



HAL
open science

Modeling, Identification and Control of a Guided Projectile in a Wind Tunnel

Guillaume Strub

► **To cite this version:**

Guillaume Strub. Modeling, Identification and Control of a Guided Projectile in a Wind Tunnel. Other. Université de Haute Alsace - Mulhouse, 2016. English. NNT : 2016MULH8492 . tel-01559031

HAL Id: tel-01559031

<https://theses.hal.science/tel-01559031>

Submitted on 10 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modeling, Identification and Control of a Guided Projectile in a Wind Tunnel

Modélisation, identification et commande d'un projectile guidé en soufflerie

THÈSE

présentée et soutenue publiquement le 20 juillet 2016

pour l'obtention du

Doctorat de l'Université de Haute-Alsace

(mention automatique)

par

Guillaume STRUB

Composition du jury

<i>Président :</i>	JEAN-MARC BIANNIC	Professeur, ISAE, ONERA, Toulouse
<i>Rapporteurs :</i>	LUC DUGARD FRANCK PLESTAN	Directeur de Recherche CNRS, GIPSA-Lab, Grenoble Professeur, Ecole Centrale de Nantes
<i>Directeur :</i>	MICHEL BASSET	Professeur, ENSISA, Mulhouse
<i>Encadrants :</i>	SIMONA DOBRE SPILIOS THEODOULIS	Chercheuse, ISL, Saint-Louis Chercheur, ISL, Saint-Louis
<i>Invité :</i>	VINCENT GASSMANN	Chercheur, ISL, Saint-Louis

Remerciements

Ce travail est le fruit d'une collaboration entre l'Institut franco-allemand de recherches de Saint-Louis (ISL) et le laboratoire Modélisation, Intelligence, Processus et Systèmes (MIPS EA2332) de l'université de Haute-Alsace, avec le soutien financier de la Direction Générale de l'Armement (DGA).

Je tiens tout d'abord à remercier les membres du jury, qui m'ont fait l'honneur de participer à l'évaluation de ce travail. Je remercie Luc Dugard, Directeur de recherches CNRS à l'INP Grenoble et Franck Plestan, professeur à l'École Centrale de Nantes, d'avoir accepté d'examiner mon travail en tant que rapporteurs. J'adresse mes remerciements à Jean-Marc Biannic, professeur à l'ISAE/SUPAERO et chercheur à l'ONERA pour avoir présidé ce jury et pour l'intérêt qu'il a porté à ces travaux.

Je tiens à exprimer ma gratitude envers mon directeur de thèse, Michel Basset, professeur à l'ENSISA, pour m'avoir accueilli dans son équipe de recherche et m'avoir guidé au cours de cette aventure, bien que je n'aie été que trop peu présent sur place. J'adresse un très grand merci à mes encadrants de l'Institut franco-allemand de recherches de Saint-Louis : Simona Dobre, du groupe ABX, Vincent Gassmann et Spilios Theodoulis, du groupe GNC. Merci à vous tous pour votre disponibilité, le temps que vous m'avez consacré, les discussions enrichissantes et votre soutien tout au long de ces travaux.

J'aimerais également remercier l'ensemble des personnes ayant participé à la réalisation de ce projet, et notamment sur la réalisation de la partie expérimentale. Je remercie Denis Bidino et Kevin Méder pour la conception et la mise en plans des différents éléments mécaniques, et l'ensemble du personnel de l'atelier central pour la réalisation de ces pièces. Je souhaite également remercier Joseph Juncker, Emmanuel Pêcheur et Emmanuel Roussel pour leurs conseils lors de la conception de l'électronique, ainsi que Christophe Bissel pour la gravure des cartes. Merci également à Corentin Chauffaut de l'Heudiasyc pour le code source des drivers temps réel du port série et I²C.

Je remercie les permanents, thésards et stagiaires de l'ISL avec qui j'ai partagé de bons moments à midi, autour d'un café ou en terrasse : Nicolas, Pierre, Maxime, Thomas, Florian, Sebastian, Nathalie, Cédric, Bastien, Christophe, Dominique, Daniel, Erwann, Axel, Julien, Sylvain. Un grand merci à Marie pour ses conseils et son soutien moral dans la dernière ligne droite de la thèse ! Merci aussi aux grimpeurs de l'ISL, Alexander, Michel, Carole, Arnaud, Robert, pour les séances d'escalade au COSEC ou en falaise qui m'ont permis de décompresser pendant la rédaction. J'aimerais aussi remercier l'équipe MIAM pour leur accueil et leur bonne humeur, même si je n'étais pas présent souvent. Merci à Jean-Philippe, Benjamin, Abderazik, Rodolfo, Joël, Jonathan, Jérémie, aux thésards, Jean-Nicola, Sabra, Mohamed, Florent, Rachid, Olivier, et aussi à Luc et Aurélia, pour les moments de convivialité au RU et autour d'un café. Mention spéciale à Thomas, collègue de stage puis de thèse, avec qui j'ai partagé les mêmes moments de galère et d'innombrables projectiles en mousse dans la salle Sun !

Enfin, je ne remercierai jamais assez mes parents et ma famille pour leur éternel soutien et leur investissement tout au long de mes études, dont cette thèse constitue le dernier chapitre.

Contents

Remerciements	i
Conferences and Publications	vii
Résumé de thèse	ix
Nomenclature	xxi
Introduction	1

Chapter 1

Experimental Setup

1.1	Introduction	11
1.2	The ACHILES Test Setup	13
1.3	Hardware Components	14
1.3.1	Mechanical Structure and Aerodynamics	14
1.3.2	Actuators	15
1.3.3	Sensors	21
1.3.4	Embedded Computer and Support Electronics	27
1.3.5	Wind Tunnel	29
1.3.6	Support Structure	29
1.4	Software Environment	31
1.4.1	Embedded Software	31
1.4.2	Development Environment	35
1.5	Concluding Remarks	37

Chapter 2**Modeling and Identification**

2.1	Introduction	40
2.2	Flight Mechanics of a Free-Flying Projectile	41
2.2.1	Frames and Coordinate Systems	41
2.2.2	Free-Flight 6-Degrees of Freedom Model	43
2.2.3	Aerodynamic Forces and Moments	47
2.3	Flight Mechanics of a Translation-Denied Projectile	49
2.3.1	Dynamic Model of the Translation-Denied Projectile	50
2.3.2	Airframe incidence angles	52
2.3.3	Roll-Locked Case	52
2.3.4	Roll/Yaw-Locked Case	54
2.4	Trimming and Linearization	55
2.4.1	Trimming	55
2.4.2	Linearization	58
2.5	System Identification	60
2.5.1	Identification Procedure	60
2.5.2	Model Postulation	61
2.5.3	<i>A priori</i> Identifiability	62
2.5.4	<i>A posteriori</i> Identifiability	63
2.5.5	Experiment Design	65
2.5.6	Experimental Data Collection	67
2.5.7	Parameter Estimation and Uncertainty Analysis	68
2.6	Parameter Estimation Results	69
2.6.1	Overall Trends	70
2.6.2	Model Fit Evaluation	72
2.6.3	Uncertainty Evaluation	73
2.6.4	Constant-Airspeed Trends	73

2.6.5	Comparison with PRODAS-derived Aerodynamic Coefficients	77
2.6.6	Pitch-only case	78
2.7	Conclusion	79

Chapter 3 Control Design

3.1	Introduction	82
3.2	Design and Analysis Tools	84
3.2.1	Nominal Model Selection	84
3.2.2	Uncertainty Modeling	85
3.2.3	Standard \mathcal{H}_∞ Control	87
3.2.4	Multi-objective \mathcal{H}_∞ Control	89
3.2.5	Robustness Analysis	91
3.3	Fixed-Airspeed Autopilot	94
3.3.1	Synthesis Model	94
3.3.2	Autopilot Design	100
3.3.3	Robustness Analysis	108
3.3.4	Experimental Validation	110
3.3.5	Concluding Remarks	116
3.4	Variable-Airspeed Autopilot	117
3.4.1	Gain-Scheduling Strategy	117
3.4.2	Local Controller Synthesis	120
3.4.3	Controller Implementation and Experimental Validation	131
3.4.4	Concluding Remarks	134
3.5	Conclusion	134

Conclusion and Perspectives **137**

Bibliography **141**

Appendix A**Modeling & Identification Complements**

A.1	Attitude equations of motion	151
A.1.1	Roll Angular Dynamics	152
A.1.2	Pitch Angular Dynamics	152
A.1.3	Yaw Angular Dynamics	153
A.2	Inverse Trim Map Interpolation	155
A.3	Controllability and Observability of the Linearized Pitch-Yaw Model	155
A.4	<i>A priori</i> Identifiability of the Model Structure	156
A.5	Sensitivity Functions for a Linear System	157

Appendix B**Pitch-Only Autopilot**

B.1	Synthesis Model	160
B.1.1	Nominal Model Selection	160
B.1.2	Uncertainty Modeling	160
B.2	Autopilot Structure	163
B.2.1	Rate Damping Controller	163
B.2.2	Disturbance Rejection Controller	164
B.2.3	Reference Tracking Controller	167
B.2.4	Multi-objective Synthesis	168
B.3	Controller Synthesis Results	169
B.3.1	Disturbance Rejection Controller	170
B.3.2	Reference Tracking	171
B.4	Robustness Analysis	172
B.5	Experimental Validation	174
B.5.1	Disturbance Rejection	175
B.5.2	Reference Tracking	175
B.6	Concluding Remarks	177

Conferences and Publications

Articles in International Peer-Reviewed Journals

STRUB, G., THEODOULIS, S., GASSMANN, V., DOBRE, S., AND BASSET, M. Skid-to-Turn Autopilot Design and Validation for an Experimental Guided Projectile Prototype. Submitted to *Journal of Guidance, Control, and Dynamics*.

STRUB, G., DOBRE, S., GASSMANN, V., THEODOULIS, S., AND BASSET, M. Pitch Axis Identification for a Guided Projectile using a Wind Tunnel-based Experimental Setup. *IEEE/ASME Transactions on Mechatronics*, 21(3):1357–1365, 2016. doi:10.1109/tmech.2016.2525719.

STRUB, G., THEODOULIS, S., GASSMANN, V., DOBRE, S., AND BASSET, M. Pitch Axis Control for a Guided Projectile in a Wind Tunnel Hardware-in-the-Loop Setup. *Journal of Spacecraft and Rockets*, 52(6):1614–1626, 2015. doi:10.2514/1.A33330.

Papers in International Peer-Reviewed Conferences with Proceedings

STRUB, G. AND BASSET, M. Skid-to-Turn Autopilot Design and Validation for an Experimental Guided Projectile Prototype. In *AIAA Guidance, Navigation and Control Conference*. San Diego, CA, 2016. doi:10.2514/6.2016-1875. GNC Graduate Student Paper Competition finalist.

STRUB, G., THEODOULIS, S., GASSMANN, V., DOBRE, S., AND BASSET, M. Pitch Axis Control for a Guided Projectile in a Wind Tunnel-based Hardware-In-the-Loop Setup. In *AIAA Modeling and Simulation Technologies Conference*. Kissimmee, FL, 2015. doi:10.2514/6.2015-0153.

STRUB, G., GASSMANN, V., THEODOULIS, S., DOBRE, S., AND BASSET, M. Hardware-In-the-Loop Experimental Setup Development for a Guided Projectile in a Wind Tunnel. In *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 458–463. Besançon, France, 2014. doi:10.1109/AIM.2014.6878120.

Oral Presentations in National and International Conferences without Proceedings

STRUB, G., GASSMANN, V., THEODOULIS, S., DOBRE, S., AND BASSET, M. Pitch Axis Control for a Guided Projectile in a Wind Tunnel-based Hardware-In-the-Loop Setup. In *ISL Scientific Symposium*. Saint-Louis, France, 2015.

STRUB, G., GASSMANN, V., THEODOULIS, S., DOBRE, S., AND BASSET, M. Pitch Axis Control for a Guided Projectile in a Wind Tunnel-based Hardware-In-the-Loop Setup. In *Séminaire GT MOSAR*. Saint-Louis, France, 2015.

STRUB, G., BASSET, M., DOBRE, S., GASSMANN, V., AND THEODOULIS, S. Contribution à l'identification et à la commande de systèmes non linéaires à paramètres variants. Application à un projectile guidé en soufflerie. In *5èmes Journées Doctorales/Journées Nationales MACS*. Strasbourg, France, 2013.

Résumé de thèse

L'amélioration de la précision des munitions d'artillerie est une problématique faisant l'objet d'un certain nombre de travaux de recherche d'initiative privées et publiques. Les projectiles balistiques se caractérisent traditionnellement par une faible précision liée aux incertitudes et perturbations extérieures, telles que les conditions de tir et les rafales de vent. La dispersion élevée qui en résulte limite l'emploi de ces munitions dans les scénarios de conflit modernes, qui interviennent de plus en plus souvent en terrain complexe ou urbain. Ces études visent à améliorer la précision du point d'impact afin de réduire les dommages collatéraux et les coûts de traitement associés, tout en maintenant un faible coût unitaire. Dans ce contexte, les projectiles guidés constituent une réponse possible à cette problématique. L'amélioration de la précision est obtenue grâce à l'ajout de fonctionnalités de guidage, navigation et contrôle (GN&C) inspirées des systèmes de missiles et illustrées à la Figure 1. Cette forme générique est composée de deux boucles imbriquées. La boucle extérieure, dite de guidage, génère les consignes permettant de mener à bien la mission au moyen des données issues de la solution de navigation. La boucle interne est une boucle de régulation et d'asservissement, ayant pour objectif de piloter le projectile afin que les consignes issues de la boucle de guidage soient effectivement réalisées. C'est sur la mise au point de cette dernière que portent les présents travaux de thèse.

La synthèse des lois de pilotage est une tâche complexe. Les outils de l'Automatique moderne issus de la commande robuste nécessitent la connaissance d'un modèle mathématique du système étudié, afin de synthétiser les lois de commande et d'en étudier la stabilité et la robustesse, notamment vis-à-vis des incertitudes liées à la modélisation. Dans le cas des projectiles guidés, la modélisation est le plus souvent réalisée au moyen des outils de la mécanique du vol, où les forces et moments subis par le projectile sont décrits par des décompositions en coefficients aérodynamiques. Les modèles ainsi obtenus sont typiquement non-linéaires et leurs paramètres sont dépendant des conditions de vol (pression dynamique, nombre de Mach, angles d'incidence, etc.).

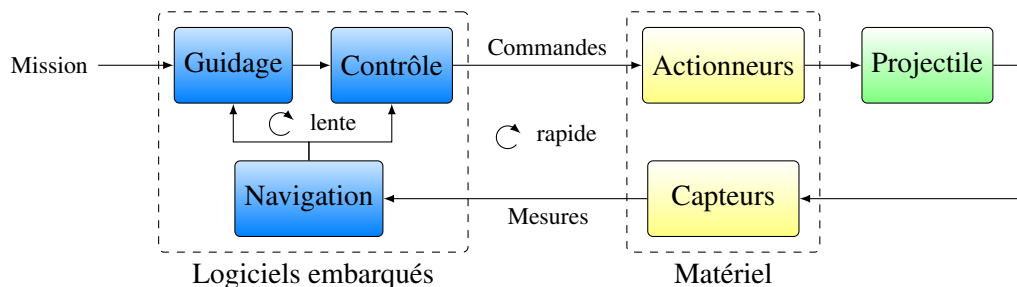


FIGURE 1: Boucle de guidage, navigation et contrôle (GNC).

Dans les processus de développement actuels, la détermination des coefficients aérodynamiques est une tâche complexe, faisant appel à différentes techniques complémentaires : mesures en soufflerie, essais en vol libre, codes de calcul aérodynamiques type CFD (*Computational Fluid Dynamics*), codes de calcul semi-empiriques. Cette dernière technique permet d'obtenir un résultat rapidement mais de faible précision, tandis que les méthodes CFD, plus précises, sont coûteuses en temps de calcul. Enfin, les mesures en vol ou en soufflerie requièrent de longues campagnes d'essais et font appel à du matériel coûteux. La validation expérimentale des lois de commande préalable aux essais en vol réel se fait le plus souvent au moyen de tables tournantes, dispositifs onéreux et ne reproduisant qu'imparfaitement la dynamique du vol. Ces étapes, nécessaires au développement de lois de commande pour les projectiles guidés, sont lourdes à mettre en œuvre tant en temps qu'en coût. Cette thèse se propose ainsi de mettre au point une méthodologie novatrice, bas-coût et plus rapide pour la mise au point et la validation desdites lois lors de phases d'avant-projet. L'approche envisagée consiste à placer un prototype fonctionnel de projectile guidé dans une soufflerie au moyen d'un support à rotule, permettant la rotation de la maquette dans tous les axes (cf. Figure 2). Ce banc d'essais est par la suite exploité afin de construire un modèle de projectile au moyen de données recueillies au cours d'essais d'identification, puis dans un second temps afin de valider les lois de commande développées.

Présentation du dispositif expérimental

Dans ce travail, nous nous intéressons à l'évaluation de cette méthodologie sur un cas pratique. L'application présentée consiste à développer des autopilotes de type *skid-to-turn* afin de commander la dynamique en tangage et en lacet d'un projectile. Pour ce faire, un dispositif expérimental complet a été entièrement développé durant ces travaux et mis à contribution pour l'application de la méthodologie. Celui-ci est basé sur un prototype de projectile empenné de calibre 80 mm, muni de quatre surfaces de contrôle au niveau de son nez. Cette configuration présente une symétrie de rotation, pour laquelle les approches *skid-to-turn* sont particulièrement adaptées. Ce prototype est placé dans une soufflerie subsonique par le biais du support à rotule illustré à la Figure 2.

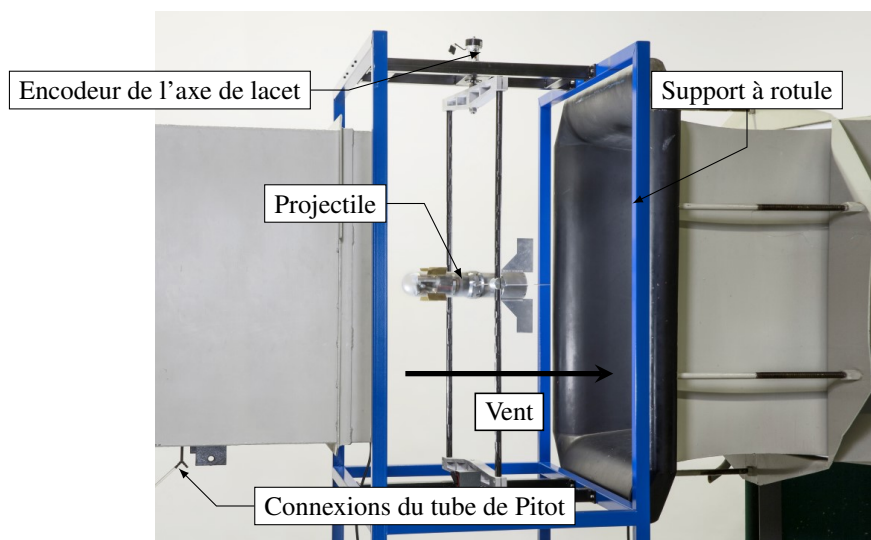


FIGURE 2: Vue d'ensemble du dispositif expérimental développé au cours de ce travail.

Une première phase de travaux a porté sur la conception et la réalisation du banc d'essais. Les critères de choix et les solutions technologiques retenues pour la mise au point de ce dispositif expérimental sont exposés dans le Chapitre 1. Un des objectifs principaux de ce développement est la mise en place d'un environnement de prototypage rapide de lois de commande (*Rapid Controller Prototyping*), afin d'accélérer le déploiement des algorithmes qui seront conçus par la suite. Pour cela, une approche de type *Hardware-In-the-Loop* a été appliquée, dans laquelle le prototype de projectile est conçu comme un système autonome. Les surfaces de commande sont actionnées au moyen de servomoteurs, et l'orientation du projectile est mesurée au moyen d'une centrale inertielle et de capteurs externes. Un calculateur embarqué coordonne l'ensemble des capteurs et actionneurs, et évalue les lois de commande en temps réel. Enfin, un environnement de développement basé sur MATLAB/Simulink permet la mise au point des algorithmes de commande, leur déploiement rapide sur l'ordinateur embarqué, la supervision de l'exécution et la collecte des données mesurées. Ce dispositif et l'ensemble de la chaîne d'acquisition et de traitement ont été validés et ont donné entière satisfaction par rapport à l'application envisagée.

Modélisation et identification

La seconde phase a porté sur l'exploitation du dispositif expérimental pour le développement de lois de commande. Les méthodes de l'Automatique moderne, telle que la commande robuste \mathcal{H}_∞ , permettent de concevoir des lois de commandes robustes et à performances élevée à partir de modèles relativement simples. Dans cette optique, nous nous sommes attachés à caractériser le comportement dynamique du projectile au moyen de familles de modèles linéaires dont les paramètres sont obtenus expérimentalement. Ces études sont présentées dans le Chapitre 2.

Dans un premier temps, le projectile est modélisé au moyen des outils de la mécanique du vol, en tenant compte des contraintes induites par le banc d'essais. Le modèle ainsi obtenu se distingue de celui décrivant le vol libre d'un projectile, par l'absence des équations décrivant la cinématique et la dynamique en translation (le projectile étant maintenu en son centre de gravité par la rotule), et par la présence de composantes d'inertie non négligeables induites par la rotule. Les forces et moments aérodynamiques appliqués au projectile sont décrits par des décompositions en coefficients aérodynamiques, dont les valeurs ne sont pas connues *a priori*. L'ensemble des équations décrivant le comportement du projectile résulte en un modèle non-linéaire à paramètres variants, dépendant notamment du nombre de Mach et de l'altitude. Néanmoins, les méthodes de commandes envisagées portent sur des structures de modèles linéaires, nécessitant une linéarisation du comportement du système au voisinage de ses points de fonctionnement. Ceux-ci sont caractérisés expérimentalement et conduisent à des surfaces d'équilibrage, dont un exemple est donné à la Figure 3. Cette cartographie traduit les points d'équilibre du système obtenus en fonction d'une entrée donnée. Elle conduit à un domaine de fonctionnement symétrique et linéaire, de forme identique pour la quasi-totalité des vitesses d'écoulement de la soufflerie étudiées. La linéarisation du modèle non-linéaire conduit alors au modèle d'état suivant :

$$\begin{bmatrix} \dot{\mathbf{x}}_\delta \\ \mathbf{y}_\delta \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x}_\delta \\ \mathbf{u}_\delta \end{bmatrix} = \left[\begin{array}{cccc|cc} 0 & 1 & 0 & 0 & 0 & 0 \\ M_{q\alpha} & M_{qq} & M_{q\beta} & 0 & M_{q\delta_m} & 0 \\ 0 & 0 & 0 & -\frac{1}{\cos \alpha} & 0 & 0 \\ M_{r\alpha} & 0 & M_{r\beta} & M_{rr} & 0 & M_{q\delta_n} \end{array} \right] \begin{bmatrix} \alpha_\delta \\ q_\delta \\ \beta_\delta \\ r_\delta \\ \delta_{m,\delta} \\ \delta_{n,\delta} \end{bmatrix} \quad (1)$$

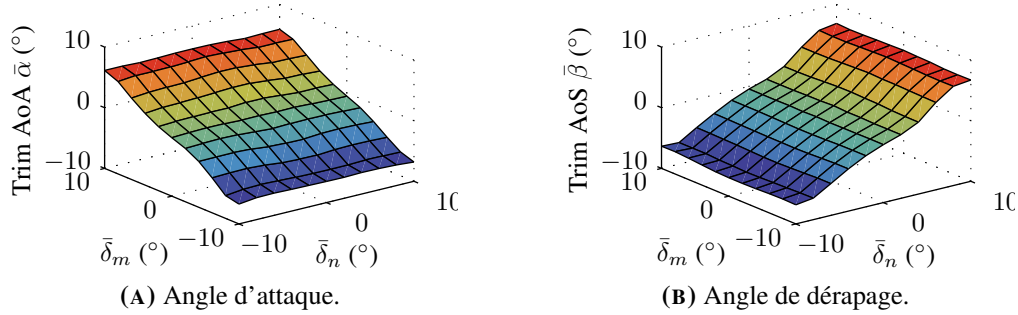


FIGURE 3: Surfaces d'équilibre en tangage et en lacet du projectile à $V = 25$ m/s.

Dans ce modèle, les paramètres M_{ij} sont dépendants du point de fonctionnement, tandis que les variables vectorielles \mathbf{x}_δ , \mathbf{u}_δ et \mathbf{y}_δ caractérisent les variations autour du point de fonctionnement. Le vecteur d'état \mathbf{x} du modèle non-linéaire correspondant contient les angles d'incidence (angle d'attaque α et angle de dérapage β) et les vitesses angulaires en tangage (q) et en lacet (r). Le vecteur d'entrée se compose des commandes virtuelles δ_m et δ_n , qui sont distribuées sur les 4 surfaces de contrôle selon un plan horizontal et un plan vertical, tel qu'illustré à la Figure 4.

Dans un second temps, une étude d'identification a visé à estimer les coefficients M_{ij} intervenant dans les matrices d'état de l'Équation (1) au moyen de données expérimentales. La détermination de ces paramètres constitue un problème inverse, qui peut être impossible à résoudre s'il est mal posé ou difficile à résoudre s'il est mal conditionné. Une procédure d'identification a été suivie afin de garantir l'existence et la qualité des estimées des paramètres du modèle. Celle-ci consiste à vérifier si le problème est bien posé et bien conditionné au moyen d'études d'identifiabilité respectivement *a priori* et *a posteriori*. Ces études ont permis de démontrer que les paramètres décrivant le couplage entre les deux axes ($M_{q\beta}$ et $M_{r\alpha}$) ne sont pas identifiables *a posteriori*, bien que la structure de modèle soit globale-

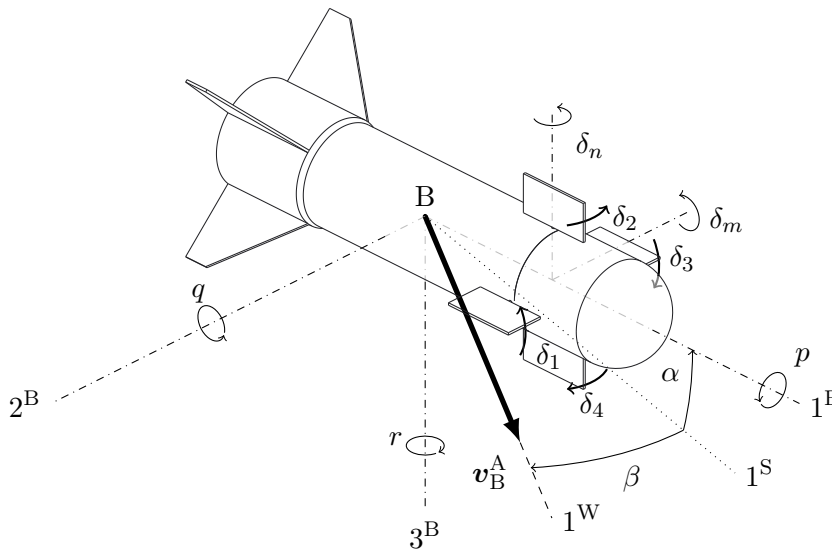


FIGURE 4: Position des surfaces de contrôles du projectile (δ_i), commandes virtuelles (δ_m , δ_n), vitesses angulaires (p , q , r) et angles d'incidence (α , β).

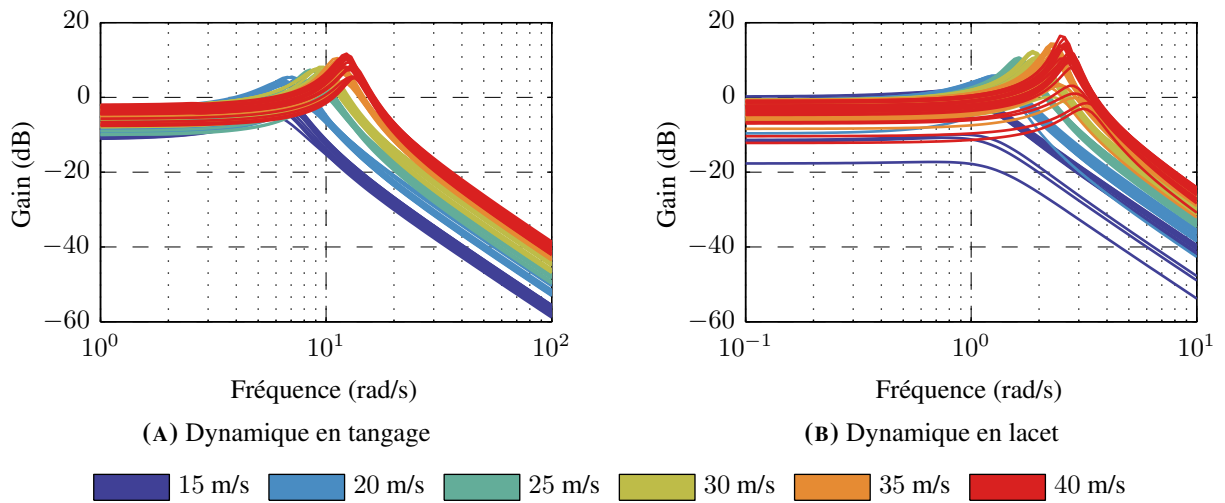


FIGURE 5: Evolution de la dynamique en boucle ouverte des modèles estimés, en fonction de la vitesse de l'air.

ment identifiable. Par conséquent, il est plus difficile d'obtenir un résultat précis pour ces paramètres. Comme chaque modèle ne décrit que le comportement local du projectile, les paramètres ont été estimés sur plusieurs points de fonctionnement afin de couvrir le domaine de vol complet. Pour cela, le projectile est stimulé autour de chacun de ces points de fonctionnement. L'algorithme d'estimation consiste alors à chercher le vecteur de paramètres minimisant l'erreur quadratique entre la réponse du modèle et les sorties mesurées. L'incertitude relative des paramètres obtenus est acceptable, à l'exception des paramètres de couplage ($M_{q\beta}$ et $M_{r\alpha}$) et de certains points de fonctionnement proches des limites du domaine de vol. Le comportement fréquentiel des modèles ainsi estimés est illustré à la Figure 5 et montre une forte dépendance vis-à-vis de la vitesse de l'air, illustrée par les différentes couleurs. En revanche, pour une vitesse donnée, le comportement du système varie modérément en fonction des angles d'incidence à l'équilibre.

Développement de lois de commande

Les familles de modèles issues de l'étude d'identification sont mises à contribution pour la conception de lois de commande du projectile au Chapitre 3. Ces lois de commande, ou autopilotes, visent à remplir différents objectifs : augmentation de la stabilité, rejet de perturbations, suivi de consigne et robustesse vis-à-vis de l'incertitude de modèle. Ces correcteurs sont par la suite implémentés sur le dispositif expérimental pour validation, ce qui requiert une étape de discrétisation introduisant des retards supplémentaires. Les méthodes de commande basées sur la synthèse \mathcal{H}_∞ sont particulièrement intéressantes pour le développement des lois de commande. En effet, ces méthodes fournissent des correcteurs robustes dont la stabilité et les performances nominales sont garanties. L'analyse de la robustesse vis-à-vis des incertitudes sur le modèle se fait dans un second temps et fait intervenir des techniques telles que la μ -analyse.

Trois applications de complexité croissante visent à valider l'approche envisagée. Une première étude a porté sur l'asservissement en tangage du projectile à vitesse constante et permet de comparer les performances de plusieurs méthodes de synthèse de contrôleurs basées sur la commande \mathcal{H}_∞ . Elle est détaillée en Annexe B. La seconde application (à la Section 3.3) tire parti des enseignements du cas

simple et vise à étendre le domaine de vol à l'axe de lacet, formant un autopilote de type *skid-to-turn* à vitesse constante.

Les points de fonctionnement du système dépendent de la vitesse de l'air et des angles d'incidence α et β . Cependant, ces derniers ne sont en général pas mesurés pour les missiles et projectiles guidés. De plus, l'adaptation du contrôleur en fonction de ces angles est un problème complexe car ceux-ci exhibent des dynamiques rapides induisant des termes de couplage cachés. Pour ces raisons, la commande à vitesse constante est réalisée via un unique contrôleur robuste. Dans le cas présent, les faibles variations du comportement du projectile à vitesse constante liées aux angles d'incidence, permettent de caractériser celui-ci au moyen d'un modèle nominal unique. Comme les angles d'incidence atteignables en pratique restent faibles (moins de 10°), l'incertitude induite par cette simplification reste modérée, et les correcteurs obtenus via cette approche conservent des marges de robustesse suffisantes.

La dernière application (présentée dans la Section 3.4) constitue une première extension de l'autopilote *skid-to-turn* au domaine de vol complet et tient compte des variations de la vitesse de l'écoulement de l'air dans la soufflerie.

Formulation du problème pour la synthèse

La structure de commande considérée pour les autopilotes *skid-to-turn* est illustrée à la Figure 6 et vise à asservir les angles d'incidence α et β en fonction des consignes α_r et β_r . Cette structure se compose de deux boucles imbriquées. La première est une boucle de régulation consistant en un retour de sortie stabilisant via la matrice de gain \mathbf{K}_R . La seconde est une boucle de rejet de perturbations et d'asservissement, où \mathbf{K}_S est un correcteur intégral agissant sur l'erreur de suivi de consigne afin de l'annuler en régime permanent. Finalement, un correcteur *feedforward* \mathbf{K}_F permet de façonner la réponse en boucle fermée du système afin qu'elle suive un modèle de référence donné à l'avance, dont les caractéristiques sont modulables par le concepteur.

La synthèse de ces trois correcteurs est réalisée en une seule étape, au moyen de techniques de synthèse \mathcal{H}_∞ à ordre et structure fixes. Ces méthodes permettent de fixer la structure des correcteurs afin de privilégier des architectures de commande peu complexes, par opposition aux techniques classiques produisant un correcteur d'ordre plein. Pour cela, le problème est reformulé sous la forme standard illustrée à la Figure 7. Les performances souhaitées en rejet de perturbations et en suivi de consigne sont imposées par le biais des filtres de pondération \mathbf{W}_{ij} , tandis que la réponse souhaitée en boucle fermée est spécifiée par le modèle de référence \mathbf{T}_{ref} . Ces deux objectifs sont traités indépendamment

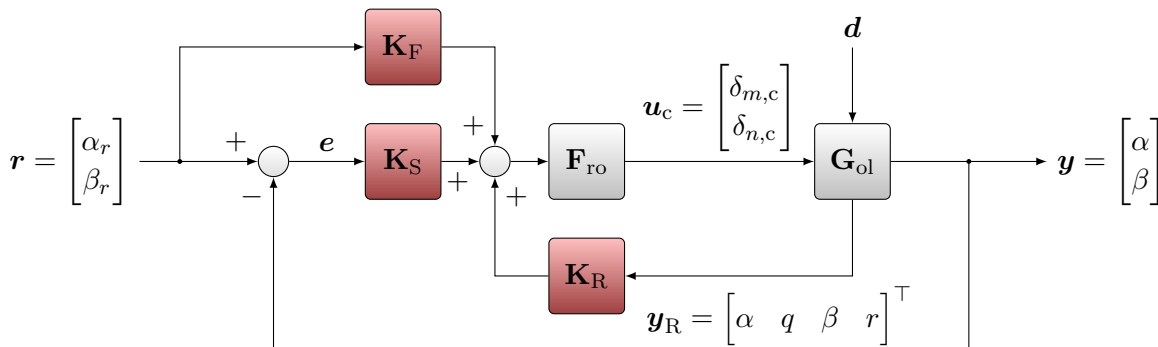


FIGURE 6: Structure de commande de l'autopilote.

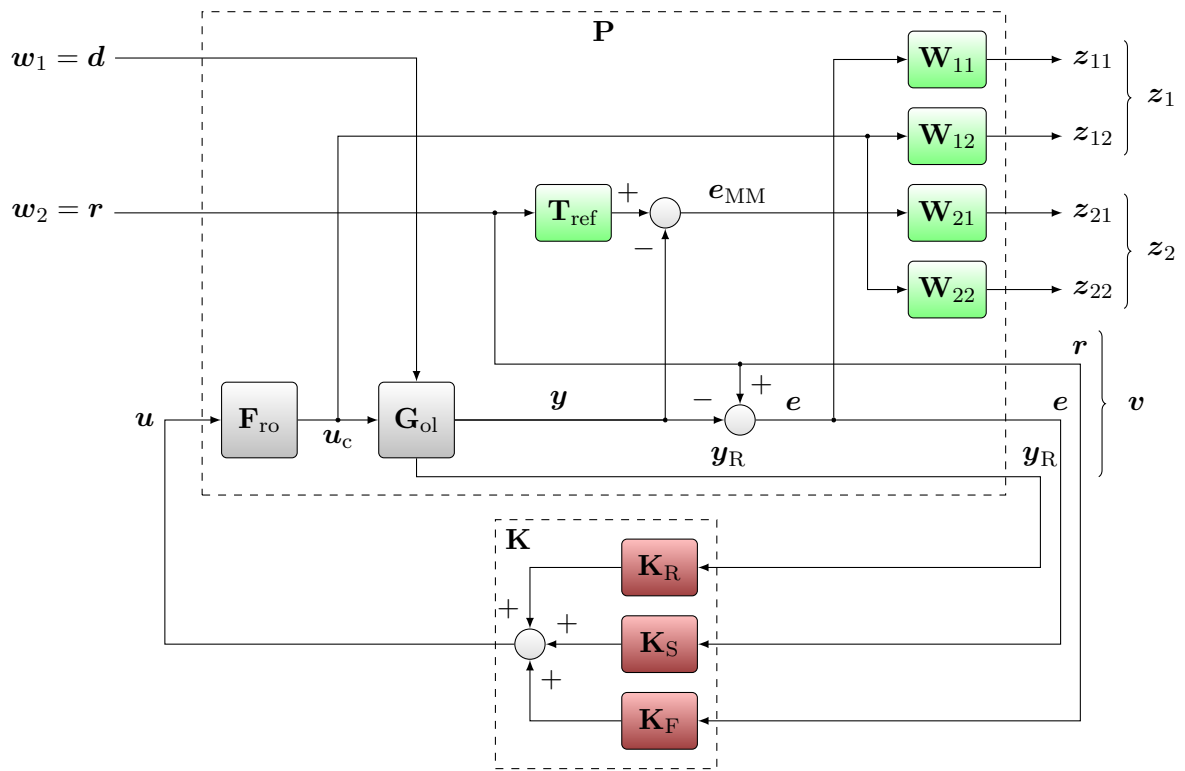


FIGURE 7: Schéma de synthèse de l'autopilote *skid-to-turn* sous forme standard.

mais simultanément au moyen d'une méthode de synthèse multi-objectifs. Le problème de synthèse du contrôleur $\mathbf{K} = \{\mathbf{K}_R, \mathbf{K}_S, \mathbf{K}_F\}$ peut alors se formuler de la façon suivante: pour γ donné, trouver l'ensemble des correcteurs \mathbf{K} stabilisant le système tels que

$$\max \{ \|\mathbf{T}_{w_1 \rightarrow z_1}(\mathbf{P}, \mathbf{K})\|_\infty, \|\mathbf{T}_{w_2 \rightarrow z_2}(\mathbf{P}, \mathbf{K})\|_\infty \} < \gamma \quad (2)$$

où les $\mathbf{T}_{w_i \rightarrow z_i}(\mathbf{P}, \mathbf{K})$ décrivent les fonctions de transfert en boucle fermée relatives à chaque objectif de commande. Une étape supplémentaire vise à analyser la stabilité robuste des contrôleurs ainsi obtenus au moyen de la μ -analyse, en tenant compte du modèle incertain du système.

Cas du tangage-lacet à vitesse constante

Cette approche a tout d'abord été appliquée pour la conception d'un autopilote de tangage-lacet à vitesse constante $V = 25$ m/s. Les objectifs de commande, à savoir le rejet des perturbations et le suivi des consignes, ont été validés par des simulations du modèle nominal ainsi que des essais expérimentaux. Le rejet de perturbations, en particulier le facteur d'atténuation et le temps de retour à l'équilibre, a été évalué sur plusieurs points de fonctionnement par le biais de créneaux de perturbations appliqués à l'entrée de l'actionneur. Ces résultats sont illustrés à la Figure 8 et montrent des performances conformes à la simulation, avec une annulation complète de l'erreur statique.

Une seconde étude expérimentale, illustrée à la Figure 9, a visé à valider les performances de suivi de consigne sur l'ensemble de l'enveloppe de vol. Bien que les dynamiques en tangage et en lacet soient sensiblement différentes, le contrôleur obtenu offre un niveau de performance similaire au modèle de référence utilisé lors de la synthèse.

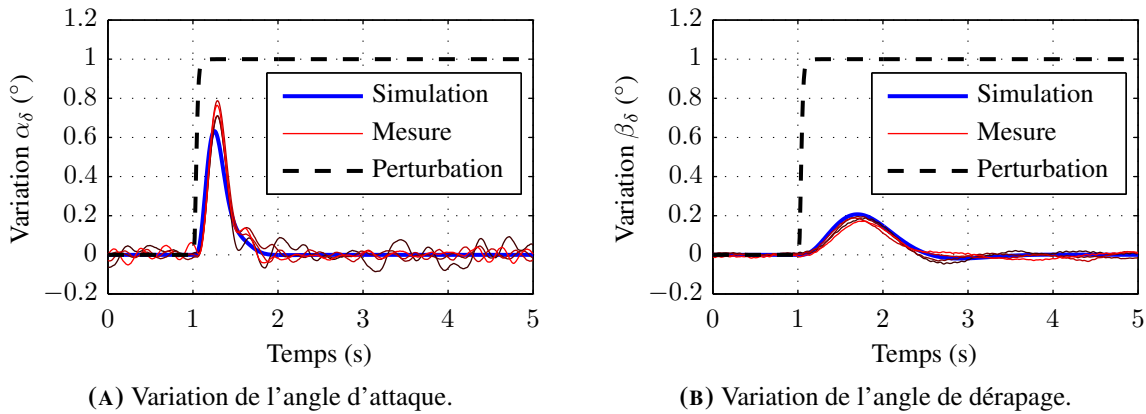


FIGURE 8: Performance en rejet de perturbations, à $V = 25$ m/s et pour plusieurs points de fonctionnement $(\bar{\alpha}, \bar{\beta}) = (0^\circ, 0^\circ), (0^\circ, 3^\circ), (2^\circ, 3^\circ)$ et $(2^\circ, 0^\circ)$.

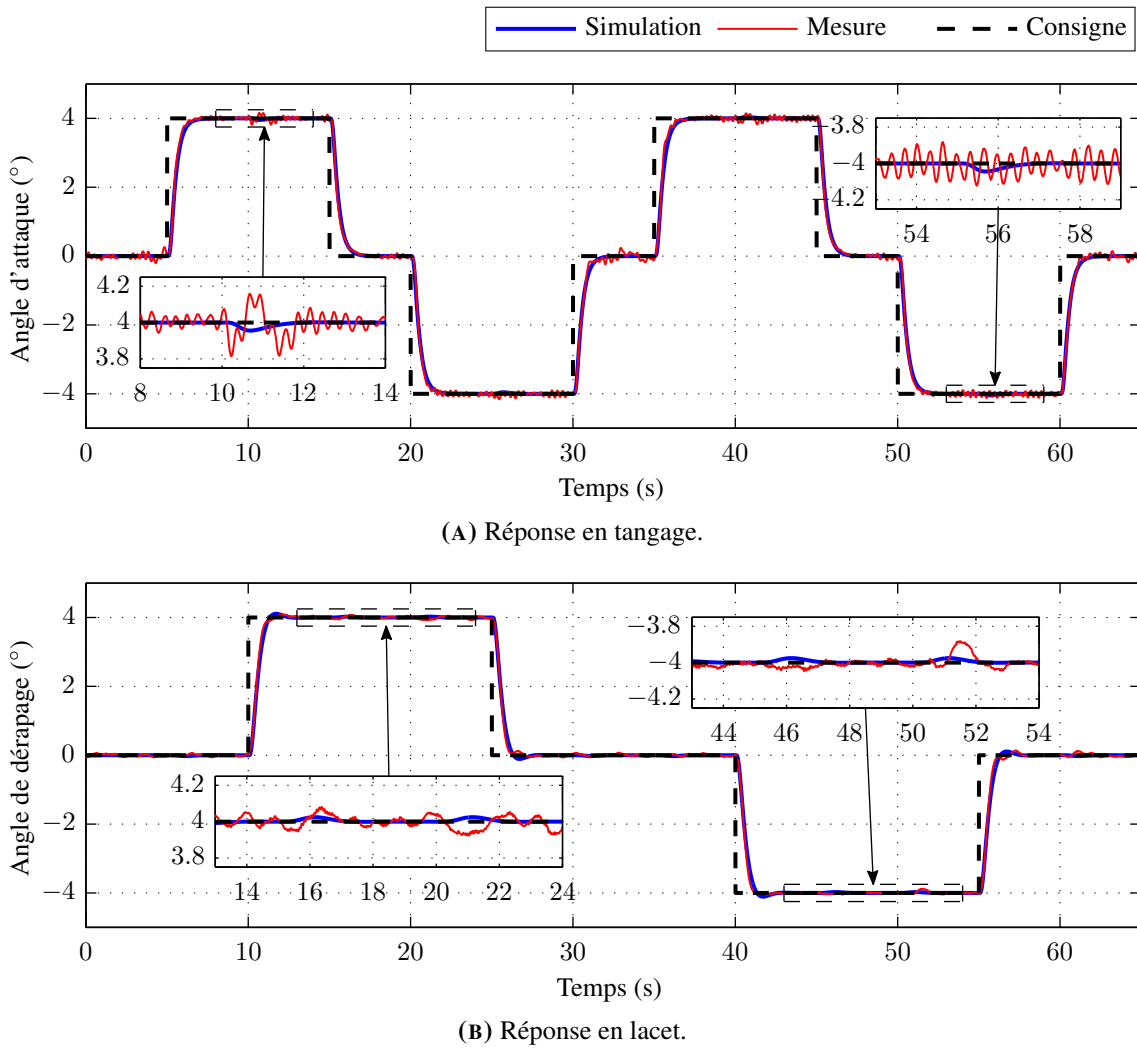


FIGURE 9: Performance en suivi de consigne à $V = 25$ m/s pour des consignes simultanées d'angle d'attaque et d'angle de dérapage.

Cas du tangage-lacet à vitesse variable

Dans le dernier cas considéré, les variations de la dynamique du système liées à la vitesse de l'air ne permettent plus d'envisager une approche de commande basée sur un correcteur unique. Afin de prendre en compte ces variations, les techniques de séquençement de gains permettent d'adapter les contrôleurs à un ensemble réduit, mesurable et représentatif des conditions de vol, afin de maintenir un niveau constant de performance et de stabilité de l'autopilote. Parmi les techniques existantes, les approches de type *divide-and-conquer* permettent de réemployer les résultats obtenus à vitesse constante. Ces approches consistent à diviser le domaine de fonctionnement via un maillage de points de séquençement, et à calculer un contrôleur local pour chaque point de séquençement. Dans le cas présent, des correcteurs similaires à ceux illustrés à la Figure 6, sont calculés pour des vitesses de l'air comprises entre 15 et 40 m/s, par pas de 5 m/s.

Pour la construction du contrôleur complet, une approche *ad-hoc* couramment employée en pratique consiste à interpoler les gains, zéros et pôles de ces contrôleurs locaux en fonction du point de fonctionnement. Bien que cette approche ne fournisse aucune garantie théorique de stabilité, de performance et/ou de robustesse, les résultats obtenus expérimentalement, illustrés à la Figure 10 sont conformes aux spécifications attendues. En effet, les performances en suivi de consigne restent identiques malgré les variations de la vitesse de l'air dans la soufflerie. Ces résultats montrent également que le système reste stable dans toute sa plage de fonctionnement, bien que le projectile soit plus sensible aux perturbations aux vitesses élevées.

Conclusion et perspectives

Cette thèse propose une approche novatrice permettant la mise au point de lois de commande de projectiles guidés durant des phases d'avant-projet, au moyen d'une méthodologie basée sur l'exploitation d'un environnement expérimental mettant en œuvre une maquette du concept étudié. Un premier démonstrateur centré sur un projectile empenné dans une soufflerie subsonique a visé à évaluer la faisabilité de cette approche sur un cas concret. Ses caractéristiques ont permis la mise en place d'approches de commande robuste utilisant des familles de modèles linéaires identifiés expérimentalement. Dans l'ensemble des cas étudiés (conception d'autopilotes de tangage seul et de tangage-lacet à vitesse fixe puis à vitesse variable), les mesures expérimentales sont sensiblement identiques aux résultats de simulation, ce qui confirme la validité des approches présentées.

Plusieurs axes de recherche se dégagent des résultats de cette première étude. Sur le plan expérimental, il sera nécessaire de se rapprocher des conditions du vol réel d'un projectile (nombre de Mach $> 0,5$) au moyen de souffleries pouvant atteindre des vitesses plus élevées. À ces vitesses plus élevées, les efforts induits sur les surfaces de commande (moments de charnière) ainsi que les modes flexibles ne sont alors plus négligeables et doivent être pris en compte lors de la conception. Afin d'améliorer la fidélité du dispositif expérimental en vue de réaliser des simulations HIL à 6 degrés de liberté, les forces induites par l'aérodynamique doivent être prises en considération, au moyen de balances aérodynamiques par exemple. Enfin, il serait intéressant d'adapter cette approche à d'autres architectures, tel que les projectiles gyroscopiques munis d'une fusée découplée, qui ont reçu une attention accrue ces dernières années.

Les prolongements sur le plan méthodologique portent à la fois sur la modélisation et l'identification du comportement du projectile et la mise au point de lois de commande. Afin d'améliorer la fidélité

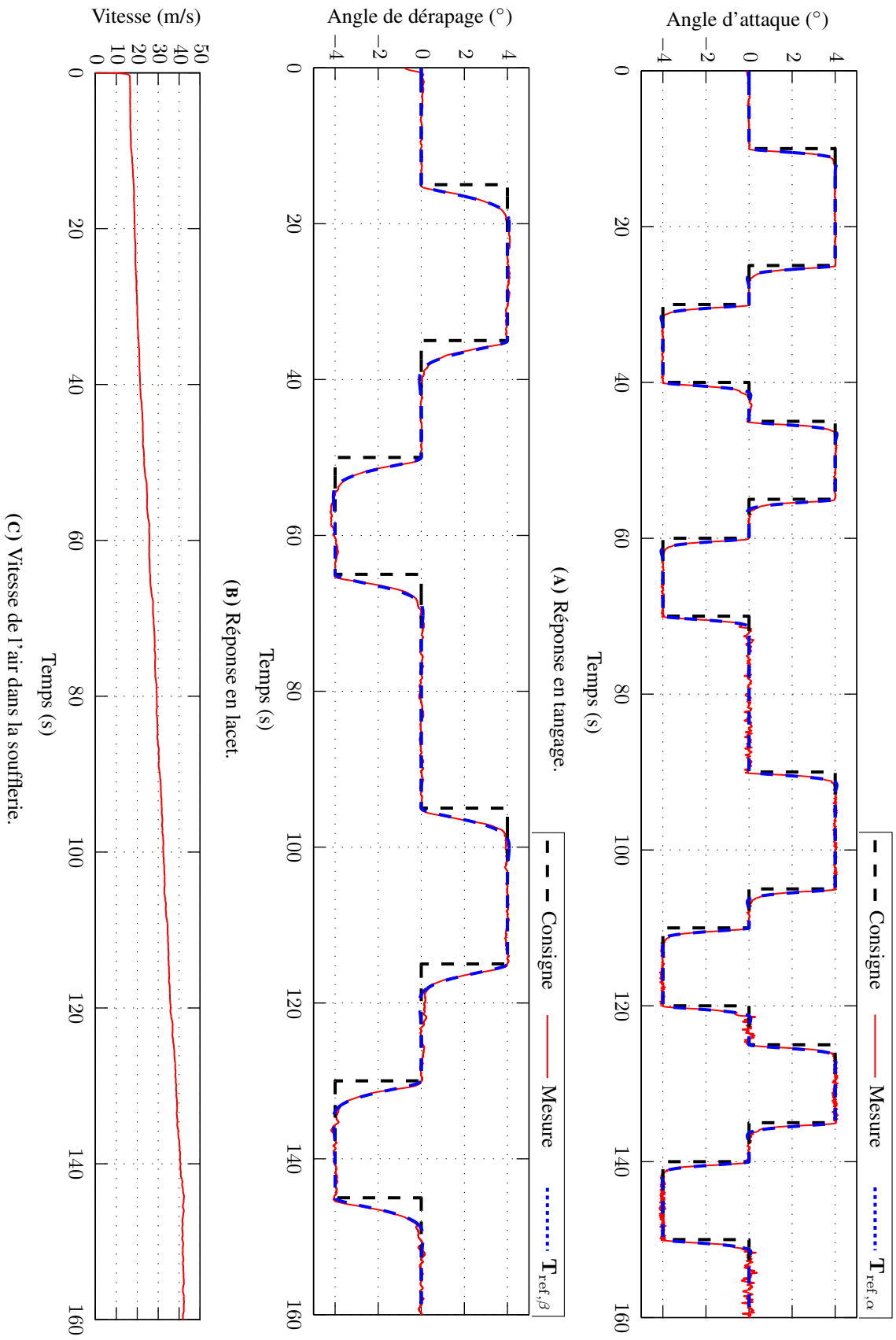


FIGURE 10: Performance en suivi de consigne de l'autopilote par séquençement de gains sur le domaine de fonctionnement complet.

du modèle non-linéaire, notamment dans le cas d'architectures faisant apparaître une dépendance plus marquée envers les conditions de vol, il peut être intéressant de considérer une description polynomiale des coefficients aérodynamiques intervenant dans le modèle. La difficulté principale consiste alors à estimer ces coefficients de polynômes à partir de mesures en soufflerie car la structure de modèle est non-linéaire. Du point de vue commande, les différents axes de recherche portent sur l'amélioration de la méthodologie présentée. Ainsi, des méthodes de commande multi-modèles permettraient de tenir compte de tous les membres des familles de modèle estimés, évitant ainsi l'approximation par un modèle nominal unique. Il sera également nécessaire de s'intéresser à des méthodes de séquençement de gain présentant des garanties de stabilité et de performance, en portant une attention particulière aux termes de couplage cachés pouvant apparaître. La performance des autopilotes aux limites de l'enveloppe de vol pourrait être améliorée au moyen de techniques anti-saturation et anti-windup. Sur le plus long terme, il serait intéressant de mettre en œuvre des méthodes de commande LPV, qui présentent des garanties de stabilité et de performance plus rigoureuses. Cependant, certains points du cadre théorique associé à l'identification de modèles LPV à temps continu sont encore des problèmes ouverts à ce jour.

Nomenclature

Notation Rules

x	Scalar
\boldsymbol{x}	Vector
\mathbf{X}	Matrix
\boldsymbol{x}^\top (or \mathbf{X}^\top)	Transpose of \boldsymbol{x} (or \mathbf{X})
\mathbb{I}_n	Identity matrix of size $n \times n$
$\mathbb{O}_{n \times m}$	Zero-filled matrix of size $n \times m$

Rigid-Body Mechanics Notations

\boldsymbol{v}_A^B	Velocity of point A <i>w.r.t.</i> frame B
\boldsymbol{a}_A^B	Acceleration of point A <i>w.r.t.</i> frame B
$\boldsymbol{\omega}^{AB}$	Rotational velocity vector of frame A <i>w.r.t.</i> frame B
$\boldsymbol{\Omega}^{AB}$	Skew-symmetric rotational velocity tensor of frame A <i>w.r.t.</i> frame B
$[\cdot]^A$	Projection in coordinate system A
$[\mathbf{T}]^{BA}$	Transformation matrix from coordinate system A to B
\mathcal{D}^A	Rotational time derivative <i>w.r.t.</i> frame A
\boldsymbol{l}_C^{AB}	Angular momentum of body A <i>w.r.t.</i> frame B expressed at point C
\boldsymbol{h}_C^{AB}	Rate of change of the angular momentum of body A <i>w.r.t.</i> frame B expressed at point C
\mathbf{I}_B^A	Inertia tensor of body A expressed at point B

Frames and Coordinate Systems

B	Body frame
E	Earth frame
$]^B$	Body coordinate system
$]^L$	Local coordinate system
$]^W$	Wind coordinate system

Projectile Model Quantities and Variables

α	Angle of attack (deg)
β	Angle of sideslip (deg)
ϕ, θ, ψ	Euler angles (roll, pitch and yaw) (deg)
$\mathbf{f}_a, \mathbf{m}_B$	Aerodynamic forces and moments (N) (Nm)
h	Altitude (m)
C_X, C_Y, C_Z	Axial force, sideforce and normal force aerodynamic coefficients
C_l, C_m, C_n	Roll, pitch and yaw moment aerodynamic coefficients
\mathcal{M}	Mach number
S	Projectile reference area (m ²)
V	Airspeed (m/s)
m	Mass (g)
u, v, w	Translational velocities along the body axes (m/s)
p, q, r	Body angular rates (deg/s)
$\delta_l, \delta_m, \delta_n$	Virtual roll, pitch and yaw fin deflections (deg)
g, \mathbf{g}	Gravity norm (m/s ²) and vector
\bar{q}	Dynamic pressure (N/m ³)

Linearized Model Quantities and Variables

M_{ij}	Linear model parameters
$\boldsymbol{\theta}$	Parameters vector
$\boldsymbol{\rho}$	Equilibrium (trim) point vector
$\boldsymbol{\sigma}$	Operating point (external parameters) vector
$\bar{\cdot}$	Value of \cdot at equilibrium
\mathbf{F}	Fisher information matrix
$\mathbf{S}_y, \hat{\mathbf{S}}_y$	Sensitivity and scaled sensitivity matrices
s_y/θ_i	Sensitivity of output y w.r.t. parameter θ_i

Control-related Symbols

Notations

$\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i$	State-space matrices associated to transfer function \mathbf{G}_i
$\mathcal{F}_l(\cdot, \cdot)$	Lower linear fractional transformation
$\mathcal{F}_u(\cdot, \cdot)$	Upper linear fractional transformation
$\ \mathbf{G}(s)\ _\infty$	\mathcal{H}_∞ norm of the transfer function $\mathbf{G}(s)$
$\ \mathbf{G}(s)\ _2$	\mathcal{H}_2 norm of the transfer function $\mathbf{G}(s)$
$\bar{\sigma}(\mathbf{M})$	Maximum singular value of \mathbf{M}
$\underline{\sigma}(\mathbf{M})$	Minimum singular value of \mathbf{M}
$\mu(\mathbf{M})$	Structured singular value of \mathbf{M}

$\mathbf{T}_{i \rightarrow j}$	Transfer function from channels i to j
$\mathbf{S}_{i \rightarrow e}$	Closed-loop sensitivity function to disturbance channel i
$\bar{\mathbf{P}}$	Nominal or mean-parameters plant
$\tilde{\mathbf{P}}$	Perturbed plant
$\mathbf{\Pi} = \{\mathbf{P}_i\}$	Family of systems

Models and Subsystems

d	Disturbance signal
u	Input signal
y	Output signal (measurement)
r	Reference signal
\mathbf{G}_{ol}	Open-loop plant
\mathbf{G}_A	Actuator model
\mathbf{G}_B	Airframe model
\mathbf{G}_C	Control delay
\mathbf{G}_Z	Sample-and-hold delay
\mathbf{T}_{ref}	Reference model
ω_0, ζ	Natural frequency (rad/s) and damping ratio of a second-order system
\mathbf{W}_{ij}	Weighting filter
\mathbf{K}	Controller
\mathbf{P}	Augmented plant
w	Exogenous inputs
z	Performance outputs
Δ	Perturbation matrix

Abbreviations and Acronyms

ACHILES	Automatic Control Hardware-In-the-Loop Experimental Setup
AHRS	Attitude Heading Reference System
BMI	Bilinear Matrix Inequality
CFD	Computational Fluid Dynamics
DoF	Degree(s) of Freedom
GNC	Guidance, Navigation & Control
GM	Gain Margin
GPS	Global Positioning System
HIL	Hardware-In-the-Loop
I ² C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
LBGS	Linearization-Based Gain Scheduling
LHP	Left-Hand Plane
LMI	Linear Matrix Inequality

LPV	Linear Parameter-Varying
LTI	Linear Time-Invariant
NP	Nominal Performance
NS	Nominal Stability
MEMS	Micro Electro-Mechanical System
MIMO	Multiple Inputs Multiple Outputs
MIPS	Million Instructions Per Second
MPC	Model Predictive Control
PM	Phase Margin
RHP	Right-Hand Plane
RP	Robust Performance
RS	Robust Stability
RTOS	Real-Time Operating System
SISO	Single Input Single Output
UAV	Unmanned Air Vehicle

Introduction

War theaters in the last two centuries have seen the increasing use of artillery as one of the most lethal class of weapons. While the origins of artillery can be traced back as far as the Roman siege engines or the invention of gunpowder, the age of modern artillery started during the Industrial Revolution [Kopp, 2005; Bastable, 1992]. Advances in metallurgy allowed the introduction of the rifled barrel, which imparts a spin movement to the munition improving the in-flight stability (gyrostabilization). The majority of past and current artillery projectiles are non-propelled, unguided gun-fired shells, which can be mass-produced at very low unit costs. However, these projectiles follow an unguided ballistic trajectory after launch, rendering them ineffective against moving targets. The trajectory is also affected by the launch conditions and in-flight disturbances such as wind gusts, degrading the impact point precision and limiting the usable range to a few0 tens of kilometers. While conventional artillery has traditionally been used for barrage fires, using a high volume of rounds to create a psychological effect on enemy combatants, attacking a specific target also requires a large number of rounds due to the impact dispersion associated with unguided shells. The large expenditure of ammunition necessary to accomplish the mission results in a high logistics demand as well as a high risk of collateral damage.

Between the First and the Second World War, several efforts were made to overcome these limitations, leading to the development of the first guided missiles. Early examples, such as the German Fritz X, were unpowered aircraft-dropped guided bombs. Advances in rocketry led to the development of the first long-range missiles, such as the German *Vergeltungswaffe* (retaliation weapons) 1 and 2. The V2 was the first operational guided ballistic missile, carrying a 1 tonne payload over a range of 300 km [Becklake, 1995] and relied upon a simple guidance system to keep the missile stable on a preplanned trajectory [MacKenzie, 1993]. Later developments of ballistic, or *strategic* missiles culminated during the Cold War with the introduction of intercontinental ballistic missiles (ICBMs) such as the Atlas missile [Siouris, 2004], with ranges over 10000 km. Another class of missiles are *tactical* missiles, which are characterized with a shorter range but high maneuverability and the ability to engage and follow fast moving targets. These improvements in precision, range and maneuverability are obtained using advanced, high-performance Guidance, Navigation and Control subsystems. However, the unit cost of missiles remains several orders of magnitude higher than classical artillery shells.

The last decades have also seen an increasing interest in the development of cannon-launched guided projectiles. These munitions aim at improving the precision and range of artillery by providing integrated guidance, navigation and control functions while remaining significantly cheaper than missiles. The improved precision of these projectiles reduces the number of shells required for a given target, hereby decreasing the logistic burden and limiting the risk of collateral damage. Although the development of these munitions has started in the 1960s [Ross, 1963], they have recently received a renewed interest

for both technical and operational reasons. Advances in semiconductor manufacturing have resulted in the widespread availability of low-cost, small-footprint sensors and embedded computers, allowing their integration in small munitions. At the same time, army budgets are becoming tighter while the operational requirements in artillery precision and range are increasingly stricter. Indeed, modern combat scenarios may take place in complex and urban terrain, where collateral damage, supply chain demand and operational effectiveness are primary concerns.

Guided Projectile Technologies

Guided projectile concepts can be classified upon the means of achieving airframe stability: fin-stabilization or spin-stabilization. *Fin-stabilized* projectiles, such as those illustrated in Figure 1, achieve airframe stability using aerodynamic surfaces such as wings and tail fins, and are similar in shape and aerodynamics to missiles. The US Army M712 Copperhead [Morrison and Amberntson, 1977] and XM982 Excalibur [Wells, 2000] as well as the Russian GRAN¹ are field-operational examples of this class of projectiles, using canards for steering the airframe. Similar concepts, such as MPM² (Metric Precision Munition) and Vulcano³ have been studied or are developed in Europe. *Spin-stabilized* projectiles are similar to classical artillery shells in that they use a high spin rate for achieving gyroscopic stabilization. In this context, an interesting guided projectile concept are Course-corrected fuzes (CCF), which are screw-on replacements for conventional fuzes, allowing the reuse of existing unguided ammunition stockpiles with added guidance and control functionality. Several CCFs have been studied or developed for operation with 155 mm caliber ammunition, such as the Spacido fuze [Campion, 2007], the European Correction Fuze [Perrin, 2011] and the XM1156 Precision Guidance Kit [Burke and Pergolizzi, 2009], which are illustrated in Figure 2. The Spacido and ECF fuzes provide one-dimensional course correction using a deployable air brake, while the PGK provides two-dimensional guidance using an roll-orientable set of fixed canards.

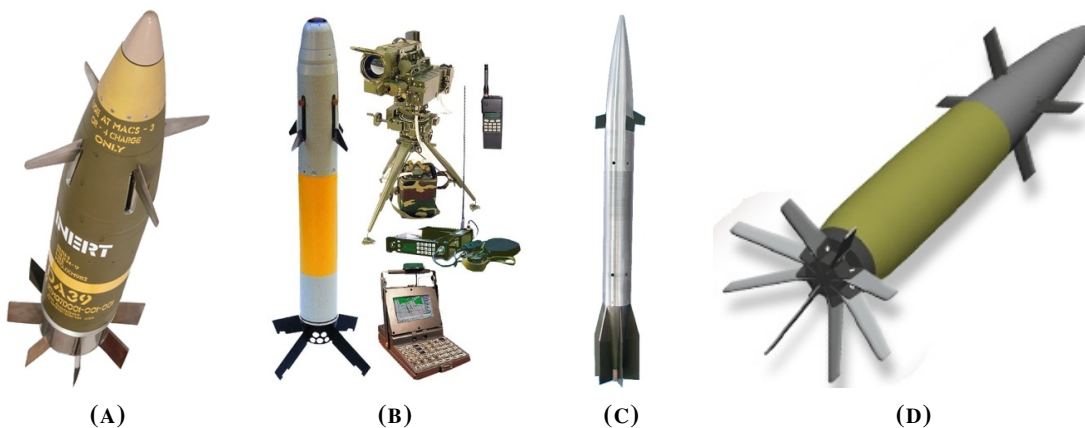


FIGURE 1: Fin-stabilized projectile concepts: (A) XM982 Excalibur, (B) GRAN, (C) Vulcano and (D) MPM projectiles.

¹<http://kbptula.ru/>

²<http://www.nexter-group.fr/>

³<http://www.diehl.com/>

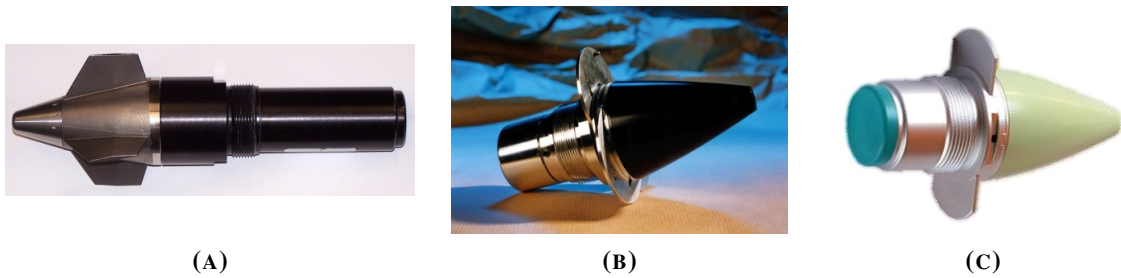


FIGURE 2: Course-Correction Fuzes: (A) XM1156 Precision Guidance Kit, (B) Spacido fuze and (C) European Correction Fuze.

Several technological solutions have been explored for influencing the trajectory of the projectile during flight, and three main categories of steering mechanisms can be distinguished:

- **Aerodynamic effectors**, which are by far the most common: all operational or almost operational concepts presented in the above paragraph fall under this class. The most common type of effector is the rotating wing surface, usually mounted at the aft or at the nose, in which case it is also called a *canard*. Other types of aerodynamic effectors include deflectable cones [Costello and Agarwalla, 2000; Landers et al., 2003] and retractable pins [Fresconi et al., 2011; Libsig, 2016].
- **Impulse Thrusters**, which are based on the reaction force provided by gas generators [English et al., 2006] or miniature explosive charges [Corriveau et al., 2011].
- **Inertial effectors** based on internal moving parts, which act on the mass distribution by offsetting the center of gravity from the main axis of the projectile in order to create control authority [Rogers and Costello, 2008].

The number and type of actuators as well as the means of achieving airframe stability have a great influence on the design of the autopilot. Two types of maneuvering strategies are commonly employed in the design of the autopilot and guidance system: *bank-to-turn* and *skid-to-turn*. The *bank-to-turn* maneuver is similar to banking turns performed with aircraft. It consists in rolling the airframe such that the maximum aerodynamic normal force is oriented in the plane of the commanded direction and is carried out in two steps: first the airframe is rolled towards the commanded direction, then a pitching maneuver is performed to achieve the commanded acceleration [Carter and Shamma, 1996]. This method is interesting in that actuators are only required for controlling the roll and pitch of the munition, and is particularly suited to airframe architectures with no rotational symmetry. However it is sensitive to sideslip disturbances and has a limited maneuver capability. By contrast, the *skid-to-turn* maneuver consists in actuating the airframe in its two lateral planes, generating normal and sideslip forces, while the roll angle is kept constant or ignored. Since there is no need to perform an initial roll maneuver, *skid-to-turn* vehicles have high maneuvering capabilities. This maneuver is especially adapted to architectures with rotational symmetry, however more actuators may be required than in the *bank-to-turn* case.

The design of guided projectiles involves some unique technical and scientific challenges. The gun launch represents one of the major issues due to the high shock forces involved (over 10,000 *gs*), which requires *g*-hardening of the components. The cost-driven philosophy tends to favor off-the-shelf, low-cost actuators and sensors which come at a fraction of the cost of missile components but with severe

limitations in performance and accuracy. In the context of spin-stabilized projectiles, the high spin rate induces complex aerodynamic effects (e.g. the Magnus moment [Klatt et al., 2014]) as well as significant coupling between the pitch and yaw axes caused by the gyroscopic effect. While these effects are well-mastered in conventional ballistics, they represent challenging control issues due to the high dynamics involved, the cross-coupling effects and limitations in the actuators and sensors. One interesting solution to reduce the actuator demand are dual-spin architectures [Wernert and Theodoulis, 2011], which decouple the nose, where the canards are installed, from the fast spinning projectile body.

The Guidance, Navigation and Control Loop

The improvements in precision of guided projectiles are obtained with the integration of Guidance, Navigation and Control (GN&C) functions. These functions are responsible for piloting the attitude and position of mobile systems using high-level commands, reducing or avoiding human intervention during operation. GN&C functions are present at the core of almost all autonomous or semi-autonomous systems such as manned and unmanned aircraft, spacecraft, missiles, robots and autonomous cars [Marin et al., 2014; Paull et al., 2013]. They are implemented using a combination of actuators, sensors, and embedded software processing. The roles of each function in a GN&C system is as follows:

- *Guidance* consists in determining the system trajectory according to a high-level mission (e.g. go to a given location, follow another vehicle) and generate reference signals according to the current navigation solution.
- *Navigation* consists in determining the current attitude, velocity and position of the system using a combination of proprioceptive sensors (e.g. accelerometers, gyroscopes) and exteroceptive sensors (e.g. magnetometers, GPS).
- *Control* consists in driving the system so as to follow the guidance reference signals. This low-level function is responsible for ensuring the stability and performance of the system.

The GN&C functionality is typically implemented using two cascaded loops as illustrated in Figure 3. In this block diagram, the high-level guidance functionality is implemented in the outer loop and the inner loop deals with the low-level control function. This thesis focuses on the design of the control loop, which is one essential step in the integration of GN&C functionality since it directly affects the stability of the vehicle as well as the overall performance of the system.

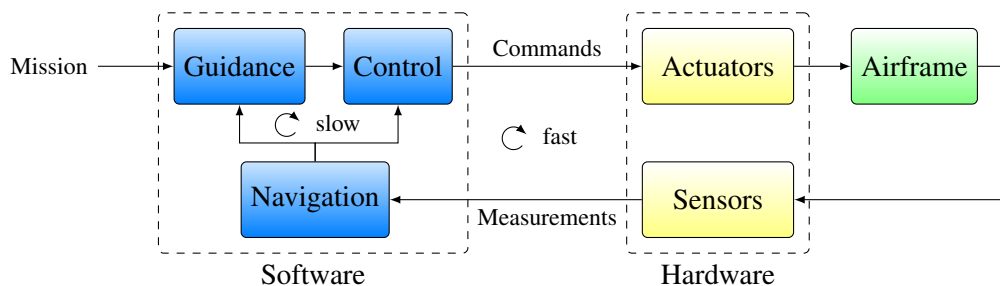


FIGURE 3: The Guidance, Navigation and Control loop.

Autopilot design for Guided Projectiles

Designing control laws for aerospace systems is a complex task, as their behavior is usually nonlinear and depends on the craft's flight conditions. Typically, the control law development processes can be divided in three main steps: system modeling, control law synthesis, and control law validation (or certification). The first step consists in capturing the behavior of the plant using a mathematical model, which is not only used for controller synthesis, but also required for evaluating the stability, performance, and robustness of the control laws. Flight dynamics is the study of the motion of flying vehicles through the air or space. In this framework, the vehicle kinematics and dynamics are modeled using classical rigid-body mechanics involving Newton's and Euler's laws. The aerodynamic forces and moments, acting on the vehicle and induced by its motion through the air, are governed by fluid dynamics laws. These aerodynamic forces and moments depend on the flight conditions and the vehicle geometry, and are notoriously difficult to model. They are generally described using decompositions in aerodynamic coefficients, of which the determination is an extensive and complex research subject. To this end, several complementary techniques may be employed: numerical methods and experimental techniques based on wind tunnel or free-flight tests. Some of these techniques are illustrated in Figure 4.

Aerodynamics Modeling

Multiple numerical methods for determining aerodynamic coefficients have been developed. Empirical and semi-empirical techniques are based on theoretical models, and databases of experimental and simulation results for known projectile architectures. These techniques are computationally efficient, making them especially useful in preliminary design phases. However the precision of the results largely depends on the quality of the database, and whether the studied configuration is close to reference cases contained in the database. These codes are hence application-specific and dedicated to one class of vehicles, for instance the US Air Force DATCOM for aircraft, Missile DATCOM [Rosema et al., 2011] and ONERA MISSILE for missiles, or PRODAS for projectiles.

Computational Fluid Dynamics (CFD) is another class of numerical techniques, which consists in solving the fundamental equations of fluid dynamics in a discretized domain around the vehicle [Anderson, 1995]. This technique can provide the three-dimensional flow structure as well as the pressure

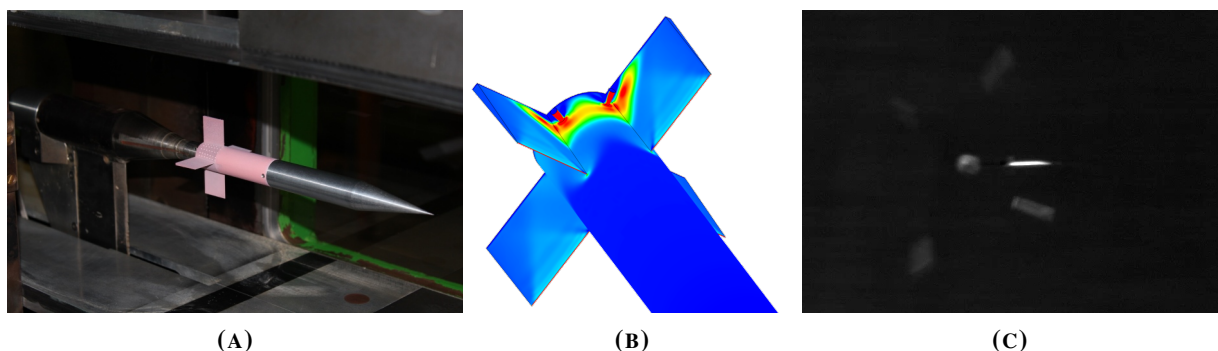


FIGURE 4: Illustration of several techniques for investigating the aerodynamics of a Basic Finner reference projectile [Libsig, 2016]: (A) Wind tunnel measurements using pressure sensitive paint, (B) pressure distribution in the presence of actuating pins computed using CFD predictions, (C) high-speed camera capture showing the separation of the projectile and its sabot in free-flight tests.

and velocity distribution, allowing the study of complex, nonlinear aerodynamic phenomena. However, this powerful technique comes at a high computational cost. A CFD simulation describes the vehicle behavior for a given, static operating point, and covering the complete flight envelope would require a very long computation time. Moreover, the characterization of dynamic effects (i.e. dynamic stability) requires even more complex simulations.

Wind tunnel testing is carried out on a scaled or full-size model of the craft to be studied and involves multiple measurement techniques, such as force and moment balance measurements, pressure distribution measurements or flow visualization [Barlow et al., 1999]. Using positioning systems, multiple combinations of incidence angles and Mach numbers may be tested in a timely manner. However the similarity with free flight is only partial, and interactions between the model, support system and the wind tunnel boundaries may result in incorrect determinations of some aerodynamic coefficients (e.g. the damping, the Magnus or the drag coefficients).

Finally, flight tests allow the characterization of the vehicle behavior in real operating conditions. In this method, aerodynamic forces and moments are not measured directly, but estimated from state measurements performed during the flight. This technique is frequently invoked for the development of human-piloted air vehicles, where the pilot applies specific stimuli to the aircraft during a test flight [Jategaonkar, 2006; Klein and Morelli, 2006]. In the case of artillery projectiles, the fast dynamics and short flight times, as well as the absence of means of control, pose significant challenges for the estimation of the aerodynamic coefficients [Albisser, 2015; Demailly, 2011]. While the free-flight techniques can provide the most accurate results, these techniques require a sufficiently mature, flightworthy prototype and hence may only be used late in the development process.

Control Law Design

Control law design for aerospace applications has been one of the major users and drivers of research on control theory, due to the complexity of flying vehicle dynamics and increasingly stringent safety requirements. While the literature is abundant with aircraft and missile applications, flight control for projectiles has received interest only recently; however most low-spin rate projectiles share common dynamics with missiles.

Even though classical methods on successive SISO loop closures have been successfully applied to early guided projectiles such as the M712 Copperhead [Morrison and Amberntson, 1977], a single linear controller may not provide sufficient stability and performance guarantees throughout the flight envelope due to the nonlinear, parameter-varying behavior of projectiles. To this end, nonlinear controllers based on adaptive methods such as Model Predictive Control (MPC) [Ollerenshaw et al., 2008] or dynamic model inversion [Calise et al., 1998], have been applied with success. However, these fully adaptive control techniques usually involve high computation costs, which are prohibitive for low-cost embedded computers, or lack theoretical robustness guarantees.

Another widely used class of adaptive techniques is gain scheduling (illustrated in Figure 5), either based on linear parameter varying (LPV) control theory [Wu et al., 2002], or based on the interpolation of linear time-invariant (LTI) controllers with respect to the flight condition [Theodoulis et al., 2015] (linearization-based gain scheduling). Linearization-based gain-scheduling is especially interesting in that for each considered operating point, the plant model is an LTI system, enabling the use of well-

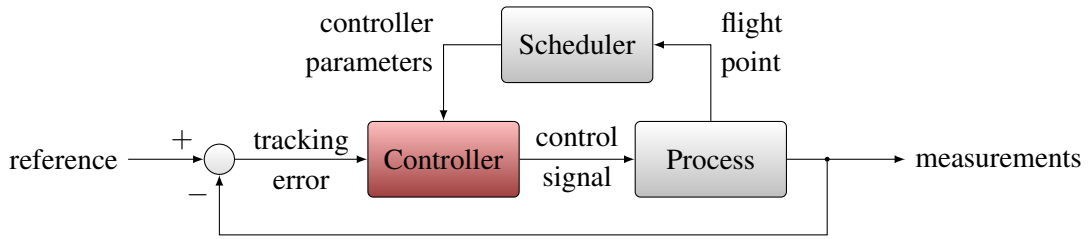


FIGURE 5: Gain-scheduling control configuration.

mastered linear control techniques. A design procedure for this control strategy is exposed in Jackson [2010]. Several techniques may be employed to design the corresponding controllers, for instance optimal control techniques such as linear quadratic regulators [Fresconi et al., 2015], or robust control techniques such as \mathcal{H}_∞ synthesis [Nichols et al., 1993; Sève et al., 2014]. The latter class of techniques has received significant interest as a result of the development of efficient solutions to the \mathcal{H}_∞ control problem [Doyle et al., 1989]. \mathcal{H}_∞ techniques are attractive in that they are able to deal with multivariable systems exhibiting cross-coupling between channels in an efficient yet practical manner, providing stability, performance and robustness guarantees on the obtained controllers.

Control Law Validation

The final step of typical flight control law development processes consists in the validation of the control law. This step assesses whether the system complies to the design requirements, in terms of stability, performance and robustness, in order to ensure safe operation throughout the flight envelope.

In general, validation procedures include classical gain, phase and delay margin analysis, nonlinear Monte-Carlo simulations, and Hardware-In-the-Loop (HIL) simulations [Kharisov et al., 2008]. Monte-Carlo simulations are based on the full nonlinear aerodynamic model and consist in running multiple simulations using randomly sampled values of uncertain parameters (e.g. aerodynamic coefficients, initial conditions). HIL simulation consists in including real-world components in the simulation loop, which then runs in real time. The interest of this technique is twofold: First, it allows testing the control laws using components which may be difficult to model, e.g. actuators [Ilg, 2008]. Secondly, as the simulation runs in real time, it can be used to ensure whether the controller implementation can deliver the control output within a specified control period. In these simulations, the vehicle dynamics and aerodynamics are generally simulated using the nonlinear model.

Flight testing is the final step of flight control law validation, in which the real-world performance of the controllers can be assessed. However these test campaigns are usually expensive and may only occur towards the end of the development.

The main limitation of these traditional validation approaches is that they can miss worst-case combinations of environmental conditions, uncertain parameters, input vectors and flight conditions which lead to degraded stability and performance margins. In order to assess the robustness of the designed control laws in a systematic manner, methods based on uncertain modeling such as the structured singular value μ framework have received significant interest in the last decades [Miotto and Paduano, 1995; Bates and Postlethwaite, 2002]. Under this framework, the various sources of uncertainty and limitations in the system are cast into a single uncertain model, and the robustness of the system is then tested for the worst-case combination of the uncertain parameters.

Motivation and Contributions

This thesis proposes a novel approach for the design of guided projectile control laws, which aims at overcoming the difficulties of the conventional approach, namely the determination of aerodynamic coefficients involved in the nonlinear model, and the lack of experimental validation facilities in early project phases. Since free-flight testing is not tractable in early project phases, an innovative experimental technique, based on the combination of dynamic wind-tunnel testing and Hardware-in-the-Loop simulation, is proposed. To this end, a fully-functional projectile prototype including its actuators and sensors is installed in a wind tunnel, by the means of a support structure which allows all three rotational degrees of freedom. This technique aims at reproducing the in-flight behavior of the studied projectile in a controlled environment, in order to gather experimental data describing the projectile behavior as well as to perform control law validation.

In order to assess the validity of this approach, this thesis investigates the design of a skid-to-turn autopilot for a proof-of-concept setup based on a fin-stabilized, canard-guided projectile prototype with tetragonal symmetry. While this class of projectiles does not present the various issues associated with spin-stabilized projectiles, there are still some challenges due to the nonlinear and uncertain nature of the system, as well as aerodynamic interaction between the fins and the canards.

