att}$	54.66	49.23	48.21	48.67	43.25	40.52	43.69	43.88	39.28
γ	2.243	2.249	2.286	2.576	2.562	2.603	2.995	2.957	3.013

[(i)] The time constant τ and the settling time $t_{\rm s}$ (taken for 95% of the final value) are measured in ms, the overshoot $M_{\rm p}$ in %, the gain margin (GM) and the open loop magnitude attenuation d_{att} in dB, the phase margin (PM) in degrees and the gain crossover frequency $\omega_{\rm c}$ in rad/s.

[(ii)] The GM, PM, $\omega_{\rm c}$ and d_{att} are all computed for the open loop transfer function, with the loop opened before the actuator.

`Controller results`

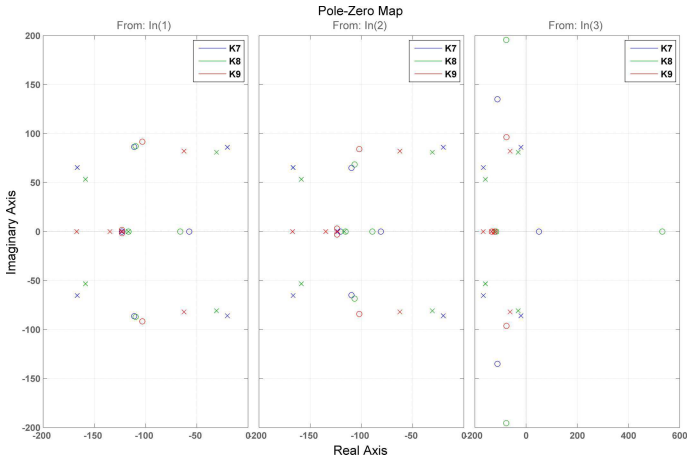
The poles of each $\mathscr{H}_\infty$ controller are shown in Table 5.3 whereas the poles and transmission zeros for each of the three I/O channels are shown in Fig. 5.4 [11]. It may be observed that the synthesis algorithm provides well-behaved controllers in terms of pole location (avoids excessively fast modes) and I/O zeros (except for some cases in controllers No. 8, 9 where some non-minimal phase zeros appear).

In Figs. 5.5a-5.5b, simulations of each of the nine closed loops (corresponding to the synthesis points) are demonstrated. In the first figure, step responses of the vertical acceleration are shown whereas in the second the corresponding control signal rates are presented.

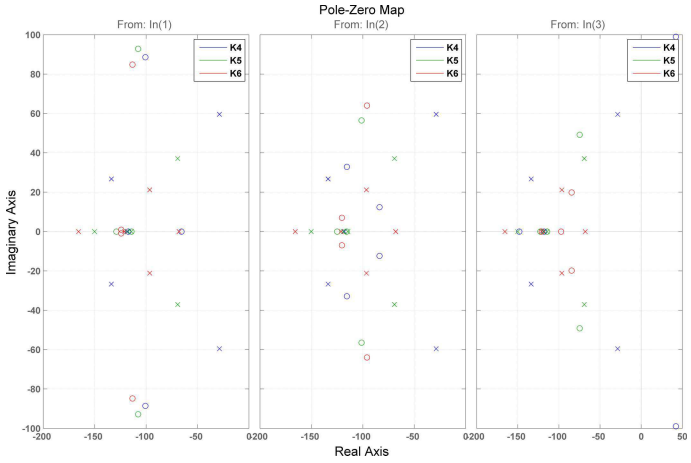
Table 5.3: Controller poles.

Points	**Poles**
1	$-48.71 \pm 9.63j, -120.94, -129.22 \pm 8.01j$
2	$-38.42, -103.16 \pm 25.32j, -123.32, -130.64$
3	$-55.76, -99.39 \pm 35.15j, -123.19, -136.45$
4	$-28.66 \pm 59.56j, -118.1, -133.8 \pm 26.71j$
5	$-69.23 \pm 37.07j, -114.51, -118.92, -150.16$
6	$-68.02, -96.49 \pm 21.16j, -121.05, -165.68$
7	$-20.08 \pm 86j, -123.41, -166.74 \pm 65.34j$
8	$-30.75 \pm 80.9j, -116.81, -158.64 \pm 53.34j$
9	$-62.46 \pm 82.08j, -123.01, -134.77, -167.26$

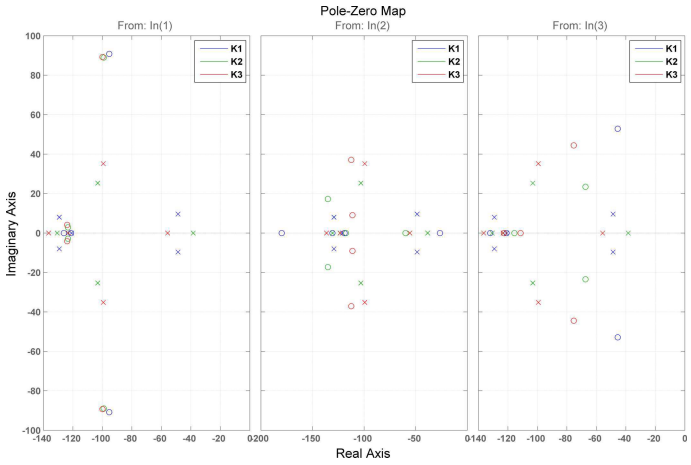
[11] The controllers are depicted in triplets, corresponding to constant Mach numbers: $M = 1.5, 2.25, 3$.



(a) *Controllers No. 7,8,9*

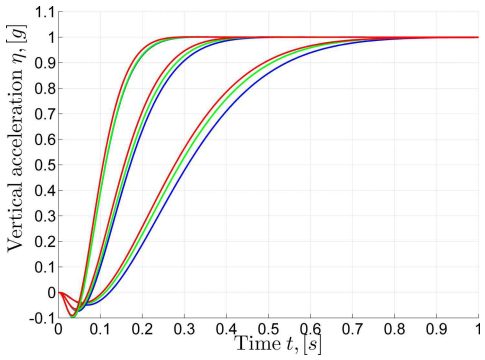


(b) *Controllers No. 4,5,6*

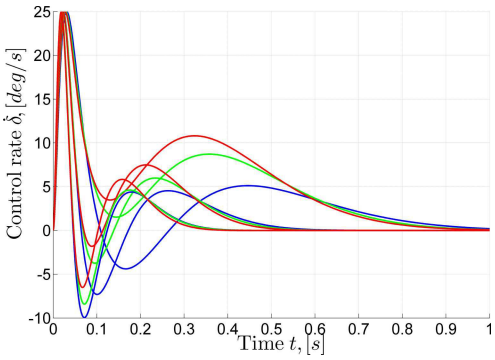


(c) *Controllers No. 1,2,3*

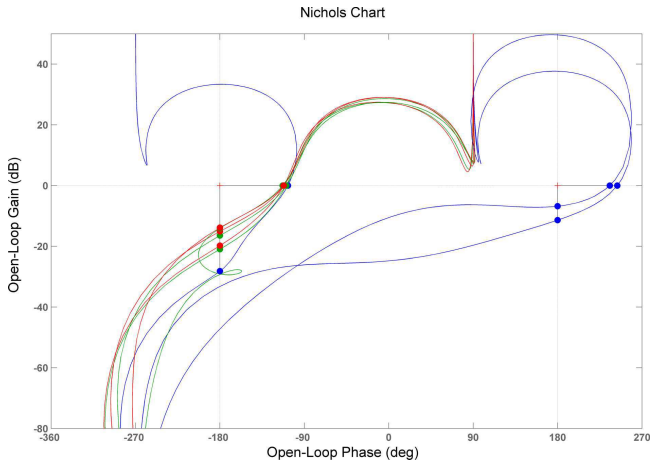
Figure 5.4: Controllers' I/O poles & zeros.



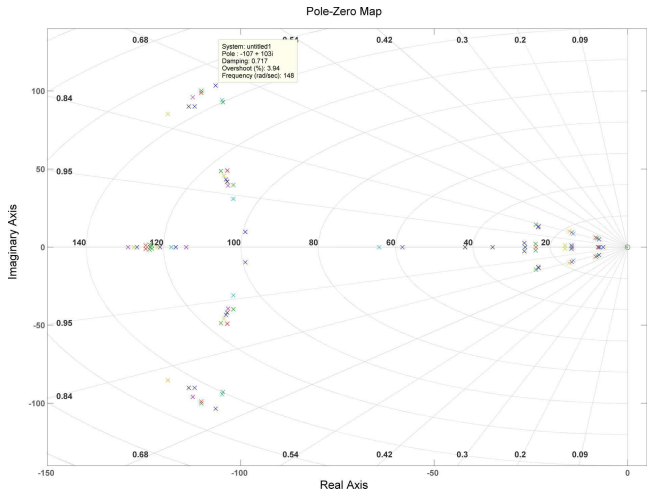
(a) *Output step responses (closed loop)*



(b) *Control signal rates (closed loop)*



(c) *Nichols charts (open loop)*



(d) *Poles-zeros (closed loop)*

Figure 5.5: Simulation results for the linear plants.

Concerning the outputs[12] (see Fig. 5.5a), it may be remarked that the controllers provide excellent damping and thus the corresponding step responses demonstrate practically no overshoot (see also Table 5.2). In addition, it may be remarked from ig. 5.5b that the controllers exploit all available bandwidth (reaching the maximum of 25deg/s) and thus provide a time constant in most cases well below the performance specifications ($\tau \leq 350$ms) [13].

Concerning now the Nichols charts, they present a very good visualization of the open loop gain & phase margins for every linear synthesis point. Once again the smallest, but still adequate values, are obtained for synthesis point No. 7. The stability margins may also be seen from the corresponding Bode diagrams depicted in the following Fig. 5.6. In addition to these margins, and the gain & phase crossover frequencies, the very good magnitude attenuation margins d_{att} of the robustness objective R_2 can be observed (the grey box shows the attenuation constraint). Finally, in Fig. 5.5d are shown the closed loop poles of the transfer function $T_{w\zeta_\infty}(s)$ for every synthesis point. It may be observed that they are indeed inside the convex LMI region $\mathcal{D}(\lambda_{\min}, r_{\max}, \vartheta_{\min})$[14].

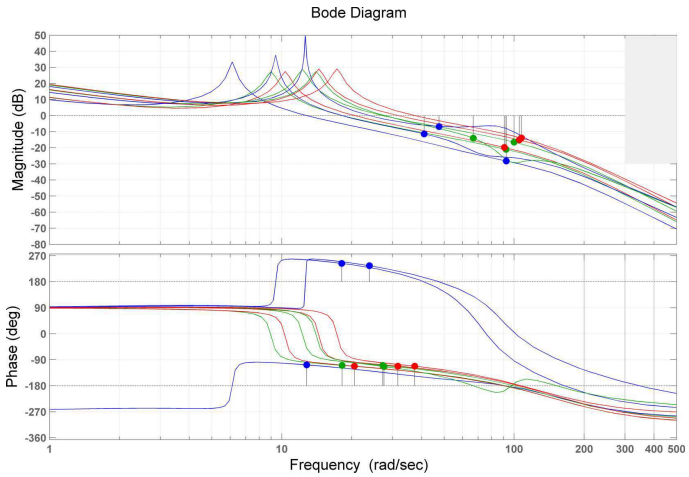


Figure 5.6: Bode diagrams (open loop).

As a last comment, it should be stressed that even though the $\mathscr{H}_\infty$ controllers are really of very good performance, they remain complex since the final controller to be implemented is the fifth order controller $K(s)$ plus the integrator; all this for a second order plant. In addition, it remains to be seen in the next section if the interpolation strategy chosen (controller blending) merits such a complicated LTI synthesis approach.

[12] All the time responses are again grouped in triplets for constant Mach numbers (blue corresponds to synthesis points No. 1,4,7, green to No. 2,5,8 and red to No. 3,6,9 respectively).

[13] Except for synthesis point No. 7 where there is a small violation. Now this may be expected since the open loop dynamics are here *unstable* (for more details see the missile open loop stability discussion of Chapter 4).

[14] The minimum decay rate $\lambda_{\min}$ changes for each synthesis point according to Table 5.2.

5.4.2 Gain-scheduled Controller

In this section the gain-scheduled controller, using the controller blending method and the $\mathscr{H}_\infty$ controllers of the previous section, will be detailed. The analysis starts with some details on some practical issues concerning the controller interpolation/realization; the next section presents simulation results.

5.4.2.1 Practical Issues

Inner control structure

The Simulink block diagram of the total simulated plant and the gain-scheduled controller is presented in Fig. 5.7. The grey boxes correspond to the missile nonlinear dynamics, Mach number generation trajectory and their appropriate initialization blocks. The blue boxes are the error integrator, the gain-scheduled controller and a block providing trim values for the pitch rate q according to the scheduling vector value in the missile flight envelope[15]. The red box is the reference trajectory generating block and finally the yellow boxes are the actuator and a first order filter acting on the reference trajectory. The latter is used in order to smoothen the passage of the scheduling vector through the four scheduling regions $\Gamma^1, \Gamma^2, \Gamma^3, \Gamma^4$ and thus facilitate the interpolating procedure.

The gain-scheduled controller block interior is more complicated and is shown in Fig. 5.8; its major functions are performed mainly by four block families depicted in different colors. The *blue* colors depict the four adjacent controllers the interpolation procedure needs, in order to interpolate between their signals and provide the final control law. Each block realizing a dynamic controller, needs the appropriate controller matrices $\mathbf{A}_\mathrm{k}, \mathbf{B}_\mathrm{k}, \mathbf{C}_\mathrm{k}$ and $\mathbf{D}_\mathrm{k}$, appropriate state initialization and reset when moving from one scheduling region to the next, according to the values of the scheduling vector $\varrho = [\eta_\mathrm{r}\ M]^T$.

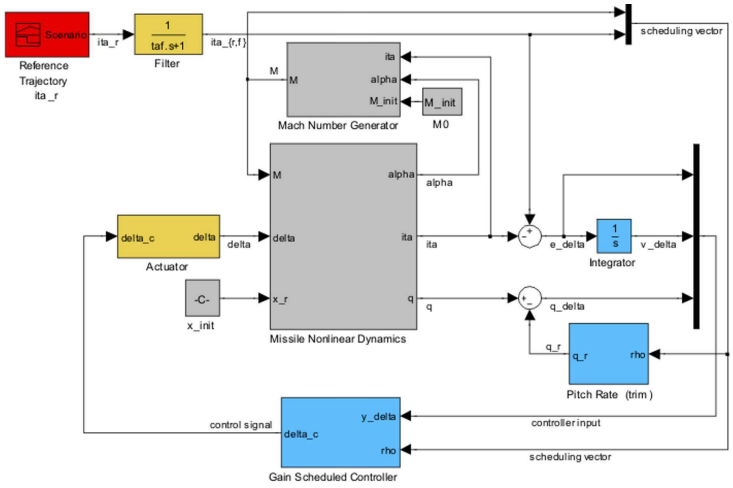


Figure 5.7: Total simulation block diagram (controller blending).

[15]The last is a necessary box since the input to the controller is $q_\delta = q - q_\mathrm{r}$, where $q_\mathrm{r} = q(\varrho_\mathrm{r})$.

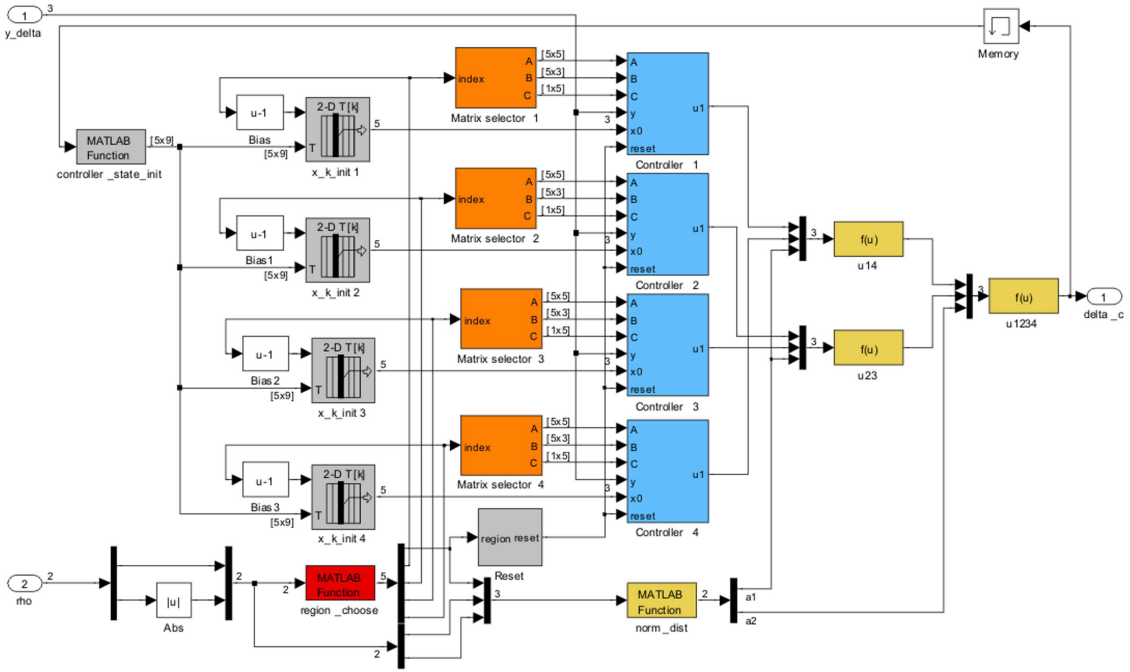


Figure 5.8: Total simulation block diagram (controller interior).

State matrices

The controller state matrices of each controller are provided by an *orange* block that is in fact a look-up table. Inside this table are stored the controller matrices of all nine synthesis points and they are appropriately selected according to a signal *index* that dictates the current scheduling region. This signal is the output of another block (red color) that outputs this region number ($\Gamma^1, \Gamma^2, \Gamma^3$ or Γ^4) [16] according to the value of the scheduling vector.

Interpolation

Now for each region number corresponds a different quartet of controllers (in total there are nine controllers and each time only four are used). This is important when coming to interpolation since each time, in order to compute the final interpolated signal, the controllers on the left (respectively right) side corners of the current scheduling region are first combined, each combination providing a control signal. Then, these two interpolated signals are once again combined to compute the final control law (see Eqs. 1.88-1.90 for more details). This quartet of controllers in the lower left (ll), lower right (lr), upper right (ur) and upper left (ul) corners of the scheduling region is also the output of the red block that gives the scheduling region number[17].

Controller reset

Recall now from the analysis of Section 1.3.2.2, concerning the controller blending method, that when changing scheduling regions and charging different controller matrices, the control signal will be discontinuous due to the incompatibility of the DC gains of the various controllers entering and leaving the algorithm. This may be corrected by guaranteeing a bump-less control signal using the analysis of the aforementioned section. Briefly, this is done by *re-initializing* all controllers of the new scheduling region at the transition time to an appropriate state. This state is calculated from Eq. 1.91 using the control signal at the transition time and the output controller matrices of the newly entered scheduling region. This is done using the *grey* blocks of Fig. 5.8: the four vertically aligned ones re-initialize the controllers at the appropriate state using the grey block on the upper left whereas the state reset command is given by the grey block on the center.

Normalized distances

Finally, the yellow blocks calculate the interpolated final control signal. The two vertically aligned yellow boxes on the left blend the controllers' signals by pairs as detailed above (ll-ul and lr-ur respectively) and the one in the right blends these two, to provide the final control signal. The yellow block on the page center computes the normalized distances a_1, a_2 (with $0 \leq a_i \leq 1$) used by the aforementioned control signal interpolation blocks (see Eqs. 1.86-1.87)[18].

[16] See Fig. 5.2.

[17] For example, for the scheduling region Γ^4, the ll, lr, ur, ul, corners correspond to controllers No. $5, 6, 9, 8$ respectively.

[18] As a last comment it should be stressed out that given the fact that the interpolation regions Γ^i are not rectangular as assumed in Section 1.3.2.2 (for simplicity) but trapezoids, with the upper and lower sides being parallel to each other, the interpolation procedure needs some more trigonometry. For the generic trapezoid of Fig. 5.9, the normalized distances are defined as: $a_1 = l^1/l^{\{1,4\}}$ and $a_2 = l^2/l^{\{1,4\}\leftrightarrow\{2,3\}}$. The difference from a rectangular region is that the distances $l^{\{1,4\}}, l^{\{1,4\}\leftrightarrow\{2,3\}}$ are also time varying (except for the other ones l^1, l^2 that are always time varying because of the scheduling vector motion inside the flight envelope.

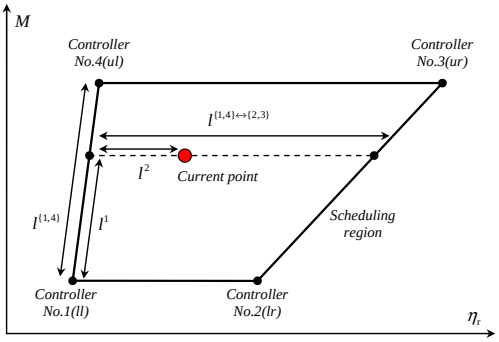
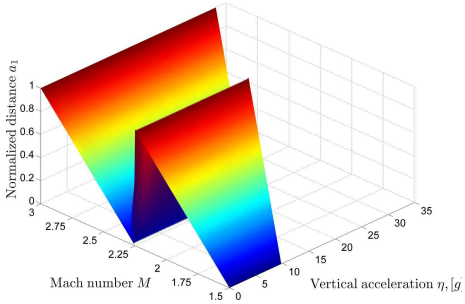


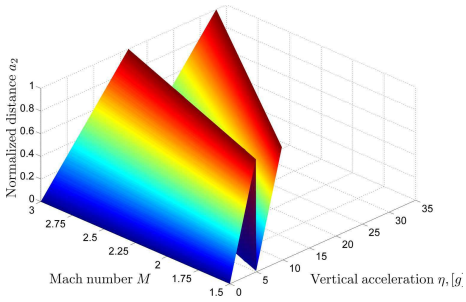
Figure 5.9: Generic trapezoidal scheduling region.

These normalized distances a_1, a_2 are visualized for the whole missile flight envelope in Figs. 5.10a-5.10b. As expected, the first one is clearly augmenting inside the scheduling regions Γ^1, Γ^2 and Γ^3, Γ^4 as the Mach increases, whereas the second one is augmenting inside the regions Γ^1, Γ^4 and Γ^2, Γ^3 as the vertical acceleration increases.

As a last comment it may be added that the trajectory reference tracking is assured by the integrator at the input of the controller. An alternative to that is to decompose the control signal δ_c into an open loop control signal providing a trim control input $\delta_{c,r}$ (as a function of the scheduling vector) and a closed loop interpolated control signal $\delta_{c,\delta}$ stabilizing the missile (see for example the conference paper [137]).



(a) *Normalized distance* a_1



(b) *Normalized distance* a_2

Figure 5.10: Scheduling region normalized distances.

5.4.2.2 Simulation Results

In this section, the simulation results of the nonlinear controller will be presented. Contrary to other existing works, where the nonlinear controller is not thoroughly tested due to scheduling vector reference trajectories not covering the whole missile flight envelope, here some rather stringent scenarios are considered.

Simulation scenario

The Mach number trajectory given by Eq. 4.10 (taken from the original benchmark paper [103]) covers only a small part of the missile operating domain when combined to the corresponding vertical acceleration reference trajectory given in the same work. To obtain a more realistic scenario, here the drag coefficient A_{x} (see Table 4.1) has been augmented in order to provide a steeper descent for the Mach number. In addition, the vertical acceleration profile has been slightly modified with respect to the aforementioned work.

Mach & output

In Fig. 5.11a this Mach trajectory is visualized whereas in Fig. 5.11b the vertical acceleration reference trajectory $\eta_{\mathrm{r}}(t)$ (black), the filtered reference trajectory $\eta_{\mathrm{r,f}}(t)$ (red) and the actual response of the system $\eta(t)$ (blue) are demonstrated. Finally in Fig. 5.11c, the output trajectories are plotted on the missile flight envelope (using the same controllers). The general behavior of the gain-scheduled controller is rather acceptable; however there many issues that will be detailed further on.

Control signals

In Fig. 5.12a is shown the total control command $\delta_{\mathrm{c}}(t)$ of the global gain-scheduled controller given to the actuator (red) and the filtered one $\delta(t)$ that is the actual input to the system. In the same figure is also depicted the scheduling region number (either 1, 2, 3 or 4 corresponding to regions $\Gamma^1, \Gamma^2, \Gamma^3$ or Γ^4).

In Fig. 5.12b are shown the four controller outputs (corresponding each time to each of the four corners of each scheduling region) and once again the interpolated control signal $\delta_{\mathrm{c}}(t)$ (in red and blue respectively). In addition, the controller state reset signal is shown (being in fact accordant to the scheduling region number signal of the previous figure). Finally, in Fig. 5.12c the normalized distance signals a_1, a_2, that are used to interpolate the four controller signals, are illustrated; taking obviously values between zero and one.

5.4.2.3 Discussion

Even though the time performance of the controller blending method is good (see Fig. 5.11b), there exist several inconveniences due to the fact that the controllers need to re-initialize when changing scheduling region.

This fact causes control signal transients, chattering and degrades the overall time performance of the system[19]. Consider for example to different cases observing Figs. 5.11-5.12: switching due to the vertical acceleration (Case 1) and due to the Mach number (Case 2).

[19] When referring to time performance, output tracking is considered in most cases since this is the primary goal of control systems in this work.

Case 1. The simulation scenario chosen involves four changes on the desired output vertical acceleration (see Fig. 5.11b):

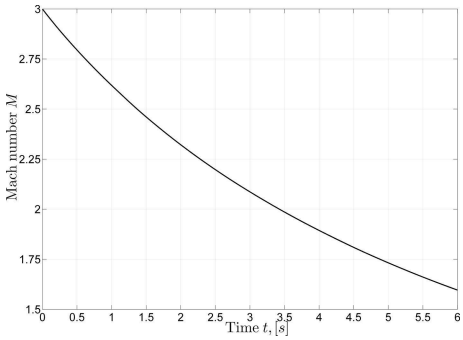
$$\eta_{\mathrm{r}}(t) : 0g \rightarrow 25\mathrm{g} \rightarrow -15\mathrm{g} \rightarrow -10\mathrm{g} \rightarrow 5\mathrm{g}$$

These changes correspond to the first three and the fifth state resets of Fig. 5.12b (see reset signal impulses). Consider now just one of them (the first) since the controller behavior is similar for all four. At $t = 0s$, the output reference signal (see Figs. 5.11b-5.11c) changes its value. Given that the state reset signal is based on the *filtered* reference signal, it gives the command to reset the states of the four controllers approximately at $t = 0.15s$, the time it takes the filtered signal to cross the boundary of the fourth and third scheduling regions respectively (see Fig. 5.11c).
As a consequence, the control signals exhibit a transient behavior right afterwards (see Fig. 5.12a) before settling down and controlling the plant. This fact may be observed equally on the four controller signals of Fig. 5.12b. This behavior is clearly undesirable and undermines the plant performance but is unavoidable if this interpolation method is used.
This is once again due to the need to change the *whole* controller when passing on to a subsequent region; in general two controllers are switched on and off respectively, except for the extremely improbable case that the scheduling vector crosses the scheduling boundary at a synthesis point and thus three controllers need to be switched.

