PolyPublie
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Commande tolérante aux fautes pour multicoptères
Auteur: Author:	Duc Tien Nguyen
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Nguyen, D. T. (2021). Commande tolérante aux fautes pour multicoptères [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/6573/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/6573/
Directeurs de recherche: Advisors:	David Saussié, & Lahcen Saydy
Programme: Program:	Génie électrique

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Commande tolérante aux fautes pour multicoptères

DUC TIEN NGUYEN
Département de génie électrique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie électrique

Avril 2021

© Duc Tien Nguyen, 2021.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Commande tolérante aux fautes pour multicoptères

présentée par **Duc Tien NGUYEN**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Richard GOURDEAU, président
David SAUSSIÉ, membre et directeur de recherche
Lahcen SAYDY, membre et codirecteur de recherche
Sofiane ACHICHE, membre
André DESBIENS, membre externe

DÉDICACE

À mes parents...

REMERCIEMENTS

Le travail présenté ici a été soutenu financièrement par mes deux directeurs de recherche, Prof. David Saussié et Prof. Lahcen Saydy, ainsi que par le programme de demi-bourse de thèse de Polytechnique Montréal ; je leur suis reconnaissant pour cette aide financière qui m'a permis de poursuivre mes études et de réaliser ce projet de thèse.

Tout d'abord, je tiens à remercier particulièrement mes deux directeurs de recherche. Ils m'ont offert un cadre de travail stimulant et flexible, et ont toujours cru en moi. Je voudrais adresser toute ma reconnaissance à M. David Saussié pour m'avoir encadré, orienté, aidé et conseillé dès mes premiers jours au Canada. Il m'a offert l'opportunité de m'initier au domaine de la recherche et m'a toujours soutenu dans les hauts et les bas de mon parcours scolaire. Sa rigueur, sa passion pour l'enseignement et son attitude scientifique sont des modèles que j'espère pouvoir refléter. J'aimerais également remercier M. Lahcen Saydy qui a accepté d'être mon codirecteur de recherche. Je suis honoré d'être son étudiant et chanceux de travailler sur sa théorie des applications gardiennes. Ses suggestions et ses corrections ont été déterminantes dans la progression de ce projet.

Je remercie ensuite les membres du jury, MM. Richard Gourdeau, Achiche Sofiane et André Desbiens d'avoir accepté de lire ce manuscrit et de participer à la soutenance.

Je remercie également tous mes collègues du laboratoire qui ont su m'encourager, me soutenir et me donner plusieurs conseils précieux : Alexandre Borowczyk, André Phu-Van Nguyen, Catherine Massé, Jérémie Pilon, Justin Cano et Olivier Gougeon. Un grand merci à André qui m'a beaucoup aidé pour réaliser mes essais expérimentaux.

Je voudrais remercier ma chère Mai Nguyễn pour tout ce qu'elle a fait pour moi.

Mes derniers remerciements sont destinés à ma famille : mon père Nguyễn Văn Tình, ma mère Phan Thị Quý, mon frère Nguyễn Đức Trọng et ma petite sœur Nguyễn Đức Quỳnh Như. Sans votre soutien inconditionnel et vos encouragements, je n'aurais pas pu accomplir ce travail. Un grand merci à Trọng qui a pris soin de notre famille ces dernières années. Je n'aurais pas pu me concentrer sur mes études sans ton dévouement et ton sacrifice. Finalement, j'aimerais exprimer ma profonde gratitude envers mes parents à qui je dédie cette thèse. Malgré la pauvreté, vous avez tout fait pour que vos enfants accèdent à l'éducation. Tout ce que j'ai aujourd'hui est le fruit de vos efforts.

RÉSUMÉ

De nos jours, les multicoptères sont largement utilisés dans plusieurs secteurs grâce à leur capacité à décoller et à atterrir verticalement, ainsi qu'à effectuer un vol stationnaire. Cependant, les défaillances de tels systèmes, surtout dans des conditions de navigation défavorables, peuvent avoir des conséquences sérieuses sur leur environnement et les êtres humains. Les multicoptères devraient donc posséder une capacité de tolérance aux fautes, notamment lors de la défaillance des moteurs, afin de garantir leur sûreté de fonctionnement. Cette thèse consiste à développer un système de commande tolérante aux fautes (CTF) permettant à un multicoptère de poursuivre sa mission avec une performance adéquate en dépit des fautes.