As stated in the previous section, control techniques such as linearization-based gain scheduling and \mathcal{H}_∞ synthesis operate on LTI approximations of the nonlinear dynamics at specific operating points, and robustness validation of the control laws can be carried out using uncertain models. In this regard, the complete knowledge of the aerodynamic coefficients in the nonlinear model may not be necessary for control law design. Instead, the approach considered herein consists in linearizing the nonlinear analytical flight mechanics model into an LTI model structure, whose parameters are identified from experimental data.

To this end, a standard identification procedure is followed. It aims at evaluating the existence and uniqueness of solutions to the parameter estimation problem, first in an idealized context and then taking the actual input/output data into account. This knowledge is leveraged to build an optimal excitation signal, in order to maximize the fidelity of estimated models. Experimental data for parameter estimation are then collected for several operating points at different airspeeds, in order to build linear model families. The autopilot design is then carried out with \mathcal{H}_∞ -based control techniques for these families of models. The resulting control laws are validated against stability, performance and robustness objectives, first using theoretical robustness conditions, then by actual implementation on the experimental setup.

This thesis has been done in collaboration with the French-German Research Institute of Saint-Louis (ISL) and the MIPS laboratory (Modeling, Intelligence, Process and Systems) from the Université de Haute-Alsace in Mulhouse. The core mission of ISL is to conduct research and predevelopment studies in the defence and security fields. The GNC group, which initiated this project, focuses on the development and performance evaluation of guidance, navigation and control algorithms for gun-launched guided projectiles. Flight techniques for projectiles are one of the main research topics at ISL, with other teams focusing on aerodynamics and exterior ballistics, measurement and simulation techniques, and sensors, telemetry and communication. The MIAM team (Modélisation, Identification en Automatique et Mécanique) from the MIPS laboratory focuses on modeling, identification and control of complex systems, diagnosis and reconfiguration applied to the development of intelligent vehicles.

Related Work

While this type of approach has been used for other types of systems, or for the design and validation of projectile subsystems, there are very few examples in the literature regarding Hardware-In-the-Loop testing of a complete guided projectile.

An example of HIL simulation for a projectile subsystem is given in Ilg [2008]. In his thesis, the author investigated the design of a voice-coil-actuated canard system for a spinning projectile. Instead of de-spinning the projectile or decoupling the canard system from the body, the actuators are attached to the rolling frame and modulated *w.r.t.* the roll angle. In order to assess the performance of the actuator design and associated controller, HIL simulations are performed, in which a real prototype of the canard actuation system is integrated in the loop, while the 6-DoF aerodynamics are simulated.

In Hann et al. [2012], the nonlinear roll dynamics of a sounding rocket are investigated using a parameter identification technique. To this end, a scale model of a sounding rocket is installed in a vertical wind tunnel, with a support allowing the rocket to roll freely about its primary axis. In this setup, the roll motion of the rocket is controlled by servo-actuated tail fins, while the roll angle is measured by an optical encoder. Using piecewise-constant fin inputs, the roll dynamics of the rocket are stimulated and the roll response is recorded to perform estimation.

Perhaps one of the most similar approaches and setups is proposed by Fresconi et al. [2014, 2013]. These articles detail the design of a fin-stabilized, canard-guided projectile using low-cost commercial-off-the-shelf actuators and sensors, with a strong focus on qualifying these components for use in high-g environments. The performance of a linear quadratic roll autopilot is demonstrated in a wind tunnel, where the projectile is mounted on a sting allowing rotation along the roll axis. In this work, the projectile is modeled using aerodynamic coefficients given by PRODAS, which were further refined using parameter identification.

Both cases presented above currently focus only on the roll dynamics, with possibly future extensions to the pitch and yaw axes, whereas the proof-of-concept setup proposed here potentially aims at allowing motion on all three axes.

Manuscript Organization

Chapter 1 introduces the wind-tunnel-based experimental setup. A detailed review of the hardware components, namely the actuators, sensors, embedded computer, wind tunnel and support structure, is given. The software environment designed for rapid controller prototyping is also presented.

Chapter 2 deals with modeling and identification of the projectile prototype. In the first part of this chapter, the flight mechanics of gun-launched projectiles are developed, first in the free-flight case, then in the wind-tunnel environment where linear motion is prevented. More specifically, this study focuses on a roll-fixed configuration, where only pitch and yaw motions are allowed. The corresponding non-linear, parameter-dependent model is then linearized around equilibrium points, to derive a linear model structure suitable for identification and control. The second part of this chapter consists in the application of a parameter identification procedure, and on the estimation of families of linear models which describe the projectile behavior on its complete flight envelope.

Chapter 3 leverages the knowledge obtained in Chapter 2 to design skid-to-turn autopilots for the ACHILES projectile, with disturbance rejection, reference tracking and robust stability objectives. The cases of constant and variable airspeed are considered and discussed. The autopilots are synthesized using multi-objective fixed-structure \mathcal{H}_∞ synthesis, which yields low-complexity, robustly stable controllers with guaranteed performance requirements. The robustness of the proposed autopilots is assessed using an uncertain model, which takes into account the differences between estimated models and the nominal model, as well as uncertainties on the estimated parameter values. These controllers are then validated on the experimental setup and results are compared with numerical simulations.

Chapter 1

Experimental Setup

Contents

1.1	Introduction	11
1.2	The ACHILES Test Setup	13
1.3	Hardware Components	14
1.3.1	Mechanical Structure and Aerodynamics	14
1.3.2	Actuators	15
1.3.3	Sensors	21
1.3.4	Embedded Computer and Support Electronics	27
1.3.5	Wind Tunnel	29
1.3.6	Support Structure	29
1.4	Software Environment	31
1.4.1	Embedded Software	31
1.4.2	Development Environment	35
1.5	Concluding Remarks	37

1.1 Introduction

In this chapter, a custom test bench for guided projectile research and development is presented in detail. The ACHILES (for *Automatic Control Hardware-in-the-Loop Experimental Setup*) testbench is the basis of the modeling, identification, control and validation studies presented in the subsequent chapters. It is composed of a fully autonomous projectile prototype, installed in the test section of a wind tunnel by the means of a support structure allowing only angular motion. This setup is illustrated in Figure 1.1. The development procedure presented in the introduction involves frequent control law testing on the projectile hardware in laboratory conditions, a procedure known as *rapid control prototyping*, which shares multiple characteristics with Hardware-in-the-Loop (HIL) testing.

Hardware-in-the-Loop simulation is a technique combining *real* hardware with a *real-time* simulation loop [Isermann et al., 1999; Bacic, 2005; Schuette and Waeltermann, 2005]. Thus, the modeled process can consist of both real and simulated components. This technique has found widespread use in the

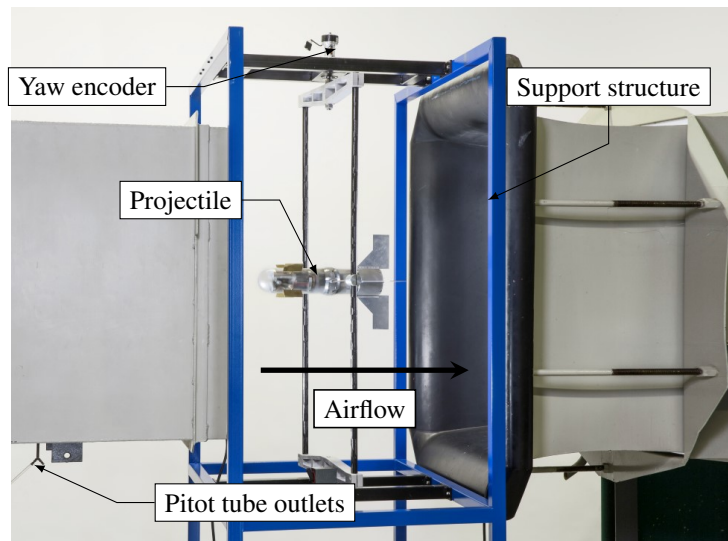


FIGURE 1.1: Overview of the ACHILES test setup, including the projectile prototype, the support structure and the test section of the subsonic wind tunnel.

industry, as it fills the gap between full software simulation and real-world testing. A typical use case is the development of control algorithms for complex plants, which cannot be tested on a real prototype in operating conditions for timing, cost and/or safety reasons. For example, consider the development of a missile guidance algorithm based on an optical seeker and a laser designator. In this case, the actual missile actuators and seeker are employed, while the aerodynamics, the sensors and the control loop are simulated in real-time. Another typical use case is the validation of control units against edge cases⁴ or failure conditions. These test cases would prove unrealizable or create hazards when assessed on the real system, while HIL simulation provides a safe and repeatable testing environment. One typical application is the development of Electronic Control Units (ECUs) in the automotive industry, for example in the design of Anti-lock Braking Systems (ABS). In this case, the hardware consists in the ECU under test, and the vehicle dynamics and sensors are simulated.

The ACHILES test setup stands at the intersection between HIL simulation and (rapid) control prototyping. In this case, the complete control loop runs on the projectile prototype and no part of the system is emulated by an external computer. The free-flight is emulated through a *local* environment, formed by the wind tunnel airflow and the supporting structure. Bacic [2005] introduces the notion of *transparency* in a HIL simulation: in a perfectly transparent HIL simulation, the real system does not feel the difference between the HIL simulator and operation in real conditions. In the current version of the setup, this local environment is fixed and cannot be influenced by the simulation. The simulation transparency is then adequate for fixed-operating conditions studies but presents some limitations when considering complete flight trajectories.

This chapter is organized as follows: the test setup design goals and the overall ACHILES architecture are presented in Section 1.2. Section 1.3 describes the different hardware components in both the projectile prototype and the supporting environment, namely the airframe, the actuators, sensors, embedded computer, wind tunnel and support structure. In this section, the focus is mainly set on the

⁴An *edge case* means the system is operating outside of its nominal parameters or conditions.

custom-designed actuators, for which a detailed description is given. The counterpart of the hardware is the software environment, which consists in both embedded software and a development environment, presented in Section 1.4.

1.2 The ACHILES Test Setup

The main objective of the ACHILES testbench is to provide a rapid prototyping environment for the development of guided projectile control laws, the ultimate goal being the development of a proof of concept for the design methodology presented in the introduction. A synopsis of its architecture is given in Figure 1.2 and shows the interactions between the projectile prototype and the development environment.

The central component of the ACHILES setup is the projectile prototype. It is designed as an autonomous system, where the developed algorithms are executed on an embedded computer onboard and aims at reproducing the behavior of a free-flying projectile. The industry trends, as well as the Ministry of Defense requirements, tend towards the use of low-cost actuators and sensors for guided ammunition applications. These components are subject to harsh conditions in this application, mainly due to accelerations at gun launch on the order of ten thousand g s [Fresconi et al., 2013]. In the present case, the acceleration survivability was not retained as a design criteria since this projectile is not meant to be used in free-flight tests. As the presented setup is a first iteration of the concept, the projectile should present a stable open-loop behavior, while providing sufficient control authority.

The key goal of the ACHILES software environment is to provide a *rapid control prototyping* environment, enabling the user to quickly test and iterate on control strategies. Such an environment provides a seamless transition from controller simulation using a numerical model to HIL testing, allowing shorter development-to-test cycles. In this regard, industry-standard tools such as MATLAB/Simulink offer code generation functionalities, which relieve the user from manual implementation of the control algorithms in the HIL embedded computer. The identification and control algorithms are developed in the MATLAB/Simulink environment and deployed on the embedded computer using code generation toolboxes (see Section 1.4.2). The resulting program consists of a main task corresponding to the Simulink diagram and support tasks for interacting with the projectile hardware.

In respect to the HIL nature of this setup, these tasks are executed in real time to guarantee the execution time of the algorithms. The simulation program must guarantee a deterministic, bounded response time within a sampling period. In order to streamline the simulation process, an embedded

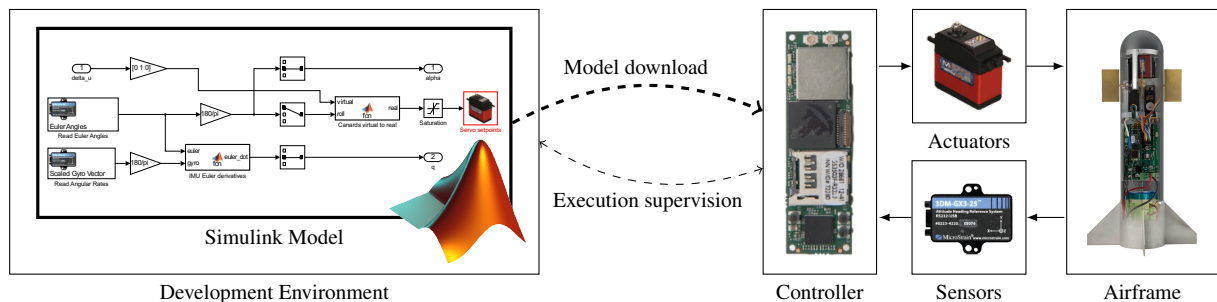


FIGURE 1.2: Overview of the ACHILES test setup hardware and software architecture.

operating system is preferable over a standalone programming approach⁵ since it provides a complete programming interface with multi-tasking support, as well as hardware drivers, standard protocols and interfaces, such as TCP/IP. Combined with the above mentioned real-time constraint, this application mandates the use of a real-time operating system (RTOS), which is presented in Section 1.4.1.

1.3 Hardware Components

1.3.1 Mechanical Structure and Aerodynamics

The ACHILES projectile airframe is based on a missile-like projectile architecture, consisting of an 80mm caliber, 400mm long outer shell with aerodynamic control and stabilization surfaces. The various hardware components are placed on the internal structure shown in Fig. 1.3, and are placed around the projectile's longitudinal axis so as to minimize imbalance. The internal structure enables fine adjustment of the projectile center of gravity by moving the central sled with an externally accessible screw.

As presented in the introduction, there are several technological solutions for both airframe control and stabilization. In the present case, aerodynamic stabilization was preferred over gyroscopic stabilization. The high roll rate in the latter case induces inter-axis coupling, which prevents independent studies of each axis, and also imposes several constraints on the mechanical design, as the rolling motion must be maintained (e.g. using a motor). On the other hand, a projectile is aerodynamically stable when its center of gravity (CoG) lies ahead of the aerodynamic center of pressure. The position of the latter is influenced by the geometry of aerodynamic surfaces and can be moved towards the aft with the addition of tail fins. This missile-like configuration is much simpler mechanically and results in an open-loop stable projectile even at null roll rate.

Guided projectiles may be steered using aerodynamic surfaces, impulse thrusters, or inertial masses. In the present case, the aerodynamic surfaces solution is retained for its ease of implementation, and consists of four independently actuated canards located at the front of the shell. The sizing of these

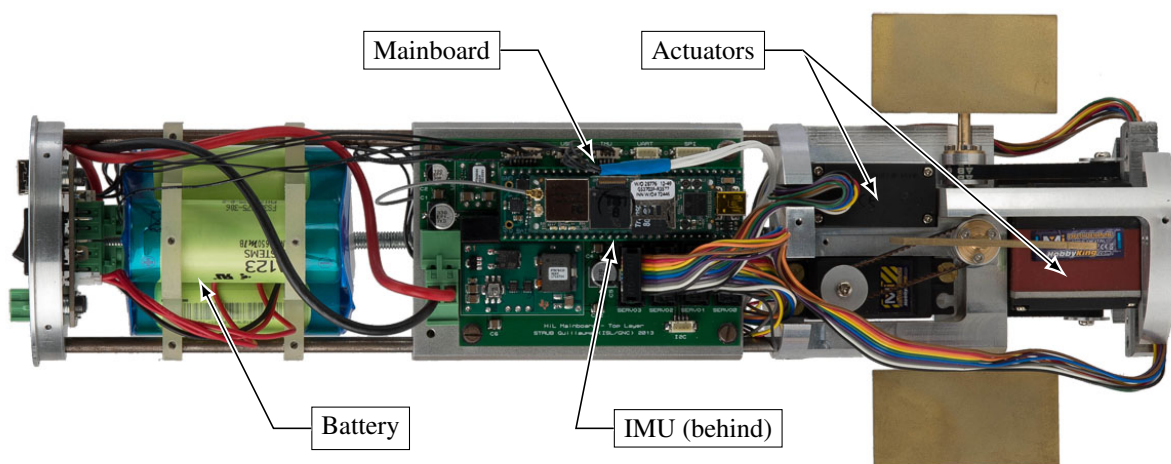


FIGURE 1.3: Internal view of the ACHILES projectile exposing the structure and hardware components.

⁵The program directly runs on the target hardware without an operating system

surfaces is the result of a compromise between open-loop stability and control authority. Indeed, larger canards produce a higher steering moment but tend to shift the aerodynamic center forward, which decreases the airframe stability. The canards are mounted at a 45° angle relative to the tail fins (see Figure 1.4) in order to minimize aerodynamic interaction between the two sets of surfaces. Tests conducted with both surface sets aligned showed a decrease in stability, which is due to canard trailing vortex interactions with the tail fins [Silton and Fresconi, 2015].

The projectile's mass, center of gravity position and moments of inertia have been measured using a commercial Space Electronics measurement system and are presented in Table 1.1. These results show a homogeneous mass distribution with identical pitch and yaw moments of inertia, while the center of gravity is only off the main axis by a negligible amount.

TABLE 1.1: ACHILES projectile mass properties

Mass (g)	Center of Mass (mm)			Moments of Inertia ^a (kg.m ²)		
	x_{CG}^b	y_{CG}	z_{CG}	I_{xx}	I_{yy}	I_{zz}
2282.5	168.72	0.07	-0.15	$3.801 \cdot 10^{-3}$	$3.353 \cdot 10^{-2}$	$3.346 \cdot 10^{-2}$

^aAt the projectile's center of mass

^bRelative to the aft

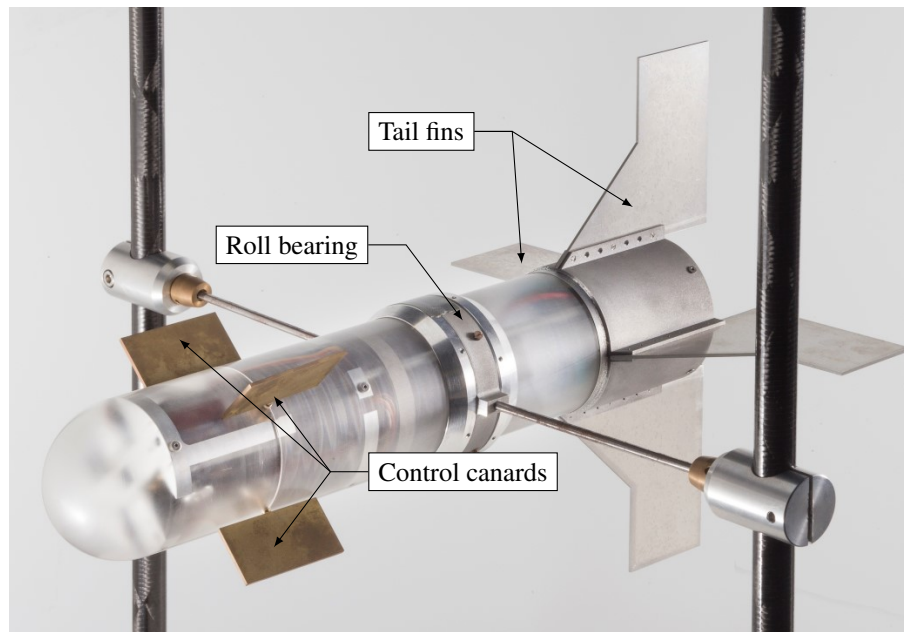


FIGURE 1.4: The ACHILES projectile prototype installed in the support structure.

1.3.2 Actuators

In the ACHILES setup, the actuators are responsible for driving the guidance canards with respect to commands provided by the embedded algorithms. For this task, hobby radio control servomotors were selected for their wide availability, very reasonable cost and standardized dimensions. Each actuator is mounted on the internal structure with an offset from the canard axis and is linked to the corresponding canard through a toothed belt drive with a 16/18 reduction ratio, as shown in Figure 1.5.

However, as hobby servomotors were initially designed for remote control of model aircraft, there is no provision for measuring the output shaft position, nor acting on the control algorithm parameters. Indeed, these actuators aim at tracking a reference angle change as fast as possible, without concern for saturations and the linearity of their response. Furthermore, the accuracy of standard hobby servomotors is usually limited due to the use of low-cost potentiometers as the main position sensor, and this information is seldom indicated on servo datasheets. For these reasons, a custom actuator has been developed, based on the mechanical parts of a standard servomotor and a purpose-built control board.

The actuator design objectives were set as follows: first, the communication between the servomotor and the embedded computer should be bidirectional. This enables reporting of the achieved angular position over time, setting the home position of the actuator and fine-tuning of the controller gains. Secondly, the output position should be accurate, with trueness⁶ better than $\pm 0.5^\circ$ and repeatability under $\pm 0.1^\circ$ ⁷. Finally, the control laws should be user-adjustable, and provide a linear response for any reference step under $\pm 30^\circ$ in amplitude. This last requirement allows the servomotor to be modeled as an LTI system, which greatly simplifies the design of projectile control laws.

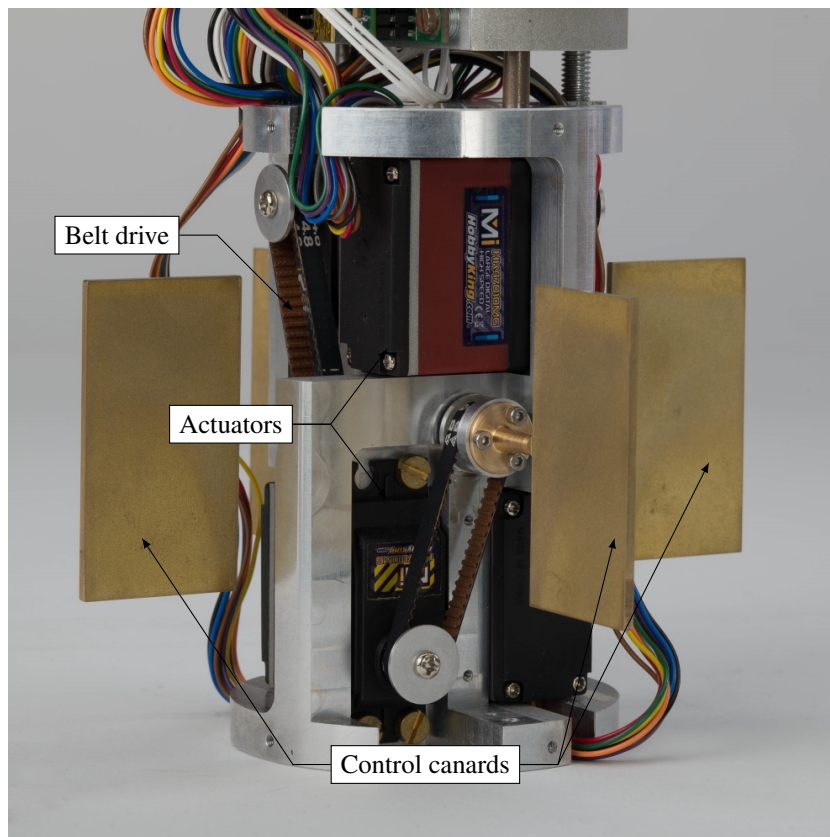


FIGURE 1.5: The ACHILES control canards and driving actuators.

⁶*Trueness*, or *accuracy*, designates the proximity of a measurement to the true value (systematic error).

⁷*Repeatability*, or *precision*, relates to the variation of a measurement repeated under identical condition (random error).

1.3.2.1 Hardware Design

The overall architecture of the ACHILES actuator is presented in Figure 1.6. The motor drives the output shaft through a geared reduction mechanism, which increases the available torque at the expense of speed. The position sensor measures the output shaft angle y , which is used by the control algorithm to compute the error between the actual and target positions. This algorithm is implemented on a microcontroller, which communicates with the embedded computer over a bidirectional I²C bus. The controller output is the PWM control signal u_{PWM} , which is amplified by an H-bridge amplifier and applied to the motor. In this case, the average motor voltage is proportional to the duty cycle of the PWM signal.

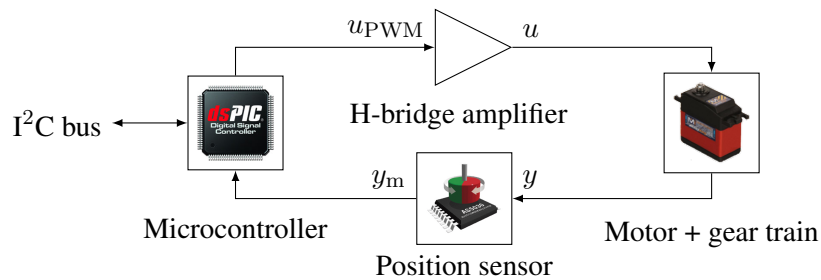


FIGURE 1.6: ACHILES actuator architecture.

The accuracy of the actuator is entirely defined by the position sensor. It is located at the output shaft in order to avoid the effects of backlash in the geartrain. Absolute magnetic sensors are a good alternative to potentiometers as they provide excellent characteristics (in terms of linearity, repeatability, resolution) in a small footprint. They consist of two parts: a 2-pole, diametrically-polarized magnet is mounted on the output shaft, and the sensor integrated circuit (IC) is mounted on the chassis below the magnet. The IC combines an array of Hall sensors with an analog front-end and a digital signal processor (DSP) in a single package, and provides analog and/or digital readout of the measured angle. As there is no mechanical contact between the magnet and the IC, the reliability of the sensor is excellent.

A HobbyKing HK47010MG servomotor based on this principle was chosen as the mechanical base for the ACHILES actuators. This model features a coreless motor, metal gearbox and double shaft bearings, and the original sensor is replaced with an AMS AS5045 12-bit sensor [ams AG, 2010]. This resolution leads to a 0.088° quantization step, and the chosen sensor provides better than 0.03° repeatability and a maximum $\pm 0.5^\circ$ integral nonlinearity error (INL)⁸.

The servo control laws are implemented on a Microchip dsPIC30F3012 microcontroller. This 16-bit device has 24 KiB⁹ of Flash memory, 2 KiB of RAM and 1 KiB of EEPROM memory and provides up to 20 MIPS (Millions of Instructions Per Second) when running at 80 MHz [Microchip Technology, 2010]. Although this device specification may be oversized for the application, the development time was significantly reduced with the use of the C programming language and the large memory resources.

The microcontroller program performs several tasks: running the control loop, handling the communication bus and processing the command messages. Considering the actuator bandwidth, the control loop (highest priority task) runs at a fixed rate of 1 kHz, set by a hardware timer interrupt. The communication bus handler is also implemented as an interrupt routine and stores the received messages in

⁸The maximum difference between actual and measured values. This corresponds to the sensor's trueness.

⁹1 KiB equals 1024 bytes.

a buffer for processing by the message parser, which runs at a lower priority. Finally, the control loop can be turned off and the clock speed reduced when the actuators are not in use so as to reduce power consumption.

Communication is carried out over an I²C bus, which is a multi-master/multi-slave bidirectional serial bus supporting speeds from 100 kbit/s to 5 Mbit/s [NXP Semiconductors, 2014]. Each I²C device has a unique 7-bit address, thus up to 127 devices are supported on a single bus (address 0x00 is reserved for general calls)¹⁰. In the present configuration, the embedded computer acts as an I²C master and each of the 4 actuators is an I²C slave with a unique address, with all actuators sitting on the same bus. The actuators are accessed sequentially when setting the target position and reading the achieved position, while the general call address is used for commands which have to be executed by all servos at the same time (acquire position and define reference), as further explained in Section 1.4.1.2.

1.3.2.2 Control Design

The embedded servo controller is designed using the same principles that are applied to the complete guided projectile prototype in the subsequent chapters of this manuscript. Hence the reader is encouraged to consult these chapters for more details on the identification (Chapter 2) and control techniques (Chapter 3), as only the results will be discussed in this section. The overall controller structure is shown in Figure 1.7 and exhibits a cascaded structure, with an inner velocity regulation loop and an outer position control loop. Compared to a single position control loop, this structure showed a better compensation of friction inside the gear train.

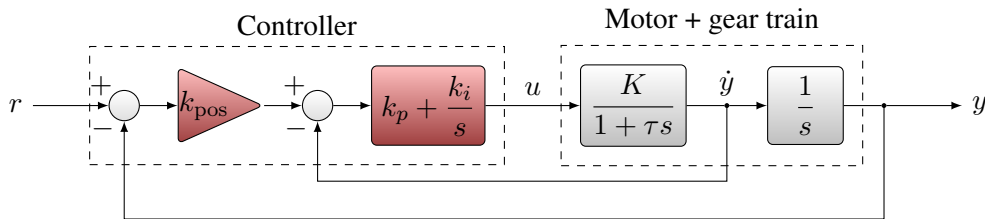


FIGURE 1.7: Servo controller structure.

Modeling and Identification The open-loop plant corresponds to the complete servo chain, which consists in the PWM modulator, H-bridge amplifier, DC motor, gear train and the absolute magnetic position encoder (Figure 1.6). In this chain, the PWM modulator and H-bridge acts as a power digital-to-analog converter that provides the motor with a voltage proportional to the discrete controller output. As the slowest part in the servo chain is the mechanical time constant of the motor, the open-loop velocity dynamics are modeled as a first-order system [Butler et al., 1989]:

$$\mathbf{G}_{A,OL}(s) = \frac{K}{1 + \tau s} \quad (1.1)$$

The gain K and time constant τ were determined by Recursive Least Squares (RLS) estimation using experimental data. The experimental procedure consists in exciting the actuator with a piecewise-constant random input signal $u(t)$ and observing the sensor measurement $y_m(t)$. As this is a sampled-data system, the first step consists in discretizing the open-loop model (1.1). In the controller implementation, the output (PWM duty cycle) is held constant between sampling instants (zero-order hold),

¹⁰Greater capacity can be achieved using the 10-bit addressing variant and/or port expanders.

leading to the discrete open-loop plant:

$$\mathbf{G}_{A,OL}(z) = (1 - z^{-1})\mathcal{Z} \left\{ \frac{1}{s} \frac{K}{1 + \tau s} \right\} \quad (1.2)$$

$$= \frac{k_1 z^{-1}}{1 - k_2 z^{-1}} = \frac{y(z)}{u(z)} \quad (1.3)$$

with $k_1 = K(1 - e^{-T_s/\tau})$ and $k_2 = e^{-T_s/\tau}$, where T_s is the sampling period.

The RLS technique consists in computing the least squares estimates of the model parameters k_1 and k_2 *recursively*, instead of collecting data and performing the estimation afterwards [Simon, 2006; Wellstead and Zarrop, 1991; Krneta et al., 2005]. This algorithm was applied to the open-loop servomechanism using a sampling period of 5 ms. This period is the result of a compromise between sampling bandwidth and sensitivity to noise, since the velocity signal is derived numerically from the position sensor measurements. Under these conditions and after conversion to the continuous domain, the following parameter estimates were obtained for a 7 V supply voltage: $K = 326.2$ °/s/V and $\tau = 0.0182$ s.

Controller Design and Synthesis The servo controller is designed around the two nested loops shown in Fig. 1.7. The inner velocity control loop is based on a proportional-integral (PI) controller, while the outer position controller consists in the proportional gain k_{pos} , giving the continuous transfer function:

$$u(s) = \left(k_p + \frac{k_i}{s} \right) [k_{pos} \quad -1] \begin{bmatrix} r(s) - y_m(s) \\ \dot{y}_m(s) \end{bmatrix} \quad (1.4)$$

The closed-loop transfer function of the actuator is then given by:

$$\mathbf{G}_A(s) = \frac{K k_{pos}(k_p s + k_i)}{\tau s^3 + (1 + K k_p)s^2 + K(k_i + k_{pos} k_p)s + K k_{pos} k_i} \quad (1.5)$$

and a state-space representation of \mathbf{G}_A is given by:

$$\mathbf{G}_A \stackrel{ss}{=} \left[\begin{array}{c|c} \mathbf{A}_A & \mathbf{B}_A \\ \hline \mathbf{C}_A & \mathbf{D}_A \end{array} \right] = \left[\begin{array}{ccc|c} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline -K k_{pos} k_i & -K(k_i + k_{pos} k_p) & -\frac{1}{\tau}(1 + K k_p) & \frac{1}{\tau} \\ \hline K k_{pos} k_i & K k_{pos} k_p & 0 & 0 \end{array} \right] \quad (1.6)$$

The three controller gains k_{pos} , k_i and k_p are determined using a structured \mathcal{H}_∞ design procedure [Apkarian and Noll, 2006], which is presented in greater detail in Chapter 3. The synthesis aims at obtaining a closed-loop response similar to a second-order system \mathbf{T}_{ref} with bandwidth $\omega_0 = 50$ rad/s and damping $\zeta = 0.78$ (fastest 2% response). An additional constraint is that the motor control voltage must remain under the supply voltage of 7 V for a reference step of $\pm 30^\circ$, in order to avoid saturation.

These constraints are implemented as weighting filters in a mixed sensitivity and model-matching formulation, which is illustrated in Figure 1.8. The weighting filters \mathbf{W}_S , \mathbf{W}_{KS} and \mathbf{W}_{MM} respectively act on the position error transfer function $\mathbf{S}_{r \rightarrow e_p}$, control transfer $\mathbf{T}_{r \rightarrow u_v}$ and model-matching transfer $\mathbf{T}_{r \rightarrow e_{MM}}$, and are selected according to the principles presented in Chapter 3. The structured \mathcal{H}_∞ optimization problem then consists in finding the controller gains k_{pos} , k_i and k_p such that the system is stable and

$$\|\mathbf{T}_{w \rightarrow z}\|_\infty < \gamma, \quad \gamma > 0 \quad (1.7)$$

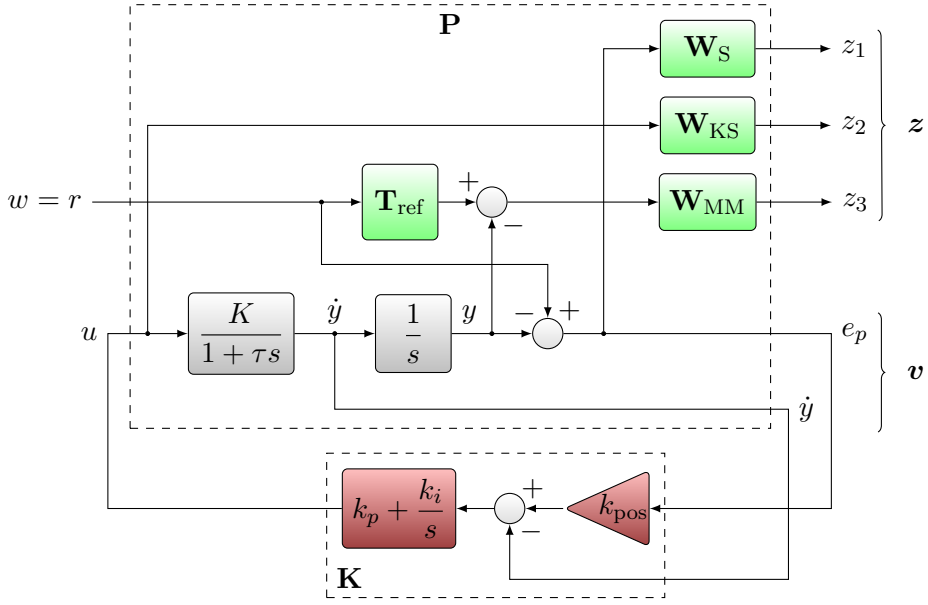


FIGURE 1.8: Mixed-sensitivity S/KS and model-matching servo controller synthesis diagram in standard form.

where $\mathbf{T}_{w \rightarrow z}$ is the global exogenous input to performance output transfer obtained by the lower linear fractional transformation (LFT) $\mathbf{T}_{w \rightarrow z} = \mathcal{F}_l(\mathbf{P}, \mathbf{K})$. Applying the structured \mathcal{H}_∞ synthesis procedure yields the following gain values:

$$k_{\text{pos}} = 29.4, \quad k_i = 0.3941, \quad k_p = 0.052$$

The cascaded controller (1.4) is implemented numerically in the microcontroller firmware with a control update rate set to 1 kHz. This control period is small *w.r.t.* the open-loop system bandwidth, hence the continuous controller may be directly converted to a discrete equivalent instead of performing a complete redesign in the discrete domain [Åström and Wittenmark, 2011]. The integrator in the velocity controller is approximated with the backward Euler equivalent $\frac{zT_s}{z-1}$, while the velocity is estimated using a numerical derivative, yielding the discrete controller transfer function:

$$u(z) = \left(k_p + \frac{k_i T_s z}{1-z} \right) \begin{bmatrix} k_{\text{pos}} & -\frac{z-1}{T_s z} \end{bmatrix} \begin{bmatrix} r(z) - y_m(z) \\ y_m(z) \end{bmatrix} \quad (1.8)$$

The Bode magnitude diagrams and simulated step responses of the reference model as well as the continuous and discrete closed-loop systems are visible in Figure 1.9. The responses achieved with the continuous and discrete controllers are fairly similar and present a 5 % settling time of 70 ms. These responses are close to the desired response but do not present overshoot as they contain a real dominant pole at 40 rad/s.

Implementation and Results An important aspect of the controller implementation on the real actuator is the presence of saturations, due to the finite supply voltage. An increase of the controller output above the supply voltage introduces integral error buildup, or windup, which can lead to overshoot or instability when the system stops saturating. To avoid this issue, a simple anti-windup scheme consists in bounding the integral error variable to the supply voltage.

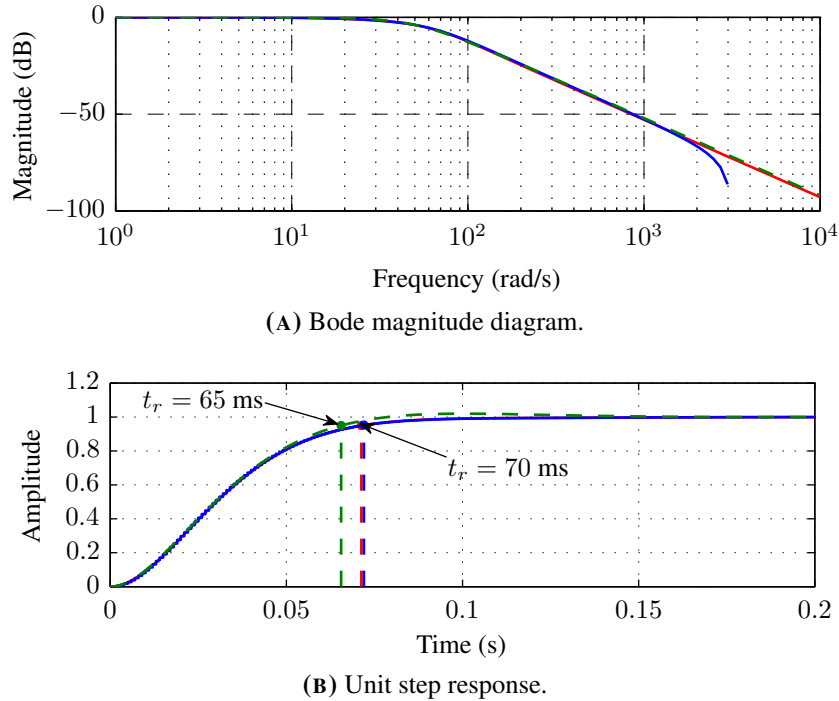


FIGURE 1.9: Bode diagram and unit step response of the continuous (solid red) and discrete (solid blue) closed-loop systems, and the reference model (dashed green).

The response of the implemented digital controller to various step signals is given in Figure 1.10. In this figure, the actuator reference signal consists in pulses of various amplitudes, and the measured output is normalized to 1 for comparison. The time response of the actuator transfer function \mathbf{G}_A for a unit pulse signal is shown in dashed thick black for reference. For angular amplitudes under 60° , the measured responses are almost indistinguishable and follow the simulated response closely, with the same 5% response time of 70 ms. Therefore the actuator behavior can be accurately modeled by the transfer function \mathbf{G}_A for angular steps smaller than 60° .

For steps larger than 60° , the actuator does not behave linearly anymore due to saturation: the controller commands a motor voltage higher than the available supply voltage. This results in a gentler rising slope and a higher response time, as the maximal velocity is bounded. However the response does not present overshoot when the controller output no longer saturates, thanks to the implemented anti-windup scheme.

1.3.3 Sensors

The guidance, navigation and control algorithms require a precise knowledge of the projectile's state vector, which is measured using onboard sensors. The state vector typically consists in the projectile attitude (orientation), angular rates, accelerations, velocities, and absolute position. These quantities are measured and/or estimated using an Attitude and Heading Reference System (AHRS) or an Inertial Measurement Unit (IMU), usually associated to a Global Navigation Satellite System (GNSS) for absolute positioning [Grewal et al., 2013]. Depending on the guidance scheme, imaging sensors may be used to locate the target, e.g. using a four-quadrant seeker with a laser designator on the ground [Katulka et al., 2008].

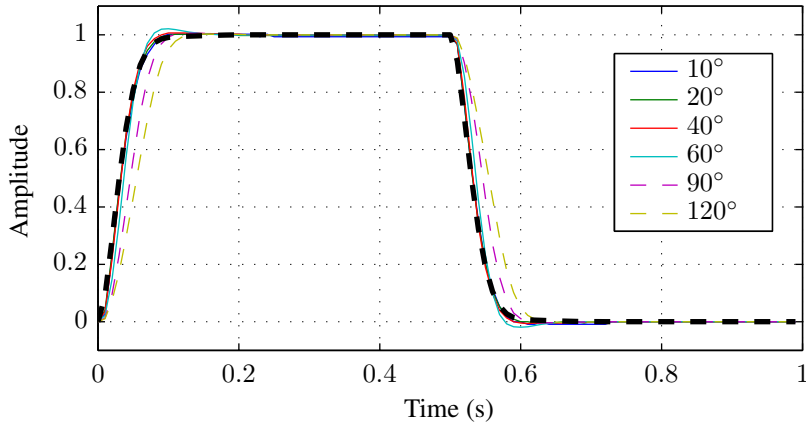


FIGURE 1.10: Measured and normalized actuator time response for a set of reference angles. Thin lines: measured responses, thick dashed line: simulated response of the actuator transfer function G_A .

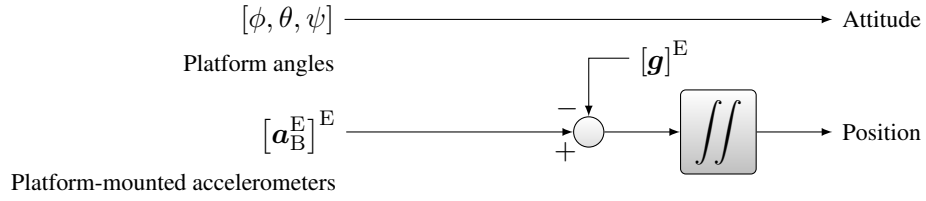
The ACHILES setup focuses on the control part of the GNC functionalities, while the support structure restrains the projectile's linear motion. As the center of mass remains static, there is no need for measuring the absolute position of the projectile, leaving only the angular motion quantities and the relative wind velocity to be measured. These quantities are measured using an onboard Inertial Measurement Unit with integrated data fusion, and an external Pitot tube and yaw encoder (see their location in Figure 1.1).

1.3.3.1 Inertial Measurement Unit

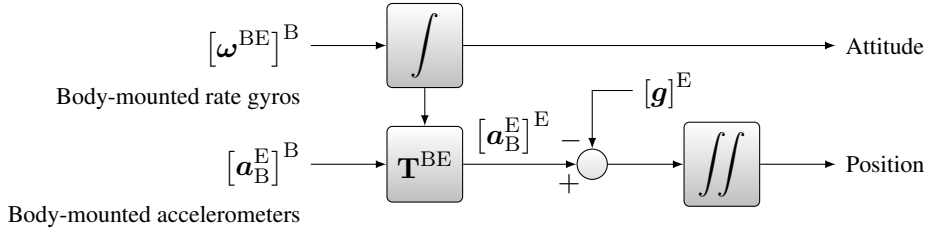
An inertial measurement unit (IMU) is a navigation device based on triplets of orthogonal accelerometers, rate gyroscopes, often associated with magnetometers. These sensors respectively measure linear acceleration, angular velocity and magnetic bearing. The orientation and position of an object, relative to a starting point, can be determined using inertial navigation, or *dead reckoning*, which consists in integrating the measured accelerations and angular variations [Woodman, 2007].

Inertial measurement units are used in a broad range of applications, and are typically used in naval and air vehicle navigation, including ships, aircraft, spacecraft and missiles. A recent breakthrough in inertial navigation devices comes from the MEMS (Micro ElectroMechanical System) semiconductor technology, which enables the manufacturing of single-chip, solid-state inertial sensors. The resulting availability of low-cost, lightweight and small inertial measurement units has broadened the range of applications of inertial navigation, including motion capture devices used by the film industry, smartphones, gaming devices and light UAVs. IMUs can be divided in two categories depending on the integration in the vehicle: stable platform and strapdown [Titterton and Weston, 2004; Barbour and Schmidt, 2001].

Stable platform IMUs In stable platform systems, the sensors are mounted on a stabilized platform kept in alignment with the Earth coordinate system. This platform is isolated from the vehicle's motion using a gimbal mount, which allows three rotational degrees of freedom. Angular deviations of the platform are detected by the platform-mounted gyroscopes. They are canceled out using motors to rotate the gimbal, effectively maintaining constant alignment of the platform with the global frame. The object's orientation is directly obtained using angular sensors which measure the gimbal angles, while position is obtained by double integration of the accelerometer readings (after subtraction of the vertical gravity



(A) Stable platform inertial navigation.



(B) Strapdown inertial navigation.

FIGURE 1.11: Overall principle of (A) stable platform inertial navigation and (B) strapdown inertial navigation. E denotes the Earth frame, B denotes the body-attached frame and the $[\cdot]^X$ notation expresses projection in the coordinate system X. $[g]^E$ is the gravity vector expressed in the Earth coordinate system.

vector), as shown in Figure 1.11A. Stable platform IMUs are mechanically complex devices and are thus usually large and expensive. However, they deliver excellent precision and the required computation power is low. These systems are typically used in heavy vehicle applications, where the very high estimated position precision is essential, as in aircraft, ships and missiles.

Strapdown IMUs In strapdown IMUs, the sensors are rigidly mounted on the vehicle body and thus measure quantities in the vehicle's coordinate system. The orientation of the vehicle is obtained by integration of the gyroscope measurements. The accelerometer readings are first projected into the Earth coordinate system using the estimated orientation, then doubly integrated to estimate the vehicle's position, as shown in Figure 1.11B. Compared to stable platform systems, strapdown IMUs are very simple mechanically, with little or no moving parts. This greatly reduces the cost and footprint, but come at the cost of much higher computational power requirements. The obtained precision is also less than stable platform systems, due to the extra integration step in the orientation calculation. Thanks to the decreasing costs of computational power, these sensors are widespread in all fields requiring a low-cost, low-footprint navigation solution, such as UAVs and motion capture systems.

Microstrain 3DM-GX3-25 IMU The choice of an inertial measurement unit for the ACHILES project was mostly constrained by the available footprint and the cost factor, since guided projectiles are typically based on low-cost actuators and sensors. For these reasons, a Microstrain 3DM-GX3-25 IMU strapdown-type IMU based on MEMS sensors was selected for this task. This device is relatively low-cost (~ 2000 euro), lightweight (18 g) and has a small footprint. It is equipped with MEMS accelerometers, rate gyros and magnetometers, and contains a data fusion algorithm for attitude estimation and drift compensation.

In this application, the most critical measurements are the orientation and angular velocities, which are derived from the rate gyro measurements. MEMS rate gyros are based on the effect of the Coriolis force on micro-machined silicon vibrating elements. Despite their many advantages, including low cost,



FIGURE 1.12: ACHILES Inertial sensors. Left: Microstrain 3DM-GX3 IMU, Right: MBDA Germany Mini-IMU (not used in this study).

high reliability, ruggedness and small size, these devices have limited accuracy compared to mechanical and optical gyroscopes due to high angular random walk and bias instability [Grewal and Andrews, 2010]. For these reasons, they are associated to other types of sensors (accelerometers and magnetometers) in order to provide an absolute reference to readjust the orientation estimate and compensate angular drift. Therefore, even if we are only interested in the orientation, the IMU should be positioned as close as possible to the projectile's center of mass (which is also its center of rotation) so as to avoid measuring rotation-induced accelerations. The Microstrain IMU is visible in Figure 1.12 and its characteristics are summarized in Table 1.2.

TABLE 1.2: IMU Specifications [MicroStrain Inc., 2012]

	Accelerometers	Rate gyros	Magnetometers
Measurement range	$\pm 5 g$	$\pm 300^\circ/s$	± 2.5 Gauss
Non-linearity	$\pm 0.1 \% fs$	$\pm 0.03 \% fs$	$\pm 0.4 \% fs$
Bias stability	$\pm 0.04 mg$	$18^\circ/h$	–
Initial bias error	$\pm 2 mg$	$\pm 0.25^\circ/s$	$\pm 3 mGauss$
Scale factor stability	$\pm 0.05 \%$	$\pm 0.05 \%$	$\pm 0.1 \%$
Noise density	$80 \mu g/\sqrt{Hz}$	$0.03^\circ/s/\sqrt{Hz}$	$100 \mu Gauss/\sqrt{Hz}$
Alignment error	$\pm 0.05^\circ$	$\pm 0.05^\circ$	$\pm 0.05^\circ$
Bandwidth	225 Hz	440 Hz	230 Hz
Sampling rate	30 kHz	30 kHz	7.5 kHz

The IMU provides orientation estimates in quaternion, rotation matrix or Euler angles format. As the quaternion form is used internally, there are no gimbal lock issues [Diebel, 2006]. The inertial navigation integrals are updated at a 1 kHz rate, and the estimates can be retrieved at a user-definable rate between 1 Hz and 1 kHz. Communication with the IMU takes place over an RS232 serial link at a baudrate up to 921600 baud using a message-based protocol. Two operating modes are provided by the IMU. In the *polling* mode, the embedded computer requests the most recent data from the IMU, while in the *continuous mode* the IMU periodically sends the latest estimates at the user-defined data rate. The latter mode is preferred, as it guarantees the lowest delay between the acquisition of inertial sensor values and the availability of the estimates on the embedded computer.

1.3.3.2 External Sensors Acquisition Interface

Data measured by the external sensors, namely the Pitot tube and the yaw encoder (visible in Figure 1.1), are gathered with a custom-built acquisition interface shown in Figure 1.13. This interface transmits the data to the projectile using a wireless link, and is built around a Microchip dsPIC33FJ128MC804 microcontroller and a Nordic Semiconductor nRF24L01+ 2.4GHz radio. The microcontroller acquires the angular data from the yaw encoder using an integrated quadrature encoder interface, and the differential pressure from the Pitot tube is digitized using an Amphenol NPA-700B-10WD digital pressure sensor interfaced on the microcontroller's I²C bus. An USB-to-Serial interface allows the user to fine-tune the interface parameters, such as the radio channel and address, and the sensor offsets.

The radio link is built upon two nRF24L01+ transceivers, one on the acquisition board and the other in the projectile prototype. These modules provide a fast, low-latency radio link with a maximum radio data rate of 2 Mbit/s and are easily interfaced with the host processor (dsPIC on the acquisition board, Gumstix in the projectile prototype) using the SPI bus. However, despite the short distance between the two radio modules, packet loss may occur at about 1 packet dropped ever 100 packets sent. This limitation has thus to be taken into account when using the external sensor data.

1.3.3.3 Yaw Encoder

As the projectile evolves in a severely disturbed magnetic environment (presence of large steel masses, magnetic encoder in the actuators), the IMU magnetometer cannot be relied upon to generate an absolute heading reference (North vector). Moreover the gyrometers are subject to drift, and the yaw angle thus cannot be estimated using simple integration. To this end, an absolute external reference is provided by an incremental encoder mounted on the support structure's yaw axis. This encoder has a 2000 optical step/turn resolution, which amounts to 8000 edges per revolution using quadrature decoding. As an

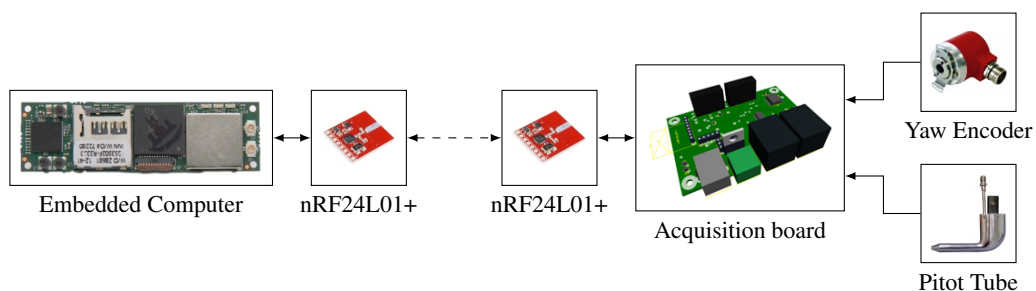


FIGURE 1.13: Interfacing of external sensors with the projectile prototype.

incremental encoder only provides a relative position measurement, the angular position of the yaw gimbal is not known when the acquisition interface is powered on. The encoder provides an index signal, which is active only one step per turn. This signal is employed to determine the axis origin at system startup in a procedure known as *homing*.

The encoder measurements are transmitted wirelessly to the embedded computer thanks to the external acquisition interface. However, this measurement cannot be used directly, since packet loss has a direct impact on the control algorithms. Furthermore, the encoder resolution leads to a 0.045° quantization step, and thus the numerical derivative of the yaw angle (e.g. when used in a PID scheme) may become noisy. To overcome these shortcomings, the yaw encoder and the IMU-derived yaw rate measurements are combined using a complementary filter. This simple type of filter corresponds to a steady-state Kalman filter, where no detailed description of the noise processes are considered [Higgins, 1975]. A complementary filter operates on two measurements x and y of a same quantity z , where x is affected by low-frequency noise and y is affected by high-frequency noise. An estimate \hat{z} of z is obtained by filtering y with a low-pass filter $G(s)$ and x with the complementary high-pass filter $(1 - G(s))$, and summing the result.

The complementary filter scheme is presented in Figure 1.14. The yaw encoder measurement ψ_Q is subject to high-frequency noise induced by the quantization steps, while the gyro yaw rate $\dot{\psi}_G$ is subject to low-frequency bias variations and white noise. The integration of the yaw rate $\dot{\psi}_G$ attenuates the high-frequency noise and gives ψ_G , which is subject to drift due to the gyro bias. The yaw encoder measurement ψ_Q is then fed through a first-order low-pass filter with time constant τ , while ψ_G is fed to the complementary high-pass filter, giving the yaw estimate $\hat{\psi}$. By combining the integrator and the high-pass filter, the complementary filter equation is written as:

$$\hat{\psi} = \frac{1}{\tau s + 1} (\tau \dot{\psi}_G + \psi_Q) \quad (1.9)$$

This filter equation is then discretized using the backward Euler approximation $s = \frac{z-1}{zT_s}$ [Franklin et al., 1998] with sampling time $T_s = 0.01$ s, yielding the difference equation:

$$\hat{\psi}_k = (1 - f) (\hat{\psi}_{k-1} + T_s \dot{\psi}_{G_k}) + f \psi_{Q_k} \quad (1.10)$$

where $f = \frac{T_s}{T_s + \tau}$ is the filter parameter. The optimal value of τ was found to be 5 seconds during initial testing, and corresponds to the usual range (2 to 6 seconds) found in the literature [Higgins, 1975]. This difference equation is implemented in the embedded software source code. When no yaw encoder data is available due to packet loss, f is set to zero and $\dot{\psi}_G$ is directly integrated.

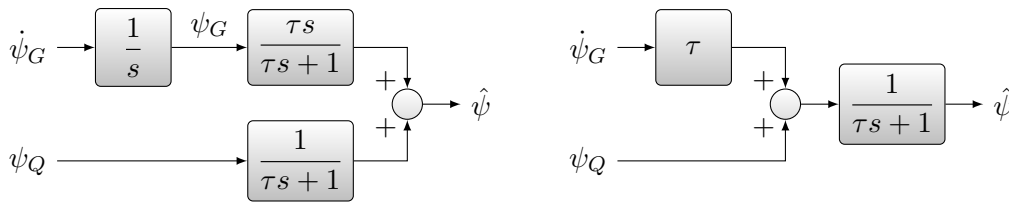


FIGURE 1.14: Yaw angle complementary filter. Left: basic complementary filter with integrator. Right: actual continuous-time realization.

1.3.3.4 Pitot Tube

The second external sensor used in the ACHILES setup is a Pitot-static tube, or Prandtl antenna, which is installed in the wind tunnel test section. This device is composed of a flow-facing tube and one or more static pressure ports perpendicular to the flow direction, as shown in Figure 1.15. The pressure inside these ports corresponds to the ambient, or *static* pressure. The innermost tube is open at its forward-facing extremity. As there is no outlet for the flow to exit the tube, the pressure inside this tube corresponds to the *total* pressure, which is the sum of the *static* pressure and the velocity-dependent *dynamic* pressure. A differential pressure sensor measures the difference of pressure between these two tubes, which is the *dynamic* pressure. The air velocity is then obtained with the following equation:

$$v = \sqrt{\frac{2(p_t - p_s)}{\rho}} \quad (1.11)$$

where v is the air velocity, p_t is the total pressure, p_s is the static pressure and ρ is the air density (1.2 kg/m^3 at 20° at sea level). In our application, the *dynamic* pressure ($p_t - p_s$) is measured using an Amphenol NPA-700B-10WD I²C differential pressure sensor. This digital sensor is temperature-compensated and provides a 14-bit resolution over a $\pm 10 \text{ inH}_2\text{O}$ range ($\pm 2490 \text{ Pa}$), corresponding to a maximum measurable speed of 64 m/s.

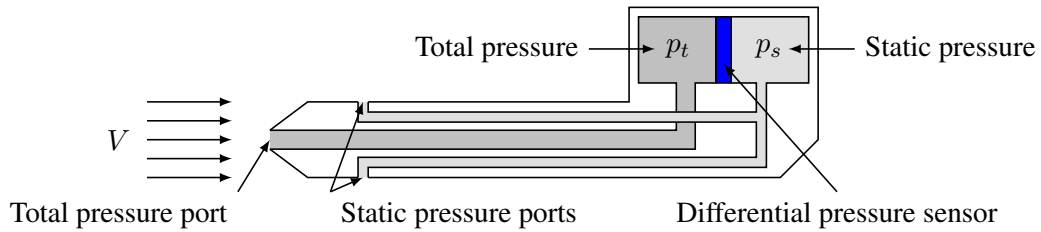


FIGURE 1.15: Pitot-static tube with static and total pressure ports.

1.3.4 Embedded Computer and Support Electronics

The control algorithms are executed on an onboard embedded computer, which acquires the sensor measurements and drives the actuators accordingly. This fully autonomous approach is common in the field of mobile robotics, where it may be complex to offload computation on a separate computer: physical tethers may be impractical or unfeasible, and wireless links introduce latency, which has negative impact on control loop performance and stability. In any case, a communication channel is necessary to upload the algorithm executable code, monitor the Hardware-In-the-Loop simulation and download the recorded data. To this end, Linux-based solutions reduce the development time since they provide standard protocols and functions for general-purpose computing (e.g. file management, networking) which would otherwise require a custom implementation.

Embedded Linux computer Several embedded Linux-compatible computers-on-board are available on the market, using mostly microprocessors based on the ARM architecture, with processing power ranging from low-end routers to high-end quad-core smartphone processors and with prices starting as low as 25 euro (Raspberry Pi Model A). The ACHILES embedded computer is a Gumstix Overo FireSTORM board, which is based on a 800 MHz Texas Instruments DM3730 ARM Cortex-A8 processor with 512 MiB RAM. This module is easily integrated into the projectile shell thanks to its small

dimensions (58 mm x 17 mm x 4.2 mm). Contrary to most System-on-Chips (SOC) on embedded Linux boards, the Texas Instruments processor documentation is thorough and detailed [Texas Instruments, 2012], which is critical for writing custom device drivers. The Gumstix Overo FireSTORM supports several standard embedded serial bus protocols, with three available UARTs (asynchronous serial bus) ports, one I²C bus and one SPI bus. In the ACHILES prototype, these busses are allocated as follows: one serial port is dedicated to communication with Microstrain IMU, and another serial bus provides a console session. The I²C bus is employed for communication with the four actuators, and the SPI bus is connected to a nRF24L01+ module for establishing a wireless link with the external sensors acquisition board. The embedded computer is also equipped with a WiFi chipset, which is used for communication with the development computer for development, monitoring and data-logging purposes.

Main board The Gumstix embedded computer is installed on the projectile's main board, which is represented in Figure 1.16. In addition to providing mechanical support to the Gumstix, the main board also provides signal level translation, different power supplies as well as battery monitoring. The signal level translation enables the interconnection of the Gumstix CPU with the actuators and sensors. Indeed, the DM3730 input/output voltage is 1.8 V, whereas the actuators and sensors use higher voltage levels: the Microstrain IMU uses ± 12 V RS-232 signaling, the nRF24L01+ I/O voltage is 3.3 V and the actuators I²C bus uses 5 V signaling.

Power supply The ACHILES prototype is powered with a LiFePO₄ battery composed of three A123 ANR26650 cells in series, resulting in a 9.90 V nominal voltage and a 2.5 Ah capacity. The battery voltage is stepped down to 5 V for the Gumstix board, the Microstrain IMU and the actuator logic using a Texas Instruments PTH08080W switching regulator module. A second Texas Instruments PTH78020W regulator generates the 7.5 V rail for the actuator drive train. Finally, a switchable voltage divider enables battery monitoring using one of the Gumstix's analog inputs. The system can then warn the user when the battery runs low, and turn off the system to prevent battery overdischarge.

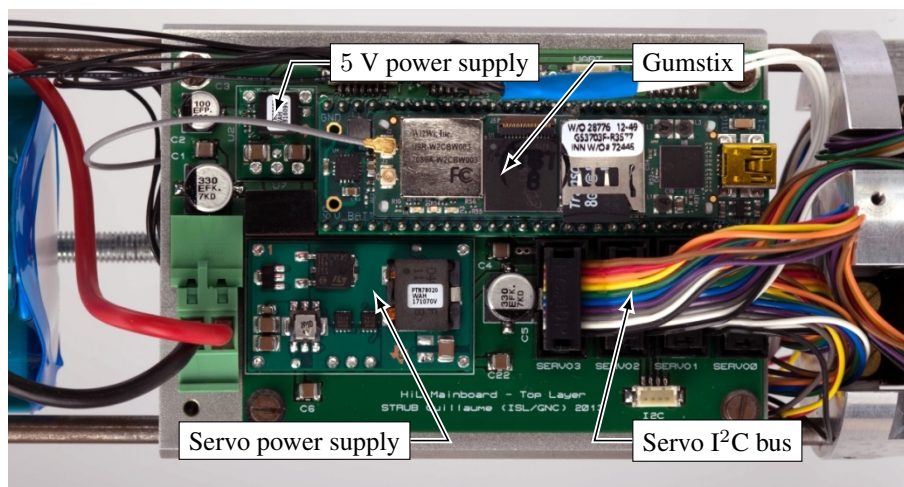


FIGURE 1.16: Gumstix Overo Computer-on-Module installed on ACHILES main board.

1.3.5 Wind Tunnel

In the ACHILES experimental setup, the projectile free-flight motion is emulated with the help of a wind tunnel. The French-German research institute of Saint-Louis has three wind tunnels in service: a subsonic (Mach 0 to 0.15) continuous wind tunnel with a test section of 70 cm x 90 cm, visible in Figure 1.17, a supersonic (Mach 1 to 4.5) blow-down wind tunnel with a test section of 20 cm x 20 cm and a trisonic blow-down wind tunnel (Mach 0.5 to 4.5) with a test section of 30 cm x 30 cm. Even though the flight speed of guided projectiles is usually in the high subsonic (Mach > 0.5) to supersonic range, we selected the subsonic wind tunnel to carry out this study. The larger test section enables the use of a full-size projectile prototype and simplifies the mechanical design. As the wind speed remains low, the support structure can be lighter than if a supersonic wind tunnel were employed. Finally, the continuous airflow enables long-duration identification and control experiments. In the case of blow-down wind tunnels, the experiment duration is usually limited to 1 or 2 minutes.

The ISL's subsonic wind tunnel was commissioned in 1979 and was initially dedicated to the study of two-dimensional flows on wing profiles using laser velocimetry [Jaeggy, 1982]. It is a closed-return design, where the air is recirculated through the 70 cm x 90 cm test section. The latter is open, which facilitates the installation of experimental setups. Air is set in motion using a 1.40 meter fan driven by a 40 kW DC motor for generating airflows up to 50 m/s. The motor is driven using a three-phase variable rectifier and regulator. In its current configuration, the airspeed is not regulated. Instead, the motor speed reference is set using a dial on the wind tunnel control panel. An overall schematic drawing of the wind tunnel is given in Figure 1.18.

1.3.6 Support Structure

The projectile prototype is installed in the test section of the subsonic wind tunnel by the means of a support structure presenting three angular degrees of freedom, which is presented in Figure 1.19. This structure consists in a support frame and two concentric gimbals.

The outer gimbal frame rotates along the vertical axis, which corresponds to the yaw axis in the flight mechanics Euler angles convention (see Chapter 2). The rotation of this frame relative to the chassis is

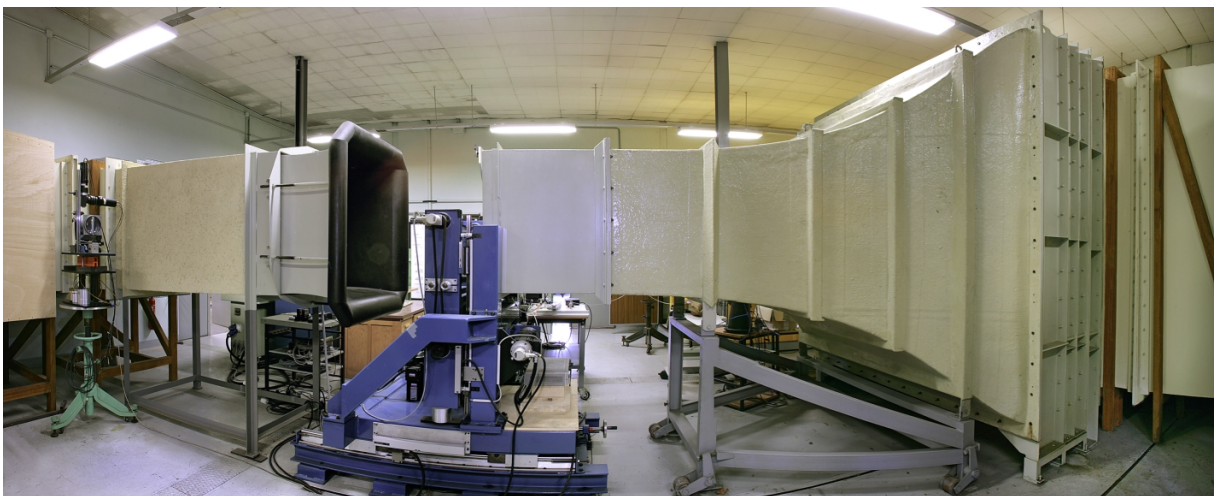


FIGURE 1.17: Panoramic view from inside the subsonic wind tunnel loop.

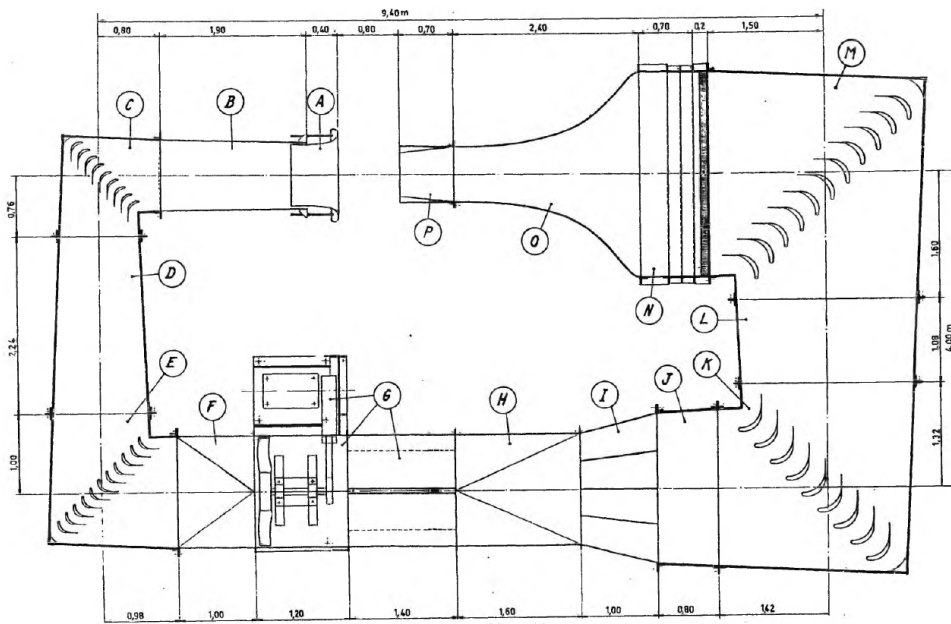


FIGURE 1.18: ISL subsonic wind tunnel schematic drawing [Jaeggy, 1982].

measured using the incremental encoder introduced in Section 1.3.3.3. The outer gimbal has a large size of about $1\text{ m} \times 1\text{ m}$ so that it does not interfere with the airflow. Nevertheless, this gimbal has a reasonable moment of inertia along the vertical axis thanks to the selected materials (aluminium bracing and carbon fiber tubing), and thus does not modify significantly the yaw dynamics.

The inner gimbal frame rotates along the horizontal axis of the outer gimbal frame, corresponding to the Euler pitch angle in the chosen convention. It consists in a central ring and two lateral axes, which connect to the outer gimbal through ball bearings. These axes are put under tension so as to limit sag on the central ring. However, due to the 180° angular spacing of the anchor points, sagging is unavoidable. An 80 mm ultra-thin ball bearing connects the projectile prototype to this central ring. The resulting degree of freedom corresponds to the Euler roll angle, effectively completing the gimbal-like structure. The high ball density of the roll bearing, combined with a very small ball diameter (approximately 1 mm) result in considerable friction, which unfortunately cannot be overcome by the aerodynamic roll moment. For this reason alone, the roll axis will not be considered in this work. The high spread of the outer gimbal results in low torsional rigidity along the vertical axis (the horizontal and out-of-plane motions are restricted thanks to the top and bottom bearings). Combined with the carbon fiber tubing, this results in a torsional vibrating mode around 12.27 Hz affecting the yaw axis.

Other mechanical solutions, such as an inner gimbal or a tensioned wire mesh, have also been considered. The inner gimbal solution severely restricts the hardware integration inside the projectile and requires careful mass balancing. However, in this solution the Euler angles may be measured directly using angular sensors, removing the need of an IMU. Most of the support electronics can be moved outside the projectile, as the wiring can follow or go through the sting support. Finally, in this solution the aerodynamic forces and moments can be measured directly using an aerodynamic balance. This solution was not retained though due to the implementation complexity, but may be considered in future work. The tensioned wire mesh solution was not retained, as it does not solve the roll friction issue. It also results in parasitic angular moments at high incidence angles and may also present vibrating modes.

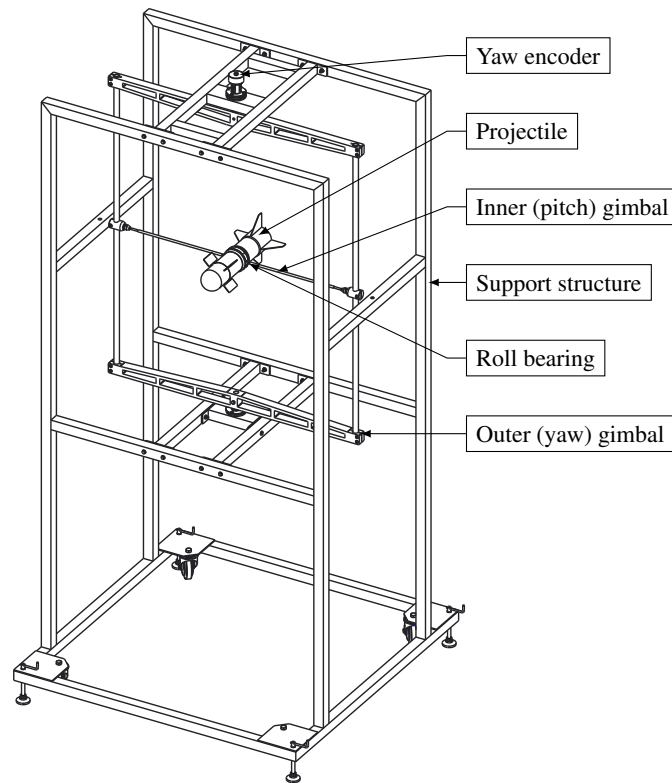


FIGURE 1.19: Three-degrees of freedom support structure components.

1.4 Software Environment

This section describes the ACHILES software environment, which consists in the embedded operating system and associated real-time framework on the one hand, and on the development environment based on MATLAB/Simulink on the other hand.

1.4.1 Embedded Software

The main purpose of the ACHILES embedded software is to provide an execution environment for the developed algorithms on the target hardware. To this end, this software provides several support functionalities, namely interfaces with the embedded actuators, embedded sensors and external sensor interfaces and a communication link with the development environment for monitoring and interacting with the simulation. The embedded software also provides a remote console access for housekeeping tasks, such as zeroing the actuators, and for retrieval of the collected data.

In the context of HIL simulation and rapid controller prototyping, the embedded software is subject to a real-time constraint. This constraint arises when a system has to guarantee a response within specified time constraints. Several classes of real-time constraints can be defined, depending on the application and the consequences of a missed deadline [Bernat et al., 2001; Brown and Martin, 2010]:

Hard real-time : an event must be reacted to within a strict deadline. A failed deadline results in total system failure, possibly endangering human lives and/or property. For example, automotive powertrain and chassis control ECUs, medical devices such as pacemakers, industrial process

controllers are hard real-time systems.

Firm real-time : these systems tolerate deadline misses up to a certain amount as long as they are sufficiently spaced, but the usefulness of the result is null after the deadline. An example of such a system is an automated production line, where missed deadlines result in manufacturing defects.

Soft real-time : these systems allow frequently missed deadlines. The result has still value after the deadline, but the quality of service is degraded. For instance, a video streaming system can tolerate deadline misses, resulting in lower quality or playback stall.

Based upon this classification, a projectile guidance, navigation and control system is a hard real-time system, as missed deadlines could result in collateral damage. In general, hard real-time systems require very specific real-time operating systems, and are subject to strict code quality and certification requirements, like in the automotive or aerospace industries. The ACHILES setup on the other hand is a firm real-time system, as a missed deadline could only result in a failed experimental run. In order to meet these real-time requirements, the embedded software is built upon a Linux-based real-time operating system, and the developed algorithms are implemented using a multi-threaded, real-time software framework. These components are described in the following subsections.

1.4.1.1 Real-Time Operating System

Depending on the application, real-time constraint, processing power, there are several real-time operating system (RTOS) options available for the designer, both from commercial sources and from the free/open-source software community. Commonly used RTOSes in the industry include Wind River VxWorks, QNX Neutrino, FreeRTOS and Real-time Linux distributions. Most of these solutions are designed for highly integrated, hard real-time embedded development and lack the flexibility of a fully-fledged desktop operating system. Moreover, proprietary solutions usually have high licensing costs.

Contrary to traditional embedded RTOSes, which aim at minimizing their computational and memory footprint at the expense of flexibility, real-time Linux-based operating systems retain full functionality with the addition of soft or hard real-time capabilities. Several methods are available for extending Linux with real-time capabilities [Vun et al., 2008]. *PREEMPT* patches such as *CONFIG_PREEMPT_RT* try to reduce the blocking durations of high-priority operations, at the expense of overall throughput¹¹. Co-kernel approaches, such as Xenomai [Gerum, 2004], add a real-time subsystem besides the Linux kernel and expose their functionality to Linux applications. They require the application to be written using a specific application programming interface (API) but provide better low-level hardware control. This latter approach was retained in the development of ACHILES.

The overall architecture of the Xenomai solution is exposed in Figure 1.20. In a *vanilla* Linux distribution, the kernel has direct access to the processor and peripherals, and the user-space software interact with the hardware through the Linux kernel. In the Xenomai co-kernel approach, the hardware is shared between the two kernels by the Adeos/I-pipe nano-kernel [Yaghmour, 2001]. This software shim¹² handles hardware interrupt requests (IRS), such as the system timer IRQ and serial busses IRQs

¹¹Number of transactions per unit of time.

¹²Lightweight software layer that transparently intercepts and rewrites, handles or redirects API calls.

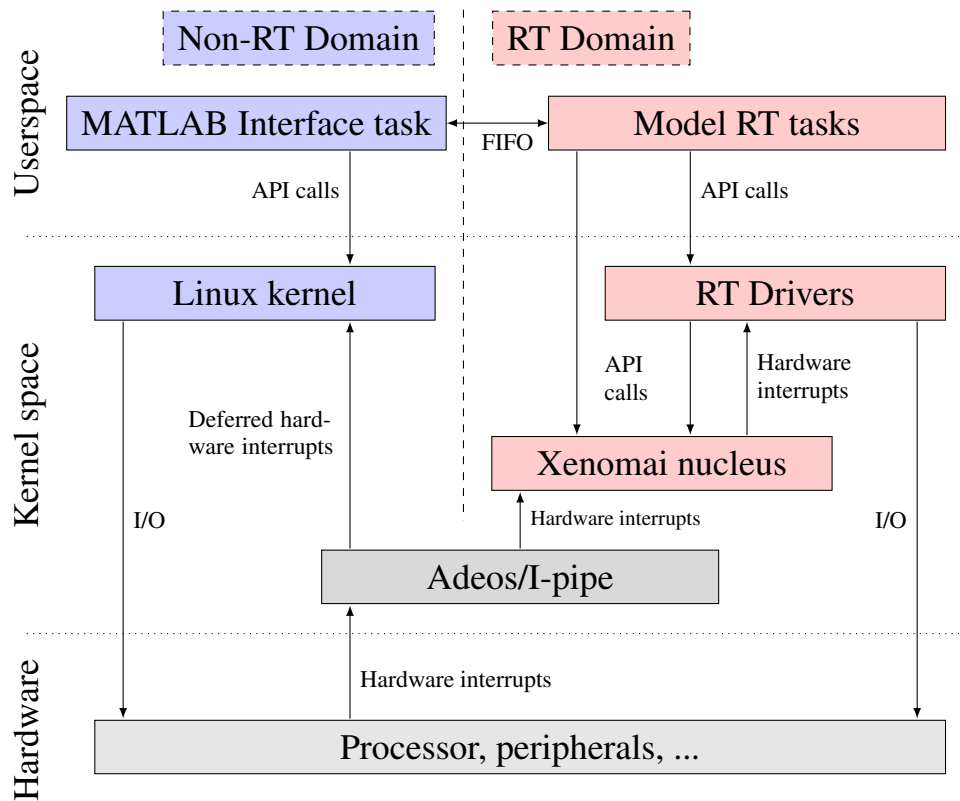


FIGURE 1.20: Embedded software structure.

and passes them to the upper software layer in order of priority. The Xenomai kernel has the highest priority and therefore has the opportunity of responding first.

As the two kernels operate on different models, device drivers for the peripherals used in the real-time application (namely, the serial, I²C and SPI bus peripherals) have been written for use with the Xenomai kernel. These drivers coexist in a mutually exclusive way with the existing Linux drivers, enabling access to these peripherals to non-realtime applications. When used in a real-time application, the existing Linux driver is entirely bypassed and the Xenomai driver interacts with the hardware instead.

1.4.1.2 Application Framework

The data-collection experiments and control algorithms are designed in MATLAB/Simulink. Using code-generation facilities, the Simulink diagram (hereafter called *model*) are converted to C source code, which is implemented on the embedded computer with the help of a purpose-built application framework. This software environment provides an interface with the projectile hardware, a communication channel with the development environment, and executes the model code at the specified sampling frequency.

This framework is a multi-threaded application, composed of both real-time and non-real-time tasks. The starting thread, or *main* task, runs in the Linux domain and manages the simulation startup, execution and shutdown. More specifically, this thread configures the IMU to send the desired data values, activates the actuators control loops and starts the real-time tasks. During the simulation, it provides a remote management channel with the development environment over a wireless network link. When the simulation is terminated (either by user action or after a certain duration), this thread stops the real-time

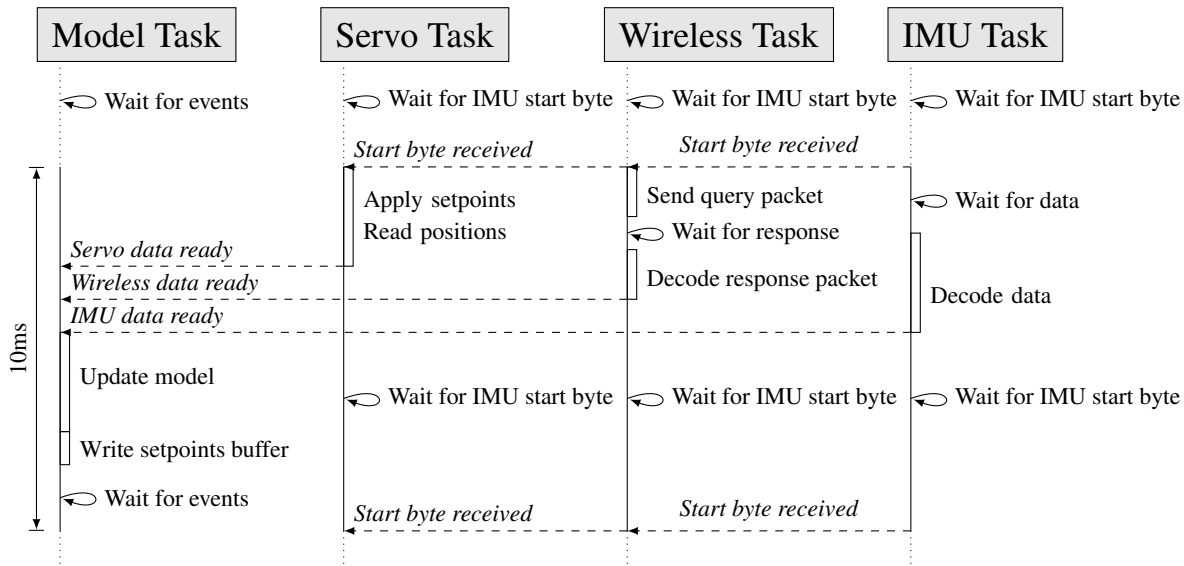


FIGURE 1.21: Real-time tasks execution sequence.

tasks, writes the collected data to the embedded SD card and cleans up before exiting.

The real-time part of the application framework consists in the *model* task, as well as hardware interface tasks for the IMU, actuators and external sensors. These timing-sensitive threads are executed in the Xenomai domain in order to minimize latency and meet the real-time constraints. The main timing source in the ACHILES application framework is the IMU update rate, which is set by default to 100 Hz. This solution is preferred over a polling approach with a system timer, as it guarantees the lowest latency between the sampling of inertial sensors, and the availability of inertial estimates.

The real-time task execution sequence is detailed in Figure 1.21. In order to synchronize sampling instants across all actuators and sensors, the interface tasks are in a waiting state until the first byte of an IMU message is received. The following actions are then performed simultaneously:

- The **IMU Task** waits for the complete IMU message to be received. The inertial data are then stored in memory, and the *IMU data ready* flag is set.
- The **Servo Task** sends a general call command to all actuators over the I²C bus. This command stores the actual fin position in an internal buffer, and defines the new position setpoint to the value defined at the previous cycle. Then, all fin positions are read from the actuators sequentially and stored to memory, then the *servo data ready* flag is set.
- The **Wireless Task** sends a request packet to the external sensors acquisition board, and waits for the response packet. If the packet is received within a specified timeframe, the response is processed and stored to memory. Otherwise, the packet is considered lost and the wireless data for this cycle are considered invalid. The task then sets the *wireless data ready* flag.