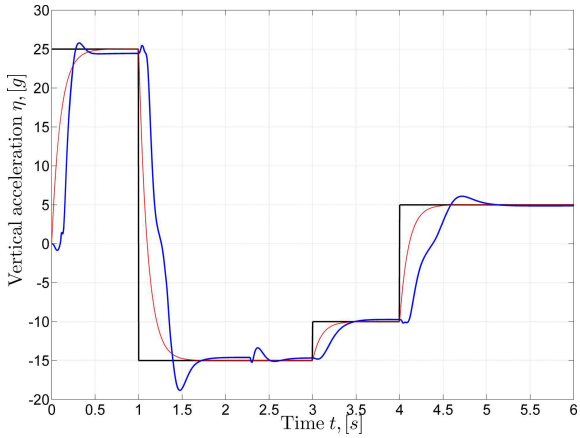
Case 2. Consider now a scheduling region change due to the Mach number. Refer again to Figs. 5.11b-5.11c and consider the crossing of the reference trajectory at $t = 2.3s$ (this can be equally seen from the reset signal of Fig. 5.12b) due to the Mach number falling below the value 2.25.
This causes the scheduling vector passing from region Γ^4 to region Γ^2 and thus the controllers switch and re-initialize[20]. The latter fact causes a transient behavior and the output η demonstrates a small oscillation around its steady state value (that had already been established at $t = 2.3s$).
It is clear that this issue is even more important; if the user is not so lucky and this switch due to the Mach number falls during the transient of a switch due to the vertical acceleration also, then the performance of the system is even more deteriorated because of the combination of both effects.

These two study cases demonstrate in practice the greatest disadvantage of the *controller blending* method: controller re-initialization. In the next section an alternative method is considered based on state feedback/observer interpolation, solving this problem since the controllers are not switched but rather structurally modified.

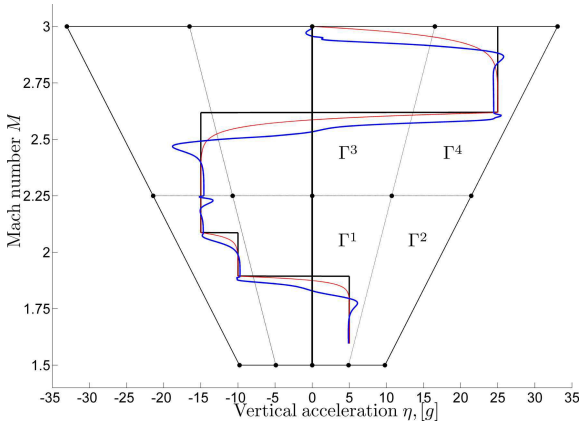
[20]Recall that the missile flight envelope is symmetrical and at the switching time $\eta_{\mathrm{r}} < 0$; thus the system functions on the left symmetric side of the flight envelope.



(a) *Mach number*

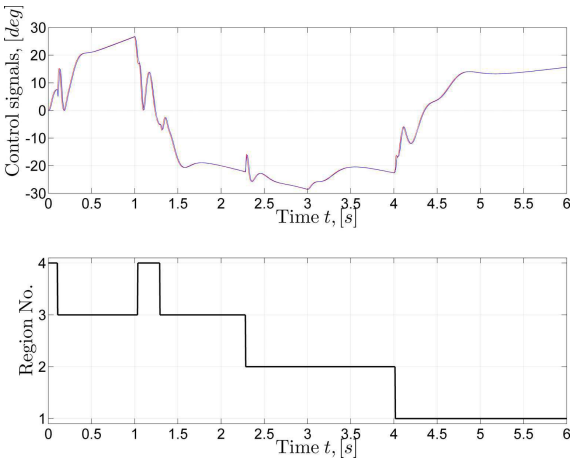


(b) *Vertical acceleration*

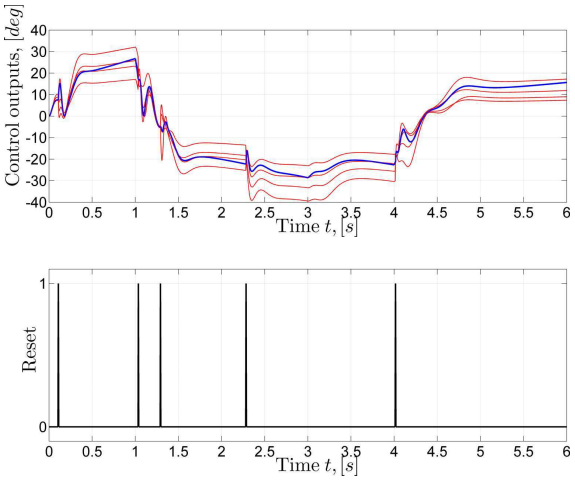


(c) *Flight envelope trajectories*

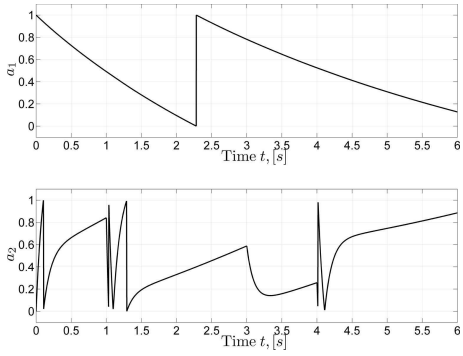
Figure 5.11: Controller blending nonlinear simulations (output).



(a) *Actuator command and region number*



(b) *Controller outputs and reset signal*



(c) *Normalized distances*

Figure 5.12: Controller blending nonlinear simulations (control).

5.5 Observer/State Feedback Interpolation

In the previous section, $\mathscr{H}_\infty$ controllers were computed at nine operating points and then their outputs interpolated at four scheduling regions, in order to compute the final gain-scheduled control law. In this section another interpolation method for the control of the Reichert missile will be detailed, based on the state feedback/observer interpolation technique presented in Section 1.3.2.6.

5.5.1 LTI Synthesis

LTI control structure

The scheduling regions and the interpolation geometry used for this method are the same as the ones used with the controller blending method. The key difference is in the *interpolation* method itself. The idea here is to convert all nine $\mathscr{H}_\infty$ controllers of the previous section in estimator-controller form (see Section 3.2) and then obtain a gain-scheduled controller by updating/interpolating its inherent structure (gains, matrices).

As far as LTI controllers are concerned, they have the following general standard form:

$$\dot{\hat{x}}_\delta = \mathbf{A}\hat{x}_\delta + \mathbf{B}u_\delta + \mathbf{K}_\mathrm{o}(y_\delta - \mathbf{C}\hat{x}_\delta) \tag{5.14}$$

$$u_\delta = -\mathbf{K}_\mathrm{c}\hat{x}_\delta. \tag{5.15}$$

The matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are in fact the open loop dynamics matrices $\mathbf{A}_\mathrm{aug}, \mathbf{B}_u$ and $\mathbf{C}_y$ of the standard model of Eq. 5.7. As a result the estimator shall inevitably reconstruct all the state vector x_aug of the open loop system and perform then a state feedback pole placement in order to control it.

5.5.2 Gain-scheduled Controller

Controller realization

A simplified block diagram of the state feedback/observer-based gain-scheduled controller is shown in Fig. 5.13. The controller is constructed using a standard state space realization of a Kalman observer (see Eq. 5.14 above). The inputs to the observer are the plant's input and outputs plus the structure of the observer (state feedback/observer matrices and plant matrices). Now all these matrices should be supplied to the observer/controller according to the system's operation using the block named 'Interpolator' for each corner of the scheduling region and the value of the scheduling vector ϱ.

Method issues

Consider first the state feedback/observer matrices $\mathbf{K}_\mathrm{c}, \mathbf{K}_\mathrm{o}$: the dimensions of these matrices is 1×5 and 5×3 respectively, thus the total number of coefficients needing interpolation is twenty (!), which is a rather big number for real world implementation[21]. This is of course done with the same procedure as before concerning controller blending, using normalized distances between controllers for each synthesis points and linear interpolation.

[21] Compare with the controller blending interpolation where only four signals were interpolated.

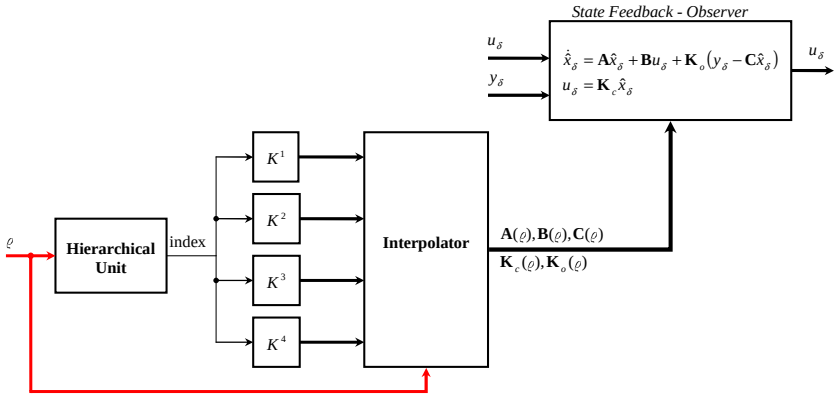


Figure 5.13: State Feedback - Observer Gain-Scheduled Controller.

`Inter/tion` The most unrealistic thing however concerning this method is the fact that in order to reconstruct the state of the plant, the observer needs information on the system's structure, that is the matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$. In order for the observer to reconstruct the state, these matrices should be computed for each reference/operating point of the plant. This may done in three different ways: *symbolic computation*, *tabulation* or *interpolation*.

Symbolic computation means that the formulas in Chapter 4 (e.g. Eqs. 4.36-4.44) concerning in fact the LPV model of the missile should be evaluated in real time for each reference/equilibrium point. Tabulation means that the corresponding surfaces of these equations for every value of the scheduling vector (see Fig. 4.6) should be stored in memory and retrieved using the value of the scheduling vector. Finally interpolation means that this storing procedure may be done only for the nine synthesis points and then use interpolation for every other intermediate point.

`Implem/tion` It is clear that each of these methods presents advantages and disadvantages but globally, all three are not so realistic. The first one is evident that is totally not feasible for real world implementation (even though it offers the best results) since symbolic calculations are very costly in terms of hardware implementation. The second one could be considered but it would be also costly in terms of storage memory for a real world system or for a system that has more than two scheduling variables where coefficients hyper-surfaces should be calculated. Finally, the third method lacks precision because it is clear that nine points could not necessarily cover the whole domain of operation correctly. In addition, interpolation would be needed in order to obtain a value for the system matrix coefficients for every operating point.

This method however, even though it presents all these disadvantages demonstrates a clearly better performance than the controller blending one on various aspects. Simulation results will not be presented here since they may be found in the comparison paper [137] where the two approaches are put side by side and all their advantages and disadvantages stressed out.

Performance

Concerning the advantages of this method, it should be at least cited its excellent output time performance and the avoidance of problems regarding controller re-initialization as with the controller blending method. Clearly, it has been already stressed out in the previous section that the main disadvantage of the controller blending method is the not coherent state initialization when passing from one scheduling region to the next and the resulting control transients during this procedure. With this method, this annoying fact is avoided since the controller structure is fixed and only its inherent parameters interpolated whereas with the controller blending method, four controllers need to be realized at each scheduling region with possibly inconsistent I/O representations.

Pole partitioning

A last delicate matter concerning this interpolation method based on state feedback/observer control has to do with the issue mentioned in the discussion at the end of Section 3.2.2, concerning the partitioning of the closed loop eigenvalues between the controller and observer. It is really important that this partition be done in an automatic way since this is the essence of gain scheduling: have a systematic and repetitive manner of doing things for a generic parameter-dependent system. However, this fact is not always easy since the closed loop poles resulting from the $\mathscr{H}_\infty$ controllers may be also complex conjugate or real or a mixture of two and of *different* multiplicity/speed for each synthesis point. Thus the choice of closed loop assignment is clearly not trivial as it may be also seen from the analysis in [4] or even in [20, 21].

5.6 Conclusions

In this chapter two ad-hoc interpolation strategies using the *controller blending* and *state feedback/observer scheduling* methods were tested and compared. Much attention has been given on issues when using these approaches for the control of real world systems.

As far as these issues are concerned, it has been stressed out that the controller blending method seems to be the simplest one in terms of calculations (only four signals are interpolated) with respect to the state feedback/observer one (all control/system matrices are interpolated). However the latter one is easier to implement since only one controller is considered and only its structure interpolated whereas the first one needs always four controller that run in real time. The state feedback/observer method is also of higher performance since it avoids control signal transients and chattering caused by controller re-initialization. However, it remains an open issue on how its LTI controllers should be calculated: should they be chosen in a standard 'controller poles three times slower than the observer ones' or via the Youla parametrization-based conversion detailed in the previous section. If the first method is used, the user loses the highly desirable robustness properties of $\mathscr{H}_\infty$ control theory, whereas if the second method is used, the partitioning of the closed loop poles is not trivial when more than one operating/synthesis points are considered.

Seeing things in a global manner, it is evident that these two methods are both complicated and ad-hoc. The latter issue comes from the fact that regarding LTI controller computation there is practically *no guideline* if the nine synthesis points considered were too few, too many, appropriately partitioned in the flight envelope etc. In addition, the control structure used is too complicated; a sixth order controller for a second order plant.

As a result, a simpler, more efficient and more systematic way to treat the problem is needed that will offer an elegant and practical solution for the control of this type of nonlinear parameter-dependent systems. The next and most important chapter of this work is exactly devoted to that.

Chapter 6

Systematic Control Strategies

Overview

In this chapter the main results of this work are presented. A new, systematic and easy to implement gain-scheduled control strategy is proposed: the extended Loop Shaping Design Procedure (or *e*-LSDP). This method is based on the *gain blending* interpolation method and uses the McFarlane&Glover standard LSDP coupled with the gap metric theory in order to provide a gain-scheduled controller that takes into account the nonlinear dynamics variation as a function of the system's operating conditions. This procedure is applied both to the Reichert and ARV benchmark examples to obtain a gain-scheduled autopilot; for both systems an exhaustive analysis is presented focusing on the features and advantages that this method presents over the *ad-hoc* ones analyzed in the previous chapter.

Chapter contents

6.1 Introduction

In this chapter a novel method, for the control of the two benchmark examples (the 'Reichert' missile model and the atmospheric re-entry vehicle) introduced in Chapter 4, will be presented. This method is the *gain blending* method detailed in Section 1.3.2.7 and is coupled with the $\mathscr{H}_\infty$ loop shaping and gap metric theory of Sections 3.3-3.4, in order to provide a systematic control strategy that treats the inconveniences of the two methods detailed in the previous chapter. Novel approach

As it has been already remarked, the two major disadvantages of these methods are *complexity* and *lack of performance-robustness*. Complexity results from two factors: relatively high-order LTI controllers to be interpolated and unknown number of synthesis points required in order to obtain a good coverage of the system's operating domain. Lack of performance results from the interpolation strategy chosen (see for example control transients from the controller blending method) and from the number of synthesis points considered, whereas lack of robustness results from the absence of a systematic/general way to take into account uncertainty in the feedback loop of the gain-scheduled controller. Motivation

The solution of these problems is not easy; to the author's opinion, one straightforward way to treat complexity is by selecting simple to tune yet performing controllers such as PID in order to obtain a basic compensation of the nonlinear system at a relatively small number of synthesis points (e.g. using the corners of the flight envelope or an intuitive selection like in the previous chapter). By interpolating the gains of such controllers an acceptable compensation is taken as a basis, in order to compute an *additional* number of synthesis points where simple enough static $\mathscr{H}_\infty$ controllers (based on the initial loop shaping by the PID ones) are obtained. These $\mathscr{H}_\infty$ controllers capture the nonlinearity of the system by the use of the notion of the gap metric as it will be shown in this chapter. Thus, they correct in a way the somewhat 'ad hoc' loop shaping of the PID controllers since they act on the control signal components and are afterwards interpolated using a triangulation of the system's operating domain. In order to show in fact this amelioration, the 'ad hoc' controllers are compared to the combination of these loop shaping controllers and the additional $\mathscr{H}_\infty$ ones. Solution

The control strategy proposed is applied to the control of the 'Reichert' missile benchmark and to the ARV provided by the EADS foundation. It is clear that the work presented here corresponds to the three last steps of the LBGS procedure detailed in Section 1.3.1; namely to the *LTI Controller Synthesis*, *Gain Interpolation* and *Global Controller Implementation & Validation* steps.

This chapter is divided in two parts: in the first one the gain blending method is applied on the ARV to obtain a regulating autopilot[1], whereas the second is devoted to the missile where the same control strategy is adapted with slight modifications (due to the different control objectives).

[1] The regulating autopilot is computed for the ARV re-entry vehicle using only the systematic *gain blending* method and not the ad-hoc methods of Chapter 4 since this corresponds to the main work of this thesis.

6.2 Gain Blending (re-entry vehicle)

The ARV vehicle

Recall from Section 4.2 that the ARV nonlinear dynamics are described by a state vector that is comprised by the AoA α (in rad) and the pitch rate q (in rad/s). The control input is elevator deflection signal δ_{e} (in rad) whereas the measured output is the AoA. The elevator dynamics are governed by a second order ODE and the rectangular flight envelope $\mathbf{\Gamma}_{\mathrm{fe}}^{[\alpha,M]}$ is a function of the Mach number M and the AoA. The nonlinear aerodynamic functions $C_{\mathrm{m0}}(\alpha, M), C_{\mathrm{me}}(\alpha, M)$ associated with the pitch rate are tabulated for every value of the scheduling vector $\varrho = [\alpha\ M]^T$ inside this envelope[2].

Control objectives

The problem here is to obtain a regulating autopilot that will maintain the AoA to a constant value α_{r} for a given variation profile of the Mach number and the dynamic pressure $Q(M)$, which is an additional non-measurable time-varying variable.

The ARV control objectives are mostly precision and robustness ones and are the following:

Regulation & Flight Envelope. The autopilot should be able to regulate the AoA around a pre-defined reference value α_{r} with $\pm 1\%$ step response steady-state error accuracy when the Mach number follows a given time trajectory inside the vehicle flight envelope.

LTI Synthesis Objectives. The linear controllers designed on the synthesis points should provide at least 50° phase margin and 12dB gain margin. In addition, the dominant closed loop poles must have a damping of at least 0.45 and natural frequencies greater or equal to the dominant open loop ones. Finally, the control effort should be minimized in terms of variation rate.

Gain-scheduled Controller Objectives (nominal). The gain-scheduled controller should be implemented with a sampling period $T_{\mathrm{s}} < 0.15$s and the frozen time open loops during all the Mach time variation range should have at least 30° phase margin, 6dB gain margin and 1 control period delay margin.

Gain-scheduled Controller Objectives (uncertain). The gain-scheduled controller should provide at least 3dB gain margin and one half control period delay margin when heavy additive uncertainties are introduced on the dynamic pressure, on the moment of inertia and on the aerodynamic functions.

The next section details the extended loop shaping design procedure (*e*-LSDP) that corresponds to the third step of the LBGS procedure (LTI Controller Synthesis) of Section 1.3.1. Of course all trim analysis-control, linear models are based on the discussion in Section 4.2 and may be equally found in [139].

[2] For more details see Section 4.2.1.

6.2.1 LTI Controller Synthesis

The e-LSDP (extended-Loop Shaping Design Procedure) devised in this work is based on the standard one of Section 3.3.2 and is used to compute a global gain-scheduled controller for a nonlinear parameter-dependent plant. It incorporates not only the LTI controller synthesis phase of Section 1.3.1 but also a systematic way of choosing the operating points where this synthesis will occur.

The e-LSDP corresponds as said, to the third step of the LBGS procedure of Section 1.3.1 concerning the LTI controllers used in order to cover the operating domain of the system. Additionally, in the robust $\mathscr{H}_\infty$ loop shaping - gap metric context of Chapter 3 an operating point choice algorithm is proposed. The e-LSDP is decomposed in three steps:

The e-LSDP

Step 1 - Loop Shaping. A linearized model of the vehicle $G(s)$ is obtained (see Section 4.2.3) for five synthesis points inside the flight envelope, namely on the corners and on the center of the envelope (see Fig. 6.1):

$$[\alpha, M] : [30^\circ, 4], [50^\circ, 4], [30^\circ, 26], [50^\circ, 26], [40^\circ, 15]$$

The linearized plant is then augmented using a pre-compensator $W_1(s)$ (which is actually the actuator dynamics) and a post-compensator $W_2(s)$ (which is a filtered PID controller) in order to provide basic performance and robustness requirements for the aforementioned five synthesis points corresponding linearized plants. The analysis on how to obtain these controllers is detailed in the next section. For any other point of the flight envelope, a linear interpolation of these gains is used by considering the four corresponding triangular interpolation regions $\Gamma^1, \Gamma^2, \Gamma^3$ and Γ^4.

Step 2 - Operating Point Algorithm. Given that a simple linear interpolation of the f-PID controllers is not sufficient as it will be seen in Section 6.2.3, an additional number of controllers should be used in order to treat the variation of the nonlinear dynamics of the vehicle. A variation indicator is the undamped natural frequency of the complex conjugate poles of the linearized plant for every value of the scheduling vector $\varrho = [\alpha \ M]^T$ (see Fig. 4.15). However, given that a *closed loop* criterion would be preferred, here the *gap metric* notion is used to quantify such variation.
Recall once again from Section 3.4.2, Theorem 3.6, that the gap metric between a nominal LTI plant G and a perturbed one G_Δ are closely related to the stabilizability of both plants by the same $\mathscr{H}_\infty$ controller K_∞ (designed for the nominal plant), and the robustness margin $\epsilon_{\max}$[3].
The autopilot regulates α around a constant α_r for all M; thus these additional synthesis points are sought on this line of the flight envelope; for the rest of the analysis concerning this algorithm see Section 6.2.1.2.

[3]Recall that the maximum robustness margin is smaller in the static case than in the full order one (see Section 3.3.4).

Step 3 - $\mathscr{H}_\infty$ Controller Synthesis. After the algorithm using the gap metric described in Section 6.2.1.2 has found the additional set of synthesis points, robust $\mathscr{H}_\infty$ controllers are computed at these points to increase the robustness of the final gain-scheduled controller. The additional controllers are static ones to reduce complexity and are of course also interpolated at intermediate points using the Mach number (and for constant α), in the fourth step of the LBGS procedure (see Section 6.2.2).
These two last steps of the *e*-LSDP are closely connected; the robust controller K_∞ may be computed after all synthesis points are found (since the robustness margin associated to a given synthesis point depends *only* on the initial loop shaping) or immediately when this point is found.
The nominal open loop plant $G(s)$ when using the operating point choice algorithm, is in fact the augmented plant $G_\mathrm{s}(s)$ obtained by the series interconnection of the pre/post compensators of Step 1 and is written as $G_\mathrm{s}(s) = W_2(s)G(s)W_1(s)$, for any operating point. As it has already been mentioned, the linearized vehicle model $G(s)$ varies as function of ϱ and so does the post-compensator $W_2(s)$, since its gains are the interpolation of the controller gains at the initial five synthesis points given in Step 1.
The final implemented controller $K_\mathrm{s}(s)$ for a synthesis point issuing from the algorithm is in fact the series interconnection of the pre/post compensators and the static robust controller. The robust controller becomes itself also interpolated at the fourth step (Controller Interpolation) of the LBGS procedure as it will be presented in Section 6.2.2.

`Discussion` The operating point choice algorithm treats in fact the variation of the open loop shaped linear dynamics $G_\mathrm{s}(s)$ between a nominal operating point and a subsequent one, as uncertainty. This uncertainty is in fact visualized in the gap δ_g between these two systems. Leaving details for Section 6.2.1.2, it will only be mentioned that by computing the gap between subsequent 'uncertain' systems and a nominal one, the designer can find out until what point this dynamics variation is tolerable by a robust controller K_∞ designed for the nominal point.

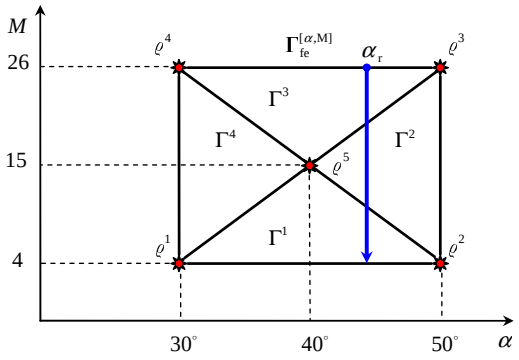


Figure 6.1: ARV flight envelope.

6.2.1.1 Loop Shaping

Basic control action

As it has been mentioned in the previous section, in order to obtain a basic control action for the ARV, five f-PID controllers are designed. These controllers are designed using the linearized, scheduling vector-dependent state space model $\mathcal{S}_{\text{LPV}}(\varrho_{\text{r}})$ of the ARV at these points given by Eqs. 4.58-4.68. The equivalent matrix transfer function of the vehicle $G(s)$ at each operating point gives the I/O relation of the control input δ_{e} to each state variable (AoA and pitch rate). Given that only the first state variable is measured, the plant is SISO with $G_\alpha(s)$ being the transfer function from the control input to the AoA.

Loop shaping

In the loop shaping context (see also Section 3.3), the transfer function $G_\alpha(s)$ is augmented by the actuator acting as a pre-compensator (with $W_1(s) = G_{\text{a}}(s)$) on the control signal, and by a filtered PID controller (f-PID) $G_{\text{c}}(s)$ acting on the regulation error $e_\delta = \alpha_\delta - \alpha_{\text{r}}$[4]. The control signal $\delta_{\text{e,c}}$ ('c' for commanded) is fed to the actuator that produces the final control stabilizing control signal $\delta_{\text{e},\delta}$. The loop shaping block diagram is shown in Fig. 6.2 with the corresponding open loop transfer function being:

$$G_{\text{s}}(s) = G_{\text{c}}(s)G_\alpha(s)G_{\text{a}}(s). \tag{6.1}$$

f-PID controller

The filtered PID controller used for the AoA regulation has the following transfer function:

$$G_{\text{c}}(s) = \frac{1}{\frac{1}{N}s + 1}\left(K_{\text{p}} + \frac{K_{\text{i}}}{s} + K_{\text{d}}s\right) \tag{6.2}$$

with $K_{\text{p}}, K_{\text{i}}, K_{\text{d}}$ being the PID gains and N the filter's time constant inverse.