Dans la première partie de la thèse, l'approche CTF basée sur la synthèse robuste et l'autoséquencement de gains est proposée. La première solution consiste à paramétrer les gains de commande en fonction de la perte d'efficacité des actionneurs (PEA), fournie par un système de détection et de diagnostic de fautes. La synthèse $\mathcal{H}_\infty$ structurée permet alors d'intégrer les contraintes de performance et de robustesse, déduites du cahier des charges. Afin de faciliter le processus de synthèse, la deuxième solution consiste à paramétrer les gains avec la perte d'efficacité de commande virtuelle (PEV). La structure du contrôleur est alors plus simple et plus générale, de sorte qu'elle peut être mise en œuvre sur différents types de multicoptères. Une combinaison de cette technique avec la ré-allocation de contrôle est également envisagée pour résoudre le même problème. Un tel correcteur est capable de tolérer les fautes sans modifier les lois de commande. Des tests expérimentaux sur un hexacoptère démontrent l'efficacité de ces lois de commande en rejetant de multiples défaillances critiques.

La deuxième partie de cette thèse vise à la conception d'un système CTF générique qui peut être implanté sur la plupart des multicoptères traditionnels. Ce système se divise en deux boucles de commande : un correcteur interne qui génère la commande virtuelle normalisée pour le problème de suivi de trajectoire en présence de fautes et un correcteur adaptatif qui permet de rejeter certaines incertitudes du système et aussi d'adapter les contraintes de conception pour une plateforme spécifique. Afin d'améliorer la robustesse, la loi adaptative est augmentée d'une modification robuste basée sur l'opérateur de projection. Plusieurs simulations et tests expérimentaux ont alors été effectués sur différents multicoptères en utilisant un correcteur unique avec les mêmes paramètres constants.

Dans le but de valider les lois de commande, une analyse de robustesse reposant sur les applications gardiennes est effectuée. Le lien entre cette méthode d'analyse et la valeur singulière structurée (VSS) est en particulier étudié. On propose alors des nouvelles conditions

nécessaires et suffisantes pour la stabilité généralisée robuste sous la forme de la VSS d'une matrice constante. Cette approche est applicable aux systèmes linéaires dépendant de façon affine ou rationnelle de paramètres incertains.

ABSTRACT

The ability of multicopter unmanned aerial vehicles (UAVs) to perform vertical take-off and landing (VTOL) as well as stationary hover flight makes them intensively useful for various commercial and military applications. Due to the increasing requirement for high autonomy and safety, UAVs should possess a fault-tolerant ability to accommodate malfunctions during flight. The objective of this thesis is to design Fault-Tolerant Control (FTC) systems for a multicopter UAV subject to actuator faults and system uncertainties.

In the first part of the thesis, FTC approaches based on gain-scheduling (GS) with structured $\mathcal{H}_\infty$ synthesis are proposed. The first solution is to parameterize the controller gains according to the loss of actuator effectiveness (LAE), given by an appropriate fault detection and diagnosis system. Then, the structured $\mathcal{H}_\infty$ synthesis allows to integrate the performance and robustness constraints, deduced from the specifications. In order to facilitate the tuning process, the second solution is to parameterize the gains with the loss of virtual control effectiveness (LVE). The controller structure is then simpler, and more general, so that it can be implemented on different types of multicopter UAVs. A combination of GS and control allocation (CA) technique is also considered to achieve FTC. Using a CA technique has the major benefit that it can deal directly with actuator faults without modifying the control laws. Experimental results performed on an hexacopter UAV show the effectiveness of these control laws by rejecting multiple critical failures.

The second part of this thesis aims at the design of a generic FTC system that can be implemented on most traditional multicopters. The system consists of a two-loop control structure: an internal corrector that generates normalized virtual control inputs for the trajectory tracking problem in the presence of faults, and an adaptive corrector that allows to reject some uncertainties of the system and also to adapt the design requirements for a specific platform. In order to improve robustness, the adaptive law is augmented by a robust modification based on the projection operator. Several simulation and experimental tests were successfully performed on different multicopter UAVs using a single controller with the same constant parameters.

In order to validate the control laws, a robustness analysis based on guardian maps is performed. The link between this analysis method and the structured singular value (SSV) is particularly studied. Thus, we propose novel necessary and sufficient conditions in terms of the SSV of a single constant matrix for uncertain systems with affine dependent parametric uncertainties as well as rational ones.

TABLE DES MATIÈRES

DÉDICACE iii

REMERCIEMENTS iv

RÉSUMÉ v

ABSTRACT vii

TABLE DES MATIÈRES viii

LISTE DES TABLEAUX xi

LISTE DES FIGURES xii

LISTE DES SIGLES ET ABRÉVIATIONS xv

LISTE DES ANNEXES xvii

CHAPITRE 1 INTRODUCTION 1
1.1 Contexte 1
1.2 Axes de recherche 2
1.2.1 Axe de recherche portant sur la synthèse robuste et autoséquencement de gains 3
1.2.2 Axe de recherche portant sur la commande adaptative et robuste . . 3
1.3 Plan de la thèse 4