These tasks interact with the projectile hardware through custom-developed real-time drivers running on top of the Xenomai nucleus. The collected sensor data are stored in memory and made accessible to the *model* task through access functions.

The model task is in a locked state until all interface task flags are set. When these tasks are ready, one simulation step is performed by calling the generated source code, which may retrieve the stored sensor data using the access functions provided by the interface tasks. The model outputs are then written to the servo setpoint buffers for application on the next cycle, thus ensuring a fixed 1-cycle delay between the controller inputs and outputs.

1.4.2 Development Environment

A key element in the design of a HIL simulation or a rapid control prototyping platform is the development environment. Ideally, this environment should allow the complete algorithm development workflow to be followed using a single paradigm, to avoid re-implementation between design phases. The ACHILES development is based on MATLAB/Simulink, which is a widely used tool in the control community. This environment is also well-suited for the development of HIL and rapid control prototyping setups [Hercog and Jezernik, 2005; Bucher and Balemi, 2006], thanks to flexible code-generation tools suitable for a wide selection of platforms [The Mathworks, 2012].

In this regard, identification and control algorithms are developed in the MATLAB environment, while the controllers are implemented as Simulink diagrams, which can be exploited for numerical simulation as well as for testing on the projectile prototype. The deployment of Simulink diagrams is performed with the help of the MATLAB Coder and Simulink Coder toolboxes, for which an ACHILES-specific code generation target has been developed. This target specifies the embedded computer characteristics, defines the compilation process and also provides a Simulink block library for interacting with the projectile's hardware through the embedded application framework.

1.4.2.1 Code Generation and Compilation Process

Simulink diagrams for implementation on the projectile hardware differ from numerical simulation diagrams on several points. The diagram is no longer executed on the development computer, but has to run in *External* mode, where the *Run/Stop* buttons on the main toolbar are completed with a *Connect* button and a *Build* button. As the main timebase is set by the IMU, only fixed-step solvers (continuous or discrete) may be used, with the major timestep set to the IMU update period (10 ms). Moreover, the code generation toolbox imposes some restrictions on the blocks that may be used in the diagram (see the Simulink Coder User's Guide [The Mathworks, 2012], pages 128 to 147), and more specifically, the use of different sample times in a same diagram must be defined explicitly.

The code generation and compilation process is detailed in Figure 1.22A. The *Generate Code* step consists in converting all blocks appearing in the Simulink diagram and its subsystems to the corresponding C source code. The generated model code is then contained in the function blocks detailed in Figure 1.22B, which presents the model execution flow chart. Functions in shadowed blocks are visible from the global application scope and are called from the *model* task of the application framework during execution. The next step, *Generate Makefile*, generates a compilation script involving the model source code as well as the embedded application framework source files. Then, all source files are cross-compiled using the ARM-gcc compiler, and the produced executable is uploaded to the embedded computer over the Wifi connection.

Finally, the executable is started over a remote login session, and the user may connect the Simulink

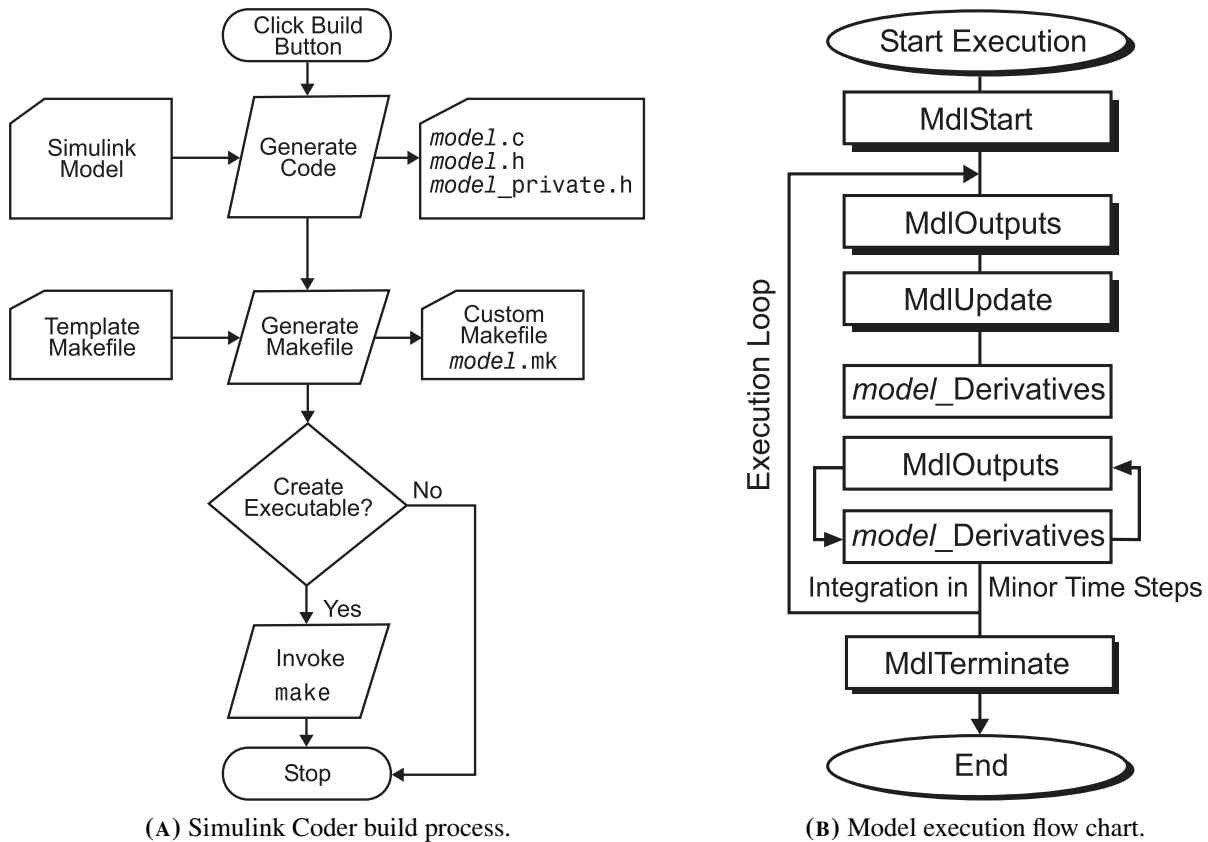


FIGURE 1.22: Simulink Coder build process and model execution flow chart [The Mathworks, 2012].

diagram to the target using the External Mode controls of the main toolbar. The execution of the simulation can then be supervised using standard blocks (e.g. the *Scope* block), and parameters can be tuned during execution. At the end of the simulation, the monitored variables are available in the MATLAB workspace for post-processing, and are also saved in a MAT-file on the embedded computer.

1.4.2.2 Interface with the Application Framework

For HIL and rapid control prototyping simulations, the controller is no longer interfaced with a simulated plant but directly acts on the projectile hardware. To this end, a custom Simulink block library has been developed to provide an easy-to-use interface with the projectile actuators and sensors. These blocks are implemented using *fully-inlined S-Functions*, which consist in a C/C++ MEX S-function and a target language compiler (TLC) file. The MEX S-function is not used during code generation but defines the block behavior, notably the number of inputs and outputs with their data types, the number of discrete or continuous states. It also indicates if the block has a direct feedthrough, as this influences on the execution order of the blocks. The corresponding TLC file is processed during code generation, and contains the actual block algorithm source code. In the ACHILES case, this corresponds to the access functions mentioned in Section 1.4.1.2.

1.5 Concluding Remarks

In this chapter, an innovative guided projectile Hardware-In-the-Loop setup for model identification and controller validation has been presented. This setup aims at providing an intermediate step in a guided projectile design workflow, between numerical simulation and free-flight testing campaigns. The developed experimental setup is composed of a fully autonomous guided projectile prototype based on a missile-like structure with four independently-actuated, front-located control canards and four aft-located stabilization fins. This projectile is suspended in the test section of a wind tunnel using a gimbaled support structure, which enables the emulation of the projectile's free-flight angular behavior in a controlled environment.

One of the main design goals was to provide a complete rapid controller prototyping environment in order to streamline the controller development workflow and reduce time spent on implementation details. This objective is fulfilled by a customized software environment based on real-time Linux on the projectile prototype, and MATLAB/Simulink on the development computer.

The experimental setup satisfies all of its design objectives, and has been used with success throughout this thesis. There are some minor limitations due to the support structure, which affects the dynamic response due to the gimbals' inertia, and the limited usability of the roll axis due to excessive bearing frictions. In future developments, the projectile will be held in the airflow by the means of an internal gimbal. Although this solution is mechanically more complex and restricts the angular range, it has less impact on the projectile behavior thanks to lower inertia. Moreover, the actual forces and moments acting on the projectile may be measured using a 6-DoF internal balance, allowing direct control of the load factor.

Chapter 2

Modeling and Identification

Contents

2.1	Introduction	40
2.2	Flight Mechanics of a Free-Flying Projectile	41
2.2.1	Frames and Coordinate Systems	41
2.2.2	Free-Flight 6-Degrees of Freedom Model	43
2.2.3	Aerodynamic Forces and Moments	47
2.3	Flight Mechanics of a Translation-Denied Projectile	49
2.3.1	Dynamic Model of the Translation-Denied Projectile	50
2.3.2	Airframe incidence angles	52
2.3.3	Roll-Locked Case	52
2.3.4	Roll/Yaw-Locked Case	54
2.4	Trimming and Linearization	55
2.4.1	Trimming	55
2.4.2	Linearization	58
2.5	System Identification	60
2.5.1	Identification Procedure	60
2.5.2	Model Postulation	61
2.5.3	<i>A priori</i> Identifiability	62
2.5.4	<i>A posteriori</i> Identifiability	63
2.5.5	Experiment Design	65
2.5.6	Experimental Data Collection	67
2.5.7	Parameter Estimation and Uncertainty Analysis	68
2.6	Parameter Estimation Results	69
2.6.1	Overall Trends	70
2.6.2	Model Fit Evaluation	72
2.6.3	Uncertainty Evaluation	73
2.6.4	Constant-Airspeed Trends	73
2.6.5	Comparison with PRODAS-derived Aerodynamic Coefficients	77
2.6.6	Pitch-only case	78
2.7	Conclusion	79

2.1 Introduction

Modeling is the first and foremost step of model-based control design, aiming at describing the plant's behavior via a mathematical model. This plant model consists of a *model structure*, which describes the relationships between the model variables, and of *model parameters*. The model structure may be derived from first principles, such as physical or chemical equations, or inferred from observations of the plant behavior. The model parameters are either directly measured physical properties (e.g. mass, resistance) or estimated via system identification techniques, in which case the estimated parameters may have no direct physical interpretation.

This chapter focuses on the derivation of a model structure representing the flight behavior of the ACHILES projectile, and then on the estimation of the model parameters around fixed operating points. This task is carried out using a flight mechanics framework widely used in the aircraft and missile fields [Zipfel, 2007; Hull, 2007; Siouris, 2004]. In this context, the model can be divided into two parts. The *first part* describes both the dynamic and kinematic behavior of the airframe using classical rigid-body mechanics. The *second part* aims at modeling the aerodynamic forces and moments acting on the flying vehicle. However, the aerodynamic behavior of a flying body (as opposed to the rigid-body behavior) relies on complex partial differential equations which cannot be solved analytically in the general case [Nielsen, 1988]. The aerodynamic forces and moments are thus approximated using polynomial descriptions based on aerodynamic coefficients, whose determination is a complete field of study in its own. The combination of both parts results in a highly nonlinear model with a large number of parameters.

As mature robust control techniques are more easily applicable to LTI or LPV systems, a further linearization or transformation step is required. In order to reduce model complexity as well as the number of parameters to be estimated, the non-linear model is linearized around a number of operating points. The parameters of these linear models are estimated for each operating point, and the set of models then forms a family of linear models (which through a slight abuse in notations can be called a *quasi-LPV* model), suitable for autopilot design [Leith and Leithead, 2000].

The estimation of the model parameters is an inverse problem, which can be challenging or impossible to solve if ill-posed or ill-conditioned. To this end, a systematic identification procedure is applied. This procedure starts with evaluating whether the parameter estimation is well-posed and well-conditioned through respective *a priori* and *a posteriori* identifiability analyses. In the former, the existence and unicity of a solution is assessed in an ideal noise-free case, in which the input and output signals can be chosen freely. In the latter, sensitivity functions are computed from the actually available input and output signals in order to determine which parameters are identifiable in practice. The following step consists in the design of an optimal input signal, which satisfies an excitation criterion and aims at minimizing the estimates variance. This signal is applied to the system during data-collection experiments. These measurements are used in the final steps of the procedure to estimate the model parameters and the uncertainties on these estimates.

This chapter is organized as follows. The general case of a free-flying projectile is considered first in Section 2.2, which covers the involved frames and coordinate systems, the six-degrees of freedom mechanical model and the modeling of the aerodynamic forces and moments acting on the projectile. In the ACHILES case, the projectile is no longer in free flight and translational degrees of freedom are restricted by the support structure. These considerations are taken into account in Section 2.3, where

an ACHILES-specific model is derived and various axis-locked configurations are detailed. Section 2.4 then deals with the trimming and linearization of the ACHILES-specific nonlinear models. Finally, the identification procedure necessary for estimating the model parameters is presented in Section 2.5. The parameter estimation results are finally presented and discussed in Section 2.6

2.2 Flight Mechanics of a Free-Flying Projectile

This section focuses on the derivation of a nonlinear model describing the behavior of a projectile in free-flight. The frames and coordinate systems involved in the mechanical modeling are presented in Section 2.2.1. The complete 6-DoF mechanical model of a free-flying projectile is then derived. Finally, Section 2.2.3 deals with the aerodynamic forces and moments modeling.

2.2.1 Frames and Coordinate Systems

This section aims at describing the frames and coordinate systems which are involved in the derivation of a flying craft's equations of motion in the flight mechanics framework. Although they are commonly considered as synonyms and used interchangeably, the terms *frame* and *coordinate system* refer to different notions: a frame models a physical reference, such as the Earth or the vehicle, whereas a coordinate system associates a vector with scalar coordinates in the Euclidean space [Zipfel, 2007]. A frame is a set of points with invariant distances containing at least three non-collinear points. Hence, for any given frame, an infinite number of coordinate systems may be defined. An orthonormal triad, consisting of a base point and three mutually orthogonal unit vectors, is sufficient to define a frame. In this case, the *preferred* coordinate system is the one aligned with the triad's base vectors.

Earth Frame Modeling an aerospace vehicle consists in expressing its position and motion relatively to an inertial reference frame (in the Newtonian sense). In the flight mechanics framework, three reference frames are most used: the Sun-centered (heliocentric) frame, the Earth-centered (Geocentric-inertial) frame and the Earth frame. Depending on the application, either of these frames can be considered as inertial. The Sun-centered frame is employed in space travel applications, while the Geocentric-centered frame is used for most Earth-orbiting applications. The Geocentric-centered frame and the Earth frame differ in that the Geocentric-centered frame keeps a constant orientation along the Earth orbit, while the Earth frame rotates with the Earth itself. In the case of Earth-bound flights, the rotation of the Earth is neglected and the Earth frame is considered as inertial. This is the case for most guided projectile applications, and especially in the ACHILES case. The Earth frame is defined as follows. The base point E is located at the Earth's center, and the first base vector e_1 points towards the intersection of the Greenwich meridian and the Equator plane. The third base vector e_3 is directed along the Earth rotation axis, pointing towards the North Pole, and the second base vector e_2 completes the triad.

Body Frame In the general case, the studied flying vehicle is considered to be a rigid body. The vehicle can then be represented by the *body* frame, which is used for modeling its position and orientation *w.r.t.* a (potentially inertial) reference frame. The base point B of the body frame is located at the vehicle's center of mass, and the base vectors are directed along the main axes of the moment-of-inertia tensor. The first base vector b_1 is directed along the main axis of the vehicle, pointing out of the nose. The second base vector b_2 exits the vehicle through the right wing and the third base vector b_3 points downwards.

Although each of the above-defined frames embeds a preferred coordinate system, additional coordinate systems are required in the development of an aerospace vehicle model, depending on the application. As for reference frames, several coordinate systems are defined in the flight mechanics framework, however only the coordinate systems relevant to ACHILES and the associated transformations will be presented in the sequel.

Local Coordinate System Let us consider an arbitrary point on the Earth surface, e.g. the projectile launch point. The local coordinate system, denoted \mathcal{J}^L , is then defined such that its 1^L and 2^L axes define a plane tangent to the Earth at the considered point, with the 1^L axis pointing North and the 2^L axis pointing East. The 3^L axis points towards the center of the Earth, completing a North-East-Down (NED) coordinate system.

Body Coordinate System The body coordinate system, denoted \mathcal{J}^B , corresponds to the preferred coordinate system of the body frame, defined previously by the triad $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$. The orientation of the body coordinate system relative to the local coordinate system is described using the aircraft convention for Euler angles, which is highlighted in Figure 2.1. The Euler angles transformation consists in three consecutive rotations by the roll, pitch and yaw angles, respectively ϕ , θ and ψ . In the aircraft or 'ZYX' convention, the first rotation, of angle ψ , is about the vertical 3^L axis, giving the first intermediate X coordinate system. The second rotation, of angle θ , is about the 2^X axis and yields the second intermediate coordinate system, Y. Finally, the last rotation is of angle ϕ about the 1^Y axis, closing the chain between the local and the body coordinate systems.

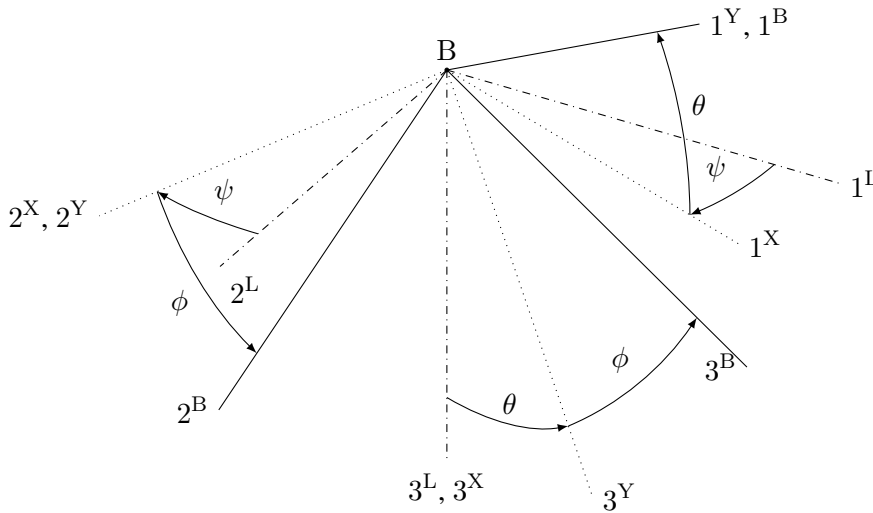


FIGURE 2.1: Body coordinates *w.r.t.* local coordinates using the aircraft Euler angles convention.

These transformations are expressed in matrix form as:

$$[\mathbf{T}]^{XL} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad [\mathbf{T}]^{YX} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}, \quad [\mathbf{T}]^{BY} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}$$

The chain multiplication of these matrices then yields the complete transformation:

$$[\mathbf{T}]^{BL} = [\mathbf{T}]^{BY} [\mathbf{T}]^{YX} [\mathbf{T}]^{XL} \quad (2.1)$$

$$[\mathbf{T}]^{\text{BL}} = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \cos \theta \sin \phi \\ \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \cos \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.2)$$

Wind Coordinate System The Cartesian¹³ wind coordinate system $]^{\text{W}}$ is defined upon the vehicle relative wind vector v_{B}^{A} , describing the velocity of the vehicle's center of mass B *w.r.t.* the air frame A, illustrated in Figure 2.2. The 1^{W} axis is collinear and pointing in the same direction as v_{B}^{A} , while the 2^{W} and 3^{W} are defined using an intermediate stability coordinate system $]^{\text{S}}$. The 1^{S} axis is parallel and in the direction of the projection of the velocity vector in the plane $(1^{\text{B}}, 3^{\text{B}})$, and the 2^{S} axis coincides with 2^{B} . By convention, the angle of attack (AoA) α is taken positive from 1^{S} to 1^{B} and the angle of sideslip (AoS) β is positive from 1^{S} to 1^{W} . This leads to the transformation matrix [Zipfel, 2007]:

$$[\mathbf{T}]^{\text{WB}} = \begin{bmatrix} \cos \alpha \cos \beta & \sin \beta & \sin \alpha \cos \beta \\ -\cos \alpha \sin \beta & \cos \beta & -\sin \alpha \sin \beta \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (2.3)$$

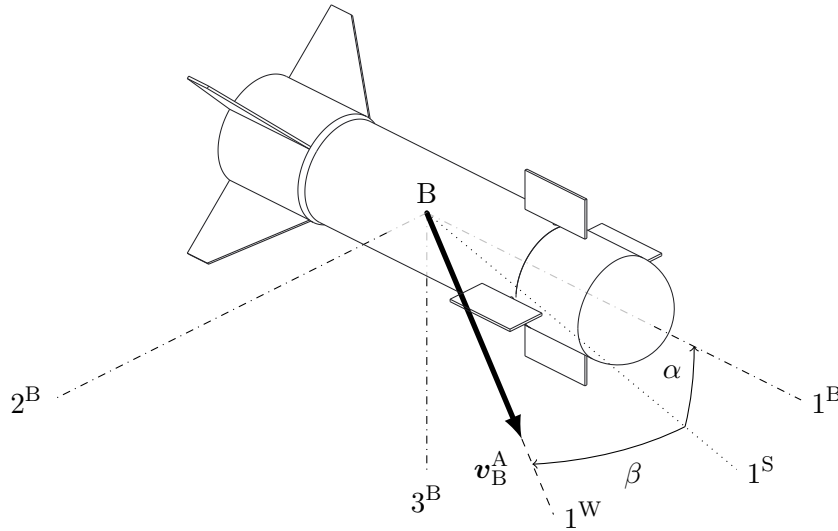


FIGURE 2.2: Wind coordinates *w.r.t.* body coordinates, showing the angle of attack α and the angle of sideslip β .

2.2.2 Free-Flight 6-Degrees of Freedom Model

Before deriving the actual equations of motion involved in the ACHILES experimental setup, it is first desirable to consider the more general case of a free-flying projectile. In the context of gun-fired guided and artillery projectiles, the following assumptions are made:

1. The projectile is a rigid body with invariant mass m and mass distribution,
2. The Earth frame is considered as an inertial reference frame,

¹³For missiles with rotational symmetry, the aeroballistic wind coordinate system is also commonly used. The relative wind direction is then expressed by the polar aeroballistic angles: the total angle of attack α' , which is the angle formed by the wind vector and the 1^{B} axis, and the aerodynamic roll angle ϕ' , which is the angle between the projection of the wind vector in the $(2^{\text{B}}, 3^{\text{B}})$ plane and the 3^{B} axis.

3. The Earth surface can be approximated by a plane.

After the initial launch acceleration, the only remaining efforts on the projectile are its own weight $m\mathbf{g}$ and aerodynamic forces \mathbf{f}_a and moments \mathbf{m}_B . The forces are applied at the projectile's center of mass (*c.m.*), and moments are also expressed at this point. As the projectile is an isolated rigid body, Newton's and Euler's second laws apply directly [Zipfel, 2007]:

$$m\mathcal{D}^E\mathbf{v}_B^E = m\mathbf{g} + \mathbf{f}_a \quad (2.4)$$

$$\mathcal{D}^E\mathbf{l}_B^{BE} = \mathbf{m}_B \quad (2.5)$$

In the above equations, \mathbf{v}_B^E is the projectile *c.m.* velocity *w.r.t.* the Earth frame and $\mathbf{l}_B^{BE} = \mathbf{I}_B^B\boldsymbol{\omega}^{BE}$ is the angular momentum of the projectile body *w.r.t.* the Earth frame referred to its *c.m.*, where \mathbf{I}_B^B is the moment of inertia (MOI) tensor of the projectile body referred to its *c.m.* and $\boldsymbol{\omega}^{BE}$ is the angular velocity vector of the projectile body *w.r.t.* the Earth frame. The \mathcal{D}^E notation designates the rotational time derivative with respect to the inertial frame E. Shifting the rotational time derivative of Equations (2.4)–(2.5) through the Euler transformation so as to derive these equations in the body frame then gives:

$$m\mathcal{D}^B\mathbf{v}_B^E + m\boldsymbol{\Omega}^{BE}\mathbf{v}_B^E = m\mathbf{g} + \mathbf{f}_a \quad (2.6)$$

$$\mathcal{D}^B\mathbf{l}_B^{BE} + \boldsymbol{\Omega}^{BE}\mathbf{l}_B^{BE} = \mathbf{m}_B \quad (2.7)$$

where $\boldsymbol{\Omega}^{BE}$ is the skew-symmetric tensor of $\boldsymbol{\omega}^{BE}$. These equations are then expressed in the body coordinate system $]^B$, and in this case the rotational time derivative \mathcal{D}^B becomes an ordinary time derivative:

$$m \left[\frac{d\mathbf{v}_B^E}{dt} \right]^B + m [\boldsymbol{\Omega}^{BE}]^B [\mathbf{v}_B^E]^B = m [\mathbf{T}]^{BL} [\mathbf{g}]^L + [\mathbf{f}_a]^B \quad (2.8)$$

$$[\mathbf{I}_B^B]^B \left[\frac{d\boldsymbol{\omega}^{BE}}{dt} \right]^B + [\boldsymbol{\Omega}^{BE}]^B [\mathbf{I}_B^B]^B [\boldsymbol{\omega}^{BE}]^B = [\mathbf{m}_B]^B \quad (2.9)$$

The gravity vector is available in the local coordinate system as $[\mathbf{g}]^L = [0 \ 0 \ g]^T$, where T denotes matrix transposition. Its expression in the body coordinate system is obtained by premultiplication with the transformation matrix $[\mathbf{T}]^{BL}$. The linear and angular velocities \mathbf{v}_B^E and $\boldsymbol{\omega}^{BE}$ are expressed in the body coordinate system as (see Figure 2.3):

$$[\mathbf{v}_B^E]^B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}^B, \quad [\boldsymbol{\omega}^{BE}]^B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}^B \quad (2.10)$$

The skew-symmetric tensor $\boldsymbol{\Omega}^{BE}$ related to $\boldsymbol{\omega}^{BE}$ is defined in $]^B$ as:

$$[\boldsymbol{\Omega}^{BE}]^B = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}^B \quad (2.11)$$

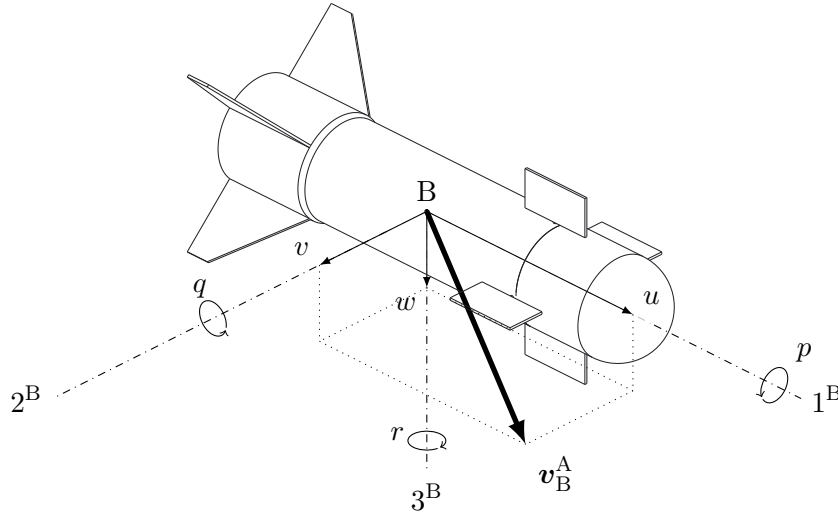


FIGURE 2.3: Projectile angular velocity ω^{BE} and linear velocity v_{B}^{A} expressed in body coordinates.

The projectile's inertia tensor can be considered diagonal and is expressed in the body coordinate system as:

$$[\mathbf{I}_{\text{B}}^{\text{B}}]^{\text{B}} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}^{\text{B}} \quad (2.12)$$

Combining Equations (2.8) to (2.12), the scalar differential equations describing both the translational and rotational dynamic behavior of a flying vehicle can be expressed as:

$$\dot{u} = rv - qw + 1/mf_{a_1} + t_{13}g \quad (2.13)$$

$$\dot{v} = pw - ru + 1/mf_{a_2} + t_{23}g \quad (2.14)$$

$$\dot{w} = qu - pv + 1/mf_{a_3} + t_{33}g \quad (2.15)$$

$$\dot{p} = I_1^{-1} [(I_2 - I_3)qr + m_{\text{B}1}] \quad (2.16)$$

$$\dot{q} = I_2^{-1} [(I_3 - I_1)pr + m_{\text{B}2}] \quad (2.17)$$

$$\dot{r} = I_3^{-1} [(I_1 - I_2)pq + m_{\text{B}3}] \quad (2.18)$$

In the above equations, t_{ij} denotes the i th row, j th column component of the Euler transformation matrix $[\mathbf{T}]^{\text{BL}}$. Aerodynamic forces \mathbf{f}_a and moments \mathbf{m}_{B} are expressed in the body coordinate system, and the i subscript in f_{a_i} and m_{B_i} denotes the respective vector component.

In the development of a projectile model for controller synthesis, the air is assumed to be static *w.r.t.* the air, hence the geographic velocity vector v_{B}^{E} is equal to the relative wind vector v_{B}^{A} . From the definition of the wind coordinate system, the expression of the relative wind is $[v_{\text{B}}^{\text{A}}]^{\text{W}} = [V_w \ 0 \ 0]^{\text{T}}$ where V_w also corresponds to the geographic velocity $V = \sqrt{u^2 + v^2 + w^2}$. Applying the coordinate

system transformation $[\mathbf{v}_B^A]^W = [\mathbf{T}]^{WB} [\mathbf{v}_B^A]^B$ then yields:

$$\begin{bmatrix} V \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \alpha \cos \beta & \sin \beta & \sin \alpha \cos \beta \\ -\cos \alpha \sin \beta & \cos \beta & -\sin \alpha \sin \beta \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.19)$$

The incidence angles α and β are then given by:

$$\alpha = \arctan \left(\frac{w}{u} \right) \quad (2.20)$$

$$\beta = \arcsin \left(\frac{v}{V} \right) \quad (2.21)$$

Conversely, the linear velocities u , v and w can be expressed as:

$$u = V \cos \alpha \cos \beta \quad (2.22)$$

$$v = V \sin \beta \quad (2.23)$$

$$w = V \sin \alpha \cos \beta \quad (2.24)$$

The set of differential equations (2.13)–(2.15) governing the translational dynamics may then be rewritten in terms of V , α and β by taking the derivative of the definition of V and Equations (2.20) and (2.21):

$$\dot{V} = \frac{u\dot{u} + v\dot{v} + w\dot{w}}{V} \quad (2.25)$$

$$\dot{\alpha} = \frac{u\dot{w} + w\dot{u}}{u^2 + w^2} \quad (2.26)$$

$$\dot{\beta} = \frac{-uv\dot{u} + (u^2 + w^2)\dot{v} - vw\dot{w}}{V^2\sqrt{u^2 + w^2}} \quad (2.27)$$

Combining the above with the values of u , v , w from Equations (2.22)–(2.24) and their derivatives \dot{u} , \dot{v} , \dot{w} from Equations (2.13)–(2.15), the final translational dynamics equations are obtained:

$$\begin{aligned} \dot{V} &= \frac{1}{m} (f_{a1} \cos \alpha \cos \beta + f_{a2} \sin \beta + f_{a3} \sin \alpha \cos \beta) \\ &\quad + g (t_{13} \cos \alpha \cos \beta + t_{23} \sin \beta + t_{33} \sin \alpha \cos \beta) \end{aligned} \quad (2.28)$$

$$\begin{aligned} \dot{\alpha} &= q - \tan \beta (p \cos \alpha + r \sin \beta) \\ &\quad + \frac{1}{mV \cos \beta} (-f_{a1} \sin \alpha + f_{a3} \cos \alpha) + \frac{g}{V \cos \beta} (-t_{13} \sin \alpha + t_{33} \cos \alpha) \end{aligned} \quad (2.29)$$

$$\begin{aligned} \dot{\beta} &= p \sin \alpha - r \cos \alpha + \frac{1}{mV} (-f_{a1} \cos \alpha \sin \beta + f_{a2} \cos \beta - f_{a3} \sin \alpha \cos \beta) \\ &\quad + \frac{g}{V} (-t_{13} \cos \alpha \sin \beta + t_{23} \cos \beta - t_{33} \sin \alpha \sin \beta) \end{aligned} \quad (2.30)$$

In the above, Equation (2.28) describes the projectile's axial dynamics while Equations (2.29) and (2.30) respectively describe the normal and the lateral projectile dynamics¹⁴.

¹⁴For a projectile exhibiting rotational symmetry, both dynamics are equivalent and are designated as the lateral dynamic.

2.2.3 Aerodynamic Forces and Moments

The equations of motion reviewed in the previous section involve the external efforts applied on the projectile. Aside from gravitational forces, projectiles are subject to aerodynamic forces and moments induced by their motion through the air. These forces are influenced by the flow characteristics (Mach number, Reynolds number), the incidence angles and their derivatives, the body rates, the control deflections, the projectile shape, surface and *c.m.* location [Zipfel, 2007]. However, the aerodynamic behavior of a projectile cannot be obtained analytically, and thus the aerodynamic forces and moments are usually modeled through aerodynamic coefficients given in lookup tables or from polynomial decompositions. The force and moment vectors \mathbf{f}_a and \mathbf{m}_B are expressed in body coordinates as:

$$[\mathbf{f}_a]^B = \begin{bmatrix} f_{a1} \\ f_{a2} \\ f_{a3} \end{bmatrix} = \bar{q}S \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} \quad [\mathbf{m}_B]^B = \begin{bmatrix} m_{B1} \\ m_{B2} \\ m_{B3} \end{bmatrix} = \bar{q}Sd \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} \quad (2.31)$$

where \bar{q} is the dynamic pressure, S is the projectile reference area (cross section) and d is the projectile reference length (caliber). The dynamic pressure is a function of the airspeed V and the air density ρ at altitude h , expressed as:

$$\bar{q} = \frac{1}{2}\rho(h)V^2 \quad (2.32)$$

The aerodynamic coefficients C_X , C_Y , C_Z , C_l , C_m and C_n in Equation (2.31) are nondimensional. This property enables the use of reduce-scaled models in wind tunnel studies, where the obtained coefficients also describe the full-scale craft. These coefficients are functions of the flight conditions, typically the Mach number \mathcal{M} , the altitude h , the incidence angles α and β , the body rates p , q , r and the control surface deflections δ :

$$C_j = C_j(\mathcal{M}, \alpha, \beta, p, q, r, \delta), \quad j = X, Y, Z, l, m, n \quad (2.33)$$

The ACHILES projectile has four front-located, independently-actuated control canards of which the deflections δ_i , $i = 1 \dots 4$ are illustrated in Figure 2.4. In order to simplify the expressions of the aerodynamic coefficients, these canards are grouped together horizontally and vertically to form virtual roll, pitch and yaw control surfaces δ_l , δ_m and δ_n using the following mixing logic [Siouris, 2004]:

$$\begin{bmatrix} \delta_l \\ \delta_m \\ \delta_n \end{bmatrix} = \begin{bmatrix} -1/4 & -1/4 & -1/4 & -1/4 \\ 1/2 & 0 & -1/2 & 0 \\ 0 & 1/2 & 0 & -1/2 \end{bmatrix} \cdot \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} \quad (2.34)$$

The sign convention of the virtual deflection angles is chosen such that positive control angles δ_m and δ_n induce positive incidence angles α and β , and a positive virtual roll control δ_l induces a positive rolling moment. This convention is non-unique and alternative hypotheses may be made [Zipfel, 2007; Siouris, 2004].

Using the above virtual control deflections, an approximation of the aerodynamic coefficients in Equation (2.31) can be proposed. As the ultimate goal of the presented modeling procedure is to obtain a linearized state-space model, a linear decomposition in first-order terms is employed, based on the

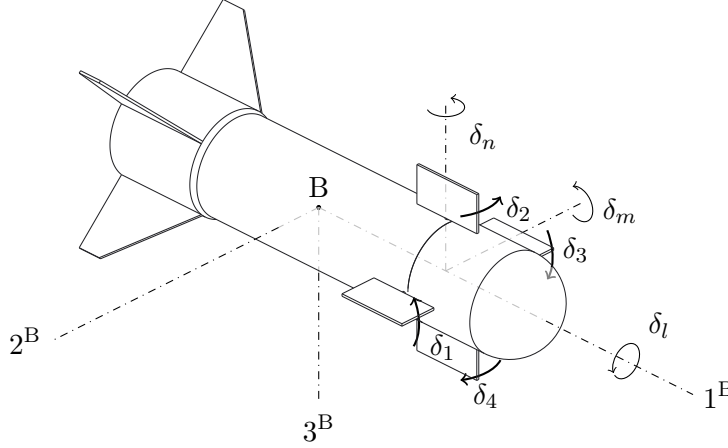


FIGURE 2.4: Configuration of the control canards on the ACHILES projectile and virtual deviation angles.

variables mentioned in Equation (2.33). One suitable form is given in Tayfun [2011] for a missile model, using the incidence angles α and β and is adjusted to keep only the first-order terms. Additional terms have been included to take into account the cross-coupling between the pitch and yaw channels, giving the following decomposition of force and moment coefficients:

$$C_X = C_{X0} + C_{X\alpha'}\alpha' + C_{X\delta_{\text{eff}}}\delta_{\text{eff}} \quad (2.35)$$

$$C_Y = C_{Y0} + C_{Y\beta}\beta + \left(\frac{d}{2V}\right)C_{Yr}r + C_{Y\delta_n}\delta_n \quad (2.36)$$

$$C_Z = C_{Z0} + C_{Z\alpha}\alpha + \left(\frac{d}{2V}\right)C_{Zq}q + C_{Z\delta_m}\delta_m \quad (2.37)$$

$$C_l = C_{l0} + \left(\frac{d}{2V}\right)C_{lp}p + C_{l\delta_m}\delta_l \quad (2.38)$$

$$C_m = C_{m0} + C_{m\alpha}\alpha + C_{m\beta}\beta + \left(\frac{d}{2V}\right)C_{mq}q + C_{m\delta_m}\delta_m \quad (2.39)$$

$$C_n = C_{n0} + C_{n\alpha}\alpha + C_{n\beta}\beta + \left(\frac{d}{2V}\right)C_{nr}r + C_{n\delta_n}\delta_n \quad (2.40)$$

where the *aerodynamic derivatives* $C_{ij} = C_{ij}(\mathcal{M}, \alpha, \beta)$ are functions of the Mach number and the incidence angles. The expression of the axial force coefficient C_X involves the *total* angle of attack $\alpha' = \arccos(\cos \alpha \cos \beta)$ and the effective fin deflection $\delta_{\text{eff}} = \frac{1}{4}\sum_{i=1}^4|\delta_i|$. In the expression of these aerodynamic coefficients, the terms C_{i0} are static coefficients, terms in α and β are the *static stability derivatives*¹⁵, terms preceded by $\left(\frac{d}{2V}\right)$ are the *dynamic stability derivatives*¹⁶ and terms in δ are the *control derivatives*. Finally, $C_{m\beta}$ and $C_{n\alpha}$ describe the cross-axis coupling between the pitch and yaw axes.

¹⁵These derivatives are representative of the static stability of a vehicle, i.e. its tendency to return to an equilibrium when disturbed: negative values indicate a statically stable vehicle while positive values imply static instability.

¹⁶These derivatives characterize the damping of the vehicle's motion. Negative values indicate a dynamically stable vehicle and positive values imply dynamic instability.

2.3 Flight Mechanics of a Translation-Denied Projectile

In the ACHILES framework, the projectile is no longer in free flight. The support structure, presented in Section 1.3.6, inhibits the linear motion of the projectile, while allowing rotation on all axes. In this case, the projectile is no longer an isolated rigid body, but is attached to the Earth frame *via* the inner and outer gimbals of the support structure and the roll bearing, which form an equivalent ball joint. The linear and angular dynamics derived for the free-flight case are then not directly applicable, as the gimbals' inertia and joint reactions have to be considered in the model.

The kinematic diagram of the complete system is given in Figure 2.5 and illustrates the connections between the different components of the support assembly. In the ACHILES case, the air is no longer static *w.r.t.* the Earth, but moves at velocity V in the wind tunnel test section. The definition of the local coordinate system \mathcal{J}^L is also modified such that 1^L is directed along the wind direction in the wind tunnel. In addition to assumptions made for the free-flight case, the inner and outer gimbals are considered as rigid bodies, and all joints are frictionless. The coordinate systems associated to the inner gimbals as well as the joints between the components of the support structure are discussed in the following paragraphs.

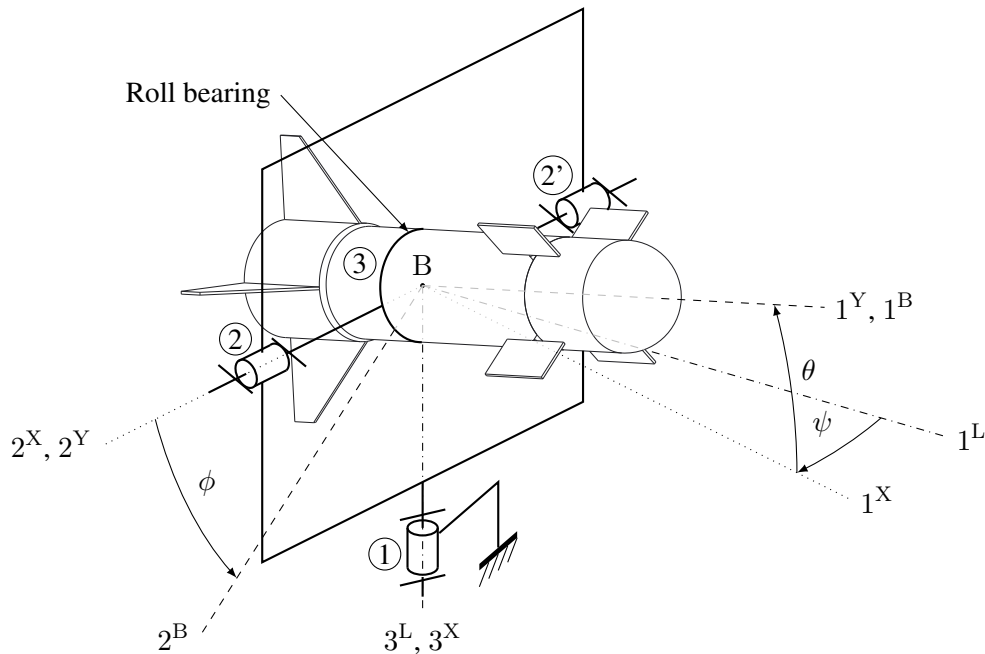


FIGURE 2.5: Kinematics diagram of the ACHILES projectile and support structure.

Outer gimbal The outer gimbal is connected to both the support frame and the inner gimbal by two revolute joints with perpendicular axes. These axes intersect at point B, which is the base point of the reference frame X attached to the outer gimbal. Its preferred coordinate system is defined such that 3^X is vertical (parallel with 3^L) and 2^X is parallel to the inner gimbal revolution axis. The revolute joint ① connecting the outer gimbal with the Earth frame is along 3^L with joint angle ψ . Due to its size and required strength, the outer gimbal has a non-negligible mass and moment of inertia. Since it has a single rotation *w.r.t.* the Earth frame, only the moment of inertia component along the vertical axis I_{X_3} is considered in the model.

Inner gimbal The inner gimbal is connected to the outer gimbal and the projectile body by two revolute joints with perpendicular axes, intersecting at point B. A reference frame Y is attached to the inner gimbal, with base point B and its preferred coordinate system such that 2^Y is aligned with 2^X and 1^Y is along the projectile's main axis 1^B . The first revolute joint ②, connecting the inner gimbal to the outer gimbal, is along 2^Y , with joint angle θ and the second revolute joint ③ connects the inner gimbal to the projectile body along 1^B with joint angle ϕ . Note that the angular joint parameters are designated with the same nomenclature as for the Euler angles, as the order of transformation is identical to the definition of the Euler angles ('ZYX' convention). The support structure was indeed designed such that the Euler angles measured by the onboard IMU correspond to the revolute joint angles. Finally, the inner gimbal's mass and moment of inertia are negligible compared to the other elements of the system.

2.3.1 Dynamic Model of the Translation-Denied Projectile

The presence of a linkage chain between the projectile body and the Earth frame induces internal reaction forces and moments at each joint. The equations of motion for this multibody system can be obtained either using analytical methods, such as the Lagrange equations or the principle of virtual power, or by successive applications of Newton's and Euler's laws for each body in the system [Wittenburg, 2008]. In the latter case, the joint reaction forces and moments introduce supplementary unknown variables into the equations of motion for each isolated body. These unknown variables can either be solved for (e.g. in order to determine the loads on a bearing), or removed in order to obtain the equations of motion for the complete system. This systematic method however requires the derivation of the complete equation set.

For an open kinematic loop (as it is the case here), the reaction unknowns can also be eliminated by carefully selecting bodies (or subsystems of bodies) in order to keep only one joint between the set of isolated bodies and other bodies in the system. For any given joint, the reaction effort corresponding to the joint's mobility is zero since the joints are assumed to be ideal (frictionless): e.g. for a revolute joint the reaction moment is zero along the joint axis. Then, the projection of Euler's and Newton's laws along the mobility axes does not contain reaction unknowns.

In the present configuration, the linear motion of the *c.m.* is prevented by the support structure, hence only the angular kinematics and dynamics are considered in the following. The motion unknowns are the angular rate derivatives \dot{p} , \dot{q} and \dot{r} , therefore three independent equations are required to obtain the complete angular dynamics. To this end, three subsystems are successively isolated: first the projectile alone ($\Sigma^{17} = B$), then the projectile with the inner gimbal ($\Sigma = B + Y$) and finally the complete system ($\Sigma = B + Y + X$).

The derivation of the associated equations is carried out in Appendix A.1. For each considered subsystem, Euler's law is applied and the resulting vector equation is projected along the adequate joint axis, yielding three scalar equations in total.

For $\Sigma = B$, the projection along 1^B gives:

$$\dot{p}I_1 + (I_3 - I_2)qr = m_{B1} \quad (2.41)$$

¹⁷Here Σ denotes the set of bodies to which Euler's and Newton's laws are applied. For convenience, the individual bodies are designated with the same letter as their preferred coordinate system.

For $\Sigma = B + Y$, the projection along 2^Y yields:

$$[\dot{q}I_2 + pr(I_1 - I_3)] \cos \phi - [\dot{r}I_3 + pq(I_2 - I_1)] \sin \phi = m_{B_2} \cos \phi - m_{B_3} \sin \phi \quad (2.42)$$

For $\Sigma = B + Y + X$, the projection along 3^L gives:

$$\begin{aligned} \ddot{\psi}I_{X_3} - [\dot{p}I_1 + qr(I_3 - I_2)] \sin \theta + [\dot{q}I_2 + pr(I_1 - I_3)] \cos \theta \sin \phi + [\dot{r}I_3 + pq(I_2 - I_1)] \cos \theta \cos \phi \\ = -m_{B_1} \sin \theta + m_{B_2} \cos \theta \sin \phi + m_{B_3} \cos \theta \cos \phi \end{aligned} \quad (2.43)$$

These equations are then solved for the angular rate derivatives \dot{p} , \dot{q} and \dot{r} , resulting in the angular dynamics of the ACHILES projectile in the support structure. In opposition to the free-flight case, these expressions are far from trivial due to the outer gimbal inertia I_{X_3} acting on the yaw axis:

$$\dot{p} = I_1^{-1} [(I_2 - I_3)qr + m_{B_1}] \quad (2.44)$$

$$\dot{q} = \frac{1}{A_2} \left[k_2 \left(1 + \frac{I_{X_3} \cos^2 \phi}{I_3 \cos^2 \theta} \right) - k_3 \frac{I_{X_3} \sin \phi \cos \phi}{I_3 \cos^2 \theta} - k_\psi I_{X_3} \frac{\sin \phi}{\cos^2 \theta} \right] \quad (2.45)$$

$$\dot{r} = \frac{1}{A_3} \left[-k_2 \frac{I_{X_3} \sin \phi \cos \phi}{I_2 \cos^2 \theta} + k_3 \left(1 + \frac{I_{X_3} \sin^2 \phi}{I_2 \cos^2 \theta} \right) - k_\psi I_{X_3} \frac{\cos \phi}{\cos^2 \theta} \right] \quad (2.46)$$

where symbols A_2 , A_3 , k_2 , k_3 and k_ψ are defined as follows:

$$A_2 = I_2 + I_{X_3} \frac{1}{\cos^2 \theta} \left[\sin^2 \phi + \frac{I_2}{I_3} \cos^2 \phi \right] \quad (2.47)$$

$$A_3 = I_3 + I_{X_3} \frac{1}{\cos^2 \theta} \left[\cos^2 \phi + \frac{I_3}{I_2} \sin^2 \phi \right] \quad (2.48)$$

$$k_2 = m_{B_2} - pr(I_1 - I_3) \quad (2.49)$$

$$k_3 = m_{B_3} - pq(I_2 - I_1) \quad (2.50)$$

$$k_\psi = \frac{1}{\cos \theta} [pq \cos \phi - pr \sin \phi + 2(q^2 - r^2) \tan \theta \cos \phi \sin \phi + 2qr \tan \theta (1 - 2 \sin^2 \phi)] \quad (2.51)$$

The projectile's angular kinematics can be described by the Euler angle differential equations:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ q \cos \phi - r \sin \phi \\ q \sin \phi / \cos \theta + r \cos \phi / \cos \theta \end{bmatrix} \quad (2.52)$$

These equations are seldom used in the development of flight simulations due to a singularity at $\theta = \pm\pi/2$ which can severely deteriorates the numerical integration accuracy. Modern flight simulators avoid this singularity by using either the direction cosine matrix (DCM) differential equations or quaternions to compute the angular kinematics [Shuster, 1993]. These methods rely on a higher number of differential equations and require extra transformation steps to obtain the Euler angles. However in the present case, the singularity in θ is not reached since θ remains small, and hence the Euler angle differential equations may be used directly.

2.3.2 Airframe incidence angles

In the ACHILES case, the expressions of the incidence angles given in Equations (2.20)–(2.21) are no longer valid since the support structure ensures that the linear velocities u , v and w are zero. However, recall that the local coordinate system is modified such that 1^L is directed along the wind direction in the wind tunnel test section. As by definition 1^W is directed along the wind, the projection of unit vectors directed respectively along 1^L and 1^W in the body coordinate system yields:

$$[\mathbf{T}]^{BW} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^W = [\mathbf{T}]^{BL} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^L \quad (2.53)$$

$$\begin{bmatrix} \cos \alpha \cos \beta \\ \sin \beta \\ \sin \alpha \cos \beta \end{bmatrix} = \begin{bmatrix} \cos \psi \cos \theta \\ \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi \\ \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \end{bmatrix} \quad (2.54)$$

The incidence angles can then be expressed as functions of the Euler angles:

$$\alpha = \arctan (\tan \theta \cos \phi + \tan \psi \sin \phi \sec \theta) \quad (2.55)$$

$$\beta = \arcsin (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi) \quad (2.56)$$

2.3.3 Roll-Locked Case

In addition to the complete three-degrees of freedom configuration, the roll and yaw axes of the ACHILES support structure can be locked individually to focus on the pitch-yaw and pitch-only behaviors of the guided projectile prototype. This thesis concentrates on these two situations. On one hand, the roll behavior of projectiles is well-known and has already been studied by other research groups [Fresconi et al., 2014; Hann et al., 2012], and on the other hand mechanical difficulties with the roll bearing (high friction) prevent the exploitation of the roll axis.

The roll-locked case consists in imposing $\dot{\phi} = 0$ by locking the joint between the projectile and the inner gimbal. In addition, the projectile roll angle is considered to be zero ($\phi = 0$), such that the 2^B axis lies in the horizontal plane. In this case, the body frame B and inner gimbal frame Y are identical. The procedure for deriving the motion equations is similar to the three-degrees of freedom case, except there are only two remaining unknowns \dot{q} and \dot{r} . The projectile angular velocity vector then becomes:

$$[\boldsymbol{\omega}^{BE}]^B = \begin{bmatrix} -\dot{\psi} \sin \theta \\ \dot{\theta} \\ \dot{\psi} \cos \theta \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.57)$$

In the above equation, both p and r are related to the yaw velocity $\dot{\psi}$. The angular velocity p is no longer independent and can be written as $p = -r \tan \theta$. This situation corresponds to the body-fixed frame (BFF) used for modeling spinning projectiles [Costello and Agarwalla, 2000]. Equation (2.41) is no longer applicable as there is a reaction moment induced by the locked roll joint. However, Equation (2.42) still applies as the locked roll joint reaction forces and moments are internal to the isolated system

$\Sigma = B + Y$, yielding the pitch dynamics:

$$\dot{q} = \frac{1}{I_2} [r^2 \tan \theta (I_1 - I_3) + m_{B_2}] \quad (2.58)$$

Similarly, Equation (2.43) still holds, and its application in the roll-locked case gives:

$$\begin{aligned} \ddot{\psi} I_{X_3} - [\dot{p} I_1 + qr(I_3 - I_2)] \sin \theta + [\dot{r} I_3 + pq(I_2 - I_1)] \cos \theta \\ = -m_{B_1} \sin \theta + m_{B_3} \cos \theta \end{aligned} \quad (2.59)$$

This expression involves the second derivative of ψ and the derivative of p , which are respectively in the roll-locked case:

$$\ddot{\psi} = \frac{d}{dt} \left(\frac{r}{\cos \theta} \right) \quad (2.60)$$

$$= \frac{1}{\cos^2 \theta} (\dot{r} \cos \theta - qr \sin \theta) \quad (2.61)$$

$$\dot{p} = \frac{d}{dt} (-r \tan \theta) \quad (2.62)$$

$$= -\dot{r} \tan \theta - qr \frac{1}{\cos^2 \theta} \quad (2.63)$$

Rewriting Equation (2.59) using the above then yields the angular dynamics in r :

$$\dot{r} = \frac{\cos^2 \theta}{I_{X_3} + I_1 \sin^2 \theta + I_3 \cos^2 \theta} \left[qr \tan \theta \left(I_3 - I_1 - \frac{I_{X_3} + I_1}{\cos^2 \theta} \right) - m_{B_1} \tan \theta + m_{B_3} \right] \quad (2.64)$$

The projectile angular dynamics about the 3^B axis are much more complicated in the roll-locked case than in the free-flight case. In the roll-free cases, the roll moment m_{B_1} only acts on the projectile body, thanks to the roll bearing which decouples it from the support structure. In the roll-locked case, this moment is transferred to the support structure, which acts as a universal (or Cardan) joint between the projectile body and the outer gimbal. This results in a term in m_{B_1} in the expression of \dot{r} . Note that this expression is no longer valid if the projectile is close to vertical ($\theta = \pm \frac{\pi}{2}$).

Finally, the incidence angles equations (2.55), (2.56) trivially reduce to $\alpha = \theta$ and $\beta = -\psi$ in the roll-locked case, while the angular kinematics are simply:

$$\dot{\theta} = q \quad (2.65)$$

$$\dot{\psi} = -r / \cos \theta \quad (2.66)$$

Equations describing the dynamics (2.58), (2.64) and the kinematics (2.65), (2.66) are rewritten in the state-space form using the incidence angle relations and the actual description of the aerodynamic forces and moments given in Section 2.2.3. The state vector is selected as $\mathbf{x} = [\alpha \quad q \quad \beta \quad r]^T$, leading to the non-linear parameter-dependent state-space model:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f} [\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\sigma}(t)] \\ \mathbf{y}(t) &= \mathbf{g} [\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\sigma}(t)] \end{aligned} \quad (2.67)$$

where $\mathbf{u} = [\delta_m \ \delta_n]^\top$ is the input vector and $\boldsymbol{\sigma} = [V \ h]^\top$ is an external parameter vector. The output of this system, which is measured by the ACHILES sensors, is given by $\mathbf{y} = \mathbf{x}$, i.e., the state vector is completely known.

The state functions $\mathbf{f} = [f_\alpha \ f_q \ f_\beta \ f_r]^\top$ are then expressed as:

$$f_\alpha = q \quad (2.68)$$

$$f_q = \frac{1}{I_2} \left[r^2 \tan \alpha (I_1 - I_3) + \bar{q} S d \left(C_{m0} + C_{m\alpha} \alpha + C_{m\beta} \beta + \left(\frac{d}{2V} \right) C_{mq} q + C_{m\delta_m} \delta_m \right) \right] \quad (2.69)$$

$$f_\beta = -r / \cos \alpha \quad (2.70)$$

$$f_r = \frac{\cos^2 \alpha}{I_{X3} + I_1 \sin^2 \alpha + I_3 \cos^2 \alpha} \left[\bar{q} S d \left(C_{n0} + C_{n\alpha} \alpha + C_{n\beta} \beta + \left(\frac{d}{2V} \right) C_{nr} r + C_{n\delta_n} \delta_n \right) + q r \tan \alpha \left(I_3 - I_1 - \frac{I_{X3} + I_1}{\cos^2 \alpha} \right) + \bar{q} S d \left(-C_{l0} + \left(\frac{d}{2V} \right) C_{lp} r \right) \tan \alpha \right] \quad (2.71)$$

2.3.4 Roll/Yaw-Locked Case

A third configuration of the ACHILES support structure consists in locking both the roll and yaw axes, leaving only the pitch axis as the single degree of freedom. The projectile angular velocity vector is then $\boldsymbol{\omega}^{\text{BE}} = \dot{\theta} \mathbf{b}_2$, thus $p = 0$, $q = \dot{\theta}$ and $r = 0$. The equations of motion describing this configuration are trivially obtained from the two-degrees of freedom configuration, as the reaction forces of the locked joints do not impact the pitching motion. The pitch dynamics are then expressed by:

$$\dot{q} = \frac{1}{I_2} \bar{q} S d \left(C_{m0} + C_{m\alpha} \alpha + C_{m\beta} \beta + \left(\frac{d}{2V} \right) C_{mq} q + C_{m\delta_m} \delta_m \right) \quad (2.72)$$

Using the same formalism as for the two-degrees of freedom case, the pitch dynamics and kinematics can be formulated in the nonlinear state-space form of Equation (2.67). The state vector then becomes $\mathbf{x} = [\alpha \ q]^\top$ and the input and outputs are respectively $u = \delta_m$ and $\mathbf{y} = [\alpha \ q]^\top$. An additional assumption is that the yaw angle is set to zero, that is, $\psi = 0$. The nonlinear state equation is then written as:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} q \\ \frac{1}{I_2} \bar{q} S d \left(C_{m0} + C_{m\alpha} \alpha + \left(\frac{d}{2V} \right) C_{mq} q + C_{m\delta_m} \delta_m \right) \end{bmatrix} \quad (2.73)$$

Despite being limited to a single degree of freedom, this model was used with success in the initial design phases for the identification, control and validation studies [Strub et al., 2014, 2016]. This proof of concept stage was a necessary step towards the design of identification and control methods targeted at the two-axes case.

2.4 Trimming and Linearization

The model structures developed in the previous section are nonlinear and are thus not directly suitable for robust control techniques operating on linear parameter-varying (LPV) or linear time-invariant (LTI) systems. To this end, these nonlinear dynamics are approximated using a family of linearized models computed for a selection equilibrium points of the system. This procedure is divided in two steps. First, a *stick-fixed* trimming procedure is carried out to obtain the trim maps of the projectile, then the linearized dynamics are derived for any given operating point. The obtained linearized structure then serves as the basis for system identification and controller design.

2.4.1 Trimming

In aviation, *trimming* refers to a procedure which consists in setting an aircraft's controls such that the vehicle keeps a constant attitude without requiring constant pilot action. When properly trimmed, the system is effectively at an equilibrium: aerodynamic moments acting on the body and wings are entirely compensated by moments induced by the control surfaces. More generally, finding the trim map of any flying vehicle consists in solving the equilibrium condition for a given value of the parameter vector $\bar{\sigma}$:

$$\dot{\bar{x}} \triangleq 0 = \mathbf{f}(\bar{x}, \bar{u}, \bar{\sigma}) \quad (2.74)$$

where the bar symbols denote the equilibrium values of the involved variables. In this equation, both the equilibrium state vector $\bar{x} = [\bar{\alpha} \ \bar{q} \ \bar{\beta} \ \bar{r}]^\top$ and the equilibrium input vector \bar{u} are unknowns. This results in an under-constrained problem as there are more unknowns than equations.

This issue is solved by imposing a given value for the inputs or a part of the state (e.g. in aerospace applications it may be the incidence angles α and β), hence forming an extended parameters (or trim) vector $\rho = [\sigma \ x]$. The equilibrium solutions are usually solved numerically [Garza and Morelli, 2003], as an analytical solution may be difficult to compute. The flight envelope is discretized and the trimming procedure is repeated for each operating point, resulting in a multi-dimensional equilibrium manifold.

In the ACHILES case, the actual values of the aerodynamic coefficients are *a priori* unknown. Consequently, the numerical resolution of (2.74) is not possible, and hence the trim map is may only be determined experimentally. In the two-degrees-of-freedom case, applying the equilibrium condition (2.74) to the nonlinear state equations (2.68)–(2.71) yields $\bar{q} = 0$ and $\bar{r} = 0$: the projectile has a constant attitude at equilibrium. The projectile's trim map can then be determined by measuring the steady-state angles of incidence $\bar{\alpha}$ and $\bar{\beta}$ for any given control inputs $\bar{\delta}_m$ and $\bar{\delta}_n$, corresponding to the so-called *stick-fixed* trimming case. In this case, the trim vector is composed of the control inputs $\mathbf{u} = [\delta_m \ \delta_n]^\top$ and the operating conditions $\sigma = [V \ h]^\top$, that is, $\rho = [\sigma \ \mathbf{u}]^\top$.

2.4.1.1 Experimental Trimming Procedure

The experimental trimming procedure consists of the following steps:

1. Impose the projectile operating conditions $\sigma = [V \ h]^\top$. In the ACHILES experimental setup, the altitude h remains constant and the airspeed V can be adjusted from the wind tunnel's control panel.
2. Define an equidistant gridding for the controls δ_m and δ_n along their operating range.

3. For each trim condition ρ , apply the virtual deflections δ_m and δ_n to the four canards using the mixing logic of Equation (2.34), and record the steady-state value of the incidence angles α and β . These quantities are respectively measured by the IMU and the structure-mounted yaw encoder.
4. The measured trim incidence angles are subsequently stored in two respective two-dimensional arrays indexed by the trim fin deflections.

This procedure assumes that the projectile is open-loop stable. For unstable projectiles, a baseline stabilizing controller would be required. In this case, the equilibrium points are unstable and may be determined by fixing either the initial control inputs δ_m and δ_n , or the target incidence angles α and β , then measuring the steady-state value of the free parameters. The flight envelope then corresponds to the domain in which the projectile can be brought to a static equilibrium.

2.4.1.2 Experimental Results

The above trimming procedure was applied to the ACHILES projectile prototype for an airspeed range of $15 \leq V \leq 40$ m/s. The virtual fin deflections δ_m and δ_n are considered in the $[-10^\circ, 10^\circ]$ range with 2° increments, resulting in a uniform 11×11 grid. These limits have been determined experimentally and roughly correspond to the usable projectile flight domain. Outside of this domain the canards enter a stalling regime, which results in a reduced control efficiency and may also lead to quasi-unstable behavior (limit cycles) for high deflections.

The overall shape of the projectile's flight envelope is illustrated in Figure 2.6, which represents the trim angle of attack $\bar{\alpha}$ and angle of sideslip $\bar{\beta}$ for each considered trim condition $\bar{\rho}$. This three-dimensional representation is built upon 'slices' corresponding to the evaluation of the trimming procedure for each value of the airspeed. These slices have identical dimensions and shapes, except for $V = 15$ m/s and $V = 20$ m/s, for which the slices are slightly smaller and offset from the center axis towards positive values of the yaw trim. The smaller slice size at low airspeeds correspond to a reduced flight envelope due to lower canard effectiveness, while the bias in the yaw measurements is due to a small torsional moment in the support structure's yaw axis caused by a minor misalignment in the support's yaw axis. At higher airspeeds, this biasing moment becomes insignificant *w.r.t.* the aerodynamic moments.

The shape of these flight envelope slices are discussed in greater detail thereafter, using the trim data for $V = 25$ m/s. The individual trim surfaces corresponding to the incidence angles α and β are respectively represented in Figures 2.7A and 2.7B, while the corresponding slice has been isolated in Figure 2.7C. The trim surfaces exhibit good linearity for control inputs in the $[-8^\circ, 8^\circ]$ range, and show no visible static cross-axes coupling: the trim angle of attack $\bar{\alpha}$ depends almost only of the virtual pitch control $\bar{\delta}_m$, and the situation is similar for the yaw axis. Cross-axes couplings would manifest as a dependency of each trim angle of incidence to both control inputs. These observations carry over to the flight envelope slice, which corresponds to the mapping of the $(\bar{\delta}_m, \bar{\delta}_n)$ grid to a $(\bar{\alpha}, \bar{\beta})$ grid. The resulting grid is relatively uniform with almost straight constant-deflection lines. The boundaries of the flight envelope are visible at high values of the fins deflections : the constant-deflection lines are closer, which corresponds to reduced canard efficiency.

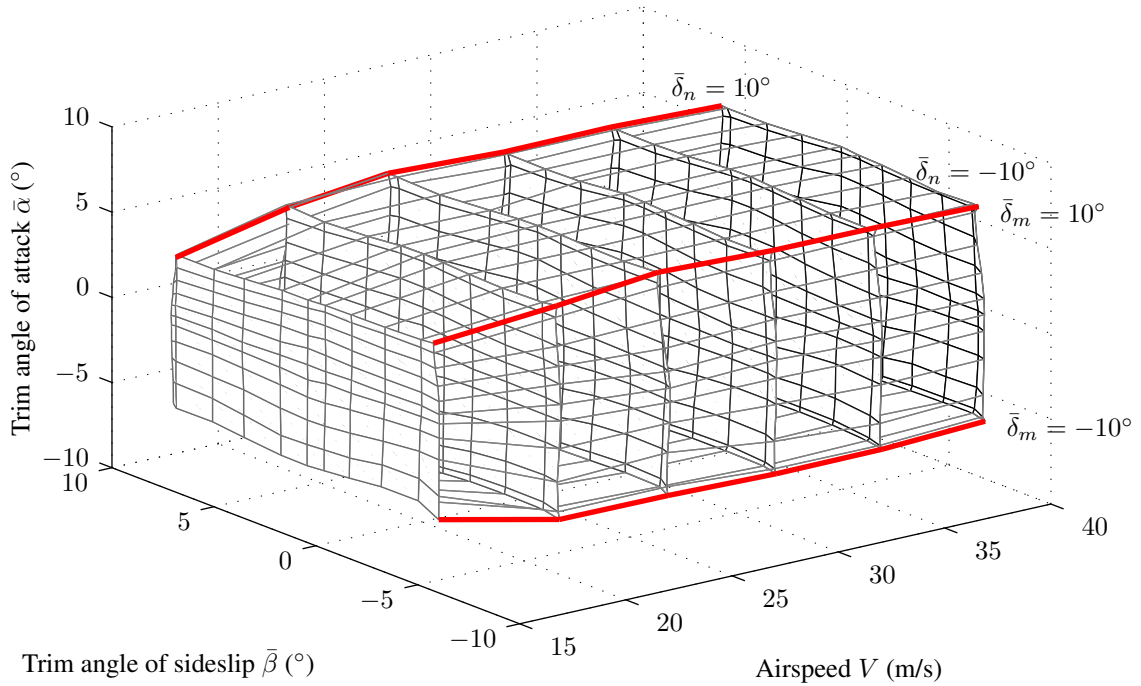


FIGURE 2.6: Overview of the ACHILES projectile flight envelope for airspeeds ranging from 25 m/s to 40 m/s. Each vertical slice corresponds to the mapping from a uniform 11×11 grid in $(\bar{\delta}_m, \bar{\delta}_n)$ to an $(\bar{\alpha}, \bar{\beta})$ grid for a constant airspeed. Red lines denote the extreme constant-deflection values.

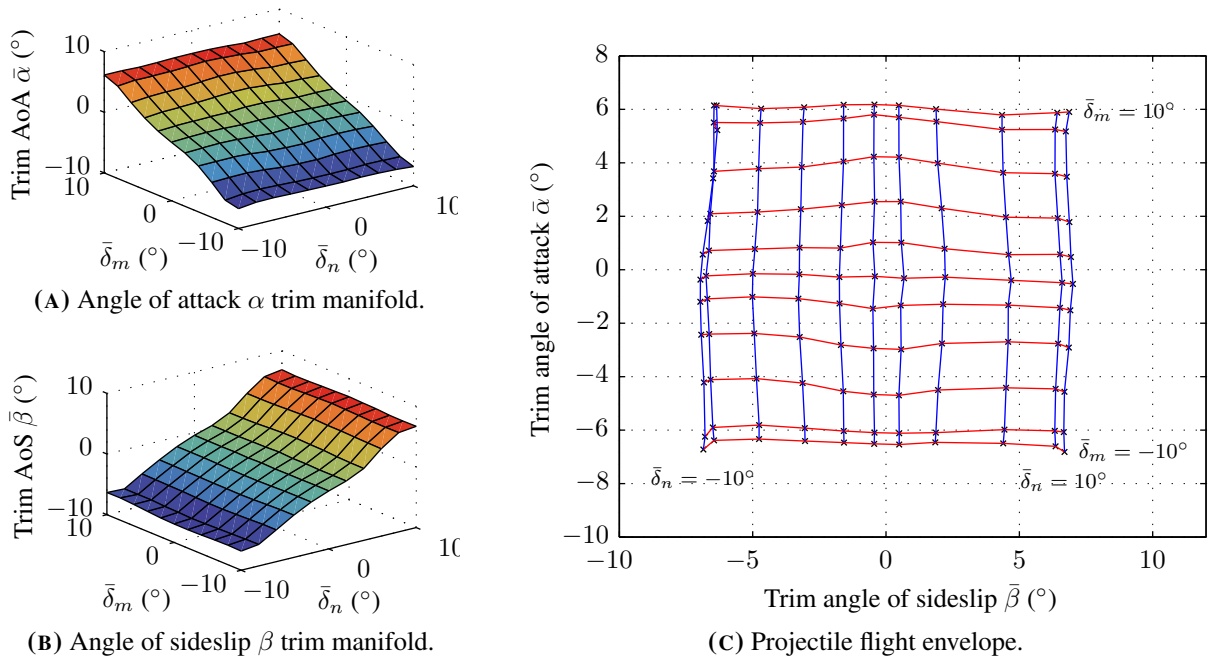


FIGURE 2.7: Trim surfaces for (A) the angle of attack α and (B) the angle of sideslip β , and (C) projectile trim map, obtained at $h = 0$ and $V = 25$ m/s and based on an uniform 11×11 grid of control deflections $(\bar{\delta}_m, \bar{\delta}_n)$ in the range $[-10^\circ, 10^\circ]$.

2.4.2 Linearization

The linearization step consists in approximating the nonlinear projectile behavior described in Equations (2.67-2.71, 2.73) using a family of linear systems. For any trim point $\bar{\rho}$ and its associated equilibrium conditions \bar{x} , \bar{u} , \bar{y} , the nonlinear state-space equation (2.67) can be approximated using series expansion theory [Leith and Leithead, 2000]. Neglecting the higher-order terms, the projectile behavior in the vicinity of $\bar{\rho}$ is given by the following linear state-space system:

$$\begin{aligned}\dot{\mathbf{x}}_\delta(t) &\simeq \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\bar{\rho}} \mathbf{x}_\delta(t) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\bar{\rho}} \mathbf{u}_\delta(t) \\ \mathbf{y}_\delta(t) &\simeq \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\bar{\rho}} \mathbf{x}_\delta(t) + \left. \frac{\partial \mathbf{g}}{\partial \mathbf{u}} \right|_{\bar{\rho}} \mathbf{u}_\delta(t)\end{aligned}\quad (2.75)$$

where $\mathbf{x}_\delta = \mathbf{x} - \bar{\mathbf{x}}$, $\mathbf{u}_\delta = \mathbf{u} - \bar{\mathbf{u}}$ and $\mathbf{y}_\delta = \mathbf{y} - \bar{\mathbf{y}}$ are deviations from equilibrium. However, this description is only valid in the local neighborhood of the considered operating condition ρ , restricting the validity domain of the linearized model.

In order to span the complete projectile flight envelope, multiple operating points covering the trim map at each airspeed are considered, forming a family of linear systems indexed by the trim conditions¹⁸:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(\bar{\rho})\mathbf{x}(t) + \mathbf{B}(\bar{\rho})\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}(\bar{\rho})\mathbf{x}(t) + \mathbf{D}(\bar{\rho})\mathbf{u}(t)\end{aligned}\quad (2.76)$$

where the \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} matrices correspond to the partial derivatives found in Equation (2.75). This representation is similar to a *quasi*-linear parameter-varying (*quasi*-LPV) form, since the model depends on both external and state-dependent parameters. However, it must be stressed that this family of linear models is not a direct representation of the nonlinear projectile behavior. Indeed, the state and output equations (2.76) describe perturbation quantities around a given equilibrium point, while the nonlinear state and output equations (2.67) describe the complete system behavior. As a result, the *quasi*-LPV form may not be used directly for numerical simulations of the projectile. However, it is possible to establish *a relationship between the local stability of a non-linear system and the stability of the associated series expansion equilibrium linearizations* [Leith and Leithead, 2000], which allows to use families of linear systems as an approximation of a nonlinear system for control law design.

The state-space matrices \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} are obtained by applying the first-order series expansion of Equation (2.75) to the ACHILES nonlinear state-space dynamics. In the roll-locked 2-DoF case, these dynamics are represented by Equations (2.68)–(2.71) and the state vector $\mathbf{x} = [\alpha \ p \ \beta \ q]^\top$ is entirely measured, thus $\mathbf{y} = \mathbf{x}$. Recalling that at equilibrium, $\bar{q} = 0$ and $\bar{r} = 0$, the state, input and output matrices are:

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array} \right] = \left[\begin{array}{cccc|cc} 0 & 1 & 0 & 0 & 0 & 0 \\ M_{q\alpha} & M_{qq} & M_{q\beta} & 0 & M_{q\delta_m} & 0 \\ 0 & 0 & 0 & -\frac{1}{\cos \alpha} & 0 & 0 \\ M_{r\alpha} & 0 & M_{r\beta} & M_{rr} & 0 & M_{q\delta_n} \\ \hline & & \mathbb{I}_4 & & \mathbb{O}_{4 \times 2} & \end{array} \right] \quad (2.77)$$

¹⁸In the following, the δ subscript and the approximation sign are omitted in order to simplify the notation.

where \mathbb{I}_4 denotes the 4×4 identity matrix and $\mathbb{O}_{4 \times 2}$ denotes a 4×2 null matrix, i.e., the system is strictly proper (no direct feedthrough). This model structure is observable and controllable, as shown in Appendix A.3

The M_{ij} coefficients in the expression of **A** and **B** are functions of the trim vector ρ , and their detailed expressions are obtained by evaluating the partial derivatives of the state functions with respect to the state and input vectors, as in Equation (2.75):

$$M_{q\alpha} = \frac{1}{I_2} \bar{q} S d \left(C_{m\alpha} + \frac{\partial C_{m0}}{\partial \alpha} + \bar{\alpha} \frac{\partial C_{m\alpha}}{\partial \alpha} + \bar{\beta} \frac{\partial C_{m\beta}}{\partial \alpha} + \bar{\delta}_m \frac{\partial C_{m\delta_m}}{\partial \alpha} \right) \quad (2.78)$$

$$M_{qq} = \frac{1}{I_2} \bar{q} S d \left(\frac{d}{2V} \right) C_{mq} \quad (2.79)$$

$$M_{q\beta} = \frac{1}{I_2} \bar{q} S d \left(C_{m\beta} + \frac{\partial C_{m0}}{\partial \beta} + \bar{\alpha} \frac{\partial C_{m\alpha}}{\partial \beta} + \bar{\beta} \frac{\partial C_{m\beta}}{\partial \beta} + \bar{\delta}_m \frac{\partial C_{m\delta_m}}{\partial \beta} \right) \quad (2.80)$$

$$M_{q\delta_m} = \frac{1}{I_2} \bar{q} S d C_{m\delta_m} \quad (2.81)$$

$$M_{r\alpha} = \frac{I_{X3} + 3(I_3 - I_1) \cos^2 \alpha + I_1}{(I_{X3} + I_1 \sin^2 \alpha + I_3 \cos^2 \alpha)^2} \cdot \bar{q} S d (C_n \cos \alpha \sin \alpha - C_{l0} \sin^2 \alpha) + A \bar{q} S d \left[C_{l0} (2 \cot \alpha - 3) - 3C_n + C_{n\alpha} + \frac{\partial C_{n0}}{\partial \alpha} + \bar{\alpha} \frac{\partial C_{n\alpha}}{\partial \alpha} + \bar{\beta} \frac{\partial C_{n\beta}}{\partial \alpha} + \bar{\delta}_n \frac{\partial C_{n\delta_n}}{\partial \alpha} - \frac{\partial C_{l0}}{\partial \alpha} \tan \alpha \right] \quad (2.82)$$

$$M_{r\beta} = A \bar{q} S d \left(C_{n\beta} + \frac{\partial C_{n0}}{\partial \beta} + \bar{\alpha} \frac{\partial C_{n\alpha}}{\partial \beta} + \bar{\beta} \frac{\partial C_{n\beta}}{\partial \beta} + \bar{\delta}_n \frac{\partial C_{n\delta_n}}{\partial \beta} - \frac{\partial C_{l0}}{\partial \beta} \tan \alpha \right) \quad (2.83)$$

$$M_{rr} = A \bar{q} S d \left(\frac{d}{2V} \right) [C_{nr} + C_{lp} \tan^2 \alpha] \quad (2.84)$$

$$M_{r\delta_n} = A \bar{q} S d C_{n\delta_n} \quad (2.85)$$

where all the C_{ij} are evaluated at the trim point $(\bar{V}, \bar{\alpha}, \bar{\beta})$ and where the factor A is introduced for readability purposes, with

$$A = \frac{\cos^2 \alpha}{I_{X3} + I_1 \sin^2 \alpha + I_3 \cos^2 \alpha}$$

Note that the complexity of the $M_{r\alpha}$ coefficient is mainly due to the trigonometric terms in α in the expression of \dot{r} in Eq. (2.71), which are differentiated *w.r.t.* α during linearization. If these terms are assumed to be constant during linearization, the resulting expression of $M_{r\alpha}$ has the same form as $M_{r\beta}$:

$$M_{r\alpha} = A \cdot \bar{q} S d \left(C_{n\alpha} + \frac{\partial C_{n0}}{\partial \alpha} + \bar{\alpha} \frac{\partial C_{n\alpha}}{\partial \alpha} + \bar{\beta} \frac{\partial C_{n\beta}}{\partial \alpha} + \bar{\delta}_n \frac{\partial C_{n\delta_n}}{\partial \alpha} - \frac{\partial C_{l0}}{\partial \alpha} \tan \alpha \right)$$

For any operating point ρ , the projectile behavior in its neighborhood is entirely determined by the values of the M_{ij} coefficients. The exact knowledge of the aerodynamic coefficients and the projectile mass properties is thus no longer necessary, as the M_{ij} coefficients can be determined directly through system identification. This is a more direct approach compared to the painstaking procedure of aerodynamic coefficient determination through numerical simulations (CFD) or aerodynamic balance measurements.

2.5 System Identification

In the general case, a system identification problem consists of two inverse problems, namely determining a model structure and estimating its parameters. In the present case, the model structure is derived from a linearized first-principles model and some parameters are free unknowns. Such a model is called a *grey-box* model [Bohlin, 2006], and in this case the system identification problem is reduced to parameter estimation. In both cases, it is necessary to ensure the problem is well-posed and well-conditioned in order to guarantee the validity of the solution. In this respect, these conditions are assessed in the first steps of the identification procedure presented thereafter and followed throughout this section.

This section focuses on estimating the parameters of the linearized state-space model from experimentally collected data. These parameters cannot be determined directly in an easy way, because of their dependence upon unknown aerodynamic derivatives. This inverse problem, does not only depend on the model structure, but also on the quality of the input/output data and thus requires careful attention.

2.5.1 Identification Procedure

Due to the nature of the parameter estimation problem, several tasks must be performed before actually collecting any experimental data and proceeding with estimation, in order to guarantee the validity of the results. The identification procedure presented hereafter, and illustrated in Figure 2.8 was adapted from [Franceschini and Macchietto, 2008; Dobre, 2010; Albisser, 2015]. This procedure consists of several steps introduced below:

1. Construct a mathematical model structure describing the behavior of the system.
2. Assess the theoretical or *a priori* identifiability of the model structure. This step consists in analyzing the existence and uniqueness of a solution to the parameter estimation problem in an idealized framework. This is a necessary condition which results in a set of theoretically identifiable parameters, but does not guarantee the success of parameter estimation from real data.
3. Determine an initial estimate of the model parameters, using for example *a priori* knowledge or a first data-collection experiment.
4. Perform an *a posteriori* identifiability analysis, which consists in assessing the uniqueness of the solution for given input-output data. In order to ensure the estimation problem is well-conditioned, the set of practically identifiable parameters is selected based on the sensitivity of the outputs with respect to the model parameters.
5. Design an optimal input signal, that is, a sufficiently exciting input signal such that the collected experimental data are informative, in order to maximize parameter influence on the output and minimize uncertainty on the parameter estimates. In practice, this step is carried out simultaneously with the sensitivity analysis.
6. Collect experimental data using the previously designed input signal.
7. Estimate the model parameters and compute the uncertainty on the estimates.

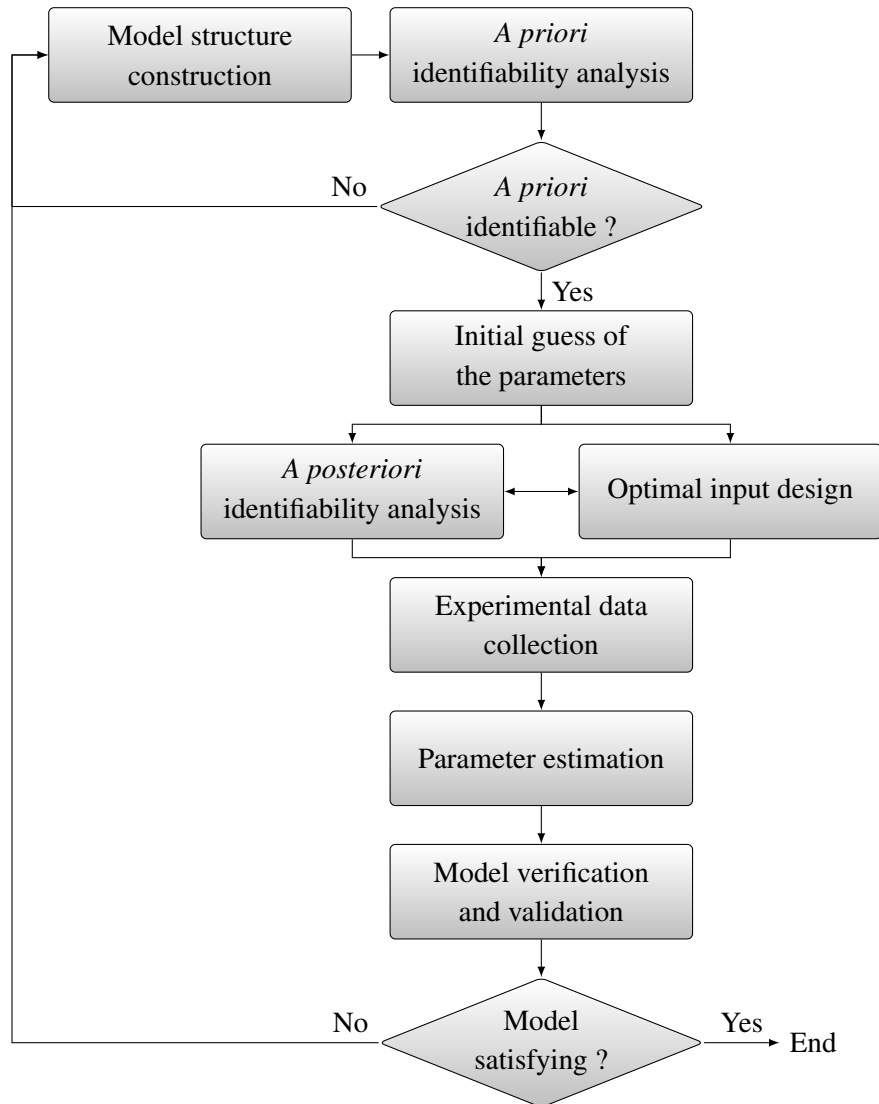


FIGURE 2.8: System identification procedure.