Control challenge

As it has been detailed in Chapter 4 concerning the stability analysis of the linearized models of the vehicle, its open loop dynamics are conditionally unstable given that the poles are purely imaginary. The nonlinear control problem is challenging since their undamped natural frequency ω_0 changes as a function of the Mach number and the AoA, as it may be observed from Fig. 4.15. Here however the latter is mostly important since the autopilot is a regulation one around a constant α_{r}.

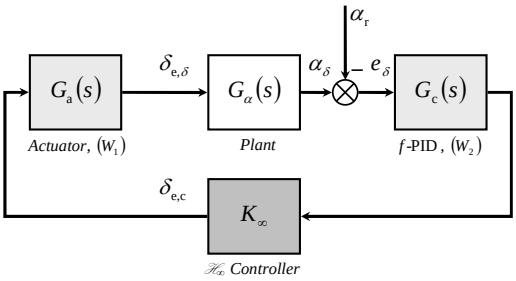


Figure 6.2: Loop shaping block diagram.

[4]The subscript 'δ' notation is used to emphasize around equilibrium operation.

The control goal is somewhat different from the Reichert missile one; the need for a classic signal tracking performance with appropriate rise times, settling times etc. is not crucial. Here the autopilot should mostly provide good stability, delay margins and damping while minimizing the control effort.

Consider now one synthesis point (e.g. point No. 1 for $\varrho^1 = [30^\circ, 4]$) in order to detail the correction needed by the f-PID controller. The vehicle's open loop transfer function $G_\alpha(s)$ presents two complex conjugate poles with an undamped natural frequency ω_0 and zero damping since these poles are purely imaginary. The PID controllers shall correct this fact by using its two complex conjugate zeros (adjusted by the three gains $K_\mathrm{p}, K_\mathrm{i}, K_\mathrm{d}$) and attract these poles into the negative complex plane. The integrator of the controller will provide zero steady-state error whereas the filtering part that is a first order transfer function limits the control effort and adjusts the bandwidth of the system.

In terms of frequency response, the choice of the controller's parameters is not trivial; here a classic Bode response correction is used to provide an initial adjustment whereas fine-tuning is performed by using MATLAB® Simulink Control Design toolbox and its optimization routines.

Controller zeros influence

The natural frequency of the controller's complex zeros $\omega_{0,z}$ is equal to $\sqrt{K_\mathrm{i}/K_\mathrm{d}}$ and plays a significant role in providing the correct gain and phase margins for the open loop plant as well as the bandwidth, combined by the filter action. It should be chosen near but a bit smaller than the open loop natural frequency of the plant's complex conjugate poles. Reducing this frequency by moving the zeros nearer the origin, the gain magnitude increases starting from a lower frequency and thus the gain crossover frequency ω_gc is increased[5]. In addition, given that the ω_gc increases and the phase continuously decreases to -180°, the phase margin gets smaller[6]. The phase crossover frequency ω_pc is almost one decade further on and is not so much influenced by the movement of the zero, however given that by reducing the zeros' frequency the loop gain increases, the gain margin decreases.

The damping now of the controller zeros is governed by K_p if the other two gains are fixed; its influence is more complicated on the frequency response. In general, if the damping is increased, the step performance of the plant is ameliorated with the cost of deteriorating the stability margins and augmenting the control signal amplitude needed.

[5]The open loop magnitude starts at low frequencies from a value dictated by the controller zeros natural frequencies and drops with -20dB/dec until the point where the zeros start to act and increase the gain. Then the gain increases even more (mathematically to infinity around ω_0) due to the imaginary poles of the plant before falling once again due to the filter pole with -40dB/dec for higher frequencies (-80dB/dec if the actuator poles are added).

[6]The open loop phase starts from -90° due to the integrator and then starts to increase due to the complex controller zeros until ω_0; then it suddenly loses -180° due to the plant's complex conjugate purely imaginary poles. However the phase remains sufficient due to the total phase added until ω_0 by the zeros (about 150°). It then continues to decrease due to the filter pole until -180°; if the actuator is counted also, then it continues to drop further on until -360° at high frequencies.

`Controller pole influence`

Concerning now the influence of the filter's pole, things are also a bit complicated: given that the pole's frequency is bigger than the gain crossover frequency, it does not directly affect it for small displacements. If it is reduced, it starts also reducing the total phase added by the controller (being the combination of the phase due to the complex zeros and the pole) and thus deteriorate the phase margin. However, once this frequency is chosen (roughly at the middle of the zone $[\omega_{\mathrm{gc}}, \omega_{\mathrm{pc}}]$) it may be fine-tuned using Simulink Control Design.

`Controller results`

These concepts may be seen in Fig. 6.3[7] where the transfer functions of the open loop plant $G_\alpha(s)G_\mathrm{a}(s)$, the compensator $G_\mathrm{c}(s)$ and the combined, corrected (or 'shaped') open loop $G_\mathrm{s}(s) = G_\mathrm{c}(s)G_\alpha(s)G_\mathrm{a}(s)$ are shown together.

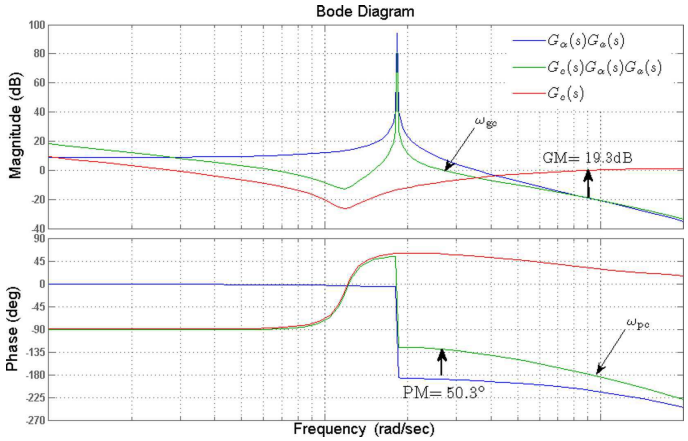


Figure 6.3: Correction open loop transfer function.

In Fig. 6.4 is shown the root locus diagram of the closed loop; in order to view the closed loop poles, the loop gain should be chosen as unitary.

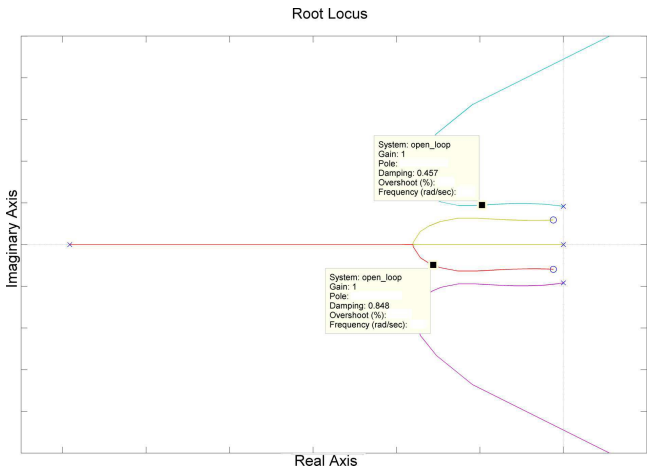


Figure 6.4: Open/closed loop poles diagram.

[7] Frequency values are omitted for confidentiality reasons.

The upper box in the previous figure shows the location of the vehicle's closed loop poles when the loop gain is unitary; the damping is satisfactory (0.457) as is demanded by the LTI synthesis objectives of Section 6.2. In addition, the controller closed loop poles are shown by the lower box; for high gains they tend to the open loop complex zeros. Here the actuator poles are not shown since they are much further on the left.

Fine-tuning & final results

Concerning now the fine-tuning performed using MATLAB® Simulink Control Design toolbox, it should be pointed out that it permits to optimize all four controller gains by putting constraints on the closed loop pole minimum damping and natural frequencies as well as on the stability margins. Performing this optimization for all points yields the results of Table 6.1.

Finally the five Nichols charts of the open loop systems are shown in Fig. 6.5a; the correct GM, PM achieved may be observed. In Fig. 6.5b are also shown the closed loop poles and zeros for all synthesis points.

Table 6.1: Loop shaping results[(i),(ii)]

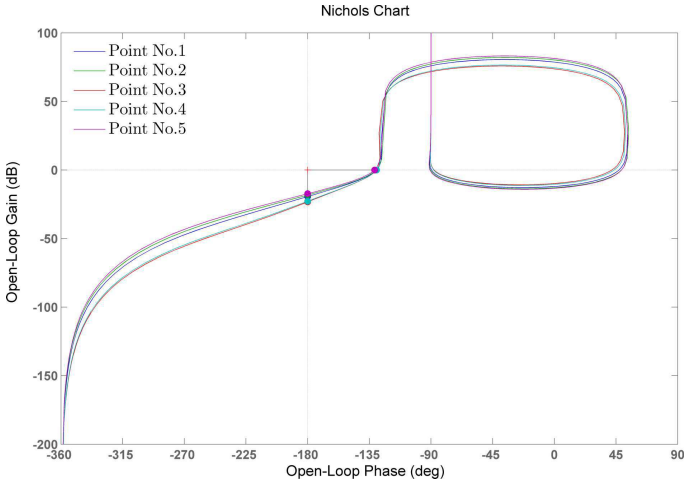
Points	**1**	**2**	**3**	**4**	**5**
K_p	0.04904	0.04879	0.06209	0.06868	0.06867
K_i	0.29011	0.36260	0.18882	0.22324	0.58750
K_d	0.20454	0.18348	0.38169	0.41269	0.23058
N	5.91630	7.43050	3.04130	3.25040	8.55560
GM	19.3	18.1	23.2	22.7	17.3
PM	50.3	50.2	50.3	50.4	49.0
ω_pc	2.65	3.15	1.54	1.64	3.59
ω_gc	9.18	10.2	6.71	6.93	10.8
t_s	4.77	3.98	8.31	7.84	3.46
$\lvert\dot{\delta}_{\mathrm{e},\delta}\rvert_\mathrm{max}$	12.7	13.8	12.8	14.8	20.4

[(i)] The gain margin (GM), phase margin (PM), phase crossover frequency (ω_pc), gain crossover frequency (ω_gc), settling time (t_s) and maximum control rate ($\lvert\dot{\delta}_{\mathrm{e},\delta}\rvert_\mathrm{max}$) are measured in dB, deg, rad·s^{-1}, rad·s^{-1}, s and deg·s^{-1} respectively.

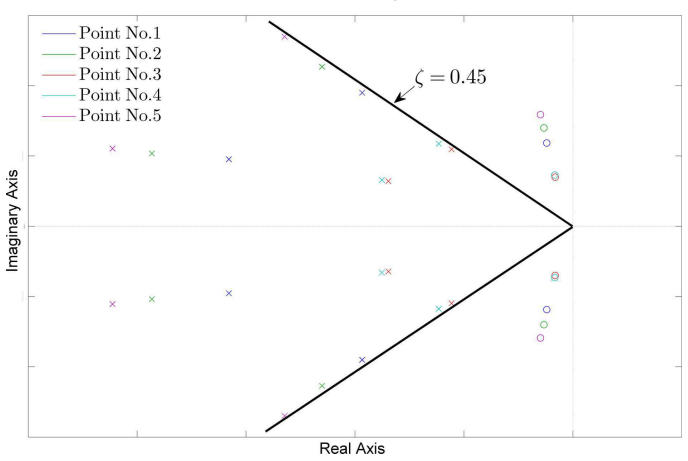
[(ii)] The settling time is measured within a 2% envelope.

If a gain-scheduled controller is constructed using only the family of five points and the corresponding interpolated controller gains are computed using the four scheduling regions of Fig. 6.1, then the results are not satisfactory in terms of stability margins both for the nominal and uncertain cases, as it will be presented in Section 6.2.3.

For this reason, some additional points are added on the line for constant α_r and robust $\mathscr{H}_\infty$ controllers are designed and also interpolated using the Mach number. The operating point choice algorithm is detailed in the next section.



(a) *Nichols charts*



(b) *Pole-zero maps*

Figure 6.5: Loop shaping results.

6.2.1.2 Operating Point Algorithm

Motivation

The operating point choice algorithm will add some additional synthesis points on the line α_r in order to ameliorate the robustness of the gain-scheduled controller detailed in Section 6.2.2. On these additional points, static $\mathscr{H}_\infty$ controllers are designed using the analysis of Section 3.3.4. The algorithm is based on the analysis of Section 3.4.2 that merges the loop shaping control theory and the gap metric.

Discussion

The general idea behind the algorithm is the following: the designer computes the open loop corrected transfer function of the system $G_s(s)$ at a nominal point (e.g. $\alpha = \alpha_r, M_0 = 26$), using interpolation for the PID gains and computes the corresponding robustness margin for this point. This is done of course *after* a static robust controller K_∞ is computed using the analysis of Section 3.3.4[8]. Then, for a neighbor operating point (say for $M_0 + \delta M$), the new corrected open loop $G_{s,\Delta}(s)$ is computed and then the gap between these two open loops is calculated. If this gap is smaller than the robustness margin associated with the nominal point, then this means that the robust controller is satisfactory for the neighbor point; if not a new operating point is chosen, a new robust controller and robustness margin computed and the algorithm continues until the flight envelope is covered. The algorithm is formally divided in the following steps:

Step 1 - Initialization. Choose a gridding (e.g. equidistant) over the Mach number range [26,4] and thus obtain a set of candidate synthesis points $\Sigma_M = [M_1, \ldots, M_k]$. Then take as the initial operating point P^j (corresponding to a scheduling vector value ϱ^j) the one corresponding to $M = 26$.

Step 2 - Interpolated Loop Shaping. For the operating point P^j, compute the open loop shaped plant $G_s^j = G_c^j G_\alpha^j G_a$. The plant G_α^j is simply the linearization of the nonlinear parameter-dependent vehicle model at $\varrho = \varrho^j$, whereas the f-PID controller gains are obtained using a triangular interpolation of the five synthesis points of the previous section[9]. For the shaped plant G_s^j, a static $\mathscr{H}_\infty$ controller is calculated using Theorem 3.5 and the corresponding to the point P^j robustness margin ϵ^j is computed.

Step 3 - Line Search or Reset. Performing a line search using subsequent candidate points belonging to Σ_M, successive shaped plants G_s^f are computed, until the gap $\delta_g(G_s^j, G_s^f)$ between the nominal initial plant and the successive one is greater or equal than the robustness margin ϵ^j. If this is the case, then a new operating point P^j is chosen and then the algorithm jumps back to Step 2, except for the case when the end of the flight envelope is reached. In this case, even if $\delta_g(G_s^j, G_s^f) < \epsilon^j$, the final point is selected and the procedure terminated.

[8] Recall that in the full order case the robustness magin is computed before actually computing the controller; however here it is not possible.

[9] For more details on the triangulation process see Section 6.2.2 further on.

Table 6.2: Robust controller gains

Mach	$K_{\mathrm{p},\infty}$	$K_{\mathrm{i},\infty}$	$K_{\mathrm{d},\infty}$	ϵ
26	0.5181	0.5965	0.9424	0.3297
23.5	-0.1472	0.7028	1.0769	0.3374
20.75	-0.1559	0.6507	1.0353	0.3363
18.25	0.4020	0.5894	0.8927	0.3358
16	-0.0899	0.5609	0.8148	0.3162
4	-0.0318	0.6395	0.9429	-

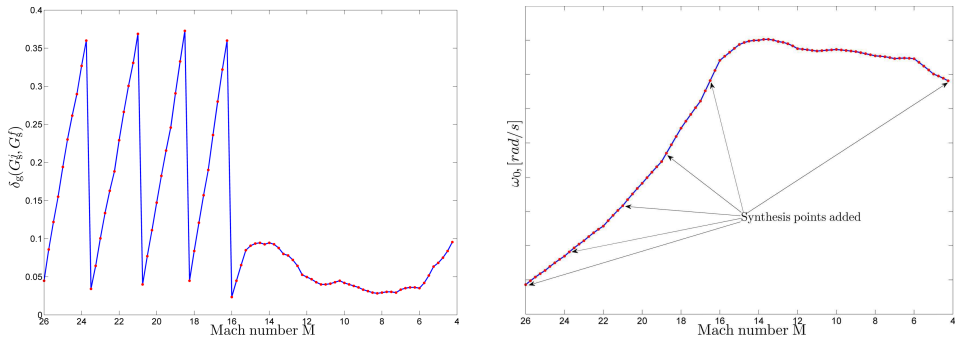
`Algorithm results`

Using this algorithm, totally six additional $\mathscr{H}_\infty$ controllers are computed. The static controller $K_\infty \in \mathbb{R}^{3\times 1}$ treats each of the three channels of the f-PID controller[10]. Each of the gain elements $K_{\mathrm{p},\infty}, K_{\mathrm{i},\infty}, K_{\mathrm{d},\infty}$ (one for each channel) as well as the corresponding Mach numbers are given in Table 6.2.

In Figs. 6.6a-6.6b some results on the operating point choice algorithm are presented. In the first figure, the gap $\delta_{\mathrm{g}}(G_{\mathrm{s}}^{j}, G_{\mathrm{s}}^{f})$ evolution with respect to M is given for a gridding performed each 0.25 units (totally 88 points) whereas in the second, the natural frequency ω_0 evolution with respect to M is shown.

From Fig. 6.6a it may be observed that the gap increases until the first robustness margin $\epsilon^1 = 0.3297$ (see Table 6.2) is surpassed; the algorithm is then re-initialized until all Mach range is covered. It may also be observed that all re-initializations take place (and thus synthesis points added) until approximately $M = 16$; further on, the gap is rather small. The algorithm thus continues until the flight envelope is finished and adds the final point at $M = 4$.

This behavior is explained from Fig. 6.6b showing the linearized plant's natural frequency ω_0 variation, being an indicator of the system's 'nonlinearity'. This frequency increases rapidly until $M = 16$ but then remains almost constant; this is captured by the algorithm which decides that the plant's dynamics do not change significantly to justify another synthesis point until $M = 4$.



(a) *Gap between subsequent shaped plants* **(b)** *Natural frequencies & synthesis points*

Figure 6.6: Operating point choice algorithm results.

[10]For more details on the synthesis scheme refer to the next section.

6.2.1.3 $\mathscr{H}_\infty$ Controller Synthesis

Control goal

In this section, the synthesis procedure concerning the static, robust $\mathscr{H}_\infty$ controllers of the previous section is detailed. Recall from Section 3.3.2 concerning the LSDP that the robust controllers are in fact designed for a shaped open loop plant $G_\mathrm{s}(s)$; additionally nothing changes for the synthesis problem in terms of posing (except of course for the LMI's) be the designed compensator of full or zero order. The final goal is to compute a static controller K_∞ for $G_\mathrm{s}(s)$ in order to guarantee a stable loop and additionally:

$$\left\| \begin{bmatrix} K_\infty \\ \mathbb{I} \end{bmatrix} (\mathbb{I} - G_\mathrm{s} K_\infty)^{-1} \tilde{M}^{-1} \right\|_\infty \leq \gamma, \quad \gamma = \epsilon^{-1} > 0. \tag{6.3}$$

Recall from the previous section that these controllers computed totally at six additional synthesis points (see Table 6.2) yield a robustness margin ϵ^j for the corresponding linearized shaped plants $G_\mathrm{s}(s)$; neighbor plants are also well-behaved under the same corresponding controller due to the gap metric theory.

Synthesis structure

The open loop shaped plant is SISO and thus a robust controller would be a simple gain on the output of the f-PID controller, thus not permitting significant amelioration on the feedback loop. However, if the f-PID controller's control signal is broken in three parts (proportional, integral and derivative) then the robust controller is a three element matrix $(K_{\mathrm{p},\infty}, K_{\mathrm{i},\infty}, K_{\mathrm{d},\infty})$. The synthesis block diagram corresponding to Fig. 6.2 is shown in Fig. 6.7[11].

As a final comment concerning the robust controllers, it is clear that the control structure is really simple (compared to a dynamic robust controller that would be of order five), easy to implement-interpolate and of high performance, as it will be presented in Section 6.2.3.

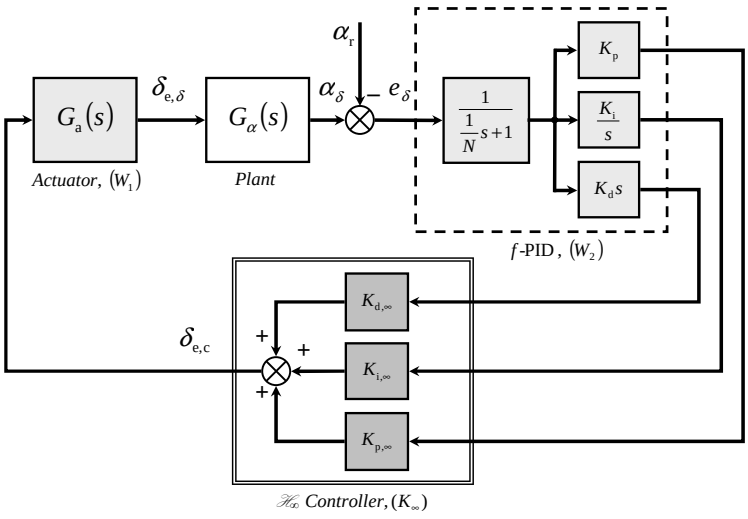


Figure 6.7: Robust controller synthesis block diagram.

[11] For details on solving the synthesis problem refer to Section 3.3.4.

6.2.2 Gain Interpolation

The gain-scheduled controller presented in the next section uses gain interpolation in order to update the LTI controllers' parameters, based on the scheduling vector values. The parameters interpolated are the f-PID gains $K_\mathrm{p}, K_\mathrm{i}, K_\mathrm{d}, N$ and additionally the robust controller gains $K_{\mathrm{p},\infty}, K_{\mathrm{i},\infty}, K_{\mathrm{d},\infty}$.

`f-PID interp/tion`

The first are interpolated using the five initial synthesis points of Table 6.1 and the corresponding four triangular scheduling regions $\Gamma^1, \Gamma^2, \Gamma^3, \Gamma^4$. To create the scheduling regions, Delaunay triangulation is used (refer to Section 1.4) using the coordinates $[\alpha^i \; M^i]^T, i = 1, \ldots, 5$ of all five points. Each gain is interpolated by considering the corresponding plane equation defined by the coordinates of each triangle corner of every scheduling region.

Consider for example the f-PID controller derivative gain K_d, the three corner gains $K_\mathrm{d}^1, K_\mathrm{d}^2, K_\mathrm{d}^5$ of scheduling region Γ^1 (see Fig. 6.1) and the scheduling vector coordinates $\varrho^1 = [\alpha^1 \; M^1]^T, \varrho^2 = [\alpha^2 \; M^2]^T, \varrho^5 = [\alpha^5 \; M^5]^T$. Then the interpolated gain $K_\mathrm{d}(\varrho)$ with $\varrho \in \Gamma^1$ is computed by solving the plane equation leading to the following solution[12]:

$$K_\mathrm{d}(\varrho) = \frac{c_1 - c_2\alpha(t) - c_3 M(t)}{c_4}. \tag{6.4}$$

The constants c_1, c_2, c_3 and c_4 are dependent only to the data concerning the synthesis points and are calculated as:

$$c_1 = \begin{vmatrix} \alpha^1 & M^1 & K_\mathrm{d}^1 \\ \alpha^2 & M^2 & K_\mathrm{d}^2 \\ \alpha^5 & M^5 & K_\mathrm{d}^5 \end{vmatrix}, c_2 = \begin{vmatrix} 1 & M^1 & K_\mathrm{d}^1 \\ 1 & M^2 & K_\mathrm{d}^2 \\ 1 & M^5 & K_\mathrm{d}^5 \end{vmatrix}, c_3 = \begin{vmatrix} \alpha^1 & 1 & K_\mathrm{d}^1 \\ \alpha^2 & 1 & K_\mathrm{d}^2 \\ \alpha^5 & 1 & K_\mathrm{d}^5 \end{vmatrix}, c_4 = \begin{vmatrix} \alpha^1 & M^1 & 1 \\ \alpha^2 & M^2 & 1 \\ \alpha^5 & M^5 & 1 \end{vmatrix}. \tag{6.5}$$

$\mathscr{H}_\infty$ `controller interpolation`

The three $\mathscr{H}_\infty$ controller gains are linearly interpolated as a function of the Mach number for the constant regulation value of the AoA α_r, considering the five intervals formed by the six additional synthesis points added by the gap metric operating point choice algorithm of Section 6.2.1.2. Consider for example the proportional channel gain $K_{\mathrm{d},\infty}$, the middle interval $[M^3, M^4]$ and the corresponding gains $K_{\mathrm{d},\infty}^3, K_{\mathrm{d},\infty}^4$. Then the interpolated value $K_{\mathrm{d},\infty}(M)$ is given by:

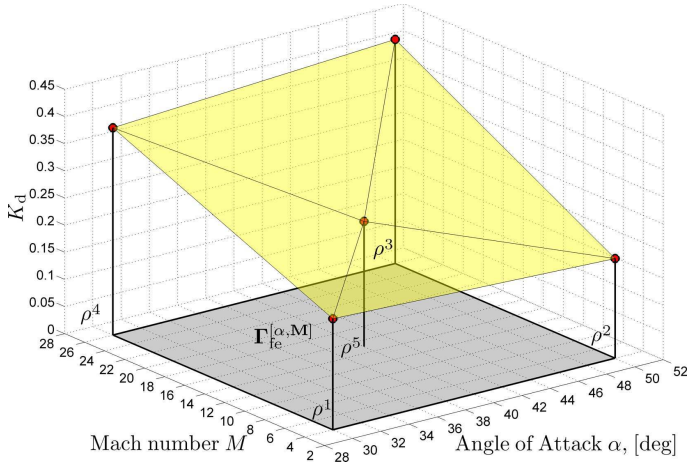
$$K_{\mathrm{d},\infty}(M) = K_{\mathrm{d},\infty}^4 a_M(t) + \big[1 - a_M(t)\big] K_{\mathrm{d},\infty}^3 \tag{6.6}$$

with $0 \leq a_M(t) \leq 1$ being the normalized distance given by:

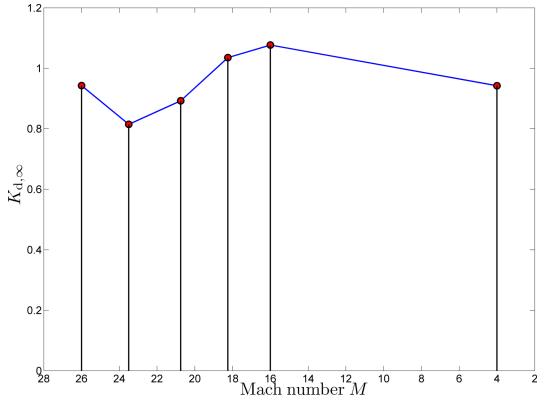
$$a_M(t) = \left| \frac{M(t) - M_3}{M_4 - M_3} \right|. \tag{6.7}$$

In Fig. 6.8a are shown the interpolation surfaces corresponding to the derivative gain K_d for all four triangular scheduling regions whereas in Fig. 6.8b is shown the interpolated derivative robust gain $K_{\mathrm{d},\infty}$ for all five linear interpolation regions for constant AoA.

[12] The interpolated gain may be also seen as the projection of the current scheduling vector coordinates on the plane defined by the three corners of each scheduling region.



(a) *Interpolation surfaces for* K_d



(b) *Interpolation line for* $K_{\mathrm{d},\infty}$

Figure 6.8: Gain interpolation results.

6.2.3 Controller Implementation & Validation

The global gain-scheduled controller will be detailed in this section; in the first section its structure and some other minor issues will be detailed whereas in the following section some simulation results will be presented.

6.2.3.1 Nonlinear Gain-scheduled Controller

The global gain-scheduled controller is implemented by discretizing the f-PID controller using bilinear transformation[13] with a sampling time $T_\mathrm{s} < 0.15s$ and an ideal sampler. All seven gains are then interpolated using the procedure described in the previous section.

Global controller

The total control signal $\delta_{\mathrm{e,tot}}$ supplied to the actuator is the sum of the trim control signal $\delta_{\mathrm{e,r}} = \delta_\mathrm{e}(\varrho_\mathrm{r})$ (see Eq. 4.57 in the trim analysis Section 4.2.2) and the closed loop scheduled stabilizing signal $\delta_{\mathrm{e,c}}$. The transfer function $\mathcal{K}(s, \varrho)$ providing the gain-scheduled control signal before discretizing is:

$$K(s,\varrho) = \frac{1}{\frac{1}{N(\varrho)}s + 1}\left[K_\mathrm{p}(\varrho)K_{\mathrm{p},\infty}(\varrho) + \frac{K_\mathrm{i}(\varrho)K_{\mathrm{i},\infty}(\varrho)}{s} + K_\mathrm{d}(\varrho)K_{\mathrm{d},\infty}(\varrho)s\right] \quad (6.8)$$

Controller block diagram

The following figure shows the Simulink diagram of the gain-scheduled controller. The big grey block represents the ARV nonlinear dynamics, the small one generates the Mach reference time trajectory illustrated in Fig. 4.12a and the red block generates the AoA reference value. The upper small blue block generates the trim control signal whereas the other two big blue blocks represent the discretized controller of Eq. 6.8 and the interpolating functions used to update its gains as a function of the scheduling vector. Finally the yellow block represents the actuator dynamics.

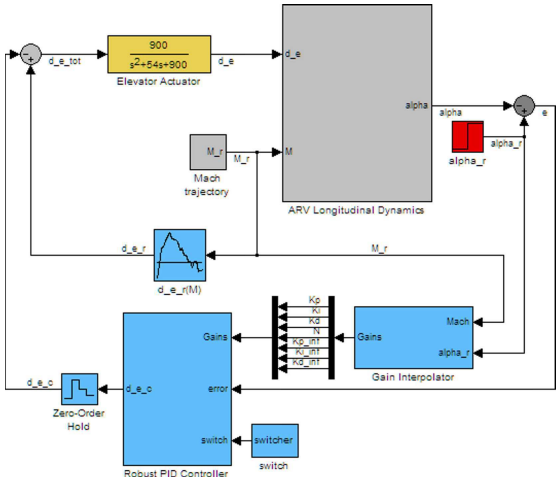


Figure 6.9: Robust controller synthesis block diagram.

[13] Recall that using this transformation the Laplace variable is replaced with $s = \frac{2}{T_\mathrm{s}}\frac{z-1}{z+1}$.

6.2.3.2 Simulation Results

In this section some simulation results will be presented both for the nominal and the uncertain case corresponding to the control objectives of Section 6.2, demonstrating the effectiveness of the proposed control scheme.

Nominal case results

For the *nominal case*, the simulation profiles used for the Mach and the dynamic pressure are already presented in Figs. 4.12a-4.12b. The goal for the autopilot is to regulate the AoA around the reference value with $\pm 1\%$ steady-state accuracy. The time simulation of the gain-scheduled controller is shown in Fig. 6.10; the blue curve shows the response if only the f-PID controller is used whereas the red one if both the f-PID controller and the $\mathscr{H}_\infty$ controllers are used. The steady state margins are satisfied for both cases with slight differences in the amplitude; however the stability margin performance is not good if the robust controllers are not used, as it is demonstrated further on.

To test these stability margins, the gain-scheduled system is linearized every 10s and thus totally 57 frozen time open loop systems are obtained[14] when using either the f-PID controller or both the f-PID and $\mathscr{H}_\infty$ controllers. In Fig. 6.11a the Nichols charts for these two cases are shown whereas in Figure 6.11b the corresponding gain & phase margins (GM & PM) are plotted for each system. Using these figures it may be seen that the GM lower limit of 6dB is never violated and the robust controllers provide an amelioration of up to 2dB's. The results are even better for the PM since the f-PID controller by itself is not sufficient with the lower limit of 30° being violated for $350 \lesssim t \lesssim 435$s and reaching its worst point of 24° at $t \simeq 380s$. The robust controller, using the additional synthesis points, succeeds at augmenting the PM up to 7.5° and thus helps the gain-scheduled controller meeting the robustness constraints[15].

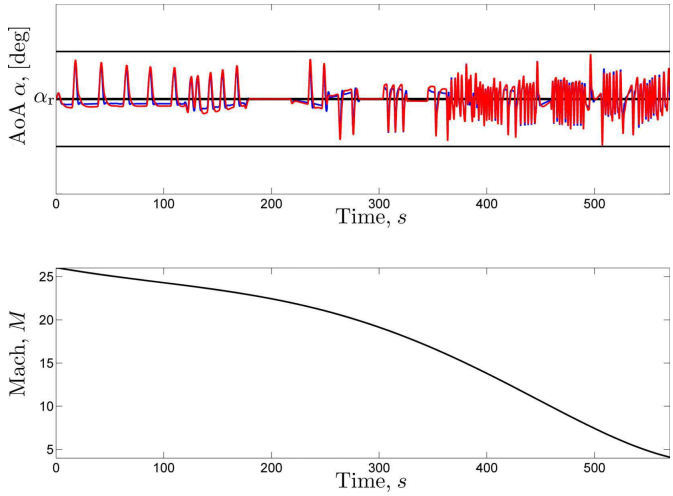
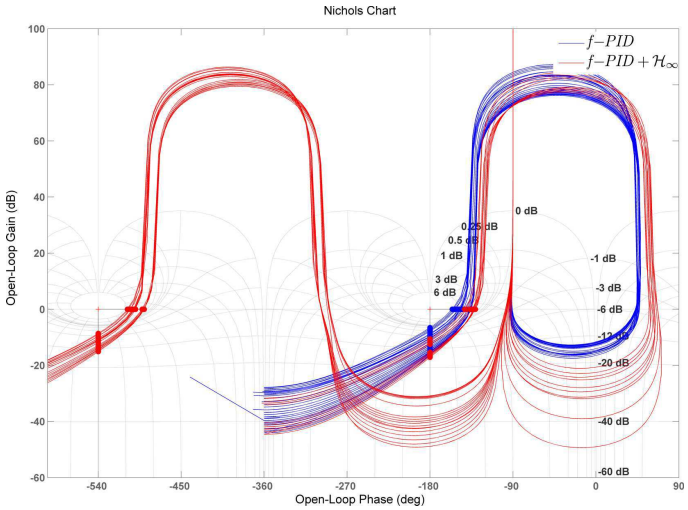
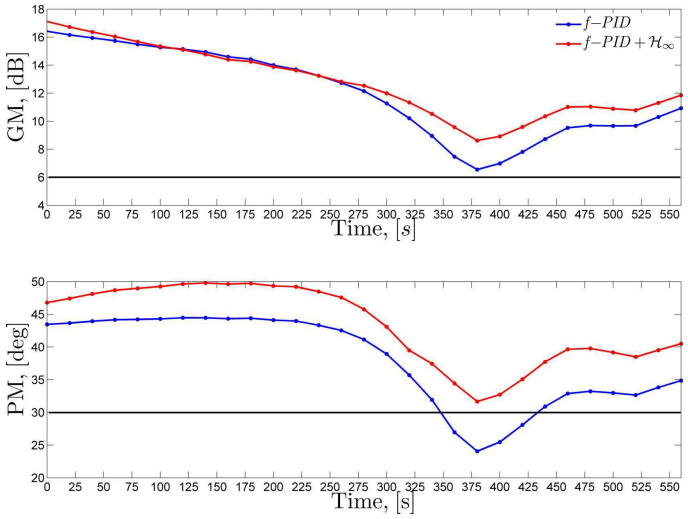


Figure 6.10: Gain-scheduled controller time performance.

[14] This is done using the MATLAB® Simulink Control Design toolbox.
[15] The biggest augmentation is observed in fact for the worst point ($t \simeq 380s$).



(a) *Nichols charts*



(b) *Gain & Phase Margins*

Figure 6.11: Simulation results (nominal).

Uncertain case results

Concerning now the *uncertain* case, the gain-scheduled autopilot is also tested in the face of uncertainties over the dynamic pressure Q, the moment of inertia $\mathrm{I_{yy}}$ and the aerodynamic functions C_{m0} and C_{me}. Two worst case scenarios are considered in total; $\pm 35\%$ and $\mp 50\%$ additive uncertainties on C_{m0} and C_{me}, $+35\%$ on Q and $+10\%$ on $\mathrm{I_{yy}}$.

The delay margin (DM) is tested for both scenarios by considering the *nominal* case plus the *robust* cases (with totally 9+9 uncertain runs obtained by increasing the uncertainty over the four variables by 10% each time until reaching the maximum uncertainty limit) and the results are shown in Figures 6.12a, 6.12b for all 57 frozen time models. Moreover, the Nichols charts (see Figures 6.13a, 6.13b) show the stability margins of these linearized open loops for the worst cases (maximum uncertainty norm) of both uncertain scenarios.

The minimum gain, phase and delay margin (in sampling periods) for the nominal case and both uncertain scenarios[16] are found in Table 6.3. It can be observed that the additional points added with the synthesis point selection algorithm of Section 6.2.1.2 have clearly assisted the gain-scheduled controller meeting the specifications imposed in Section 6.2 with only small violations on the delay margins. Obviously, no uncertain cases are considered for the simple PID tuning since not even the nominal ones are satisfied; this in fact shows the necessity of the $\mathscr{H}_\infty$ controllers.

Table 6.3: Stability margin results[(i)]

Study case	GM	PM	DM
PID (nominal)	6.5(6.0)	24.1 (30)	0.70 (1.0)
PID+$\mathscr{H}_\infty$ (nominal)	8.7(6.0)	32.2 (30)	1.00 (1.0)
PID+$\mathscr{H}_\infty$ (uncertain case No.1)	6.4(3.0)	17.3 (na)	0.44 (0.5)
PID+$\mathscr{H}_\infty$ (uncertain case No.2)	3.4(3.0)	15.9 (na)	0.40 (0.5)

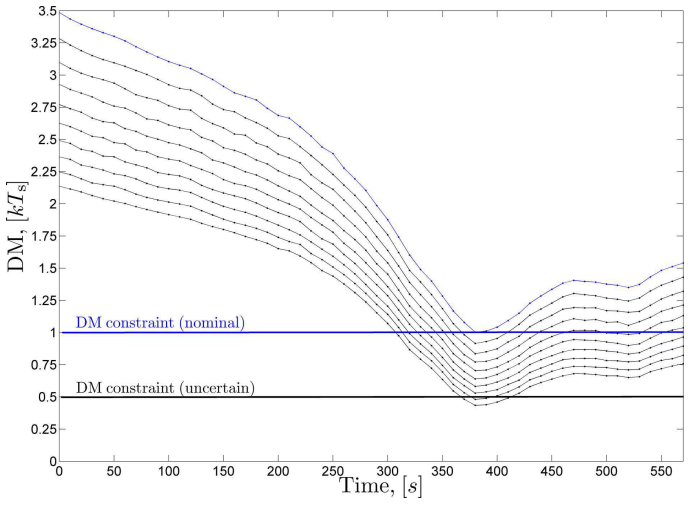
[(i)] The constraint values are given in parentheses.

6.2.3.3 Discussion

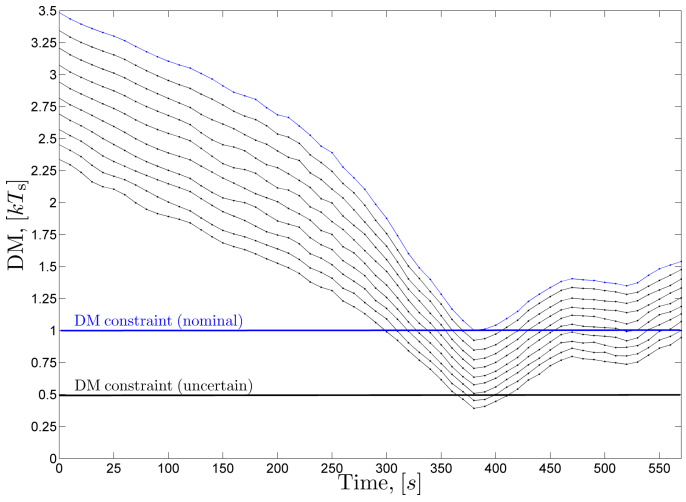
The autopilot designed in this section is used for the regulation of the ARV AoA around a reference value α_{r} during the atmosphere re-entry phase, when the Mach number is time-varying and this reference value must be held constant during the flight phase considered.

However, the procedure used here may be applied for any other reference AoA; this is easily done by re-running the e-LSDP operating point selection algorithm for this new value and re-storing the $\mathscr{H}_\infty$ controller gains. This fact really proves the generality of the approach followed here since the operating point choice algorithm is designed to fine-tune the f-PID controllers for *any* value of $30 \leq \alpha_{\mathrm{r}} \leq 50$ of the ARV's flight envelope.

[16] In the uncertain cases only the combined loop shaping plus robust controller based gain-scheduled controller is considered.

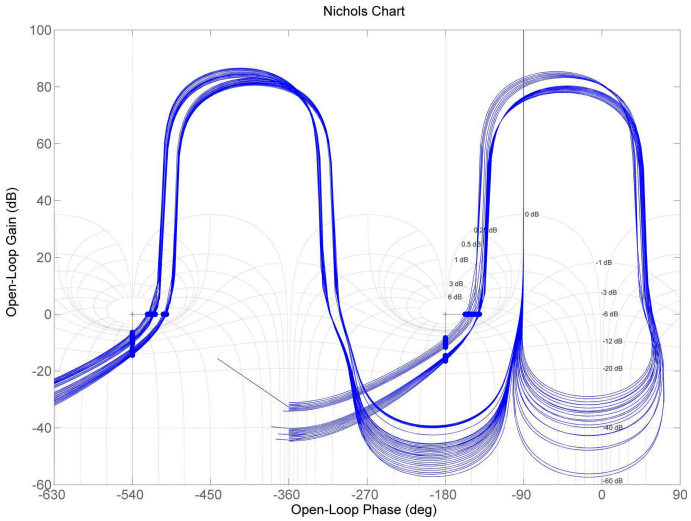


(a) *Uncertain case No.1*

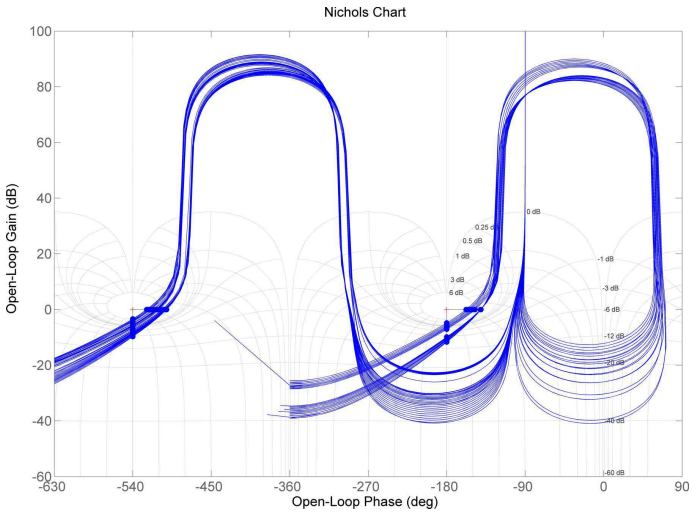


(b) *Uncertain case No.2*

Figure 6.12: Delay margins (nominal & uncertain).



(a) *Uncertain case No.1*



(b) *Uncertain case No.2*

Figure 6.13: Nichols charts (worst cases).

6.3 Gain Blending (missile)

Reminder & motivation

The Reichert benchmark missile autopilot problem has been analyzed in Chapter 5 and two scheduling methods have been applied to obtain a gain-scheduled autopilot. These methods (the controller blending and the state feedback/observer-based interpolation) are of common use in the gain scheduling practice; however they result to a conservative and complicated controller due to the fact that they give no indication on the number of synthesis points needed, due to the high order of the LTI controller and also due to practical issues concerning the methods (e.g. initialization, eigenvalue partitioning etc.). A new method is proposed here based on *gain blending* interpolation and on an extended loop shaping procedure (the e-LSDP) permitting to conceive an interpolation strategy that addresses the aforementioned issues.

In this second part of the chapter this procedure is detailed, following the analysis of the first part concerning the ARV autopilot design. The e-LSDP is adjusted to take into account that the additional points needed must now be added on a plane and not only across a line as with the ARV since the scheduling vector ϱ may follow any trajectory on the missile flight envelope.

As it has been already detailed in Section 5.3, the missile control objectives are performance ones (adequate time constant, overshoot, steady-state error & control rate) as well as robustness ones (robust stability in the face of parametric uncertainties & high frequency open loop magnitude attenuation). The difference between the missile autopilot problem and the ARV one is mainly that here the problem is a tracking and not a regulation one and that stability margin constraints do not appear explicitly; even though it is desired that they are maximized as for any feedback control system.

6.3.1 LTI Controller Synthesis

The e-LSDP (re-visited)

Following the discussion for the ARV autopilot of Section 6.2.1, the LTI controllers for the Reichert benchmark problem are calculated using the e-LSDP procedure that is divided in the three standard steps: *loop shaping*, *operating point algorithm* and finally $\mathscr{H}_\infty$ *controller synthesis*. Briefly these steps involve the following analysis:

Step 1 - Loop Shaping. A linearized model of the missile $G(s)$ is obtained for a small number of synthesis points (9) on the flight envelope, the same as in Chapter 5 (see Table 5.1). The missile linearized dynamics are preceded by the actuator dynamics $G_\mathrm{a}(s)$ (acting as a pre-compensator $W_1(s)$) and followed by a specific *outer-inner loop* PI/P controller (acting as a post-compensator $W_2(s)$), in order to shape the open loop frequency response. This procedure corresponds to the LSDP of Section 3.3.2 and provides some basic compensation for these synthesis points; additional synthesis points are added using the algorithm that follows.

Step 2 - Operating Point Algorithm. Similarly to the ARV autopilot problem considered in the first part of this chapter, if the loop shaping controllers only are used to obtain an interpolated gain-scheduled controller, the results are not satisfactory. Once again, the *gap metric* coupled with $\mathscr{H}_\infty$ *loop shaping* theories are used to devise an operating point choice algorithm that will capture the nonlinear dynamics variation; a glimpse of this variation may be observed from the linearized dynamics results presented in Fig. 4.7. As it has been already mentioned, the algorithm chooses points for a family of values for the scheduling vector ϱ, inside all the flight envelope and not only across a line as with the ARV problem ; for additional details on the algorithm see Section 6.3.1.2.

Step 3 - $\mathscr{H}_\infty$ Controller Synthesis. The robust controller synthesis algorithm follows closely the theory presented in the first part and thus the static $\mathscr{H}_\infty$ controller synthesis of Section 3.3. The static controllers are once again designed at the synthesis points deducted from the previous step and then interpolated to provide an additional corrective action over the loop shaping PI/P controllers; for more details see Section 6.3.2.

6.3.1.1 Loop Shaping

As it has been detailed in the previous section, the first step of the e-LSDP is the initial loop shaping performed over a small number of synthesis points, using corresponding transfer functions $G(s) = [G_\eta(s)\ G_q(s)]^T$ issued from the initial missile nonlinear parameter-dependent model $\mathcal{S}_{\text{pd}}$[17].

The control structure chosen is a special type of external/internal (PI/P type) compensation; this strategy has been chosen among others due mainly to its simplicity and ease of tuning as it will be shown in the following analysis. It is evident that in terms of performance and robustness it may be inferior than the full-order $\mathscr{H}_\infty$ controllers considered in the previous chapter; however in terms of implementation and aided by the additional static $\mathscr{H}_\infty$ controllers designed in the next sections, it results to a better gain-scheduled controller.

`Control structure`

The control structure used is depicted in Fig. 6.14; an inner simple proportional feedback (P controller) is applied first on the pitch rate q_δ with positive feedback[18] in order to reduce its corresponding open loop gain and augment its gain margin. Then, an external proportional plus integral feedback (PI controller) is added to the tracking error $e_\delta = \eta_\delta - \eta_{\text{r}}$ in order to achieve good tracking performance[19]. The three gains $K_{\text{p}}, K_{\text{i}}, K_q$ are first adjusted in a two step procedure considering first the inner and then the outer loop, using standard frequency domain techniques, and then are optimized using MATLAB® Simulink Control Design and Simulink Response Optimization routines.

[17] For details on the missile trim analysis and linearization refer to Sections 4.1.2-4.1.3.

[18] Positive feedback is used since the pitch rate open loop gain is negative (see Eq. 6.10).

[19] The feedback sign convention for the tracking error is conformable to a standard robust control notation maintaining *positive feedback*.

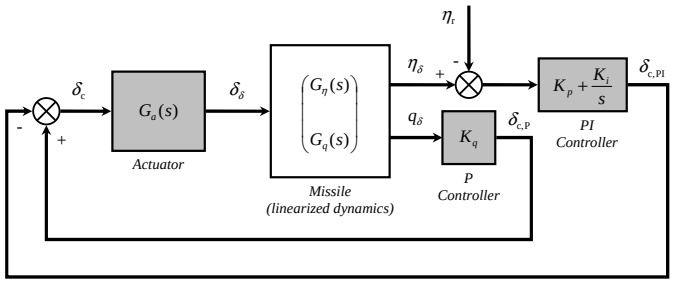


Figure 6.14: PI/P compensation block diagram.

Tuning example

To illustrate the PI/P controller tuning, consider the missile linearized state-space model $\mathcal{S}(\varrho_r)$ for the fifth synthesis point ($\varrho_r = [\eta_r \; M_r]^T = [10.7132 \; 2.25]^T$) corresponding to the middle of the missile flight envelope:

$$\mathcal{S}_{\text{LPV}}(\varrho_r) \overset{\text{ss}}{:} \begin{cases} \begin{pmatrix} \dot{\alpha}_\delta \\ \dot{q}_\delta \end{pmatrix} = \begin{pmatrix} -0.9945 & 1 \\ -151.03 & 0 \end{pmatrix} \begin{pmatrix} \alpha_\delta \\ q_\delta \end{pmatrix} + \begin{pmatrix} -0.0888 \\ -73.611 \end{pmatrix} \delta_\delta \\ \begin{pmatrix} \eta_\delta \\ q_\delta \end{pmatrix} = \begin{pmatrix} -75.873 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_\delta \\ q_\delta \end{pmatrix} + \begin{pmatrix} -6.5705 \\ 0 \end{pmatrix} \delta_\delta \end{cases} \tag{6.9}$$

or the corresponding matrix transfer function $G(s)$, with:

$$G = \begin{bmatrix} G_\eta \\ G_q \end{bmatrix} = \frac{\begin{bmatrix} -6.57s^2 + 0.1996s + 4593 \\ -73.61s - 59.8 \end{bmatrix}}{s^2 + 0.9945s + 151}. \tag{6.10}$$

The aforementioned linearized system presents two badly damped but stable poles $p_{1,2} = -0.4972 \pm 12.279$ whereas the transmission zeros are $z_{1,2} = \pm 26.453$ and $z_3 = -0.812$ for the vertical acceleration and pitch rate channels respectively.

P controller tuning

The tuning of the pitch rate K_q is done by considering the inner loop that may be seen as the positive *feedback* interconnection of the actuator transfer function $G_a(s)$ with the *series* interconnection of the pitch rate transfer function $G_q(s)$ and the P controller. The input to this loop is the output of the PI controller $\delta_{c,PI}$ with negative sign whereas the output is the filtered total control signal δ_δ; the block diagram of this loop is shown in Figure 6.15.

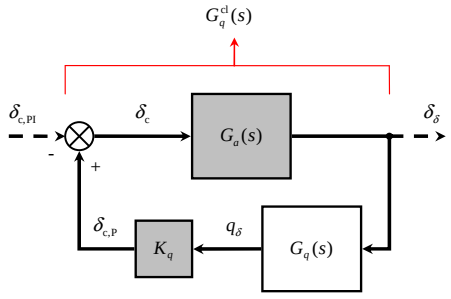


Figure 6.15: Inner loop block diagram.

The closed loop transfer function $G_q^{\rm cl}(s)$ formed by this interconnection is:

$$G_q^{\rm cl}(s) = \frac{G_a(s)}{1 - K_q G_q(s) G_a(s)}. \tag{6.11}$$

The pitch rate feedback gain K_q permits to play on the magnitude of the corresponding open loop transfer function $G_q^{\rm ol}(s)$ given by:

$$G_q^{\rm ol}(s) = K_q G_q(s) G_a(s). \tag{6.12}$$

This gain is computed by the Evans root locus method; the open loop poles of $G_q^{\rm ol}(s)$ are comprised by the badly damped missile ones plus the actuator's; given that the latter are very fast, only the former are considered for the tuning. The gain is chosen so that these poles obtain a good damping corresponding to the 10% overshoot constraint P_1 of the missile control objectives; this gives a damping of 0.59. The root locus diagram for synthesis point No. 5 is shown in Fig. 6.17a and the gain computed is $K_q = 0.183$; resulting to a loop gain decrease of $20\log_{10}(K_q) = -14.75$dB (see also the Bode diagram of Fig. 6.17b).