2.5.2 Model Postulation

The model structure considered herein was presented in Equations (2.76) and (2.77). In this case, the parameters to be estimated are the M_{ij} coefficients of the \mathbf{A} and \mathbf{B} matrices. In the considered model structure, the body rates p and q are directly related to the derivatives of the incidence angles α and β . Since these measurements do not bring supplementary knowledge on the system state, they are removed from the output vector to relax constraints on the estimation problem. In order to account for measurement and process noise, an error model consisting in a 4×2 disturbance matrix \mathbf{K} and a residuals signal $e(t)$, assumed to be uncorrelated white noise, are incorporated into the model. This results in the following model structure:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(\boldsymbol{\theta})\mathbf{x}(t) + \mathbf{B}(\boldsymbol{\theta})\mathbf{u}(t) + \mathbf{K}(\boldsymbol{\theta})\mathbf{e}(t) \\ \mathbf{y}(t) &= \mathbf{C}(\boldsymbol{\theta})\mathbf{x}(t) + \mathbf{D}(\boldsymbol{\theta})\mathbf{u}(t) + \mathbf{e}(t) \end{aligned} \quad (2.86)$$

where $\boldsymbol{\theta} = [M_{q\alpha} \ M_{qq} \ M_{q\beta} \ M_{r\alpha} \ M_{r\beta} \ M_{rr} \ M_{q\delta_n} \ M_{r\delta_n}]^\top$ is the vector of parameters to be estimated and the output vector is $\mathbf{y} = [\alpha \ \beta]^\top$. The \mathbf{A} and \mathbf{B} matrices were defined in Equation (2.77), while the expressions of \mathbf{C} and \mathbf{D} are now

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.87)$$

since the body rate measurements are not exploited.

The discrete form of this model structure is called a *directly parametrized innovations form* [Ljung, 1999], where $e(t)$ is the prediction error. This form naturally leads to the estimation of a *predictor* model, where the input and previous outputs are used to estimate the future outputs. In the present case, the estimated model will be used for control law synthesis, therefore the model will *simulate* the projectile dynamics, without using measured outputs. This difference is of great importance when considering the identification procedure. Indeed, in the *predictor* focus, the parameters and the error model have to be estimated simultaneously, whereas in the *simulation* focus, the error model is estimated separately.

2.5.3 *A priori* Identifiability

The *a priori* or theoretical identifiability property of a model structure indicates the existence and uniqueness of a solution to the parameter estimation problem [Walter and Pronzato, 1997]. It is evaluated in an idealized framework where

1. the process and the model have an identical structure (no characterization error),
2. data are noise-free, and
3. input signals u and measurement times can be chosen at will.

Let $\boldsymbol{\theta}$ denote the parameters of the model and $\boldsymbol{\theta}^*$ the parameters of the process. The parameter θ_i will be *globally identifiable* if for almost any admissible parameter vector $\boldsymbol{\theta}^*$:

$$\exists u \in \mathbb{R}^{\mathbb{R}^+}, y(t, \boldsymbol{\theta}, u(t)) = y(t, \boldsymbol{\theta}^*, u(t)) \forall t \in \mathbb{R}^+ \Rightarrow \theta_i = \theta_i^* \quad (2.88)$$

Proving global identifiability of a model structure is not always realizable [Dobre, 2010]. A similar *local* identifiability condition exists, which considers the above condition in the neighborhood of $\boldsymbol{\theta}^*$.

There are several techniques for assessing the *a priori* identifiability of a model structure M , based on state isomorphisms [Peeters and Hanzon, 2005], power series expansion [Walter and Pronzato, 1997], Laplace transform [Bellman and Åström, 1970], etc. In the present study, as the model structure of Equation (2.77) is an observable and controllable linear state-space system, a similarity transformation approach [Walter and Pronzato, 1997] has been employed. The application of this method to the problem at hand (detailed in Appendix A.4) concluded that all model parameters θ_i are *globally* identifiable. This global property, which indicates the uniqueness of the parameter vector, should not be confused with the fact that the postulated model is only valid locally (in the neighborhood of an operating point).

2.5.4 A posteriori Identifiability

A *posteriori* identifiability differs from a *priori* identifiability in that the input signal and measurement times are imposed. Even if a parameter is *a priori* identifiable, it may not be identifiable in practice because of a lack of information in the experimental data. For a given model structure, input signal u and measurement times t_k , the parameter θ_i is a *posteriori* identifiable if for almost any admissible parameter vector θ^* :

$$\mathbf{y}(t_k, \boldsymbol{\theta}, \mathbf{u}_k) = \mathbf{y}(t_k, \boldsymbol{\theta}^*, \mathbf{u}_k) \quad \forall k \in \llbracket 1, N \rrbracket \implies \theta_i = \theta_i^*$$

where N is the number of samples. As with *a priori* identifiability, there is a *local* definition where a neighborhood of $\boldsymbol{\theta}^*$ is considered. For a single-output system, this condition can be expressed, with n_p the number of parameters, as [Dobre, 2010]:

$$\forall k \in \llbracket 1, N \rrbracket, \sum_{i=1}^{n_p} \underbrace{\frac{\partial y(t_k, \boldsymbol{\theta}, \mathbf{u}_k)}{\partial \theta_i}}_{s_{y/\theta_i}(t_k, \boldsymbol{\theta}^*, \mathbf{u})} \Big|_{\boldsymbol{\theta}^*} \cdot d\theta_i = 0 \implies d\boldsymbol{\theta} = 0 \quad (2.89)$$

where $s_{y/\theta_i}(t_k, \boldsymbol{\theta}^*, \mathbf{u}_k)$ is the sensitivity of the output with respect to the parameter θ_i , for an input signal u and measurement time t_k , and $d\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}_0$ is the difference with the nominal parameters vector $\boldsymbol{\theta}_0$.

The parameters and model outputs may have different measurement units and/or orders of magnitude. In order to allow dimension-free comparison of the sensitivity functions, they must first be normalized *w.r.t.* the output variables and the parameters. In the present study, as both outputs have the same units and order of magnitude, the sensitivity functions are only scaled *w.r.t.* the model parameters as follows:

$$\hat{s}_{y/\theta_i}(t_k, \boldsymbol{\theta}^*, \mathbf{u}_k) = \theta_i \cdot s_{y/\theta_i}(t_k, \boldsymbol{\theta}^*, \mathbf{u}_k) \quad (2.90)$$

From the above two equations, the parameters $\boldsymbol{\theta}^*$ is a *posteriori* identifiable if the empirical Fisher Information Matrix

$$\mathbf{F} = \hat{\mathbf{S}}_y(\boldsymbol{\theta}^*, \mathbf{u})^\top \hat{\mathbf{S}}_y(\boldsymbol{\theta}^*, \mathbf{u}) \quad (2.91)$$

is full rank, with $\hat{\mathbf{S}}_y(\boldsymbol{\theta}^*, \mathbf{u})$ the sensitivity matrix defined as:

$$\hat{\mathbf{S}}_y(\boldsymbol{\theta}^*, \mathbf{u}) = \begin{bmatrix} \hat{s}_{y/\theta_1}(t_1, \boldsymbol{\theta}^*, \mathbf{u}_k) & \cdots & \hat{s}_{y/\theta_{n_p}}(t_1, \boldsymbol{\theta}^*, \mathbf{u}_k) \\ \vdots & \ddots & \vdots \\ \hat{s}_{y/\theta_1}(t_N, \boldsymbol{\theta}^*, \mathbf{u}_k) & \cdots & \hat{s}_{y/\theta_{n_p}}(t_N, \boldsymbol{\theta}^*, \mathbf{u}_k) \end{bmatrix} \quad (2.92)$$

This condition can be extended to a system with n_y outputs [Dobre, 2010] with the $N \cdot n_y \times n_p$ global sensitivity matrix:

$$\hat{\mathbf{S}}_y(\boldsymbol{\theta}^*, \mathbf{u}) = \left[\hat{\mathbf{S}}_{y_1}(\boldsymbol{\theta}^*, \mathbf{u})^\top, \cdots, \hat{\mathbf{S}}_{y_{n_p}}(\boldsymbol{\theta}^*, \mathbf{u})^\top \right]^\top \quad (2.93)$$

where the $\hat{\mathbf{S}}_{y_i}(\boldsymbol{\theta}^*, \mathbf{u})^\top$ matrices are defined as in Equation (2.92). This corresponds to a vertical concatenation of the sensitivity matrices of each individual output.

If the empirical Fisher Information Matrix \mathbf{F} is not full rank, or has full rank but is ill-conditioned, then some model parameters are not *a posteriori* identifiable and a subset of *a posteriori* identifiable parameters has to be determined. In that direction, several procedures exist in the literature [Banks et al., 2012; Weijers and Vanrolleghem, 1997] which consist in finding parameter subsets satisfying the following two conditions:

- (\mathcal{C}_1) The model outputs must be sufficiently sensitive to individual changes in each parameter.
- (\mathcal{C}_2) The sensitivity functions must be linearly independent, as changes in the model output due to a variation of one parameter may be compensated by appropriate changes in the other parameters.

The (\mathcal{C}_1) condition is assessed by ranking the parameters in decreasing order of importance, using parameter importance indices based on the scaled sensitivity functions [Brun et al., 2001; Omlin et al., 2001]. One suitable ranking criterion is the mean-square distance δ_i^{msqr} , defined as:

$$\delta_i^{msqr} = \sqrt{\frac{1}{N} \sum_{k=1}^N \hat{\mathbf{s}}_{\mathbf{y}/\theta_i}(t_k, \boldsymbol{\theta}^*, \mathbf{u}_k)^2} \quad (2.94)$$

This first condition only takes into consideration the effect of an individual parameter variation on the output. Even if the individual parameters have strong influence on the model output, the solution may be non-unique due to linear dependence between the sensitivity functions. In order to evaluate the degree of colinearity, the second condition (\mathcal{C}_2) is based on a colinearity index γ_K computed for all parameter subsets. The colinearity index γ_K is defined as:

$$\gamma_K = \text{cond}(\mathbf{F}_K(\boldsymbol{\theta}^*, \mathbf{u})) \quad (2.95)$$

with \mathbf{F}_K being the empirical Fisher Information Matrix related to the parameter subset $\boldsymbol{\theta}_K^*$, $n_K < n_p$:

$$\mathbf{F}_K = \hat{\mathbf{S}}_{\mathbf{y}}(\boldsymbol{\theta}_K^*, \mathbf{u})^\top \hat{\mathbf{S}}_{\mathbf{y}}(\boldsymbol{\theta}_K^*, \mathbf{u}) \quad (2.96)$$

The index γ_K is computed for each possible subset and quantifies the degree of linear dependence between the sensitivity functions of the model outputs *w.r.t.* the parameters in the subset. In order to qualify the degree of colinearity, indicative thresholds are given in [Gujarati and Porter, 2004, p. 362] as follows:

$$\left\{ \begin{array}{ll} \gamma_K < 100 & \text{low colinearity} \\ 100 \leq \gamma_K < 1000 & \text{moderate colinearity} \\ 1000 \leq \gamma_K & \text{strong colinearity} \end{array} \right.$$

A subset of parameters is then *a posteriori* identifiable if its colinearity index γ_K indicates a low to moderate colinearity.

In the present application, the sensitivity functions were derived from the output equation of the considered model (2.86) using the procedure described in Appendix A.5, leaving out the error model as it is estimated separately. Since the parameter vector is not known *a priori*, a first set of parameters must be estimated using experimental data. Results presented hereafter were obtained with the refined input vector designed in the next step of the procedure (see Section 2.5.5), at the zero-incidence trim point for $V = 25$ m/s. Initial tests showed similar trends on the entire flight envelope, hence only a single operating point has been considered throughout this study.

The (\mathcal{C}_1) condition assesses the relative importance of the model parameters $\boldsymbol{\theta}$. To this end, the parameters are ranked in decreasing order of importance using the δ_i^{msqr} criterion, as illustrated in Figure 2.9. The most influential parameters are $M_{r\beta}$ and $M_{r\delta_n}$, which characterize the input/output gain of the yaw channel, and are followed in terms of importance by the pitch channel parameters $M_{q\alpha}$ and $M_{q\delta_m}$.

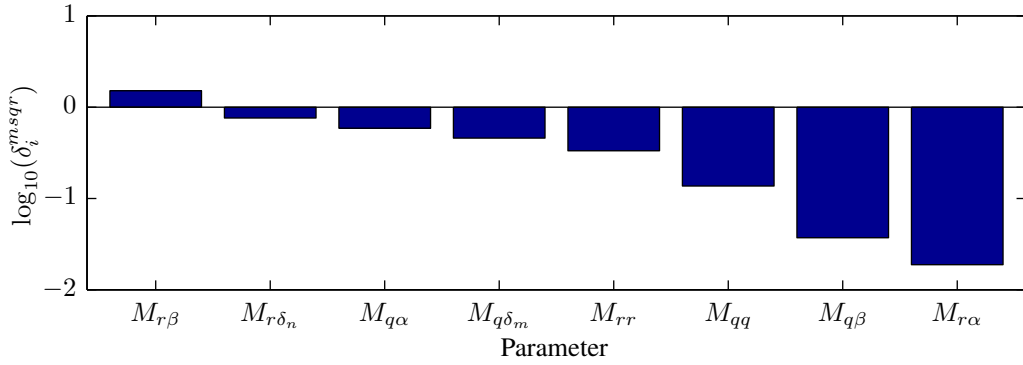


FIGURE 2.9: Classification of model parameters by decreasing importance index δ_i^{msqr} .

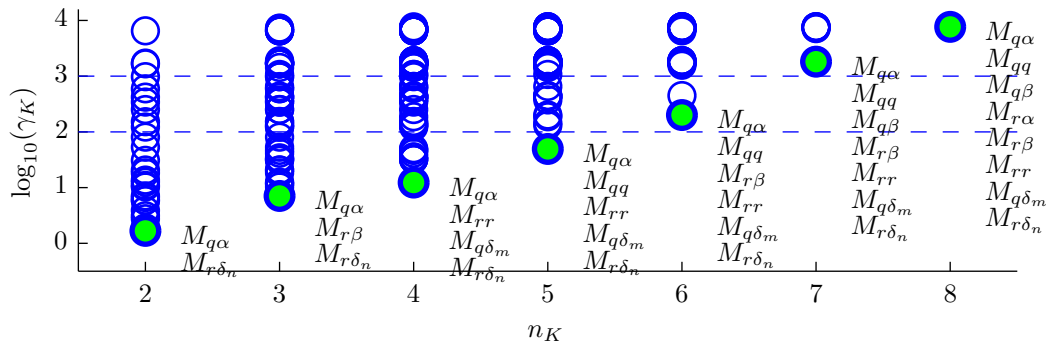


FIGURE 2.10: Classification of model parameters by the multicollinearity index γ_K for parameter subsets, where n_K denotes the size of the subset.

The damping parameters M_{rr} and M_{qq} are less influent, and are associated to the aerodynamic damping derivatives C_{mq} and C_{nr} , which are usually difficult to estimate using traditional techniques [McGowan et al., 2014]. Finally, the coupling coefficients $M_{q\beta}$ and $M_{r\alpha}$ are the least influent parameters, and therefore may not be estimated precisely.

The degree of colinearity is evaluated using the (C_2) condition. The resulting colinearity indices γ_K are represented in Figure 2.10, where the green-filled circles represent the minimal γ_K for each subset dimension. From these results, it appears the uncoupled system exhibits moderate colinearity, while the addition of the coupling parameters $M_{q\beta}$ and $M_{r\alpha}$ introduces severe colinearity, suggesting these parameters are non-identifiable given the chosen *a priori* parameter vector. This limitation may be overcome by estimating the uncoupled model parameters first, by forcing $M_{q\beta}$ and $M_{r\alpha}$ to zero. These coupling parameters are then estimated in a second step, with the other parameters fixed to the values found in the first step.

2.5.5 Experiment Design

The design of the data-collecting experiment has a direct influence on the observed data and is crucial for successful estimation of the model parameters. In general, experiment design involves the selection of several design variables: the measured signals and manipulated inputs, the sample time, the input signal characteristics and the number of samples.

Depending on the application, several types of input signals may be considered, such as pseudo-

random binary sequences (PRBS), multi-sines or filtered white noise, where the main decision factors are the signal spectrum and crest factor [Ljung, 1999]. For the experiment to be informative enough (i.e. the data contains enough information to allow the complete determination of the model parameters), the persistence of excitation order of the input signals must be higher or equal than the model order: a signal $u(t)$ is said to be persistently exciting of order n if its spectrum $\Phi_u(\omega)$ is different from zero on at least n points in $-\pi \leq \omega \leq \pi$ [Ljung, 1999]. In order to select the input signal parameters, the most commonly used optimal signal design criterion is the maximization of the determinant of the Fisher Information Matrix [Walter and Pronzato, 1997]:

$$J_D(\mathbf{u}) = \det \mathbf{F}(\boldsymbol{\theta}, \mathbf{u}) \quad (2.97)$$

An input signal \mathbf{u}_D which maximizes J_D is called *D-optimal* and minimizes parameter variance.

In the studied case, the outputs of (2.77) are the angles of incidence α_m ¹⁹ and β_m , which are measured using the onboard and external sensors at the main clock rate of 100 Hz. The inputs of this model are the fin deflections $\delta_{m,m}$ and $\delta_{n,m}$, which are measured on the actuators, while the excitation signal is applied to the actuators inputs using the virtual fin commands $\delta_{m,c}$ and $\delta_{n,c}$.

Since the behavior of the projectile prototype around an equilibrium point is modeled using a linear system, a suitable input signal in this case is the PRBS. This class of signal has white-noise-like properties but is deterministic, periodic and most importantly bounded, as the input amplitude must be limited to remain in the neighborhood of the equilibrium point. The clock periods T_{δ_m} and T_{δ_n} of the PRBS signals are chosen based on the j_D criterion, as illustrated in Figure 2.11. In order to limit the duration of the data-collection experiments, the PRBS signal duration is limited to 90 s. In practice, the *D-optimal* bit durations lead to a degraded model fit for the edge cases as the system may not reach steady-state. Therefore, the actually implemented clock periods are four times the *D-optimal* values so that the fit is improved and the input is still close to optimality. The chosen clock periods are then $T_{\delta_m} = 1.28$ s and $T_{\delta_n} = 5.12$ s.

Taking into account the duration limit, this has resulted in a 64 bits PRBS sequence for the pitch axis and a 16 bits sequence for the yaw axes. These sequences are respectively generated using 6-bit and 4-bit linear feedback shift registers (LFSR) with different polynomials and seeds for the estimation

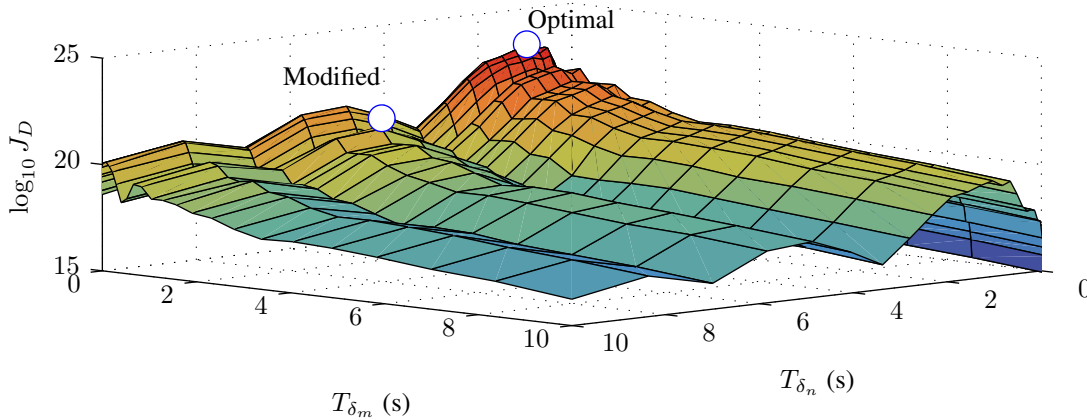


FIGURE 2.11: Input signal D-optimality criterion as a function of the PRBS bit periods T_{δ_m} and T_{δ_n} .

¹⁹The _m subscript denotes a measured quantity.

and validation sequences. As the sequence generated by an LFSR is at most $2^n - 1$ bits long, with n the depth of the LFSR, an extra '0' bit is added at the end of the sequence such that the input signals are zero-mean. For a PRBS sequence of length M , the order of persistence of excitation is $M - 1$. In the present case where a fourth-order model is considered, the data-collection experiment is informative enough since the two input signals are persistently exciting of orders 15 and 63.

2.5.6 Experimental Data Collection

In this step, the optimal input signal is applied to the ACHILES guided projectile in data-collection experiments. In order to cover the complete flight envelope, these experiments are carried out in the neighborhood of several operating points, of which the selection is detailed hereafter.

The equilibrium points on which parameter identification is performed are defined by the operating conditions $\bar{\sigma} = [\bar{V} \quad \bar{h}]^T$ and the trim angles of incidence $\bar{\alpha}$ and $\bar{\beta}$. The selection of these operating points is based on the constant-airspeed trim maps determined in Section 2.4. Therefore, the same airspeed range was considered as for the trim map determination, with \bar{V} varying from 15 m/s to 40 m/s in 5 m/s increments. The selection of the trim angles of incidence must cover adequately the constant-airspeed flight domain while keeping a reasonable experiment duration. Since the trim maps exhibit horizontal and vertical symmetry, it is reasonable to assume that only a single quadrant needs to be estimated. The chosen equilibrium angles of incidence are summarized in Table 2.1 and are considered for each airspeed, except at $V = 15$ m/s where trim points with $\bar{\alpha} > 3$ are removed because of the reduced flight envelope at this speed.

TABLE 2.1: Selected operating points for system identification.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
AoA $\bar{\alpha}$ ($^\circ$)	0	0	0	0	0	1	1	1	1	2	3	3	3	3	5	5	5	5
AoS $\bar{\beta}$ ($^\circ$)	0	1	2	4	6	6	4	2	0	0	0	2	4	6	0	2	4	6

In order to reduce the number of experiments, all operating points for a given airspeed are considered sequentially in the data-collection experiments. The initialization procedure for these experiment first consists in obtaining the corresponding trim fin deflections using the inverse interpolation procedure described in Appendix A.2. These coordinates are projected back on the trim surfaces, where the local slope is used to compute the input signal amplitude such that each input channel (pitch and yaw) produce a $\pm 1^\circ$ steady-state deflection on the incidence angles around the operating point. These experiments are then performed using a data acquisition program built using the MATLAB-based development environment presented in Chapter 1.

A sample sequence of the collected data corresponding to the operating point $V = 25$ m/s, $\bar{\alpha} = 2^\circ$, $\bar{\beta} = 3^\circ$ is presented in Figure 2.12. The pitch and yaw sequences are offset relative to each other: at the beginning and at the end of the sequences, only a single channel is active and the other is at rest, in order to better capture the interaxis coupling. This behavior is notably visible on the pitch axis at the beginning of the sequence. These data also show the sharp contrast between the projectile pitch and yaw dynamics. Even though the aerodynamic structure presents tetrahedral symmetry, the yaw axis is largely affected by the inertia of the support structure, which results in a noticeably lower natural frequency.

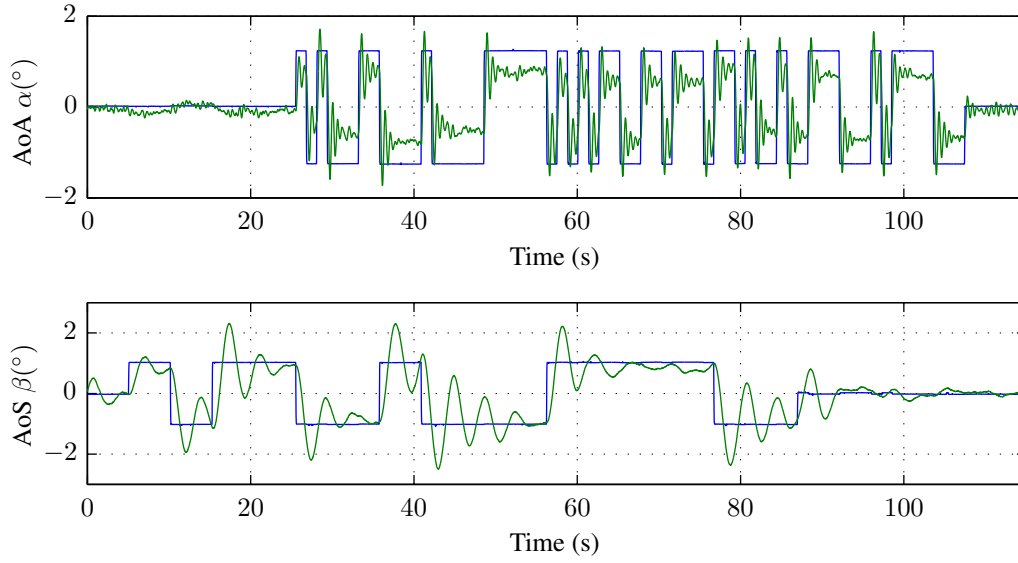


FIGURE 2.12: System input/output behavior around ($V = 25$ m/s, $\bar{\alpha} = 2^\circ$, $\bar{\beta} = 3^\circ$). Top plot is the angle of attack channel, bottom plot is the angle of sideslip channel; blue curves correspond to the input signals $\delta_{m,m}$ and $\delta_{n,m}$, green curves correspond to the measured outputs α_m and β_m .

2.5.7 Parameter Estimation and Uncertainty Analysis

The parameter estimation step is a numerical optimization problem which consists in finding the parameter vector θ that minimizes the quadratic cost function

$$J(\theta) = \sum_{k=1}^N |\mathbf{y}_m(t_k) - \mathbf{y}(t_k, \theta, \mathbf{u}_k)|^2 \quad (2.98)$$

where $\mathbf{y}(t_k, \theta, \mathbf{u}_k)$ is the model output and $\mathbf{y}_m(t_k)$ is the corresponding measured output vector at sampling instant t_k . In general, this problem is non-convex and algorithms such as the Gauss-Newton method or the Levenberg-Marquardt procedure [Walter and Pronzato, 1997] are employed.

Since the experimental data are affected by noise, and some parameters have a low influence on the output, there is a need for establishing a degree of confidence on the estimated parameter values. The uncertainty on parameters estimates is also useful for building uncertain models that are used for robustness analysis during control law design. Intuitively, this uncertainty can be determined by repeating the data-collection and parameter estimation several times for each operating point, then observing the distribution of estimated parameters. By all means, such an approach is hardly practical due to the duration of the experiments. The experiments can however be repeated virtually using *bootstrap* techniques [Efron and Tibshirani, 1993; Zoubir and Boashash, 1998], which generate fictive data vectors from the measured data using the following procedure:

- Estimate the parameter vector $\hat{\theta}$ from the measured input and output vectors \mathbf{u} and \mathbf{y} .
- Compute the residual vector \mathbf{e} using these values and the noise model.
- The fictive data sets are generated as follows:
 - Create a fictive residuals vector \mathbf{e}_f by applying random permutations to \mathbf{e}

- Using the original input \mathbf{u} , the fictive residuals e_f and the estimated parameters vector $\hat{\boldsymbol{\theta}}$, generate a fictive output vector \mathbf{y}_f .

The parameter estimation procedure is then repeated for each of these fictive data sets, yielding a population of estimated parameters of which the statistical characteristics can be studied. The confidence interval for each parameter may then be determined using the standard error or percentiles of this population, and the relative uncertainty on each parameter θ_i is:

$$\Delta\theta_i = \frac{\max(|\hat{\theta}_i - \theta_{i\max}|, |\hat{\theta}_i - \theta_{i\min}|)}{|\hat{\theta}_i|} \quad (2.99)$$

In order to ensure convergence to the global minimum of $J(\boldsymbol{\theta})$, the initial parameter vector $\boldsymbol{\theta}_0$ must be chosen carefully. To this end, the initial values are based on the observed behavior of the open-loop system and *a priori* aerodynamics knowledge. Static stability implies the stability derivatives $M_{q\alpha}$, $M_{r\beta}$ are negative with respect to the incidence angles ($M_{r\beta} > 0$ as $\beta = -\psi$). Similarly, dynamic stability implies the damping coefficients M_{qq} , M_{rr} are all negative. Finally, the fin deflections are defined such that a positive deflection induces a positive change in the incidence angles, thus $M_{q\delta_m} > 0$ and $M_{r\delta_m} < 0$.

In practice, the estimation problem is solved in two steps to take into account practical limitations on parameters identifiability: the parameters describing uncoupled dynamics ($M_{q\alpha}$, M_{qq} , $M_{r\beta}$, M_{rr} , $M_{q\delta_n}$ and $M_{r\delta_n}$) are estimated first, with the coupling parameters $M_{q\beta}$, $M_{r\alpha}$ set to zero. These parameters are estimated separately using the same input/output data and with the uncoupled dynamics parameters being fixed to the values found in the first step. Finally, an identical procedure aims at minimizing the quadratic norm of the prediction error $\mathbf{y}_p(t_k, \boldsymbol{\theta}, \mathbf{u}_k, \mathbf{y}_k) - \mathbf{y}_m(t_k)$ to determine the error matrix \mathbf{K} .

2.6 Parameter Estimation Results

As mentioned previously, the parameters of the model depend on three trim parameters: the airspeed and the incidence angles. Thus, in order to properly cover the projectile flight envelope, 104 different operating points have been considered for the estimation step, as presented in Section 2.5.6. In addition to presenting the estimated parameter values, it is also essential to assess the model validity for each of these operating points. One of the main objectives of this work is to develop a full flight-envelope autopilot, based on gain scheduling techniques. In this regard, the airspeed V is a *slow* parameter, while the incidence angles are *fast* parameters. It then makes sense to study the dependence of the parameters on the airspeed and on the incidence angles separately.

For practical reasons, this section is divided as follows. First, the overall trends over the complete flight envelope are discussed in Section 2.6.1. The model evaluation results over the flight domain are investigated in Section 2.6.2, and the parameter uncertainties are discussed in Section 2.6.3. Constant-speed results are discussed for $V = 25$ m/s and extended to the other airspeeds in Section 2.6.4. The zero-incidence aerodynamic coefficients are computed and compared with PRODAS results in Section 2.6.5. Finally, results for the pitch-only configurations are mentioned in Section 2.6.6.

2.6.1 Overall Trends

The evolution of the projectile dynamics *w.r.t.* the trim incidence angles and the airspeed is illustrated in Figure 2.13, and the variations of the estimated parameters *w.r.t.* these variables are represented in Figure 2.14. For each considered airspeed, the model parameters are represented as a function of $\bar{\alpha}$ and $\bar{\beta}$. As the operating points grid is sparse, the estimated values are interpolated to form tri-dimensional surfaces. The color of these surfaces represents the trim airspeed, where dark blue corresponds to $V = 15$ m/s and red corresponds to $V = 40$ m/s. Several conclusions can be drawn from these overall results. The stability derivatives $M_{q\alpha}$ and $M_{r\beta}$ as well as the input coefficients $M_{q\delta_m}$ and $M_{r\delta_n}$ are highly speed-dependent. On the other hand, the damping coefficients M_{qq} and M_{rr} as well as the coupling coefficients $M_{q\beta}$ and $M_{r\alpha}$ exhibit a lower amount of dependence with respect to the airspeed, as the surfaces are much closer to each other or even superposed. An exception is the higher yaw damping coefficient for $V = 15$ m/s. Due to the size and weight of the yaw gimbal, friction in the yaw bearing induces additional damping. As the speed increases, the aerodynamic damping becomes more prevalent, reducing the impact of the bearing friction.

The shape of the constant-speed surfaces also suggests that the model parameters do not change significantly with respect to the chosen operating point, for a given value of the airspeed. Put in perspective with the above discussion on *fast* and *slow* scheduling parameters, this behavior is interesting for gain-scheduling control design. Indeed, while gain-scheduling on a *slow* parameter like the airspeed is relatively simple, scheduling on *fast* parameters like the incidence angles, which are also part of the linearized system state vector, introduces additional complex dynamics that must be considered carefully.

In the present case, the constant-speed projectile behavior may then be described with a single model, enabling a simple airspeed-based gain-scheduling control scheme. This is illustrated by the Bode diagrams in Figure 2.13, which show that the system bandwidth and damping ratio does not vary significantly for all operating points at a given airspeed, except for the yaw channel where the observed deviations correspond to operating points close to the flight envelope boundaries. The observation of the Bode diagrams also indicate that the system bandwidth increases with an increase in airspeed, while the damping factor decreases.

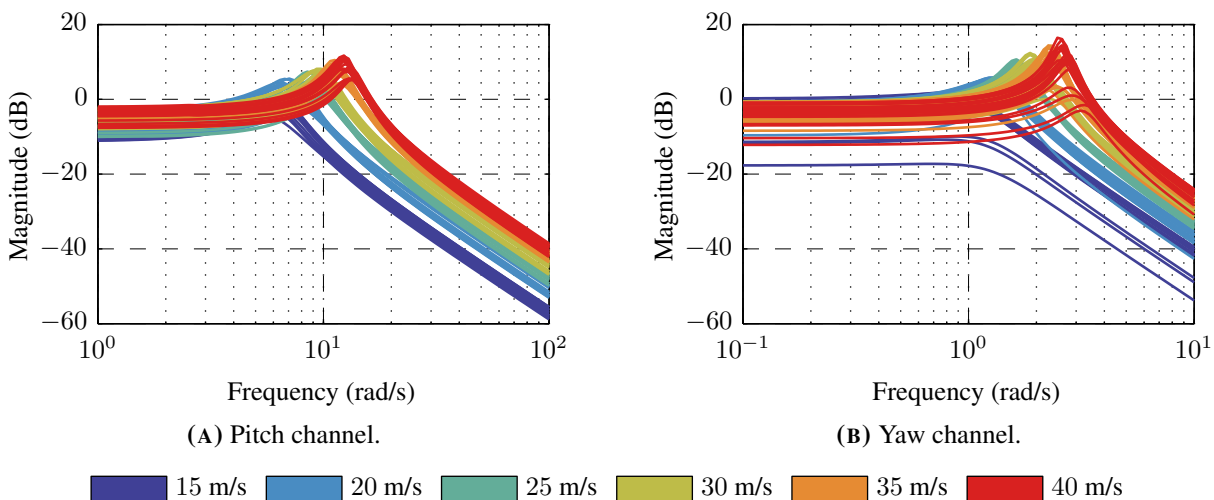


FIGURE 2.13: Open-loop plant Bode diagrams for all considered operating points.

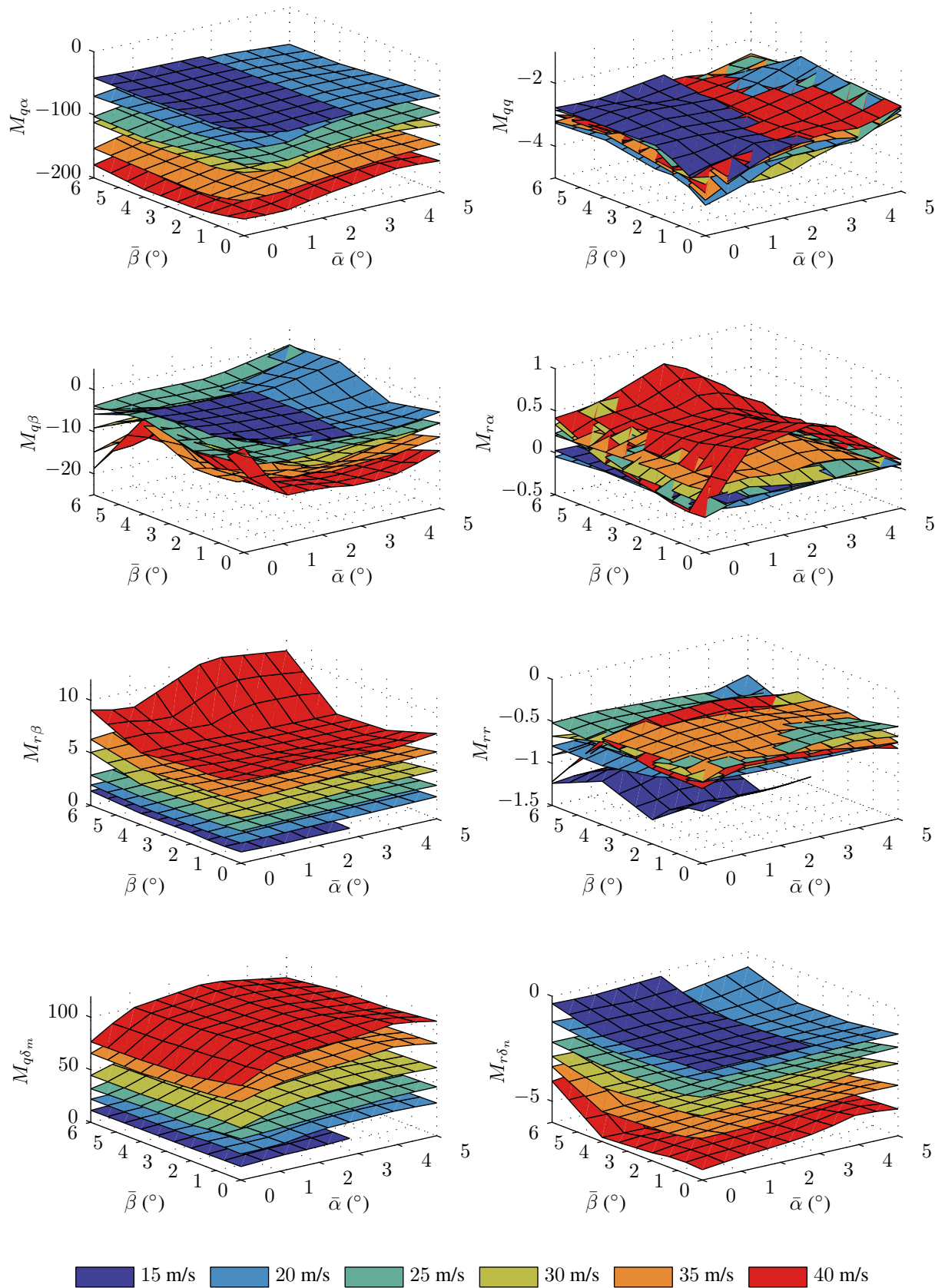


FIGURE 2.14: Estimated parameter surfaces as a function of the trim angles of incidence $\bar{\alpha}$ and $\bar{\beta}$. Each individual surface corresponds to a single trim airspeed \bar{V} .

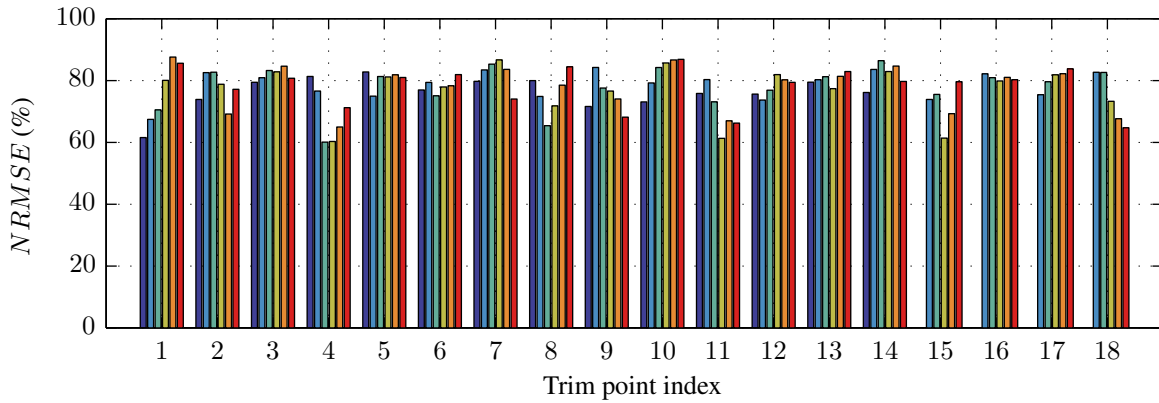
2.6.2 Model Fit Evaluation

Evaluating a model consists in assessing its ability to reproduce the measured output signals. This consists in comparing the measured outputs with simulated results obtained with the estimated parameters. This evaluation is thus carried out on an independent data set, which has not been used for parameter estimation. A suitable measurement of the model's ability to reproduce a given data set is the normalized root mean square error (NRMSE):

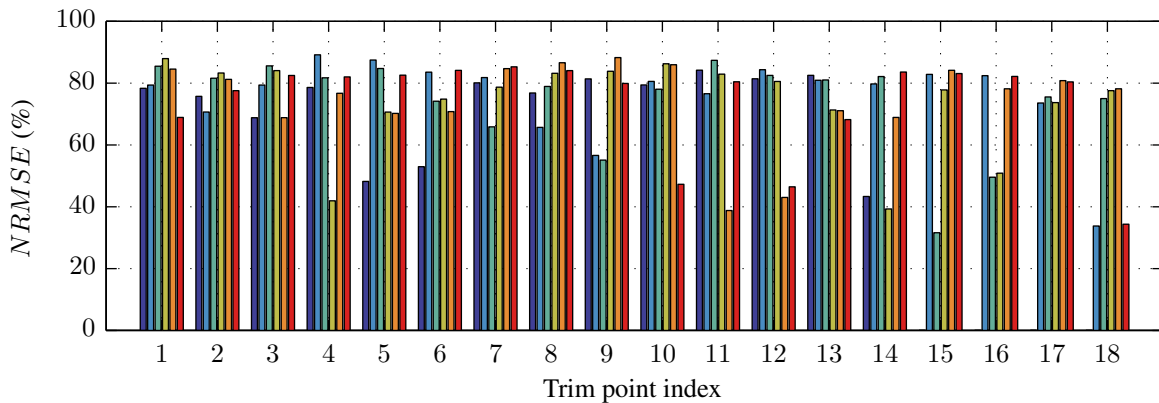
$$NRMSE = 1 - \frac{\|\mathbf{y}_m - \mathbf{y}_e\|_2}{\|\mathbf{y}_m - \bar{\mathbf{y}}_m\|_2} \quad (2.100)$$

where \mathbf{y}_m is the measured validation output vector, \mathbf{y}_e is the simulated output vector, $\bar{\mathbf{y}}_m$ is the mean of \mathbf{y}_m and $\|\cdot\|_2$ denotes the 2-norm. The NRMSE varies between $-\infty$ to 1, where 1 is a perfect fit.

The fit values have been computed for each operating point and are presented in Figure 2.15. The horizontal scale corresponds to the trim point index, where the corresponding trim incidence angles are given in Table 2.1. The fit results are grouped by the trim airspeed, with the leftmost bar corresponding to $V = 15$ m/s. The average fit values for the pitch and yaw channels are respectively 77.9% and 74.2%, which show that the models are able to simulate the system output around the considered operating



(A) Model fit for the pitch channel.



(B) Model fit for the yaw channel.



FIGURE 2.15: Model fit to the validation data, measured using the normalized root mean square error (NRMSE). The horizontal scale refers to the index of the trim point in Table 2.1.

points. The model fit degrades for extreme values of the incidence angles, at the edge and corners of the trim map, with the yaw channel being the most affected. This degradation of fit is due to unmodeled and nondeterministic perturbations of the output signal. As the projectile reaches the edge of the trim map, the efficiency of the canards is reduced and the projectile is more sensitive to aerodynamic disturbances, which induce pitching and/or oscillations superposed with the modeled input/output behavior.

2.6.3 Uncertainty Evaluation

Another indicator of model quality is the confidence interval of the estimated parameters and their relative uncertainty. These values, together with the estimated parameter values, are illustrated in Figures 2.16 and 2.17. In both figures, the horizontal scale also corresponds to the index of the trim points in Table 2.1. The stability coefficients $M_{q\alpha}$, $M_{r\beta}$ and the input coefficients $M_{q\delta_m}$, $M_{r\delta_n}$ exhibit the tightest confidence intervals and lowest relative uncertainty figures. These values are better for the pitch channel, with relative uncertainties under 5% for both coefficients. The yaw-related coefficients exhibit a higher uncertainty, especially for trim points close to the edge of the trim map and at low speeds. Under these conditions, the uncertainty can attain 30% and even more than 100% for the trim point ($\bar{V} = 15$ m/s, $\bar{\alpha} = 3^\circ$, $\bar{\beta} = 6^\circ$), which indicates this point may not be usable in practice. Indeed, at this point the yaw data shows severe asymmetry, indicating operation outside the nominal flight domain. At higher speeds and lower incidence angles, the uncertainty on the yaw static coefficients is under 15%. As predicted by the sensitivity analysis, the pitch and yaw damping coefficients M_{qq} and M_{rr} exhibit higher uncertainty figures than the static coefficients, respectively under 20% and 40% (excluding edge cases). Finally, the coupling coefficients $M_{q\beta}$ and $M_{r\alpha}$ show the highest uncertainty figures. This observation has two root causes. First, these coefficients were deemed non-identifiable by the *a posteriori* identifiability analysis, thus poor confidence in these results is expected. Secondly, these coefficients have values close to zero on some points, resulting in high relative uncertainty.

These observations suggest that the projectile's in-flight behavior is well-defined for most of the flight domain, and that the extreme-incidence points should be considered with caution as they result in high parameter uncertainty on the yaw axis. Secondly, the coupling coefficients may not be represented using relative (multiplicative) uncertainty, as their nominal values are often close to zero.

2.6.4 Constant-Airspeed Trends

Let us now discuss the parameter trends at a constant airspeed. The following discussion concerns the trim points for $V = 25$ m/s, for which the parameter surfaces are illustrated in Figure 2.18, but the results are mostly similar at other airspeeds, and differences will be discussed thereafter.

In Section 2.6.1, the overall observed trend is that the parameters are more dependent on the airspeed than on the incidence angles, as shown by the flatness of most of the parameter surfaces in Figure 2.14. Following the assumption made in this subsection, a nominal constant-air-speed model can be defined, with its parameters being the respective means of the model parameters at the considered airspeed. In this case, the incidence angle dependence of the model parameters is relatively moderate, since at $V = 25$ m/s the observed relative variation range for the direct channel parameters with respect to the nominal (mean) value is less than 30%. This relative variation is much higher for the coupling parameters, which have however little influence on the output as shown in Section 2.5.4. An additional observation is that parameters $M_{q\alpha}$, $M_{r\alpha}$, and $M_{q\delta_m}$, which are associated with the pitch-related states and input are more

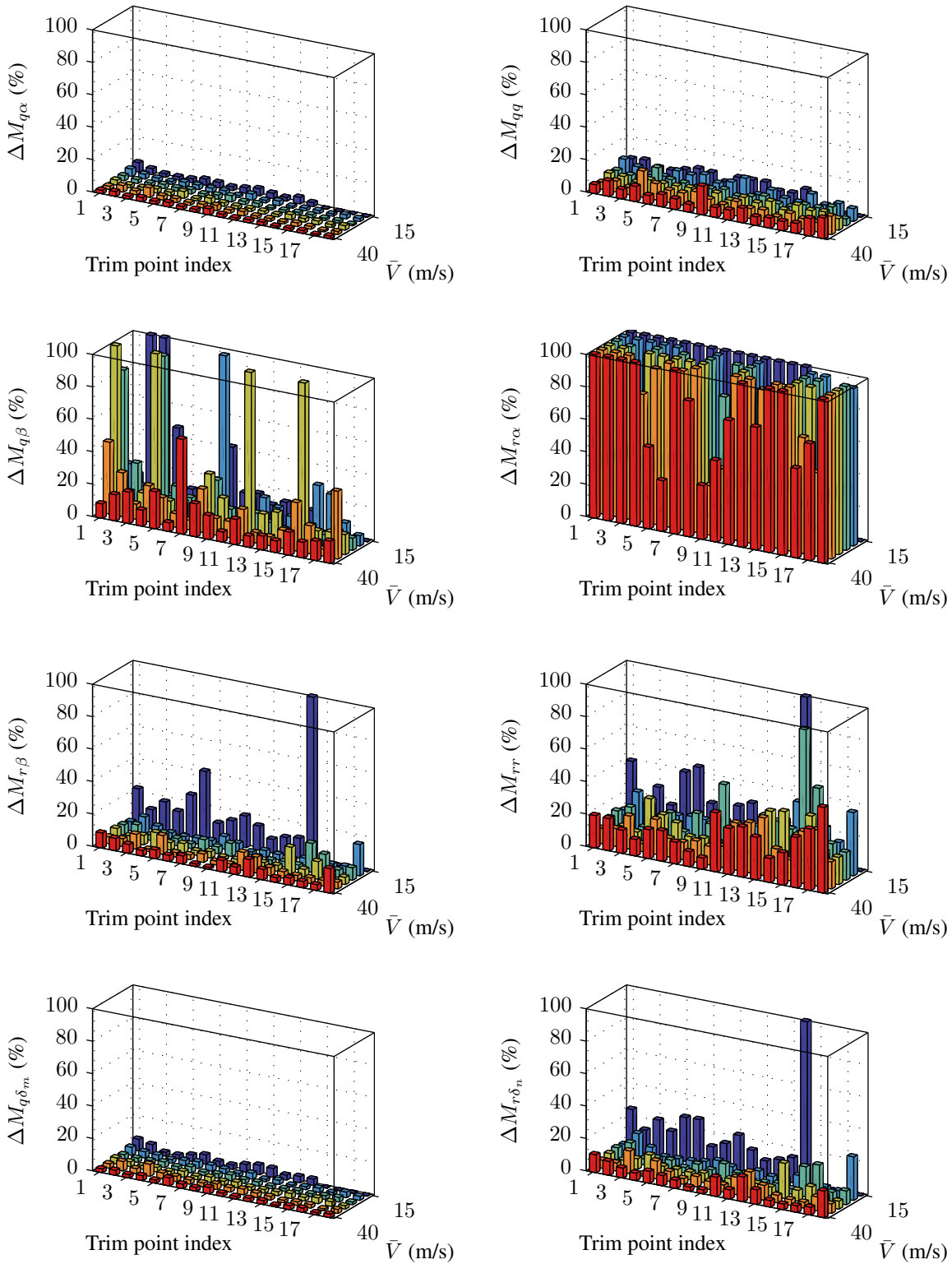


FIGURE 2.16: Relative uncertainty on the estimated parameters. The vertical scale has been clipped to 100% for readability.

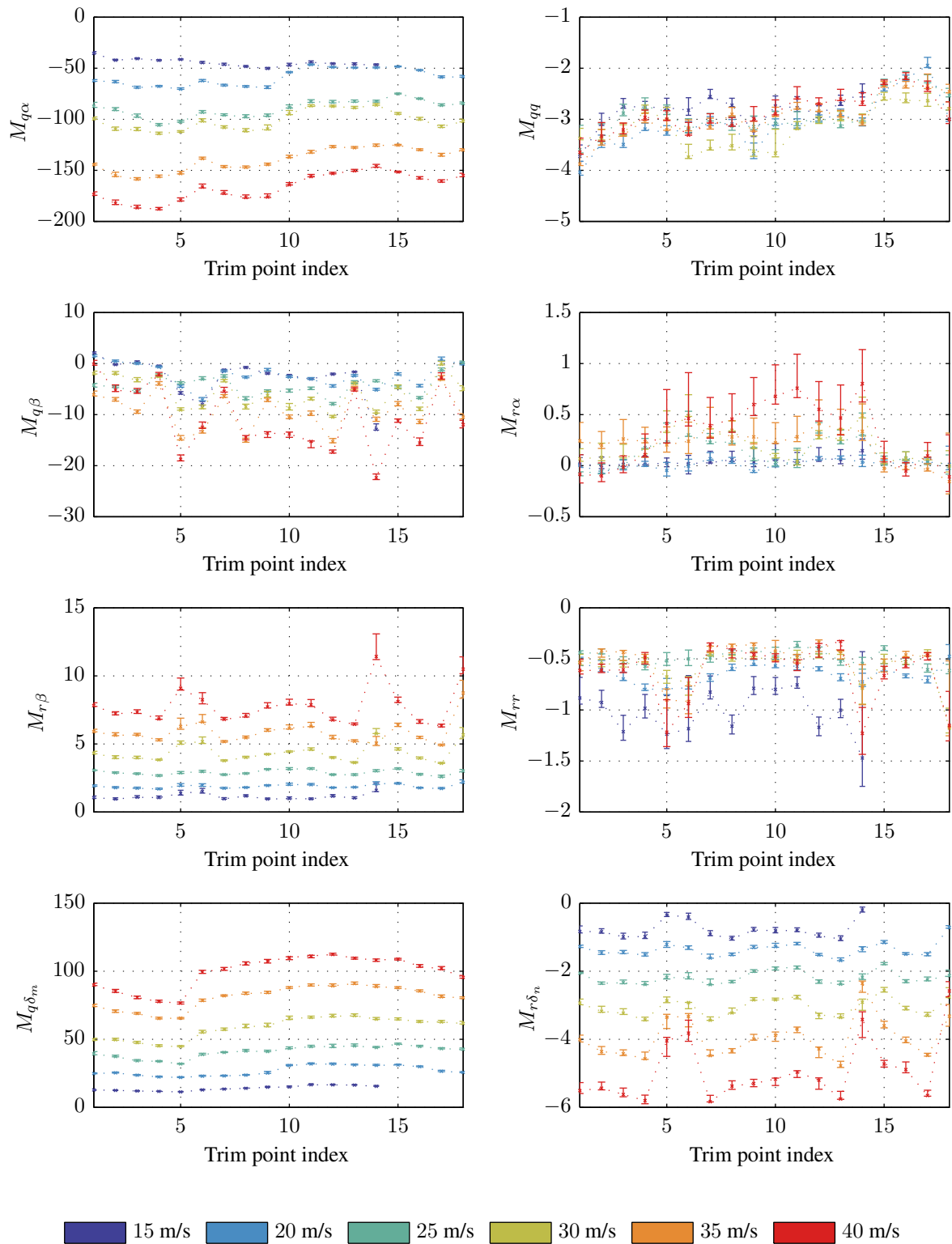


FIGURE 2.17: 95% Confidence intervals on the estimated parameters. The horizontal scale refers to the index of the trim point in Table 2.1.

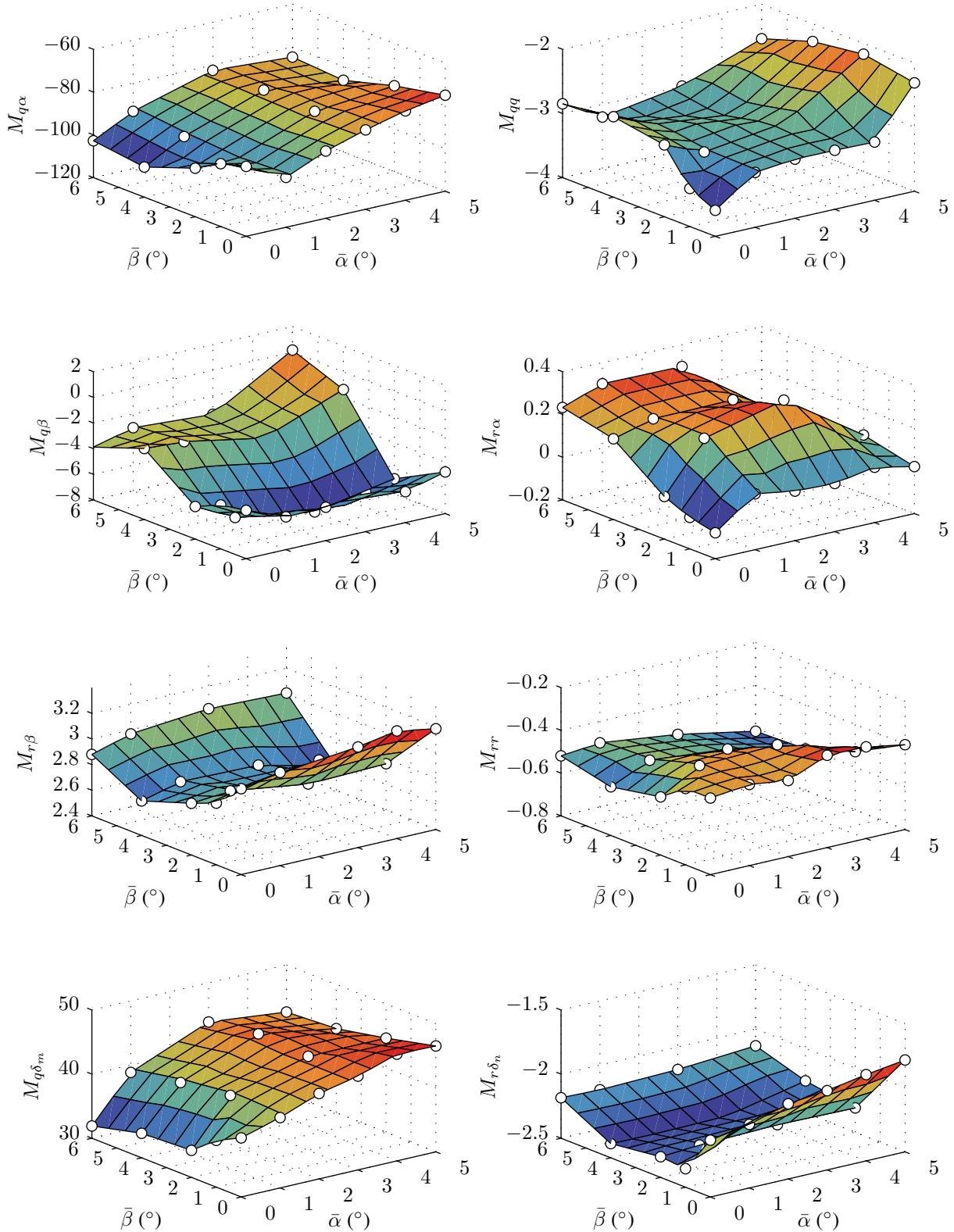


FIGURE 2.18: Estimated parameter surfaces for $V = 25$ m/s as functions of the trim angles of incidence $\bar{\alpha}$ and $\bar{\beta}$. White dots denote the actually sampled operating points.

dependent on the value of $\bar{\alpha}$ and do not vary significantly about the $\bar{\beta}$ axis. The same holds true for the yaw-dependent parameters $M_{q\beta}$, $M_{r\beta}$, and $M_{r\delta_n}$, which are more sensitive to $\bar{\beta}$ than $\bar{\alpha}$.

This discussion mostly holds for other airspeeds, except at the lowest setting ($V = 15$ m/s) and for the yaw damping coefficient at high speeds. At the lowest speed, the variation on the yaw channel input parameter $M_{r\delta_n}$ is over 75%. At this setting, the flight envelope is reduced and the observed effect is due to reduced canard efficiency at high angles of sideslip. At high speeds, the yaw damping coefficient M_{rr} variation attains 100% with respect to the nominal value along the flight envelope, with higher absolute values for high incidence angles.

2.6.5 Comparison with PRODAS-derived Aerodynamic Coefficients

In Section 2.4.2, the model parameters M_{ij} were defined as gradients of the nonlinear model state functions, and are related to the local derivatives of the aerodynamic coefficients by the expressions given in Equations (2.78)–(2.85). However, it must be remembered that the linear models only describe the local, tangent behavior around an equilibrium point and are not a direct representation of the nonlinear projectile behavior. The same goes for the model parameters, which cannot describe the overall shape of the aerodynamic coefficients functions. Nevertheless, at the zero-incidence trim points, the model parameters are effectively related to the actual values of the aerodynamic coefficients since the unknown trim-related terms cancel themselves. As the mass properties of the projectile have been measured, it is then possible to compare these aerodynamic coefficients obtained through parameter estimation against values obtained with other methods.

The determination of aerodynamic coefficients using wind tunnel tests and Computational Fluid Dynamics software (CFD) is particularly time-consuming, especially when dynamic coefficients are required, such as the damping coefficient C_{mq} . Conversely, semi-empirical codes such as PRODAS can provide with an answer in reasonable time, but the accuracy of these results is highly dependent on the quality and thoroughness of the databases used for interpolation. In particular, these tools are mainly suited for spinning supersonic projectiles and results may not be precise for finned projectiles, especially at low speeds. An additional assumption in these tools is that the coefficients are almost constant in the low subsonic regime ($\mathcal{M} < 0.4$), also called the incompressible regime.

Aerodynamic coefficients concerning the pitch axis static and dynamic stability were obtained using a model of the ACHILES projectile in PRODAS. For low Mach numbers, the software only provides values for $\mathcal{M} = 0.01$ and $\mathcal{M} = 0.4$. These values have been represented together with the coefficients derived from the estimated parameters in Figure 2.19. These figures show an excellent agreement between these two methods for the static stability derivative $C_{m\alpha}$, and a larger difference for the dynamic stability derivative C_{mq} . Nonetheless, both methods produces values of C_{mq} of about the same order of magnitude. The difference between these two methods originates from several sources. First, the PRODAS coefficients have an expected error of about 25% in the transonic and supersonic regimes. As this tool is based on databases of wind tunnel and free-flight results, the accuracy of these databases at low Mach number may also be questioned. Moreover, the ACHILES fin and nose geometry is not common along classical projectiles, and thus this configuration may not be well covered by PRODAS. Finally, differences in the estimation of C_{mq} are to be expected, as this coefficient is notoriously difficult to measure or estimate, and comparisons in the literature also show high dispersions between experimental results and semi-empirical aeroprediction codes [Albisser, 2015; Bhagwandin and Jubaraj, 2014].

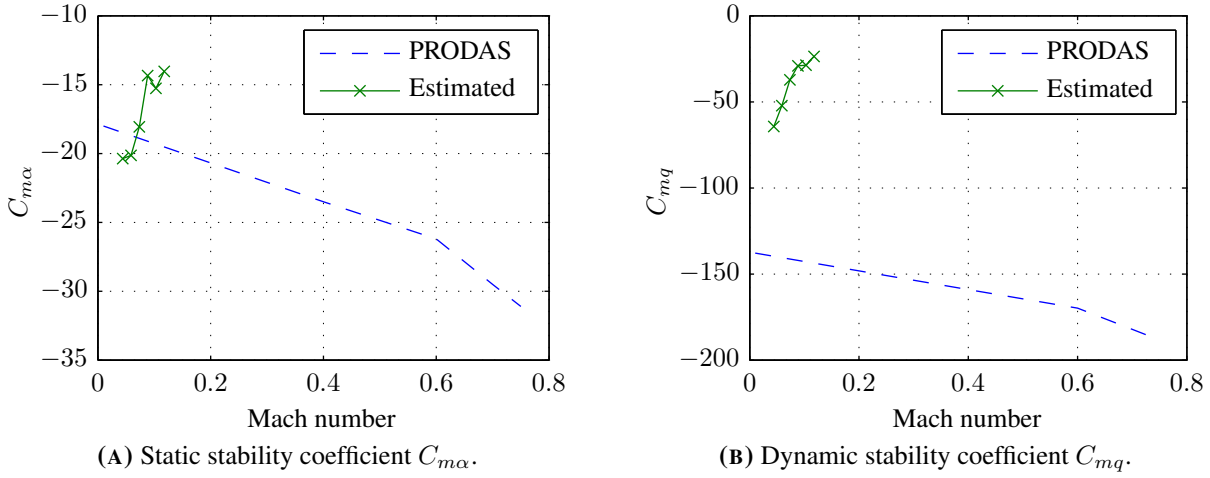


FIGURE 2.19: Comparison of aerodynamic stability derivatives obtained using PRODAS and from the estimated parameters.

2.6.6 Pitch-only case

The pitch-only case is a particular case of the two-degrees of freedom discussed in this chapter. As the yaw axis is locked, $\beta = 0$ and $r = 0$ and thus the fourth-order linear model of Equation (2.86) reduces to the following second-order single-input, two-outputs model:

$$\mathbf{G}_b : \begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ M_{q\alpha} & M_{qq} & M_{q\delta_m} \\ \hline \mathbb{I}_2 & & \mathbb{O}_{2 \times 1} \end{array} \right] \begin{bmatrix} \alpha \\ q \\ \delta_{m,c} \end{bmatrix} \quad (2.101)$$

where $\mathbf{x} = \mathbf{y} = [\alpha \quad q]^\top$ is the state and output vector, and $u = \delta_{m,c}$ is the input signal.

This simple model has served as the first step in developing the identification and control procedures presented in this work [Strub et al., 2014, 2016]. These results have been exploited in the development of a pitch-only autopilot, detailed in Appendix B and [Strub et al., 2015a,b].

As this simpler model only involves three parameters ($M_{q\alpha}$, M_{qq} , $M_{q\delta_m}$), it is not necessary to present the identification procedure again. The identification procedure presented in Section 2.5 indeed also applies to this reduced-order case and results concerning the system *a priori* and *a posteriori* identifiability are similar. In particular, the three parameters keep the same relative importance order as in the two-degrees of freedom, and there are no parameter collinearity issues.

The chosen aerodynamic configuration is different than in the two-degrees of freedom case, with the canards at a 45° angle and the tail fins lying in the horizontal and vertical planes. This configuration indeed extends the flight domain for a single degree of freedom, and roughly corresponds to a diagonal of the two-degrees of freedom trim map. The corresponding trim map is presented in Figure 2.20 and exhibits a larger flight domain with a sharp boundary. In this configuration, the accessible flight domain extends from $\bar{\alpha} = -8^\circ$ to $\bar{\alpha} = 8^\circ$, for fin deflections $\bar{\delta}_m$ in the $[-12^\circ, 12^\circ]$ range. For higher angles of attack, the efficiency of the canards rapidly degrades and the projectile enters a stalling behavior.

Model parameters have been estimated for values of the trim angle of attack $\bar{\alpha}$ from 0° to 7° in 1° steps, at a constant airspeed $V = 25$ m/s. As this was the first step in a bottom-up design approach for

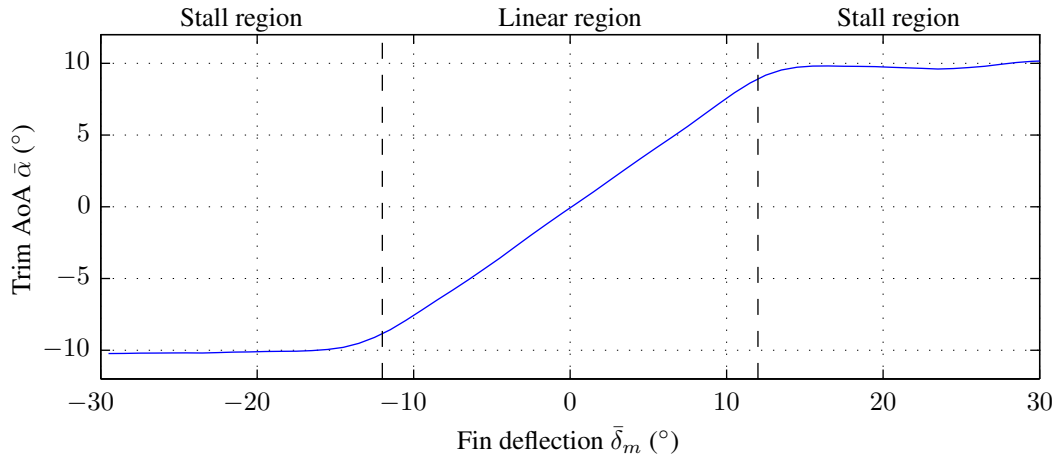


FIGURE 2.20: Projectile pitch-only trim map at $V = 25$ m/s.

the identification procedure, other values of the airspeed were not considered at the time. The estimated parameter values and relative uncertainties are summarized in Table 2.2. These parameters show similar trends to the two-axes case, with only a moderate variation of the values with respect to the trim angle of attack. Similarly, the uncertainty figures for the static parameters $M_{q\alpha}$, $M_{q\delta_m}$ are lower than for the damping parameter M_{qq} , and have a similar range (respectively under 10% and 20%).

TABLE 2.2: Estimated parameters and associated uncertainties for values of $\bar{\alpha}$ from 0° to 7° in 1° steps.

$\bar{\alpha}$	$M_{q\alpha}$	$\Delta M_{q\alpha}$	M_{qq}	ΔM_{qq}	$M_{q\delta}$	$\Delta M_{q\delta}$	Fit
0°	-50.3	6%	-2.9	15%	38.8	6%	74%
1°	-53.0	9%	-2.7	20%	34.3	10%	75%
2°	-54.6	7%	-2.8	16%	36.2	7%	82%
3°	-56.1	2%	-2.6	8%	40.8	3%	86%
4°	-52.5	4%	-2.4	12%	40.5	4%	89%
5°	-51.5	5%	-2.2	15%	40.3	5%	89%
6°	-52.7	3%	-2.1	17%	41.5	4%	89%
7°	-54.4	3%	-2.0	8%	38.5	3%	86%

2.7 Conclusion

In this chapter, the complete modeling process for the ACHILES projectile prototype has been carried out, from the derivation of the nonlinear differential equations to estimation of the model parameters through system identification. Before taking into account the specificities of the ACHILES setup, the flight mechanics of a generic free-flying vehicle have been presented in order to introduce the notations and modeling of the aerodynamic forces and moments. This model also serves as a comparison point to highlight the differences with the ACHILES projectile models.

The ACHILES projectile has been modeled using the same framework as in the free-flying case, taking into account the mechanical restraints imposed by the support structure. As a result, the translational equations of motion vanish and only the angular equations of motion remain. Since the support structure introduces a non-negligible inertia component on an intermediary axis, the equations of motion are much more complicated, despite similarities in form. Two additional cases were considered and

modeled, where first the roll axis is locked at a zero roll angle and secondly the yaw axis is also locked to zero, resulting in respectively two- and one-degree of freedom models.

All these models are highly nonlinear, due to the involved trigonometric functions and to the unknown formulation of the aerodynamic coefficients. In order to perform control design using modern robust control techniques targeted to linear systems, these nonlinear models are approximated using families of LTI models defined around the equilibrium points of the system. The projectile equilibrium manifold has been assessed for airspeeds between 15 m/s and 40 m/s, which correspond to the usable range of the wind tunnel. Results show that the usable flight domain remains constant across the airspeed range, except at low speeds which conducts to a smaller flight envelope.

The second part of the chapter presented the system identification procedure and parameter estimation results for the linear model describing the projectile pitch and yaw behavior. A standard identification procedure was applied, in which the identifiability of the model structure and of the individual parameters was assessed. Results indicate that the cross-axes coupling parameters are not *a posteriori* identifiable and that the estimation may not be precise for these parameters.

The model parameters were estimated for 18 angular trim points at 6 different airspeeds, resulting in a total of 103 considered operating points (several points were removed at low speeds due to the reduction in the flight envelope). These results show a strong dependence of several coefficients ($M_{q\alpha}$, $M_{q\delta_m}$, $M_{r\beta}$, $M_{r\delta_n}$) with respect to the airspeed, while damping M_{qq} , M_{rr} and coupling coefficients $M_{q\beta}$, $M_{r\alpha}$ are less sensitive to the airspeed. The parameter estimation results show good confidence in the estimated parameter values, except for the coupling coefficients and extreme values of the incidence angles. At these conditions, the projectile behavior is severely affected and the linearity hypotheses required for the considered system identification method are no longer respected. Except for the coupling coefficients, the model parameters also exhibit a moderate relative variation across the flight envelope for a constant airspeed, which is essential for the development of an airspeed-based gain-scheduled control law. As a result, the projectile behavior at a constant airspeed can be approximated by a single nominal model, and deviations from this model can be taken into account as parametric uncertainty.

Chapter 3

Control Design

Contents

3.1	Introduction	82
3.2	Design and Analysis Tools	84
3.2.1	Nominal Model Selection	84
3.2.2	Uncertainty Modeling	85
3.2.3	Standard \mathcal{H}_∞ Control	87
3.2.4	Multi-objective \mathcal{H}_∞ Control	89
3.2.5	Robustness Analysis	91
3.3	Fixed-Airspeed Autopilot	94
3.3.1	Synthesis Model	94
3.3.2	Autopilot Design	100
3.3.3	Robustness Analysis	108
3.3.4	Experimental Validation	110
3.3.5	Concluding Remarks	116
3.4	Variable-Airspeed Autopilot	117
3.4.1	Gain-Scheduling Strategy	117
3.4.2	Local Controller Synthesis	120
3.4.3	Controller Implementation and Experimental Validation	131
3.4.4	Concluding Remarks	134
3.5	Conclusion	134

3.1 Introduction

This chapter focuses on autopilot design for the ACHILES projectile prototype, leveraging the model structures and parameter estimates presented in Chapter 2. The autopilot lies at the core of the Guidance, Navigation and Control (GNC) loop, in which it is responsible for the "Control" function. The main objective of any control system is to drive the plant input such that a desired output behavior is obtained, while ensuring the stability of the plant. Depending on the plant and application, the desired behavior may consist in *disturbance rejection* and/or *reference tracking*. *Disturbance rejection* corresponds to the *regulator* problem, which consists in attenuating the effects of external disturbances on the plant input and output. *Reference tracking* corresponds to the *servo* problem, in which the plant output must follow an external reference signal. Both objectives are characterized by performance specifications, such as the response time, overshoot level, or noise/disturbance attenuation level.

In the general case, the exact plant behavior may not be known, either due to various sources of uncertainties, or by deliberately neglecting some aspects so as to simplify the model. In order to deal with these inaccuracies, the plant may be modeled as a *nominal* part associated to an *uncertainty* description. This *uncertain* model aims at capturing all the possible realizations of the (perturbed) plant. Under this framework, the *stability* and *performance* objectives of the control system can be classified depending on whether the uncertainty is taken into account for evaluation or not [Skogestad and Postlethwaite, 2007]: *nominal stability* (NS) and *nominal performance* (NP) are evaluated with no model uncertainty, while *robust stability* (RS) and *robust performance* (RP) are evaluated for any perturbed realization of the plant, up to the worst-case uncertainty.

In the aerospace field, the design of an autopilot satisfying the nominal and robust objectives is a complex task. On the one hand, aerospace vehicles have typically highly nonlinear dynamics with time- and parameter-dependent behavior. On the other hand, the designer is faced with multiple limitations, requiring a careful trade-off between performance and robustness. Limitations and constraints include limited actuator bandwidth and saturation-free operation range, unmodeled and neglected dynamics in the system, parameter uncertainty, delays, nonlinearities, sensor noise and external disturbances.

Aerospace applications have been major users and drivers of research on automatic control [Åström and Kumar, 2014], from the first airplane autopilots to the space race, ballistic missile systems and fighter jets. Even though classical control methods based on successive closures of SISO loops have been successfully employed for decades, the challenges posed by modern high-performance aircraft configurations, as well as safety requirements, call for multivariable robust control methods [Bates and Postlethwaite, 2002] such as \mathcal{H}_∞ or \mathcal{H}_2 [Doyle et al., 1989]. These methods are able to treat the above limitations efficiently, yet in a practical manner for the control designer, and have been applied successfully to a wide variety of applications, including missile autopilot design [Apkarian et al., 1995; Nichols et al., 1993].

In its classical formulation, the \mathcal{H}_∞ synthesis method consists of a design step followed by an analysis step. The actual controller synthesis is performed in the design step, by shaping several closed-loop transfer functions (mixed-sensitivity or signal-based \mathcal{H}_∞ synthesis). At this step, the \mathcal{H}_∞ controller ensures the nominal stability and performance objectives. The robust stability and performance objectives are assessed *a posteriori* in the analysis step, using an uncertain model of the plant that takes into account the various sources of uncertainties in the system. Another technique called \mathcal{H}_∞ loop-shaping consists

in shaping the open-loop transfer functions to satisfy performance requirements, then robustifying the augmented plant to coprime factor uncertainty [McFarlane and Glover, 1992]. Finally, while techniques such as μ -synthesis are able to tackle all four objectives at once, they rely on tight bounds of the structured singular value μ which are notoriously difficult to compute and may yield high-order controllers, hence they are seldom used in the industry [Bates and Postlethwaite, 2002].

Control design for the ACHILES projectile prototype provides with some interesting challenges. While the operating envelope is not as wide as for a free-flying projectile, the ACHILES dynamics are still nonlinear and parameter-dependent. Additionally, the experimental setup is subject to different dynamics on the pitch and yaw axes as well as flexible modes due to the large size of the support structure. Furthermore, the autopilot is to be implemented in a Hardware-In-the-Loop scheme with a main clock rate of 100 Hz and a one-cycle controller delay due to software considerations, as presented in Section 1.4.1. All these constraints, along with modeling uncertainty, make the autopilot design a complex task and justify the need for \mathcal{H}_∞ control methods.

The main difference in autopilot design between the ACHILES projectile and a free-flying projectile is on the controlled variables. In the present case, the autopilot reference tracking acts on the projectile angles of incidence (the angle of attack α and the angle of sideslip β), which are directly related to the Euler angles measured by the IMU. In the free-flying case, the tracked outputs are the projectile accelerations, as in most cases the incidence angles cannot be measured directly. In the present monograph, only the first three autopilot objectives (namely nominal stability and performance, and robust stability) will be discussed since the robust performance objective is not as critical in the present case. The autopilot performance requirements consist in both disturbance rejection and reference tracking, the latter being specified by the means of a reference model exhibiting the desired behavior.

An incremental approach was followed for the design of the ACHILES autopilot, with increasing complexity and increasing dimension of the allowable flight domain at each step. Three cases have been considered:

- A pitch motion control at a constant airspeed of 25 m/s. In this case, the flight envelope is reduced to a single direction, which results in a SIMO control problem.
- A combined pitch and yaw motion control at a constant airspeed of 25 m/s. Naturally, the flight domain is extended to a second dimension and coupling effects have to be considered between both axes.
- Pitch and yaw motion control with a variable airspeed, allowing the full identified flight envelope to be exploited.

For every case, the full controller synthesis procedure is detailed and the robustness of the closed-loop system is studied. Results are discussed from both a simulation and an experimental point of view. Nevertheless, to shorten the presentation, only the combined pitch and yaw motion control at constant and variable airspeeds will be described in the following. The pitch-only case has been discussed in [Strub et al., 2015a,b] and is presented in Appendix B. Before treating these application cases, this chapter begins with some theoretical considerations.

3.2 Design and Analysis Tools

This section aims at introducing the design and analysis tools that will be used throughout this chapter. The first step in applying any model-based control technique such as \mathcal{H}_∞ consists in modeling the plant behavior. Recalling the control system objectives, namely nominal/robust stability and performance, two models are actually required: a nominal model for control design, and an uncertain model taking into account the various sources of uncertainty in the plant for robustness analysis. Determining these models is straightforward when an analytical model is available and the plant is linear time-invariant (LTI).

For systems with moderate nonlinearities (e.g the fixed-airspeed ACHILES case), instead of building a nonlinear controller, a sensible design approach is to construct a single robust controller. To this end, the plant behavior at any operating point may be approximated using a single linear nominal model, and plant variations due to the nonlinearities may be taken into account as uncertain dynamics. In the ACHILES case, the analytical model cannot be used directly for control due to the aerodynamic coefficients being modeled as unknown functions. Instead, the airframe behavior is approximated using families of linear models which have been estimated from experimental data. Sections 3.2.1 and 3.2.2 provide tools for respectively constructing nominal and uncertain models of the airframe from these results. The standard \mathcal{H}_∞ design problem and associated synthesis techniques are introduced in Section 3.2.3. However, practical applications may include somewhat conflicting specifications, in which case standard \mathcal{H}_∞ synthesis may not offer sufficient performance, hence multi-objective design techniques suitable to these design problems are presented in Section 3.2.4. Finally, classical and modern robustness analysis tools are briefly introduced in Section 3.2.5.

3.2.1 Nominal Model Selection

Considering a family of estimated linear models issued from a nonlinear plant, a moderate nonlinearity should result in similar behavior across the members of the family. As a nominal model should be representative of the average behavior of the plant, it should then exhibit the lowest possible difference in behavior *w.r.t.* all members of the family. A good metric for assessing this difference is the ν -gap metric δ_ν [Vinnicombe, 2001; Zhou and Doyle, 1998], which describes the difference in closed-loop performance between two plants under the same controller. The ν -gap $\delta_\nu(\mathbf{P}_1, \mathbf{P}_2)$, for two plants of the same size, is defined as:

$$\delta_\nu(\mathbf{P}_1, \mathbf{P}_2) = \begin{cases} \left\| \begin{bmatrix} -\tilde{\mathbf{M}}_2 & \tilde{\mathbf{N}}_2 \end{bmatrix} \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{M}_1 \end{bmatrix} \right\|_\infty & \text{if } \det\left(\begin{bmatrix} \mathbf{N}_2 \\ \mathbf{M}_2 \end{bmatrix}^\top \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{M}_1 \end{bmatrix}\right)(j\omega) \neq 0, \forall \omega \in (-\infty, \infty) \\ & \text{and } \text{wno}\left(\det\left(\begin{bmatrix} \mathbf{N}_2 \\ \mathbf{M}_2 \end{bmatrix}^\top \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{M}_1 \end{bmatrix}\right)\right) = 0 \\ 1 & \text{otherwise} \end{cases} \quad (3.1)$$

where $\mathbf{P}_i = \mathbf{N}_i \mathbf{M}_i^{-1} = \tilde{\mathbf{M}}_i^{-1} \tilde{\mathbf{N}}_i$ are normalized right (resp. left) coprime factorizations of the plants, and wno denotes the winding number²⁰. The distance $\delta_\nu(\mathbf{P}_1, \mathbf{P}_2)$ is then small when the two plants \mathbf{P}_1

²⁰The winding number of a function $g(s)$ is the the number of encirclements of the origin made by $g(s)$ as s follows any closed path, which is equal to the numbers of zeros of $g(s)$ enclosed by the path minus the number of poles of $g(s)$ enclosed by the path [Vinnicombe, 2001]

and \mathbf{P}_2 exhibit a similar behavior with the same feedback controller, and tends towards unity when the two systems exhibit large differences in behavior [Cantoni and Vinnicombe, 1999].

Using this metric, a nominal model selection algorithm minimizing the largest ν -gap distance between the nominal model and the family members can be constructed using the following criterion:

$$J(\mathbf{P}) = \max_{\mathbf{P}_i \in \Pi} \delta_\nu(\mathbf{P}, \mathbf{P}_i) \quad (3.2)$$

where \mathbf{P} is the nominal model and $\mathbf{P}_i \in \Pi$ the family of estimated models.

Finding the optimal nominal model \mathbf{P} is a complex, non-convex optimization problem and hence two more computationally tractable strategies are considered. In the first strategy, the nominal model has the same structure as the family members and each parameter is the mean of the corresponding estimated parameters. The second strategy consists in selecting the nominal model as the member of the family of estimated models which exhibits the lowest maximum ν -gap distance *w.r.t.* all other models in the family:

$$\mathbf{P} = \left\{ \mathbf{P}_k \in \Pi \mid J(\mathbf{P}_k) = \min_{\mathbf{P}_j \in \Pi} J(\mathbf{P}_j) \right\} \quad (3.3)$$

3.2.2 Uncertainty Modeling

Uncertain models are used for analyzing the robustness of control laws and aim at capturing the various sources of plant model uncertainty, which can be classified in two classes [Skogestad and Postlethwaite, 2007]: dynamic (or *unstructured*) uncertainty and parametric (or *structured*) uncertainty.