PI controller tuning

Once the pitch rate loop is tuned, the vertical acceleration loop is corrected by adjusting the gains $K_{\rm p}, K_{\rm i}$ of the PI controller. The open loop $G_\eta^{\rm ol}(s)$ now is formed by the series interconnection of the PI controller transfer function $G_{\rm PI}(s)$ (with a negative sign, corresponding to the negative feedback of Fig. 6.14), the adjusted closed loop pitch rate transfer function $G_q^{\rm cl}(s)$ and the missile vertical acceleration transfer function $G_\eta(s)$:

$$G_\eta^{\rm ol}(s) = -G_\eta(s) G_q^{\rm cl}(s) G_{\rm PI}(s). \tag{6.13}$$

The closed loop transfer function $G_\eta^{\rm cl}(s)$ is obtained by the unitary positive feedback interconnection of the open loop transfer function $G_\eta^{\rm ol}(s)$[20]:

$$G_\eta^{\rm cl}(s) = \frac{G_\eta(s) G_q^{\rm cl}(s) G_{\rm PI}(s)}{1 + G_\eta(s) G_q^{\rm cl}(s) G_{\rm PI}(s)}. \tag{6.14}$$

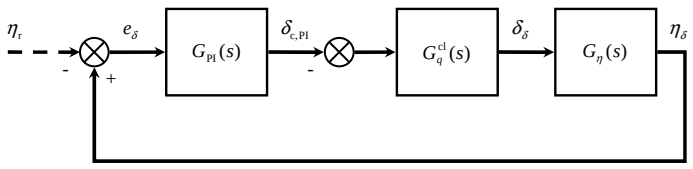
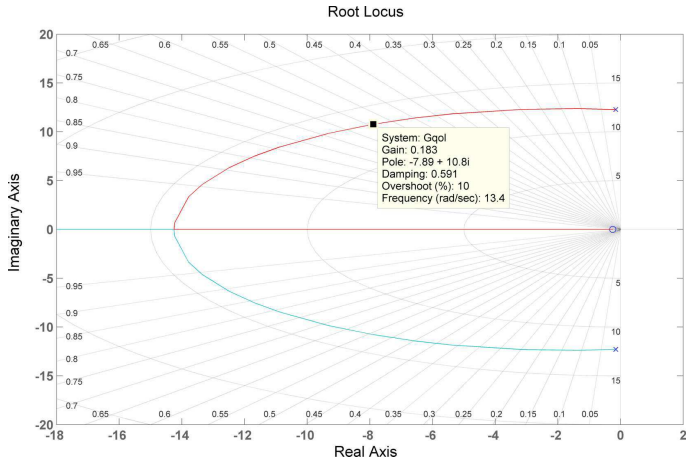


Figure 6.16: Outer loop block diagram.

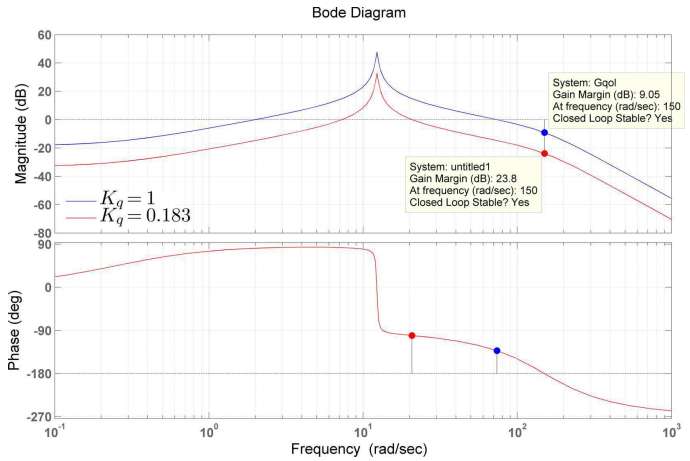
The PI controller transfer function is given by the following equivalent formulations:

$$G_{\rm PI}(s) = K_{\rm p} + \frac{K_{\rm i}}{s} = K_{\rm i}\frac{1 + \frac{K_{\rm p}}{K_{\rm i}} s}{s}. \tag{6.15}$$

[20]Note that the reference signal $\eta_{\rm r}$ is applied using a negative sign.



(a) *Evans root locus (pitch rate loop)*



(b) *Bode diagram (pitch rate loop)*

Figure 6.17: Pitch rate pre-tuning.

Pre-tuning

The PI controller is tuned in two phases: first only the *integral action* is added and the integral gain K_i is adjusted so as obtain a satisfying step response performance[21]. By augmenting the gain, the response becomes more rapid (the time constant τ is reduced) but also more oscillatory. The integral gain is chosen trying to minimize the time constant while respecting the maximum overshoot constraint. The resulting gain is $K_\mathrm{i} = 0.15314$, giving a time constant $\tau = 254.6ms$ (less than 350ms imposed by the performance objectives) and a settling time $t_\mathrm{s} = 552ms$[22] with the overshoot being $M_\mathrm{p} = 10\%$ (equal to the constraints). The results using only the integral action are good but not all the control bandwidth is used. Indeed the maximum control rate $\dot{\delta}_\delta$ is approximately 9deg/s, about three times less than the limit of 25deg/s of the performance objective P_2 of Section 5.3. The *proportional action* now of the PI controller will add a zero on the open loop transfer function $G_\eta^{\mathrm{ol}}(s)$ of Eq. 6.13 permitting a more rapid step response.

Final tuning

Based on the pre-tuning of the integral gain, both the *integral* and *proportional* gains are optimized using MATLAB® Simulink Control Design and Simulink Response Optimization; the strategy used is to try and minimize the vertical acceleration step response time constant τ while not violating the overshoot and control rate constraints. The gains obtained are $K_\mathrm{i} = 0.18593$ (re-tuned) and $K_\mathrm{p} = 0.0052173$[23] and the two step responses (integral and re-tuned integral plus proportional) are shown in Fig. 6.18a whereas the two corresponding corrected open loop Bode diagrams are visualized in Fig. 6.18b.

The faster step response with PI controller (red line) with respect to the I controller (blue line) is evident (see Fig. 6.18a); this may in fact be explained by the increased open loop bandwidth. In the second case the gain crossover frequency ω_gc is 4.87rad/s whereas in the first case 6.19rad/s (27% bigger) (see Fig. 6.18b). The time constant τ is in the second case 254.6ms (as mentioned before) whereas in the first case 203ms (25.5% faster). Despite the system being significantly faster, the gain margin is also slightly ameliorated (8.71dB compared to 7.84dB initially) whereas the phase margin is similar since the damping in both cases is the same.

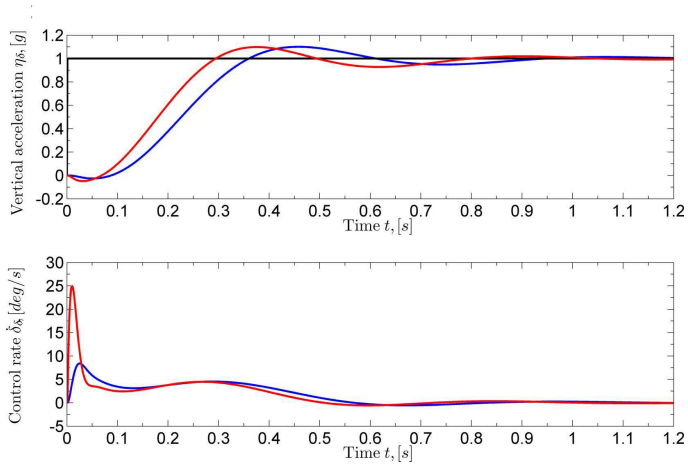
The closed loop dynamics are of fifth order; the poles (system, controller, actuator) and zeros (total outer feedback loop) are the following (the non-minimum phase step response of Fig. 6.18a may now be justified by the positive closed loop I/O zero):

$$\begin{aligned} \text{poles}: &\quad -3.84 \pm 11.6j, -6.8, -97.9 \pm 96.5j \\ \text{zeros}: &\quad \pm 26.4, -35.6. \end{aligned}$$

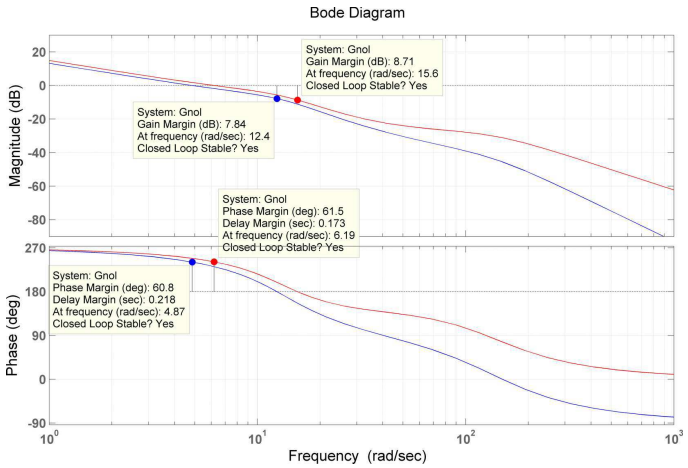
[21] This adjustment is done using MATLAB® SISOTool, permitting to observe in real-time the influence of the integral gain on the step response.

[22] Note that no settling time constraints are imposed; trying to minimize the time constant does not always mean that the settling time is also minimized. In fact this results to a greater overshoot and thus the settling time is finally augmented.

[23] This corresponds to a controller zero added at $s = -K_\mathrm{i}/K_\mathrm{p} \simeq -35.64$.



(a) *Step responses*



(b) *Open loop Bode diagrams*

Figure 6.18: PI controller tuning comparison results (blue: integral action, red: integral plus proportional actions).

Table 6.4: PI/P controller tuning results

Points	K_q	K_p	$K_\mathrm{i}^{(\mathrm{i})}$	
1	0.214	0.0056002	0.11111	0.12988
2	0.307	0.0042098	0.28452	0.30545
3	0.351	0.0031131	0.37043	0.38815
4	0.144	0.0057412	0.07762	0.10476
5	0.183	0.0052173	0.15314	0.18593
6	0.213	0.0046419	0.21069	0.24660
7	0.106	0.0057885	0.05839	0.08913
8	0.117	0.0056409	0.08030	0.11584
9	0.141	0.0053212	0.12486	0.16721

(i) The first column gives the values for the pre-tuning whereas the second the final values after the optimization.

The same procedure is applied iteratively for all synthesis points (see Section 5.4) and the results for the gains are shown in Table 6.4. The pitch rate and integral channel gains increase with the vertical acceleration for a given Mach number whereas the proportional channel gain decreases.

Tuning results & discussion

The results from the PI/P controller shaping are shown in Table 6.5. The second line shows the time constant achieved; the missile's performance ameliorates with increasing Mach and vertical acceleration. The synthesis is done trying not to violate $M_\mathrm{p} = 10\%$ and $|\dot{\delta}_\delta|_\mathrm{max} = 25\mathrm{deg/s}$; the rapidity constraint is not achieved only at the first synthesis point where $\tau = 440\mathrm{ms}(> 350\mathrm{ms})$. Comparing with the $\mathscr{H}_\infty$ controllers of Section 5.4.1 the results are really good, taking also into account the controller order considered in both cases.

Table 6.5: PI/P controller tuning results(i)

Points	1	2	3	4	5	6	7	8	9
τ(ms)	440	321	286	240	203	181	146	126	141
GM(dB)	9.62	8.15	7.78	10.5	8.71	8.00	10.3	9.44	8.25
PM(deg)	61.1	61.2	61.2	60.8	61.5	61.2	58.8	60.5	61.7
DM(ms)	370	276	248	196	173	155	109	112	106
ω_gc(rad/s)	2.89	3.87	4.31	5.44	6.19	6.88	9.39	9.39	10.1
GM(dB)	26.3	25.0	24.6	21.2	20.4	20.0	17.2	17.0	16.4
PM(deg)	78.5	79.9	79.9	67.2	69.6	70.0	52.0	54.3	55.8
DM(ms)	144	103	90.1	81.7	66.7	57.7	47.0	45.0	38.5
ω_gc(rad/s)	9.51	13.5	15.5	14.3	18.2	21.2	19.3	21.1	25.3
d_att(dB)	43.5	41.9	41.2	38.2	37.7	37.3	34.1	34.0	33.8

(i) Lines $2 \rightarrow 5$ give the frequency results with the outer loop opened, whereas lines $6 \rightarrow 10$ when the actuator loop is opened.

All nine step responses of the missile vertical acceleration and control rate are visualized in Figs. 6.19a-6.19b; the red lines correspond to the faster retuned PI controller and the blue ones to the I controller only. The uniformity of the curves is apparent with the system becoming faster through points $1 \rightarrow 9$; in addition, each response demonstrates the same overshoot and control rate (for the PI case only of course).

In Figs. 6.20a-6.20b the closed loop poles (missile+controller+actuator) and η-channel transmission zeros of each of the nine corrected systems using the final PI/P controller are presented; the first figure shows the big picture whereas the second zooms on the missile's poles only. The rapidity of the system's dominant poles (namely the missile ones) clearly increases through the synthesis points $1 \rightarrow 9$ (see Fig. 6.20b). These poles exhibit a rather constant damping (between 0.283 and 0.333) and an increasing natural frequency (between 5.59rad/s and 19.2rad/s).

The open loop frequency results with the loop opened *before the PI controller* (see Fig. 6.16) are given in lines $3 \rightarrow 6$ of Table 6.5 and the corresponding Bode and Nichols diagrams in Figs. 6.21a-6.21b. The results include the gain, phase and delay margins (GM, PM, DM) and gain crossover frequencies for all open loop transfer functions $G_{\eta}^{\mathrm{ol}}(s)$ of Eq. 6.13. Again the PM is almost constant (around 60°) since all time responses exhibit the same overshoot whereas the GM seems adequate ranging from 8 to 10.5dB. The DM follows the same pattern as the time constant one and corresponds from 75% to 87% of the time constant.

The open loop frequency results with the loop now opened *before the actuator* (see Fig. 6.14) are also given in lines $7 \rightarrow 11$ of Table 6.5 and the corresponding Bode and Nichols diagrams in Figs. 6.21a-6.21b. Suppose the total PI/P controller matrix transfer function from the missile outputs $\eta_{\delta}, q_{\delta}$ to the control input δ_{c} is denoted by:

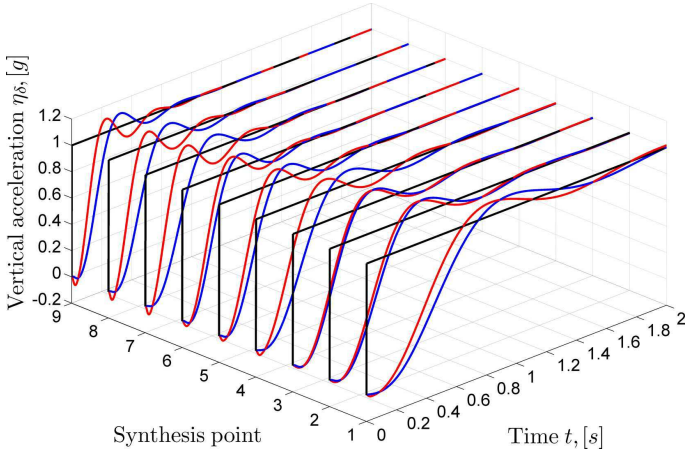
$$G_{\mathrm{c}}(s) = \begin{bmatrix} -G_{\mathrm{PI}}(s) & K_q \end{bmatrix} = \begin{bmatrix} -K_{\mathrm{p}} - \dfrac{K_{\mathrm{i}}}{s} & K_q \end{bmatrix}. \tag{6.16}$$

Then the aforementioned open loop (stabilized always with a positive feedback) is a SISO transfer function G_a^{ol} comprised by the series interconnection of the actuator, the missile linearized dynamics and the controller[24]:

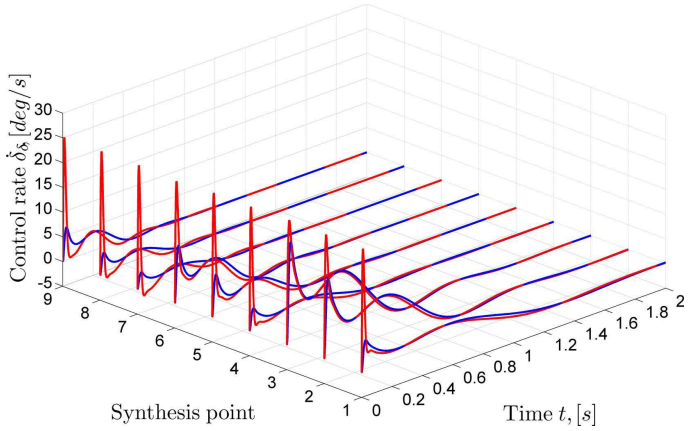
$$G_a^{\mathrm{ol}} = G_{\mathrm{c}}(s)G(s)G_a(s). \tag{6.17}$$

Besides the GM, PM, DM and gain crossover frequency, the open loop magnitude attenuation d_{att} (robustness constraint R_2 of Section 5.3) is given; the latter is maintained for all synthesis points ($d_{\mathrm{att}} > 30db$). The GM and DM decreases for synthesis points $1 \rightarrow 9$ since the system becomes more rapid with the latter being approximately one third of the system's time constant; finally, the PM is also very good.

[24]Note that a *positive feedback* is always assumed for the calculation of the closed loop transfer function.

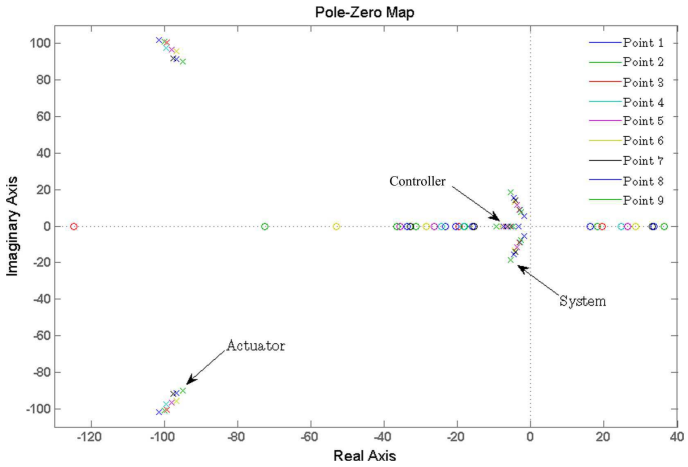


(a) *Step responses*

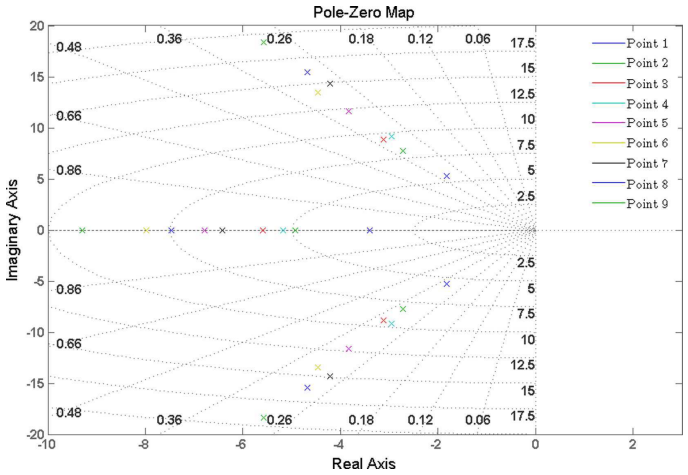


(b) *Open loop Bode diagrams*

Figure 6.19: Total controller tuning results (blue: integral action, red: integral plus proportional actions).

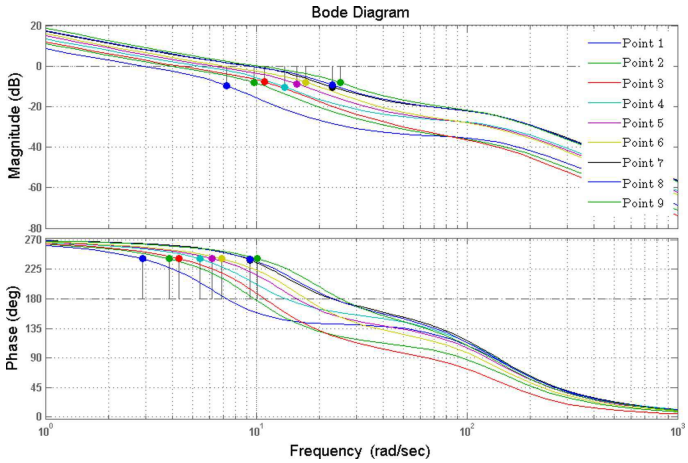


(a) *General view*

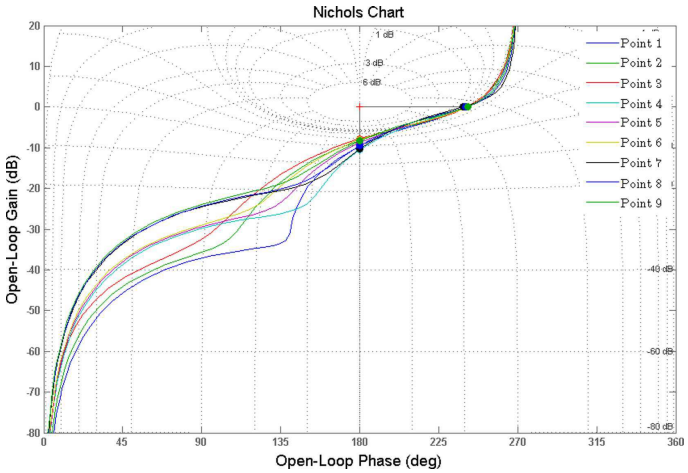


(b) *Zoomed view*

Figure 6.20: Closed loops pole-zero map.

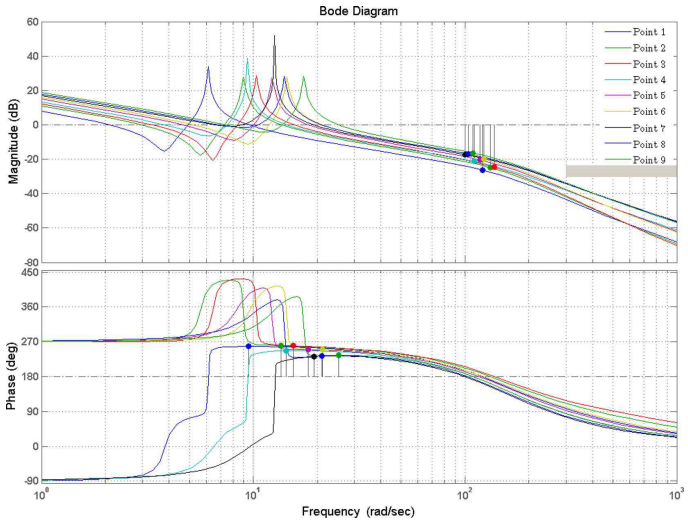


(a) *Bode diagrams*

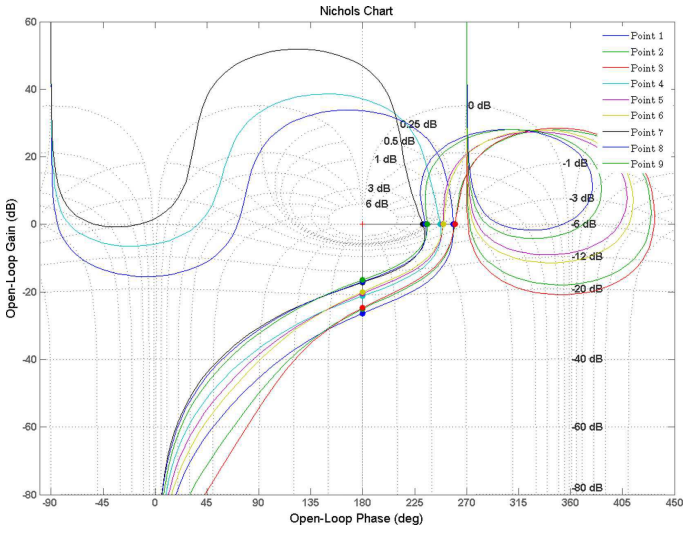


(b) *Nichols diagrams*

Figure 6.21: Open loops analysis (outer feedback loop).



(a) *Bode diagrams*



(b) *Nichols diagrams*

Figure 6.22: Open loops analysis (actuator loop).

6.3.1.2 Operating Point Algorithm

Based on the loop shaping performed in the previous section, an additional set of synthesis points for the $\mathscr{H}_\infty$ controllers will be computed, similarly to the analysis of Section 6.2.1.2 concerning the ARV benchmark model. The major difference between the two is that the missile's nonlinear dynamics will be treated in *two dimensions* and not only across a line as with the ARV.

Discussion

The algorithm proposed in this section is essentially the same as the one in Section 6.2.1.2, with the difference that here the linear search is performed for m *sets* of equidistant values $\Sigma_\eta^m = [\eta_1, \ldots, \eta_k^m]$ for the vertical acceleration. The index 'm' defines the value M^m of the Mach number corresponding to each set, since the algorithm is performed iteratively for a gridding $\Sigma_M = [M^1, \ldots, M^m]$ over the Mach number. Given the fact that for each value of Σ_M, the corresponding final value η_k^m is different because of the trapezoidal form of the flight envelope, the size of each set Σ_η^m will be different for each m. The algorithm used here proposes only three values for Σ_M; the same used for the loop shaping of the previous section (i.e. 1.5,2.25,3) and thus the additional synthesis points will be added across these three (constant Mach) lines[25]. Once again, the operating point algorithm is divided into three distinct steps:

Step 1 - Initialization. Choose an equidistant gridding (e.g. $\Sigma_M = [1.5, 2.25, 3]$) over the Mach range and then a second equidistant gridding Σ_η^m for all $m = 1, \ldots 3$ over the vertical acceleration, thus creating a planar gridding of candidate synthesis points. Take then as the initial synthesis point P^j the one corresponding to $\eta = 0$, for each value of Σ_M[26].

Step 2 - Interpolated Loop Shaping. For the initial operating point P^j (for a corresponding scheduling vector value $\varrho = \varrho^j$ with $m = 1$), compute the open loop shaped plant $G_{\mathrm{s}}^j = G_{\mathrm{c}}^j G^j G_{\mathrm{a}}$ (see Eq. 6.17). The loop shaping PI/P controller gains at this point are computed using linear interpolation of the nine initial synthesis points and the corresponding four trapezoidal scheduling regions (see Fig. 5.2). The missile model is simply the linearization of the initial parameter-dependent model for $\varrho = \varrho^j$, and the open loop shaped plant is finally completed by adding the actuator model. For this shaped plant G_{s}^j an $\mathscr{H}_\infty$ static loop shaping controller is computed following Theorem 3.5 and the discussion of the next section; in addition, the corresponding robustness margin ϵ^j is obtained.

Step 3 - Line Search or Reset. Performing a line search for subsequent values of η belonging to Σ_η^m, successive shaped plants G_{s}^f are computed, until the gap $\delta_{\mathrm{g}}(G_{\mathrm{s}}^j, G_{\mathrm{s}}^f)$ between the nominal initial plant and the successive one is greater or equal than the robustness margin ϵ^j.

[25] Of course more points could be used; however three seem to be adequate in this case.

[26] Note that the right extremal values of the flight envelope are computed using Eq. 4.22; for more details concerning the missile operating domain refer to Section 4.1.2.3.

If this is the case, then a new operating point P^j is chosen and the algorithm jumps back to Step 2, except for the case when the end of the flight envelope is reached. In this case, even if $\delta_g(G_s^j, G_s^f) < \epsilon^j$, the final point is selected and the procedure jumps to Step 2 where a new set value Σ_M is chosen ($m = m + 1$) and the algorithm continues for all m.

`Algorithm results`

Using this algorithm, totally twelve synthesis points are computed. The $\mathscr{H}_\infty$ controller treats the internal/external loops outputs and $K_\infty = [K_{\text{PI},\infty}\ K_{q,\infty}] \in \mathbb{R}^{1\times 2}$ (for more details see the next section). The coordinates of the synthesis points, the corresponding gains and robustness margins are shown in Table 6.6.

Table 6.6: Robust controller gains

Points	$[\eta, M]$	$K_{\text{PI},\infty}$	$K_{q,\infty}$	e
1	[0 1.5]	0.594	1.098	0.369
2	[4.20 1.5]	0.626	1.099	0.378
3	[9.7969 1.5]	0.657	1.129	0.381
4	[0 2.25]	0.645	1.131	0.391
5	[4.20 2.25]	0.643	1.093	0.387
6	[12.91 2.25]	0.720	1.152	0.396
7	[21.43 2.25]	0.745	1.155	0.389
8	[0 3]	0.569	0.960	0.292
9	[4.20 3]	0.544	0.874	0.369
10	[11.11 3]	0.628	1.006	0.399
11	[21.01 3]	0.770	1.167	0.395
12	[33.0559 3]	0.738	1.133	0.383

The gap evolution is shown in Fig. 6.23 for $M = 1.5$ (for the rest of the values for M the profile is similar). For this Mach value the operating point algorithm finds three points (see Table 6.6). Take for example the first one ($\eta = 0$); the robustness margin ϵ^j and the maximum robustness margin $\epsilon^j_{\max}$ (corresponding to the full order robust controller) are shown with black points. The red points show the gap between subsequent *candidate* synthesis points and the initial one; once the gap becomes greater than ϵ^j, a new synthesis point is selected.

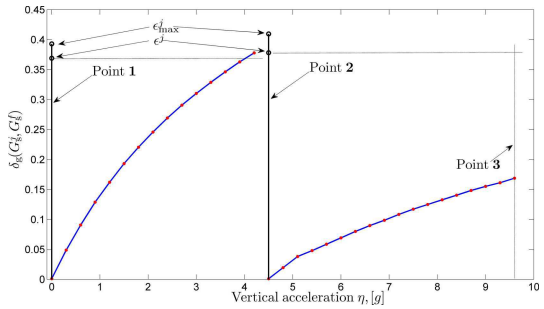


Figure 6.23: Gap evolution for $M = 1.5$.

6.3.1.3 $\mathscr{H}_\infty$ Controller Synthesis

The analysis concerning the computation of the robust $\mathscr{H}_\infty$ controllers is essentially the same as in Section 6.2.1.3 with the corresponding problem being given by Eq. 6.3, and will not be repeated.

`Control structure`

The robust controller structure uses a slightly different form here, comparing to the one concerning the ARV benchmark, where all three channels of the PID controller were treated (see Fig. 6.7). The loop now is opened using *directly* the output of the PI controller $\delta_{c,PI}$ (instead of e.g. separately the proportional & integral channels) and the output of the P controller $\delta_{c,P}$. Two static gains $K_{PI,\infty}, K_{q,\infty}$ are thus computed treating each controller output and their outputs are summed in order to provide the final control signal δ_c (see Fig. 6.24).

The step responses for the PI/P shaped plant (blue) and the PI/P shaped plant plus the robust controller (red) for the synthesis point (No. 12) are shown in Fig. 6.25 (similar behavior holds for all points); with the output now being more damped. The PI/P controllers could have very well been adjusted initially to provide such damping; however it should not be forgotten that the robust controllers, coupled with the gap metric theory, additionally capture the plant's nonlinear dynamics variation using the algorithm of the previous section.

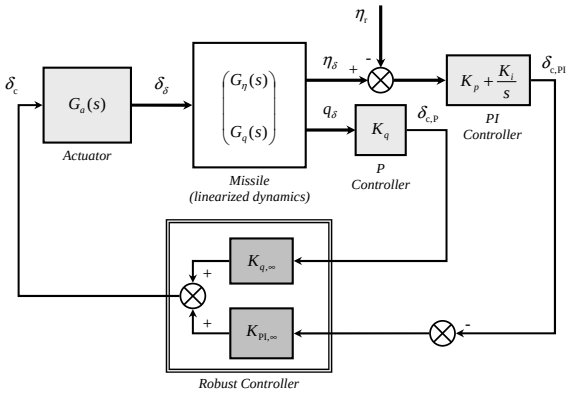


Figure 6.24: Robust controller synthesis block diagram.

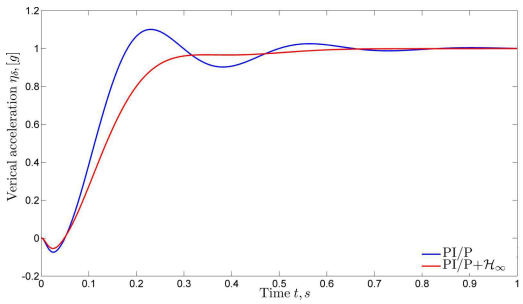


Figure 6.25: Step response comparison.

6.3.2 Gain Interpolation

The gain-scheduled controller detailed in the next section uses gain interpolation in order to update its parameters as a function of the scheduling vector $\varrho = [\eta \ M]^T$. Gain interpolation is also used by the operating point selection algorithm of Section 6.3.1.2 in order to provide PI/P controller gain values for the loop shaping needed at any point on the missile's flight envelope.

The PI/P controller gains $K_q, K_\mathrm{p}, K_\mathrm{i}$ are designed using the analysis of Section 6.3.1.1 at nine synthesis points (see Table 6.4) forming four *trapezoidal* scheduling regions $\Gamma^1, \Gamma^2, \Gamma^3$ and Γ^4 (the same used in Chapter 5). The robust controller gains $K_{\mathrm{PI},\infty}, K_{q,\infty}$ are designed at twelve synthesis points (see Table 6.6) forming twelve *triangular* scheduling regions.

PI/P gain interpolation

Concerning the PI/P controller gains, the trapezoidal interpolation is performed like the one used for the controller blending method in Section 5.4.2.1 (see especially Fig. 5.9). Consider for example the gain K_q and the first scheduling region Γ^1; define as $K_q^{\mathrm{ll}}, K_q^{\mathrm{lr}}, K_q^{\mathrm{ur}}, K_q^{\mathrm{ul}}$ the gain values at the lower-left, lower-right, upper-right and upper-left corners of the scheduling region. The normalized quantities a_1, a_2 (with $0 \leq a_i \leq 1$) give the relative distance of the current interpolated value $K_q(t)$ from the left (lower & upper) points and from the lower (left & right) points respectively. These distance need some trigonometry to be computed and the calculations will not be given here; it must however be stressed out that once the normalized quantities are found, the interpolated value for the controller gain is simply obtained by[27]:

$$K_q(t) = \big[1 - a_1(t)\big] K_q^{\mathrm{l}}(t) + a_1(t) K_q^{\mathrm{u}}(t) \tag{6.18}$$

where:

$$K_q^{\mathrm{l}}(t) = \big[1 - a_2(t)\big] K_q^{\mathrm{ll}} + a_2(t) K_q^{\mathrm{lr}} \tag{6.19}$$

$$K_q^{\mathrm{u}}(t) = \big[1 - a_2(t)\big] K_q^{\mathrm{ul}} + a_2(t) K_q^{\mathrm{ur}}. \tag{6.20}$$

To illustrate the interpolation method used here, a spiral scheduling vector trajectory is shown in Fig. 6.26 (red points).

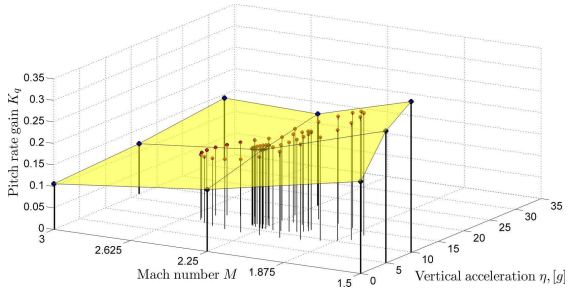


Figure 6.26: Pitch rate gain interpolation.

[27] Dependence on the scheduling vector is omitted; only time dependence is used for simplicity.

$\mathscr{H}_\infty$ controller interp/tion

The robust controller gains $K_{\mathrm{PI},\infty}, K_{q,\infty}$ now are interpolated using triangular scheduling regions as a result of Delaunay triangulation. In order to ensure a more correct triangulation of the missile flight envelope, the scheduling regions are obtained by considering only the portion of the flight envelope that corresponds to two subsequent values of the Mach number gridding.

For example, if the current value of the scheduling vector for the Mach number is $M = 1.6$ and given that the gridding values are $1.5, 2.25, 3$, only the portion of the flight envelope corresponding to $1.5 \leq M \leq 2.25$ is triangulated. In this case the triangular scheduling regions are illustrated in Fig. 6.27.

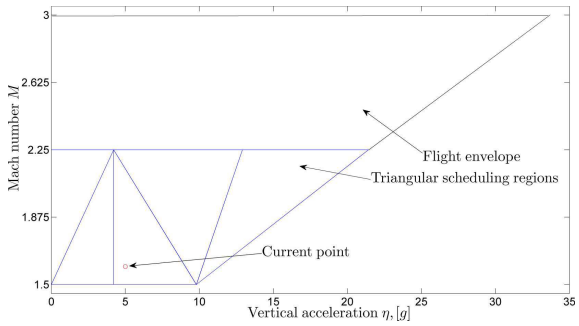
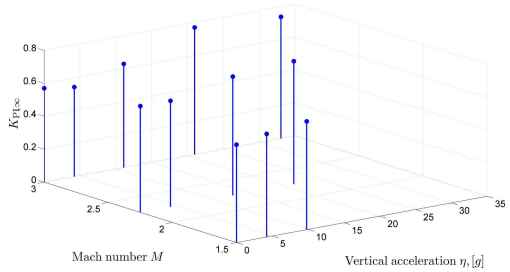
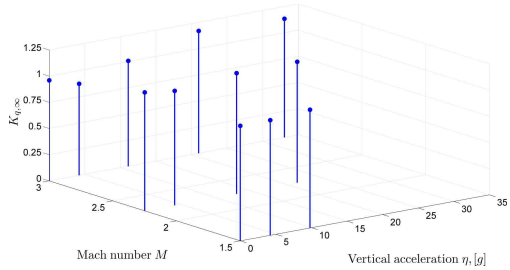


Figure 6.27: Flight envelope triangulation.

Finally in Figs. 6.28a-6.28b, the robust controller gains $K_{\mathrm{PI},\infty}, K_{q,\infty}$ are shown in 3D (see also Table 6.6).



(a) *PI-channel gain*



(b) *Pitch rate-channel gain*

Figure 6.28: Robust controller gains.

6.3.3 Controller Implementation & Validation

The missile gain-scheduled controller will be detailed in this section. The first subsection presents the controller structure whereas the second one presents the main results concerning the nominal and uncertain behavior of the missile under the specific controller; the chapter ends with a short discussion.

6.3.3.1 Nonlinear Gain-scheduled Controller

The nonlinear gain-scheduled controller block diagram is shown in Fig. 6.29. The grey blocks represent the nonlinear missile pitch-axis dynamics, the actuator dynamics and the Mach dynamics (see Eqs. 4.1-4.9, Eq. 4.11 and Eq. 4.10 respectively). The total control signal δ is the sum of the trim control δ_{r} and the closed loop control signal δ_δ.

`Trim & feed/ward control`

The trim control $\delta_{\mathrm{r}} = \delta(\varrho_{\mathrm{r}})$ is computed using the analysis of Section 4.1.2.2 as a function of the scheduling vector $\varrho = [\eta_{\mathrm{r}} \; M_{\mathrm{r}}]^T$. A feedforward controller is added before the trim control block in order to 'schedule on a slow variable' as it is often the case in gain scheduling. This controller is a simple first-order filter acting on the reference signal η_{r} dampening the system's output $\eta(t)$; as a result, the trim control is computed using the filtered reference signal $\eta_{\mathrm{r},f}$.

`Fast loop`

The closed loop control signal δ_δ is the sum of the outputs of the PI and the pitch rate controllers (see in Fig. 6.14), and scaled by the robust $\mathscr{H}_\infty$ controller. The inputs to these controllers are the 'error signals' η_δ and q_δ; the former is computed by subtracting the missile output by the reference output signal whereas the latter is also computed in a similar way[28]. This closed loop is the *fast* one since it stabilizes the system and ensures trajectory following.

`Slow loop`

The additional scheduling loop is the *slow* one, and uses the scheduling vector ϱ and the 'Interpolation Mechanism' block (in red) in order to update all five gains of the inner control loop as a function of the system's operating conditions (see Section 6.3.2). A similar feedforward controller as the trim control one (but with a different time constant) is used to smooth the gain variation[29]. The gain-scheduled controller is thus governed by the following equations[30]:

`Global controller`

$$\begin{aligned} \dot{x}_{\mathrm{c}} &= \mathbf{A}_{\mathrm{c}}(\varrho)\big[x_{\mathrm{c}} - x_{\mathrm{c}}(\varrho)\big] + \mathbf{B}_{\mathrm{c}}(\varrho)\big[y - y_{\mathrm{r}}(\varrho)\big] \\ \delta &= \mathbf{C}_{\mathrm{c}}(\varrho)\big[x_{\mathrm{c}} - x_{\mathrm{c}}(\varrho)\big] + \mathbf{D}_{\mathrm{c}}(\varrho)\big[y - y_{\mathrm{r}}(\varrho)\big] + \delta(\varrho_{\mathrm{r}}) \end{aligned} \tag{6.21}$$

with $y = [\eta \quad q]^T, \dot{x}_{\mathrm{c}} = \eta_\delta$ and:

$$\mathbf{A}_{\mathrm{c}}(\varrho) = 0 \tag{6.22}$$

$$\mathbf{B}_{\mathrm{c}}(\varrho) = [1 \quad 0] \tag{6.23}$$

$$\mathbf{C}_{\mathrm{c}}(\varrho) = K_{\mathrm{PI},\infty}(\varrho)K_{\mathrm{i}}(\varrho) \tag{6.24}$$

$$\mathbf{D}_{\mathrm{c}}(\varrho) = [K_{\mathrm{PI},\infty}(\varrho)K_{\mathrm{p}}(\varrho) \quad K_{q,\infty}(\varrho)K_q(\varrho)]. \tag{6.25}$$

[28] To compute q_r, an additional 'Trim Values' block is used (see Section 4.1.2.2 & Eq. 4.20).
[29] For details on tuning both feedforward controllers, see [140], §VI.A.
[30] The feedforward filters are not considered in the equations for simplicity.

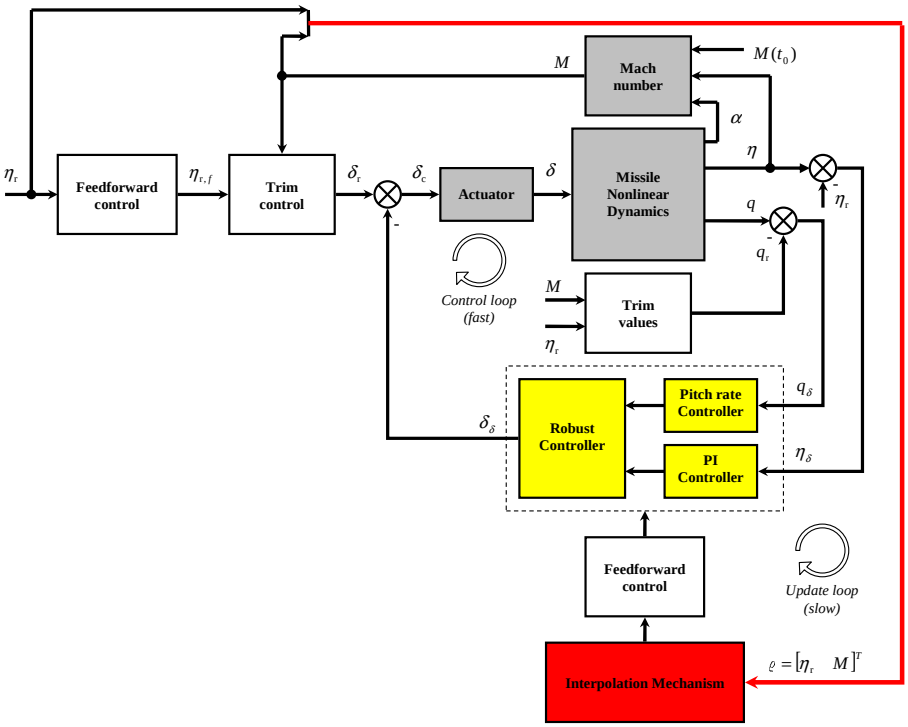


Figure 6.29: Gain-scheduled controller block diagram.

6.3.3.2 Simulation Results

The simulation results presented in this section are obtained under the same conditions as the ones in Chapter 5 (Mach number trajectory, reference signal scenario); for more information refer to Figs. 5.11a-5.11c.

Nominal case results

The first figure presented here (see Fig. 6.30) illustrates the missile's output vertical acceleration η as a function of time, when applying two gain-scheduled controllers: the PI/P scheduled controller using the nine LTI controllers of Section 6.3.1.1 (blue line), and the full PI/P plus robust $\mathscr{H}_\infty$ scheduled controller detailed in Sections 6.3.1.2-6.3.1.3 (red line). The tracking reference η_r and the filtered tracking reference $\eta_{\mathrm{r},f}$ are also shown (black lines).

When observing the responses, the robust controller is clearly superior to the simple PI/P controller, providing adequate damping to the system's output while retaining excellent time constants and steady-state errors and satisfying the missile control objectives of Section 5.3[31]. The response characteristics for the robust gain-scheduled controller are also presented in detail in Table 6.7 for all four reference operating points.

Table 6.7: Controller nonlinear performance

Points	No. 1	No. 2	No. 3	No. 4	**Limit**
τ (ms)	260	283	250	319	**350**
M_p (%)	3.88	6.25	0.00	5.33	**10**
e_ss (%)	0.19	0.00	*1.71*	*1.05*	**1**
$\lvert\dot{\delta}\rvert$ (deg/s)[(ii)]	4.44	6.03	6.64	7.20	**25**

[(i)] The violated constraints are shown in italics.
[(ii)] The control rate is normalized by the amplitude of the reference signal η_r.

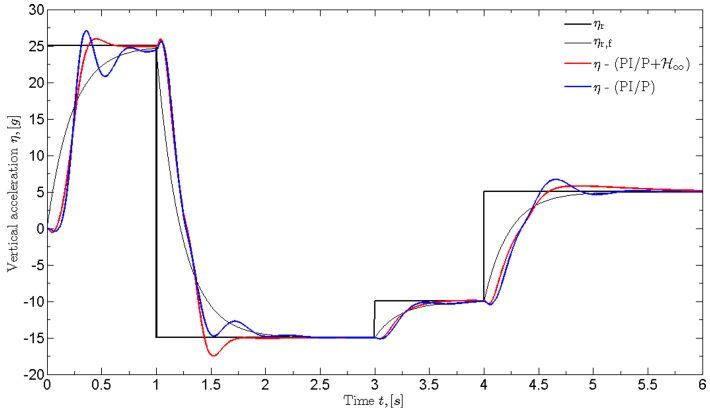
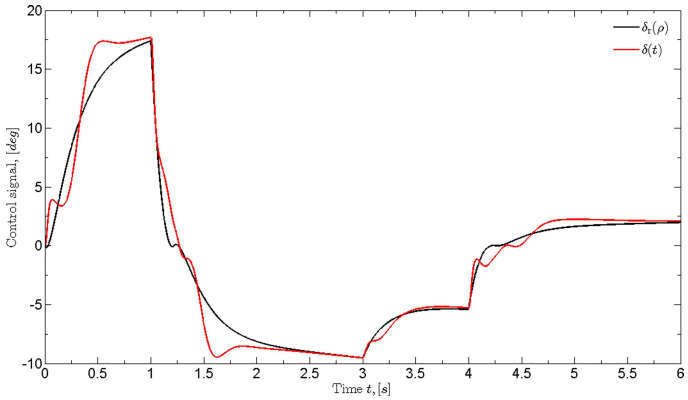
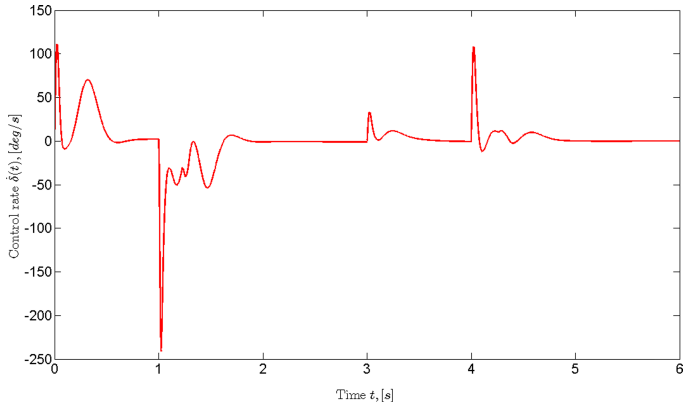


Figure 6.30: Vertical acceleration (comparison).

[31] There is a small violation on the steady-state errors for points No. 3 & 4.



(a) *Control signal*



(b) *Control rate signal*

Figure 6.31: Controller outputs.

The two figures in the previous page present the gain-scheduled controller output characteristics. In Fig. 6.31a, the total control signal $\delta(t)$[32] after the actuator is shown (in red)[33] along with the trim control signal (in blue). It may be observed that the control signal does not present any discontinuities or transients as with the *controller blending* method.

An interesting phenomenon may be also observed from Fig. 6.31a concerning moving equilibria. Take for example the transient response for the first reference point $0 \rightarrow 25$g; the output $\eta(t)$ has settled down for $t > 0.7$s (see Fig. 6.30), however the control signal $\delta(t)$ continues to increase (see Fig. 6.31a). This may be explained from the fact that during the system's operation the Mach number continues to drop rapidly and thus the trim control also augments according to Fig. 4.4c.

The control rate $\dot{\delta}(t)$ illustrated in Fig. 6.31b is also well inside the constraints (25deg/s for 1g reference commands) as it is also seen from Table 6.7. The obvious question is whether the response could be faster and exploit all the available bandwidth; the answer is of course positive but with the expense of smaller stability margins presented further down. In any case, a major role in this issue is played by the feedforward controllers that provide damping and do not let the control signal be too aggressive; the time constants however remain fast for all reference points considered in the benchmark tests (see Fig. 6.30).

The missile state vector $x = [\alpha \ q]^T$ is depicted in Fig. 6.32; the angle of attack α lies well within its domain of operation (recall from Section 4.1.1 that the nonlinear pitch-axis missile model is valid for $|\alpha| \leq 20^\circ$) and reaches its equilibrium values corresponding to the output reference trajectory, practically with no overshoot and in a smooth way. In addition, the pitch rate q is rapidly augmenting when there is a change to the output operating point and then reaches asymptotically its equilibrium value (see also Fig. 4.3c).

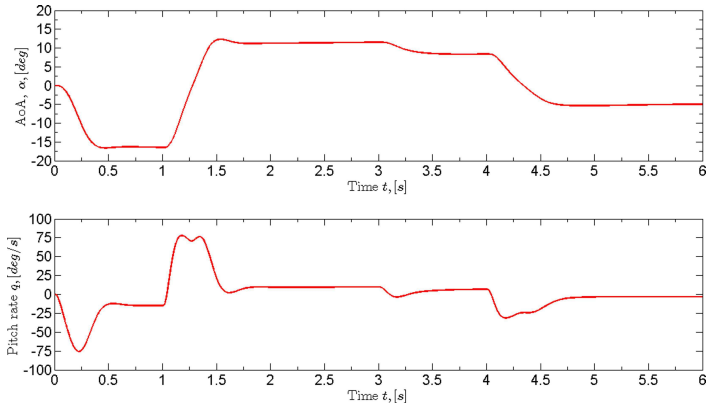


Figure 6.32: Missile state vector.

[32]Recall that the total control signal is the sum of the nominal (or trim) control signal δ_r plus the closed loop stabilization signal δ_δ.

[33]The control signal *before* the actuator δ_c is not shown here since it is very close to δ.

The robust controller gains $K_{\mathrm{PI},\infty}, K_{q,\infty}$ time evolution for the simulation scenario chosen is shown in Fig. 6.33[34]. The gain evolution is smooth and the transition rate between operating points is influenced by the feedforward controller applied on the output of the interpolation mechanism. The same phenomenon as with the control signal is also observed here; the gains do not reach steady-state values since they are interpolated not only as a function of η, but also as a function of the Mach number M that continues to drop during the system's operation.

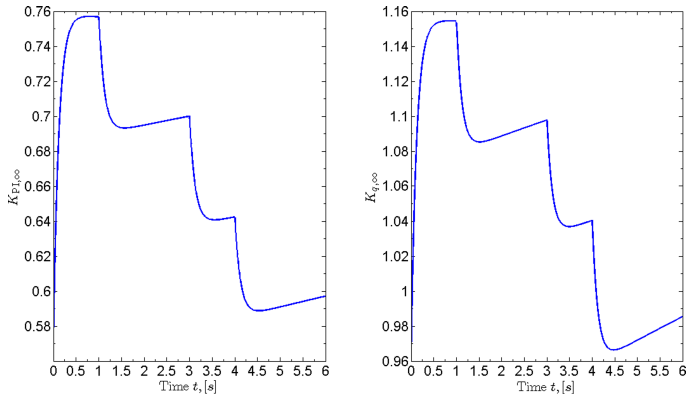


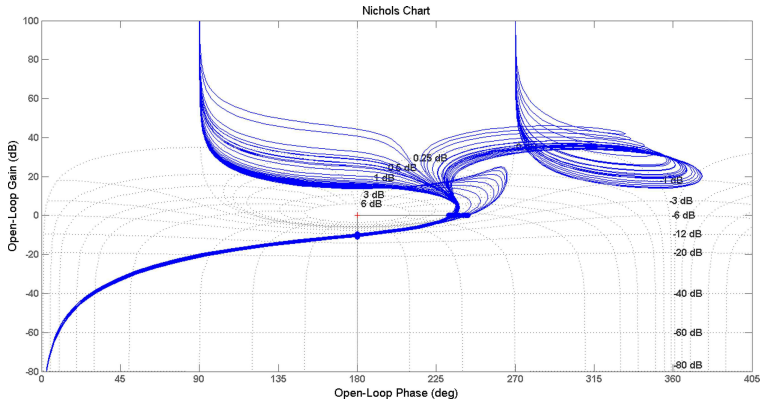
Figure 6.33: Robust controller gains time evolution.