3.2.2.1 Dynamic Uncertainty

Dynamic uncertainty takes into account unmodeled dynamics, such as the high-frequency behavior of the plant, and high-order dynamics (such as flexible modes) which may be neglected in order to carry out the controller synthesis on simpler models. This class of uncertainty considers a full (*unstructured*) complex perturbation matrix Δ scaled by frequency-dependent weights. The Δ matrix is any stable transfer function matrix satisfying $\|\Delta\|_\infty \leq 1$, and has usually compatible dimensions with those of the plant. There are several common representations for this class of uncertainty [Skogestad and Postlethwaite, 2007], one of the preferred form being *multiplicative (input) uncertainty*, which is illustrated in Figure 3.1 and contains perturbed models $\tilde{\mathbf{P}}$ of the form:

$$\Pi_I : \quad \tilde{\mathbf{P}} = \mathbf{P} (\mathbb{I} + w_I \Delta_I) \quad (3.4)$$

In the above equation, Δ_I is any stable transfer function matrix satisfying $\|\Delta_I\|_\infty \leq 1$. The multiplicative weight w_I bounds the relative difference in frequency response between the nominal model and

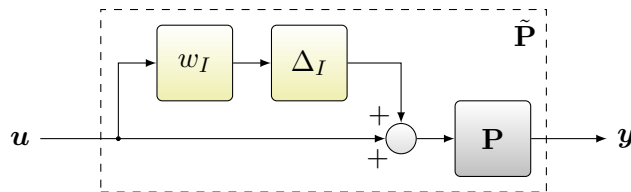


FIGURE 3.1: Multiplicative uncertainty at the plant input.

uncertain systems in the perturbed set Π . It is usually chosen as a stable, minimum-phase, scalar-valued transfer function satisfying $|w_I(j\omega)| \geq \ell_I(j\omega), \forall \omega$. The envelope of relative differences is characterized by the uncertainty radius ℓ_I , defined as:

$$\ell_I(\omega) = \max_{\tilde{\mathbf{P}} \in \Pi} \bar{\sigma} \left[\mathbf{P}^{-1} \left(\tilde{\mathbf{P}} - \mathbf{P} \right) (j\omega) \right], \forall \omega \quad (3.5)$$

This class of uncertainty has been used in the pitch-only case (see Appendix B) to model the open-loop dynamics variations across the flight domain. In this case, the uncertainty radius corresponds to the maximum difference between the nominal model \mathbf{P} and the set of experimentally estimated models, including the bootstrapped models.

3.2.2.2 Parametric Uncertainty

Parametric uncertainty considers model parameters which are known only approximately, or are subject to variations induced by nonlinearities and/or operating point variations. In this representation, each (real-valued) uncertain parameter $\tilde{\theta}_i$, with nominal value $\bar{\theta}_i$ and variation range $[\theta_{i,\min}, \theta_{i,\max}]$ is modeled as:

$$\tilde{\theta}_i = \bar{\theta}_i (1 + r_i \delta_i) \quad (3.6)$$

where δ_i is any real number satisfying $|\delta_i| \leq 1$ and r_i is the relative uncertainty defined as

$$r_i = \frac{\theta_{i,\max} - \theta_{i,\min}}{\theta_{i,\max} + \theta_{i,\min}} \quad (3.7)$$

This uncertainty representation can be employed to describe uncertain gains, poles or zeros in a transfer function, or uncertain state-space system matrices. These descriptions usually involve multiple perturbations δ_i , which may be separated from the transfer function or system matrices and collected in a larger, diagonal²¹ Δ perturbation matrix with $\Delta = \text{diag} \{ \delta_i \}$. Depending on the number of occurrences of an uncertain parameter θ_i in the model, the associated perturbation δ_i may be repeated in the Δ matrix.

To carry out this separation, the uncertain model may be written using linear fractional transformations [Hecker and Varga, 2004; Lambrechts et al., 1993]. The corresponding upper LFT is illustrated in Figure 3.2, and is given by:

$$\tilde{\mathbf{P}} = \mathcal{F}_u(\mathbf{P}^\Delta, \Delta) \quad (3.8)$$

where \mathbf{P}^Δ contains to the nominal model \mathbf{P} , augmented with the relative uncertainties r_i and interconnections $\mathbf{u}_\Delta, \mathbf{y}_\Delta$ with the Δ matrix.

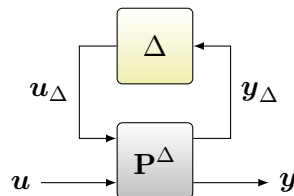


FIGURE 3.2: Uncertain plant represented using a Linear Fractional Transformation.

²¹In this case, the perturbation matrix is said to be *structured*, as opposed to *unstructured* uncertainty where Δ is full.

Both dynamic and parametric uncertainty representations may be combined in a same uncertain model, in which case the Δ matrix has a block-diagonal structure. In practice, evaluating the robust stability condition is more complex for structured uncertainty but this representation gives the least conservative results. However, for SISO systems parametric uncertainty may be approximated with a single lumped dynamic uncertainty since it often reduces conservativeness [Skogestad and Postlethwaite, 2007]. Moreover, the associated robust stability condition is much more computationally tractable.

3.2.3 Standard \mathcal{H}_∞ Control

The \mathcal{H}_∞ control problem consists in designing a controller ensuring closed-loop internal stability and a desired exogenous input to exogenous output frequency behavior. The exogenous inputs may contain reference signals and external disturbances. \mathcal{H}_∞ synthesis leads to controllers which attenuate the impact of these exogenous inputs on the exogenous outputs. If model uncertainties are also taken into account in the problem formulation, then the obtained \mathcal{H}_∞ controller is also robust *w.r.t.* these uncertainties.

3.2.3.1 Standard form

The \mathcal{H}_∞ control framework considers the standard configuration illustrated in Figure 3.3.

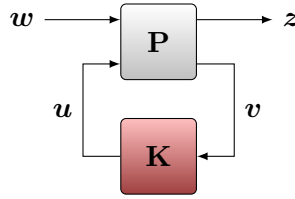


FIGURE 3.3: Standard \mathcal{H}_∞ problem.

In this representation, \mathbf{K} is the controller and \mathbf{P} is the *augmented plant*, containing the plant to be controlled \mathbf{G} as well as the design specifications in the form of weighting filters. The inputs of \mathbf{P} are the exogenous inputs \mathbf{w} , such as reference signals and disturbances, and the control inputs \mathbf{u} . Its outputs are the performance (or exogenous) outputs \mathbf{z} , which are to be minimized, and the measured variables \mathbf{v} . The relations governing the standard \mathcal{H}_∞ problem are:

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{v} \end{bmatrix} = \mathbf{P}(s) \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11}(s) & \mathbf{P}_{12}(s) \\ \mathbf{P}_{21}(s) & \mathbf{P}_{22}(s) \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} \quad (3.9)$$

$$\mathbf{u} = \mathbf{K}(s)\mathbf{v} \quad (3.10)$$

with a state-space realization of the augmented plant \mathbf{P} given by:

$$\mathbf{P}^{\text{ss}} \doteq \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{array} \right] \quad (3.11)$$

From Equations (3.9) and (3.10), the closed-loop transfer function from \mathbf{w} to \mathbf{z} can be obtained using the lower linear fractional transformation (LFT):

$$\mathbf{T}_{\mathbf{w} \rightarrow \mathbf{z}} = \mathcal{F}_l(\mathbf{P}, \mathbf{K}) \quad (3.12)$$

$$= \mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(\mathbb{I} - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21} \quad (3.13)$$

3.2.3.2 \mathcal{H}_∞ Problem Resolution

The \mathcal{H}_∞ control problem consists in finding a controller \mathbf{K} which minimizes, in the sense of the \mathcal{H}_∞ norm, the influence of exogenous inputs on performance outputs. Mathematically, the optimal \mathcal{H}_∞ problem is formulated as follows: find all stabilizing controllers \mathbf{K} which minimize [Duc, 2000]:

$$\|\mathcal{F}_l(\mathbf{P}, \mathbf{K})\|_\infty = \max_{\omega} \bar{\sigma}(\mathcal{F}_l(\mathbf{P}, \mathbf{K})(j\omega)) \quad (3.14)$$

The \mathcal{H}_∞ norm has several interpretations in terms of performance: in the above equation, it minimizes the peak maximum singular value of $\mathcal{F}_l(\mathbf{P}, \mathbf{K})$. The \mathcal{H}_∞ norm can also be interpreted in the time-domain as the induced \mathcal{L}_2 norm on signals. Let $z = \mathcal{F}_l(\mathbf{P}, \mathbf{K})\mathbf{w}$, then:

$$\|\mathcal{F}_l(\mathbf{P}, \mathbf{K})\|_\infty = \max_{\mathbf{w}(t) \neq 0} \frac{\|z(t)\|_2}{\|\mathbf{w}(t)\|_2} \quad (3.15)$$

where the \mathcal{L}_2 norm on signals is defined as:

$$\|z(t)\|_2 = \sqrt{\int_0^\infty \sum_i |z_i(t)|^2 dt} \quad (3.16)$$

Thus, minimizing the \mathcal{H}_∞ norm corresponds to minimizing the amplification of the exogenous inputs, hence improving disturbance rejection and reference tracking performance.

The resolution of the \mathcal{H}_∞ problem is performed on the augmented plant, which typically consists in the system to be controlled with additional weighting filters. The design specifications are enforced by the means of these filters, which are applied to signals of interest in order to shape their frequency content in the closed-loop system. Two approaches for selecting these filters are commonly employed: the mixed-sensitivity approach and the signal-based approach. In the mixed-sensitivity approach [Kwakernaak, 1993], the weighting filters are applied to the closed-loop sensitivity function $\mathbf{S}_O = (\mathbb{I} + \mathbf{G}\mathbf{K})^{-1}$ and one or more other closed-loop transfer functions such as $\mathbf{K}\mathbf{S}_O$ or the complementary sensitivity function $\mathbf{T}_O = \mathbb{I} - \mathbf{S}_O$. The signal-based approach is more general and considers weighting filters applied to both the exogenous inputs, where they describe the relative importance or the frequency content of these signals, and to the performance outputs, where they describe the desired frequency content. In all cases, the selection of weighting filters is usually a tedious process subject to trial-and-error, where the designer needs to conduct a careful trade-off between closed-loop performance, control signal energy and robustness, even though guidelines for their selection do exist [Skogestad and Postlethwaite, 2007].

In practice, it is generally not necessary to obtain an optimal controller for the \mathcal{H}_∞ problem and it is often more computationally tractable to find a suboptimal controller. Let γ_{\min} be the \mathcal{H}_∞ -optimal gain, i.e. the minimum value of $\|\mathcal{F}_l(\mathbf{P}, \mathbf{K})\|_\infty$ over all stabilizing controllers \mathbf{K} . The \mathcal{H}_∞ suboptimal control problem is then formulated as follows: given $\gamma > \gamma_{\min}$, find all stabilizing controllers \mathbf{K} such that

$$\|\mathcal{F}_l(\mathbf{P}, \mathbf{K})\|_\infty < \gamma \quad (3.17)$$

One of the major milestones in \mathcal{H}_∞ control was reached with the state-space solution to the suboptimal \mathcal{H}_∞ problem formulated by Doyle et al. [1989], which consists in solving two Riccati equations with iteratively reducing γ . A more recent approach is based on semidefinite programming and consists in formulating the suboptimal \mathcal{H}_∞ control problem as a system of Linear Matrix Inequalities (LMIs) [Gahinet and Apkarian, 1994]. Both techniques are based on the state-space representation of the augmented plant \mathbf{P} given in Equation (3.11).

3.2.3.3 Fixed-Structure and Fixed-Order Design

The main disadvantage of these standard synthesis techniques is that the obtained controller is full order, or *unstructured*, and its order is equal to the order of the open-loop system plus the order of the weighting filters. These controllers have many practical limitations, which has slowed the adoption of traditional \mathcal{H}_∞ synthesis in the industry [Gahinet and Apkarian, 2011]. Indeed, \mathcal{H}_∞ controllers are monolithic with high order, whereas most embedded control architectures are based on decentralized collections of low-complexity elements such as gains, lead/lag filters and PID controllers. The interpretation of \mathcal{H}_∞ controllers in terms of these structured architectures is often difficult, and optimization-based tuning or manual tuning remains the norm in many applications.

For these reasons it is desirable to introduce constraints on the controller structure and order. Contrary to the traditional full-order \mathcal{H}_∞ synthesis techniques, the \mathcal{H}_∞ problem with structural constraints on \mathbf{K} is typically a non-convex, non-smooth problem. The main challenge with this method was to find efficient algorithms to solve the underlying optimization problem. While solutions involving bilinear matrix inequalities (BMIs) have been explored [Chung, 2003], they are usually limited by numerical difficulties even for problems of moderate size. More efficient approaches based on state-of-the-art nonsmooth optimizers have been proposed in the last decade and have resulted in robust and user-friendly tools for the control practitioner. The technique developed by Apkarian and Noll [2006] is based on evaluating the \mathcal{H}_∞ norm using the Hamiltonian bisection algorithm in an iterative descent procedure, and forms the algorithmic foundation of the `hinfstruct` tool of the MATLAB Robust Control Toolbox. An open-source equivalent of this tool is the `HIFOO` (H-Infinity Fixed-Order Optimization) solver introduced by Burke et al. [2006] and Gumussoy et al. [2009]. Both `hinfstruct` and `HIFOO` rely on algorithms that search for local minima, hence in addition to the augmented plant and the controller structure, it is also necessary to specify initial values of the controller parameters. Even though these solvers do not aim at finding global minima, the obtained controllers often compare favorably with full-order solutions [Řezáč and Hurák, 2013]. In recent MATLAB releases, the functionalities of `hinfstruct` have been included in the `systeme` tool.

3.2.4 Multi-objective \mathcal{H}_∞ Control

Multiobjective control aims at designing a feedback control law meeting potentially conflicting specifications defined on various input/output channels [Gumussoy et al., 2009; Scherer, 2000]. For instance, in most control systems keeping the control error at a minimum conflicts with control input energy usage. Consider also the case of a simultaneous disturbance-rejection and reference-tracking problem with model-matching, discussed later in this chapter as well as in Appendix B. In classical \mathcal{H}_∞ synthesis diagrams, the model-matching error depends both on the reference and the disturbance inputs. The latter transfer is not relevant to the reference tracking objective and may result in degraded controller performance. Using multi-objective synthesis, these two objectives can be considered separately. More generally, these frameworks offer much more flexibility to the designer than the classical \mathcal{H}_∞ control framework, and design specifications may include \mathcal{H}_2 and \mathcal{H}_∞ performance, passivity, time-domain constraints, etc.

As for the classical \mathcal{H}_∞ synthesis, any control architecture can be transformed into a standard control configuration, as illustrated in Figure 3.4. Unlike classical \mathcal{H}_∞ synthesis, multi-objective control techniques do not require the exogenous inputs and performance outputs to be the same for all control objectives. Instead, multiple $w_i \rightarrow z_i$ performance channels are considered, with possibly different signals and weighting filters for each objective. Furthermore, multi-objective control techniques can also provide a practical way of taking plant variations and uncertainties into account, by tuning the controller against a set of plant models representative of plant variations during operation [Apkarian et al., 2014]. In its most general form, the multi-objective control problem can be formulated as a constrained optimization problem with hard (mandatory) and soft (desirable) requirements:

$$\begin{aligned} \underset{\mathbf{K}}{\text{minimize}} \quad & \max_{i,k} \{ \|\mathbf{T}_{w_i \rightarrow z_i}(\mathbf{P}_k, \mathbf{K})\| \} \\ \text{subject to} \quad & \max_{j,k} \{ \|\mathbf{T}_{w_j \rightarrow z_j}(\mathbf{P}_k, \mathbf{K})\| \} \leq 1 \end{aligned} \quad (3.18)$$

where \mathbf{P}_k is the k -th plant model in the set and $\|\cdot\|$ denotes either the \mathcal{H}_2 norm or the \mathcal{H}_∞ norm. This problem aims at minimizing the worst-case value of the soft requirements $\|\mathbf{T}_{w_i \rightarrow z_i}\|$ while enforcing the hard requirements $\|\mathbf{T}_{w_j \rightarrow z_j}\|$.

If only \mathcal{H}_∞ performance is considered, with no distinction between hard and soft constraints and for a single plant, the multi-objective design problem can be formulated similarly to a suboptimal \mathcal{H}_∞ optimization problem: given $\gamma > \gamma_{\min}$, find all stabilizing controllers \mathbf{K} such that

$$\max_i \{ \|\mathbf{T}_{w_i \rightarrow z_i}(\mathbf{P}, \mathbf{K})\|_\infty \} < \gamma \quad (3.19)$$

LMI solutions to the multi-objective design problem have been proposed [Scherer et al., 1997; Scherer, 2000] for objectives which can be formulated in a common Lyapunov function. However, these approaches are not tractable in the general case, nor when structural constraints on the controller are applied. More general algorithms based on nonconvex optimization such as those provided in HIFOO [Gumussoy et al., 2009] or MATLAB's `syntune` function [Apkarian et al., 2014] are formulated on finding coefficients of a fixed-structure, fixed-order controller. As finding the global optimum is a computationally difficult problem, these algorithms search for local minima, using several starting points to avoid unsatisfactory local solutions.

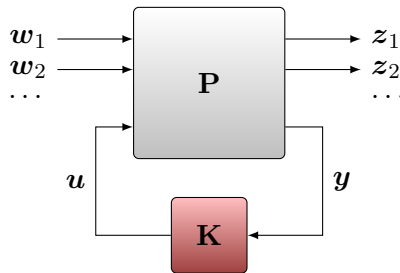


FIGURE 3.4: Multi-objective synthesis problem in standard form.

3.2.5 Robustness Analysis

The main goal of any control system is to ensure the closed-loop system remains stable at all times, with sufficient margin against instability for the considered sources of uncertainty. This section considers linear robust stability analysis techniques, which evaluate this requirement by calculating stability margins for the considered linear representations of the system.

3.2.5.1 Stability Margins

In classical control design approaches, stability robustness requirements are given in terms of minimum gain and phase margins for each loop of the system, typically $\text{GM} > 6 \text{ dB}$ and $\text{PM} > 30^\circ$. These margins are robustness measures defined using the nominal SISO open-loop transfer function $\mathbf{L}_O = \mathbf{G}\mathbf{K}$ and indicate the closeness of the curve $\mathbf{L}(j\omega)$ to the -1 point in the complex plane. For systems which include pure time delays (such as those induced by discrete-time controller implementations), a particularly important interpretation of the phase margin is the maximum time delay that may be introduced before the system becomes unstable (delay margin):

$$\theta_{\max} = \frac{\text{PM}}{\omega_c} \quad (3.20)$$

where ω_c is the gain crossover frequency, at which $|\mathbf{L}(j\omega_c)| = 1$.

For MIMO systems, the classical gain and phase margins may be calculated on a loop-by-loop basis by breaking one loop at a time, while the other loops remain closed, and analyzing the resulting SISO open-loop transfer function. However these margins do not constitute a sufficient robust stability condition for MIMO systems as they can be unreliable for systems exhibiting significant coupling, as shown in Bates and Postlethwaite [2002]. In the latter, multivariable gain and phase margins based on singular value robustness measures of the $\mathbf{S} - \mathbf{T}$ transfer with unstructured gain and phase uncertainty are proposed, where $\mathbf{S} = (\mathbb{I} + \mathbf{L})^{-1}$ is the sensitivity function and $\mathbf{T} = \mathbf{L}(\mathbb{I} + \mathbf{L})^{-1}$ is the complementary sensitivity function. Using the Small Gain Theorem, guaranteed lower bounds on these margins are defined as follows:

$$\text{GM}_{\text{ST}} = \left[\frac{1 - r_{\min}}{1 + r_{\min}}, \frac{1 + r_{\min}}{1 - r_{\min}} \right] \quad (3.21)$$

$$\text{PM}_{\text{ST}} = \left[-2 \tan^{-1} r_{\min}, 2 \tan^{-1} r_{\min} \right] \quad (3.22)$$

where

$$r_{\min} = \inf_{\omega \in \mathbb{R}} \frac{1}{\bar{\sigma} \left[(\mathbb{I} - \mathbf{L})(\mathbb{I} + \mathbf{L})^{-1} \right]} \quad (3.23)$$

These margins are valid for simultaneous uncertainty in gain **or** phase in multiple loops of the system.

The classical and multivariable gain and phase margins are necessary robust stability conditions, but they are not sufficient in that they do not consider simultaneous gain **and** phase uncertainty. They also do not take into account explicit uncertainty representations, which are essential to guarantee the identification of the real worst-cases. Modern robust control theory provides necessary and sufficient conditions for evaluating robust stability for all perturbations in a given uncertain model.

3.2.5.2 Robust Stability

The closed-loop system taking into account the modeled uncertainty can be represented using the general framework illustrated in Figure 3.5A. In this general control configuration, the uncertainty block Δ is "pulled out" by augmenting the plant model with inputs and outputs to the Δ block, forming the generalized plant \mathbf{P}^Δ . Robustness analyses are conducted using the $\mathbf{N} - \Delta$ representation illustrated in Figure 3.5B, in which the nominal closed-loop system \mathbf{N} is related to \mathbf{P}^Δ and \mathbf{K} by the lower LFT:

$$\mathbf{N} = \mathcal{F}_l(\mathbf{P}^\Delta, \mathbf{K}) \tag{3.24}$$

The resulting transfer function \mathbf{N} can be partitioned as:

$$\begin{bmatrix} \mathbf{y}_\Delta \\ \mathbf{z} \end{bmatrix} = \mathbf{N} \begin{bmatrix} \mathbf{u}_\Delta \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} \\ \mathbf{N}_{21} & \mathbf{N}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Delta \\ \mathbf{w} \end{bmatrix} \tag{3.25}$$

The uncertain closed-loop transfer function is then given by the upper LFT of \mathbf{N} and Δ :

$$\mathbf{T}_{\mathbf{w} \rightarrow \mathbf{z}}^\Delta = \mathcal{F}_u(\mathbf{N}, \Delta) \tag{3.26}$$

$$= \mathbf{N}_{22} + \mathbf{N}_{21}\Delta(\mathbb{I} - \mathbf{N}_{11}\Delta)^{-1}\mathbf{N}_{12} \tag{3.27}$$

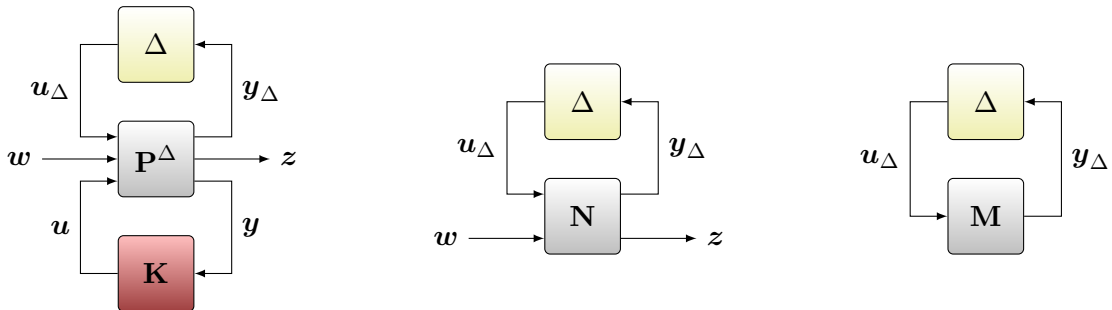
Assuming the nominal system \mathbf{N} and the uncertain block Δ are stable, the only remaining source of instability in (3.27) is $(\mathbb{I} - \mathbf{N}_{11}\Delta)^{-1}$. Hence, robust stability analysis consists in assessing the stability of the $\mathbf{M}\Delta$ transfer represented in Figure 3.5C, where $\mathbf{M} = \mathbf{N}_{11}$.

When \mathbf{M} and Δ are SISO transfer functions, the Nyquist stability criterion may be applied. The system is then robustly stable (RS) if and only if the loop transfer function $\mathbf{M}\Delta$ does not encircle the $(-1, 0)$ point for all Δ [Skogestad and Postlethwaite, 2007]:

$$\text{RS} \Leftrightarrow |1 + \mathbf{M}\Delta| > 0, \quad \forall \omega, \forall |\Delta| \leq 1 \tag{3.28}$$

The worst case is obtained for Δ selected at each frequency such that $|\Delta| = 1$ and $\mathbf{M}\Delta$ has negative sign. Then, the above condition can be rewritten as:

$$\text{RS} \Leftrightarrow 1 - |\mathbf{M}(j\omega)| > 0, \quad \forall \omega \tag{3.29}$$



(A) General control configuration with uncertainty.

(B) $\mathbf{N} - \Delta$ structure for robust performance analysis.

(C) $\mathbf{M} - \Delta$ structure for robust stability analysis.

FIGURE 3.5: Uncertain control configurations.

$$\text{RS} \Leftrightarrow |\mathbf{M}(j\omega)| < 1, \quad \forall \omega \quad (3.30)$$

which is a necessary and sufficient stability condition.

For MIMO systems, a similar robust stability condition can be obtained using the small gain theorem [Bates and Postlethwaite, 2002]. Assuming the nominal system is stable and Δ is a full (unstructured) complex transfer function matrix satisfying $\|\Delta\|_\infty \leq 1$,

$$\text{RS} \Leftrightarrow \bar{\sigma}[\mathbf{M}(j\omega)] < 1, \quad \forall \omega \quad (3.31)$$

When structured uncertainty representations are considered, the Δ is no longer full but has a block-diagonal structure $\Delta = \text{diag}\{\Delta_i\}$. In this case, the robust stability condition given in Equation (3.31) is only sufficient. The result will be conservative since the small gain theorem considers a full Δ matrix, while most elements are zero in a structured Δ matrix. In order to take into account the structure of Δ in the analysis and thus to reduce conservatism, the structured singular value μ has been introduced by Doyle [1982]. This function is based on finding the smallest factor k_m making the matrix $\mathbb{I} - k_m \mathbf{M} \Delta$ singular, where Δ is structured and $\bar{\sigma}(\Delta) \leq 1$:

$$\mu(\mathbf{M}) = \frac{1}{\min[k_m \text{ s.t. } \det(\mathbb{I} - k_m \mathbf{M} \Delta) = 0 \text{ for structured } \Delta, \bar{\sigma}(\Delta) \leq 1]} \quad (3.32)$$

The robust stability condition for structured uncertainty is then given in terms of μ . Assuming \mathbf{M} and Δ are stable, with $\|\Delta\|_\infty \leq 1$,

$$\text{RS} \Leftrightarrow \mu(\mathbf{M}) \leq 1, \quad \forall \omega \quad (3.33)$$

The main difficulty associated with this robust stability condition is that the computation of μ is a NP-hard problem. This means there may not be a polynomial-time algorithm for finding the exact value of μ and thus the number of required operations is an exponential function of the problem size. In order to reduce the problem complexity, the exact value of μ is not computed but approached with upper and lower bounds. The upper bound then provides a sufficient stability condition for a given amount of structured uncertainty, while the lower bound gives a sufficient condition for instability. Thus the tighter these bounds are, the less conservative this robustness condition is.

An additional difficulty is that polynomial-time algorithms for determining an accurate upper bound of μ are only available for complex or mixed (complex and real) structured uncertainties. In the purely real case, for which the exact μ plot is often a discontinuous function of the frequency, μ upper bounds (necessarily obtained by a mixed μ algorithm for a high dimensional problem) are generally conservative. Moreover, for such cases, it is quite difficult to evaluate this conservatism since the μ lower bound will often fail to converge. A practical workaround is then to introduce a small amount of fictitious complex uncertainty to the system until a pseudo lower bound is obtained for the modified system. Next, by a migration technique, using standard optimization routines, a lower bound can be deduced for the initial system. Such a method is detailed in Ferreres and Biannic [2001] and implemented in the latest version of the SMART library of the SMAC toolbox [Biannic et al., 2016]. See also Roos and Biannic [2015] for a detailed comparison of all existing algorithms.

3.3 Fixed-Airspeed Autopilot

The mathematical tools presented in the last sections will now be used for the design of a skid-to-turn autopilot for the ACHILES prototype. In this first application, the objective is to control simultaneously the pitch and yaw motions of the projectile, while the roll axis remains fixed at zero. The airspeed in the wind tunnel is set to $V = 25$ m/s. The flight domain is thus two-dimensional and the projectile's dynamic behavior can be locally described by the 2-inputs, 4-outputs linear model given in Equation (2.77). The full design procedure will be detailed in the next paragraphs and consists in the following steps:

- Definition of a nominal model for the control synthesis and computation of the associated uncertain model.
- Choice of a control architecture. Low-complexity controller structures are desirable in order to allow implementation on low-power, low-cost embedded computers, and to serve as the basis for the gain-scheduled autopilot.
- Controller synthesis, based on the desired specifications.
- Robustness analysis.
- Simulation and experimental validation to demonstrate the ability of the proposed approach to achieve the expected performance.

This procedure naturally relies on the estimation results obtained in Chapter 2. The combined pitch and yaw case extends the pitch-only case, which is reproduced in Appendix B and discussed in [Strub et al., 2015a,b]. In this preliminary study, the different controller synthesis techniques introduced in the previous section were compared for the design of an autopilot achieving reference tracking and disturbance rejection objectives. While both full-order \mathcal{H}_∞ synthesis and fixed-order \mathcal{H}_∞ synthesis produced high-performance disturbance rejection controllers, a supplementary step was necessary to achieve satisfying reference tracking with the addition of an inverse-based feedforward controller. Multi-objective \mathcal{H}_∞ synthesis aims at achieving both objectives in a single design step, which greatly simplifies the designer's task, and results in a single controller structure with similar performance to the above mentioned two-step approaches.

The reader is encouraged to consult the Appendix and [Strub et al., 2015a,b] for greater detail concerning the pitch-only case. Both for the pitch-only case and the pitch and yaw case, similar conclusions can be drawn, and the main differences between these two cases will be pointed out.

3.3.1 Synthesis Model

3.3.1.1 Open-Loop Plant

The open-loop model considered for controller synthesis is illustrated in Figure 3.6. It consists in the linearized plant dynamics \mathbf{G}_B given in Equation (2.77), augmented with the actuator dynamics \mathbf{G}_A . The $\mathbf{G}_C\mathbf{G}_Z$ block is added to account for the delays introduced by the embedded electronics. Indeed, the designed autopilot will be implemented on the ACHILES embedded computer using a discrete-time approximation of the continuous-time controllers. The discretized controller is interfaced with

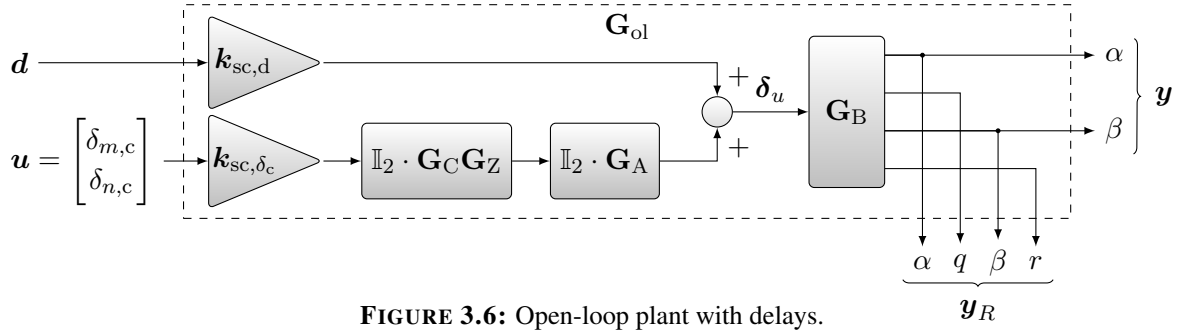


FIGURE 3.6: Open-loop plant with delays.

the continuous plant using sample and hold circuits, which introduce a hold delay in the control loop. Moreover, the software framework introduces an additional 1-cycle computation delay to account for latencies in the actuator and sensor communication interfaces. Contrary to the pitch-only case, where the controllers were designed with a delay-free nominal model, in the present case these delays are included in the synthesis models to improve the performance and stability of the discrete autopilots [Stevens and Lewis, 2003]. The transfer functions of the computation delay and the sample-and-hold circuit are respectively given by, with T the sampling period:

$$\mathbf{G}_C = e^{-sT} \quad \mathbf{G}_Z = \frac{1 - e^{-sT}}{sT} \quad (3.34)$$

These non-rational transfer functions can be approximated using a first-order Padé approximation of the e^{-sT} term, resulting in the following rational transfer functions, suitable for controller synthesis:

$$\hat{\mathbf{G}}_C = \frac{1 - sT/2}{1 + sT/2} \quad \hat{\mathbf{G}}_Z = \frac{1}{1 + sT/2} \quad (3.35)$$

The open-loop transfer function of the system illustrated in Figure 3.6 can then be expressed as:

$$\mathbf{G}_{ol} : \begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \\ \mathbf{y}_R \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{d} \\ \mathbf{u} \end{bmatrix} \quad (3.36)$$

where the state vector \mathbf{x} is of dimension 14 and contains the 4 airframe dynamics state variables α, q, β, r , the actuator states (2×3 states) and the linearized delays (2×2 states). The outputs $\mathbf{y} = [\alpha \ \beta]^T$ are the Cartesian angles of incidence to be tracked and $\mathbf{y}_R = [\alpha \ q \ \beta \ r]^T$ are the available measurements for control, where q and r are the body angular rates. \mathbf{d} is the disturbance input and $\mathbf{u} = [\delta_{mc} \ \delta_{nc}]^T$ are the control inputs. These inputs are virtual fin deflections, which are distributed over the four actuating canards. In the considered projectile configuration, the canards are placed so that they form a ‘+’ shape at rest as illustrated in Figure 2.4. The virtual fin deflections are applied to the control canards using the mixing logic of Equation (2.34).

Finally, the actuator and disturbance inputs are scaled with respective factors $\mathbf{k}_{sc,\delta_c}, \mathbf{k}_{sc,d}$ such that the open-loop plant has unity steady-state gain. This scaling ensures both channels are treated similarly during \mathcal{H}_∞ synthesis [Skogestad and Postlethwaite, 2007].

Partitioning the state vector in terms of the blocks appearing in Figure 3.6, i.e.

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B^\top & \mathbf{x}_A^\top & \mathbf{x}_A^\top & \mathbf{x}_C^\top & \mathbf{x}_C^\top & \mathbf{x}_Z^\top & \mathbf{x}_Z^\top \end{bmatrix}^\top \quad (3.37)$$

the matrices of Equation (3.36) can be detailed as follows ²²:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_B & \mathbf{B}_B \otimes \mathbf{C}_A & 0 & 0 \\ 0 & \mathbb{I}_2 \otimes \mathbf{A}_A & \mathbb{I}_2 \otimes \mathbf{B}_A \mathbf{C}_C & \mathbb{I}_2 \otimes \mathbf{B}_A \mathbf{D}_C \mathbf{C}_Z \\ 0 & 0 & \mathbb{I}_2 \otimes \mathbf{A}_C & \mathbb{I}_2 \otimes \mathbf{B}_C \mathbf{C}_Z \\ 0 & 0 & 0 & \mathbb{I}_2 \otimes \mathbf{A}_Z \end{bmatrix} \quad (3.38)$$

$$\mathbf{B}_1 = \begin{bmatrix} \mathbf{B}_B \mathbf{k}_{sc,d} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbb{I}_2 \otimes \mathbf{k}_{sc,\delta_c} \end{bmatrix} \quad (3.39)$$

$$\mathbf{C}_1 = \left[\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbb{O}_{2 \times 10} \right] \quad \mathbf{C}_2 = \left[\mathbb{I}_4 \quad \mathbb{O}_{4 \times 10} \right] \quad (3.40)$$

$$\mathbf{D}_{11} = 0 \quad \mathbf{D}_{12} = 0 \quad \mathbf{D}_{21} = 0 \quad \mathbf{D}_{22} = 0 \quad (3.41)$$

where $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i$ are the state-space matrices corresponding to the transfer function \mathbf{G}_i , and \otimes is the Kronecker product. The state-space representations of \mathbf{G}_B and \mathbf{G}_A were respectively given in Equations (2.77) and (1.6), and state-space representations of the Padé approximations of \mathbf{G}_C and \mathbf{G}_Z are given as follows:

$$\hat{\mathbf{G}}_C : \begin{bmatrix} \dot{x}_C \\ y_C \end{bmatrix} = \begin{bmatrix} -\frac{2}{T} & | & \frac{1}{T} \\ 4 & | & -1 \end{bmatrix} \begin{bmatrix} x_C \\ u_C \end{bmatrix} \quad \hat{\mathbf{G}}_Z : \begin{bmatrix} \dot{x}_Z \\ y_Z \end{bmatrix} = \begin{bmatrix} -\frac{2}{T} & | & \frac{2}{T} \\ 1 & | & 0 \end{bmatrix} \begin{bmatrix} x_Z \\ u_Z \end{bmatrix} \quad (3.42)$$

3.3.1.2 Nominal Model Selection

In the open-loop system of Figure 3.6, careful attention should be paid to the subsystem \mathbf{G}_B , which characterizes the dynamic behavior of the ACHILES projectile in the wind tunnel. Indeed, the linearized model (2.77) is only valid in the neighborhood of an equilibrium point. Even if the airspeed is fixed at $V = 25$ m/s, the model parameters are subject to non-negligible variations as the angle of attack α and the angle of sideslip β change (see Figure 2.18). In this case, a family of 18 linear models has been estimated to cover the flight envelope, with the corresponding operating points summarized in Table 2.1. The parameter estimation results at constant airspeed $V = 25$ m/s were discussed in Section 2.6.4. Since the incidence angles are fast variables which cannot be used for controller scheduling, and since the variations of the model parameters remain moderate *w.r.t.* changes in the trim angles of incidence, it becomes then preferable to consider a single controller to account for these variations. Consequently, a nominal model has to be selected to cover the entire flight envelope. The resulting approximations will be treated as model uncertainty in a second step.

In order to select an appropriate nominal model, the two selection strategies based on the ν -gap metric and outlined in Section 3.2.1 are applied. The mutual ν -gap distances between each model in the estimated set are presented in Figure 3.7A, and the ν -gap distances between the estimated models and the mean-parameters model are illustrated in Figure 3.7B. In the set of estimated models, \mathbf{P}_{10} (which corresponds to the trim conditions $\bar{\alpha} = 2^\circ$ and $\bar{\beta} = 0^\circ$) is the best nominal model candidate,

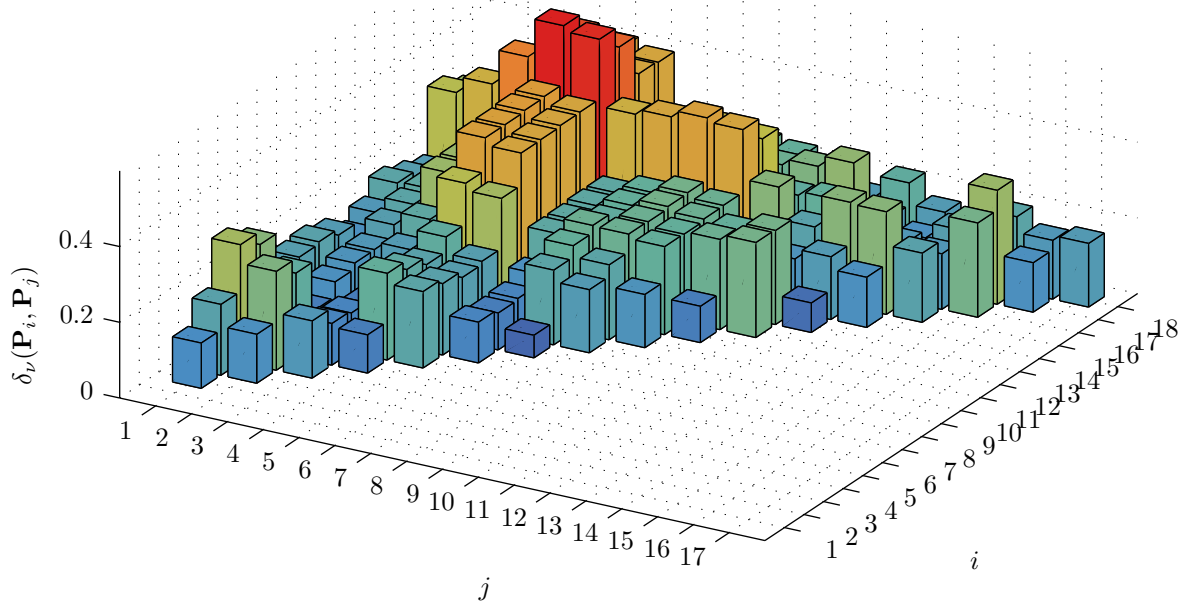
²²For clarity, in matrices of matrices, 0 denotes a matrix of zeros with compatible dimensions; the dimensions are given only when necessary.

with $J(\mathbf{P}_{10}) = 0.33$. Applying the mean-parameters strategy leads to the mean-parameters model $\bar{\mathbf{P}}$, with $J(\bar{\mathbf{P}}) = 0.31$. From Figure 3.7, models \mathbf{P}_4 , \mathbf{P}_5 , \mathbf{P}_{15} and \mathbf{P}_{16} exhibit the highest ν -gap distances with respect to the mean-parameters model and to other models in the set. These models correspond to operating points close to the edges of the flight envelope, where the model fit is degraded due to high external disturbances.

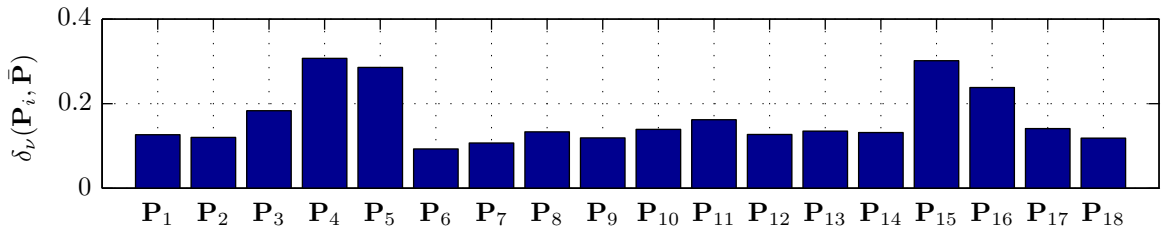
The Nyquist diagrams for the complete set of estimated models and the two nominal model candidates (\mathbf{P}_{10} and $\bar{\mathbf{P}}$) are illustrated in Figure 3.8 and show that both candidate model responses lie within the set of estimated models. Since the mean-parameters model $\bar{\mathbf{P}}$ presents the lowest value of the J criterion defined in Equation (3.2), it is chosen as the nominal model.

3.3.1.3 Uncertain Model

The differences between the nominal model and the set of estimated models, as well as the uncertainty on each estimated model, are taken into account into an uncertain model. This representation allows the designer to assess the robustness of the closed-loop system against the modeled uncertainty.



(A) Mutual ν -gap distances.



(B) ν -gap distances against the mean-parameters model $\bar{\mathbf{P}}$.

FIGURE 3.7: Mutual ν -gap distances (A) for all estimated models at $V = 25$ m/s and (B) against the mean-parameters nominal model $\bar{\mathbf{P}}$. For readability, only the upper diagonal part of the ν -gap distances matrix is shown, as $\delta_\nu(\mathbf{P}_i, \mathbf{P}_j) = \delta_\nu(\mathbf{P}_j, \mathbf{P}_i)$.

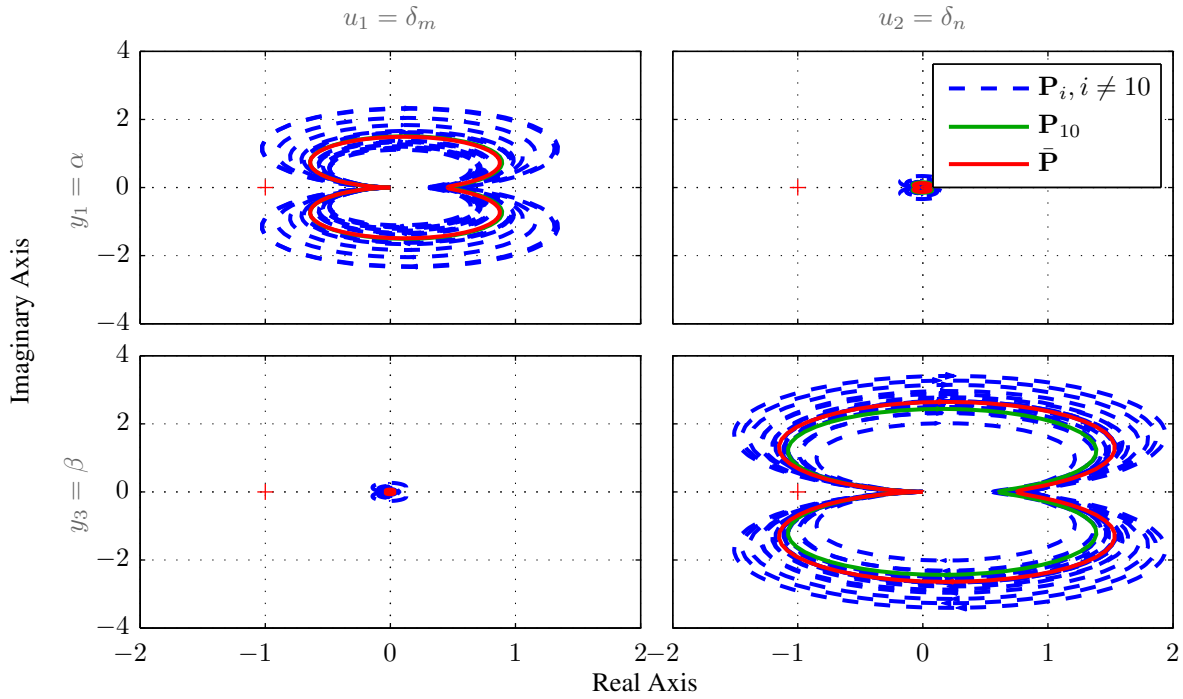


FIGURE 3.8: Nyquist diagrams of the set of estimated models (dashed blue lines), the nominal model candidate from this set (green line) and the mean-parameters nominal model candidate (red line).

In Section 3.2.2, two approaches for uncertainty modeling were presented. In the present case, the model is a 2-input, 4-output, 14th-order system. The unstructured uncertainty representation is frequently used for SISO systems, where all sources of uncertainty are lumped into a single complex perturbation. However this simple approach is often unsuitable for MIMO systems and can result in a too conservative representation, i.e. the set of perturbed models captured by the uncertain model is much larger than the set of observed models [Skogestad and Postlethwaite, 2007].

For this reason, a parametric uncertainty representation is preferred. In this framework, the model parameters M_{ij} are modeled as uncertain parameters using Equations (3.6-3.7). In these equations, the minimum and maximum values $\theta_{i,\min}$ and $\theta_{i,\max}$ are obtained from the minimum and maximum values of the estimated parameters, taking the uncertainty on parameter estimates (obtained with the bootstrap procedure in Section 2.5.7) into account. Rewriting the plant dynamics (2.77) with $M_{ij} = \bar{M}_{ij}(1+r_{ij}\delta_{ij})$ results in an uncertain state-space system, which can be written as the upper LFT interconnexion:

$$\mathbf{G}_{B_p} = \mathcal{F}_u(\mathbf{G}_B^\Delta, \Delta) \quad (3.43)$$

where Δ is a diagonal matrix of dimension 8, with $\|\Delta\|_\infty \leq 1$. $\mathbf{G}_B^{\Delta 23}$ corresponds to the nominal plant \mathbf{G}_B augmented with the relative uncertainty, and inputs and outputs to the Δ block:

$$\mathbf{G}_B^\Delta : \begin{bmatrix} \dot{x} \\ \mathbf{y}_\Delta \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_\Delta & \mathbf{B} \\ \mathbf{C}_\Delta & \mathbf{D}_{\Delta\Delta} & \mathbf{D}_{0,\Delta} \\ \mathbf{C} & \mathbf{D}_{\Delta,0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u}_\Delta \\ \mathbf{u} \end{bmatrix} \quad (3.44)$$

²³Can also be denoted as \mathbf{H}_B following standard notation in [Skogestad and Postlethwaite, 2007; Ferreres, 1999].

with matrices \mathbf{B}_Δ , \mathbf{C}_Δ , $\mathbf{D}_{\Delta\Delta}$, $\mathbf{D}_{\Delta,0}$ and $\mathbf{D}_{0,\Delta}$ defined as follows:

$$\mathbf{B}_\Delta = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (3.45)$$

$$\mathbf{C}_\Delta = \begin{bmatrix} \bar{M}_{q\alpha} r_{q\alpha} & 0 & 0 & 0 \\ 0 & \bar{M}_{qq} r_{qq} & 0 & 0 \\ 0 & 0 & \bar{M}_{q\beta} r_{q\beta} & 0 \\ 0 & 0 & 0 & 0 \\ \bar{M}_{r\alpha} r_{r\alpha} & 0 & 0 & 0 \\ 0 & 0 & \bar{M}_{r\beta} r_{r\beta} & 0 \\ 0 & 0 & 0 & \bar{M}_{rr} r_{rr} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{D}_{0,\Delta} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \bar{M}_{q\delta_m} r_{q\delta_m} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \bar{M}_{r\delta_n} r_{r\delta_n} \end{bmatrix} \quad (3.46)$$

$$\mathbf{D}_{\Delta\Delta} = \mathbb{O}_{8 \times 8} \quad \mathbf{D}_{\Delta,0} = \mathbb{O}_{4 \times 8} \quad (3.47)$$

Time- and frequency-domain responses of the uncertain model $\mathbf{G}_{B_p} = \tilde{\mathbf{P}}$, the nominal model $\mathbf{G}_B = \bar{\mathbf{P}}$ and the family of estimated models $\mathbf{\Pi}$ are illustrated in Figure 3.9. In Figure 3.9C, the difference in pitch and yaw dynamics is clearly visible. Even though the uncertain model is slightly more conservative for the yaw channel than for the pitch channel, the nominal and uncertain models offer an excellent coverage of the family of estimated models. The higher coverage provided by $\tilde{\mathbf{P}}$ comes from considering directions of the hypercube Δ which were not present in $\mathbf{\Pi}$.

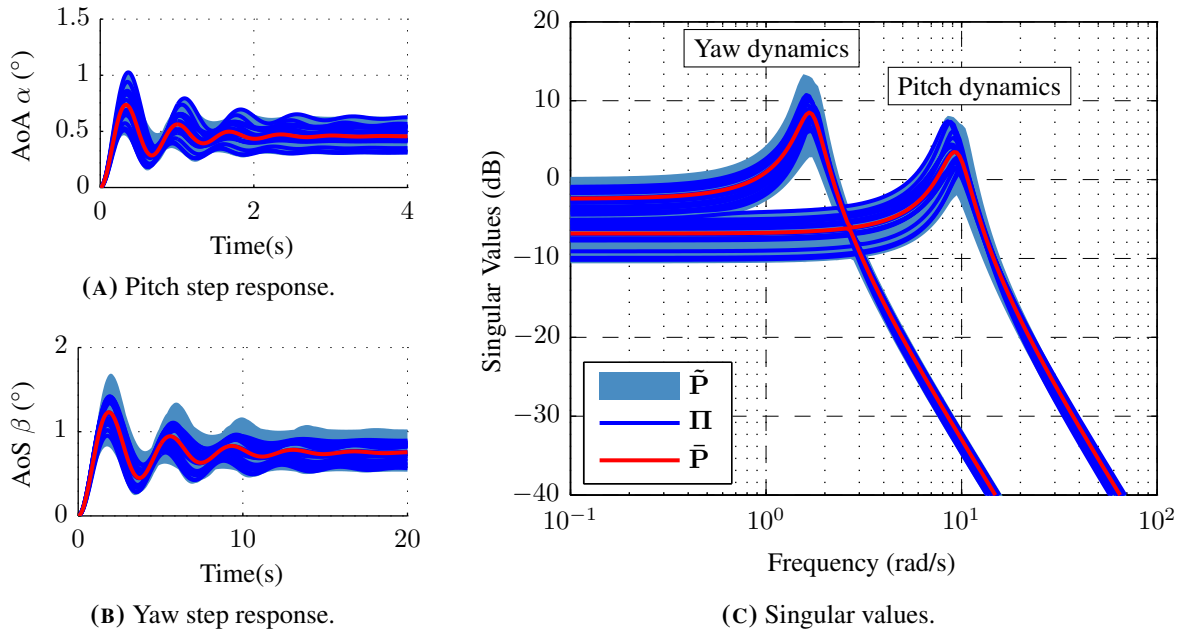


FIGURE 3.9: Time- and frequency-domain responses of the nominal model $\bar{\mathbf{P}}$, the family of estimated models $\mathbf{\Pi}$ and the envelope of uncertain models $\tilde{\mathbf{P}}$, obtained from 1000 randomly sampled realizations of $\tilde{\mathbf{P}}$.

3.3.2 Autopilot Design

The tasks of the autopilot are threefold; its most important task being to ensure the stability of the system at all times. Secondly, the influence of external disturbances d on the regulated outputs should be minimized, and its key goal is to drive the plant input so as to track reference angle of incidence signals α_r and β_r issued by the guidance module. Since the open-loop dynamics exhibit non-negligible variations throughout the flight envelope, closed-loop control configurations are preferred over open-loop schemes, which are highly sensitive to disturbance and uncertainty. The controller structure should present low complexity, for practical implementation issues and to allow adaptation to large operating condition variations with simple interpolation schemes.

In this regard, simple controller structures involving low-complexity elements such as gains and PIDs and that consider each axis separately have been successfully applied to missile and projectile autopilot design problems [Theodoulis et al., 2015; Sève et al., 2014; Jackson, 2010; Nichols et al., 1993]. For the above reasons, low-complexity controllers are often preferred to full-order or nonlinear controllers, yet may offer comparable performance and robustness if carefully designed.

3.3.2.1 Autopilot Structure

The choice of the autopilot structure is motivated by the design specifications and the limitations of the system. These requirements typically include constraints on the plant inputs, acceptable steady-state tracking errors and/or minimum stability margins. As stated above, practical implementation also fosters low-complexity controller structures. The choice of a structure is often not unique, since in general several structures may fit these requirements.

The skid-to-turn autopilot structure that has been considered here is illustrated in Figure 3.10 and is inspired from multiloop structures found in [Mracek and Ridgely, 2005; Gahinet and Apkarian, 2011]. It is based around three controller blocks arranged in two nested feedback loops: an inner *regulation* loop with proportional gain matrix \mathbf{K}_R and an outer *servo* loop based on the error-tracking controller \mathbf{K}_S . These two loops are responsible for stability augmentation (\mathbf{K}_R), rejection of external disturbances and steady-state tracking error cancellation (\mathbf{K}_S). The feedforward controller \mathbf{K}_F shapes the frequency content of the reference signal to provide the desired closed-loop reference tracking behavior. An additional roll-off filter \mathbf{F}_{ro} protects the plant from high-frequency control signal injection. The structure of the autopilot blocks is discussed thereafter. The tunable controller parameters of each block are denoted

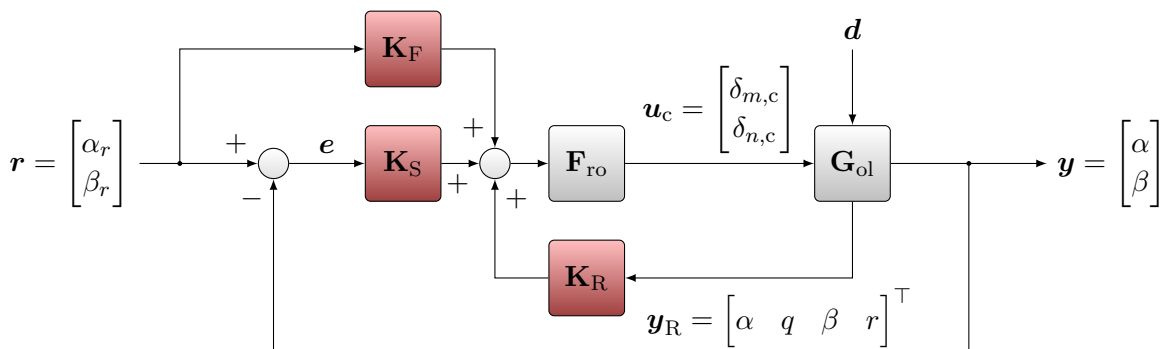


FIGURE 3.10: Structure of the constant-air-speed skid-to-turn autopilot.

$k_A^{j,m}$, where k indicates a gain (resp. p for a pole and z for a zero), A corresponds to the role of the block, j corresponds to the signal on which the parameter acts, and m refers to the pitch channel (resp. n for the yaw channel).

The regulation controller \mathbf{K}_R is chosen as a 2×4 proportional controller which is applied to the measured incidence angles and body rates. The off-channel gains are fixed to zero since the coupling terms of the model are highly uncertain but moderate in magnitude, resulting in the following structure:

$$\mathbf{K}_R = \begin{bmatrix} k_R^{\alpha,m} & k_R^{q,m} & 0 & 0 \\ 0 & 0 & k_R^{\beta,n} & k_R^{r,n} \end{bmatrix} \quad (3.48)$$

The servo controller \mathbf{K}_S is chosen as a diagonal integral controller applied to the angle of attack and the angle of sideslip tracking errors, e_α and e_β respectively:

$$\mathbf{K}_S = \begin{bmatrix} \frac{1}{s} k_S^{e_\alpha,m} & 0 \\ 0 & \frac{1}{s} k_S^{e_\beta,n} \end{bmatrix} \quad (3.49)$$

This controller ensures the output does not present static error. Together with the regulation loop based on \mathbf{K}_R , this structure is analogous to a PID controller but presents the advantage of not introducing an extra zero and hence the reference-to-actuator transfer rolls off naturally.

The feedforward controller is implemented in *injection* form, i.e. the filtered reference signal is summed with the feedback controllers outputs as shown in Figure 3.10. In the present case, the feedforward block is a diagonal array of first-order lead or lag filters:

$$\mathbf{K}_F = \begin{bmatrix} k_F^{\alpha,m} \frac{z_F^{\alpha_r,m} s + 1}{p_F^{\alpha_r,m} s + 1} & 0 \\ 0 & k_F^{\beta,n} \frac{z_F^{\beta_r,n} s + 1}{p_F^{\beta_r,n} s + 1} \end{bmatrix} \quad (3.50)$$

This filter will be tuned so that the closed-loop systems matches the expected tracking performance, specified in terms of a reference model \mathbf{T}_{ref} explained further below.

The open-loop system exhibits a flexible mode near $\omega_{\text{flex}} = 77$ rad/s on the yaw channel due to the long reach between the pitch bearings and the roll bearing, and initial experimental tests showed that sufficient excitation of this mode can cause closed-loop instability. In order to prevent excitation of this mode, a first-order roll-off filter with corner frequency $\omega_{\text{ro},\delta_n} = 15$ rad/s is applied to the input signal corresponding to this channel. On the other hand, the pitch axis does not exhibit flexible behavior and thus the pitch channel of \mathbf{F}_{ro} has unity gain at all frequencies. The expression of \mathbf{F}_{ro} is:

$$\mathbf{F}_{\text{ro}} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{1/\omega_{\text{ro},\delta_n} s + 1} \end{bmatrix} \quad (3.51)$$

In this configuration, the control input \mathbf{u}_c , the tracked output \mathbf{y} and the associated closed-loop transfer functions $\mathbf{T}_{r \rightarrow \mathbf{u}_c}$, $\mathbf{T}_{d \rightarrow \mathbf{u}_c}$, $\mathbf{T}_{r \rightarrow \mathbf{y}}$ and $\mathbf{T}_{d \rightarrow \mathbf{y}}$ are expressed as:

$$\mathbf{u}_c = (\mathbb{I} - \mathbf{L}_{\mathbf{u}_c})^{-1} \mathbf{F}_{\text{ro}} (\mathbf{K}_S + \mathbf{K}_F) \mathbf{r}$$

$$\begin{aligned}
& + (\mathbb{I} - \mathbf{L}_{u_c})^{-1} \mathbf{F}_{ro} (\mathbf{K}_R \mathbf{G}_{ol,d \rightarrow y_R} - \mathbf{K}_S \mathbf{G}_{ol,d \rightarrow y}) d \\
& = \mathbf{T}_{r \rightarrow u_c} \mathbf{r} + \mathbf{T}_{d \rightarrow u_c} d
\end{aligned} \tag{3.52}$$

$$\begin{aligned}
\mathbf{y} & = \mathbf{G}_{ol,u_c \rightarrow y} (\mathbb{I} - \mathbf{L}_{u_c})^{-1} \mathbf{F}_{ro} (\mathbf{K}_S + \mathbf{K}_F) \mathbf{r} \\
& + [\mathbf{G}_{ol,d \rightarrow y} + \mathbf{G}_{ol,u_c \rightarrow y} (\mathbb{I} - \mathbf{L}_{u_c})^{-1} \mathbf{F}_{ro} (\mathbf{K}_R \mathbf{G}_{ol,d \rightarrow y_R} - \mathbf{K}_S \mathbf{G}_{ol,d \rightarrow y})] d \\
& = \mathbf{T}_{r \rightarrow y} \mathbf{r} + \mathbf{T}_{d \rightarrow y} d
\end{aligned} \tag{3.53}$$

with

$$\mathbf{L}_{u_c} = \mathbf{F}_{ro} (\mathbf{K}_R \mathbf{G}_{ol,u_c \rightarrow y_R} - \mathbf{K}_S \mathbf{G}_{ol,u_c \rightarrow y}) \tag{3.54}$$

3.3.2.2 Controller Synthesis Framework

The structured nature of the autopilot configuration in Figure 3.10 naturally calls for fixed-structure \mathcal{H}_∞ synthesis techniques. The design of the three controller blocks \mathbf{K}_R , \mathbf{K}_S , and \mathbf{K}_F is carried out using a \mathcal{H}_∞ mixed-sensitivity formulation [Kwakernaak, 1993] with model-matching constraints [Ho et al., 1992]. The latter technique consists in specifying the closed-loop tracking performance requirements by the means of a reference model exhibiting the desired time-domain response. The model-matching constraint is enforced during synthesis and consists in minimizing the difference between the closed-loop system response and the reference model.

To this end, the design problem is cast into the standard control configuration illustrated in Figure 3.11. The augmented plant \mathbf{P} includes the open-loop system \mathbf{G}_{ol} whose state-space representation is

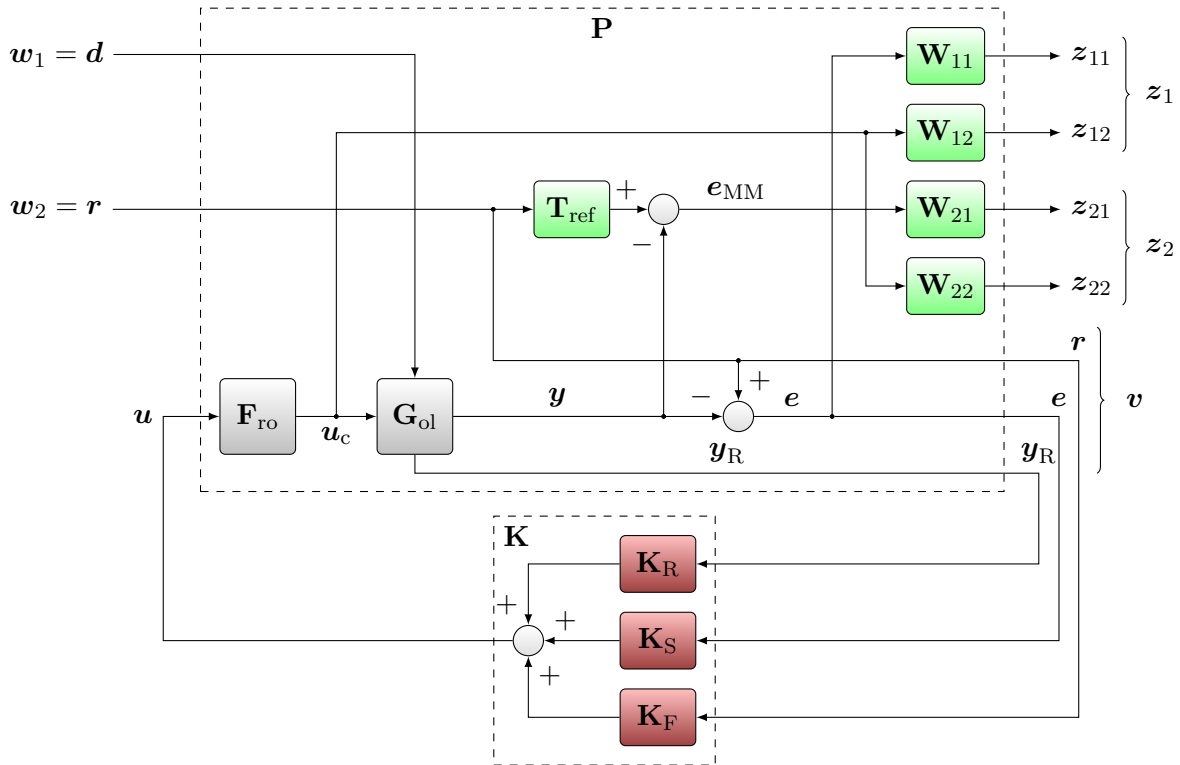


FIGURE 3.11: Skid-to-turn multi-objective autopilot synthesis diagram.

given in Equation (3.36), the roll-off filter of Equation (3.51), the reference model \mathbf{T}_{ref} and the weighting filters \mathbf{W}_{ij} . Below the augmented plant, the block $\mathbf{K} = [\mathbf{K}_R \quad \mathbf{K}_S \quad \mathbf{K}_F]$ contains the three controllers to be determined during the synthesis. The inputs of \mathbf{P} are the *exogenous inputs* $\mathbf{d} = \mathbf{w}_1$, i.e. the disturbances, and $\mathbf{r} = [\alpha_r \quad \beta_r]^\top = \mathbf{w}_2$, i.e. the references, and the control inputs \mathbf{u} containing the virtual canards deflections. Likewise, the outputs of \mathbf{P} combine the *performance signals* (or exogenous outputs) $\mathbf{z} = [z_1^\top \quad z_2^\top]^\top$, which will be further detailed, and the signals used by the controllers, i.e. the references \mathbf{r} , the tracking errors \mathbf{e} and the open-loop system outputs \mathbf{y}_R .

The augmented plant \mathbf{P} has been chosen in order to distinguish two control objectives, namely the disturbance rejection objective and the reference tracking one, using the multiobjective framework of Section 3.2.4. This kind of approach is also often referred to as a 2 degrees-of-freedom design problem [Skogestad and Postlethwaite, 2007]. First, the disturbance rejection issue considers the transfer function from the disturbance input \mathbf{d} to the performance signal $\mathbf{z}_1 = [z_{11}^\top \quad z_{12}^\top]^\top$. The considerations are double. On the one hand, it has to limit the effect of the disturbance on the tracking error \mathbf{e} by shaping the closed-loop sensitivity function $\mathbf{S}_{\mathbf{d} \rightarrow \mathbf{e}} = -\mathbf{T}_{\mathbf{d} \rightarrow \mathbf{y}}$ via the weighting filter \mathbf{W}_{11} . On the other hand, the design objective takes care of the actuator limitations, i.e. bandwidth and amplitude, via the weighting filter \mathbf{W}_{12} , such that the control signal \mathbf{u}_c does not exceed the actuator input specifications.

In the same way, the reference tracking objective aims at tuning the controller \mathbf{K} accordingly with the expected closed-loop performance in terms of precision, response time and overshoot. To this end, a reference model \mathbf{T}_{ref} is introduced in order to specify explicitly the desired performance. The main idea is that the closed-loop transfer function $\mathbf{T}_{\mathbf{r} \rightarrow \mathbf{y}}$ fits the reference model in the desired frequency band. \mathbf{T}_{ref} is traditionally specified as a second-order transfer function:

$$\mathbf{T}_{\text{ref}} = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad (3.55)$$

The model-matching requirement is translated in the frequency domain with the weighting function \mathbf{W}_{21} , which permits to discriminate frequently the model-matching error $\mathbf{e}_{\text{MM}} = (\mathbf{T}_{\text{ref}} - \mathbf{T}_{\mathbf{r} \rightarrow \mathbf{y}})\mathbf{r}$. In the same manner as \mathbf{W}_{12} , the weighting filter \mathbf{W}_{22} is used to constrain the control input \mathbf{u}_c .

Under these considerations, the performance signals vector \mathbf{z} can be expressed as follows:

$$\mathbf{z} = \begin{bmatrix} z_{11} \\ z_{12} \\ z_{21} \\ z_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{11}(\mathbf{r} - \mathbf{y}) \\ \mathbf{W}_{12}\mathbf{u}_c \\ \mathbf{W}_{21}(\mathbf{T}_{\text{ref}}\mathbf{r} - \mathbf{y}) \\ \mathbf{W}_{22}\mathbf{u}_c \end{bmatrix} \quad (3.56)$$

and the closed-loop transfer function from the exogenous input \mathbf{w} to the performance signal \mathbf{z} can then be expressed under the form:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \mathbf{T}_{\mathbf{w} \rightarrow \mathbf{z}} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{\mathbf{w}_1 \rightarrow z_1} & \mathbf{T}_{\mathbf{w}_2 \rightarrow z_1} \\ \mathbf{T}_{\mathbf{w}_1 \rightarrow z_2} & \mathbf{T}_{\mathbf{w}_2 \rightarrow z_2} \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{r} \end{bmatrix} \quad (3.57)$$

$$= \begin{bmatrix} \mathbf{W}_{11}\mathbf{S}_{\mathbf{d} \rightarrow \mathbf{e}} & \mathbf{W}_{11}(\mathbb{I} - \mathbf{T}_{\mathbf{r} \rightarrow \mathbf{y}}) \\ \mathbf{W}_{12}\mathbf{T}_{\mathbf{d} \rightarrow \mathbf{u}_c} & \mathbf{W}_{12}\mathbf{T}_{\mathbf{r} \rightarrow \mathbf{u}_c} \\ \mathbf{W}_{21}\mathbf{S}_{\mathbf{d} \rightarrow \mathbf{e}} & \mathbf{W}_{21}(\mathbf{T}_{\text{ref}} - \mathbf{T}_{\mathbf{r} \rightarrow \mathbf{y}}) \\ \mathbf{W}_{22}\mathbf{T}_{\mathbf{d} \rightarrow \mathbf{u}_c} & \mathbf{W}_{22}\mathbf{T}_{\mathbf{r} \rightarrow \mathbf{u}_c} \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{r} \end{bmatrix} \quad (3.58)$$

From these expressions, it clearly appears that the off-diagonal transfers $\mathbf{T}_{w_2 \rightarrow z_1}$ and $\mathbf{T}_{w_1 \rightarrow z_2}$ are nonzero. However, these transfer functions are not relevant to the corresponding control objective and hence should not be taken into consideration for synthesis. In order to consider only the diagonal transfers, the 2-DoF synthesis problem can be formulated as the following multi-objective \mathcal{H}_∞ problem: given $\gamma > \gamma_{\min}$, find all stabilizing controllers \mathbf{K} with the structure given in Equations (3.48) – (3.50) such that:

$$\max \{ \|\mathbf{T}_{w_1 \rightarrow z_1}(\mathbf{P}, \mathbf{K})\|_\infty, \|\mathbf{T}_{w_2 \rightarrow z_2}(\mathbf{P}, \mathbf{K})\|_\infty \} < \gamma \quad (3.59)$$

3.3.2.3 Weighting Filter Selection

As mentioned previously, the selection of weighting filters is a tedious process [Kwakernaak, 1993]. In this context, the weighting filters W_{ij} in Figure 3.11 are diagonal arrays of first-order transfer functions with the same form:

$$\mathbf{W}_i^{-1} = \frac{k_{i,\text{HF}}s + k_{i,\text{LF}}\omega_i}{s + \omega_i} \quad (3.60)$$

This formulation enables the designer to directly define the shape of the filter inverse, which corresponds to the desired maximum bound of the closed-loop transfer function of interest. The filter parameters directly define this template, which is illustrated in Figure 3.12: $k_{ij,\text{HF}}$ is the maximum high-frequency gain, $k_{ij,\text{LF}}$ is the maximum low-frequency gain, and ω_{ij} the -3dB inverse filter bandwidth.

The disturbance rejection objective is specified with the weighting filter \mathbf{W}_{11} , which defines the desired shape of the closed-loop sensitivity function $\mathbf{S}_{d \rightarrow e}$. As the system already exhibits an open-loop low-pass behavior, the role of the weighting filter is to enforce low disturbance sensitivity at frequencies near the desired disturbance-free bandwidth specified by ω_{ij} , and to limit the allowed disturbance amplitude with $k_{11,\text{HF}}$. The low-frequency gain $k_{11,\text{LF}}$ sets the steady-state error, but cannot be zero in order to avoid numerical issues during synthesis. Since the open-loop pitch and yaw channels exhibit largely

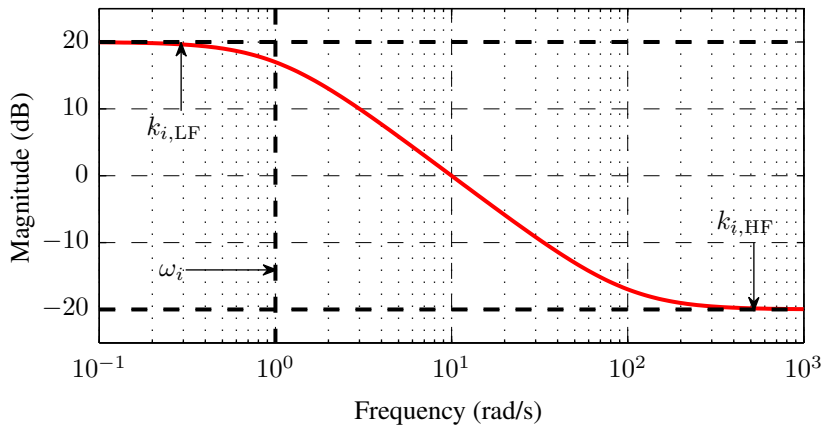


FIGURE 3.12: Inverse of the weighting filter \mathbf{W}_i and associated parameters.

²⁴ Another commonly used form is given by $\mathbf{W}_i^{-1} = \frac{k_{i,\text{HF}}s + \omega_i}{s + \omega_i/k_{i,\text{LF}}}$, for which $|\mathbf{W}_i^{-1}|_{\omega=\infty} = k_{i,\text{HF}}$, $|\mathbf{W}_i^{-1}|_{\omega=0} = k_{i,\text{LF}}$ and $|\mathbf{W}_i^{-1}|_{\omega=\omega_i} = \sqrt{\frac{k_{i,\text{HF}}^2 + 1}{(1/k_{i,\text{LF}})^2 + 1}} \approx 1 = 0 \text{ dB}$ as ω_i corresponds to the gain crossover frequency. This form is implemented in MATLAB's `makeweight` function.

different dynamics and sensitivity to disturbances, the filter parameters are adapted to each channel individually. A classical rule consists in selecting the bandwidth of \mathbf{W}_{11} close to the open-loop system bandwidth. This rule is applied for the yaw channel, but cannot be used for the pitch channel due to the limited actuator bandwidth. For the pitch channel, $\omega_{11,\alpha}$ is selected to be a good trade-off between disturbance rejection and control signal bandwidth.

The reference tracking performance specifications are implemented as a model-following constraint, which aims at shaping the closed-loop transfer function $\mathbf{T}_{r \rightarrow y}$ according to the reference model \mathbf{T}_{ref} . The corresponding weighting filter \mathbf{W}_{21} defines the maximum acceptable model-matching error e_{MM} at each frequency. It attains a maximum at medium frequencies, and the acceptable peak magnitude of the model-matching error is set with the $k_{12,\text{HF}}$ parameter. At higher frequencies, e_{MM} falls off as both the reference model and the closed-loop system exhibit low-pass behavior. The low-frequency parameter $k_{12,\text{LF}}$ controls the steady-state error and most of the time-domain response fit. It is thus set to a small non-zero value, while ω_{12} is set to the reference model bandwidth.²⁵

The actuators have a limited bandwidth and saturation-free amplitude range, while the open-loop system exhibits a flexible mode on the yaw channel. For these reasons, the control signal u_c must be bandwidth- and amplitude-limited so as not to include unmodeled dynamics in the feedback loop. These constraints are imposed using filters \mathbf{W}_{12} and \mathbf{W}_{22} , which aim at limiting the high-frequency gain to the actuators. To this end, the corner frequency ω_{i2} is set at one-third the actuator bandwidth, and the low-frequency gain is limited by parameters $k_{12,\text{LF}}$ and $k_{22,\text{LF}}$. The aforementioned limitations must be respected at all times in the disturbance rejection case, which entirely defines the feedback controller blocks \mathbf{K}_R and \mathbf{K}_S . In the reference tracking case, the actuator input constraints mostly affect \mathbf{K}_F , and the allowed low-frequency control gain $k_{22,\text{LF}}$ may be higher as the injection of higher-frequency dynamics at this point may not jeopardize system stability, the feedback loop being already band-limited.

3.3.2.4 Controller Synthesis Results

The synthesis problem described in the previous section consists in finding the 12 parameter values describing the controller blocks \mathbf{K}_R , \mathbf{K}_S and \mathbf{K}_F . The synthesis constraints are specified by the weighting filters whose parameters are given in Table 3.1, and the desired closed-loop tracking performance is specified by the second-order reference model \mathbf{T}_{ref} as in Equation (3.55) with damping factor $\zeta = 0.78$ (fastest 2% response) and bandwidth $\omega_0 = 3$ rad/s for both channels. Under these constraints, the multi-objective \mathcal{H}_∞ synthesis procedure presented in Section 3.2.4 results in a relatively low final performance level $\gamma = 1.49$, and the obtained controller responses are illustrated in Figure 3.13.

TABLE 3.1: Parameters of the weighting filters W_{ij} .

\mathbf{W}_{11} (α channel)		\mathbf{W}_{11} (β channel)		\mathbf{W}_{12}		\mathbf{W}_{21}		\mathbf{W}_{22}	
$k_{11,\text{LF}}$	0.001	$k_{11,\text{LF}}$	0.001	$k_{12,\text{LF}}$	2	$k_{21,\text{LF}}$	0.001	$k_{22,\text{LF}}$	4
$k_{11,\text{HF}}$	0.8	$k_{11,\text{HF}}$	0.8	$k_{12,\text{HF}}$	0.001	$k_{21,\text{HF}}$	0.1	$k_{22,\text{HF}}$	0.001
ω_{11}	6 rad/s	ω_{11}	2 rad/s	ω_{12}	15 rad/s	ω_{21}	3 rad/s	ω_{22}	15 rad/s

²⁵Generally, constant-valued weighting filters are employed for model-matching. In the present case, a constant filter leads to a high low-frequency model-matching error, which introduces a steady-state error in the tracking response. Hence the chosen filter structure.

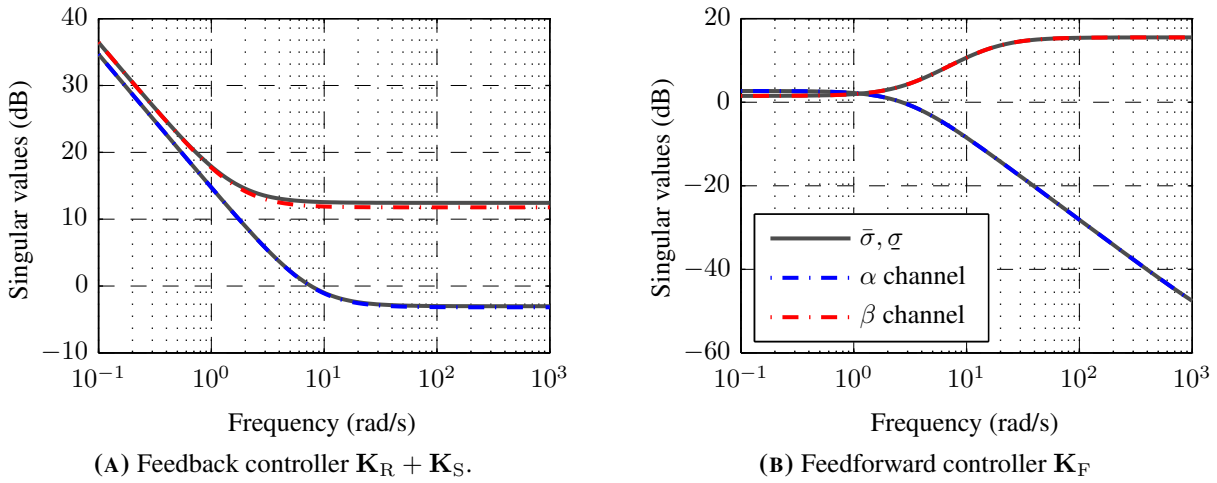


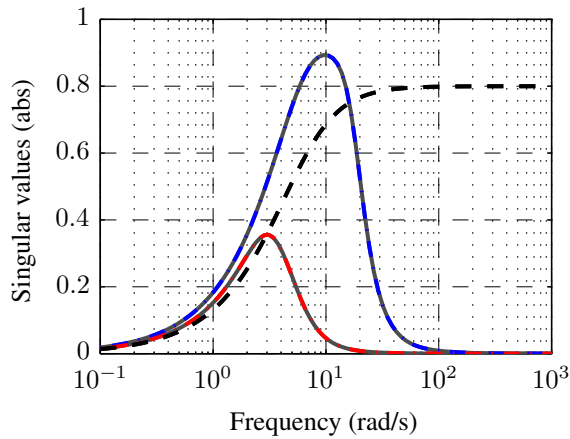
FIGURE 3.13: Singular values of (A) the feedback controller $\mathbf{K}_R + \mathbf{K}_S$ and (B) the feedforward controller \mathbf{K}_F .

The synthesis results are illustrated in Figure 3.14, both in the time domain (Figures 3.14E-3.14F) and in the frequency domain (Figures 3.14A-3.14D). In these plots, the plant model uses the actual modeled delays of Equation (3.34) and not the Padé approximation of Equation (3.35) which was required for the synthesis. The frequency results of Figure 3.14 emphasize on the four transfers which are constrained by the weighting filters \mathbf{W}_{ij} in Figure 3.11. For better understanding these plots, the dashed black line represents the inverse of the selected weighting filter, and the minimum and the maximum singular values $\underline{\sigma}$ and $\bar{\sigma}$ are shown in solid gray lines. Finally, the individual channel transfer functions, i.e. the pitch and yaw channels, are plotted in colored dash-dotted lines. One can observe that the singular value plots and the individual transfer function plots are almost superimposed on each other, proving that the cross-coupling between the two axes is small.

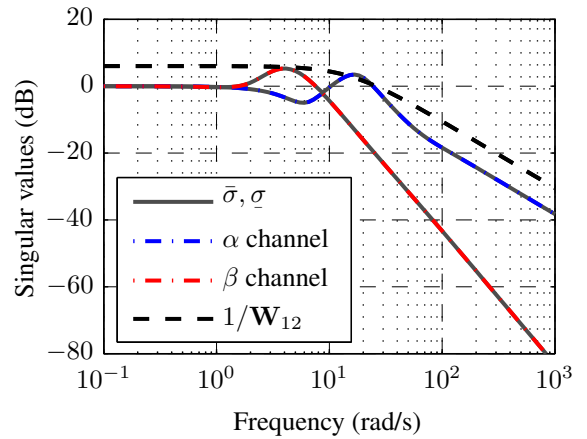
The sensitivity to disturbances $\mathbf{S}_{d \rightarrow e}$ is illustrated in Figure 3.14A. This transfer function has low gain at low frequencies as prescribed and thus constant disturbances are canceled out. This is visible in Figure 3.14E, which shows no steady-state error when a disturbance step is applied. The high-frequency gain is also close to zero, as the disturbance is filtered by the plant transfer function. At frequencies close to the desired bandwidth, this function exhibits a peak, which though exceeds a bit the desired shape for the pitch channel. This peak value is the result of a trade-off between disturbance rejection bandwidth and control input bandwidth.

Regarding the control input $\mathbf{T}_{d \rightarrow u_c}$, shown in Figure 3.14B, both channels conform to the specification and exhibit high-frequency roll-off. The yaw channel has a steeper high-frequency slope due to the additional roll-off filter \mathbf{F}_{r0} . The gain on this channel is -39 dB at the frequency of the observed flexible mode (77 rad/s), hence its dynamics are efficiently attenuated by the feedback loop.