In order to test the stability of the missile under the gain-scheduled controller, the loop is opened before the actuator and the plant is linearized freezing the time each 0.1s during the benchmark scenario of Fig. 6.30 (totally 60 open loop models are obtained). Then the Nichols & Bode diagrams of these open loops are superimposed and illustrated in Figs. 6.34a-6.34b. The worst gain and phase margins are 9.5dB and 52° respectively whereas the worst magnitude attenuation at high frequencies[35] d_{att} is approximatively 27.5db; slightly violating the 30dB robustness constraint limit R_2 (see Section 5.3).

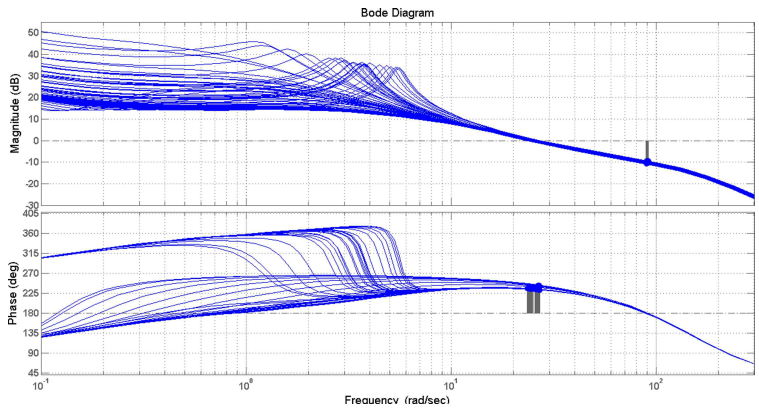
Uncertain case results

So far, the gain-scheduled controller's performance has been tested for the two performance and the second robustness constraint of the control objectives described in Section 5.3. The last test concerns the controller's robustness in the face of disturbances in the missile's pitch aerodynamic coefficients $a_{\mathrm{m}}, b_{\mathrm{m}}, c_{\mathrm{m}}$ and d_{m} (see Table 4.1). These coefficients are independently perturbed in two groups (the first three in the same manner and the fourth separately) with a maximum deviation of 25% of their nominal values and all resulting outputs $\eta(t)$ are superimposed in Fig. 6.35a. The independent perturbations follow a Gaussian distribution with zero mean and standard deviation $\sigma = 0.25/3$, with totally 150 cases being considered[36]. The envelope created around the nominal response is not too large and the plant demonstrates robust stability.

[34]The PI/P gains follow similar patterns and are not shown here for brevity.
[35]Recall from Section 5.3 that this corresponds to the robustness objective R_2.
[36]This scenario corresponds to a Monte-Carlo analysis.

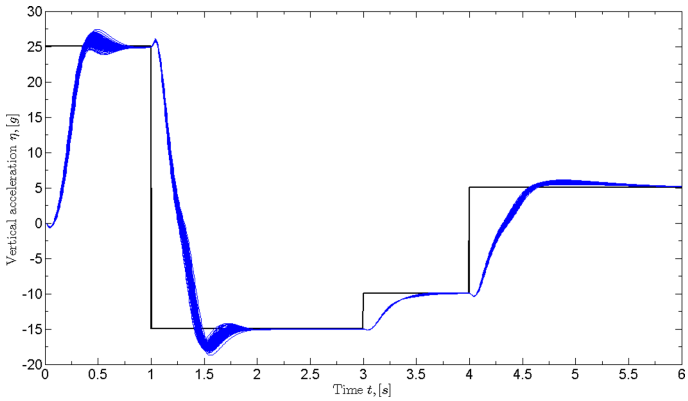


(a) *Nichols diagrams*

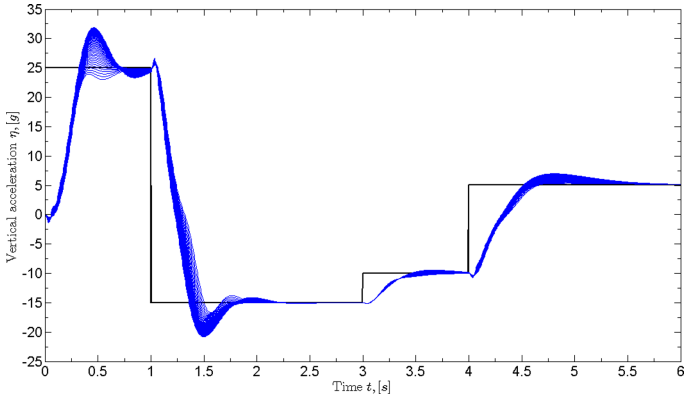


(b) *Bode diagrams*

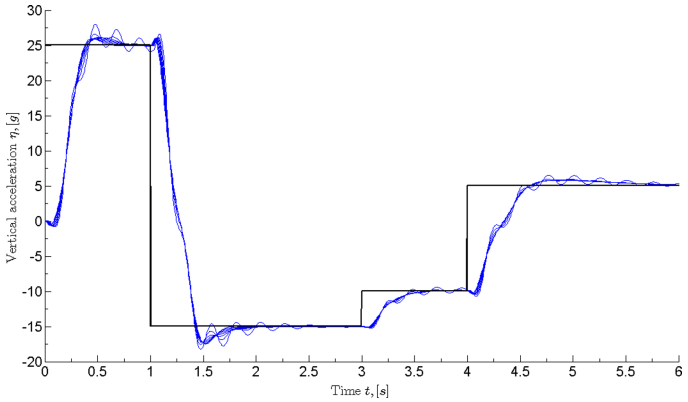
Figure 6.34: Open loop linearization results.



(a) *Parametric robustness*



(b) *Gain robustness*



(c) *Delay robustness*

Figure 6.35: Additional robustness tests.

Additional tests

Some additional tests were performed to demonstrate the robust stability of the missile in the face of open loop gain & delay augmentation. First, a variable gain was added before the actuator and the gain margin of the loop was verified by augmenting the gain from 0.7 to 3 (corresponding to the 9.5db of the worst GM of the previous frozen-time linearization analysis) and taking totally 47 simulation cases. The responses were superimposed in Fig. 6.35b; it is apparent that the output becomes more oscillatory as the loop gain increases but the system remains stable. Second, a variable loop delay was added, taking values from 0 (nominal case) to 35ms (worst case), in order to test the plant's delay margin. The totally 8 cases are also superimposed and illustrated in Fig. 6.35c; the limits of stability are clearly demonstrated as the output starts to oscillate as the loop delay increases.

6.3.3.3 Discussion

In this second part of the chapter, a novel gain-scheduling approach for the control of the Reichert missile benchmark model, based on the e-LSDP, was proposed and extensively tested in order to demonstrate both its good time performance and robustness.

The advantages of the proposed method are its simple structure (low order controllers), its ability to take into account the plant's nonlinear dynamics variation (gap metric algorithm) and its simplicity of interpolation (gain interpolation). The gain-scheduled controller depicted in Fig. 6.29 is thus very straightforward to implement on a real system since it does not demand any complex calculations or great memory as for example other gain-scheduled controllers found in the bibliography (see for example [17, 74, 103] and other approaches detailed in Section 5.2).

Conclusions & Future Work

This thesis involved research on the subject of linearization-based gain scheduling for nonlinear parameter-dependent systems. Given that these systems require controllers that are on the one hand adaptive and on the other hand easy to implement, the designer's task is not straightforward.

To obtain the required performance/robustness characteristics, most gain-scheduled control schemes reside on high-order LTI controllers designed only at a small number of synthesis points to limit memory/processing requirements (see for example [103], [17]). Other control schemes use simpler structure controllers (e.g. PID) designed at a relatively higher number of points [37]. The LTI controllers are then interpolated (or simply switched) using one of the interpolation strategies presented in the first chapter of this work. In the context of linearization-based gain-scheduling, some schemes even offer some stability guaranties for the gain-scheduled system (see for example [150]).

This thesis has shown that in some cases a gain-scheduled system need not be based on high order controllers or use a significant number of operating points so as to ensure a good behavior of the controlled plant. In fact, it has been demonstrated that low order controllers (e.g. PID) may form a good basis for a high performance gain-scheduled system, if they are complemented by specific 'corrective' gains designed around a smartly chosen set of synthesis points.

This strategy has been called the *e*-LSDP (*extended*-Loop Shaping Design Procedure) and can be briefly decomposed into three steps: *loop shaping*, *operating point selection* and $\mathscr{H}_\infty$ *controller computation*. The loop shaping is performed using simple structure LTI controllers at a relatively small number of points in order to obtain a basic tuning for the system. Then, based on this tuning, a complementary number of synthesis points is computed (using interpolated values of the PID controllers) from a specific operating point selection algorithm. This algorithm uses the gap metric as a distance measure between systems in order to capture the nonlinearity variation of the initial parameter-dependent system and this variation being regarded as unstructured normalized co-prime factor uncertainty. For this additional number of synthesis points, the PID controller outputs are treated/corrected by static $\mathscr{H}_\infty$ controllers that come as a natural connection of the gap metric tool and the standard LSDP.

[37] This is the case mostly in the industry.

To obtain the gain-scheduled controller, all gains (PID plus robust controller) are interpolated according to the scheduling vector inside the whole system's operating domain. It is clear that this interpolation strategy is the simplest possible since it does not require large memory to store the controllers or significant computational capacities to compute the interpolated control law, as is the case for example with state-space or ZPK interpolation. In addition it does not cause signal discontinuities or transients as with the controller blending method.

This method has been tested on two benchmark systems: a pitch axis missile model and an atmosphere re-entry vehicle, to obtain a tracking/regulating autopilot respectively. In the first case, the autopilot should provide adequate tracking performance for output reference step commands whereas in the second case, it should be able to regulate the AoA around a given value. In both cases there are strict constraints concerning robust stability in the face of uncertainties, control signal energy minimization, stability margin maximization etc.

In the missile case, the loop shaping controllers are designed at nine synthesis points and are of a specific internal/external feedback configuration (internal proportional feedback on the pitch rate and external proportional plus integral feedback on the vertical acceleration). These controllers offer excellent properties on the linear case but, being of relatively low order, are not adequate when considering the nonlinear gain-scheduled system. An additional set of twelve static, $\mathscr{H}_\infty$ output feedback controllers are designed at a new set of synthesis points using the gap metric-based operating point selection algorithm proposed in this thesis. These controllers are two gains treating each of the two initial feedback loops, thus robustifying the system and offering excellent tracking characteristics. All five gains are then scheduled as a function of the Mach number and the output to provide a global nonlinear gain-scheduled controller[38].

In the re-entry vehicle case, the loop shaping controllers are designed only at five synthesis points of the flight envelope, formed by the AoA (regulated variable) and the Mach number. The control structure is even simpler than in the missile case and is consisted of a PID controller preceded by a first-order filter. The controllers for both systems are designed using classical frequency domain concepts and re-tuned using optimization techniques; however, they also are not sufficient on their own. An additional set of six static robust controllers is designed at carefully chosen operating points, using the aforementioned point selection algorithm, treating each of the proportional/integral/derivative channels of the initial loop shaping controllers. Finally, all LTI controllers are discretized and the global control law is computed by appropriately scheduling the totally seven gains. Both systems are exhaustively tested using standard benchmark scenarios and Monte Carlo analysis in order to ensure their good behavior throughout all their domain of operation both in the nominal case but also when uncertainties are introduced on their aerodynamic coefficients/functions.

[38]The gain-scheduled controller is aided by a trim preserving open loop controller and a carefully tuned feedforward loop (for more details see Chapter 6).

The gain-scheduled control strategy proposed in this work is by no means perfect and several enhancements may be considered. Take for example the operating point selection algorithm, described in detail in the last chapter of this work, and consider the missile's autopilot case. The algorithm performs a linear search for candidate operating points, for constant values of the first scheduling variable (Mach number) on the trapezoidal flight envelope and increasing values of the second scheduling variable (vertical acceleration). This search thus considers the gap metric variation only as a function of a single variable, based on a given gridding for the second variable. Evidently, this is not the best way to perform the search, even though the results obtained are very good. In addition, the gap is computed with respect to a single initial system and not for a collection of systems forming say, a triangular or trapezoidal region where the gap of each system with respect of all others will be inferior than a given value. In this way, a more re-assuring partitioning of the operating domain could be performed, with the expense of course of a more demanding/complex operating point selection algorithm [39].

Possible ways to perform this could be for example exhaustive triangulation of the operating domain: start with an initial triangle, check all the gaps; if at least one is superior to a given value then divide the initial triangle in three sub-triangles and restart. Another method could be circular search: start with a given point say, at the center of the flight envelope and then check the gaps between this point and a sufficient number of neighbor points residing on the circumference of a circle of varying radius. When at least one gap is superior to a given value then stop and add an additional synthesis point on the given circle; then take out the covered area and continue the search until all the operating domain is covered.

In the same context, consider now the linear search concerning the re-entry vehicle operating point selection algorithm. The algorithm starts by comparing the gap between an initial point for $M = 26$ and subsequent points; then it continues until $M = 4$. This approach however maybe conservative since it is possible that the last two synthesis point may be close; this is due to the fact that when the algorithm is searching for the last point, the operating domain may end suddenly. As an extension to this fact, if the algorithm starts from the opposite end of the envelope, it is highly probable that the synthesis points will be *different* than before. A solution to this problem may be for instance, to start from the center of the operating domain and proceed in both directions in the same time.

Another issue concerning the gap metric-based operating point selection algorithm is the heavy demands on processing power in order to check the gap for a fine gridding of possible synthesis points. To solve this, variable-step/grid methods could be considered instead of fixed ones; that check will accelerate the search and may use for example the gap's rate of variation.

[39] See Fig. 6.36 for a brainstorming session!

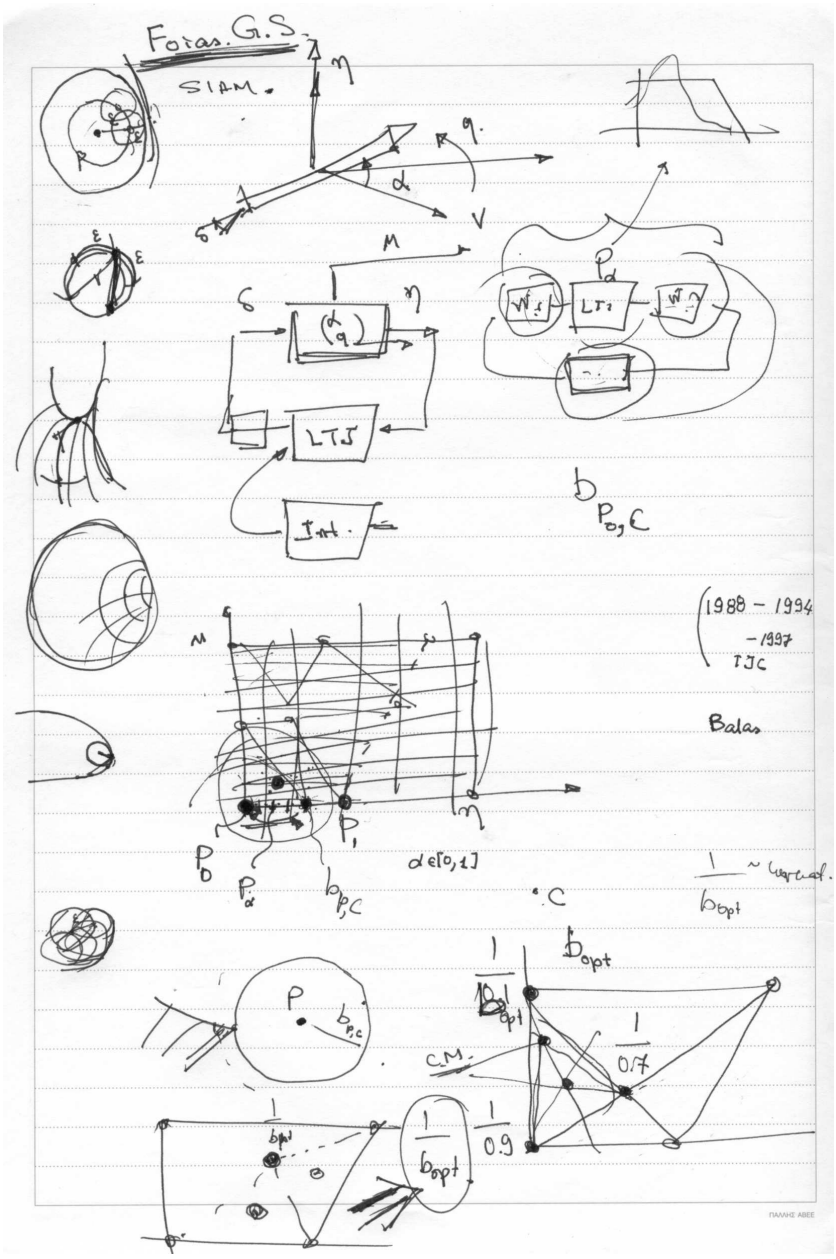


Figure 6.36: Brainstorming: Georgiou & Theodoulis, CDC'07, New Orleans, USA.

From a practical point of view, another possible research direction would be the exploitation of techniques related to the nonlinear gap-metric theory (see for example [51]) or exploitation of other types of metrics such as the ν-gap metric proposed by Vinnicombe (see for example [145]).

The next step of this work surely is the integration of the proposed gain-scheduling strategy to the 3-axes autopilot problem; notably constructing an autopilot for the roll and yaw axis of both systems. This will result to a true MIMO problem and it should be a real challenge to find out the effectiveness of the e-LSDP in this coupled case. Finally, more complex mathematical models of the systems could be considered, taking into account flexible modes, digital implementation, the guidance loop etc.

References

[1] Research on Guided Missiles. *Journal of the Franklin Institut*, 241(1):40, 1947.

[2] I.-J. Adounkpé. *Robustesse dans un Cadre Non-Linéaire : Application au Pilotage des Missiles*. PhD thesis, Université Paris-Sud XI/SUPELEC, France, 2004.

[3] I.-J. Adounkpé, E. Godoy, and J.-P. Harcaut. Robustness Analysis of a Nonlinear Missile Autopilot: An Integral Quadratic Constraint Approach. In *Proceedings of the* 5^{th} *Asian Control Conference*, pages 1833–1840, Melbourne, Australia, 2004.

[4] D. Alazard and P. Apkarian. Exact Observer-based Structures for Arbitrary Compensators. *International Journal of Robust & Nonlinear Control*, 9:101–118, 1999.

[5] F. Amato, M. Mattei, and A. Pironti. Gain-scheduled Control for Discrete-time Systems Depending on Bounded Rate Parameters. *International Journal of Robust & Nonlinear Control*, 15:473–494, 2005.

[6] P. Ambos, G. Duc, and C. Valentin-Charbonnel. Parameter-dependent Reduced Order $\mathscr{H}_\infty$ Controller Using a Loop Shaping Approach: A Solution to the EDF Benchmark Problem. In *Proceedings of the* 14^{th} *IFAC World Congress*, pages 471–476, Beijing, China, 1999.

[7] N. Aouf, D.G. Bates, I. Postlethwaite, and B. Boulet. Scheduling Schemes for an Integrated Flight and Propulsion Control System. *Control Engineering Practice*, 10:685–696, 2002.

[8] N. Aouf, B. Boulet, and R. Botez. A Gain Scheduling Approach for a Flexible Aircraft. In *Proceedings of the* 21^{th} *American Control Conference*, pages 4439–4442, Anchorage, USA, 2002. IEEE.

[9] P. Apkarian and R.J. Adams. Advanced Gain Scheduling Techniques for Uncertain Systems. *IEEE Transactions on Control Systems Technology*, 6(1):21–32, 1998.

[10] P. Apkarian, J.-M. Biannic, and P. Gahinet. Self-scheduled $\mathscr{H}_\infty$ Control of Missile via Linear Matrix Inequalities. *Journal of Guidance, Control and Dynamics*, 18(3):532–538, 1995.

[11] P. Apkarian and P. Gahinet. A Linear Matrix Inequality Approach to $\mathscr{H}_\infty$ Control. *International Journal of Robust & Nonlinear Control*, 4:421–448, 1994.

[12] P. Apkarian and P. Gahinet. A Convex Characterization of Gain-scheduled $\mathscr{H}_\infty$ Controllers. *IEEE Transactions on Automatic Control*, 40:853–864, 1995.

[13] P. Apkarian, P. Gahinet, and G. Becker. Self-scheduled $\mathscr{H}_\infty$ Control of Linear Parameter Varying Systems: A Design Example. *Automatica*, 31:1251–1261, 1995.

[14] P. Apkarian, P.C. Pellanda, and H.D. Tuan. Mixed $\mathscr{H}_2/\mathscr{H}_\infty$ Multi-channel Linear Parameter Varying Control in Discrete Time. *Systems & Control Letters*, 41:333–346, 2000.

[15] K.J. Åstrom and B. Wittenmark. *Adaptive Control*. Addison-Wesley, 1995.

[16] L. Atran. Analysis of a Nonlinear Control System for Stabilizing a Missile. *IRE Transactions on Automatic Control*, 46(3):671–675, 1958.

[17] G.J. Balas and A.K. Packard. Design of Robust, Time-varying Controllers for Missile Autopilots. In *Proceedings of the* 1^{st} *IEEE Conference on Control Applications*, pages 104–110, Dayton, USA, 1992. IEEE.

[18] G. Becker and A. Packard. Robust Performance of Linear Parametrically-varying Systems Using Parametrically-dependent Linear Feedback. *Systems & Control Letters*, 23:205–215, 1994.

[19] D.J. Bell. *Mathematics of Linear and Nonlinear Systems*. Oxford Science Publications, 1990.

[20] D.J. Bender and R.A. Fowell. Computing the Estimator-controller Form of a Compensator. *International Journal of Control*, 41(6):1565–1575, 1985.

[21] D.J. Bender and R.A. Fowell. Some Considerations for Estimator-based Compensator Design. *International Journal of Control*, 41(6):1577–1588, 1985.

[22] C.J. Bett and M.D. Lemmon. Bounded Amplitude Performance of Switched LPV Systems with Applications to Hybrid Systems. *Automatica*, 35:491–503, 1999.

[23] C. Bohn, A. Cortabarria, V. Härtel, and K. Kowalczyk. Active Control of Engine-induced Vibrations in Automotive Vehicles Using Disturbance Observer Gain Scheduling. *Control Engineering Practice*, 12:1029–1039, 2004.

[24] H. Buschek. Full Envelope Missile Autopilot Design Using Gain-scheduled Robust Control. *Journal of Guidance, Control and Dynamics*, 22(1):115–122, 1999.

[25] L.H. Carter and J.S. Shamma. Gain-scheduled Bank-to-Turn Autopilot Design Using Linear Parameter Varying Transformations. *Journal of Guidance, Control and Dynamics*, 19(5):1056–1063, 1996.

[26] V. Chérau, H. Tanguy, and G. Lebret. Interpolated Versus Polytopic Gain Scheduling Control Laws for Fin/rudder Roll Stabilization of Ships. In *Proceedings of the* 44^{th} *IEEE Conference on Decision & Control*, pages 2284–2289, Seville, Spain, 2005. IEEE.

[27] M. Chilali and P. Gahinet. $\mathscr{H}_\infty$ Design with Pole Placement Constraints: An LMI Approach. *IEEE Transactions on Automatic Control*, 41(3):358–367, 1996.

[28] M. Chilali, P. Gahinet, and P. Apkarian. Robust Pole Placement in LMI Regions. *IEEE Transactions on Automatic Control*, 44(12):2257–2270, 1999.

[29] F. Claveau, P. Chevrel, and K. Knittel. A 2DOF Gain-scheduled Controller Design Methodology for a Multi-motor Web Transport System. *Control Engineering Practice*, 16:609–622, 2008.

[30] J.R. Cleminson and R.A. Hyde. Control and Maneuverability of a Square Cross Section Missile. *Journal of Guidance, Control and Dynamics*, 29(1):25–38, 2006.

[31] C.A. Desoer and M. Vidyasagar. *Feedback Systems: Input-Output Properties*. Academic Press, 1975.

[32] E. Devaud. *Méthodologie de Prise en Compte de la Robustesse dans les Techniques de la Linéarisation*. PhD thesis, Université Paris-Sud XI/SUPELEC, France, 1999.

[33] E. Devaud, J.-P. Harcaut, and H. Siguerdidjane. Three-Axes Missile Autopilot Design: From Linear to Nonlinear Control Strategies. *Journal of Guidance, Control and Dynamics*, 24(1):64–71, 2001.

[34] S.M. Djouadi. Robustness in the Gap Metric and Coprime Factor Perturbations for LTV Systems. In *Proceedings of the* 46^{th} *IEEE Conference on Decision & Control*, pages 6400–6405, New Orleans, USA, 2007. IEEE.

[35] R. Dold and H. Buchek. Flight Test of a Scheduled μ-Synthesis Autopilot for an Air-to-Air Missile. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Montreal, Canada, 2001. AIAA.

[36] C. Döll, Y. Le Gorrec, G. Ferreres, and J.F. Magni. A Robust Self-scheduled Missile Autopilot: Design by Multi-model Eigenstructure Assignment. *Control Engineering Practice*, 9(10):1067–1078, 2001.

[37] R. Eberhardt and K.A. Wise. Automated Gain Schedules for Missile Autopilots Using Robustness Theory. In *Proceedings of the* 1^{st} *Conference on Control Applications*, pages 243–250, Dayton, USA, 1992. IEEE.

[38] A. El-Sakkary. The Gap Metric: Robustness of Stabilization of Feedback Systems. *IEEE Transactions on Automatic Control*, 30(3):240–247, 1985.

[39] A.K. El-Sakkary. *The Gap-metric for Unstable Systems*. PhD thesis, McGill University, Montreal, Canada, 1981.

[40] V. Fromion et G. Scorletti. A Theoretical Framework for Gain Scheduling. *International Journal of Robust and Nonlinear Control*, 13:951–982, 2003.

[41] D. Farret. *Embarquabilité et Synthèse de Correcteurs à Séquencement de Gains avec Contrainte d'Ordre-Application au Pilotage d'un Missile*. PhD thesis, Université Paris-Sud XI/SUPELEC, France, 2002.

[42] D. Farret, G. Duc, and J.-P. Harcaut. Multirate $\mathscr{H}_\infty$ Control: A Loop Shaping LMI-based Approach Application to the Control of a Missile. In *Proceedings of the 2001 European Control Conference*, Porto, Portugal, 2001.

[43] D. Farret, G. Duc, and J.-P. Harcaut. Reduced Order $\mathscr{H}_\infty$ Loop Shaping Control of a Missile Pitch Axis over a Wide Flight Envelope. *International Journal of Nonlinear Studies*, 10(2):199–214, 2004.

[44] J.-P. Friang. *Approche Générique $\mathscr{H}_\infty - \nu$ pour la Synthèse et l'Analyse d'un Pilote Automatique de Missile Soumis à une Aéroélasticité Importante*. PhD thesis, Université Paris-Sud XI/SUPELEC, France, 1996.

[45] J.-P. Friang, G. Duc, and J.-P. Bonnet. Robust Autopilot for a Flexible Missile: Loop Shaping $\mathscr{H}_\infty$ Design and Real ν Analysis. *International Journal of Robust and Nonlinear Control*, 8:129–153, 1998.

[46] V. Fromion. *Une Approche Incrémentale de la Robustesse Non-Linéaire: Application au Domaine Aéronautique*. PhD thesis, Université Paris-Sud XI/SUPELEC, France, 1995.

[47] V. Fromion, G. Scorletti, and G. Ferreres. Nonlinear Performance of a PI Controlled Missile: An Explanation. *International Journal of Robust and Nonlinear Control*, 9:485–518, 1999.

[48] P. Gahinet. Explicit Controller Formulas for LMI-based $\mathscr{H}_\infty$ Synthesis. *Automatica*, 32(7):1007–1014, 1996.

[49] T.T. Georgiou. On the Computation of the Gap Metric. *Systems & Control Letters*, 11:253–277, 1988.

[50] T.T. Georgiou and M. Smith. Optimal Robustness in the Gap Metric. *IEEE Transactions on Automatic Control*, 35(6):673–686, 1990.

[51] T.T. Georgiou and M. Smith. Robustness Analysis of Nonlinear Feedback Systems: An Input-output Approach. *IEEE Transactions on Automatic Control*, 42(9):1200–1221, 1997.

[52] L.E. Ghaoui, F. Oustry, and M. Aitrami. A Cone Complementarity Linearization Algorithm for Static Output Feedback and Related Problems. *IEEE Transactions on Automatic Control*, 42(8):1171–1176, 1997.

[53] K. Glover and J. Doyle. State-Space Formulae for all Stabilizing Controllers that Satisfy an $\mathscr{H}_\infty$ Norm Bound and Relations to Risk Sensitivity. *Systems & Control Letters*, 11:167–172, 1988.

[54] K.M. Grigoriadis and R.E. Skelton. Low-order Control Design for LMI Problems Using Alternating Projection Methods. *Automatica*, 32(8):1117–1125, 1996.

[55] G. Gu, J.R. Cloutier, and G. Kim. Gain-scheduled Missile Autopilot Design Using Observer-Based $\mathscr{H}_\infty$ Control. In *Proceedings of the* 14^{th} *American Control Conference*, pages 1951–1955, Seattle, USA, 1995. IEEE.

[56] S. Gutman and E.I. Jury. A General Theory for Matrix Root Clustering in Subregions of the Complex Plane. *IEEE Transactions on Automatic Control*, 26:853–863, 1981.

[57] W.M. Haddad and D.S. Bernstein. Controller Design with Regional Pole Constraints. *IEEE Transactions on Automatic Control*, 37(1):54–69, 1992.

[58] M.E. Hangstrup, J. Stoustrup, P. Andersen, and T.S. Pedersen. Gain-scheduled Control of a Fossil-fired Power Plant Boiler. In *Proceedings of the 1999 IEEE International Conference on Control Applications*, pages 905–909, Hawaii, USA, 1999. IEEE.

[59] R. Hanus, M. Kinnaert, and J.-L. Henrotte. Conditioning Technique, a General Anti-windup and Bumpless Transfer Method. *Automatica*, 23(6):729–739, 1987.

[60] B. Hencey and A.G. Alleyne. A Robust Controller Interpolation Design Technique. In *Proceedings of the* 26^{th} *American Control Conference*, pages 5347–5353, New York, USA, 2007. IEEE.

[61] A. Hiret. *Pilotage Robuste d'un Missile sur un Large Domaine de Vol : Synthèse et Analyse dans le Cadre $\mathscr{H}_\infty$ et LPV*. PhD thesis, Université Paris-Sud XI/SUPELEC, France, 1999.

[62] A. Hiret, G. Duc, and J.-P. Bonnet. The Application of Gain Scheduling $\mathscr{H}_\infty$ Controllers for a Missile Autopilot. In *Proceedings of the* 14^{th} *IFAC Symposium on Automatic Control in Aerospace*, page 59, Seoul, Korea, 1998. IFAC.

[63] A. Hiret, G. Duc, and J.-P. Friang. Self-scheduled $\mathscr{H}_\infty$ Loop Shaping Control of a Missile. In *Proceedings of the 1999 European Control Conference*, Karlsruhe, Germany, 1999.

[64] R.A. Hyde and K. Glover. VSTOL Aircraft Flight Control System Design Using $\mathscr{H}_\infty$ Controllers and a Switching Strategy. In *Proceedings of the* 29^{th} *IEEE Conference on Decision & Control*, pages 2975–2980, Honolulu, USA, 1990. IEEE.

[65] R.A. Hyde and K. Glover. The Application of Scheduled $\mathscr{H}_\infty$ Controllers to a VSTOL Aircraft. *IEEE Transactions on Automatic Control*, 38(7):1021–1039, 1993.

[66] P.A. Ioannou and J. Sun. *Robust Adaptive Control*. Prentice Hall, 1991.

[67] T. Iwasaki and G. Shibata. LPV System Analysis via Quadratic Separator for Uncertain Implicit Systems. *IEEE Transactions on Automatic Control*, 46(8):1195–1208, 2001.

[68] T. Iwasaki and R.E. Skelton. All Controllers for the General $\mathscr{H}_\infty$ Control Problem: LMI Existence Conditions and LMI Formulas. *Automatica*, 30(8):1307–1317, 1994.

[69] J. Jiang. Optimal Gain Scheduling Controller for a Diesel Engine. *IEEE Control Systems Magazine*, 14(4):42–48, 1994.

[70] B. Kadmiry and D. Driankov. A Fuzzy Flight Controller Combining Linguistic and Model-based Fuzzy Control. *Fuzzy Sets and Systems*, 146:313–347, 2004.

[71] I. Kaminer, A.M. Pascoal, P.P. Khargonekar, and E. Colemean. A Velocity Algorithm for the Implementation of Gain-scheduled Controllers. *Automatica*, 31(8):1185–1191, 1995.

[72] J. Kautski and N.K. Nichols. Robust Pole Assignment in Linear State Feedback. *International Journal of Control*, 41:1129–1155, 1985.

[73] J.H. Kelly and J.H. Evers. An Interpolation Strategy for Scheduling Dynamic Compensators. In *Proceedings of the 1997 AIAA Guidance, Navigation, and Control Conference*, pages 1682–1690, New Orleans, USA, 1997. AIAA.

[74] J.H. Kelly, J.H. Evers, R.A. Korn, and D.A. Lawrence. Scheduling Dynamic Compensators via Control Signal Interpolation. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 176–186, Portland, USA, 1999. AIAA.

[75] H.K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[76] J. Krause, M. Jackson, J. Cloutier, J. Evers, and R. Wilson. An Adaptive Autopilot for a Flexible Air-to-Air Missile. In *Proceedings of the* 30^{th} *Conference on Decision and Control*, pages 3002–3007, Brighton, UK, 1991. IEEE.

[77] J.-C. Wanner L. George, J.-F. Vernet. *La Mécanique du Vol: Performances des Avions et des Engins*. Dunod, 1969.

[78] M.-J. Lai and L.L Schumaker. *Spline Functions on Triangulations*. Cambridge University Press, 2007.

[79] D.A. Lawrence and W.J. Rugh. Gain Scheduling Dynamic Linear Controllers for a Nonlinear Plant. *Automatica*, 31(3):381–390, 1995.

[80] D.A. Lawrens. Gain-scheduled Controllers with Guaranteed Linearization Properties. In *Proceedings of the* 20^{th} *American Control Conference*, pages 4128–4133, Arlington, USA, 2001. IEEE.

[81] D.A. Lawrens, J.H. Kelly, and J.H. Evers. Gain-scheduled Missile Autopilot Design Using a Control Signal Interpolation Technique. In *Proceedings of the 1998 AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1394–1402, Boston, USA, 1998. AIAA.

[82] D.J. Leith and W.E. Leithead. Appropriate Realization of Gain-scheduled Controllers with Application to Wind Turbine Regulation. *International Journal of Control*, 65(2):223–248, 1996.

[83] D.J. Leith and W.E. Leithead. Appropriate Realization of MIMO Gain-scheduled Controllers. *International Journal of Control*, 70(1):13–50, 1998.

[84] D.J. Leith and W.E. Leithead. Comments on 'Gain Scheduling Dynamic Linear Controllers for a Nonlinear Plant'. *Automatica*, 34(8):1041–1043, 1998.

[85] D.J. Leith and W.E. Leithead. Gain-scheduled and Nonlinear Systems: Dynamic Analysis by Velocity-based Linearization Families. *International Journal of Control*, 70(2):289–317, 1998.

[86] D.J. Leith and W.E. Leithead. Gain-scheduled Controller Design: An Analytic Framework Directly Incorporating Non-equilibrium Plant Dynamics. *International Journal of Control*, 70(2):249–269, 1998.

[87] D.J. Leith and W.E. Leithead. Analytic Framework for Blended Multiple Model Systems Using Linear Local Models. *International Journal of Control*, 72(7):605–619, 1999.

[88] D.J. Leith and W.E. Leithead. Survey of Gain Scheduling Analysis and Design. *International Journal of Control*, 73(11):1001–1025, 2000.

[89] D.J. Leith and W.E. Leithead. Gain-scheduled Control of a Skid-to-turn Missile: Relaxing Slow Variation Requirements by Velocity-based Design. In *Proceedings of the* 20^{th} *American Control Conference*, pages 500–505, Arlington, USA, 2001. IEEE.

[90] D.J. Leith, A. Tsourdos, B.A. White, and W.E. Leithead. Application of Velocity-based Gain Scheduling to Lateral Autopilot Design for an Agile Missile. *Control Engineering Practice*, 9:1079–1093, 2001.

[91] A. Leonessa, W.M. Haddad, and V. Chellaboina. Nonlinear System Stabilization via Hierarchical Switching Control. *IEEE Transactions on Automatic Control*, 46(1):17–28, 2001.

[92] Z. Lin and M. Khammash. Robust Gain-scheduled Aircraft Longitudinal Controller Design Using an $\mathscr{H}_\infty$ Approach. In *Proceedings of the* 20^{th} *American Control Conference*, pages 2724–2729, Arlington, USA, 2001. IEEE.

[93] D. Limebeer M. Green. *Linear Robust Control*. Prentice-Hall, 1996.

[94] J.-F. Magni. An LFT Approach to Robust Gain Scheduling. In *Proceedings of the* 44^{th} *IEEE Conference on Decision & Control*, pages 7971–7976, Seville, Spain, 2005.

[95] M.W. McConley, B.D. Appleby, M.A. Dahleh, and E. Feron. A Computationally Efficient Lyapunov-based Scheduling Procedure for Control of Nonlinear Systems with Stability Guaranties. *IEEE Transactions on Automatic Control*, 45(1):33–49, 2000.

[96] D. McFarlane and K. Glover. *Robust Controller Design using Normalized Coprime Factor Plant Descriptions*. Springer-Verlag, 1989.

[97] D. McFarlane and K. Glover. Robust Stabilization of Normalized Coprime Factor Descriptions with $\mathscr{H}_\infty$-bounded Uncertainty. *IEEE Transactions on Automatic Control*, 34(8):821–830, 1989.

[98] D. McFarlane and K. Glover. A Loop Shaping Design Procedure Using $\mathscr{H}_\infty$ Synthesis. *IEEE Transactions on Automatic Control*, 37(6):759–769, 1992.

[99] A.R. Mehrabian and J. Roshanian. Design of Gain-scheduled Autopilot for a Highly Agile Missile. In *Proceedings of the* 1^{st} *International Symposium on System and Control in Aerospace and Astronautics*, pages 144–149, Harbin, China, 2006. IEEE.

[100] T. Meister, H. Werner, G. Lohoefer, D.M. Herlach, and H. Unbehauen. Gain-scheduled Control of an Electrostatic Levitator. *Control Engineering Practice*, 11:117–128, 2003.

[101] M.C. Mickle and J.J. Zhu. Skid to Turn Control of the APKWS Missile Using Trajectory Linearization Technique. In *Proceedings of the* 20^{th} *American Control Conference*, pages 3346–3351, Arlington, USA, 2001. IEEE.

[102] A.M. Mohamed, I.M.M. Hassan, and A.M.K. Hashem. Elimination of Imbalance Vibrations in Magnetic Bearing Systems Using Discrete Time Gain-scheduled $\mathcal{Q}$-parametrization Controllers. In *Proceedings of the 1999 Conference on Control Applications*, pages 737–742, Hawaii, USA, 1999. IEEE.

[103] R.A. Nichols, R.T. Reichert, and W.J. Rugh. Gain Scheduling for $\mathscr{H}_\infty$ Controllers: A Flight Control Example. *IEEE Transactions on Control Systems Technology*, 1(2):69–79, 1993.

[104] H. Niemann and J. Stoustrup. Gain Scheduling using the Youla Parameterization. In *Proceedings of the* 38^{th} *IEEE Conference on Decision & Control*, pages 2306–2311, Phoenix, USA, 1999.

[105] K. Ogata. *Modern Control Engineering*. Prentice-Hall, 1997.

[106] S. Oh-Hara, K. Hirata, and Y. Ohta. A Switching Gain-scheduled Approach for Wafer Transfer Robots and Its Experimental Evaluations. In *Proceedings of the* 25^{th} *American Control Conference*, pages 3898–3903, Minneapolis, USA, 2006. IEEE.

[107] M. Oosterom and R. Babuška. Design of a Gain Scheduling Mechanism for Flight Control Laws by Fuzzy Clustering. *Control Engineering Practice*, 14:769–781, 2005.

[108] B. Paijmans, W. Symens, H. Van Brussel, and J. Swevers. A Gain Scheduling Control Technique for Mechatronic Systems with Position-dependent Dynamics. In *Proceedings of the* 25^{th} *American Control Conference*, pages 2933–2938, Minneapolis, USA, 2006. IEEE.

[109] S. Patra, S. Sen, and G. Ray. Design of Static $\mathscr{H}_\infty$ Loop Shaping Controller in Four-block Framework Using LMI Approach. *Automatica*, 2008. (to appear).

[110] P.C. Pellanda, P. Apkarian, and D. Alazard. Gain Scheduling Through Continuation of Observer-based Realizations: Application to $\mathscr{H}_\infty$ and μ Controllers. In *Proceedings of the* 39^{th} *IEEE Conference on Decision & Control*, pages 2787–2792, Sydney, Australia, 2000.

[111] E. Prempain and I. Postlethwaite. Static $\mathscr{H}_\infty$ Loop Shaping Control of a Fly-by-wire Helicopter. *Automatica*, 41:1517–1528, 2005.

[112] R.T. Reichert. Dynamic Scheduling of Modern Robust Control Autopilot Designs for Missiles. *IEEE Control Systems Magazine*, 12(5):35–42, 1992.

[113] W.J. Rugh. Analytical Framework for Gain Scheduling. *IEEE Control Systems Magazine*, 11(1):79–84, 1991.

[114] W.J. Rugh and J.S. Shamma. Research on Gain Scheduling. *Automatica*, 36:1401–1425, 2000.

[115] C.S. Scherer, P. Gahinet, and M. Chilali. Multiobjective Output-feedback Control via LMI Optimization. *IEEE Transactions on Automatic Control*, 42(7):896–911, 1997.

[116] C. Schumacher and P.P. Khargonekar. A Comparison of Missile Autopilot Designs using $\mathscr{H}_\infty$ Control with Gain Scheduling and Nonlinear Dynamic Inversion. In *Proceedings of the* 16^{th} *American Control Conference*, pages 2759–2763, Albuquerque, USA, 1997. IEEE.

[117] G. Scorletti and L. El-Ghaoui. Improved LMI Conditions for Gain Scheduling and Related Control Problems. *International Journal of Robust and Nonlinear Control*, 8:845–877, 1998.

[118] S.M. Shahruz and S. Behtash. Design of Controllers for Linear Parameter Varying Systems by the Gain Scheduling Technique. *Journal of Mathematical Analysis and Applications*, 168:195–217, 1992.

[119] J. Shamma. *Analysis and Design of Gain-scheduled Control Systems*. PhD thesis, LIDS, M.I.T., USA, 1988.

[120] J.S. Shamma and M. Athans. Analysis of Gain-scheduled Control for Nonlinear Plants. *IEEE Transactions on Automatic Control*, 35(8):898–907, 1990.

[121] J.S. Shamma and M. Athans. Guaranteed Properties of Gain-scheduled Control for Linear Parameter Varying Plants. *Automatica*, 27(3):559–564, 1991.

[122] J.S. Shamma and M. Athans. Gain Scheduling: Potential Hazards and Possible Remedies. *IEEE Control Systems Magazine*, 12(3):101–107, 1992.

[123] J.S. Shamma and J.R. Cloutier. Trajectory Scheduled Missile Autopilot Design. In *Proceedings of the* 1^{st} *Conference on Control Applications*, pages 237–242, Dayton, USA, 1992. IEEE.

[124] J.S. Shamma and J.R. Cloutier. Gain-scheduled Autopilot Design Using Linear Parameter Varying Transformations. *Journal of Guidance, Control and Dynamics*, 16(2):256–263, 1993.

[125] T. Shimomura. Hybrid Control of Gain Scheduling and Switching: A Design Example of Aircraft Control. In *Proceedings of the* 22^{th} *American Control Conference*, pages 4639–4644, Denver, USA, 2003. IEEE.

[126] T. Shimomura, M. Takahashi, and T. Fujii. Extended-space Control Design with Parameter-dependent Lyapunov Functions. In *Proceedings of the* 40^{th} *IEEE Conference on Decision & Control*, pages 2157–2162, Orlando, USA, 2001.

[127] C.E. Souza and A. Trofino. Gain-scheduled $\mathscr{H}_2$ Controller Synthesis for Linear Parameter Varying Systems via Parameter-dependent Lyapunov Functions. *International Journal of Robust and Nonlinear Control*, 16:243–257, 2006.

[128] D.J. Stilwell. $\mathcal{J}$-$\mathcal{Q}$ Interpolation for Gain-scheduled Controllers. In *Proceedings of the* 38^{th} *IEEE Conference on Decision & Control*, pages 749–754, Phoenix, USA, 1999.

[129] D.J. Stilwell. State Space Interpolation for a Gain-scheduled Autopilot. *Journal of Guidance, Control & Dynamics*, 24(3):460–465, 2001.

[130] D.J. Stilwell and D.A. Lawrence. Sampled-data Implementation of a Gain-scheduled Controller. *International Journal of Robust and Nonlinear Control*, 12:855–868, 2002.

[131] D.J. Stilwell and W.J. Rugh. Interpolation Methods for Gain Scheduling. In *Proceedings of the* 37^{th} *IEEE Conference on Decision & Control*, pages 3003–3008, Tampa, USA, 1998.

[132] D.J. Stilwell and W.J. Rugh. Interpolation of Observer State Feedback Controllers for Gain Scheduling. *IEEE Transactions on Automatic Control*, 44(6):1225–1229, 1999.

[133] D.J. Stilwell and W.J. Rugh. Stability-preserving Interpolation Methods for the Synthesis of Gain-scheduled Controllers. *Automatica*, 36:665–671, 2000.

[134] F. Takemori and Y. Okuyama. Improvement of Transient Performance for a SIMO System by Gain-scheduled $\mathcal{Q}$-parameterization. In *Proceedings of the 2000 Conference on Control Applications*, pages 856–861, Anchorage, USA, 2000. IEEE.

[135] S. Tan, C.-C. Hang, and J.-S. Chai. Gain Scheduling: From Conventional to Neuro-fuzzy. *Automatica*, 33(3):411–419, 1997.

[136] W. Tan, A.K. Packard, and G.J. Balas. Quasi-LPV Modeling and LPV Control of a Generic Missile. In *Proceedings of the* 19^{th} *American Control Conference*, pages 3692–3696, Chicago, USA, 2000. IEEE.

[137] S. Theodoulis and G. Duc. Gain Scheduling Techniques for Missile Autopilot Synthesis. In *Proceedings of the 2007 European Control Conference*, pages 2258–2265, Kos, Greece, 2007. EUCA.

[138] S. Theodoulis and G. Duc. Static Interpolated $\mathscr{H}_\infty$ Loop Shaping Controllers for Missile Autopilot Synthesis. In *Proceedings of the* 46^{th} *IEEE Conference on Decision & Control*, pages 2385–2392, New Orleans, USA, 2007. IEEE.

[139] S. Theodoulis and G. Duc. Gain-scheduled Autopilot Synthesis for an Atmosphere Re-entry Vehicle. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Honolulu, USA, 2008. AIAA.

[140] S. Theodoulis and G. Duc. Missile Autopilot Design: Gain Scheduling and the Gap Metric. *Journal of Guidance, Control and Dynamics*, 2008. (to appear).

[141] A. Tsourdos, E.J. Hughes, and B.A. White. Fuzzy Multi-objective Design for a Lateral Missile Autopilot. *Control Engineering Practice*, 14(5):547–561, 2006.

[142] M. Vidyasagar. *Nonlinear System Analysis*. Prentice-Hall, 1978.

[143] M. Vidyasagar. The Graph Metric for Unstable Plants and Robustness Estimates for Feedback Stability. *IEEE Transactions on Automatic Control*, 29(5):403–418, 1984.

[144] M. Vidyasagar and H. Kimura. Robust Controllers for uncertain Linear Multivariable Systems. *Automatica*, 22(1):85–94, 1986.

[145] G. Vinnicombe. *Measuring the Robustness of Feedback Systems*. PhD thesis, University of Cambridge, UK, 1993.

[146] G. Vinnicombe. The Graph Metric for Unstable Plants and Robustness Estimates for Feedback Stability. *IEEE Transactions on Automatic Control*, 38(9):1371–1383, 1993.

[147] O. Voinot, D. Alazard, and B. Clément. Unstationnary Control of a Launcher Using Observer-based Structures. In *Proceedings of the* 22^{th} *American Control Conference*, pages 3466–3471, Denver, USA, 2003. IEEE.

[148] F. Wang and V. Balakrishnan. Improved Stability Analysis and Gain-scheduled Controller Synthesis for Parameter-dependent Systems. *IEEE Transactions on Automatic Control*, 47(5):720–734, 2002.

[149] J. Wang and W. Rugh. Feedback Linearization Familes for Nonlinear Systems. *IEEE Transactions on Automatic Control*, 32(10):935–940, 1987.

[150] D.P. White, J.G. Wozniak, and D.A. Lawrence. Missile Autopilot Design Using a Gain Scheduling Technique. In *Proceedings of the* 26^th *Southeastern Symposium on Systems Theory*, pages 606–610, Athens, USA, 1994. IEEE.

[151] F. Wu. An Unified Framework for LPV System Analysis and Control Synthesis. In *Proceedings of the* 39^th *IEEE Conference on Decision & Control*, pages 4578–4583, Sydney, Australia, 2000. IEEE.

[152] F. Wu. Switching LPV Control Design for Magnetic Bearing Systems. In *Proceedings of the 2001 IEEE International Conference on Control Applications*, pages 41–46, Mexico City, Mexico, 2001. IEEE.

[153] F. Wu and K. Dong. Gain Scheduling Control of LFT Systems Using Parameter-dependent Lyapunov Functions. *Automatica*, 42:39–50, 2006.

[154] Z. Yu, H. Chen, and P.-Y. Woo. Reduced Order Gain-scheduled Output/State Feedback Control Based on $\mathscr{H}_\infty$ Performance Preserved Interpolation. *IEEE Transactions on Control Systems Technology*, 13(4):670–681, 2005.

[155] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, 1996.

Résumé de Thèse.

Cette thèse porte sur le problème de commande des systèmes non-linéaires à paramètres variants rencontrés souvent (mais non seulement) dans le domaine aéronautique, avec la technique de séquencement de gains par linéarisation.

Une stratégie innovante, appelée *extended* - Loop Shaping Design Procedure (*e*-LSDP), qui facilite et systématise la tache du scientifique pour le calcul d'une loi de commande séquencée pour ce type de systèmes, est ici proposée.

Cette stratégie est basée sur une pré-compensation (loop shaping) faite à partir des systèmes linéarisés du système non-linéaire autour d'un petit nombre de points de fonctionnement en utilisant des compensateurs de structure simple (e.g. PID), et de plus en utilisant une compensation additionnelle/corrective type retour de sortie $\mathscr{H}_\infty$ statique. Les points de fonctionnement de la deuxième compensation sont calculés à l'aide d'un algorithme de choix de points de synthèse basé sur la connexion des théories de la gap métrique et de la commande $\mathscr{H}_\infty$ par loop shaping. La loi de commande globale non-linéaire séquencée est finalement obtenue en utilisant une interpolation de tous les gains des compensateurs impliqués pendant la phase de synthèse.

La méthode proposée ici est validée sur deux exemples d'application : le pilotage autour de l'axe de tangage d'un missile fortement manœuvrant et d'un véhicule de rentrée atmosphérique. Les deux autopilotes sont testés de façon intensive en utilisant des simulations non-linéaires, une analyse Monte Carlo et linéaire à temps figé afin de démontrer leurs excellents caractéristiques en termes de performance et de robustesse.

Mots-clés : Séquencement de gains, $\mathscr{H}_\infty$ loop shaping, gap métrique, systèmes nonlinéaires & LPV, linéarisation, analyse de robustesse, LMI, autopilotes.

Thesis Abstract.

This thesis deals with the problem of linearization-based gain scheduling control for nonlinear parameter-dependent systems often (but not exclusively) encountered in the aeronautical domain.

A novel control strategy, the *extended* - Loop Shaping Design Procedure (*e*-LSDP), that facilitates and systematizes the designer's task for the calculation of gain-scheduled control laws for such systems, is here proposed.

This strategy is based on a pre-compensation (loop shaping) of linearized instances of the nonlinear system at a small set of synthesis points using simple structure (e.g. PID) controllers and on an additional/corrective compensation at a second set of points by static $\mathscr{H}_\infty$ output feedback controllers. The latter compensation points are computed using an operating point selection algorithm based on the connection of the gap metric and standard $\mathscr{H}_\infty$ loop-shaping theories. The final global nonlinear gain-scheduled control law is finally obtained using interpolation of all controller gains involved in the synthesis phase.

The proposed method is validated on two pitch axis autopilot problems : a highly manoeuvrable missile and an atmosphere re-entry model. Both autopilots are extensively tested using nonlinear simulations, Monte Carlo and frozen time-type linear analysis in order to prove their excellent characteristics in terms of both performance and robustness.

Keywords : Gain scheduling, $\mathscr{H}_\infty$ loop shaping, gap metric, nonlinear & LPV systems, linearization, robustness analysis, LMI, autopilots.