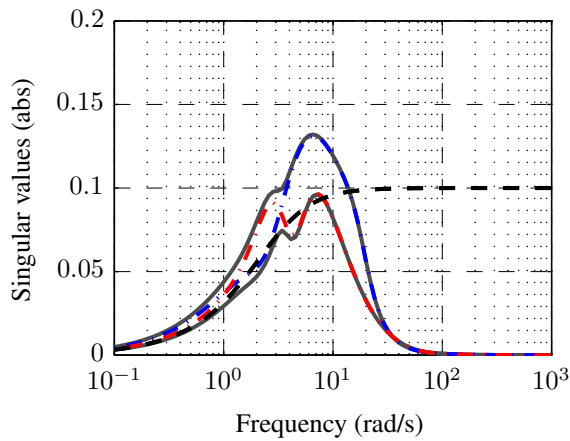
Concerning the reference tracking objective, the transfers which refer to the model-matching error are illustrated in Figure 3.14C. These transfers have zero DC gain, indicating the closed loop transfer function has the expected unity gain, as shown in the step response of Figure 3.14F. The model-matching error exhibits peaks at frequencies close to those of $\mathbf{S}_{d \rightarrow e}$, with a maximum of 0.13 on the pitch channel. This slight difference in mid-frequency behavior is also visible in the pitch step response, which is more damped and exhibits a higher deviation wrt the reference model than the yaw step response.



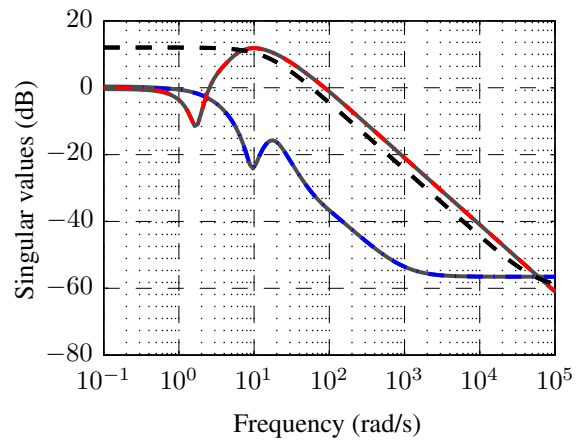
(A) Sensitivity transfers $S_{d \to e}$ and constraint $1/W_{11}$.



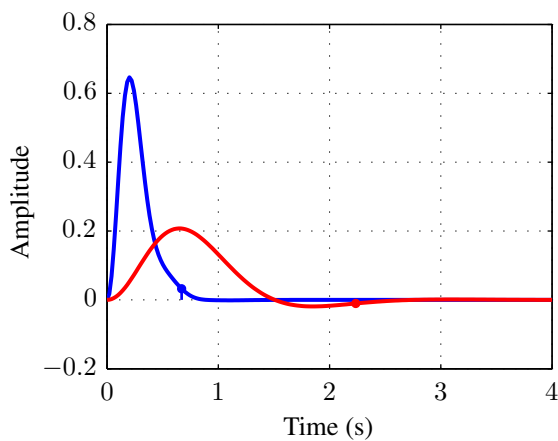
(B) Control sensitivity transfers $T_{d \to u_c}$ and constraint $1/W_{12}$.



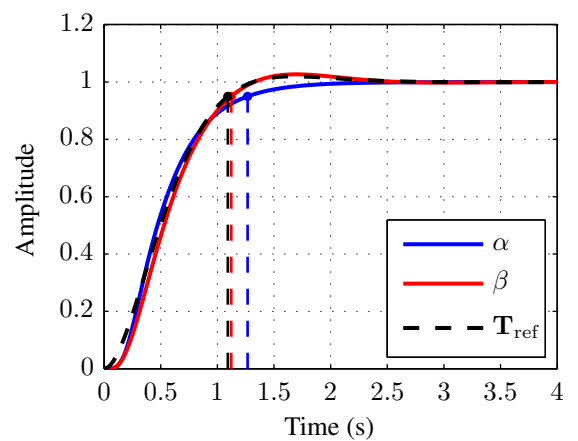
(C) Model-matching transfers $T_{ref} - T_{r \to y}$ and constraint $1/W_{21}$.



(D) Control sensitivity transfers $T_{r \to u_c}$ and constraint $1/W_{22}$.



(E) Response to a unit disturbance step.



(F) Response to a unit reference step.

FIGURE 3.14: Closed-loop transfer functions against inverse weighting filters for (A) the tracking error e and (B) the control input u in the disturbance rejection case, (C) the model-matching error e_{rmMM} and (D) the control input u in the reference tracking case, and unit step responses for (E) the disturbance rejection and (F) the reference tracking cases.

In the reference tracking case, the control input sensitivity $\mathbf{T}_{r \rightarrow u_c}$ includes the dynamics of the feedforward filter \mathbf{K}_{FF} . This filter allows more input energy for the reference tracking objective, since the dynamics in the feedback loop are already limited by \mathbf{W}_{12} . As the yaw channel is slower than the pitch channel by an order of magnitude, its associated feedforward transfer acts as a lead filter and amplifies the reference signal at medium and high frequencies to accelerate the closed-loop response. This is visible on Figure 3.14D with the yaw channel response of $\mathbf{T}_{r \rightarrow u_c}$ being slightly higher than the specified filter shape. Conversely, the feedforward filter associated to the pitch channel acts as a low-pass filter so as to provide the same closed-loop dynamics on both axes, and the associated control input sensitivity is much lower than for the yaw channel.

The time-domain performance of the autopilot is evaluated with the step responses illustrated in Figures 3.14E and 3.14F. Regarding the disturbance rejection objective, an incoming disturbance step does not introduce a steady-state offset. A unit pitch disturbance results in an output deviation less than 0.65° and is rejected (to less than 5% of its peak value) in 0.67 s. A similar disturbance on the yaw axis results in a 0.21° output deviation and is rejected in 2.25 s.

Concerning the reference tracking performance, the step response of both channels is close to the desired second-order reference model \mathbf{T}_{ref} . The 5% response time²⁶ of both channels are given by $T_{r_{5\%},\alpha} = 1.26$ s and $T_{r_{5\%},\beta} = 1.13$ s respectively, while the reference model response time is 1.10 s. Both channels exhibit unity gain as expected, and the closed-loop responses are similar despite the large difference in open-loop dynamics.

3.3.3 Robustness Analysis

A first robustness indicator is given by the classical gain and phase margins, which are calculated on a loop-by-loop basis by opening one loop at a time. In the present case, the loops are broken at the actuator input signal u_c , and the resulting open-loop gain and phase responses are illustrated in Figure 3.15. The pitch channel exhibits a 9.59 dB gain margin and a 41.7° phase margin. Concerning the yaw channel, the gain margin is 12.7 dB and the phase margin is 33.6° . Even though these margins still comply to the specifications, they are much lower than in the single-axis case (see Section B.4) due to the time delays now being included in the nominal plant.

These margins only consider either a gain **or** phase uncertainty on a single loop. A more reliable tool for MIMO systems is the multivariable gain and phase margins introduced in Section 3.2.5, which considers uncertainty on multiple loops and takes couplings into account. Applying these definitions to the open-loop MIMO plant then leads to a multivariable gain margin $GM_{ST} = [-5.42 \text{ dB}, 5.42 \text{ dB}]$ and phase margin $PM_{ST} = [-33.6^\circ, 33.6^\circ]$. These margins are lower than the loop-by-loop gain and phase margins, and the multivariable gain margin is slightly under specification. However, the definitions of these margins assume a full uncertainty matrix, whereas gain and phase uncertainty in the individual loops of a MIMO system correspond to diagonal (structured) uncertainty, hence introducing conservatism. As a result, these values are minimum bounds of the actual gain and phase margins. The combination of the classical SISO margins and these MIMO margins presumes then *a priori* sufficient stability of the nominal closed-loop system.

²⁶Although \mathbf{T}_{ref} is specified with the optimal 2% damping ratio $\zeta = 0.78$, the 5% response time is used for evaluation as it better reflects the similarity between the pitch and yaw step responses.

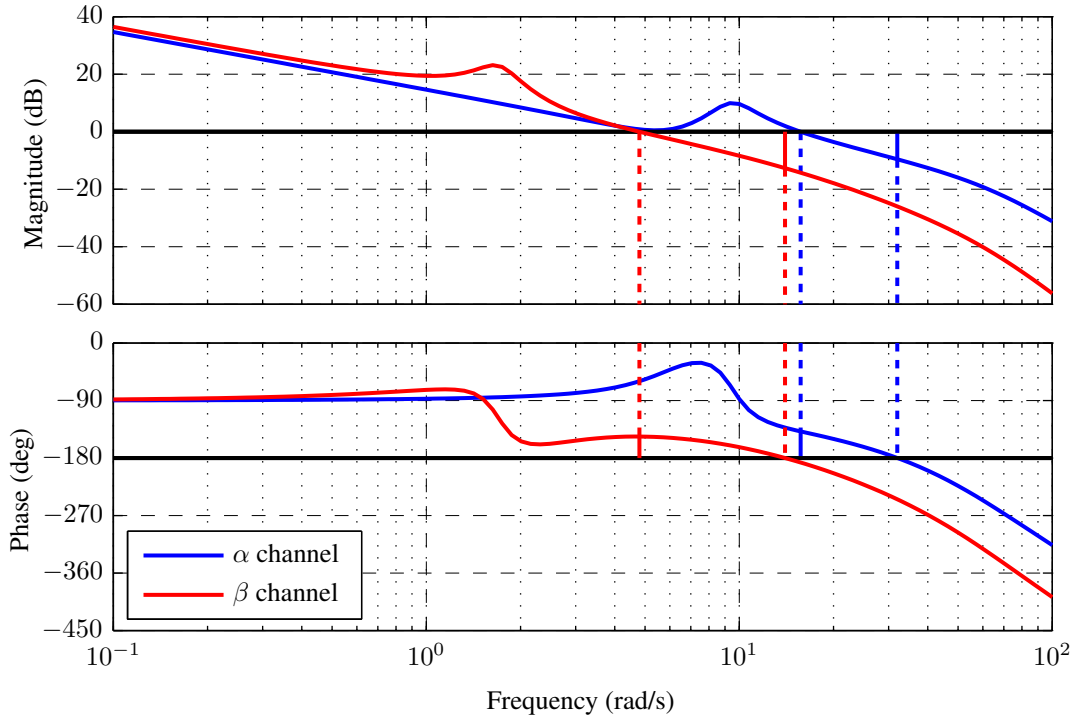


FIGURE 3.15: Loop-at-a-time classical gain and phase margins for the skid-to-turn autopilot.

To assess the robust stability of the closed-loop *w.r.t.* the modeled uncertainty, the closed-loop system is written using the uncertain airframe model given in Equation (3.36) and is cast into the $\mathbf{M} - \Delta$ configuration in Figure 3.16. In order to consider the diagonal structure of the Δ matrix in the analysis, the structured singular value robust stability criterion of Equation (3.33) is employed.

In the present case, the upper and lower μ bounds were computed using the MATLAB `mussv` function. This function operates on the pointwise frequency response of \mathbf{M} , where the chosen frequency vector contains 250 points from 10^{-2} rad/s to 10^3 rad/s for good precision near the system bandwidth. In order to obtain accurate, tight upper and lower bounds, an LMI solver is used for computing the upper bound, and a gain-based algorithm is used for the lower bound with reinitialization of computation at

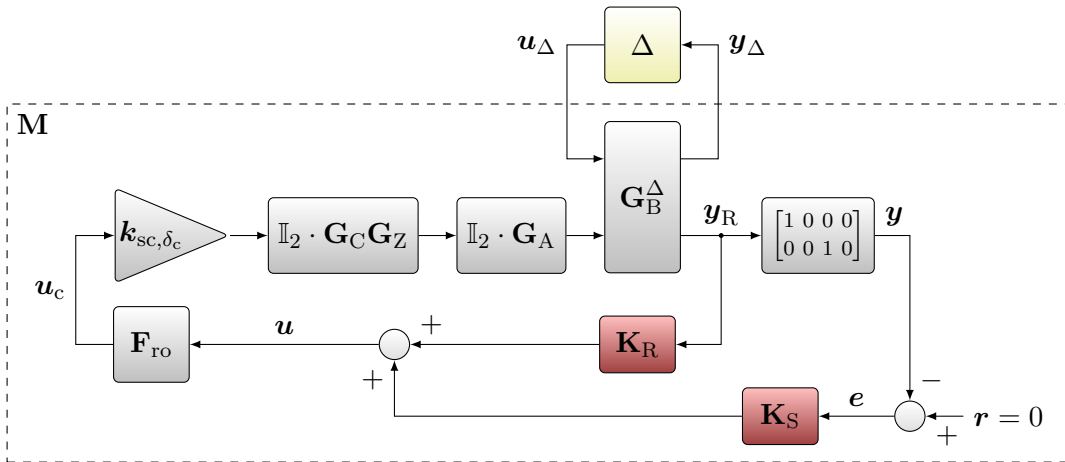


FIGURE 3.16: Pitch-yaw autopilot in the $\mathbf{M} - \Delta$ configuration.

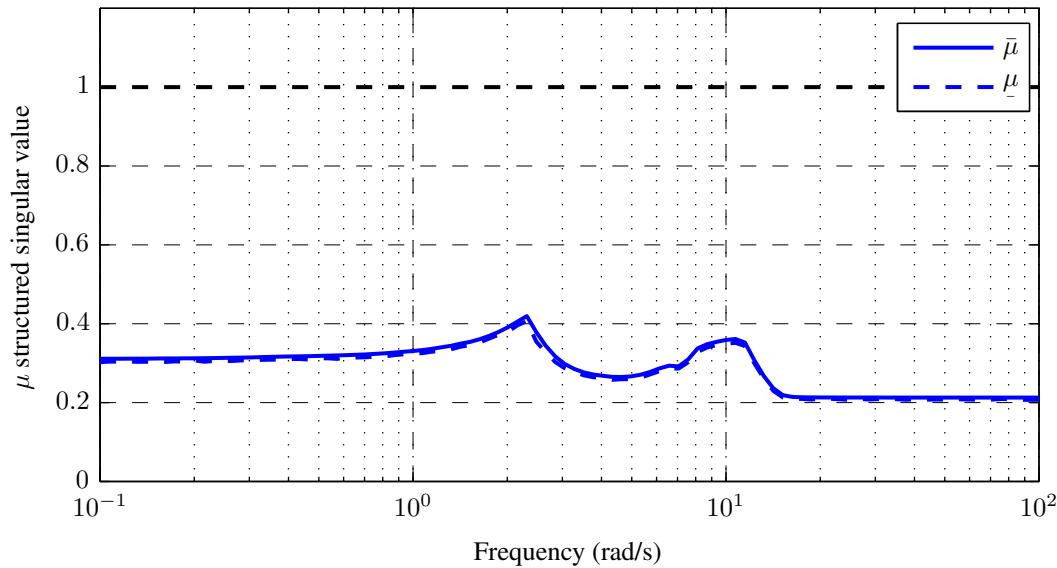


FIGURE 3.17: Upper and lower bounds of the structured singular value μ for the robust stability transfer M .

each frequency point. Using these options, the lower bound computation does not present convergence issues hence it is not necessary to introduce 'fictive' complex uncertainty in Δ . The resulting upper and lower μ bounds are tight, as shown in Figure 3.17. As the upper bound remains below 1 at all considered frequencies, the system is robustly stable and can tolerate up to 250% of the modeled uncertainty. Given that the robustness margins and the simulation results are very good, the controller may then be implemented on the projectile prototype.

3.3.4 Experimental Validation

In this section, the proposed autopilot is implemented both for disturbance rejection and reference tracking on the ACHILES test bench, and several experiments are conducted to assess its real-world performance, with comparisons to numerical simulations. The two control objectives are first assessed for each axis taken independently with the other axis remaining at rest. A second validation step evaluates the reference tracking objective for inputs combining the pitch and yaw axes, in order to assess the rejection of cross-coupling effects.

3.3.4.1 Controller Discretization

The embedded computer and software framework presented in Chapter 1 results in a discrete-time control implementation, while the plant to be controlled is a physical, continuous-time process. Digital control systems for such plants can be represented as the block diagram shown in Figure 3.18. Since the plant is a continuous-time analog process whereas the controller is a discrete-time digital system, these two systems are interfaced using analog-to-digital (A/D) and digital-to-analog converters (D/A). These blocks also contain sample-and-hold circuits for interfacing the continuous and discrete domains. Here, the A/D function is performed within the IMU and external sensors, and the D/A function is performed within the actuators.

The discrete-time controller $\mathbf{K}(z)$ can be designed either directly in the discrete domain or by approximation of a continuous-time controller. The former first involves the discretization of the contin-

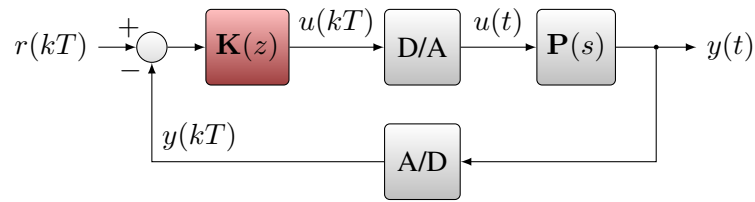


FIGURE 3.18: Typical digital control system.

uous plant $P(s)$. The controller is then designed using discrete-time control techniques. The second approach, called *design by emulation* and used here, consists in finding the discrete equivalent of a continuous-time controller, using a short sample period and discrete-time approximations of the continuous controller [Åström and Wittenmark, 2011]. This latter approach is particularly interesting, as it allows the continuous-time controllers to be implemented on the experimental setup in a straightforward way with only minor modifications. Several methods exist for approximating a continuous controller with a discrete equivalent, such as discrete derivative approximations (Euler’s method, the backward difference, or the bilinear transformation), the matched pole-zero technique, and step and ramp invariance approximations [Franklin et al., 1998].

Besides the discretization of the controller, there are several other considerations to take into account for the implementation of the control law on the real system:

- **Computation delays** due to the control law computation and input/output processing.
- **Antialias** filtering of sampled signals so as to respect the Shannon-Nyquist theorem. The high-frequency content of controller input signals must be filtered to avoid aliasing issues. Even though the plant generally acts as a low-pass filter, the measurement noise may not be bandlimited. In this case an additional antialias filter is required before the A/D converter.
- **Integrator windup** induced by magnitude and/or rate-of-change saturations in the real-world plant. Windup corresponds to a continuous increase of the controller states even though the control input cannot be further increased, which takes considerable time to unwind as the control error changes sign, hence causing large degradations of the transient behavior and even instability. To avoid this behavior, *antiwindup* techniques consist in limiting the state of the controller so that the controller output does not increase past the actuator limits.
- **Quantization errors** due to the finite resolution of the A/D and D/A converters.
- **Numerical errors** in floating-point implementations.

Concerning the ACHILES case, the selection of the sampling period T is constrained by the system dynamics, the bandwidth of the communication channel between the IMU and the embedded computer, the available computing power and the latency of the various communication channels. Furthermore, the design by emulation technique assumes that the sampling frequency is larger than the system bandwidth. The maximum open-loop bandwidth is $\omega_n = 12.96$ rad/s on the pitch channel at $V = 40$ m/s, while there is a flexible mode at $\omega_{flex} = 77$ rad/s on the yaw channel. Hence, choosing $T = 10$ ms gives sufficient computing and communication margins, and yields a sampling frequency 8 times higher than the fastest dynamics in the open-loop system.

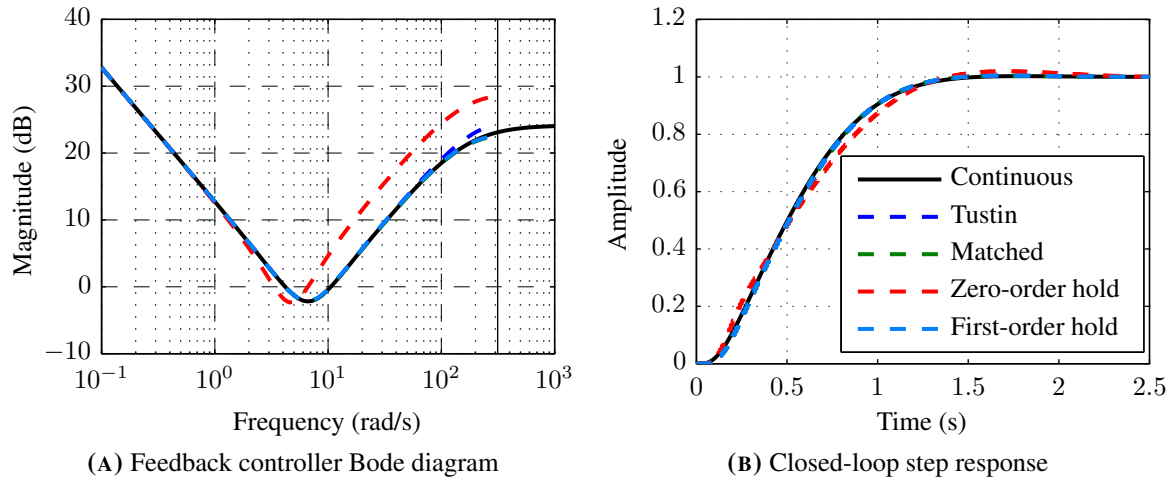


FIGURE 3.19: Bode diagram of the feedback multi-objective controller using different discretization methods and corresponding closed-loop step responses, for the pitch-only case.

Given this sampling period, discrete-time approximations are compared in a simple case, which consists in the discretization of the multi-objective pitch-only feedback controller discussed in Appendix B. These results are presented in Figure 3.19, in which the zero-order hold produces a poorly-performing discrete feedback controller while the other methods yield very similar performance to the continuous-time controller. In the present case, Tustin's method has been retained for the discretization of the feedback controller. The same discussion holds for the discretization of the feedforward controller, as the autopilot may be included in a guidance loop. In the present case, stepwise reference signals will be considered for performance evaluation and hence the zero-order hold approximation is suitable.

The input acquisition latency in the ACHILES setup is variable and may take up to 2 ms. In order to avoid introducing a variable control delay, the outputs are delayed by a full period in the digital controller implementation. The resulting delays have been taken into account during the synthesis of the pitch-yaw controllers. Regarding the windup issue, the ACHILES actuator presents a rate limitation for step inputs larger than 60° and its magnitude is limited to $\pm 30^\circ$ to avoid drag. The maximal achievable fin deflection, however, is limited by the loss of control authority at the edges of the flight envelope. The controllers designed in Chapter 3 do not integrate explicit *antiwindup* schemes as these saturations are not reached during normal operation.

3.3.4.2 Disturbance Rejection

The experimental validation of the disturbance rejection objective consists in evaluating the peak deviation, the response time and the presence of a steady-state error consecutive to a disturbance step when the system is at equilibrium. However, in the general case, the disturbance input may not be measurable and/or may not be influenced (see input $d_{(\text{real})}$ in Figure 3.20). In order to assess the disturbance rejection performance, an artificial disturbance step $d_{(\text{simulated})}$ of known amplitude is injected at the actuator input, as illustrated in Figure 3.20. Nevertheless, it must be remembered that the system can also be affected by the real disturbances during these experiments.

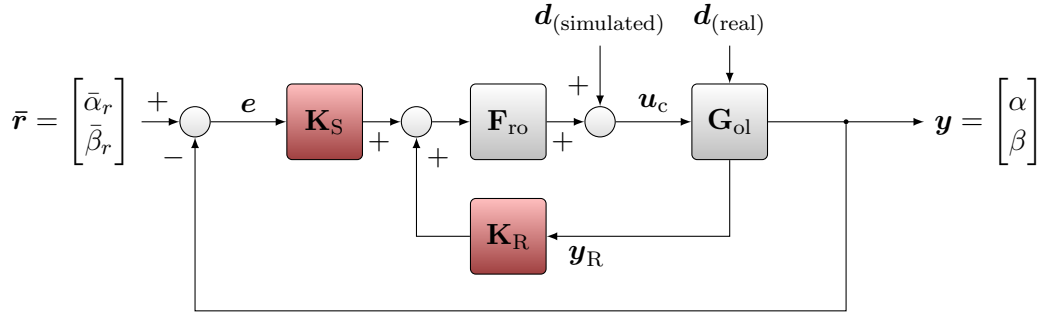


FIGURE 3.20: Injection point of the simulated disturbance signal. The feedforward block is omitted for clarity since the reference inputs remain constant.

The disturbance rejection performance of the fixed-air-speed skid-to-turn autopilot is illustrated in Figure 3.21 for the pitch and yaw channels. To this end, the rejection is evaluated for different operating points (shown as thin lines in Figure 3.21) and only the variations α_δ and β_δ of α and β *w.r.t.* these operating points are represented to make comparisons easier. The observed disturbances are the result of a unit step disturbance applied to the control surfaces.

As expected from the synthesis results, the pitch channel is more sensitive to disturbances than the yaw channel. Indeed, the experimental results exhibit visible oscillations, which are due to aerodynamic interactions between the support rods and the tail fins, flow unsteadiness around the control and tail surfaces as well as backlash in the actuator linkage. The highest level of perturbations is reached at the zero-incidence operating point ($\bar{\alpha} = 0^\circ$, $\bar{\beta} = 0^\circ$), illustrated in dark red in Figure 3.21. At these values of the incidence angles, the control fins are almost aligned with the support structure, resulting in high flow interaction. The yaw axis is less sensitive to aerodynamic disturbances as its inertia is much higher and the outer gimbal is kept out of the airflow by design.

Experimental results for both channels conform to the simulations, and the observed performance is similar to synthesis results, with a 20% attenuation in less than 1 s of disturbance steps entering the pitch channel, and a 80% attenuation in 2.5 s of disturbance steps entering the yaw channel. Both output signals shows no steady-state error, confirming that the low frequencies of the disturbance signal are effectively rejected.

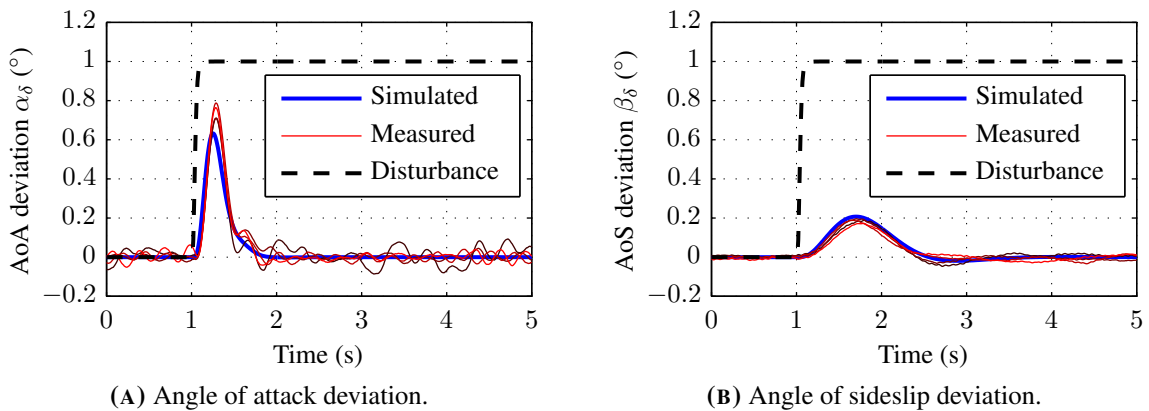


FIGURE 3.21: Disturbance rejection performance of the skid-to-turn autopilot for $V = 25$ m/s at operating points $(\bar{\alpha}, \bar{\beta}) = (0^\circ, 0^\circ), (0^\circ, 3^\circ), (2^\circ, 3^\circ)$ and $(2^\circ, 0^\circ)$.

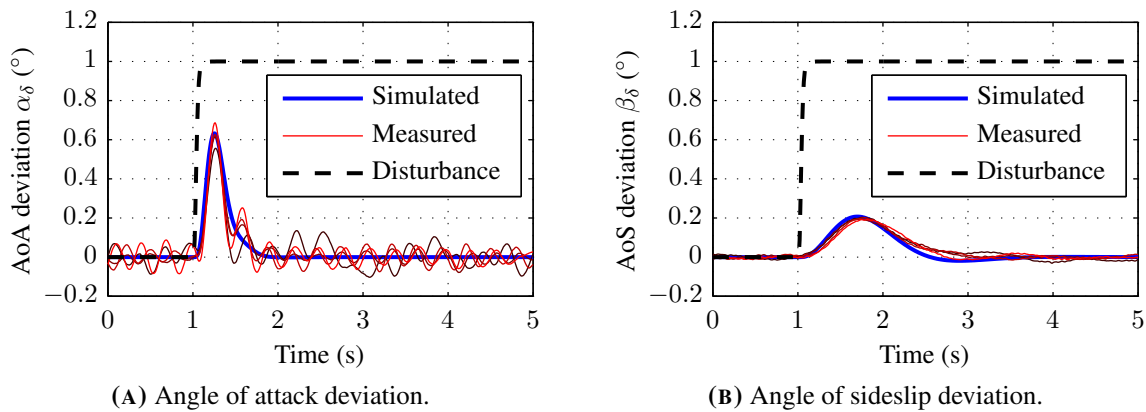


FIGURE 3.22: Disturbance rejection performance of the skid-to-turn autopilot, without taking delays into account during synthesis, under the same conditions as in Figure 3.21.

Compared to the pitch-only autopilot, the delays introduced by the discrete implementations were actually taken into account during the design of the skid-to-turn autopilot. The influence of these delays is illustrated in Figure 3.22, using controllers designed without taking them into account. This results in a degradation of the disturbance rejection performance as well as reduced stability margins, as shown by the level of oscillations in the pitch response. Even though these controllers showed sufficient gain and phase margins with the delay-free nominal plant, the actual margins are lower due to the implementation delays introducing significant phase lag.

3.3.4.3 Reference Tracking

The reference tracking performance is first evaluated separately for the pitch and yaw channels, with the other channel kept at a zero reference angle. The measured, simulated and reference outputs are illustrated in Figure 3.23, together with the associated control signals. Although the system no longer operates in the neighborhood of operating points during transients, a very good agreement between experimental and simulated responses can be observed, with only minor overshoot on the yaw channel. Both channels exhibit similar closed-loop reference-tracking dynamics as specified during synthesis, with a response time $t_{r_{5\%}}$ around 1 s, even though their open-loop bandwidths are separated by an order of magnitude. Additionally, the control inputs have similar shapes between the experimental and simulated controllers. The differences in amplitude are due to different DC gains between the nominal model and the actual system, which are compensated by the feedback controller.

In the pitch-only case, it was shown that the absence of the feedforward filter resulted in actuator saturation and a severe degradation of the reference tracking performance (see Figure B.18). The same experiment is carried out for the skid-to-turn autopilot by removing the injection-form feedforward block. This case is illustrated in Figure 3.24, with the nominal response shown in dotted blue. Compared to the pitch-only case, the control input does not show the high-amplitude peaks observed in Figure B.18B, and the absence of the feedforward filter only results in a faster pitch response and a slower, less damped yaw response. In the present autopilot structure, the high-frequency content of the tracking error signal is filtered by the servo controller, which consists in a diagonal integrator. Hence, the feedforward is not as crucial as in the pitch-only case, and its role is only to shape the reference signals so as to meet the reference model. Here, the feedforward acts as a lag filter on the pitch channel to slow down its response, and as a lead filter on the yaw channel to accelerate its response.

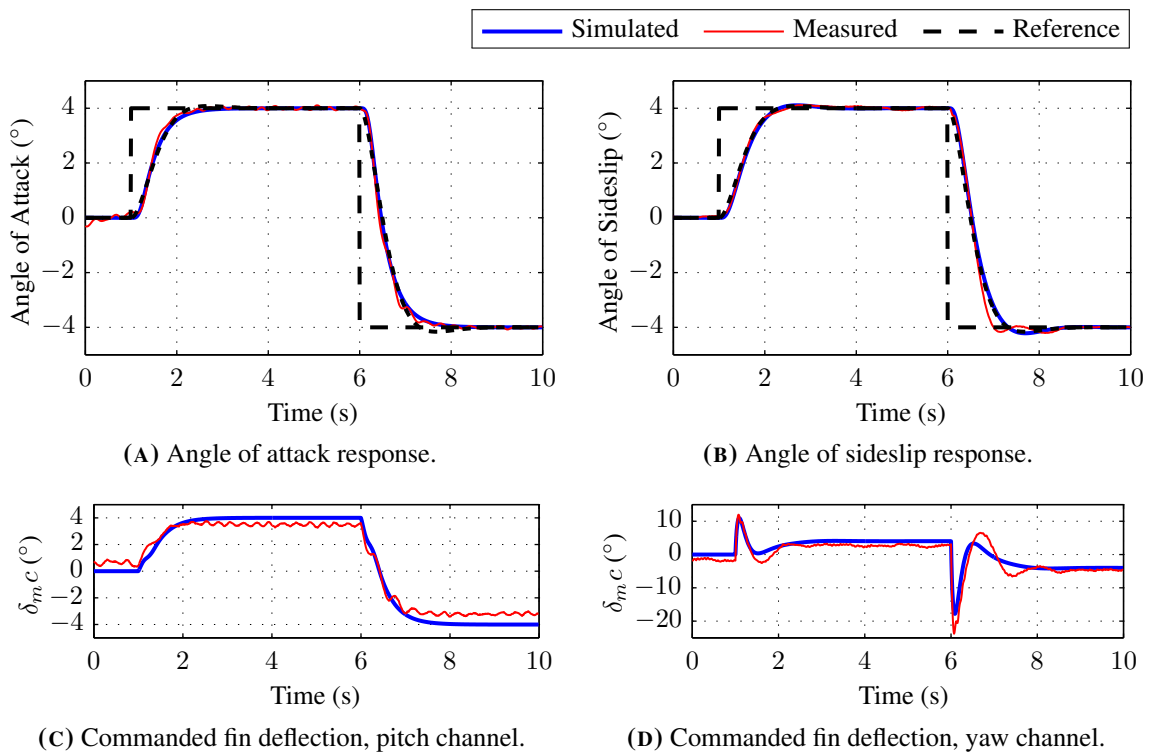


FIGURE 3.23: Per-channel reference tracking performance of the skid-to-turn autopilot for $V = 25$ m/s. The reference model T_{ref} response is shown in dotted black.

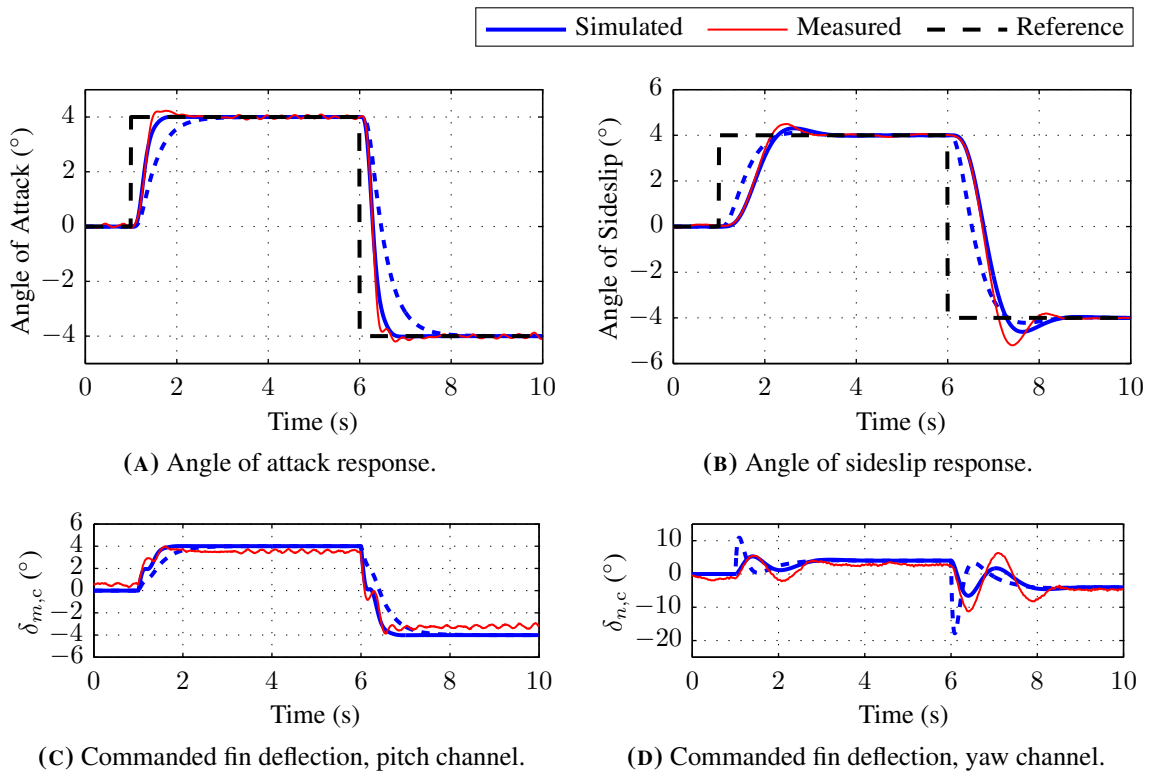
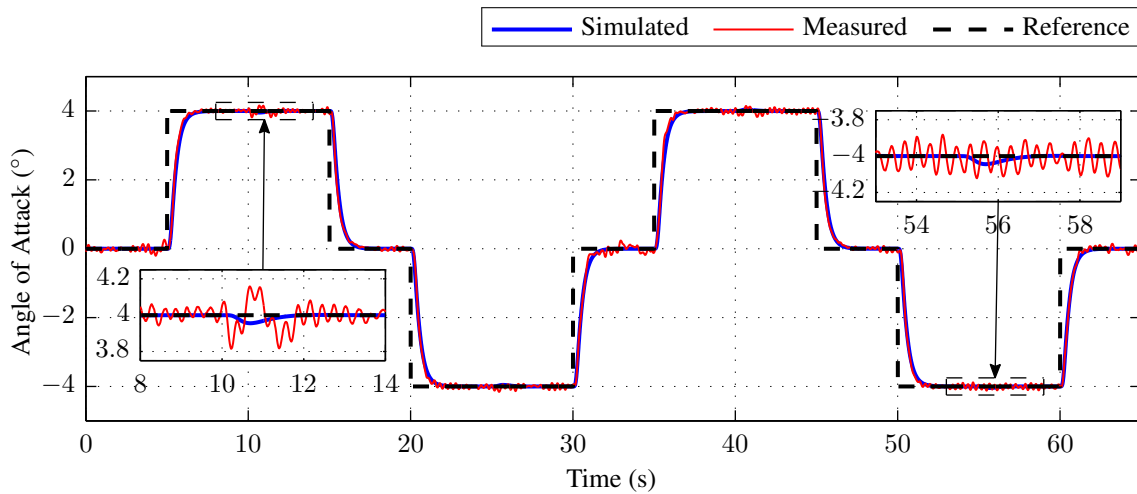
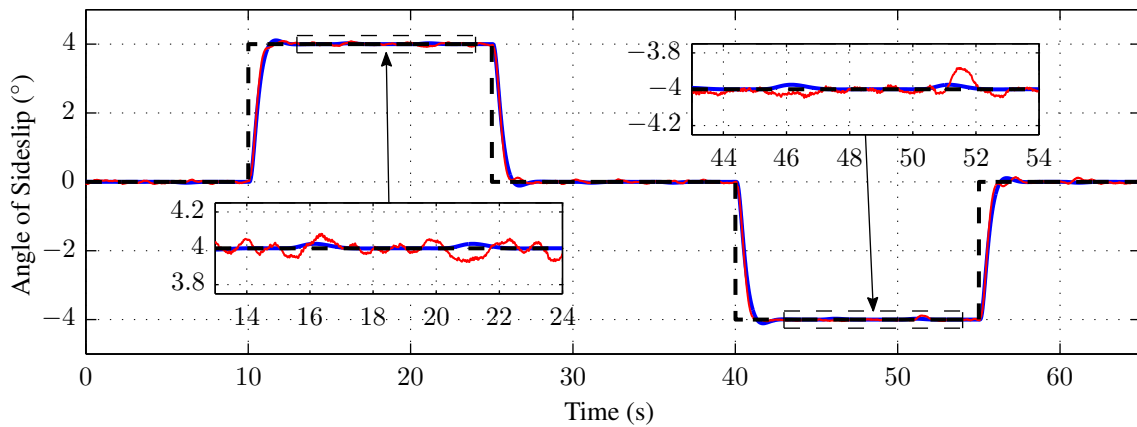


FIGURE 3.24: Per-channel reference tracking performance of the skid-to-turn autopilot without the feedforward controller for $V = 25$ m/s. The nominal response (with feedforward) is shown in dotted blue for reference.



(A) Angle of attack response.



(B) Angle of sideslip response.

FIGURE 3.25: Reference tracking performance for both channels operating simultaneously.

The cross-channel coupling effects are finally investigated in Figure 3.25 using large-amplitude steps on both channels to exercise a large part of the flight envelope. The off-axis disturbances are well rejected: as shown in the insets, there is only a barely noticeable change in the output signal when the input to the other channel changes.

3.3.5 Concluding Remarks

In this section, the development of a skid-to-turn autopilot for the ACHILES prototype has been presented for a constant airspeed of 25 m/s, and takes into account the pitch and yaw dynamics of the projectile.

The autopilot structure is based on an output-feedback regulator, augmented with a tracking integrator and an injection feedforward filter. The parameters of this structure are computed using a mixed-sensitivity, multi-objective structured \mathcal{H}_∞ synthesis technique with disturbance rejection and model-matching reference tracking constraints. The synthesized controllers perform according to specifications and despite large differences in open-loop dynamics between the pitch and yaw channels, both channels

offer similar closed-loop reference tracking performance. The robustness of the control loop with respect to the airframe model parameter uncertainty is then assessed using the structured singular value μ , and results in a large robustness margin.

In a final step, the autopilot is implemented on the experimental setup, and several exhaustive validation tests are conducted to assess the performance of the proposed solution against numerical simulations, in which the actual projectile follows the simulated responses closely and in accordance to the design requirements. The disturbance rejection results show a higher sensitivity to aerodynamic disturbances on the pitch channel than on the yaw channel, but in both cases, a unit disturbance does not introduce a steady-state error. The importance of taking the control and sample/hold delay into account was also illustrated. In terms of reference tracking, both channels exhibit similar responses despite the large difference in open-loop dynamics. These results also show that the cross-axes coupling effects are well rejected.

3.4 Variable-Airspeed Autopilot

This second control application consists in the extension of the skid-to-turn autopilot to the complete flight domain. In this context, angular motion is allowed on both the pitch and yaw axes, and the airspeed V can vary between 15 m/s and 40 m/s. The linearized dynamics model described in Equations (2.76) - (2.77) depends on the operating condition $\rho = [\alpha \ \beta \ \sigma^T]^T$, which results in a *quasi*-LPV structure. Indeed, the linearized dynamics depend both on part of the state vector, via α and β , and on the external parameters $\sigma = [V \ h]^T$, i.e. the airspeed and the altitude. In this application, the altitude is not considered as a varying parameter as it remains constant in the subsonic wind tunnel.

The previous section considered a constant airspeed $V = 25$ m/s, and dynamics variations induced by the incidence angles α and β were taken care of with a single robust controller. In order to take into account the model variations induced by the airspeed V , the single robust controller approach may not be applicable. As the controller must guarantee closed-loop stability and sufficient margins with respect to the large variations in open-loop dynamics, the system performance specifications would have to be severely degraded.

In order to keep an acceptable performance level, the controller dynamics must be adapted with respect to the airspeed. As mentioned in the introduction, nonlinear or adaptive control techniques such as model predictive control or dynamic model inversion usually involve high computation costs, which are prohibitive for low-cost embedded computers. Another widely used class of techniques is gain scheduling, which is more suited to the present case.

3.4.1 Gain-Scheduling Strategy

Gain scheduling is a widely acknowledged control strategy for non-linear systems exhibiting fast variations of their dynamics with respect to time or operating conditions. This technique has found successful application in a broad range of fields, especially for aeronautical systems such as aircrafts, space vehicles and missiles [Apkarian et al., 1995; Jackson, 2010]. The dynamics of these systems are indeed highly dependent on their operating conditions, such as the altitude, Mach number and incidence angles. In these applications, a single robust controller would often not be able to guarantee good closed-loop performance and robustness margins throughout the complete flight envelope.

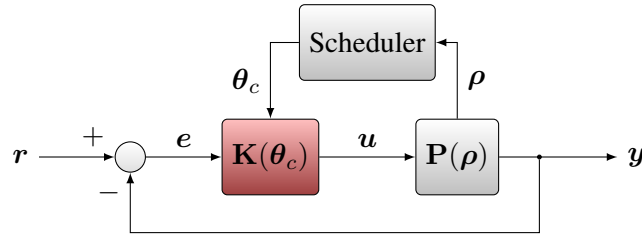


FIGURE 3.26: Gain-scheduling control configuration.

Gain-scheduling control schemes, as illustrated in Figure 3.26, consist in updating the controller parameters θ_c with respect to *scheduling variables* ρ . These variables are responsible for the parameter dependence of the system and can consist in external parameters, time or state variables. The gain-scheduling approach is most suitable to systems where the scheduling variables are slowly-varying [Shamma and Athans, 1992].

The most common gain-scheduling approaches can be classified into two categories: linearization-based gain scheduling (LBGS) and LPV-based gain scheduling, which are discussed thereafter. These techniques and other less common techniques such as neural/fuzzy gain scheduling and velocity-based scheduling are also discussed in greater detail in Theodoulis [2008]; Rugh and Shamma [2000]; Leith and Leithead [2000]. In both LBGS and LPV methods, the plant is represented as a linear parameter-varying system, where the state-space matrices are functions of a time-varying parameter vector $\rho(t)$:

$$\mathbf{P} : \begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(\rho(t))\mathbf{x}(t) + \mathbf{B}(\rho(t))\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}(\rho(t))\mathbf{x}(t) + \mathbf{D}(\rho(t))\mathbf{u}(t) \end{aligned} \quad (3.61)$$

The parameter vector ρ may also include some states of the system, in which case the model is said to be *quasi*-LPV. An LPV system can be reduced to a linear time-varying (LTV) system for a given trajectory $\rho = \rho(t)$ and to an LTI system for a constant trajectory $\rho = \rho_0$ (frozen-parameters).

LPV gain scheduling involves the direct synthesis of an LPV controller [Poussot-Vassal, 2008; Bianic, 1996]. The main advantages of these techniques are the availability of a theoretical guarantee of stability and performance for all values of the scheduling variables given a corresponding range and rate of variation of the aforementioned variables. The controller design is global with respect to the operating domain, whereas LBGS techniques focus on local properties. However, these control techniques are more restrictive than classical gain scheduling design which allows any linear control methodology. The LPV reformulation also introduces a degree of conservativeness in order to arrive to a feasible and convex problem.

3.4.1.1 Linearization-based gain scheduling

Linearization-based gain scheduling, also referred to as classical gain scheduling or divide-and-conquer, consists in constructing a nonlinear controller by combining the members of an appropriate family of LTI controllers. The associated design procedure involves the following steps:

- Divide the operating domain Γ using an appropriate gridding of synthesis points $\rho_{\text{eq},i}$, $i = 1 \dots n$. Figure 3.27 illustrates an example gridding for a planar operating domain with scheduling vector $\rho = [\rho_1 \quad \rho_2]^\top$.

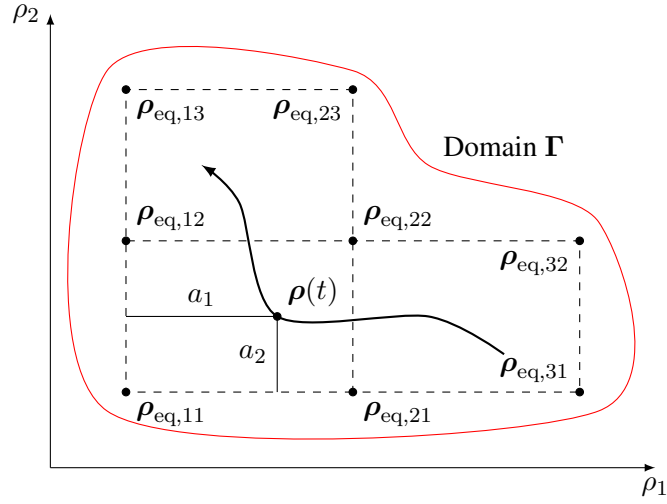


FIGURE 3.27: Operating domain gridding with synthesis points (2D case).

- Find the plant equilibrium conditions corresponding to every value of the scheduling variables $\rho_{\text{eq},i}$ inside the operating domain Γ .
- Linearize the nonlinear plant dynamics around the equilibrium conditions. For each considered value of the scheduling variables $\rho_{\text{eq},i}$, the linearization results in an LTI model describing the system behavior around its equilibrium point. Hence, the collection of scheduling variables $\rho_{\text{eq},i}$ results in a family Π of linear models \mathbf{P}_i .
- For each member \mathbf{P}_i of the family, design an appropriate controller \mathbf{K}_i ensuring the stability, performance and robustness requirements for the local plant. These controllers are of the generic form:

$$\mathbf{K}_i : \begin{cases} \dot{\mathbf{x}}_k(t) = \mathbf{A}_k(\rho_{\text{eq},i})\mathbf{x}_k(t) + \mathbf{B}_k(\rho_{\text{eq},i})\mathbf{e}_k(t) \\ \mathbf{u}_k(t) = \mathbf{C}_k(\rho_{\text{eq},i})\mathbf{x}_k(t) + \mathbf{D}_k(\rho_{\text{eq},i})\mathbf{e}_k(t) \end{cases} \quad (3.62)$$

One of the main advantages of this technique is that there are no constraints on the generic controller structure, or the synthesis technique used to obtain \mathbf{K}_i .

- For any point in the operating domain (not only those considered for synthesis), the local control law is obtained by interpolation of the family of LTI controllers \mathbf{K}_i with respect to the scheduling variables ρ .

In practice, simple ad-hoc methods such as gain blending or controller blending are sufficient in most applications. Gain blending requires a constant structure across the family of linear controllers, and consists in interpolating the controller parameters (gains, poles and zeros) with respect to the scheduling vector. In the controller blending approach, the complete set of controllers is implemented and adjacent controller outputs are blended together using a weighted sum *w.r.t.* the scheduling vector. The practical implementation of this method requires the addition of output tracking or controller reset signals in order to provide bumpless transfer between controllers [Hanus et al., 1987; Bates and Postlethwaite, 2002].

However, in general, no stability, performance and robustness guarantees can be given between the synthesis points for these ad-hoc methods. The validation of these controllers is then performed through

extensive Monte Carlo simulations using a nonlinear or LPV model of the system. More theoretically justified methods are exposed in the literature [Lawrence and Rugh, 1995] but their practical implementation is more difficult. Due to the linearization approach, classical gain-scheduling techniques are restricted to near-equilibrium operation and slowly-varying scheduling variables.

3.4.1.2 Application to the ACHILES case

In the present case, the variable-air-speed autopilot is designed using a divide-and-conquer gain-scheduling technique. This class of methods is particularly suited to the modeling and identification approaches presented in Chapter 2, where the non-linear behavior of the ACHILES projectile was approximated by families of LTI models covering the complete flight envelope. LPV gain scheduling has not been considered in this work, since the associated LPV transformation and controller design techniques are more complex and result in more conservative solutions. Furthermore, LPV identification techniques are not yet as mature as LTI identification techniques.

In the identification study, the flight envelope has been covered using 6 values of the airspeed, $V = \{V_1 V_2 \cdots V_6\}$, ranging from 15 m/s to 40 m/s in 5 m/s steps. For each considered airspeed V_i , a family of models $\Pi_i = \{\mathbf{P}_{i,j}\}$ has been estimated, where the j index corresponds to the incidence angles combinations summarized in Table 2.1²⁷. The observation of the estimation results showed that the projectile dynamics are highly dependent of the airspeed, while they exhibit a low to moderate dependency to the incidence angles (see Figure 2.13). Hence, the airspeed V constitutes a suitable controller scheduling variable, while the incidence dependency is treated in the same way as Section 3.3 using a robust approach.

This consideration effectively reduces the *quasi*-LPV model of Equations (2.76) - (2.77) to an LPV model with a single external parameter (V). For each considered airspeed value, a fixed-air-speed controller is designed using the approaches presented in Sections B and 3.3. These steps are discussed in the following paragraphs, highlighting the main differences with the fixed-air-speed cases. The resulting controllers are then interpolated using a gain-blending implementation, which is further discussed in Section 3.4.3.

3.4.2 Local Controller Synthesis

Based on the structure presented in Figure 3.6, the open-loop model considered for the synthesis of the individual controllers is illustrated in Figure 3.28. The main difference with the fixed-air-speed case is that the airframe model \mathbf{G}_B now exhibits a dependence *w.r.t.* the airspeed V . The scaling gains k_{sc,δ_c} and $k_{sc,d}$ are also adjusted *w.r.t.* V so that the plant keeps unity steady-state gain. The actuator model and the delays induced by the implementation are identical to the fixed-air-speed case. The open-loop dynamics at synthesis points V_i are then given by:

$$\mathbf{G}_{ol,i} : \begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \\ \mathbf{y}_R \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_{1,i} & \mathbf{B}_{2,i} \\ \mathbf{C}_1 & 0 & 0 \\ \mathbf{C}_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{d} \\ \mathbf{u} \end{bmatrix} \quad (3.63)$$

²⁷At low airspeeds, some of these combinations are ignored since outside the flight envelope

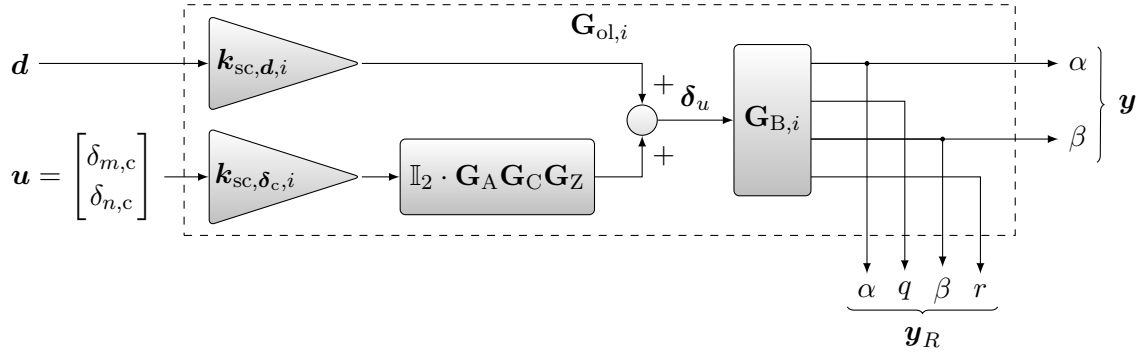


FIGURE 3.28: Open-loop plant model in the variable-air speed case.

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{A}_{B,i} & \mathbf{B}_{B,i} \otimes \mathbf{C}_A & 0 & 0 \\ 0 & \mathbb{I}_2 \otimes \mathbf{A}_A & \mathbb{I}_2 \otimes \mathbf{B}_A \mathbf{C}_C & \mathbb{I}_2 \otimes \mathbf{B}_A \mathbf{D}_C \mathbf{C}_Z \\ 0 & 0 & \mathbb{I}_2 \otimes \mathbf{A}_C & \mathbb{I}_2 \otimes \mathbf{B}_C \mathbf{C}_Z \\ 0 & 0 & 0 & \mathbb{I}_2 \otimes \mathbf{A}_Z \end{bmatrix} \quad (3.64)$$

$$\mathbf{B}_{1,i} = \begin{bmatrix} \mathbf{B}_{B,i} \mathbf{k}_{sc,d,i} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{B}_{2,i} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbb{I}_2 \otimes \mathbf{k}_{sc,\delta_c,i} \end{bmatrix} \quad (3.65)$$

$$\mathbf{C}_1 = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} & \mathbb{O}_{2 \times 10} \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} \mathbb{I}_4 & \mathbb{O}_{4 \times 10} \end{bmatrix} \quad (3.66)$$

3.4.2.1 Nominal Model selection

For each synthesis point V_i , the nominal model $\mathbf{G}_{B,i}$ is selected using the ν -gap metric-based procedure outlined in Section 3.2.1. The mutual distances between estimated models at each airspeed value are shown in Figure 3.29 and the maximum ν -gap distances for the two nominal model selection strategies (mean model and candidate from the set of estimated model) are given in Table 3.2.

At $V_3 = 25$ m/s, these distances are identical to the previously found results. However, the distances observed at the other scheduling points are much higher, especially at the highest and lowest airspeeds. At $V_1 = 15$ m/s, the best candidate is the model computed with the mean values of the parameters, with $\delta_\nu = 0.52$, while at $V_6 = 40$ m/s the best candidate is the sixth model $\mathbf{P}_{6,6}$ from the set $\mathbf{\Pi}_6$, with $\delta_\nu = 0.52$ as well. Given that the gap metric δ_ν ranges from 0 to 1, these values are indeed very high and indicate a poor fit of the nominal model with respect to the model set. This is due to some models in the set exhibiting large differences in dynamics with the rest of the set. These outliers are visible in Figure 3.29 at operating points indices 5, 6, 14 and 18 for $V_1 = 15$ m/s and $V \geq 30$ m/s, where the δ_ν distances are much higher than for other points in the set. These operating points are close to the edge of the flight envelope, with $\bar{\beta} = 6^\circ$. At these operating conditions, the projectile behavior is severely degraded due to reduced control efficiency and higher sensitivity to disturbances, as observed in Chapter 2.

In order to keep the maximum ν -gap distances at reasonable values, these operating points are excluded from the model set, corresponding to a slightly smaller flight envelope. The ν -gap distances

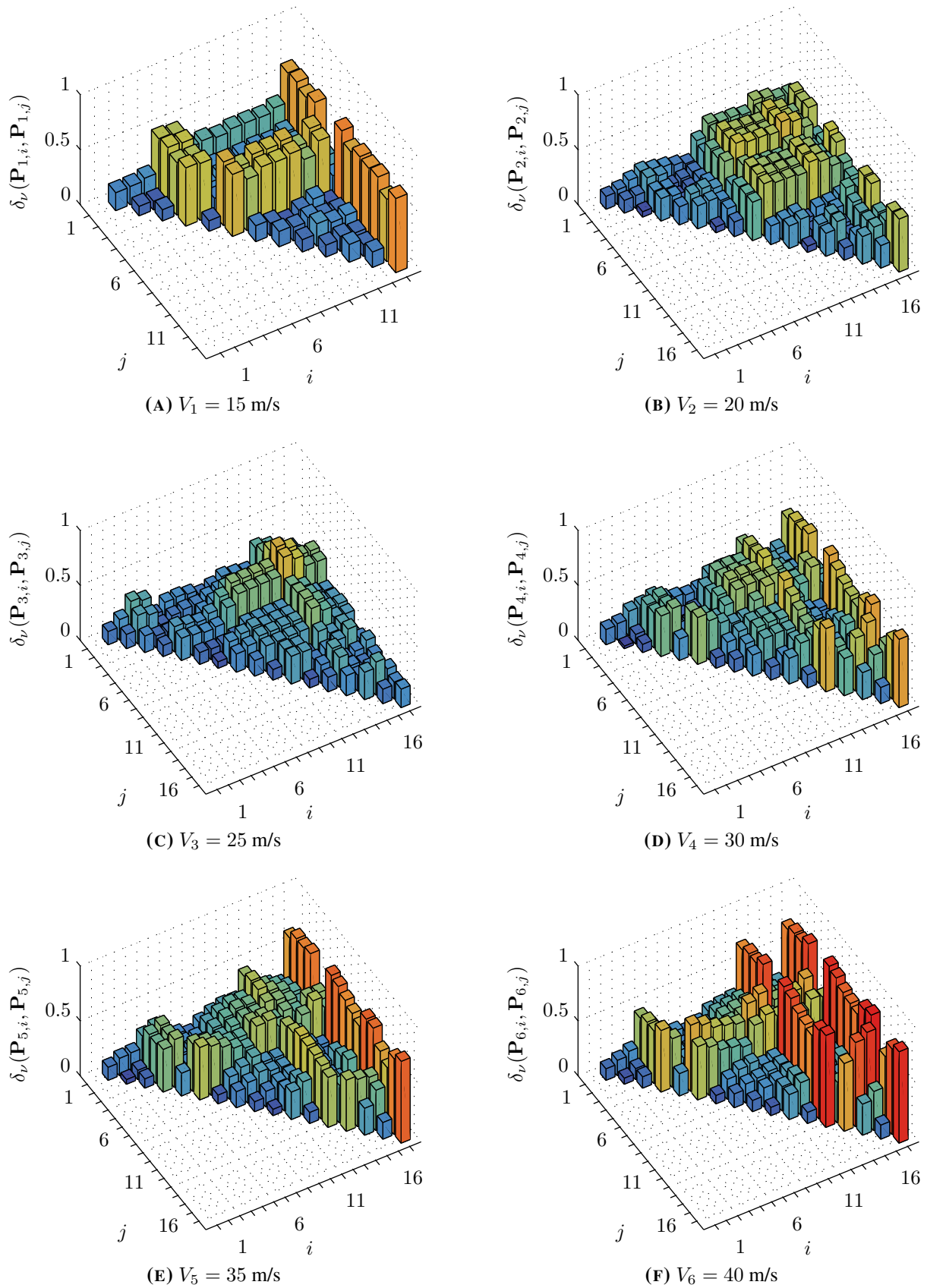


FIGURE 3.29: Mutual ν -gap distances for all estimated models, for the six considered values of the airspeed V . For readability, only the upper diagonal part of the ν -gap distances matrix is shown.

TABLE 3.2: Maximum ν -gap distances between candidate nominal models and the set of estimated models, for the complete and the reduced flight envelopes

Airspeed	Complete flight envelope			Reduced flight envelope		
	Mean model	Model from set		Mean model	Model from set	
$V_1 = 15$ m/s	0.52	0.56	$\mathbf{P}_{1,6}$	0.24	0.21	$\mathbf{P}_{1,4}$
$V_2 = 20$ m/s	0.39	0.41	$\mathbf{P}_{2,10}$	0.30	0.37	$\mathbf{P}_{2,10}$
$V_3 = 25$ m/s	0.31	0.33	$\mathbf{P}_{3,10}$	0.31	0.33	$\mathbf{P}_{3,10}$
$V_4 = 30$ m/s	0.46	0.41	$\mathbf{P}_{4,15}$	0.25	0.27	$\mathbf{P}_{4,1}$
$V_5 = 35$ m/s	0.62	0.43	$\mathbf{P}_{5,5}$	0.24	0.25	$\mathbf{P}_{5,9}$
$V_6 = 40$ m/s	0.68	0.52	$\mathbf{P}_{6,6}$	0.26	0.25	$\mathbf{P}_{6,9}$

obtained in this case are presented in Table 3.2 and show a major improvement in model distance over the full-flight envelope case.

The eigenvalues of the set of estimated models and the nominal models are illustrated in Figure 3.30. The eigenvalues of the set of estimated models are represented by colored crosses, one color per considered airspeed, while the filled squares represent the corresponding nominal models. The corresponding bandwidth and damping factor values of the nominal models are then summarized in Table 3.3. As observed during the identification study, an increase in the airspeed induces an increase of the system bandwidth and a reduction of the damping ratio.

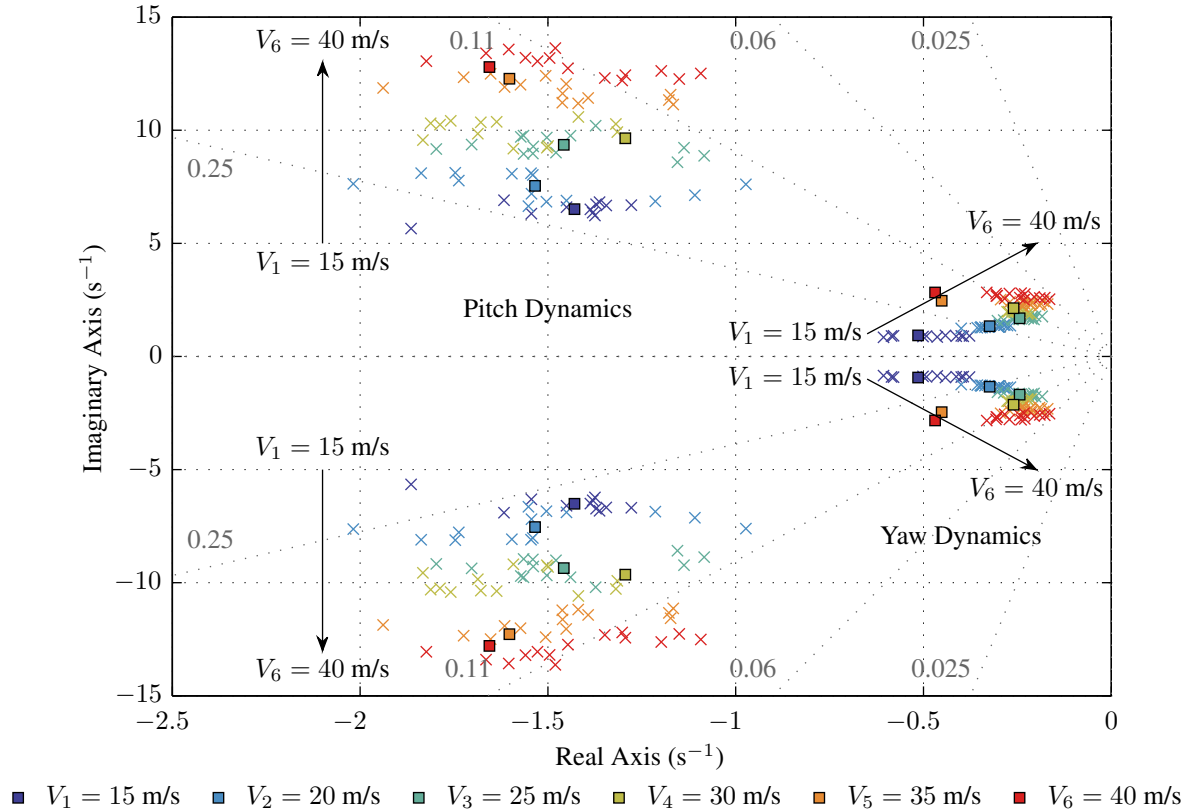
**FIGURE 3.30:** Eigenvalue loci of the estimated and selected nominal models.

TABLE 3.3: Per-channel open-loop bandwidth and damping factor.

Airspeed	Pitch Channel		Yaw Channel	
	ω_n (rad/s)	ζ	ω_n (rad/s)	ζ
15 m/s	6.67	0.22	1.02	0.47
20 m/s	7.68	0.20	1.36	0.23
25 m/s	9.45	0.15	1.70	0.14
30 m/s	10.08	0.16	2.01	0.12
35 m/s	11.86	0.12	2.37	0.09
40 m/s	12.96	0.11	2.68	0.09

3.4.2.2 Autopilot Design

The autopilot structure is identical to the skid-to-turn autopilot presented in Section 3.3, except for the feedforward filter which is implemented in filter (series) form. Initial synthesis results showed that an injection filter can lead to non-minimum phase zeros in the closed-loop $\mathbf{T}_{r \rightarrow y}$ transfer for some synthesis points, even if the feedforward filter does not contain right-hand plane (RHP) zeros. These zeros severely degrade the reference-tracking performance and may complicate the design of a guidance loop in a complete GNC design. On the opposite, a series feedforward filter is implemented in series with the closed-loop plant and does not change the location of the existing poles and zeros, but rather adds its own poles and zeros to the transfer function. The parameters of \mathbf{K}_F in (3.50) are constrained such that it has unity DC gain on both channels and all its poles and zeros lie in the left-hand plane (LHP). This revised autopilot structure is illustrated in Figure 3.31.

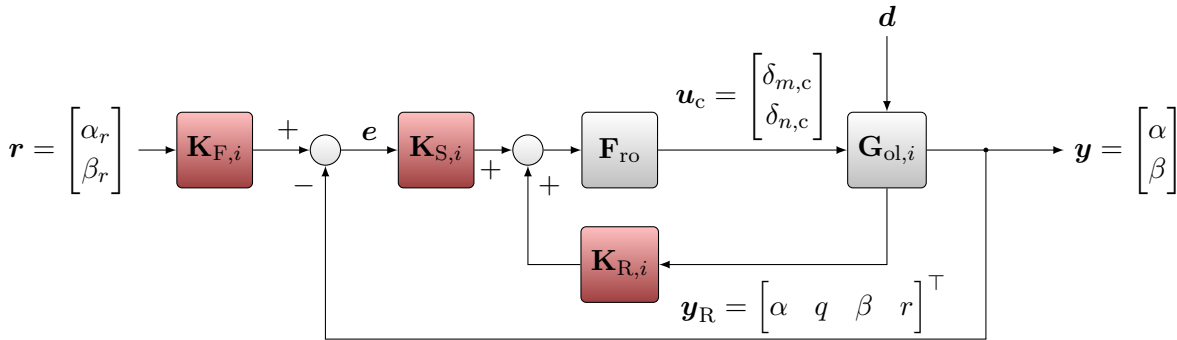


FIGURE 3.31: Individual autopilot structure.

The individual autopilot design is carried out using the synthesis diagram of Figure 3.32, which is similar to Figure 3.11 except for the location of the feedforward filter. For each considered airspeed V_i , the three controller blocks $\mathbf{K}_{R,i}$, $\mathbf{K}_{S,i}$ and $\mathbf{K}_{F,i}$ are computed at once using multi-objective \mathcal{H}_∞ synthesis under the same design constraints for all airspeeds.

Compared to the fixed-airspeed case, the bandwidth of the reference model \mathbf{T}_{ref} is reduced for the yaw channel, due to the very low open-loop bandwidth observed for low airspeeds. Indeed, if the reference model bandwidth is kept at 3 rad/s, a significant control gain would be required on the yaw channel at these airspeeds. The resulting control signal peaks would saturate the actuators and would result in large fin deviations, in addition to severely deviating from equilibrium. To avoid these issues, the model reference bandwidth is reduced to $\omega_{0,\beta} = 1$ rad/s on the yaw channel. The weighting filters \mathbf{W}_{jk} are kept

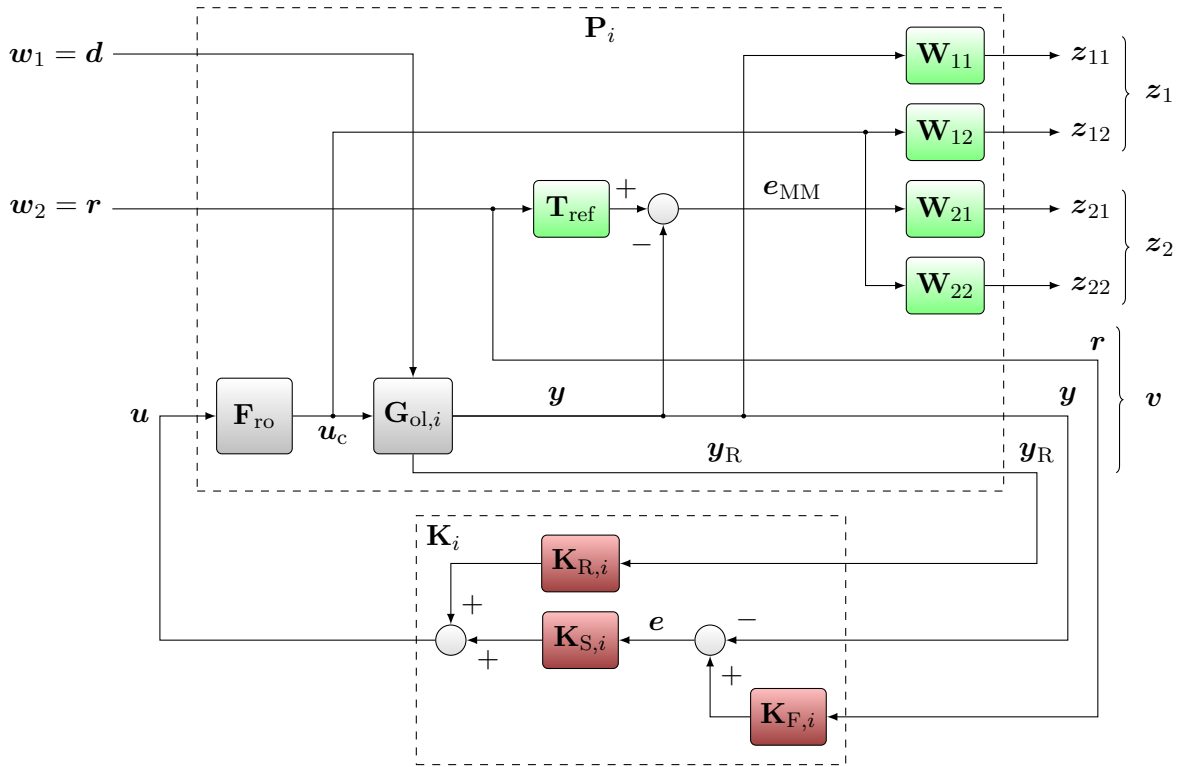


FIGURE 3.32: Variable-air-speed skid-to-turn multi-objective autopilot synthesis diagram.

identical, except for the reference-tracking roll-off constraints which are tightened thanks to the lowered reference model bandwidth, with $k_{22,LF} = 2$.

3.4.2.3 Controller Synthesis Results

The autopilot synthesis for each scheduling point is done using the multi-objective structured \mathcal{H}_∞ synthesis method discussed in the previous application cases: for each fixed-air-speed plant P_i , given $\gamma_i > \gamma_{\min}$, find all stabilizing controllers $K_i = [K_{R,i} \quad -K_{S,i} \quad K_{S,i}K_{F,i}]$ with the structure given in Equations (3.48) – (3.50) such that:

$$\max \{ \|\mathbf{T}_{w_1 \rightarrow z_1}(P_i, K_i)\|_\infty, \|\mathbf{T}_{w_2 \rightarrow z_2}(P_i, K_i)\|_\infty \} < \gamma_i \quad (3.67)$$

The peak \mathcal{H}_∞ gain of the resulting designs monotonically increases from $\gamma_1 = 1.57$ at $V_1 = 15$ m/s to $\gamma_6 = 1.88$ at $V_6 = 40$ m/s. The detailed values of γ are given in Table 3.4 and indicate an overall good respect of the design constraints.

The controller time- and frequency-domain synthesis results for each local controller are respectively illustrated in Figures 3.33 and 3.34 for the disturbance rejection and reference tracking objectives. These results are given separately for the pitch and yaw channels, since the plant exhibits negligible cross-channel couplings.

TABLE 3.4: Peak \mathcal{H}_∞ gain γ for the individual autopilots.

Airspeed	15 m/s	20 m/s	25 m/s	30 m/s	35 m/s	40 m/s
Peak \mathcal{H}_∞ gain γ	1.37	1.39	1.47	1.52	1.74	1.88

The closed-loop disturbance rejection performance transfers exhibit similar overall trends to the fixed-air-speed case. However, the mid-frequencies peak of the sensitivity transfer $\mathbf{S}_{d \rightarrow e}$ is largely dependent of the airspeed as illustrated in Figures 3.33A and 3.33B. The \mathbf{W}_{11} specification is violated at higher airspeeds on the pitch channel, and on the yaw channel to a lesser extent. This is due to the increase in open-loop bandwidth and the decrease in damping factor at these airspeeds, while the allowed control bandwidth remains constant. The value of this peak is the largest contribution to the increase of γ at high airspeeds. This behavior is also found in the time domain, with a higher peak deflection at high airspeeds as shown in Figures 3.33E and 3.33F. An incoming disturbance step is rejected in 0.55 s to 0.97 s for the pitch axis and in 3.8 s to 18 s for the yaw axis.

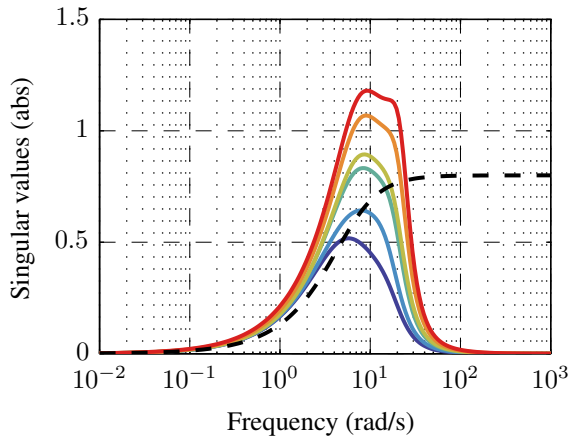
Regarding the reference tracking objective, the trends are also similar to the fixed-air-speed case. The model-matching error, illustrated in Figures 3.34A and 3.34B, does not vary significantly with the airspeed and the maximum difference is less than 15% on both channels. Thanks to the reduced yaw reference model bandwidth, the control input sensitivity transfers $\mathbf{T}_{r \rightarrow u}$ meet the specifications with a significant margin. The high-frequency gain on both channels is higher for low airspeeds, in order to speed up the system response to follow the reference model. The response to a reference step is shown in Figure 3.34E for the pitch axis and in Figure 3.34F for the yaw axis. Since the same reference model is used for all synthesis points, the time-domain performance does not change significantly with respect to the airspeed and the responses are close to the reference model response. The 5% response times are 1.20 s to 1.28 s for the pitch channel and 3.04 s to 3.88 s for the yaw channel, while the reference model response times are respectively 1.09 s and 3.28 s.

3.4.2.4 Robustness Analysis

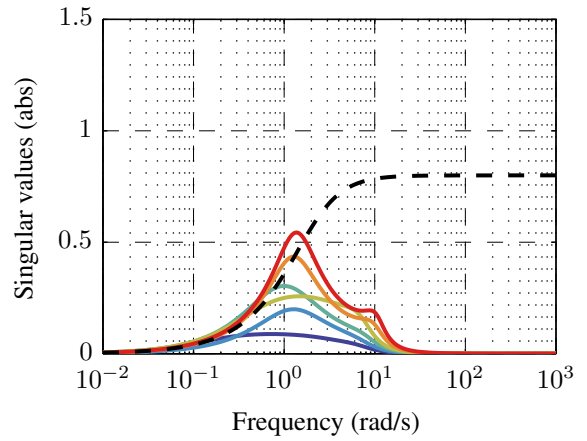
The robust stability of the individual controllers is first verified using the loop-at-a-time and multivariable gain and phase margins. The results are summarized in Table 3.5 and the pitch and yaw channels open-loop responses are illustrated in Figure 3.35. The loop-at-a-time gain and phase margin specifications are respected by all controllers, with slightly decreasing margins as the scheduling airspeed increases. However, looking at the multivariable gain and phase margins, the obtained values are slightly under specifications for airspeeds over $V_3 = 25$ m/s but do not foretell instability issues.

The robust stability of the individual controllers is analyzed in the same way as the fixed-air-speed case. For each synthesis point V_i , an uncertain model is constructed to take into account the uncertainty on model parameters, but also uncertainty on the scheduling point. To this end, a parametric uncertain model is constructed using the minimum and maximum values of the estimated parameters at the synthesis point V_i as well as the adjacent synthesis points V_{i-1} and V_{i+1} : the set of models considered for computing uncertainty is $\mathbf{\Pi}^* = \mathbf{\Pi}_{i-1}^* \cup \mathbf{\Pi}_i^* \cup \mathbf{\Pi}_{i+1}^*$, where each $\mathbf{\Pi}_j^*$ contains the set of estimated models $\mathbf{\Pi}_j$ as well as models issued from the bootstrap procedure. The robustness *w.r.t.* is then assessed for each controller on its respective scheduling point using μ -analysis. The μ upper and lower bounds are illustrated for each case in Figure 3.36. All six controllers are robustly stable as they exhibit a peak $\bar{\mu}$ under 1.

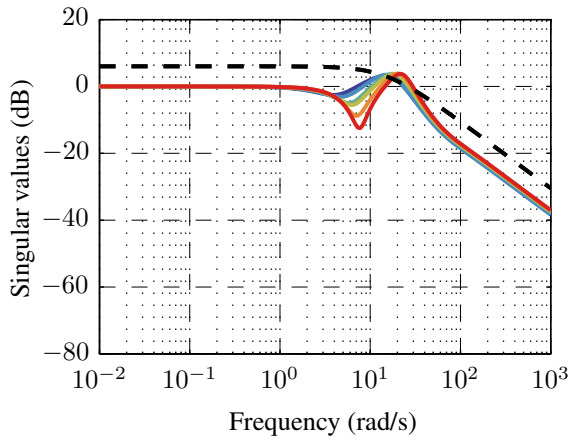
The peak $\bar{\mu}$ is the highest for controllers corresponding to $V_1 = 15$ m/s and $V_6 = 40$ m/s. This is expected from the estimation results analysis, which indicates a higher uncertainty on the estimates at these airspeeds. Between these extremes, the peak $\bar{\mu}$ is under 0.4: the system can tolerate up to 250% of the modeled uncertainty.



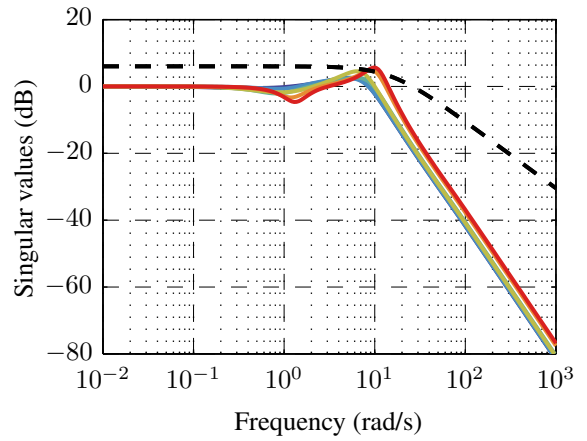
(A) Pitch sensitivity transfer $S_{d \to e}$ and constraint $1/W_{11}$ (dashed black line).



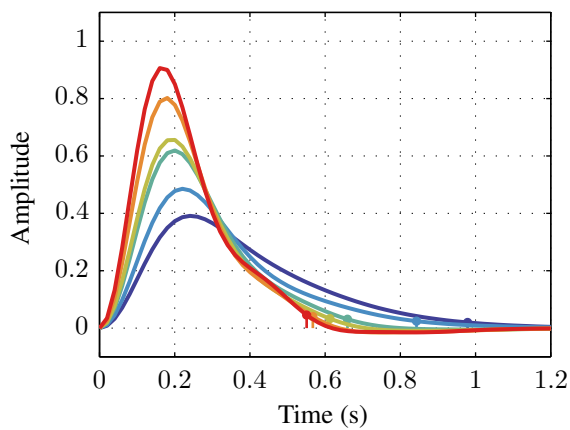
(B) Yaw sensitivity transfer $S_{d \to e}$ and constraint $1/W_{11}$ (dashed black line).



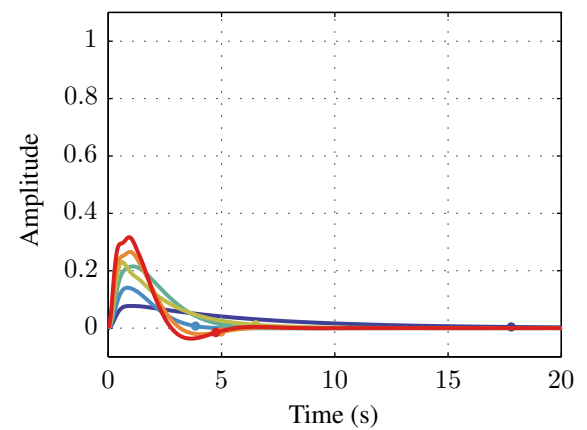
(C) Pitch control sensitivity transfer $T_{d \to u}$ and constraint $1/W_{12}$ (dashed black line).



(D) Yaw control sensitivity transfer $T_{d \to u}$ and constraint $1/W_{12}$ (dashed black line).



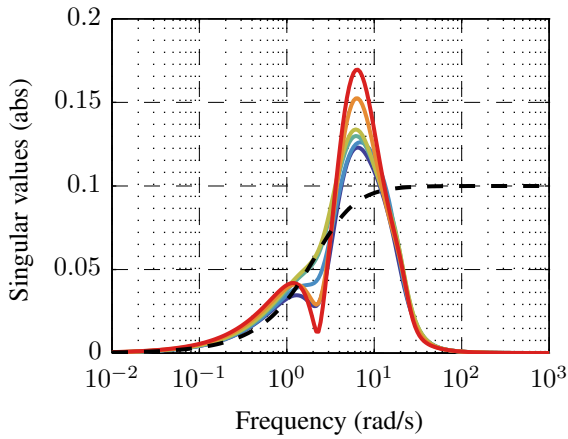
(E) Pitch response to a unit pitch disturbance step.



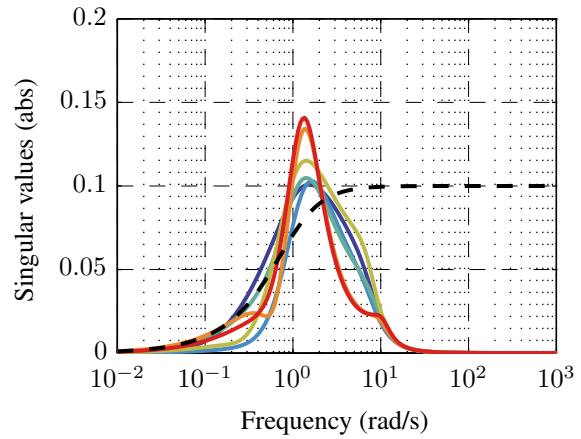
(F) Yaw response to a unit yaw disturbance step.

— $V_1 = 15$ m/s — $V_2 = 20$ m/s — $V_3 = 25$ m/s — $V_4 = 30$ m/s — $V_5 = 35$ m/s — $V_6 = 40$ m/s

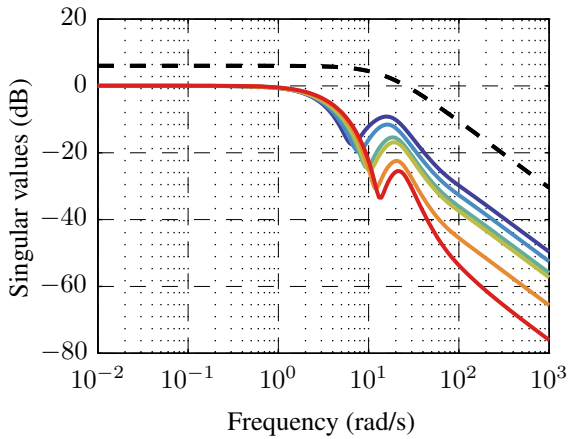
FIGURE 3.33: Closed-loop *disturbance-related* transfer functions and unit step responses



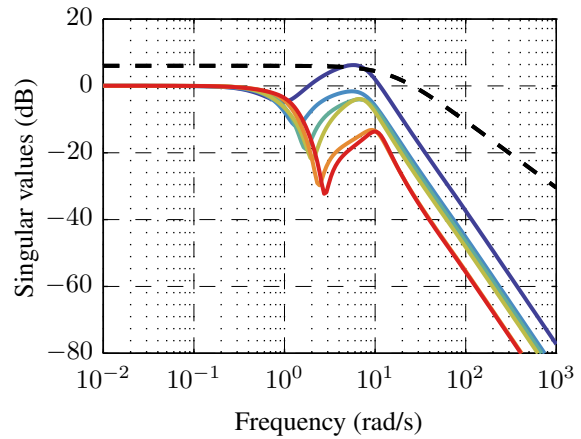
(A) Pitch model-matching error $\mathbf{T}_{\text{ref}} - \mathbf{T}_{r \rightarrow \alpha}$ and constraint $1/\mathbf{W}_{21}$ (dashed black line).



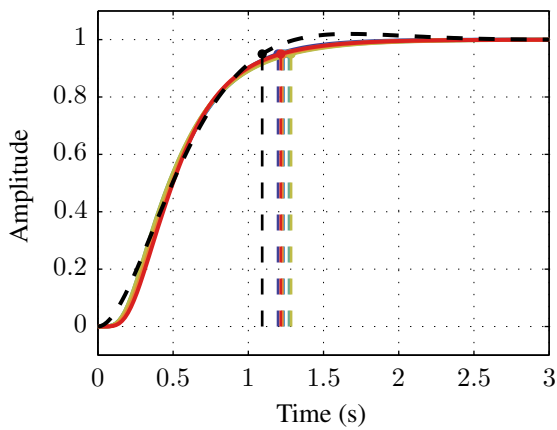
(B) Yaw model-matching error $\mathbf{T}_{\text{ref}} - \mathbf{T}_{r \rightarrow \alpha}$ and constraint $1/\mathbf{W}_{21}$ (dashed black line).



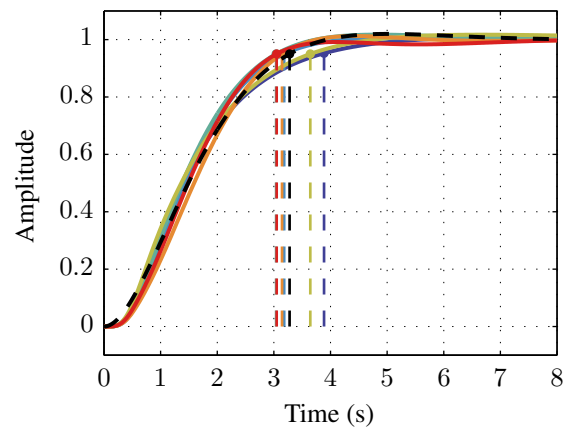
(C) Pitch control sensitivity transfers $\mathbf{T}_{r \rightarrow u}$ and constraint $1/\mathbf{W}_{22}$ (dashed black line).



(D) Yaw control sensitivity transfers $\mathbf{T}_{r \rightarrow u}$ and constraint $1/\mathbf{W}_{22}$ (dashed black line).



(E) Pitch step response against $\mathbf{T}_{\text{ref}, \alpha}$ (dashed black line).



(F) Yaw step response against $\mathbf{T}_{\text{ref}, \beta}$ (dashed black line).

— $V_1 = 15 \text{ m/s}$ — $V_2 = 20 \text{ m/s}$ — $V_3 = 25 \text{ m/s}$ — $V_4 = 30 \text{ m/s}$ — $V_5 = 35 \text{ m/s}$ — $V_6 = 40 \text{ m/s}$

FIGURE 3.34: Closed-loop reference-related transfer functions and unit step responses

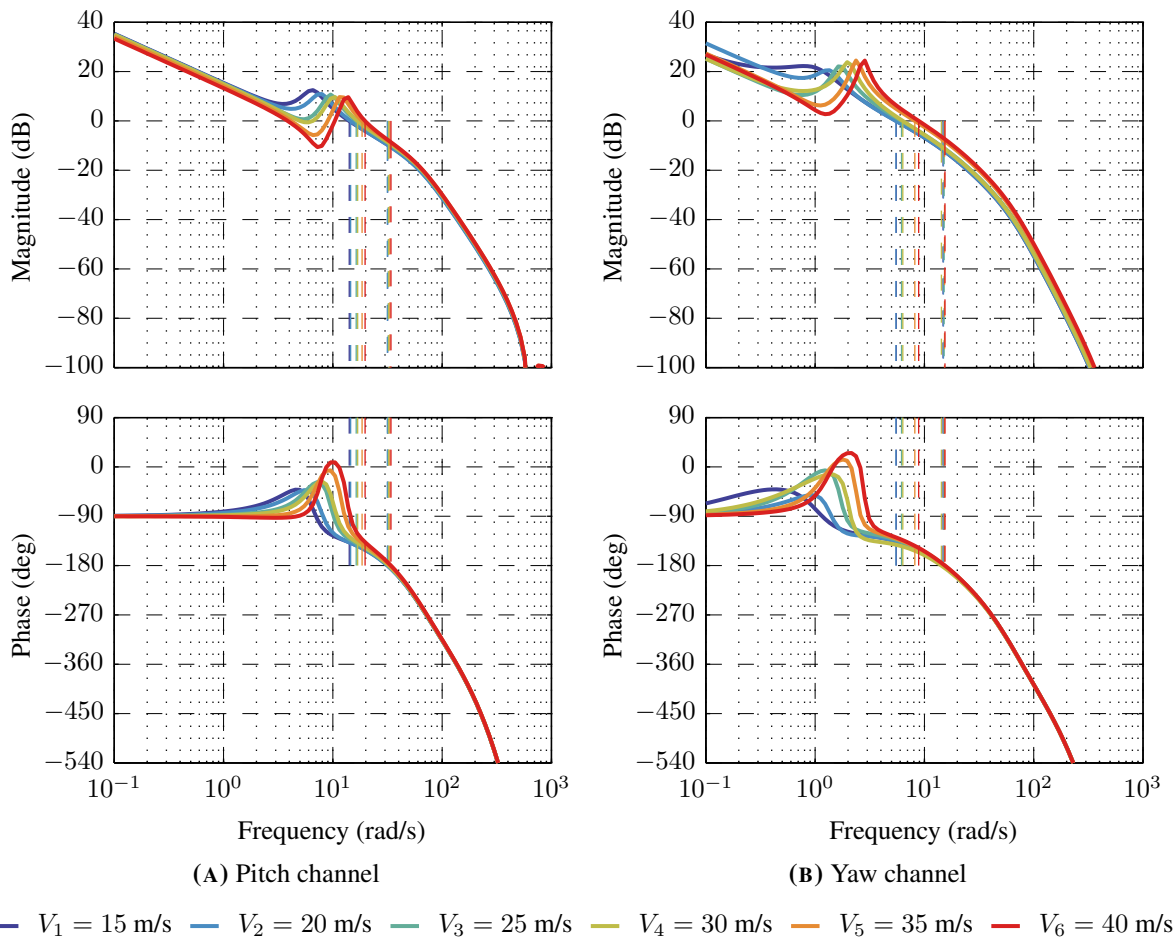


FIGURE 3.35: Loop-at-a-time gain and phase margins for the individual controllers.

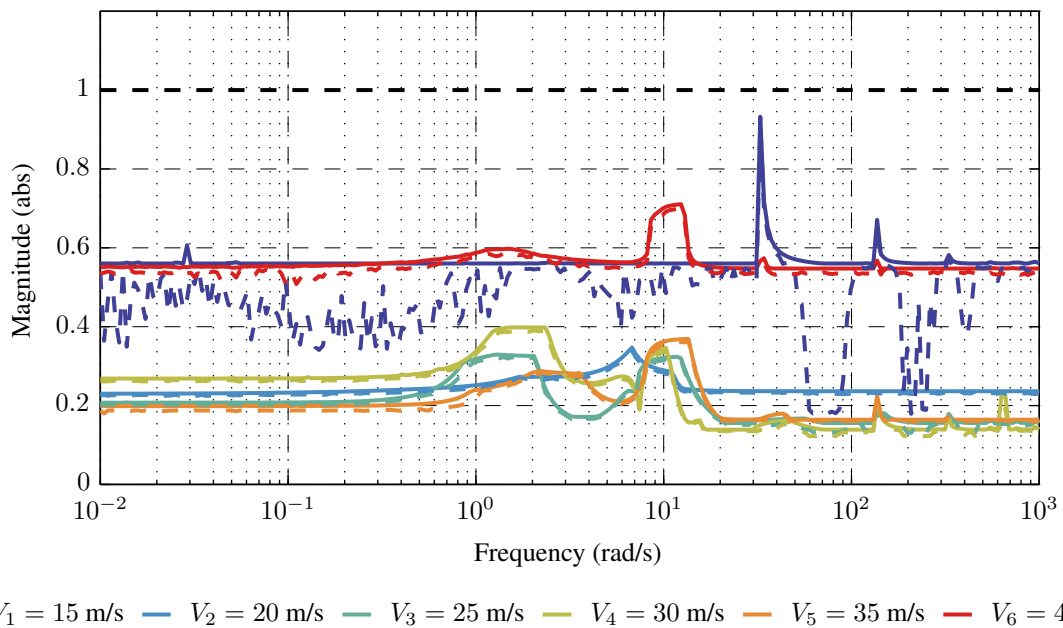


FIGURE 3.36: Structured singular value μ upper (solid) and lower bounds (dashed) of the robust stability-related transfers M^Δ for each synthesized autopilot.

TABLE 3.5: Loop-at-a-time and multivariable gain and phase margins, and peak value of the structured singular value upper bound $\bar{\mu}$.

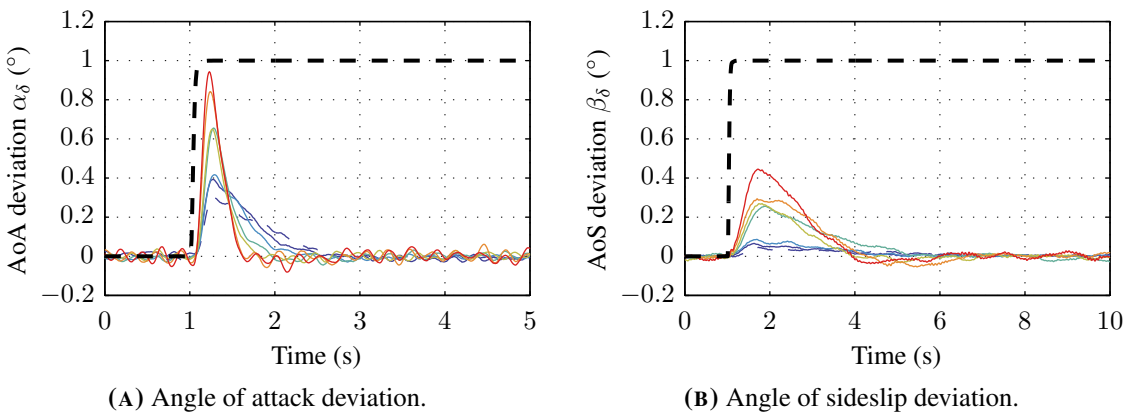
Airspeed	Pitch Channel		Yaw Channel		Multivariable		$\sup_{\omega} \bar{\mu}(\mathbf{M})$
	GM_{α}	PM_{α}	GM_{β}	PM_{β}	GM_{ST}	PM_{ST}	
15 m/s	9.66 dB	42.48°	11.93 dB	43.93°	6.22 dB	37.91°	0.9325
20 m/s	9.70 dB	41.12°	11.97 dB	44.67°	6.04 dB	36.96°	0.3474
25 m/s	8.99 dB	40.67°	11.05 dB	43.76°	5.76 dB	35.47°	0.3293
30 m/s	8.90 dB	41.59°	10.40 dB	34.82°	5.25 dB	32.71°	0.3982
35 m/s	8.63 dB	41.67°	8.17 dB	34.72°	4.74 dB	29.84°	0.3691
40 m/s	8.65 dB	42.85°	7.45 dB	32.29°	4.33 dB	27.43°	0.7106

3.4.2.5 Experimental Validation

The disturbance rejection performance of the individual controllers composing the gain-scheduled autopilot is illustrated in Figure 3.37 for each synthesis airspeed. These curves follow the trends observed in Section 3.4.2.3, i.e. the rejection ratio and the response time decrease with increasing values of the airspeed. At high airspeed values, the system is also more sensitive to aerodynamic perturbations, as shown by the higher level of oscillations for $V_6 = 40$ m/s. In all cases, the disturbances are rejected in less than 1.5 s on the pitch axis and in less than 4 s on the yaw axis and no steady-state error is visible.

The simultaneous tracking performance is illustrated in Figure 3.38. Compared to the fixed-air-speed autopilot described in the previous section, the performance requirements on the pitch and yaw channels are different in order to account for the large variations in yaw dynamics *w.r.t.* the airspeed. The reference inputs have been modified accordingly to let both channels reach steady-state. Despite these large variations in dynamics, all individual controllers achieve the same time-domain reference tracking performance and conform to the reference models $\mathbf{T}_{ref,\alpha}$ and $\mathbf{T}_{ref,\beta}$ with 5% response times in the range $1 \leq t_{r5\%} \leq 1.3$ s for the pitch channel and $3 \leq t_{r5\%} \leq 3.8$ s for the yaw channel.

The only exception to these observations is the synthesis point $V_1 = 15$ m/s, for which the system cannot reach the $\pm 4^\circ$ reference angles of attack. At this point, the reduced flight envelope at this airspeed

**FIGURE 3.37:** Disturbance rejection performance of the individual autopilots at the operating point $(\bar{\alpha}, \bar{\beta}) = (2^\circ, 3^\circ)$.

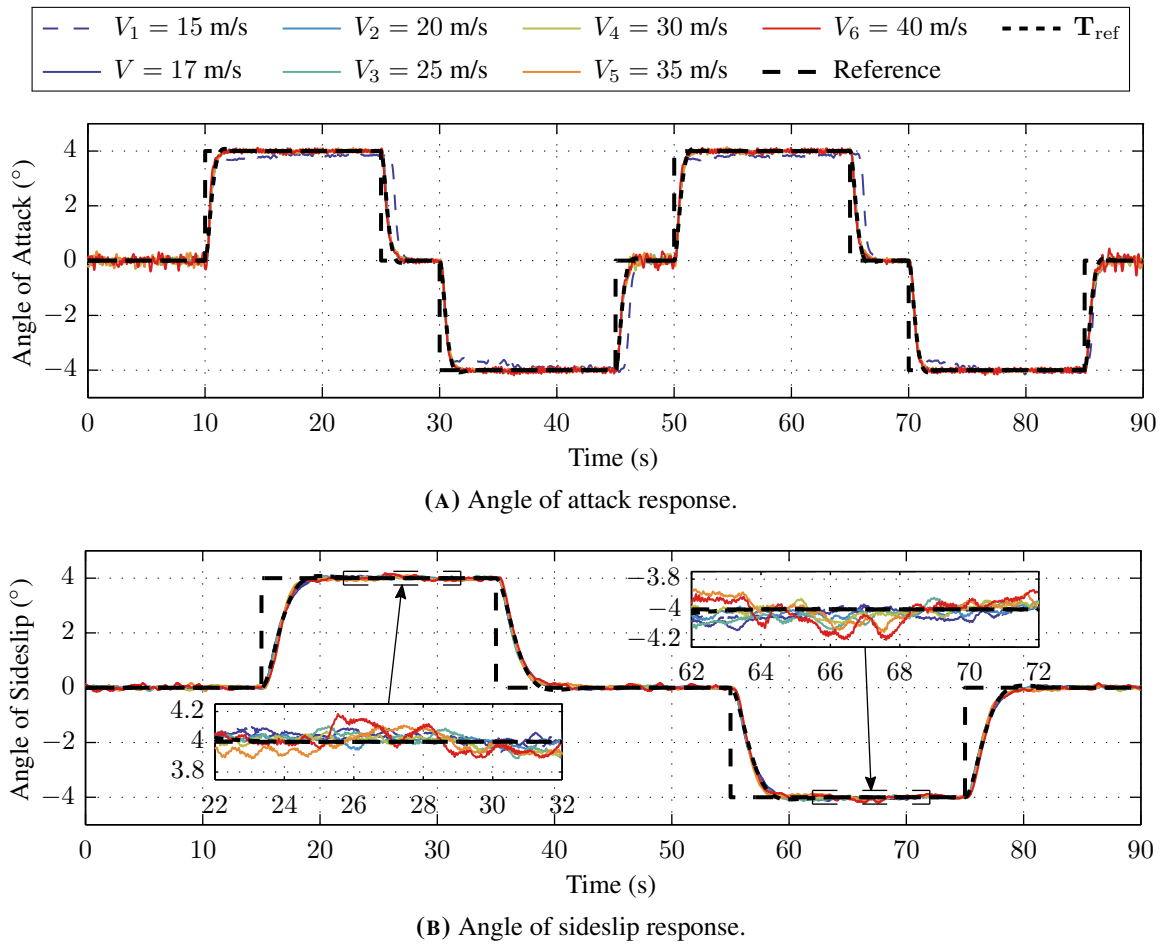


FIGURE 3.38: Reference tracking performance of the individual autopilots for both channels operating simultaneously.

results in a loss of control authority for lower values of the angle of attack, which can be modeled as an input magnitude saturation. This saturation results in integrator windup for the $\pm 4^\circ$ reference angles of attack, visible as a delayed fall in the angle of attack response at $t = 25, 45$ and 65 s. For a slightly higher airspeed ($V = 17$ m/s), these effects are no longer present. The cross-channel couplings are almost nonexistent on the pitch axis, thanks to the reduced yaw dynamics in the present case. The effects of a pitch step on the yaw angle are visible in the insets of Figure 3.38B. The yaw axis cross-channel disturbance rejection is better at lower airspeeds but remains acceptable at $V_6 = 40$ m/s with a maximum deviation of 0.2° for a 4° step on the pitch channel.

3.4.3 Controller Implementation and Experimental Validation

The individual autopilot is based around the three controller blocks illustrated in Figure 3.31. While the roll-off filter F_{ro} remains the same throughout the flight envelope, the regulator gain matrix K_R , the servo controller K_S and the feedforward filter K_F have been designed for each synthesis point.

The global gain-scheduled autopilot retains the same structure, but the controller blocks are now dependent on the scheduling parameter V . The controller parameters are obtained by linear interpolation of the individual controller parameters between the synthesis points. For the regulator K_R and the

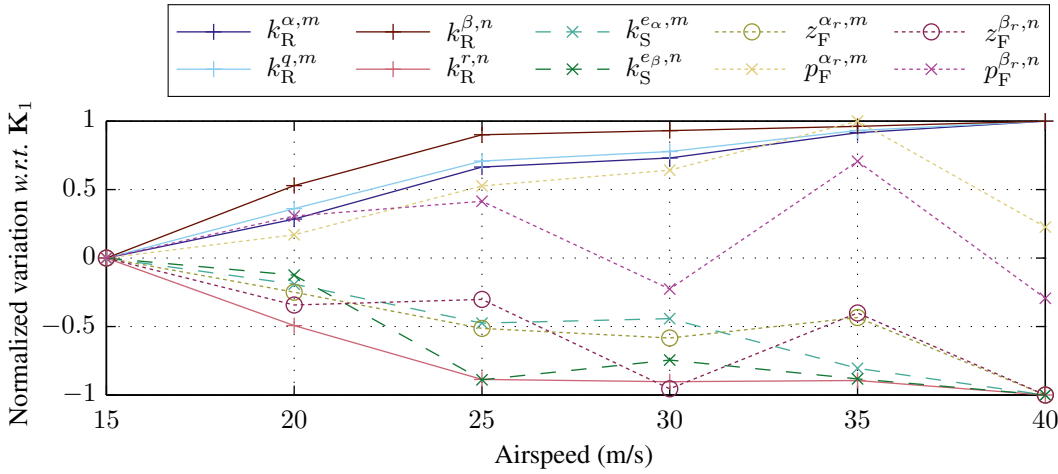


FIGURE 3.39: Normalized variation of the controller parameters w.r.t. values for \mathbf{K}_1 as a function of V .

servo controller \mathbf{K}_S , these parameters are their gains. For the feedforward filter \mathbf{K}_F , the zero and pole of the first-order lead or lag filter are interpolated and its gain remains fixed to unity. This *zero-pole-gain interpolation* technique guarantees the unity feedforward DC gain. Given $V_i < V < V_{i+1}$, the parameters $k_A^{j,m}$ of the gain-scheduled controller \mathbf{K} are then obtained as follows:

$$k_A^{j,m} = (1 - a)k_{A,i}^{j,m} + ak_{A,i+1}^{j,m} \quad (3.68)$$

with

$$a = \frac{V - V_i}{V_{i+1} - V_i} \quad (3.69)$$

where $k_{A,i}^{j,m}$ is the value of $k_A^{j,m}$ in the i -th local controller \mathbf{K}_i .

The variations of the controller parameters as a function of V are illustrated in Figure 3.39, with respect to the parameter values of \mathbf{K}_1 . The parameters of \mathbf{K}_R and \mathbf{K}_S exhibit smooth and almost monotonic variations. However, the variations of the feedforward parameters are more discontinuous, especially on the yaw channel.

This ad-hoc scheduling technique does not provide any theoretical stability, performance and robustness guarantees between the synthesis points. In general, the controller validation is carried out with extensive Monte-Carlo simulations performed on the complete nonlinear or LPV model of the plant. In the present case, the available models only allow validation in the local case, and hence the complete gain-scheduled controller is evaluated on the experimental setup instead of numerical simulations.

The performance of the complete gain-scheduled autopilot is assessed using a stepwise input signal while the airspeed is monotonically increasing from 17 m/s to 40 m/s. These results are illustrated in Figure 3.40 and show a good agreement between the measured response and the reference models $\mathbf{T}_{\text{ref},\alpha}$ and $\mathbf{T}_{\text{ref},\beta}$ throughout the airspeed variation range. In the observed results, the system remains stable and provides a constant level of performance even between scheduling points. The angle of attack response does not show steady-state error. However on the yaw axis response slight over- or undershoots can be seen near $t = 55, 100$ and 120 s. These errors can be attributed to the fact that the scheduling vector is varying, and also due to the interpolation of the controllers.

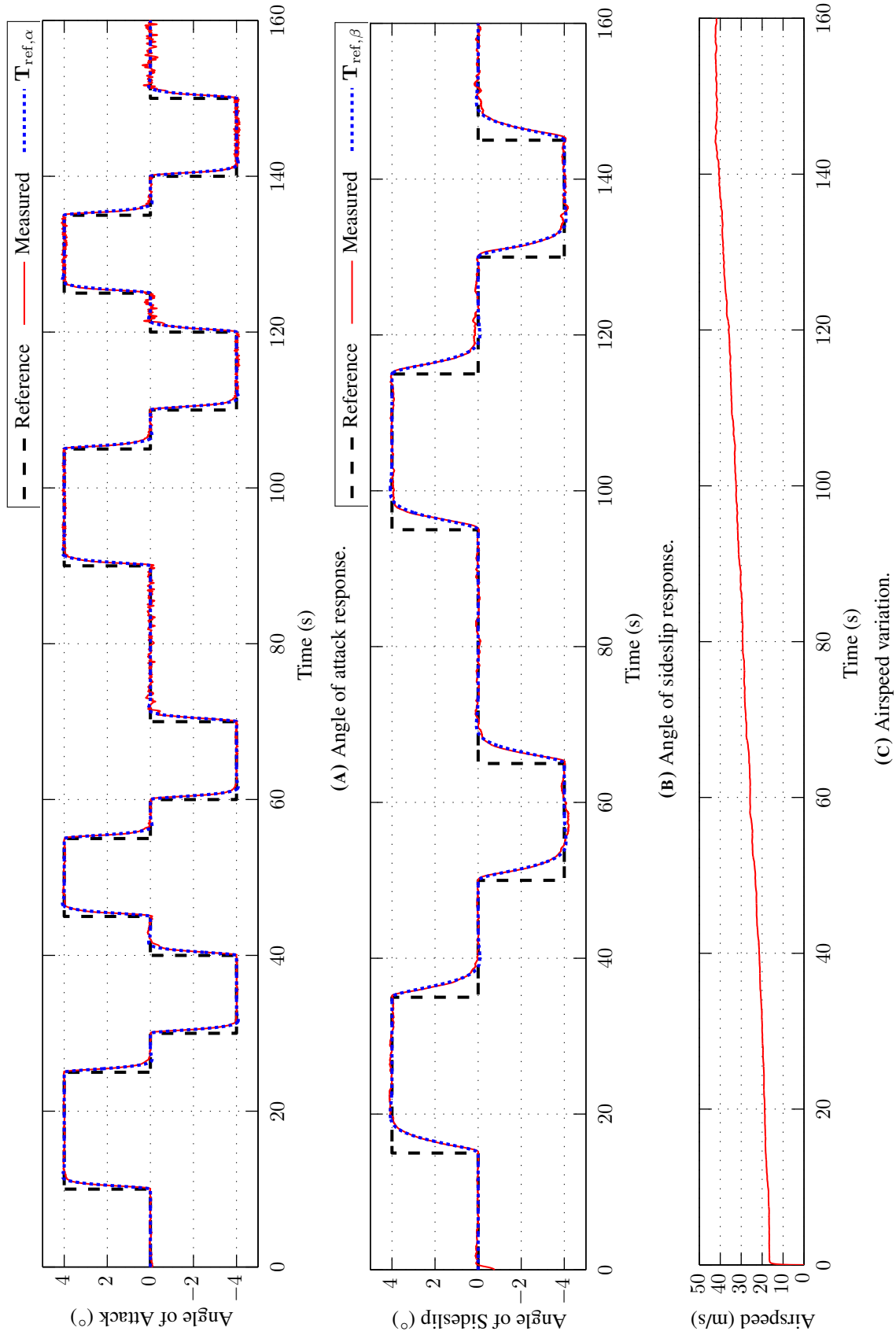


FIGURE 3.40: Reference tracking performance of the gain-scheduled autopilot.

3.4.4 Concluding Remarks

In this section, a full flight envelope gain-scheduling autopilot is designed, based on the fixed-air-speed autopilot design results. As a first approach, a divide-and-conquer strategy is employed, allowing the direct exploitation of the estimated linear model families for building a family of linear controllers, one for each value of the airspeed considered in the estimation procedure. The individual controllers are designed with objectives and techniques similar to the fixed-air-speed *skid-to-turn* autopilot, with minor alterations for taking into account the large variations of open-loop dynamics across the operating domain. The non-linear, gain-scheduling autopilot is then built upon this family of autopilots using a zero-pole-gain interpolation scheme.

This ad-hoc gain-scheduling control technique does not provide any stability or robustness guarantee between scheduling points, even if all of the individual autopilots are robustly stable on their operating point. As no complete nonlinear simulation model is available, the validation of the gain-scheduled controller for intermediate operating points has been performed on the experimental testbench. In these experiments, the gain-scheduled controller has provided consistent tracking performance across the air-speed range, while keeping the system stable at all times.

3.5 Conclusion

In this chapter, the design of autopilots for the ACHILES projectile with reference tracking, disturbance rejection and robust stability objectives has been carried out using modern multivariable robust \mathcal{H}_∞ control techniques under mixed-sensitivity formulations. The plant models are built upon the families of linear model resulting from the identification process. Since the constant-air-speed dynamics exhibit low to moderate variations with respect to the incidence angles, the projectile behavior can be approximated by a single nominal model. A corresponding uncertain model takes into account the differences between the estimated and nominal models as well as the uncertainty on the estimated parameters, and allows testing the robustness of the designed control laws.

Three application cases pertaining to the design of a *skid-to-turn* autopilot were considered. The first case considered only the pitch axis in a preliminary study and is detailed in Appendix B. These results are extended in the second case, where a pitch-yaw autopilot design is carried out for a fixed airspeed using a multi-objective \mathcal{H}_∞ synthesis technique. Despite large differences in open-loop dynamics between the pitch and yaw axes, this autopilot provided identical performance on both channels. The third application considers the fixed-air-speed autopilot results and extends the approach to consider the complete flight envelope with variable airspeed. In order to cope with the airspeed variations inducing large changes in dynamics, a nonlinear controller is built using gain-scheduling techniques. The divide-and-conquer technique is entirely compatible with the identification and control approaches developed previously, using the airspeed vector selected for model estimation. For each airspeed, a linear controller is designed using the constant-air-speed techniques applied previously. The complete nonlinear controller is then obtained by interpolation of the gains, poles and zeros of the individual controllers.

In both application cases, the obtained control laws have conformed to the performance and stability specifications for the nominal case and were proven to be robustly stable against the model uncertainty. In a final validation step, each control law was implemented on the experimental setup, and evaluated for both control objectives. The experimental results were compared with numerical simulations, showing

that the actual implementation follows the simulated responses closely and in accordance to the design requirements. Together with the models and associated parameters obtained in Chapter 2, these excellent experimental results validated the appropriateness of the experimental setup and the relative design methodology. Hence this approach provides a complete control design framework, from the modeling and characterization of the projectile's behavior to the validation of the control laws on the real system.

Conclusion and Perspectives

The objective of this thesis was to conduct a feasibility study on a novel approach for the design and validation of flight control laws for guided projectiles. In the early stages of projectile design, classical aerodynamic modeling techniques are indeed relatively time-consuming or inapplicable as some require a flightworthy prototype. The resulting models are highly complex but have a high degree of fidelity, but which may not actually be required for the design of control laws. As a matter of fact, modern control techniques such as \mathcal{H}_∞ control allow the design of robust control laws from lower-complexity models, yet still yielding excellent performance on the complete plant. Furthermore, apart from numerical simulations and free-flight testing, which rely respectively on detailed models and flight-ready prototypes, experimental means for validating projectile control laws in early design phases are inexistent.

The method proposed herein consists in approximating the free-flight behavior of a projectile using a fully actuated, sensor-equipped prototype installed in a wind tunnel by means of a support structure allowing all rotations. Using this setup, the projectile behavior can be characterized experimentally using system identification techniques. The obtained models can then be used for the design and analysis of control laws, which can be then evaluated experimentally in realistic conditions. The legitimacy of this approach was analyzed with a proof-of-concept case study, in which the design of *skid-to-turn* autopilots was carried out for a missile-like fin-stabilized, canard-guided projectile.

During this thesis, a complete experimental testbench was developed from scratch. The architecture, design and components of this setup were discussed in Chapter 1. One of the key design objectives was to construct a rapid controller prototyping environment, so as to streamline the development and testing process. The experimental setup was constructed as an Hardware-In-the-Loop platform, involving a fully autonomous projectile prototype equipped with embedded actuators, sensors and computing capabilities. The real-time requirements associated with such architectures were taken into account and guaranteed with a Xenomai-based real-time Linux environment. The flight conditions are emulated by the use of a subsonic wind tunnel and a gimbal-based support structure. This experimental setup has satisfied all of its design objectives and has been used with success throughout the various autopilot design and validation phases.

The construction of control-oriented projectile mathematical models has been considered in Chapter 2. The in-flight behavior of projectiles is governed by the laws of flight dynamics. However, taking into account the specificities of the experimental setup has yielded a slightly different model as in the free-flight case. Even though the angular dynamics are more complex in the considered configuration, the expressions are still comparable to the free-flight case. Since the resulting nonlinear model cannot be employed directly with modern \mathcal{H}_∞ -based robust control techniques, it is linearized in the neighbor-

hood of operating points. In the resulting linear model structure, the projectile dynamics and induced aerodynamic forces and moments are characterized through 12 parameters, which are to be identified. An experimental trimming procedure has then been carried out to determine the equilibrium conditions of the projectile in the complete airspeed range allowed by the wind tunnel. Results have shown that the usable flight domain remains constant and symmetric across the airspeed range, except at low speeds where the flight envelope is reduced.

The second part of Chapter 2 focused on the estimation of the linear model parameters from experiments carried out on the projectile testbench. The determination of parameters from measured data is an inverse problem, which can be impossible to solve if it is ill-posed, or difficult to solve if it is ill-conditioned. In order to guarantee the existence and quality of the estimated parameters, a standard system identification procedure has been applied. The well-posedness and well-conditioning of the problem were respectively evaluated through *a priori* and *a posteriori* identifiability analyses. These studies concluded that even though the model structure is globally identifiable, the cross-axes coupling parameters were not *a posteriori* identifiable and that the estimation may not be precise for these parameters. In order to cover the flight envelope, data-collection experiments were conducted for several operating conditions. The model parameters were estimated for each data set and presented good confidence intervals. Analysis of the resulting family of linear models showed a strong dependence of the projectile behavior with respect to the airspeed, and a moderate variation across the flight envelope for a constant airspeed.

The estimated model families were exploited in Chapter 3 for the design of robust *skid-to-turn* attitude control laws for the projectile prototype. The objectives of these autopilots are threefold. At all times, the autopilots must ensure the stability of the closed-loop system and be robust with respect to uncertainty. Secondly, the autopilot should provide efficient rejection of external disturbance signals, mainly induced by aerodynamic perturbations. Finally, the plant should track reference angle of attack and angle of sideslip signals, with a fast response time and no steady-state error. Based on the estimation results, it was observed that the projectile behavior at a constant airspeed can be approximated by a single nominal model, and deviations from this model can be considered as parametric uncertainty. Three application cases were then considered: the first and the second respectively dealt with the design of fixed-airspeed pitch-only and pitch-yaw autopilots, and the third case extended the autopilot design to take variations of the airspeed into account using a gain-scheduled approach. To this end, low-complexity controller structures were favored, as they can be adapted to varying operating conditions with simple interpolation schemes, and they present a low implementation computational burden. The controller gains were optimized in accordance to the design requirements using a multi-objective fixed-structure \mathcal{H}_∞ synthesis technique, leading to robustly stable controllers for all considered cases.

In a final step, the autopilots were implemented on the experimental setup, and several exhaustive validation tests were conducted to assess the performance of the proposed solution against numerical simulations. In all considered cases, the actual projectile followed the simulated responses closely and in accordance to the design requirements, hence validating the legitimacy of the proposed approaches. Using the proposed experimental setup the complete control design process can be mastered, from the modeling and identification steps to the validation of control laws in a realistic environment.

Perspectives

This first application allowed the presented methodology to be tested and validated on a concrete real-world case. Identification and control results showed this approach is suitable for complete control design studies based on an experimental setup. Several research axes emerge for improving the realism of this experimental approach and generalizing this methodology to more complex cases.

From an experimental viewpoint, one of the first research axes aims at improving the realism of the experimental setup. In order to take into account the actual free-flight airspeeds (Mach number over 0.5), further testbenches should be based on transsonic or supersonic wind tunnels. However, the technical solutions presented here may not apply due to the smaller test sections of these wind tunnels, and the higher airspeed induces larger wind forces. Hence, a redesign of the projectile prototype and support structure will be required and additional aspects, such as hinge moments on the control surfaces and flexible modes, will have to be taken into consideration. Furthermore, the considered experimental setup limited the study to the angular motion of the projectile. Even though it is not possible to allow linear motion on all three axes, it may still be possible to take the aerodynamic forces into account, for instance with a 6-DoF aerodynamic balance. Hence, the projectile accelerations and linear dynamics may be simulated using force measurements in order to conduct 6-DoF HIL simulations.

In further studies, other projectile configurations may be considered. It would be interesting to apply this methodology to spin-stabilized projectile concepts, notably spin-decoupled course-correction fuzes. The main difficulties in this case are from an engineering perspective, due to size and integration constraints, and the presence of multiple fast-spinning elements. From a control perspective, the high spin rates lead to a very strong coupling between axes and complex aerodynamic effects.

From an algorithmic viewpoint, there are also several perspectives. Regarding the modeling and identification of the projectile dynamics, in the present case, the parameters of a linear model structure were estimated for each operating point, which resulted in a family of linear models. In order to improve the fidelity of the projectile model, especially in more complex cases with higher parameter dependency, the aerodynamic coefficients in the nonlinear model may be described using polynomial decompositions. The main difficulty then consists in the estimation of the polynomial coefficients from experimental data. In addition to the selection of the polynomial description, proving the *a priori* and *a posteriori* identifiability of the model structure as well as the estimation of the parameters, are much more difficult due to the nonlinear model structure.

Regarding control design, the proposed methodology may be improved in several axes. Multi-model synthesis techniques may be applied, in order to avoid the approximation introduced by using a single nominal model at each airspeed, by taking into account all members of the families of estimated models. Gain-scheduling techniques with better theoretical stability and performance guarantees should be investigated, and careful attention should be paid to the hidden coupling terms which may be introduced. Furthermore, anti-windup and anti-saturation control techniques could improve the autopilot response near the limits of the operating domain. In the longer term, linear parameter varying (LPV) control techniques with theoretical stability and performance guarantees may also be investigated. To this end, continuous-time state-space LPV representations of the system are interesting in that they keep the physical interpretation of the model parameters. However the associated identification techniques are much more complex, and several theoretical issues are still open.

Bibliography

- Albisser, M. *Identification of Aerodynamic Coefficients from Free-Flight Data*. Ph.D. thesis, Université de Lorraine, Nancy, France, 2015.
- ams AG. AS5045 12 Bit Programmable Magnetic Rotary Encoder. 2010. Rev. 1.7.
- Anderson, J. D. *Computational Fluid Dynamics: The Basics with Applications*. McGraw-Hill, 1995.
- Apkarian, P. and Noll, D. Nonsmooth \mathcal{H}_∞ Synthesis. *IEEE Transactions on Automatic Control*, 51(1):71–86, 2006. doi:10.1109/TAC.2005.860290.
- Apkarian, P., Biannic, J., and Gahinet, P. Self-Scheduled \mathcal{H}_∞ Control of Missile via Linear Matrix Inequalities. *Journal of Guidance, Control, and Dynamics*, 18(3):532–538, 1995. doi:10.2514/3.21419.
- Apkarian, P., Gahinet, P., and Buhr, C. Multi-Model, Multi-Objective Tuning of Fixed-Structure Controllers. In *2014 European Control Conference (ECC)*, pages 856–861. IEEE, 2014. doi:10.1109/ecc.2014.6862200.
- Åström, K. J. and Kumar, P. Control: A Perspective. *Automatica*, 50(1):3–43, 2014. doi:10.1016/j.automatica.2013.10.012.
- Åström, K. J. and Wittenmark, B. *Computer-Controlled Systems Theory and Design*. Dover, Mineola, NY, third edition, 2011.
- Bacic, M. On Hardware-in-the-Loop Simulation. In *IEEE Conference on Decision and Control and European Control Conference*, pages 3194–3198. 2005. doi:10.1109/CDC.2005.1582653.
- Banks, H. T., Cintrón-Arias, A., and Kappel, F. Parameter Selection Methods in Inverse Problem Formulation. In *Mathematical Modeling and Validation in Physiology*, volume 2064 of *Lecture Notes in Mathematics*, pages 43–73. Springer, 2012.
- Barbour, N. and Schmidt, G. Inertial Sensor Technology Trends. *IEEE Sensors Journal*, 1(4):332–339, 2001. doi:10.1109/7361.983473.
- Barlow, J. B., Rae, W. H., and Pope, A. *Low-Speed Wind Tunnel Testing, Third Edition*. Wiley New York, 1999.
- Bastable, M. J. From Breechloaders to Monster Guns: Sir William Armstrong and the Invention of Modern Artillery, 1854-1880. *Technology and Culture*, 33(2):213, 1992. doi:10.2307/3105857.

- Bates, D. and Postlethwaite, I. *Robust Multivariable Control of Aerospace Systems*. Delft University Press, Delft, 2002.
- Becklake, J. The V2 Rocket – a Convergence of Technologies? *Transactions of the Newcomen Society*, 67(1):109–123, 1995. doi:10.1179/tns.1995.005.
- Bellman, R. and Åström, K. J. On Structural Identifiability. *Mathematical Biosciences*, 7(3):329–339, 1970. doi:10.1016/0025-5564(70)90132-X.
- Bérard, C., Biannic, J.-M., and Saussié, D. *La commande multivariable. Application au pilotage d'un avion*. Dunod, Paris, 2012.
- Bernat, G., Burns, A., and Liamosi, A. Weakly Hard Real-Time Systems. *IEEE Transactions on Computers*, 50(4):308–321, 2001. doi:10.1109/12.919277.
- Bhagwandin, V. A. and Jubaraj, S. Numerical Prediction of Pitch Damping Stability Derivatives for Finned Projectiles. *Journal of Spacecraft and Rockets*, 51(5):1603–1618, 2014. doi:10.2514/1.A32734.
- Biannic, J.-M. *Commande robuste des systèmes à paramètres variables. Applications en Aéronautique*. Ph.D. thesis, École nationale supérieure de l'aéronautique et de l'espace, Toulouse, 1996.
- Biannic, J.-M., Burlion, L., Demourant, F., Ferreres, G., Loquen, T., and Roos, C. SMAC: A Toolbox for Systems Modeling, Analysis and Control with MATLAB/Simulink. 2016. Freely available for evaluation and academic use at: <http://w3.onera.fr/smac>.
- Bohlin, T. P. *Practical Grey-box Process Identification: Theory and Applications*. Springer Science & Business Media, 2006.
- Brown, J. H. and Martin, B. How Fast is Fast Enough? Choosing Between Xenomai and Linux for Real-Time Applications. In *12th OSADL Real-Time Linux Workshop*. 2010.
- Brun, R., Reichert, P., and Künsch, H. R. Practical Identifiability Analysis of Large Environmental Simulation Models. *Water Resources Research*, 37(4):1015–1030, 2001. doi:10.1029/2000wr900350.
- Bucher, R. and Balemi, S. Rapid Controller Prototyping with MATLAB/Simulink and Linux. *Control Engineering Practice*, 14(2):185–192, 2006. doi:10.1016/j.conengprac.2004.09.009.
- Burke, P. and Pergolizzi, A. XM1156 Precision Guidance Kit (PGK) Overview. In *53rd Annual Fuze Conference*. 2009.
- Burke, J. V., Henrion, D., Lewis, A. S., and Overton, M. L. Stabilization via Nonsmooth, Nonconvex Optimization. *IEEE Transactions on Automatic Control*, 51(11):1760–1769, 2006. doi:10.1109/TAC.2006.884944.
- Butler, H., Honderd, G., and van Amerongen, J. Model Reference Adaptive Control of a Direct-Drive DC Motor. *IEEE Control Systems Magazine*, 9(1):80–84, 1989. doi:10.1109/37.16756.
- Calise, A., Sharma, M., and Corban, J. An Adaptive Autopilot Design for Guided Munitions. In *AIAA Guidance, Navigation and Control Conference and Exhibit*. 1998. doi:10.2514/6.1998-4490.

- Campion, B. SPACIDO 1D Course Correction Fuze. In *51st Annual Fuze Conference*. Nashville, TN, 2007.
- Cantoni, M. and Vinnicombe, G. Quantifying Uncertainty and Robust Performance Using the ν -gap Metric. In *Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 1999. doi:10.2514/6.1999-4277.
- Carter, L. and Shamma, J. Gain-Scheduled Bank-to-Turn Autopilot Design Using Linear Parameter Varying Transformations. *Journal of Guidance, Control, and Dynamics*, 19(5):1056–1063, 1996. doi:10.2514/3.21745.
- Chung, C. Control System Synthesis Using BMI: Control Synthesis Applications. *International Journal of Control Automation and System*, 1(2):184–193, 2003.
- Corriveau, D., Wey, P., and Berner, C. Thrusters Pairing Guidelines for Trajectory Corrections of Projectiles. *Journal of Guidance, Control, and Dynamics*, 34(4):1120–1128, 2011. doi:10.2514/1.51811.
- Costello, M. and Agarwalla, R. Improved Dispersion of a Fin-Stabilized Projectile Using a Passive Moveable Nose. *Journal of Guidance, Control, and Dynamics*, 23(5):900–902, 2000. doi:10.2514/6.2000-4197.
- Demailly, H. *Identification des coefficients aérodynamiques d'un projectile à partir de mesures prises en vol*. Ph.D. thesis, Université d'Orléans, France, 2011.
- Denery, D. Simplification in the Computation of the Sensitivity Functions for Constant Coefficient Linear Systems. *IEEE Transactions on Automatic Control*, 16(4):348–350, 1971. doi:10.1109/tac.1971.1099742.
- Diebel, J. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. Technical report, Stanford University, 2006.
- Dobre, S. *Global Sensitivity and Identifiability Analyses. Application to Estimation of Photophysical Parameters in Photodynamic Therapy*. Ph.D. thesis, Université Henri Poincaré-Nancy I, 2010.
- Doyle, J. Analysis of Feedback Systems with Structured Uncertainties. *IEE Proceedings on Control Theory and Applications*, 129(6):242–250, 1982. doi:10.1049/ip-d.1982.0053.
- Doyle, J., Glover, K., Khargonekar, P., and Francis, B. State-Space Solutions to Standard \mathcal{H}_2 and \mathcal{H}_∞ Control Problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989. doi:10.1109/9.29425.
- Duc, G. *Commande \mathcal{H}_∞ et μ -analyse*. Hermès, 2000.
- Efron, B. and Tibshirani, R. J. *An Introduction to the Bootstrap*. Chapman et Hall, New York, 1993.
- English, B., Gadiraju, P., Rinehart, C., Glezer, A., and Allen, M. Gas Generator Actuator Arrays for Flight Control of Spinning Body Projectiles. In *19th IEEE International Conference on Micro Electro Mechanical Systems*, page 806–809. IEEE, Istanbul, Turkey, 2006. doi:memsys.2006.1627922.

- Ferreres, G. *A Practical Approach to Robustness Analyses*. Kluwer Academic / Plenum Publishers, 1999.
- Ferreres, G. and Biannic, J.-M. Reliable computation of the robustness margin for a flexible aircraft. *Control Engineering Practice*, 9(12):1267–1278, 2001. doi:10.1016/S0967-0661(01)00039-9.
- Franceschini, G. and Macchietto, S. Model-Based Design of Experiments for Parameter Precision: State of the Art. *Chemical Engineering Science*, 63(19):4846–4872, 2008. doi:10.1016/j.ces.2007.11.034.
- Franklin, G. F., Powell, J. D., and Workman, M. L. *Digital Control of Dynamic Systems*. Addison-Wesley, 1998.
- Fresconi, F., Cooper, G., Celmins, I., DeSpirito, J., and Costello, M. Flight Mechanics of a Novel Guided Spin-Stabilized Projectile Concept. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 226(3):327–340, 2011. doi:10.1177/0954410011408385.
- Fresconi, F., Celmins, I., Ilg, M., and Maley, J. Projectile Roll Dynamics and Control with a Low-Cost Skid-to-Turn Maneuver System. Technical Report ARL-TR-6363, Army Research Laboratory, 2013.
- Fresconi, F., Celmins, I., Ilg, M., and Maley, J. Projectile Roll Dynamics and Control with a Low-Cost Maneuver System. *Journal of Spacecraft and Rockets*, 51(2):624–627, 2014. doi:10.2514/1.A32553.
- Fresconi, F., Celmins, I., Sifton, S., and Costello, M. High Maneuverability Projectile Flight Using Low Cost Components. *Aerospace Science and Technology*, 41:175–188, 2015. doi:10.1016/j.ast.2014.12.007.
- Gahinet, P. and Apkarian, P. A Linear Matrix Inequality Approach to \mathcal{H}_∞ Control. *International Journal of Robust and Nonlinear Control*, 4(4):421–448, 1994. doi:10.1002/rnc.4590040403.
- Gahinet, P. and Apkarian, P. Structured \mathcal{H}_∞ Synthesis In MATLAB. In *18th IFAC World Congress*. Milan, Italy, 2011. doi:10.3182/20110828-6-it-1002.00708.
- Garza, F. R. and Morelli, E. A. A Collection of Nonlinear Aircraft Simulations in MATLAB. Technical Report NASA/TM-2003-212145, National Aeronautics and Space Administration, Langley Research Center, Hampton, VA, 2003.
- Gerum, P. Xenomai - Implementing a RTOS emulation framework on GNU/Linux. Technical report, <http://www.xenomai.org>, 2004.
- Grewal, M. S. and Andrews, A. P. How Good is Your Gyro? *IEEE Control Systems Magazine*, 30(1):12–86, 2010. doi:10.1109/MCS.2009.935122.
- Grewal, M. S., Andrews, A. P., and Bartone, C. G. *Global Navigation Satellite Systems, Inertial Navigation, and Integration*. Wiley, third edition, 2013.
- Gujarati, D. N. and Porter, D. *Basic Econometrics*. McGraw-Hill, fourth edition, 2004.
- Gumussoy, S., Henrion, D., Millstone, M., and Overton, M. L. Multiobjective Robust Control with HIFOO 2.0. In *6th IFAC Symposium on Robust Control Design*, pages 144–149. Israel, 2009. doi:10.3182/20090616-3-il-2002.00025. ArXiv: 0905.3229.

- Hann, C. E., Snowdon, M., Rao, A., Winn, O., Wongvanich, N., and Chen, X. Minimal Modelling Approach to Describe Turbulent Rocket Roll Dynamics in a Vertical Wind Tunnel. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 226(9):1042–1060, 2012. doi:10.1177/0954410011420771.
- Hanus, R., Kinnaert, M., and Henrotte, J.-L. Conditioning Technique, a General Anti-windup and Bumpless Transfer Method. *Automatica*, 23(6):729–739, 1987.
- Hecker, S. and Varga, A. Generalized LFT-Based Representation of Parametric Uncertain Models. *European Journal of Control*, 10:326–337, 2004. doi:10.3166/ejc.10.326-337.
- Hercog, D. and Jezernik, K. Rapid Control Prototyping Using MATLAB/Simulink and a DSP-Based Motor Controller. *International Journal of Engineering Education*, 21(4):596, 2005.
- Higgins, W. T. J. A Comparison of Complementary and Kalman Filtering. *IEEE Transactions on Aerospace and Electronic Systems*, AES-11(3):321–325, 1975. doi:10.1109/taes.1975.308081.
- Ho, D. W. C., Lam, J., and Chan, T. W. K. An Application of \mathcal{H}_∞ Design to Model-Following. *International Journal of Control*, 55(2):483–509, 1992. doi:10.1080/00207179208934250.
- Hull, D. G. *Fundamentals of Airplane Flight Mechanics*. Springer, 2007.
- Ilg, M. *Guidance, Navigation, and Control for Munitions*. Ph.D. thesis, Drexel University, 2008.
- Isermann, R., Schaffnit, J., and Sinsel, S. Hardware-in-the-Loop Simulation for the Design and Testing of Engine-Control Systems. *Control Engineering Practice*, 7(5):643–653, 1999. doi:10.1016/S0967-0661(98)00205-6.
- Jackson, P. B. Overview of Missile Flight Control Systems. *Johns Hopkins APL Technical Digest*, 29(1):9–24, 2010.
- Jaeggy, B. C. Description et caractéristiques de la soufflerie subsonique de $70 \times 90 \text{ cm}^2$ de l'ISL. Technical Report ISL-N 608/82, French-German Research Institute of Saint-Louis, 1982.
- Jategaonkar, R. V. *Flight Vehicle System Identification: A Time Domain Methodology*. American Institute of Aeronautics and Astronautics, Reston, VA, 2006.
- Katulka, G., Lyon, D., Fresconi, F., and Petrick, D. Development and Characterization of Low Cost Seeker Technology for US Army Applications. In *26th Army Science Conference*. 2008.
- Kharisov, E., Gregory, I. M., Cao, C., and Hovakimyan, N. \mathcal{L}_1 Adaptive Control Law for Flexible Space Launch Vehicle and Proposed Plan for Flight Test Validation. In *AIAA Guidance, Navigation and Control Conference*. 2008. doi:10.2514/6.2008-7128.
- Klatt, D., Hruschka, R., and Leopold, F. Numerical Investigation of the Magnus Effect of a Generic Projectile at Mach 3 up to 90° Angle of Attack. In *New Results in Numerical and Experimental Fluid Mechanics IX*, volume 124 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 513–521. Springer, 2014.

- Klein, V. and Morelli, E. A. *Aircraft System Identification: Theory and Practice*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Reston, VA, 2006.
- Kopp, C. Artillery for the Army: Precision Fire with Mobility. *DefenceTODAY Magazine*, 2005.
- Krneta, R., Antić, S., and Stojanović, D. Recursive Least Squares Method in Parameters Identification of DC Motor Models. *Facta Universitatis - series: Electronics and Energetics*, 18(3):467–478, 2005. doi:10.2298/fuee0503467k. University of Niš.
- Kwakernaak, H. Robust Control and \mathcal{H}_∞ -Optimization—Tutorial Paper. *Automatica*, 29(2):255–273, 1993. doi:10.1016/0005-1098(93)90122-a.
- Lambrechts, P., Terlouw, J., Bennani, S., and Steinbuch, M. Parametric Uncertainty Modeling using LFTs. In *Proceedings of the American Control Conference*. San Francisco, CA, 1993.
- Landers, M. G., Hall, L. H., Auman, L. M., and Vaughn, M. E. Deflectable Nose and Canard Controls for a Fin-Stabilized Projectile at Supersonic and Hypersonic Speeds. *21st AIAA Applied Aerodynamics Conference*, 2003. doi:10.2514/6.2003-3805.
- Lawrence, D. and Rugh, W. Gain Scheduling Dynamic Linear Controllers for a Nonlinear Plant. *Automatica*, 31(3):381–390, 1995. doi:10.1016/0005-1098(94)00113-w.
- Leith, D. and Leithead, W. Survey of Gain-Scheduling Analysis and Design. *International Journal of Control*, 73(11):1001–1025, 2000. doi:10.1080/002071700411304.
- Libsig, M. *Contrôle d'écoulements en vue d'un pilotage alternatif pour les projectiles d'artillerie*. Ph.D. thesis, University of Franche-Comté, 2016.
- Ljung, L. *System Identification - Theory for the User*. Prentice-Hall, 1999.
- MacKenzie, D. A. *Inventing Accuracy: A Historical Sociology of Nuclear Missile Guidance*. MIT Press, 1993.
- Marin, L., Valles, M., Soriano, A., Valera, A., and Albertos, P. Event-Based Localization in Ackermann Steering Limited Resource Mobile Robots. *IEEE/ASME Transactions on Mechatronics*, 19(4):1171–1182, 2014. doi:10.1109/TMECH.2013.2277271.
- McFarlane, D. and Glover, K. A Loop-Shaping Design Procedure Using H_∞ Synthesis. *Automatic Control, IEEE Transactions on*, 37(6):759–769, 1992. doi:10.1109/9.256330.
- McGowan, G. Z., Kurzen, M. J., Nance, R. P., and Carpenter, J. G. High Fidelity Approaches for Pitch Damping Prediction at High Angles of Attack. *Journal of Spacecraft and Rockets*, 51(5):1474–1484, 2014. doi:10.2514/1.A32625.
- Microchip Technology. dsPIC30F2011/2012/3012/3013 Datasheet. 2010.
- MicroStrain Inc. 3DM-GX3-25 Datasheet. 2012.
- Miotto, P. and Paduano, J. Application of Real Structured Singular Values to Flight Control Law Validation Issues. In *AIAA Guidance, Navigation and Control Conference*. 1995. doi:10.2514/6.1995-3190.

-
- Morrison, P. and Amberntson, D. Guidance and Control of a Cannon-Launched Guided Projectile. *Journal of Spacecraft and Rockets*, 14:328, 1977. doi:10.2514/3.57205.
- Mracek, C. and Ridgely, D. Missile Longitudinal Autopilots: Comparison of Multiple Three Loop Topologies. In *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 2005. doi:10.2514/6.2005-6380.
- Nichols, R., Reichert, R., and Rugh, W. Gain Scheduling for \mathcal{H}_∞ Controllers: a Flight Control Example. *IEEE Transactions on Control Systems Technology*, 1(2):69–79, 1993. doi:10.1109/87.238400.
- Nielsen, J. N. *Missile Aerodynamics*. AIAA, 1988.
- NXP Semiconductors. I²C Bus Specification and User Manual. 2014. UM10204. Rev. 6.
- Ollerenshaw, D., Costello, M., et al. Model Predictive Control of a Direct Fire Projectile Equipped with Canards. *Journal of Dynamic Systems, Measurement, and Control*, 130:061010, 2008. doi:10.1115/1.2957624.
- Omlin, M., Brun, R., and Reichert, P. Biogeochemical Model of Lake Zürich: Sensitivity, Identifiability and Uncertainty Analysis. *Ecological Modelling*, 141(1–3):105–123, 2001. doi:10.1016/S0304-3800(01)00257-5.
- Paull, L., Saeedi, S., Seto, M., and Li, H. Sensor-Driven Online Coverage Planning for Autonomous Underwater Vehicles. *IEEE/ASME Transactions on Mechatronics*, 18(6):1827–1838, 2013. doi:10.1109/TMECH.2012.2213607.
- Peeters, R. L. and Hanzon, B. Identifiability of Homogenous Systems Using the State Isomorphism Approach. *Automatica*, 41(3):513–529, 2005. doi:10.1016/j.automatica.2004.11.019.
- Perrin, M. Course Correction Fuzes Integration Technologies. In *55th Annual Fuze Conference*. 2011.
- Poussot-Vassal, C. *Robust LPV multivariable Automotive Global Chassis Control*. phdthesis, Institut National Polytechnique de Grenoble - INPG, 2008.
- Řezáč, M. and Hurák, Z. Structured MIMO \mathcal{H}_∞ design for dual-stage inertial stabilization: Case study for HIFOO and Hinfstruct solvers. *Mechatronics*, 23(8):1084–1093, 2013. doi:10.1016/j.mechatronics.2013.08.003.
- Rogers, J. and Costello, M. Control Authority of a Projectile Equipped with a Controllable Internal Translating Mass. *Journal of Guidance, Control, and Dynamics*, 31(5):1323, 2008. doi:10.2514/1.33961.
- Roos, C. and Biannic, J. M. A detailed comparative analysis of all practical algorithms to compute lower bounds on the structured singular value. *Control Engineering Practice*, 44:219–230, 2015. doi:10.1016/j.conengprac.2015.06.006.
- Rosema, C., Doyle, J., Auman, L., Underwood, M., and Blake, W. B. Missile DATCOM User’s Manual - 2011 Revision. Technical Report AFRL-RB-WP-TR-2011-3071, Air Force Research Laboratory, Air Vehicles Directorate, Wright-Patterson AFB, OH, USA, 2011.

- Ross, S. Gun Launched, Terminal Guided Projectile. 1963. Patent US3072055A.
- Rugh, W. J. and Shamma, J. S. Research on Gain Scheduling. *Automatica*, 36(10):1401–1425, 2000. doi:10.1016/s0005-1098(00)00058-3.
- Scherer, C. An Efficient Solution to Multi-Objective Control Problems with LMI Objectives. *Systems & Control Letters*, 40(1):43–57, 2000. doi:10.1016/s0167-6911(99)00122-x.
- Scherer, C., Gahinet, P., and Chilali, M. Multiobjective Output-Feedback Control via LMI Optimization. *IEEE Transactions on Automatic Control*, 42(7):896–911, 1997. doi:10.1109/9.599969.
- Schuette, H. and Waeltermann, P. Hardware-in-the-Loop Testing of Vehicle Dynamics Controllers – A Technical Survey. SAE Technical Paper 2005-01-1660, SAE International, Warrendale, PA, 2005.
- Sève, F., Theodoulis, S., Wernert, P., Zasadzinski, M., and Boutayeb, M. Pitch/yaw Channels Control Design for a 155 mm Projectile with Rotating Canards, Using a \mathcal{H}_∞ Loop-Shaping Design Procedure. In *AIAA Guidance, Navigation, and Control Conference*. 2014. doi:10.2514/6.2014-1474.
- Shamma, J. and Athans, M. Gain scheduling: Potential hazards and possible remedies. *Control Systems Magazine, IEEE*, 12(3):101–107, 1992.
- Shuster, M. D. A Survey of Attitude Representations. *Journal of the Astronautical Sciences*, 41(4):439–517, 1993.
- Silton, S. I. and Fresconi, F. Effect of Canard Interactions on Aerodynamic Performance of a Fin-Stabilized Projectile. *Journal of Spacecraft and Rockets*, 52(5):1430–1442, 2015. doi:10.2514/1.A33219.
- Simon, D. *Optimal State Estimation: Kalman, \mathcal{H}_∞ , and Nonlinear Approaches*. John Wiley & Sons, 2006.
- Siouris, G. M. *Missile Guidance and Control Systems*. Springer, New York, NY, 2004.
- Skogestad, S. and Postlethwaite, I. *Multivariable Feedback Control: Analysis and Design, Second Edition*, volume 2. John Wiley & Sons, 2007.
- Stevens, B. L. and Lewis, F. L. *Aircraft Control and Simulation*. John Wiley & Sons, 2003.
- Strub, G., Gassmann, V., Theodoulis, S., Dobre, S., and Basset, M. Hardware-In-the-Loop Experimental Setup Development for a Guided Projectile in a Wind Tunnel. In *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 458–463. Besançon, France, 2014. doi:10.1109/AIM.2014.6878120.
- Strub, G., Theodoulis, S., Gassmann, V., Dobre, S., and Basset, M. Pitch Axis Control for a Guided Projectile in a Wind Tunnel-based Hardware-In-the-Loop Setup. In *AIAA Modeling and Simulation Technologies Conference*. Kissimmee, FL, 2015a. doi:10.2514/6.2015-0153.
- Strub, G., Theodoulis, S., Gassmann, V., Dobre, S., and Basset, M. Pitch Axis Control for a Guided Projectile in a Wind Tunnel Hardware-in-the-Loop Setup. *Journal of Spacecraft and Rockets*, 52(6):1614–1626, 2015b. doi:10.2514/1.A33330.

-
- Strub, G., Dobre, S., Gassmann, V., Theodoulis, S., and Basset, M. Pitch Axis Identification for a Guided Projectile using a Wind Tunnel-based Experimental Setup. *IEEE/ASME Transactions on Mechatronics*, 21(3):1357–1365, 2016. doi:10.1109/tmech.2016.2525719.
- Tayfun, Ç. A Generic Approach to Missile Autopilot Design using State-Dependent Nonlinear Control. In *18th IFAC World Congress*, pages 9587–9600. 2011. doi:10.3182/20110828-6-IT-1002.03744.
- Texas Instruments. AM/DM73xx Technical Reference Manual. 2012. Rev. R.
- The Mathworks. Simulink Coder User’s Guide. 2012.
- Theodoulis, S. *Robust Control in a Nonlinear Context for Large Operating Domains*. Ph.D. thesis, Université Paris-Sud, 2008.
- Theodoulis, S., Sève, F., and Wernert, P. Robust Gain-Scheduled Autopilot Design for Spin-Stabilized Projectiles with a Course-Correction Fuze. *Aerospace Science and Technology*, 42:477–489, 2015. doi:10.1016/j.ast.2014.12.027.
- Titterton, D. and Weston, J. L. *Strapdown Inertial Navigation Technology*. IET, second edition, 2004.
- Vinnicombe, G. *Uncertainty and Feedback: \mathcal{H}_∞ Loop-shaping and the ν -gap Metric*. World Scientific, 2001.
- Vun, N., Hor, H. F., and Chao, J. W. Real-Time Enhancements for Embedded Linux. In *14th IEEE International Conference on Parallel and Distributed Systems*, pages 737–740. 2008. doi:10.1109/ICPADS.2008.108.
- Walter, E. and Pronzato, L. *Identification of Parametric Models from Experimental Data*. Springer, 1997.
- Weijers, S. R. and Vanrolleghem, P. A. A Procedure for Selecting Best Identifiable Parameters in Calibrating Activated Sludge Model No. 1 to Full-Scale Plant Data. *Water Science and Technology*, 36(5):69–79, 1997. doi:10.1016/s0273-1223(97)00463-0.
- Wells, L. L. The projectile GRAM SAASM for ERGM and Excalibur. In *IEEE Position Location and Navigation Symposium*, page 106–111. IEEE, 2000. doi:10.1109/PLANS.2000.838290.
- Wellstead, P. E. and Zarrop, M. B. *Self-Tuning Systems*. John Wiley & Sons, 1991.
- Wernert, P. and Theodoulis, S. Modelling and Stability Analysis for a Class of 155 mm Spin-stabilized Projectiles with Course Correction Fuse (CCF). In *AIAA Atmospheric Flight Mechanics Conference*. Portland, OR, 2011.
- Wilkie, D. F. and Perkins, W. R. Generation of Sensitivity Functions for Linear Systems Using Low-Order Models. *IEEE Transactions on Automatic Control*, AC-14(2):123–130, 1969. doi:10.1109/tac.1969.1099129.
- Wittenburg, J. *Dynamics of Multibody Systems*. Springer, second edition, 2008.
- Woodman, O. J. An Introduction to Inertial Navigation. Technical report, University of Cambridge, Computer Laboratory, 2007.

-
- Wu, F., Packard, A., and Balas, G. Systematic Gain-Scheduling Control Design: A Missile Autopilot Example. *Asian Journal of Control*, 4(3):341–347, 2002. doi:10.1111/j.1934-6093.2002.tb00362.x.
- Yaghmour, K. Adaptive Domain Environment for Operating Systems. Technical report, Opersys Inc, 2001.
- Zarchan, P. *Tactical and Strategic Missile Guidance*. Number 239 in Progress in Astronautics and Aeronautics. AIAA, sixth edition, 2012.
- Zhou, K. and Doyle, J. C. *Essentials of Robust Control*. Prentice Hall, 1998.
- Zipfel, P. H. *Modeling and Simulation of Aerospace Vehicle Dynamics*. American Institute of Aeronautics and Astronautics, 2007.
- Zoubir, A. M. and Boashash, B. The Bootstrap and its Application in Signal Processing. *IEEE Signal Processing Magazine*, 15(1):56–76, 1998. doi:10.1109/79.647043.

Appendix A

Modeling & Identification Complements

Contents

A.1 Attitude equations of motion	151
A.1.1 Roll Angular Dynamics	152
A.1.2 Pitch Angular Dynamics	152
A.1.3 Yaw Angular Dynamics	153
A.2 Inverse Trim Map Interpolation	155
A.3 Controllability and Observability of the Linearized Pitch-Yaw Model	155
A.4 <i>A priori</i> Identifiability of the Model Structure	156
A.5 Sensitivity Functions for a Linear System	157

This Appendix provides additional information regarding the derivation of the nonlinear model equations, the practical exploitation of the experimental trim maps, the proofs of the controllability, observability and identifiability properties of the linear model and a simple method for computing the sensitivity functions of a linear system.

A.1 Attitude equations of motion

The equations of motion describing the angular dynamics of the ACHILES projectile are presented in Section 2.3.1. In this case, the support structure prevents all translations of the projectile and the free-flight equations of motion no longer apply, as illustrated in Figure A.1. The presence of a kinematic chain between the projectile body and the Earth introduces joint reaction forces and moments into the dynamics equations. These reaction unknowns are eliminated by selecting bodies (or subsystems of bodies) so as to keep only one joint between the set of isolated bodies and other bodies in the system. For any given joint, the reaction effort corresponding to the joint's mobility is zero since the joints are assumed to be ideal (frictionless): e.g. for a revolute joint the reaction moment is zero along the joint axis. Then, the projection of Euler's and Newton's laws along the mobility axes does not contain reaction unknowns.

In the present configuration, the motion unknowns are the angular rate derivatives \dot{p} , \dot{q} and \dot{r} , therefore three independent equations are required to obtain the complete angular dynamics. The equations of motions are obtained by successively isolating the projectile ($\Sigma^{28} = B$), then the projectile with the

²⁸Here Σ denotes the set of bodies to which Euler's and Newton's laws are applied. For convenience, the individual bodies are designated with the same letter as their preferred coordinate system.

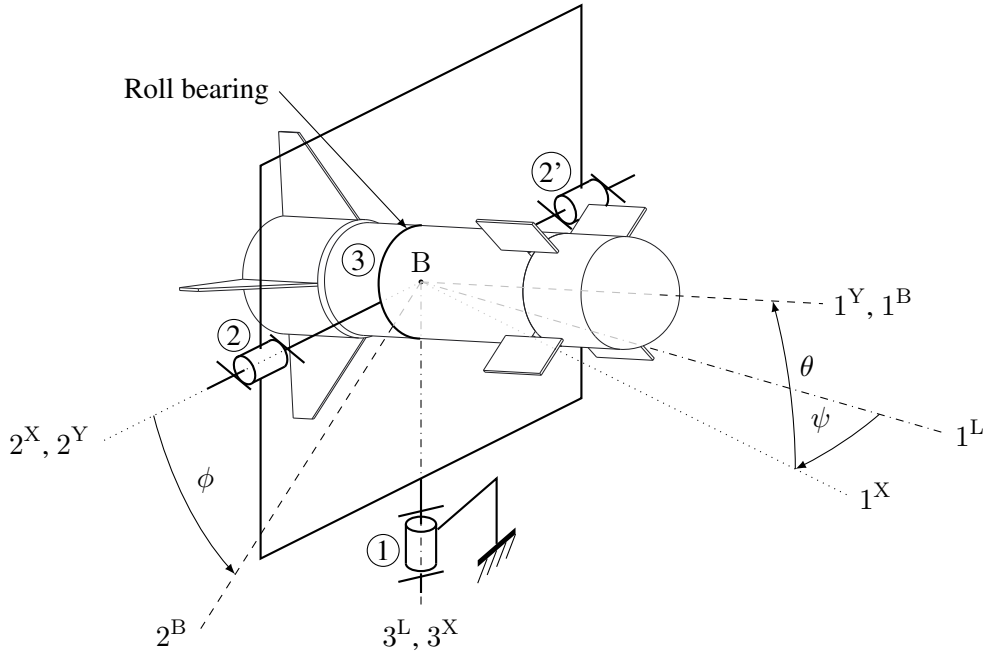


FIGURE A.1: Kinematics diagram of the ACHILES projectile and support structure.

inner gimbal ($\Sigma = B + Y$) and finally the complete system ($\Sigma = B + Y + X$), resulting in three scalar equations.

A.1.1 Roll Angular Dynamics

The first of these equations is obtained by isolating the projectile alone ($\Sigma = B$). This is similar to the situation described in Section 2.2.2, except the projectile body is also subject to the reaction moment of the roll bearing m_{B-Y} . Projecting Equation (2.9) along the roll axis 1^B eliminates the reaction unknown and yields Equation (2.16), which is still valid in the present case.

A.1.2 Pitch Angular Dynamics

The second equation is obtained by isolating the projectile body with the inner gimbal: $\Sigma = B + Y$. As the joint between the projectile and the inner gimbal is internal to the system, its reaction forces and moments do not appear in Newton's and Euler's equations. This subsystem is subject to the same efforts than the free-flying projectile, with the addition of the reaction forces and moments coming from the joint between the inner and the outer gimbal. Euler's law then becomes:

$$\mathcal{D}^E l_B^{BE} + \mathcal{D}^E l_B^{YE} = m_B + m_{Y-X} \quad (\text{A.1})$$

where m_{Y-X} is the reaction moment of the revolute joint between the inner and the outer gimbals. Since the inner gimbal's moments of inertia are negligible, the term $\mathcal{D}^E l_B^{YE}$ is assumed to be zero. The resulting form is then similar to Equation (2.5). The rate of change of the projectile's angular momentum, $\mathcal{D}^E l_B^{BE}$, is shifted to the body reference frame through the Euler transformation and expressed in the body

coordinate system:

$$[\mathbf{h}_B^{\text{BE}}]^B = [\mathcal{D}^B \mathbf{l}_B^{\text{BE}} + \boldsymbol{\Omega}^{\text{BE}} \mathbf{l}_B^{\text{BE}}]^B = \begin{bmatrix} \dot{p}I_1 + qr(I_3 - I_2) \\ \dot{q}I_2 + pr(I_1 - I_3) \\ \dot{r}I_3 + pq(I_2 - I_1) \end{bmatrix} \quad (\text{A.2})$$

As the reaction moment \mathbf{m}_{B-Y} is zero along 2^Y , the reaction unknowns are eliminated by projecting Euler's equation in the intermediate coordinate system $]^Y$ and taking the scalar product with 2^Y :

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^Y [\mathbf{T}]^{YB} [\mathbf{h}_B^{\text{BE}}]^B = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^Y [\mathbf{T}]^{YB} [\mathbf{m}_B]^B + \underbrace{\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^Y [\mathbf{m}_{B-Y}]^Y}_0 \quad (\text{A.3})$$

where $[\mathbf{T}]^{YB}$ is the transformation matrix from the body coordinate system to the intermediate coordinate system attached to the inner gimbal. Equation (A.3) then becomes:

$$[\dot{q}I_2 + pr(I_1 - I_3)] \cos \phi - [\dot{r}I_3 + pq(I_2 - I_1)] \sin \phi = m_{B_2} \cos \phi - m_{B_3} \sin \phi \quad (\text{A.4})$$

A.1.3 Yaw Angular Dynamics

The third equation is obtained by applying Euler's equation to the complete system, $\Sigma = B + Y + X$. As before, only externally applied efforts are considered when writing these equations. In this case the system is subject to the aerodynamic moments and the reaction moments in the revolute joint between the outer gimbal and the support frame. As the inner gimbal's inertial characteristics are negligible, Euler's second law can be written as:

$$\mathcal{D}^E \mathbf{l}_B^{\text{BE}} + \mathcal{D}^E \mathbf{l}_B^{\text{XE}} = \mathbf{m}_B + \mathbf{m}_{X-E} \quad (\text{A.5})$$

where \mathbf{m}_{X-E} is the reaction moment of the joint between the outer gimbal and the Earth frame. The rate of change of the outer gimbal's angular momentum is:

$$\mathbf{h}_B^{\text{XE}} = \mathcal{D}^E \mathbf{l}_B^{\text{XE}} = \mathcal{D}^E [\mathbf{I}_B^X \boldsymbol{\omega}^{\text{XE}}] \quad (\text{A.6})$$

where the angular velocity of the outer gimbal is $\boldsymbol{\omega}^{\text{XE}} = \dot{\psi} \mathbf{e}_3$. The expression of \mathbf{h}_B^{XE} in the local coordinate system is then:

$$\begin{aligned} [\mathbf{h}_B^{\text{XE}}]^L &= [\mathbf{I}_B^X]^L \left[\frac{d\boldsymbol{\omega}^{\text{XE}}}{dt} \right]^L \\ &= \ddot{\psi} I_{X_3} \end{aligned} \quad (\text{A.7})$$

The reaction moment is then eliminated by projecting Euler's equation (A.5) in the local coordinate system and taking the scalar product with the joint axis 3^L :

$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^L \left([\mathbf{h}_B^{\text{XE}}]^L + [\mathbf{T}]^{LB} [\mathbf{h}_B^{\text{BE}}]^B \right) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^L [\mathbf{T}]^{LB} [\mathbf{m}_B]^B + \underbrace{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^L [\mathbf{m}_{X-E}]^L}_0 \quad (\text{A.8})$$

which results in:

$$\ddot{\psi} I_{X_3} - [\dot{p}I_1 + qr(I_3 - I_2)] \sin \theta + [\dot{q}I_2 + pr(I_1 - I_3)] \cos \theta \sin \phi + [\dot{r}I_3 + pq(I_2 - I_1)] \cos \theta \cos \phi$$

$$= -m_{B_1} \sin \theta + m_{B_2} \cos \theta \sin \phi + m_{B_3} \cos \theta \cos \phi \quad (\text{A.9})$$

Combining Equations (2.16) and (A.9) eliminates the term in m_{B_1} , and regrouping the acceleration terms on the left-hand side yields:

$$\ddot{\psi} I_{X_3} \sec \theta + \dot{q} I_2 \sin \phi + \dot{r} I_3 \cos \phi = [m_{B_2} - pr(I_1 - I_3)] \sin \phi + [m_{B_3} - pq(I_2 - I_1)] \cos \phi \quad (\text{A.10})$$

This equation involves the second derivative of ψ , which is a function of the body rates and their derivatives. The projectile angular velocity $\boldsymbol{\omega}^{\text{BE}}$ is obtained by the following chain of rotations:

$$\boldsymbol{\omega}^{\text{BE}} = \dot{\psi} \mathbf{x}_3 + \dot{\theta} \mathbf{y}_2 + \dot{\phi} \mathbf{b}_1 \quad (\text{A.11})$$

Projecting $\boldsymbol{\omega}^{\text{BE}}$ in the body coordinate system then gives the correspondence between the time derivatives of the Euler angles and the body rates:

$$[\boldsymbol{\omega}^{\text{BE}}]^{\text{B}} = \begin{bmatrix} -\dot{\psi} \sin \theta + \dot{\phi} \\ \dot{\psi} \sin \phi \cos \theta + \dot{\theta} \cos \phi \\ \dot{\psi} \cos \phi \cos \theta - \dot{\theta} \sin \phi \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (\text{A.12})$$

Solving for $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$ gives the Euler angle differential equations:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ q \cos \phi - r \sin \phi \\ q \sin \phi / \cos \theta + r \cos \phi / \cos \theta \end{bmatrix} \quad (\text{A.13})$$

Taking the second derivative of ψ then yields:

$$\ddot{\psi} = \frac{1}{\cos \theta} [\dot{q} \sin \phi + \dot{r} \cos \phi + pq \cos \phi - pr \sin \phi + 2(q^2 - r^2) \tan \theta \cos \phi \sin \phi + 2qr \tan \theta (1 - 2 \sin^2 \phi)] \quad (\text{A.14})$$

The expressions of \dot{q} and \dot{r} are then obtained by replacing $\ddot{\psi}$ by its value in Equation (A.10), and then by combining it with Equation (A.4). In opposition to the free-flight case, these expressions are far from trivial due to the outer gimbal inertia acting on the yaw axis. Finally, the angular dynamics of the translation-denied projectile are given by:

$$\dot{p} = I_1^{-1} [(I_2 - I_3)qr + m_{B_1}] \quad (\text{A.15})$$

$$\dot{q} = \frac{1}{A_2} \left[k_2 \left(1 + \frac{I_{X_3} \cos^2 \phi}{I_3 \cos^2 \theta} \right) - k_3 \frac{I_{X_3} \sin \phi \cos \phi}{I_3 \cos^2 \theta} - k_\psi I_{X_3} \frac{\sin \phi}{\cos^2 \theta} \right] \quad (\text{A.16})$$

$$\dot{r} = \frac{1}{A_3} \left[-k_2 \frac{I_{X_3} \sin \phi \cos \phi}{I_2 \cos^2 \theta} + k_3 \left(1 + \frac{I_{X_3} \sin^2 \phi}{I_2 \cos^2 \theta} \right) - k_\psi I_{X_3} \frac{\cos \phi}{\cos^2 \theta} \right] \quad (\text{A.17})$$

where symbols A_2 , A_3 , k_2 , k_3 and k_ψ are defined as follows:

$$A_2 = I_2 + I_{X_3} \frac{1}{\cos^2 \theta} \left[\sin^2 \phi + \frac{I_2}{I_3} \cos^2 \phi \right] \quad (\text{A.18})$$

$$A_3 = I_3 + I_{X_3} \frac{1}{\cos^2 \theta} \left[\cos^2 \phi + \frac{I_3}{I_2} \sin^2 \phi \right] \quad (\text{A.19})$$

$$k_2 = m_{B_2} - pr(I_1 - I_3) \quad (\text{A.20})$$

$$k_3 = m_{B_3} - pq(I_2 - I_1) \quad (\text{A.21})$$

$$k_\psi = \frac{1}{\cos \theta} \left[pq \cos \phi - pr \sin \phi + 2(q^2 - r^2) \tan \theta \cos \phi \sin \phi + 2qr \tan \theta (1 - 2 \sin^2 \phi) \right] \quad (\text{A.22})$$

A.2 Inverse Trim Map Interpolation

The experimentally-determined, *stick-fixed* trim maps in Section 2.4.1 are discrete mappings from the sampled control deflections $(\bar{\delta}_m, \bar{\delta}_n)$ to the corresponding incidence angles $(\bar{\alpha}, \bar{\beta})$. However, for identification and control purposes it is more convenient to select the angles of incidence as trimming variables. In this case, the trim map provides the steady-state control deflections for given values of the incidence angles. The determination of this control-oriented trim map then consists in finding the inverse of the experimental trim map. In the direct case, the incidence angles for values of the fin deflections between the actually sampled points can easily be approximated using 2D interpolation. This is no longer true for the inverse mapping, which is determined as follows.

Let $\bar{\alpha}, \bar{\beta}$ be the desired angles of incidence and $\alpha = f_\alpha(\delta_m, \delta_n), \beta = f_\beta(\delta_m, \delta_n)$ the trim surfaces determined experimentally. For each angle of incidence $(\bar{\alpha}, \bar{\beta})$, find the fin deflections giving the desired value using the respective trim surface (f_α, f_β) . This corresponds to determining the following isocontour curves:

$$\mathcal{C}_\alpha = \{(\delta_m, \delta_n) | f_\alpha(\delta_m, \delta_n) = \bar{\alpha}\} \quad (\text{A.23})$$

$$\mathcal{C}_\beta = \{(\delta_m, \delta_n) | f_\beta(\delta_m, \delta_n) = \bar{\beta}\} \quad (\text{A.24})$$

The trim control deflections $(\bar{\delta}_m, \bar{\delta}_n)$ corresponding to the desired angles of incidence are then given by the coordinates of the intersection of \mathcal{C}_α and \mathcal{C}_β .

The obtained fin deflections can be validated using interpolation of the direct mapping to obtain the corresponding incidence angles. The implementation in MATLAB using the `contourc` and `interp2` functions leads to a less than 1% error between the desired incidence angles and the obtained incidence angles, which is largely sufficient for this application.

A.3 Controllability and Observability of the Linearized Pitch-Yaw Model

Controllability and observability are crucial properties of state-space systems for system identification and control. State controllability indicates whether the system input vector is able to move the state vector from any initial state to any final state in a finite time. The system is controllable if and only if the rank of the controllability matrix is equal to the number of states n (i.e. full rank), where the controllability matrix is defined as:

$$\mathcal{C} \triangleq \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \dots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \quad (\text{A.25})$$

Likewise, state observability indicates whether the state vector may be determined using the knowledge of the system outputs. The system is observable if and only if the observability matrix is full rank,

where the observability matrix is defined as:

$$\mathcal{O} \triangleq \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \dots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (\text{A.26})$$

These definitions are non-unique and are however limited to low-order systems due to numerical conditioning issues in the computation of the controllability and observability matrices.

The controllability matrix corresponding to the system of Equation (2.77) is:

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & M_{q\delta_m} & 0 \\ M_{q\delta_m} & 0 & M_{q\delta_m}M_{qq} & 0 \\ 0 & 0 & 0 & -\sec \alpha M_{r\delta_n} \\ 0 & M_{r\delta_n} & 0 & M_{r\delta_n}M_{rr} \end{bmatrix} \quad (\text{A.27})$$

which is full-rank provided $M_{q\delta_m}$ and $M_{r\delta_n}$ are non-zero.

The state vector \mathbf{x} in Equation (2.77) is directly measured using the IMU, resulting in the output matrix \mathbf{C} being identity. However, the angular rate measurements do not bring additional state information and the system remains observable when only the incidence angles are measured (corresponding to the first and third lines of the \mathbf{C} matrix). The observability matrix in this case is given by:

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\sec \alpha \end{bmatrix} \quad (\text{A.28})$$

which is full-rank. The angular rate measurements are however useful for control purposes since they avoid numerical derivatives of the incidence angles, which are more sensitive to noise.

A.4 *A priori* Identifiability of the Model Structure

The *a priori* identifiability of the model structure (2.77) is assessed in Section 2.5.3. As this model structure is observable and controllable, its *a priori* identifiability can be assessed using a similarity transformation approach [Walter and Pronzato, 1997]. Let $\hat{\mathbf{x}} = \mathbf{T}\mathbf{x}^*$, where \mathbf{T} is the invertible matrix of a state-space similarity transformation. A necessary and sufficient condition set for $M(\hat{\boldsymbol{\theta}}) = M(\boldsymbol{\theta}^*)$ is then

$$\begin{cases} \mathbf{A}(\hat{\boldsymbol{\theta}}) &= \mathbf{TA}(\boldsymbol{\theta}^*)\mathbf{T}^{-1}, \\ \mathbf{B}(\hat{\boldsymbol{\theta}}) &= \mathbf{TB}(\boldsymbol{\theta}^*), \\ \mathbf{C}(\hat{\boldsymbol{\theta}}) &= \mathbf{C}(\boldsymbol{\theta}^*)\mathbf{T}^{-1}, \\ \mathbf{D}(\hat{\boldsymbol{\theta}}) &= \mathbf{D}(\boldsymbol{\theta}^*), \\ \mathbf{x}_0(\hat{\boldsymbol{\theta}}) &= \mathbf{T}\mathbf{x}_0(\boldsymbol{\theta}^*) \end{cases} \quad (\text{A.29})$$

The state-space matrices defined in Eqs. (2.77) and (2.87) describe a structurally controllable and observable model structure for non-zero values of the input matrix coefficients ($M_{q\delta_m} \neq 0$ and $M_{r\delta_n} \neq 0$).

0), therefore the similarity transformation approach applies. Exploiting the structure of the observation matrix \mathbf{C} and the control matrix \mathbf{B} yields:

$$\left. \begin{array}{l} \mathbf{C}(\hat{\boldsymbol{\theta}})\mathbf{T} = \mathbf{C}(\boldsymbol{\theta}^*) \\ \mathbf{B}(\hat{\boldsymbol{\theta}}) = \mathbf{T}\mathbf{B}(\boldsymbol{\theta}^*) \end{array} \right\} \Rightarrow \mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ t_{21} & 1 & t_{23} & 0 \\ 0 & 0 & 1 & 0 \\ t_{41} & 0 & t_{43} & 1 \end{bmatrix} \quad (\text{A.30})$$

Applying the first condition $\mathbf{A}(\hat{\boldsymbol{\theta}}) = \mathbf{T}\mathbf{A}(\boldsymbol{\theta}^*)\mathbf{T}^{-1}$ yields:

$$\mathbf{A}(\hat{\boldsymbol{\theta}}) = \begin{bmatrix} -t_{21} & 1 & -t_{23} & 0 \\ M_{q\alpha} + \frac{t_{23}t_{41}}{\cos \alpha} & M_{qq} + t_{21} & M_{q\beta} + \frac{t_{23}t_{43}}{\cos \alpha} & -\frac{t_{23}}{\cos \alpha} \\ -t_{21}(M_{qq} + t_{21}) & 0 & -t_{23}(M_{qq} + t_{21}) & \frac{1}{\cos \alpha} \\ \frac{t_{41}}{\cos \alpha} & 0 & \frac{t_{43}}{\cos \alpha} & -\frac{1}{\cos \alpha} \\ M_{r\alpha} - t_{21}t_{41} & t_{41} & M_{r\beta} - t_{23}t_{41} & M_{rr} - \frac{t_{43}}{\cos \alpha} \\ -t_{41}(M_{rr} - \frac{t_{43}}{\cos \alpha}) & 0 & -t_{43}(M_{rr} - \frac{t_{43}}{\cos \alpha}) & 0 \end{bmatrix} \quad (\text{A.31})$$

In the state matrix \mathbf{A} , the coefficients of the first and third lines are zero, except for the second and fourth columns respectively. In application of the first condition, the remaining coefficients t_{21} , t_{23} , t_{41} , and t_{43} are then zero. The transformation matrix \mathbf{T} is then the identity matrix, therefore $\hat{\theta}_i = \theta_i^*$ and the structure is globally identifiable.

A.5 Sensitivity Functions for a Linear System

The *a posteriori* identifiability conditions presented in Section 2.5.4 involve the computation of the sensitivity functions of the model outputs *w.r.t.* changes in the parameter values. For a linear state-space system, these functions can be computed in a straightforward way by applying the definition of a sensitivity function to the output equation (assuming $\mathbf{D} = 0$):

$$s_{\mathbf{y}/\theta_i} = \frac{\partial \mathbf{y}}{\partial \theta_i} = \frac{\partial \mathbf{C}}{\partial \theta_i} \mathbf{x} + \mathbf{C} \frac{\partial \mathbf{x}}{\partial \theta_i} \quad (\text{A.32})$$

The partial derivative of the state \mathbf{x} with respect to parameter θ_i corresponds to the state sensitivity function:

$$s_{\mathbf{x}/\theta_i} = \frac{\partial \mathbf{x}}{\partial \theta_i} = \frac{\partial \mathbf{A}}{\partial \theta_i} \mathbf{x} + \mathbf{A} \frac{\partial \mathbf{x}}{\partial \theta_i} + \frac{\partial \mathbf{B}}{\partial \theta_i} \mathbf{u} \quad (\text{A.33})$$

The output sensitivity functions are then the outputs of a linear state-space system defined using the above equations. Since the system state \mathbf{x} is involved in the expressions of the sensitivity functions, the resulting system is of order $(n_p + 1) \cdot n$, with n the number of states in the original system. A major drawback of this simple method is thus the possibly high order of the sensitivity model. More sophisticated methods produce a lower-order sensitivity model by exploiting the properties of linear systems [Walter and Pronzato, 1997; Denery, 1971; Wilkie and Perkins, 1969]. However the above method is sufficient in the present case, as simulating the resulting 36th-order model is fast on modern computers.

Appendix B

Pitch-Only Autopilot

Contents

B.1	Synthesis Model	160
B.1.1	Nominal Model Selection	160
B.1.2	Uncertainty Modeling	160
B.2	Autopilot Structure	163
B.2.1	Rate Damping Controller	163
B.2.2	Disturbance Rejection Controller	164
B.2.3	Reference Tracking Controller	167
B.2.4	Multi-objective Synthesis	168
B.3	Controller Synthesis Results	169
B.3.1	Disturbance Rejection Controller	170
B.3.2	Reference Tracking	171
B.4	Robustness Analysis	172
B.5	Experimental Validation	174
B.5.1	Disturbance Rejection	175
B.5.2	Reference Tracking	175
B.6	Concluding Remarks	177

This Appendix presents a preliminary study concerning the single-axis case, which only considers the pitch motion of the projectile. This case has been presented in [Strub et al., 2015a,b] and is reproduced in greater detail here. This application considers the roll- and yaw-locked case, which has been presented and modeled in Section 2.3.4, where the airspeed is fixed to $V = 25$ m/s and the yaw axis remains locked at $\beta = 0$. This effectively reduces the flight envelope to a single dimension, where the only degree of freedom is on the angle of attack α . The corresponding model can be reduced to a single-input, single-output (SISO) model, allowing the comparison of different synthesis methods on a simpler case (*w.r.t.* the pitch/yaw case). The design goal is to obtain a low-order fixed-structure controller in order to simplify implementation as well as to prepare the ground for a future airspeed scheduling, while maintaining high performance levels.

B.1 Synthesis Model

The open-loop plant model \mathbf{G}_{ol} considered for synthesis is illustrated in Figure B.1 and consists in the series interconnection of the actuator model \mathbf{G}_A , given in Equation (1.5), and the airframe model \mathbf{G}_B , given in Equation (2.101); the sensor dynamics are neglected as the IMU is assumed to be ideal. The inputs to this model are the external disturbance d and the fin deflection command $\delta_{m,c}$. The disturbance mainly corresponds to unsteady aerodynamic effects and wind gusts, and is modeled as an input disturbance acting on the projectile body. Both inputs are scaled with respective gains $k_{sc,\delta_{m,c}}$, $k_{sc,d}$ such that the open-loop plant has unity steady-state gain. The outputs are the angle of attack α and the pitch rate q , which are measured by the IMU. The open-loop transfer function of the system is given by:

$$\mathbf{G}_{\text{ol}} : \begin{bmatrix} \alpha \\ q \end{bmatrix} = \mathbf{G}_B \begin{bmatrix} k_{sc,d} & k_{sc,\delta_{m,c}} \mathbf{G}_A \end{bmatrix} \begin{bmatrix} d \\ \delta_{m,c} \end{bmatrix} \quad (\text{B.1})$$

B.1.1 Nominal Model Selection

The selection of the nominal airframe model \mathbf{G}_B is carried out with the δ_ν -based procedure presented in Section 3.2.1. In the pitch-only case, the family of models $\mathbf{\Pi} = \{\mathbf{P}_i\}$ contains 8 linear models, which were estimated for values of the angle of attack α from 0° to 7° in 1° steps and for a constant airspeed $V = 25$ m/s (see Section 2.6.6).

Applying the first selection strategy to $\mathbf{\Pi}$ results in the mean-parameters model $\bar{\mathbf{P}}$, with $J(\bar{\mathbf{P}}) = 0.093$. The ν -gap distances of $\bar{\mathbf{P}}$ w.r.t. the members of $\mathbf{\Pi}$ are illustrated in Figure B.2B. Following the second strategy, the fifth model of the set is selected, with $J(\mathbf{P}_5) = 0.121$. The mutual ν -gap distance between any two models from $\mathbf{\Pi}$ is illustrated in Figure B.2A: the distance is lower for consecutive values of the angle of attack and higher between the extremes of the set. The frequency response of both nominal model candidates is compared against the family members on the Nyquist plot of Figure B.3. In this figure, both nominal model candidates have the same overall frequency response shape as other models in $\mathbf{\Pi}$ and are in the middle of the set of responses. As both strategies lead to low J values, the obtained models are good approximations of the overall behavior of the family, with an advantage for the mean-parameters model which has the lowest J value.

B.1.2 Uncertainty Modeling

The uncertain model is used for analyzing the robustness of the control laws and aims at capturing the various sources of uncertainty in the plant model. In the present case, the model parameters are subject to variations due to the non-linear, parameter-dependent nature of the system. They are also not known

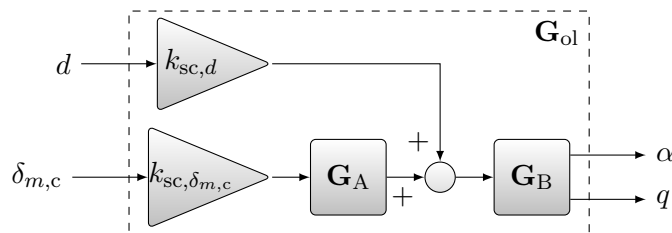


FIGURE B.1: Open-loop plant.

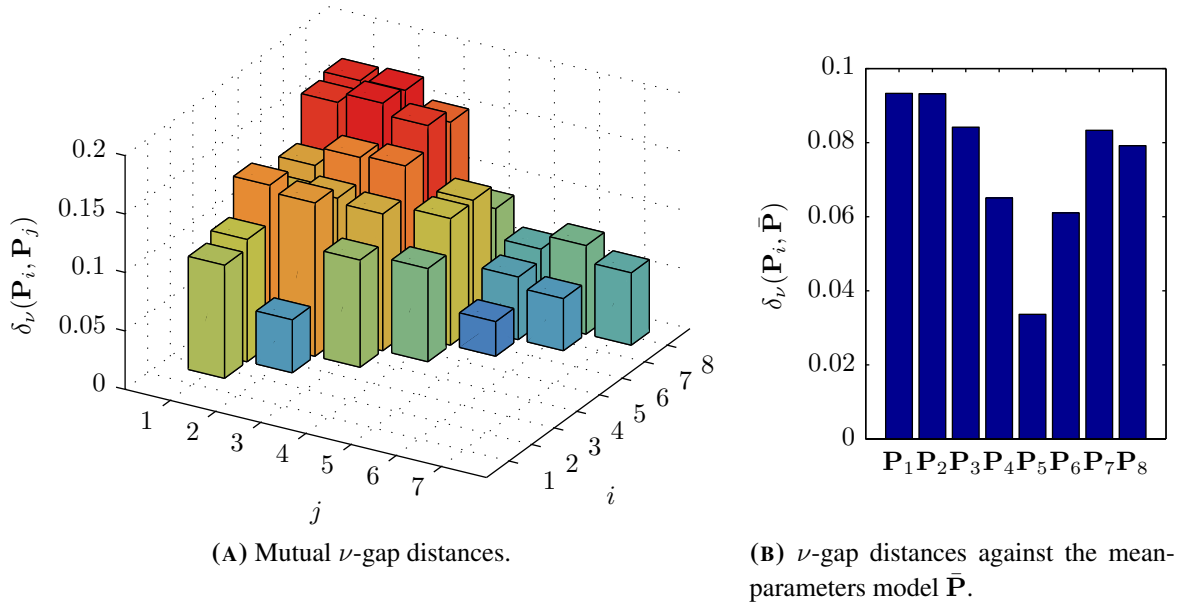


FIGURE B.2: Mutual ν -gap distances (A) for all estimated single-axis models at $V = 25$ m/s and (B) against the mean-parameters nominal model \bar{P} . For readability, only the upper diagonal part of the ν -gap distances matrix is shown, as $\delta_\nu(P_i, P_j) = \delta_\nu(P_j, P_i)$.

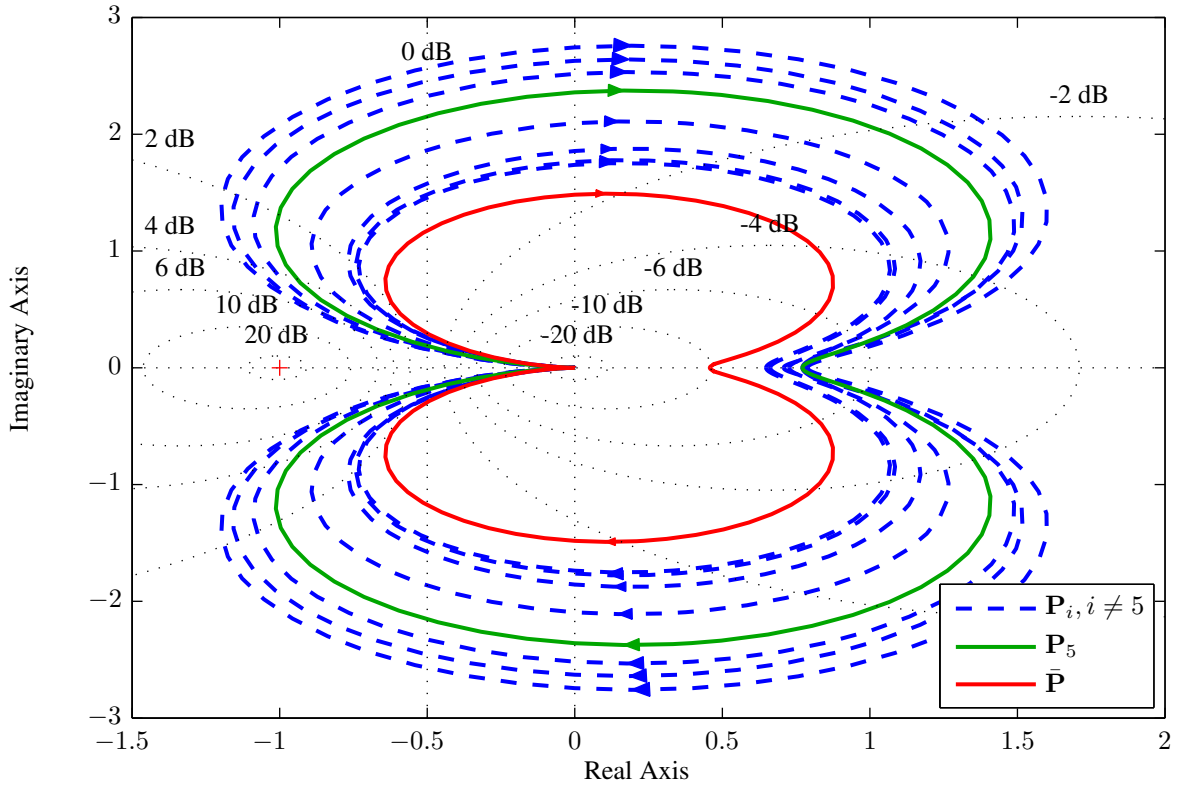


FIGURE B.3: Nyquist diagram of the set of estimated models and the two cases considered for the nominal model.

exactly but were estimated from experimental data. The plant uncertainty can then be described using only parametric uncertainty. However, the robust stability condition is generally much more complex for parametric uncertainty than for dynamic uncertainty. Moreover in the SISO case it may be interesting to approximate parametric uncertainty with a single lumped dynamic uncertainty, as it can reduce conservativeness [Skogestad and Postlethwaite, 2007]. Both approaches are discussed below.

Concerning the parametric uncertainty representation, the nominal values of the parameters correspond to the parameters of the nominal model. The variation range of each uncertain parameter corresponds to the minimum and maximum parameter values of models in the set Π^* . This set contains the estimated models Π as well as models issued from the bootstrap procedure, in order to take into account the uncertainty on each estimated model. The pitch-only linear model is governed by three parameters, leading to a 3×3 diagonal Δ matrix. This representation can be approximated by a single scalar uncertain perturbation using dynamic uncertainty. In the present case, the multiplicative input form illustrated in Figure 3.1 is selected. The multiplicative weight w_I is chosen as a second-order transfer function, which provides a good coverage of the uncertainty radius, as shown in Figure B.4.

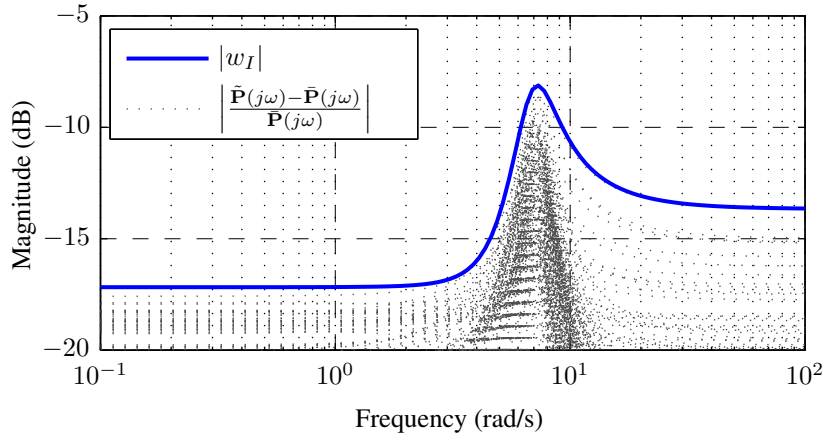


FIGURE B.4: Relative errors in the unstructured multiplicative uncertainty case and multiplicative weight w_I .

Both uncertainty representations are compared against the nominal model in Figure B.5. The response shown for the uncertain models corresponds to the worst-case (maximum) gain response, and

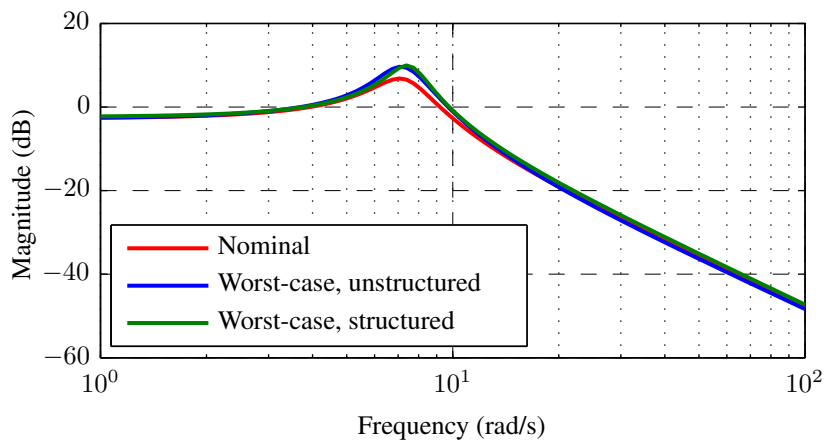


FIGURE B.5: Angle of attack response of the nominal model and the worst-case gain structured and unstructured uncertainty representations.

both uncertainty representations exhibit similar responses with the same degree of conservatism. The unstructured representation is preferred since the corresponding robust stability condition is much simpler than for parametric uncertainty, as discussed in Section 3.2.5.

B.2 Autopilot Structure

The overall structure of the pitch axis autopilot is illustrated in Figure B.6 and is inspired by autopilot structures found in the aircraft and missile literature [Zipfel, 2007; Bérard et al., 2012; Zarchan, 2012]. This autopilot is made up of three controllers, configured in two nested loops. The inner loop aims at improving the dynamic stability of the projectile in its pitching motion and is built around the rate damping gain k_q , acting on the pitch rate q . The outer loop consists in the feedback controller \mathbf{K}_{FB} acting on the angle of attack tracking error e . These two loops are responsible for the closed-loop stability and the rejection of external disturbances d . Finally, the feedforward filter \mathbf{K}_{FF} shapes the reference signal α_r to provide the desired closed-loop reference tracking behavior. In the present case, this behavior is defined by a second-order reference model \mathbf{T}_{ref} with bandwidth $\omega_0 = 3$ rad/s and damping $\zeta = 0.78$.

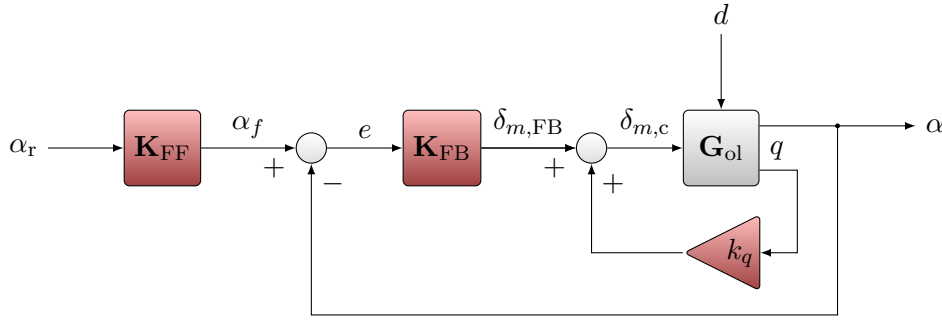


FIGURE B.6: Angle of attack autopilot structure.

Two \mathcal{H}_∞ synthesis methods for the feedback controller \mathbf{K}_{FB} are presented and compared in the sequel: a classical \mathcal{H}_∞ synthesis method resulting in a full-order controller, and a fixed-order, fixed-structured synthesis based on numerical optimization. The rate damping gain k_q is selected beforehand, as the full-order synthesis is unable to enforce a specific controller structure. In both cases, the feedforward filter is computed by interconnecting the reference model with the inverse of the closed-loop transfer function, and approximating the obtained transfer function with a low-order filter. In a third case, the feedback and feedforward controllers are synthesized in a single step using a multi-objective fixed-structure design technique.

B.2.1 Rate Damping Controller

One of the first functions of the autopilot is to improve the pitch-plane dynamic stability of the projectile. This function is accomplished by virtually increasing the aerodynamic damping derivative M_{qq} through the use of the rate feedback loop presented in Figure B.6 [Zipfel, 2007]. The airframe model \mathbf{G}_B can be decomposed in two transfers $\mathbf{G}_{B\alpha}$ and \mathbf{G}_{Bq} , with $\mathbf{G}_{Bq} = s\mathbf{G}_{B\alpha}$ from Equation (2.101). The pitch rate transfer function is then written as:

$$q = \mathbf{G}_{Bq} (\mathbf{G}_A k_{\text{sc},\delta_{m,c}} \delta_{m,c} + k_{\text{sc},d} d) \quad (\text{B.2})$$

The pitch rate q is then fed back to the actuator input and summed with the reference signal issued by the feedback controller \mathbf{K}_{FB} :

$$\delta_{m,c} = \delta_{m,\text{FB}} + k_q q \quad (\text{B.3})$$

Replacing $\delta_{m,c}$ in the pitch rate transfer function expression then yields the transfer function of the rate feedback loop:

$$q = \frac{\mathbf{G}_{\text{Bq}} \mathbf{G}_{\text{A}} k_{\text{sc},\delta_{m,c}}}{1 - k_q \mathbf{G}_{\text{Bq}} \mathbf{G}_{\text{A}} k_{\text{sc},\delta_{m,c}}} \delta_{m,\text{FB}} + \frac{\mathbf{G}_{\text{Bq}} k_{\text{sc},d}}{1 - k_q \mathbf{G}_{\text{Bq}} \mathbf{G}_{\text{A}} k_{\text{sc},\delta_{m,c}}} d \quad (\text{B.4})$$

where k_q effectively acts on the location of the closed-loop poles.

In the present case, the open-loop nominal plant is largely underdamped with a damping ratio of 0.17. Using a pole placement approach, the feedback gain k_q is selected such that the closed-loop damping approaches 0.7, while keeping the natural frequency of the actuator and body poles separated by a factor of at least three. The obtained rate damping gain is $k_q = 0.125$, achieving the desired closed-loop dominant pole damping. The poles of the resulting rate damped system are shown on Figure B.7.

The simulated and experimental responses of the closed loop against the open-loop response are presented in Figure B.8. Obviously, the rate damping controller can not constitute a tracking autopilot on its own, as plant gain uncertainty will lead to a static error, and the system output is very sensitive to disturbances. Nevertheless, the system damping is vastly improved and there is an overall good agreement between the simulated and the experimentally observed responses. The difference between these two responses is caused by aerodynamic disturbances on the real system, which are not present in the simulation.

B.2.2 Disturbance Rejection Controller

In the autopilot structure presented in Figure B.6, the feedback controller \mathbf{K}_{FB} is entirely responsible for rejection of external disturbances, as the feedforward controller \mathbf{K}_{FF} does not act on the tracking error. This paragraph focuses on the design of \mathbf{K}_{FB} , using modern robust \mathcal{H}_∞ control techniques in a S/KS-like mixed-sensitivity setup.

The \mathcal{H}_∞ design problem is represented using the standard form interconnection illustrated in Figure B.9. In this representation, the design objectives and constraints are included in the augmented plant \mathbf{P} , of which the inputs are the exogenous signal $w = d$ (disturbance) and the control input $u = \delta_{m,c}$. The outputs of \mathbf{P} are the performance signals $z = [z_1 \ z_2]^\top$, which are to be minimized, and the measured variable $v = -\alpha$. The \mathcal{H}_∞ control problem then consists in finding \mathbf{K}_{FB} , such that the system is nominally stable and

$$\|\mathbf{T}_{w \rightarrow z}\|_\infty < \gamma, \quad \gamma > 0 \quad (\text{B.5})$$

where $\mathbf{T}_{w \rightarrow z}$ is the transfer function from w to z , which is obtained by the lower linear fractional transformation (LFT) $\mathbf{T}_{w \rightarrow z} = \mathcal{F}_l(\mathbf{P}, \mathbf{K}_{\text{FB}})$.

The external disturbance d is modeled as an input disturbance to the airframe model \mathbf{G}_{B} , which exhibits a low-pass behavior thus resulting in low-frequency perturbations of the plant output. The disturbance will then be effectively rejected if the sensitivity function $\mathbf{S}_{d \rightarrow e}$, which is the transfer function between the disturbance d and the regulation error e , presents small gain at low frequencies [Skogestad and Postlethwaite, 2007, p. 359]. Furthermore, the actuators have a limited bandwidth and saturation-free amplitude range, while the open-loop plant may contain unmodeled high-frequency dynamics. In

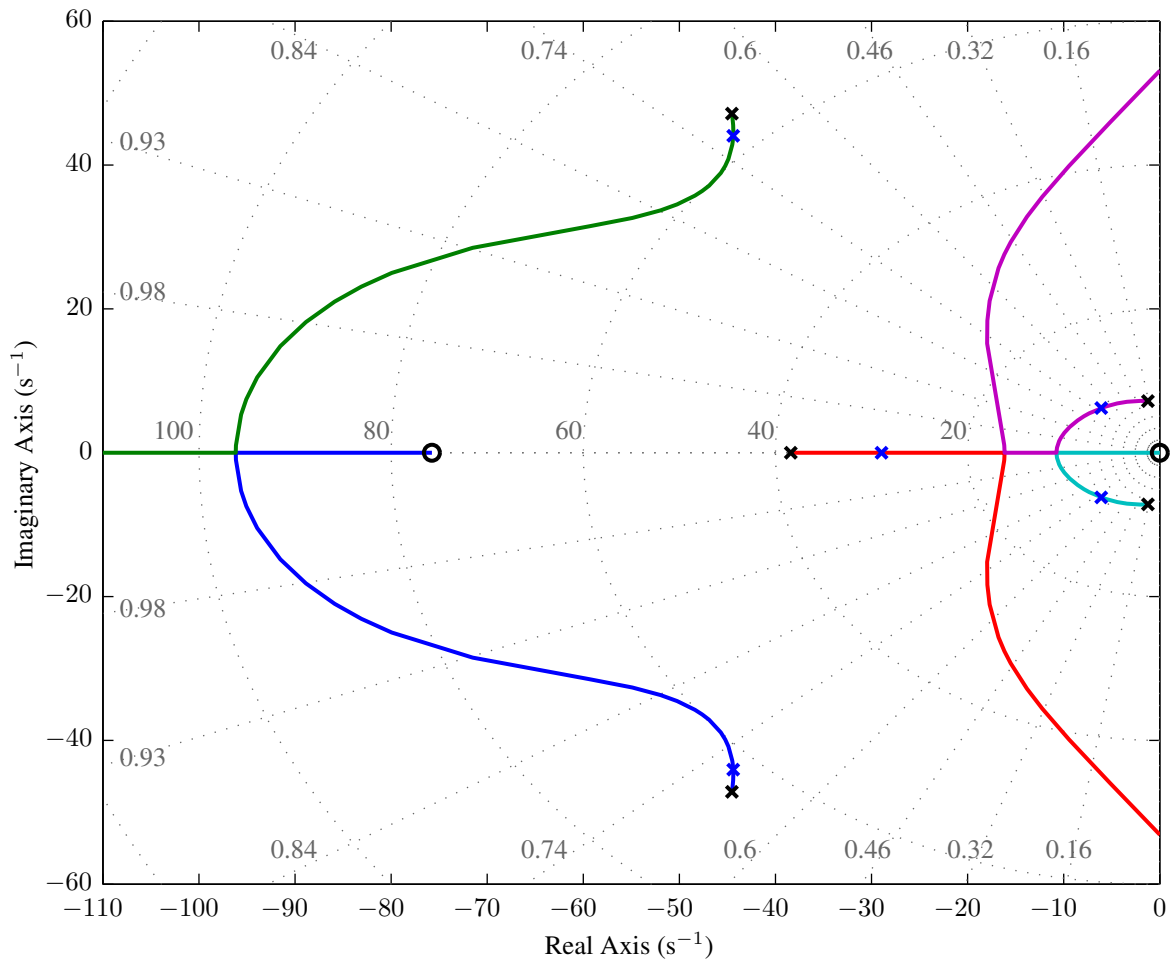


FIGURE B.7: Pole loci for the open-loop plant G_{ol} . Black crosses mark the location of open-loop G_{ol} plant poles. Bold blue crosses mark the location of rate damped plant poles.

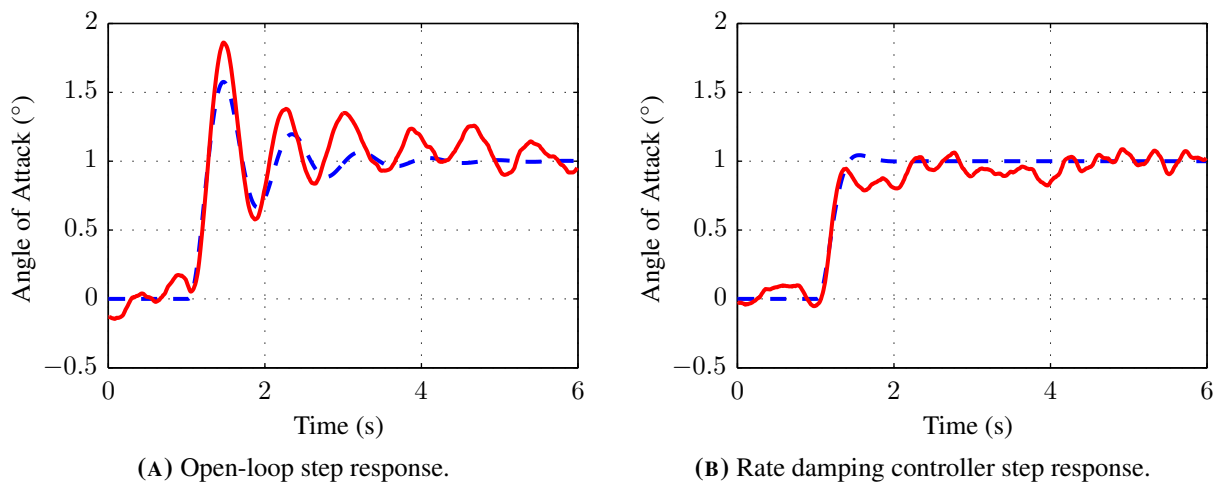


FIGURE B.8: Open-loop and rate damping controller step responses. Dashed blue line: simulated response, solid red line: measured response.

and open-loop stability margins. The performance transfer function $\mathbf{T}_{w \rightarrow z}$ is then written as:

$$\mathbf{T}_{w \rightarrow z} = \begin{bmatrix} \mathbf{S}_{d \rightarrow e} \cdot \mathbf{W}_1 \\ \mathbf{T}_{d \rightarrow \delta_{m,c}} \cdot \mathbf{W}_2 \end{bmatrix} \quad (\text{B.7})$$

As a first approach, the \mathcal{H}_∞ problem of Equation (B.5) is first solved using classical techniques, which consist in solving a Riccati equation [Doyle et al., 1989] or an LMI formulation [Gahinet and Apkarian, 1994], and result in a full-order controller structure. The high-frequency modes are subsequently eliminated using model reduction techniques, and the controller is adjusted so that the lowest frequency pole is moved to zero, thereby eliminating steady-state error. However, full-order controllers are usually not suitable for implementation on low-power control electronics, and are difficult to integrate in gain-scheduling schemes. Here, this controller serves as a performance standard for comparison against reduced-complexity controllers.

In the second approach, a fixed-order, fixed-structure technique is employed, resulting in a reduced controller order and hence a much simpler structure, which can be imposed by the designer. In guided projectile control design, PI-type structures have been used successfully in stabilizing and reference tracking controllers [Sève et al., 2014]. The PID controller provides additional damping through the derivative term, however the derivative must be approximated using a low-pass filter such that the controller is proper and so as to limit the high-frequency gain. The fixed-structure controller $\mathbf{K}_{\text{FB,FS}}$ to be synthesized here is then of the form:

$$\mathbf{K}_{\text{FB,FS}} = k_p + k_i \frac{1}{s} + \frac{k_d s}{T_f s + 1} \quad (\text{B.8})$$

with parameters k_p , k_i , k_d , and T_f to be determined. The \mathcal{H}_∞ problem is then non-convex, and the nonsmooth \mathcal{H}_∞ synthesis method by Apkarian and Noll [2006] is employed to compute the fixed-order, fixed-structure controller $\mathbf{K}_{\text{FB,FS}}$.

B.2.3 Reference Tracking Controller

The above designed feedback controllers are focused on disturbance rejection. Even if the closed-loop system has no steady-state error, the time response does not conform to the specified requirements because the complete system should behave like the reference model \mathbf{T}_{ref} . A straightforward approach for designing the feedforward filter \mathbf{K}_{FF} consists in connecting the reference model and the inverse of the closed-loop transfer function $\mathbf{T}_{\alpha_f \rightarrow \alpha}$ in series [Skogestad and Postlethwaite, 2007], where $\mathbf{T}_{\alpha_f \rightarrow \alpha}$ is the transfer function of the closed-loop system with the disturbance rejection controller:

$$\mathbf{K}_{\text{FF}} = \mathbf{T}_{\text{ref}} \mathbf{T}_{\alpha_f \rightarrow \alpha}^{-1} \quad (\text{B.9})$$

Obviously, the resulting system is non-proper and cannot be directly implemented on the real system. The approach employed to circumvent this limitation consists in approximating the magnitude frequency response of the ideal feedforward on a specific frequency band with a fixed-order transfer function. As the ideal feedforward filter behaves as a derivative filter at high frequencies, it is desirable to use a strictly proper transfer function, so as not to excite the high-frequency dynamics of the closed-loop system. The frequency band must also be chosen with care as the high-frequency dynamics of the ideal feedforward

filter may not be relevant and usually impose a higher-order transfer function and high gains. In the present case, a second-order low-pass filter with one zero provides satisfying performance with low complexity.

The selection of these parameters therefore results in a trade-off between model-matching accuracy (minimizing the difference between the reference model and the closed-loop behavior) and controller/actuator effort. However, this inverse-based design procedure may result in a too high model-matching error for a low-order feedforward filter, and the designer's actions are quite limited. Moreover, while the disturbance rejection controller satisfies the constraints on the actuator control signal, the closed-loop system with feedforward may violate these specifications. It is more desirable to design the feedback and the feedforward controllers in a single step, resulting in a 2-DoF controller such that all constraints are applied during synthesis, as presented in the next subsection.

B.2.4 Multi-objective Synthesis

A naive approach to the design of the 2-DoF controller would be to add the feedforward controller and associated model-matching and controller effort constraints to the mixed-sensitivity problem of Figure B.9. The exogenous input vector w then consists in the reference input r as well as the disturbance input d . However, it is not possible to define the weighting filters \mathbf{W}_i for individual transfers, as all inputs are used during synthesis. For example, the performance signal relative to the model-matching error would include transfers from both inputs, when only the reference is relevant. This limits the designer's actions and leads to an unsatisfactory result.

As the simultaneous feedback and feedforward design problem addresses different objectives, the multi-objective design techniques presented in Section 3.2.4 are more suitable. In the present case, the control objectives are specified on separate $w_j \rightarrow z_j$ channels, as illustrated in Figure B.10. The first channel, $w_1 \rightarrow z_1$, deals with the disturbance rejection requirements, which were specified in Section

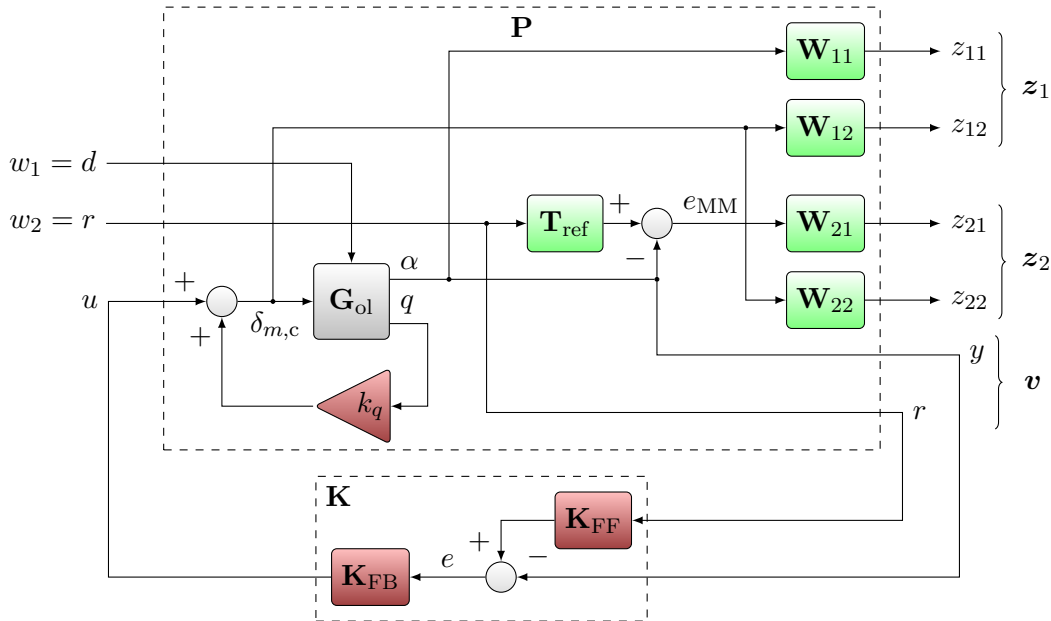


FIGURE B.10: Multi-objective two-degrees of freedom synthesis diagram in standard form.

B.2.2. The reference tracking requirements, namely a model-matching constraint with respect to \mathbf{T}_{ref} and a high-frequency roll-off constraint on $\mathbf{T}_{r \rightarrow \delta_{m,c}}$, which limits the injection of high-frequency dynamics in the plant from the reference input, are specified in the second channel $w_2 \rightarrow z_2$, where $w_2 = r$. These requirements are enforced by the means of the respective weighting filters \mathbf{W}_{21} and \mathbf{W}_{22} . The model-matching filter \mathbf{W}_{21} is a constant gain and acts on the model-matching error $e_{\text{MM}} = r(\mathbf{T}_{\text{ref}} - \mathbf{T}_{r \rightarrow \alpha})$. The gain of \mathbf{W}_{21} defines the maximum allowable difference between the target system and the closed-loop transfer function, which is the highest at medium frequencies. The roll-off filter \mathbf{W}_{22} acts on the transfer $\mathbf{T}_{r \rightarrow \delta_{m,c}}$ and is specified in the same way as in the disturbance rejection case. This filter mostly acts on the feedforward filter and its bandwidth may be set higher than the disturbance rejection roll-off filter. Finally, the performance transfer function $\mathbf{T}_{w \rightarrow z}$ is written as:

$$\mathbf{T}_{w \rightarrow z} = \left[\begin{array}{c|c} \mathbf{T}_{w_1 \rightarrow z_1} & \mathbf{T}_{w_2 \rightarrow z_1} \\ \hline \mathbf{T}_{w_1 \rightarrow z_2} & \mathbf{T}_{w_2 \rightarrow z_2} \end{array} \right] = \left[\begin{array}{c|c} \mathbf{W}_{11} \mathbf{S}_{d \rightarrow e} & \mathbf{W}_{11} (1 - \mathbf{T}_{r \rightarrow y}) \\ \mathbf{W}_{12} \mathbf{T}_{d \rightarrow \delta_{m,c}} & \mathbf{W}_{12} \mathbf{T}_{r \rightarrow \delta_{m,c}} \\ \hline \mathbf{W}_{21} \mathbf{S}_{d \rightarrow e} & \mathbf{W}_{21} (\mathbf{T}_{\text{ref}} - \mathbf{T}_{r \rightarrow \alpha}) \\ \mathbf{W}_{22} \mathbf{T}_{d \rightarrow \delta_{m,c}} & \mathbf{W}_{22} \mathbf{T}_{r \rightarrow \delta_{m,c}} \end{array} \right] \quad (\text{B.10})$$

However, off-diagonal transfers $\mathbf{T}_{w_2 \rightarrow z_1}$ and $\mathbf{T}_{w_1 \rightarrow z_2}$ are not relevant to the corresponding control objectives and hence should not be taken into consideration for synthesis. In order to consider only the diagonal transfers, the 2-DoF synthesis problem can be formulated as a multi-objective \mathcal{H}_∞ problem: given $\gamma > \gamma_{\min}$, find all stabilizing controllers $\mathbf{K} = \{\mathbf{K}_{\text{FB,MO}}, \mathbf{K}_{\text{FF,MO}}\}$ such that:

$$\max \{ \|\mathbf{T}_{w_1 \rightarrow z_1}(\mathbf{P}, \mathbf{K})\|_\infty, \|\mathbf{T}_{w_2 \rightarrow z_2}(\mathbf{P}, \mathbf{K})\|_\infty \} < \gamma \quad (\text{B.11})$$

For the sake of readability, in this diagram the disturbance rejection requirement is specified on $\mathbf{S}_{d \rightarrow y}$, which is identical to $\mathbf{S}_{d \rightarrow e}$ except for the sign in the disturbance rejection case ($r = 0$). The 2-DoF controller to be synthesized is composed of the feedback controller $\mathbf{K}_{\text{FB,MO}}$ and the feedforward controller $\mathbf{K}_{\text{FF,MO}}$, where the structure of $\mathbf{K}_{\text{FB,MO}}$ is the same as $\mathbf{K}_{\text{FB,FS}}$, as defined in Equation (B.8). In this diagram, the feedforward controller is implemented in filter form, acting on the reference signal before the error summing junction, and in first approach this filter has the same structure as in the case discussed in Section B.2.3 (inversion-based design). This structure is further reduced to a first-order lag-lead filter, showing similar performance with a lower order controller.

B.3 Controller Synthesis Results

The autopilot design constraints are implemented using the weighting filters \mathbf{W}_i in Figure B.9 and \mathbf{W}_{ij} in Figure B.10. In order to compare the three proposed synthesis methods, the weighting filters are identical across these three methods, and the parameters are summarized in Table B.1. For the fixed-structure and the multi-objective synthesis methods, the feedback controller is a PID controller as shown in Equation (B.8). In all three cases, the feedforward controller is selected to be a two-pole, one-zero minimum-phase transfer function.

TABLE B.1: Parameters of the weighting filters \mathbf{W}_i and \mathbf{W}_{ij} .

$\mathbf{W}_1 = \mathbf{W}_{11}$		$\mathbf{W}_2 = \mathbf{W}_{12}$		\mathbf{W}_{21}		\mathbf{W}_{22}	
$k_{11,\text{LF}}$	0.001	$k_{12,\text{LF}}$	1	k_{21}	20	$k_{22,\text{LF}}$	1
$k_{11,\text{HF}}$	0.6	$k_{12,\text{HF}}$	0.001			$k_{22,\text{HF}}$	0.001
ω_{11}	3 rad/s	ω_{12}	15 rad/s			ω_{22}	15 rad/s

B.3.1 Disturbance Rejection Controller

The feedback controllers \mathbf{K}_{FB} were synthesized using \mathcal{H}_∞ techniques, which led to similar final gain values for all three methods, with $\gamma = 1.45$ for the full-order controller, $\gamma = 1.54$ for the fixed-structure and $\gamma = 1.57$ for the multi-objective 2-DoF synthesis. The sensitivity transfer functions with respect to the disturbance input, as well as the controller Bode diagram and the closed-loop system response to a unit disturbance step are illustrated in Figure B.11. In these plots, the solid red and blue lines correspond respectively to the 1-DoF full-order and fixed-structure controllers $\mathbf{K}_{\text{FB,FO}}$ and $\mathbf{K}_{\text{FB,FS}}$, and the green line corresponds to the feedback controller $\mathbf{K}_{\text{FB,MO}}$ computed using the multi-objective procedure. In Figure B.11A and B.11B, the dashed black line corresponds to the inverse of the weighting filters.

As specified in Section B.3, all feedback controllers were synthesized against the same requirements. The full-order controller $\mathbf{K}_{\text{FB,FO}}$ corresponds to the best achievable result, as there are no constraints on its structure. Nevertheless, it can be seen that both fixed-structure controllers exhibit similar performance, despite a less complex structure. As expected, the error sensitivity $\mathbf{S}_{d \rightarrow e}$ tends to zero at low frequencies

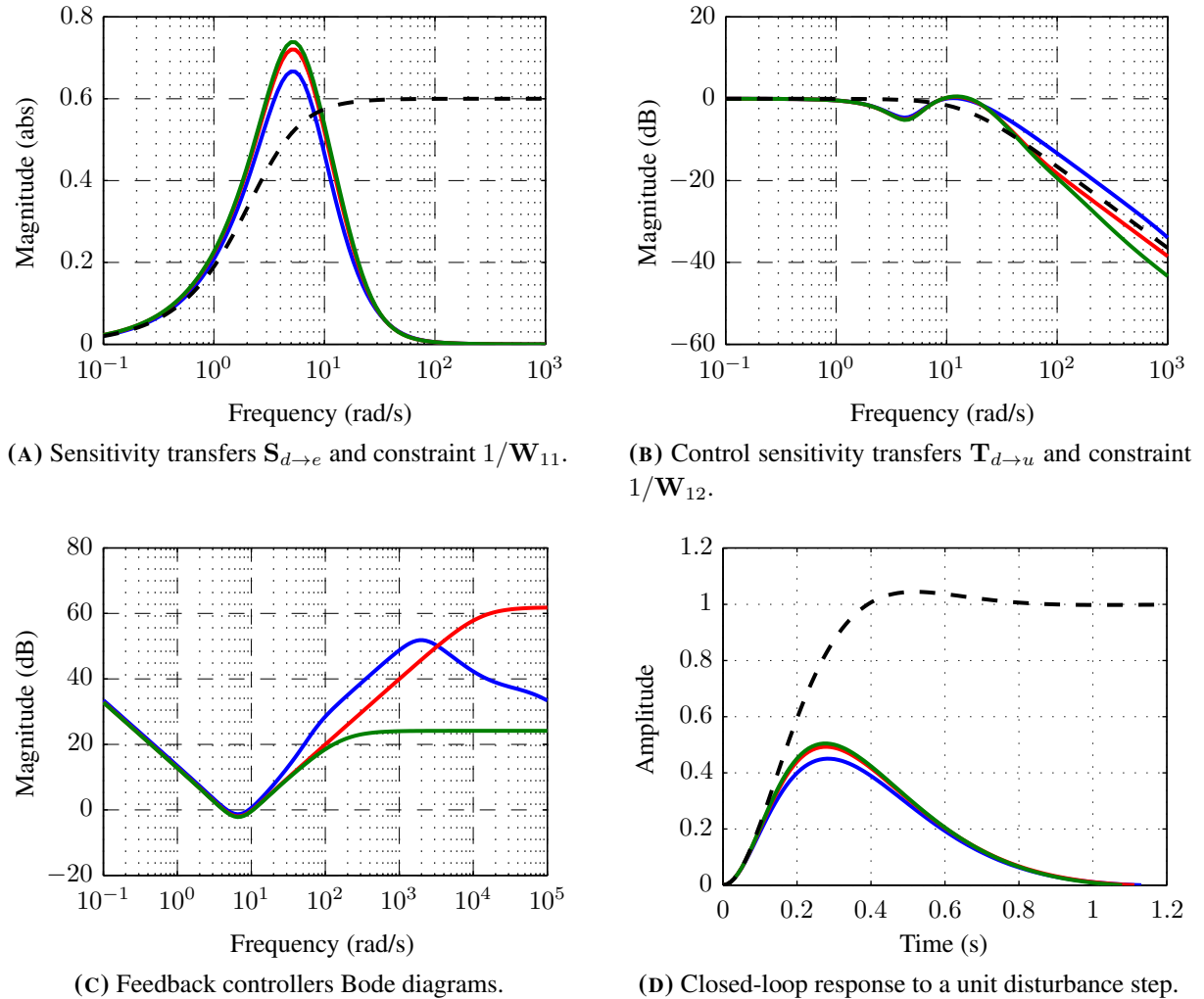


FIGURE B.11: Closed-loop transfer functions and unit step response for the disturbance rejection controllers (blue: full-order controller $\mathbf{K}_{\text{FB,FO}}$, red: fixed-structure controller $\mathbf{K}_{\text{FB,FS}}$, green: fixed-structure controller $\mathbf{K}_{\text{FB,MO}}$ in the multi-objective case).

but slightly exceeds the upper limit specified in \mathbf{W}_{11} , and the control sensitivity $\mathbf{T}_{d \rightarrow u}$ tends to unity at low frequencies and zero at high frequencies. However, the controllers do not roll-off at high frequencies, as shown on Figure B.11C. This is not an issue for disturbance rejection, as the disturbance signal d is low-pass filtered by the airframe model \mathbf{G}_B . On the other hand, the spectral content of reference signal r must be limited in order to limit the injection of high-frequency content in the feedback loop. This is guaranteed by the roll-off action of the feedforward controller \mathbf{K}_{FF} .

Finally, Figure B.11D presents the closed-loop system response to a unity disturbance step against the open-loop damped system, of which the response is represented by a black dashed line. The maximum deviation due to the disturbance is less than 0.5° , and the system reaches steady-state after about 1 second, with no static error, while the open-loop system is unable to reject the disturbance.

B.3.2 Reference Tracking

The closed-loop transfer functions relative to the reference tracking objective, as well as the feedforward controller Bode diagrams and the autopilot step response, are illustrated in Figure B.12. These plots correspond to the three considered synthesis approaches: first the 1-DoF full-order and fixed-structure feedback controllers $\mathbf{K}_{FB,FO}$ and $\mathbf{K}_{FB,FS}$ with inverse-based feedforward controllers $\mathbf{K}_{FF,FO}$ and $\mathbf{K}_{FF,FS}$, respectively in solid blue and red, and second the fixed-structure feedback and feedforward controllers $\mathbf{K}_{FB,MO}$ and $\mathbf{K}_{FF,MO}$ computed using 2-DoF multi-objective synthesis in solid green.

In Figure B.12A and Figure B.12B, the model-matching error $e_{MM} = (\mathbf{T}_{ref} - \mathbf{T}_{r \rightarrow \alpha})r$ and control sensitivity transfer $\mathbf{T}_{r \rightarrow u}$ are compared to the respective design constraints $1/\mathbf{W}_{21}$ and $1/\mathbf{W}_{22}$. These figures show that only the multi-objective controller is able to fulfill both requirements, while both designs using an inversion-based feedforward controller exhibit higher model-matching error and do not roll off at high frequencies, violating the control effort constraint $1/\mathbf{W}_{22}$. It must however be noted that these constraints are only effective during multi-objective synthesis, and the results obtained using inverse-based feedforward controllers are here presented only for comparison.

Figure B.12C presents the Bode diagrams for the three feedforward controllers against the ideal, inverse-based filters given by Equation (B.9), which are represented by dashed lines. As expected, all ideal inverse-based feedforward controllers have a similar shape, except in the high frequencies. The controllers obtained through approximation exhibit an identical response, while the multi-objective feedforward controller rolls off at higher frequencies, which is the best trade-off between the model-matching and the control sensitivity constraints respectively enforced with \mathbf{W}_{21} and \mathbf{W}_{22} .

The inverse-based feedforward controllers $\mathbf{K}_{FF,FO}$ and $\mathbf{K}_{FF,FS}$ were designed according to Section B.2.3, using 2-pole, 1-zero transfer functions to approximate Equation (B.9) for frequencies ranging from 0.1 rad/s to 10 rad/s. As observed in Figure B.12C, the obtained filters fit the ideal inverse-based feedforward in the specified frequency band. These specifications were found using trial-and-error and result in a trade-off between controller complexity, model-matching error and controller effort. If a better conformance to the design constraints is required, the inverse-based design method is not as effective as multi-objective 2-DoF design, as in the latter case the requirements are taken into account during the synthesis of both controllers, while in the former case they are verified *a posteriori*.

Finally, the time-domain performance is assessed in Figure B.12D, by comparing the system response to a unit reference step against the response of the reference model \mathbf{T}_{ref} . All three designs perform

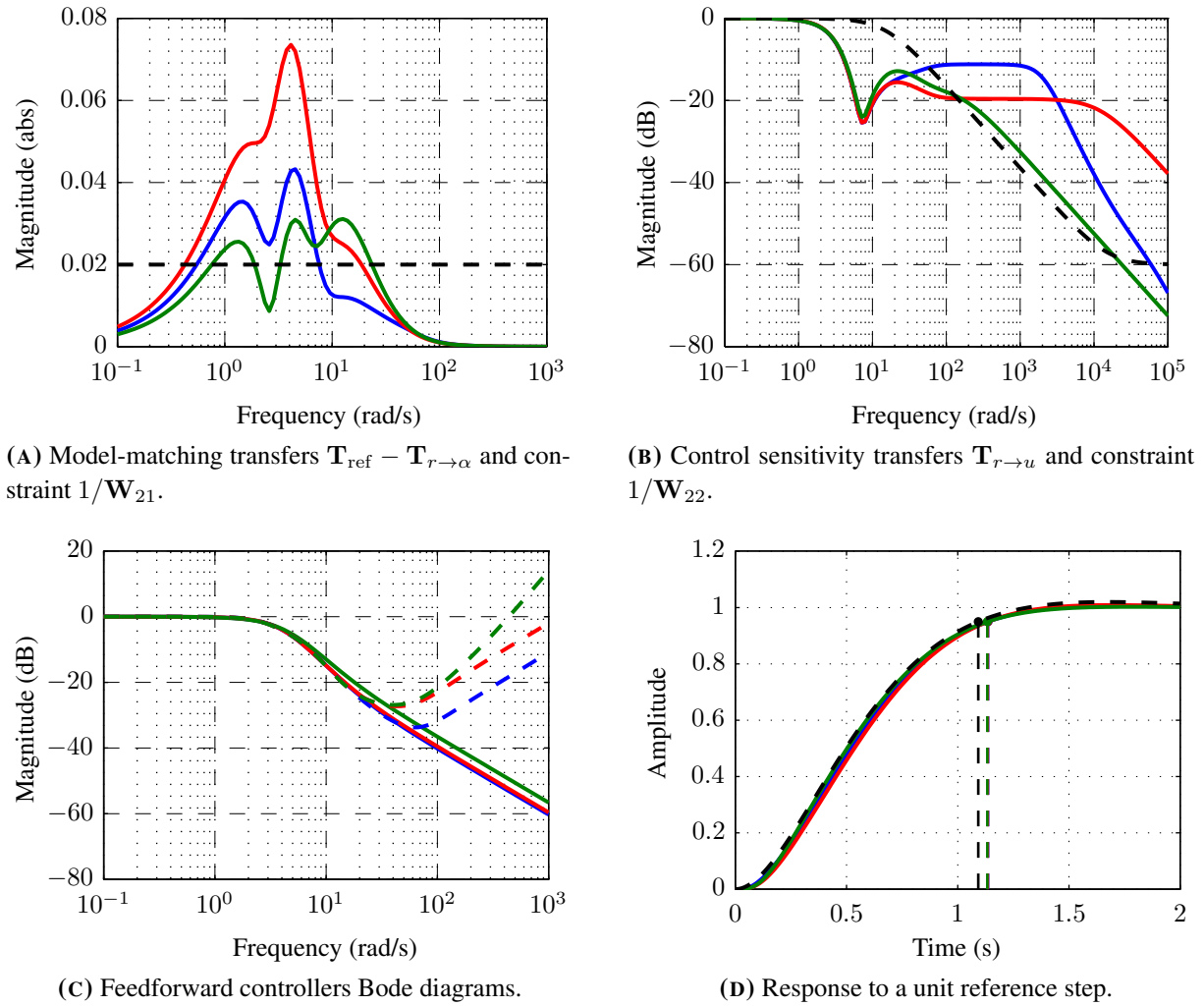


FIGURE B.12: Closed-loop transfer functions and unit step response for the complete pitch axis autopilots (blue: full-order controller $\mathbf{K}_{\text{FB,FO}}$ with inverse-based feedforward $\mathbf{K}_{\text{FF,FO}}$, red: fixed-structure controller $\mathbf{K}_{\text{FB,FS}}$ with inverse-based feedforward $\mathbf{K}_{\text{FF,FS}}$, green: fixed-structure controller $\mathbf{K}_{\text{FB,MO}}$ and equivalent series feedforward controller $\mathbf{K}_{\text{FF,MO}}$ designed using multi-objective synthesis).

similarly with a 5% response time of 1.13 s, with no steady-state error, conforming to the specified reference model.

B.4 Robustness Analysis

One of the prime goals of an autopilot is to ensure the closed-loop system remains stable at all times, even in the presence of uncertainty. A first measure of robustness consists in analyzing the open-loop gain and phase margins of the designed autopilots for the nominal system. The open-loop transfer functions for all three designs are shown in Figure B.13, for a loop opening at the plant input (commanded fin deflection $\delta_{m,c}$). The corresponding gain and phase margins are given in Table B.2.

In all three cases, the target gain and phase requirements ($GM > 6$ dB and $PM > 30^\circ$) are met with a comfortable margin. The third column in the Table B.2 corresponds to the delay margin, which must be

	Gain margin	Phase margin	Delay margin
Full-order K_{FB}	16.53 dB	65.63°	75.8 ms
Fixed-structure K_{FB}	16.26 dB	60.73°	74.9 ms
Multi-objective K_{FB}	15.36 dB	59.46°	74.6 ms

TABLE B.2: Gain, phase and delay margins of the three proposed autopilot designs.

sufficient since these autopilots are to be implemented on a discrete computer. In the ACHILES setup, the control loop runs at a frequency of 100 Hz and the software introduces a fixed 1-period delay to account for variable sensor readout delay. The hold delay and the computation delay can be approximated in first approach with a time delay of 20 ms. These delays may also be taken into account during synthesis. This approach is presented in Section 3.3, which deals with the design of a pitch/yaw skid-to-turn autopilot.

The classical gain and phase margins characterize the stability of the nominal system. However, the actual system deviates from this nominal model due to uncertainty on the estimated parameters, on the operating point and due to unmodeled and/or neglected dynamics. The various sources of uncertainty in the system are cast into a multiplicative input uncertainty representation, consisting in the uncertainty block $\Delta_I(s)$ and the uncertainty weight w_I presented in Section B.1. In order to account for the time delays introduced by the software architecture and the discretization of the controllers, an additional 20 ms time delay is inserted at the actuator input.

The complete autopilot is then cast into an $M - \Delta$ structure as illustrated in Figure B.14. Note that the feedforward controller as well as the disturbance input are not included, as they are not in the

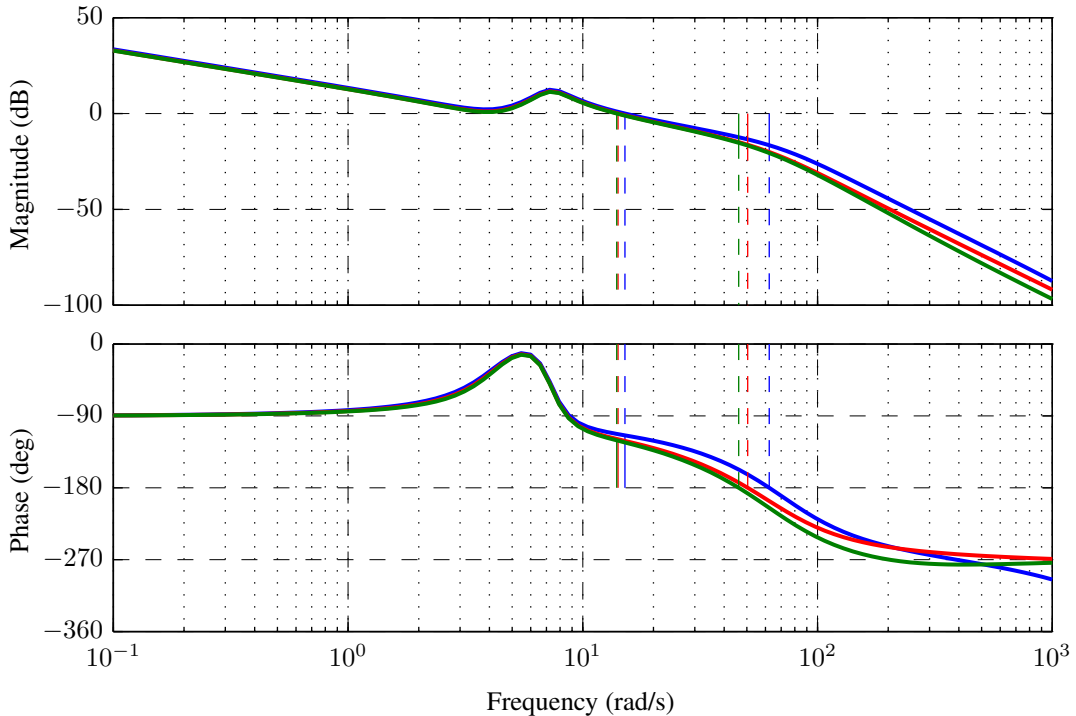


FIGURE B.13: Open-loop gain and phase margins for the disturbance rejection controllers (blue: full-order controller $K_{FB,FO}$, red: fixed-structure controller $K_{FB,FS}$, green: fixed-structure controller $K_{FB,MO}$ in the multi-objective case).

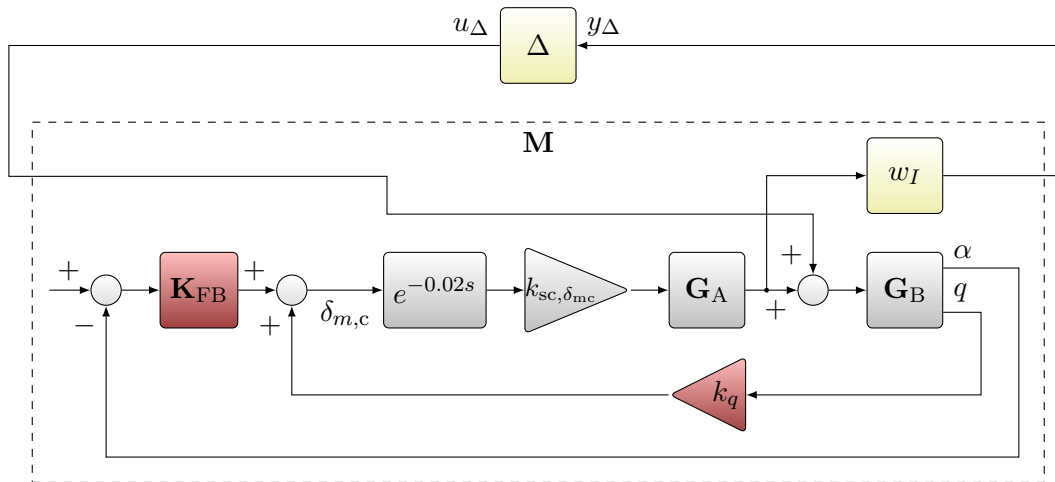


FIGURE B.14: Complete autopilot in $M - \Delta$ form.

path of any feedback loops. The magnitude of the transfer function M for each of the three proposed autopilots is presented in Figure B.15. For all considered designs, the stability condition of Equation (3.30) is respected by a considerable margin. As a result, the developed autopilots are robustly stable with respect to the modeled uncertainty, even though these autopilots were synthesized without taking the actual control delays into account.

B.5 Experimental Validation

The results presented in this section were obtained for a slightly different autopilot design, which was presented in Strub et al. [2015a]. As they remain similar to the simulation results presented in the previous section, they have nonetheless been included for reference.

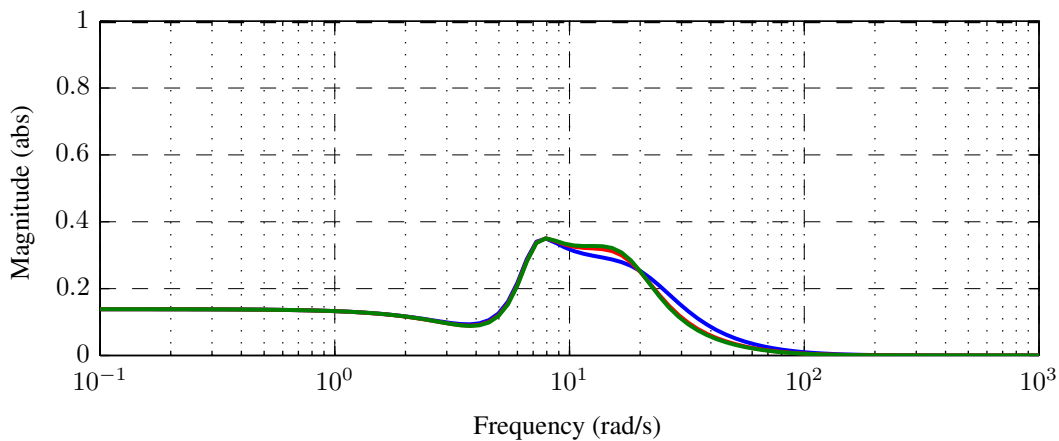


FIGURE B.15: Magnitude of the robust stability-related transfers M . Blue line: full-order controller $K_{FB,FO}$, red line: fixed-structure controller $K_{FB,FS}$, green line: fixed-structure controller $K_{FB,MO}$ designed using multi-objective synthesis. Black dashed line: stability bound on M .

B.5.1 Disturbance Rejection

The normalized response to a disturbance step is illustrated in Figure B.16, for three different values of the reference angle of attack so as to assess the effect of an operating point variation on the disturbance rejection performance. In this figure, the observed responses follow the simulation closely for all three operating points but exhibit visible superimposed oscillations. These oscillations are due to aerodynamic interaction between the support rods and the tail fins, flow unsteadiness around the control and tail surfaces as well as backlash in the actuator linkage. In all three cases, the disturbance is attenuated by more than 50%, rejected in less than 1 s and does not introduce a steady-state error. This is coherent with the results presented in Section B.3, even though the implemented controller is slightly different.

B.5.2 Reference Tracking

The autopilot reference tracking performance is also evaluated in the time domain using a step-wise reference signal. The experimental response is compared with simulated responses of the nominal closed-loop system and the reference model \mathbf{T}_{ref} in Figure B.17A, while the controller output signals are represented in Figure B.17B

The experimental angle of attack response follows the simulated responses closely and presents a response time of 1 s with no steady-state error. The associated control signals show initial peaking on reference step, but with an amplitude not higher than the steady-state value. The small steady-state differences between the simulated and the observed control signals is mainly due to uncertainty on the estimated DC gain, which does not impact performance thanks to the integrator in the feedback controller. Both the experimental output and control signals show oscillations, as in the disturbance rejection case.

The importance of the feedforward filter is highlighted in Figure B.18, in which the feedforward filter is replaced with the unity transfer function. In the present case, this filter is responsible for high-frequency roll-off of the reference signal, as the feedback controller presents relatively high gain at high frequencies due to the zero introduced by the PI controller structure. In its absence, the control signal δ_u exhibits high-amplitude, sharp peaks as illustrated in Figure B.18B. These peaks well exceed the actuator design specifications, resulting in rate saturation in the actuators. As a result, the angle of attack response in Figure B.18A considerably differs from the simulated response (for which the actuator saturation is not modeled) and presents visible overshoot and oscillations. Hence although not necessary for disturbance

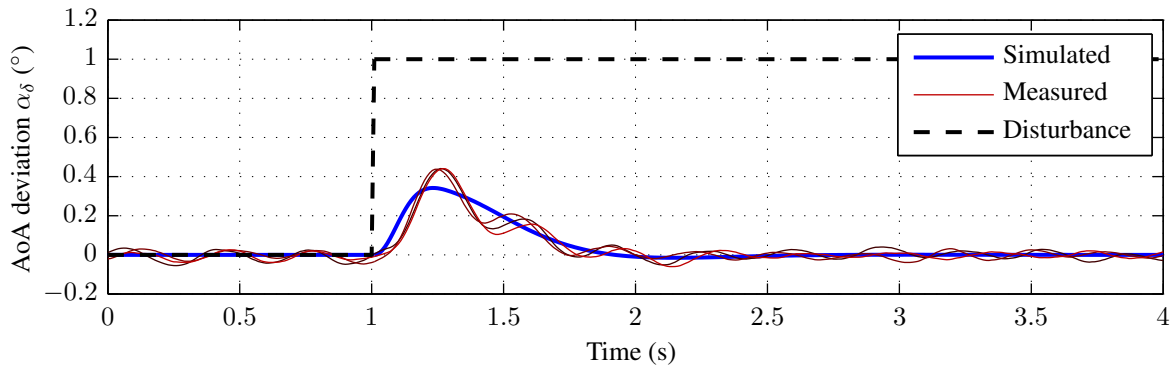
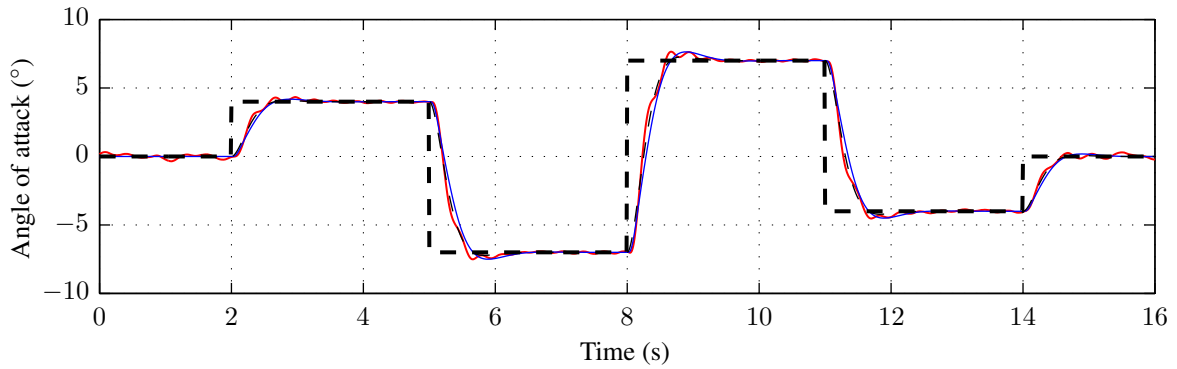
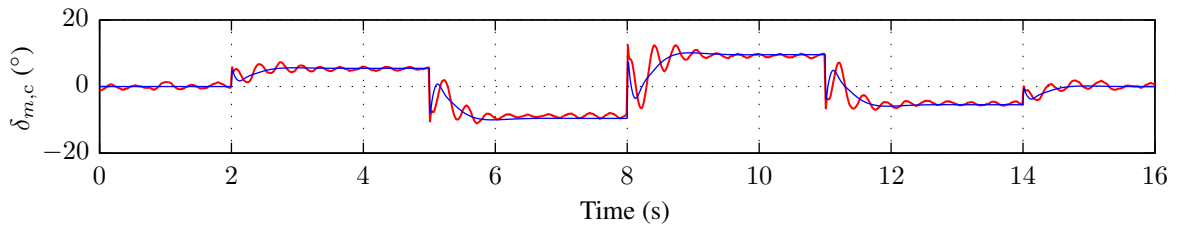


FIGURE B.16: Disturbance rejection performance of the pitch-only autopilot for three different operating points ($\bar{\alpha} = 4^\circ, 7^\circ, -2^\circ$).

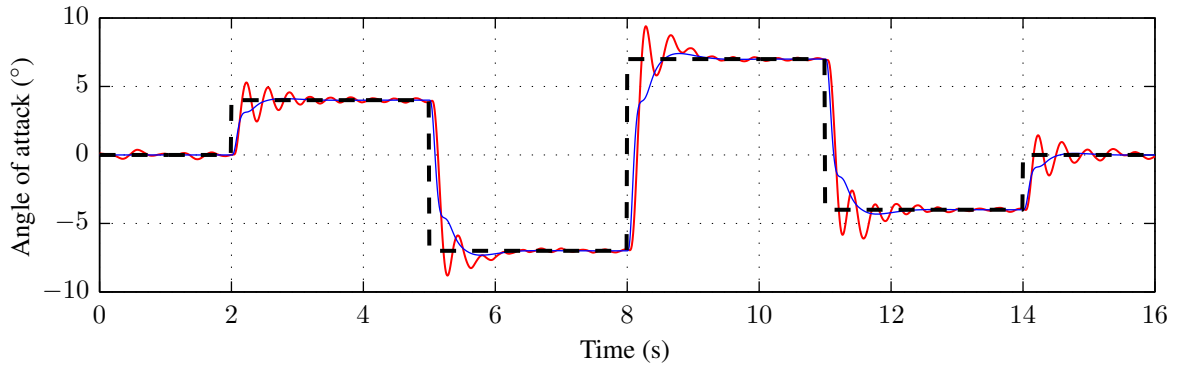


(A) Angle of attack response.

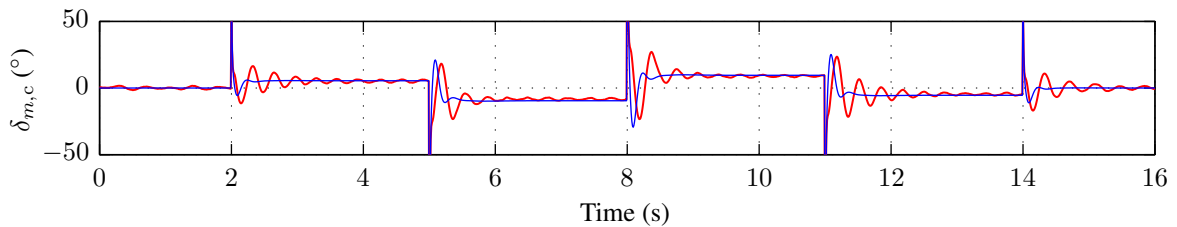


(B) Commanded fin deflections.

FIGURE B.17: Reference tracking performance of the pitch-only autopilot for $V = 25$ m/s.



(A) Angle of attack response.



(B) Commanded fin deflections. Range truncated to $\pm 50^\circ$ for readability.

FIGURE B.18: Reference tracking performance of the pitch-only autopilot without the feedforward filter for $V = 25$ m/s.

rejection, the feedforward controller is mandatory for respecting the reference tracking specifications.

B.6 Concluding Remarks

This first control application consisted in the development of a pitch-axis autopilot for the ACHILES projectile prototype. Using the model parameters estimated in Chapter 2, a nominal plant model was first build using the mean parameter values, and a multiplicative uncertainty representation describes the differences between this nominal model and the estimated models. Based on this knowledge, a two-loop autopilot structure consisting of a rate damping loop, an angle of attack feedback loop and a feedforward filter was devised. Three control design methodologies based on the \mathcal{H}_∞ framework were employed and compared for the synthesis of the controllers, yielding a full-order feedback controller in the first case and PID feedback controllers in the two other cases. For the two first cases, the feedforward controller is an approximation of the ideal feedforward controller obtained via system inversion, while in the third case the feedforward is synthesized simultaneously with the feedback controller, using multi-objective synthesis.

All three methods yield controllers with comparable time-domain performances, gain and phase margins and robustness characteristics. However, in the two first cases, the design requirements for the reference tracking objective are not respected entirely, due to the iterative approach. Indeed, in these cases, the feedback and feedforward controllers are designed in two consecutive steps. The controllers designed using the multi-objective method in the third case show a much better conformance to the design specifications, as both controllers are adjusted to conform to both the disturbance rejection and reference tracking constraints at the same time. As a result, this method is the most flexible of the three presented methods. The multi-objective pitch-only controller has been implemented on the ACHILES setup and compared against numerical simulations of the nominal closed-loop. For both the disturbance rejection and reference tracking objectives, the experimental responses followed the simulation results and conformed to the design specifications. These first application results also allowed the verification of the complete ACHILES setup and its associated development framework.

In this autopilot design, the rate damping gain k_q is selected in a preliminary step, before the synthesis of the feedback and feedforward controllers, converting the single-input, two-outputs underdamped open-loop plant into a damped SISO plant. This separate step is necessary in order to enforce the autopilot structure in the first case, which yields a full-order feedback controller: if the classical \mathcal{H}_∞ synthesis techniques were applied to the original 2×1 plant, the resulting full-order controller would exhibit a structure more complex than a single gain for the pitch rate channel. This separate specification adds unnecessary constraints for the fixed-structure cases but is maintained for comparison's sake. In these cases, specifying k_q as a tunable parameter adds one degree of freedom to the synthesis and may result in a lower final gain γ .

Furthermore, the feedback controllers designed in this section do not roll off at high frequencies due to zeros in the full-order and PID controllers. While this is not an issue for disturbance rejection as the input disturbances are bandwidth-limited by the airframe model, the closed-loop transfer function from the reference to the actuators exhibit a high frequency gain. The feedforward filter is then essential for performance and conformance to the specifications in the reference tracking case. This issue also arises from the selected autopilot structure, which was required for the synthesis of the full-order controller.

Résumé

Cette thèse présente une méthodologie de conception et d'évaluation de lois de commande pour projectiles guidés, au moyen d'un prototype placé dans une soufflerie via un support autorisant plusieurs degrés de liberté en rotation. Ce dispositif procure un environnement permettant à la fois de caractériser expérimentalement le comportement de la munition et d'évaluer les performances des lois de commande dans des conditions réalistes, et est mis en œuvre pour l'étude d'autopilotes de tangage et de lacet, à vitesse fixe et à vitesse variable, pour un prototype de projectile empenné piloté par canards.

La modélisation d'un tel système aboutit à un modèle non-linéaire dépendant de nombreuses conditions de vol telles que la vitesse et des angles d'incidence. Les méthodes de séquençement de gain basées sur des linéarisations d'un modèle non-linéaire sont couramment employées dans l'industrie pour la commande de ce type de systèmes. A cette fin, le système est représenté au moyen d'une famille de modèles linéaires dont les paramètres sont directement estimés à partir de données recueillies sur le dispositif expérimental. L'observation du comportement à différents points de vol permet de considérer la vitesse de l'air comme unique variable de séquençement. La synthèse des différents contrôleurs est réalisée au moyen d'une méthode \mathcal{H}_∞ multi-objectifs à ordre et structure fixes, afin de garantir la stabilité et la robustesse du système vis-à-vis d'incertitudes liées à la variation du point de fonctionnement. Ces lois de commande sont alors validées au moyen d'analyses de robustesse, puis par leur implémentation sur le dispositif expérimental. Les résultats obtenus lors d'essais en soufflerie correspondent aux simulations numériques et sont conformes aux spécifications attendues.

Mots-clés: Commande robuste, μ -analyse, Identifiabilité, Séquençement de gains, Commande \mathcal{H}_∞ multi-objectifs, Mécanique du vol, Systèmes temps-réel.

Abstract

This work presents a novel methodology for flight control law design and evaluation, using a functional prototype installed in a wind tunnel by the means of a support structure allowing multiple rotational degrees of freedom. This setup provides an environment allowing experimental characterization of the munition's behavior, as well as for flight control law evaluation in realistic conditions. The design and validation of pitch and yaw autopilots for a fin-stabilized, canard-guided projectile is investigated, at fixed and variable airspeeds.

Modeling such a system leads to a nonlinear model depending on numerous flight conditions such as the airspeed and incidence angles. Linearization-based gain scheduling techniques are widely employed in the industry for controlling this class of systems. To this end, the system is represented with a family of linear models whose parameters are directly estimated from experimentally collected data. Observation of the projectile's behavior for different operating points indicates the airspeed can be considered as the only scheduling variable. Controller synthesis is performed using a multi-objective, fixed-order fixed-structure method in order to guarantee the stability and robustness of the closed-loop against operating point uncertainty. The obtained control laws are validated with robustness analysis techniques and are then implemented on the experimental setup, where wind-tunnel tests results correlate with numerical simulations and conform to the design specifications.

Keywords: Robust control, μ -analysis, Identifiability, Gain scheduling, Multi-objective \mathcal{H}_∞ control, Flight mechanics, Real-time systems.