CHAPITRE 2 REVUE DE LITTÉRATURE ET OBJECTIFS 5
2.1 Revue de littérature 5
2.1.1 Synthèse de lois de commande 5
2.1.2 Système de contrôle tolérant aux fautes 9
2.1.3 Commande classique et tolérante aux fautes pour multicoptères . . . 18
2.1.4 Stabilité généralisée 23
2.2 Objectifs, méthodologie et contribution 26
2.2.1 Objectif 1 : développer un système FTC actif par des méthodes d'autoséquencement de gains 27

2.2.2 Objectif 2 : Développer un système FTC actif générique par des méthodes adaptatives et robustes . . . 28
2.2.3 Objectif 3 : analyser la robustesse du système FTC . . . 28
2.2.4 Liste de publications . . . 29

CHAPITRE 3 MODÉLISATION ET COMMANDE CLASSIQUE D'UN MULTICOPTÈRE . . . 31
3.1 Modélisation du multicoptère . . . 31
3.1.1 Préliminaires et notations . . . 31
3.1.2 Modèle dynamique d'un multicoptère . . . 35
3.2 Conception et implémentation d'un contrôleur nominal . . . 40
3.2.1 Synthèse du contrôleur nominal par LQR . . . 41
3.2.2 Implémentation du contrôleur nominal . . . 42
3.3 Montage expérimental . . . 44
3.3.1 De la synthèse à la validation . . . 44
3.3.2 AscTec Firefly . . . 44
3.3.3 Autres configurations . . . 46

CHAPITRE 4 COMMANDE TOLÉRANTE AUX FAUTES PAR DES MÉTHODES D'AUTO-SÉQUENCEMENT DE GAINS . . . 48
4.1 Estimation de fautes . . . 48
4.1.1 Représentation de fautes d'actionneurs . . . 49
4.1.2 Filtre de Kalman à deux étages . . . 50
4.1.3 Estimation de fautes basée sur la dynamique de l'hexacoptère . . . 51
4.2 Auto-séquencement de gains par LAE . . . 52
4.2.1 Synthèse multimodèle et auto-séquencement de gains . . . 53
4.2.2 Synthèse $\mathcal{H}_\infty$ structurée . . . 55
4.2.3 Résultats de synthèse . . . 57
4.2.4 Implémentation du contrôleur et technique d'ajustement . . . 60
4.3 Auto-séquencement de gains par LVE . . . 62
4.4 Auto-séquencement de CA . . . 66
4.5 Simulation et tests expérimentaux . . . 70
4.5.1 Amélioration de la stabilité . . . 70
4.5.2 Évaluation des performances . . . 72
4.5.3 Robustesse à l'estimation incorrecte de fautes . . . 74
4.5.4 Panne de rotors . . . 75
4.5.5 Auto-séquencement de CA . . . 76

4.6 Conclusion du chapitre 80

CHAPITRE 5 COMMANDE GÉNÉRIQUE TOLÉRANTE AUX FAUTES PAR DES MÉTHODES ADAPTATIVES ET ROBUSTES 81
5.1 Approche adaptative 81
5.1.1 Tolérance aux fautes générique 82
5.1.2 Augmentation par MRAC 83
5.1.3 Analyse de la stabilité 85
5.2 Approche adaptative et robuste 87
5.2.1 Modification robuste par opérateur de projection 87
5.2.2 Analyse de la stabilité 88
5.3 Simulation et tests expérimentaux 89
5.3.1 Simulation sur un quadricoptère 91
5.3.2 Simulation et tests expérimentaux sur un hexacoptère 99
5.4 Conclusion du chapitre 101

CHAPITRE 6 ANALYSE DE LA STABILITÉ GÉNÉRALISÉE ROBUSTE DU SYSTÈME FTC PAR APPLICATIONS GARDIENNES 103
6.1 Stabilité généralisée robuste du système incertain 104
6.1.1 Préliminaires et notations 104
6.1.2 Structure d'incertitude affine 108
6.1.3 Structure d'incertitude rationnelle 111
6.1.4 Complexité numérique 118
6.1.5 Exemples illustratifs 119
6.2 Validation du système FTC par applications gardiennes 123
6.2.1 Système LTI mono-entrée en présence de LOE 123
6.2.2 Biais de FDD 125
6.2.3 Système LTI multi-entrée en présence de LOE 128
6.3 Conclusion du chapitre 129

CHAPITRE 7 CONCLUSION 130
7.1 Contributions 130
7.2 Perspectives 131

RÉFÉRENCES 133

ANNEXES 152

LISTE DES TABLEAUX

Tableau 2.1 Exemples de FTC pour les multicoptères 22
Tableau 4.1 Cahier des charges 56
Tableau 4.2 Auto-séquencement par LAE - Résultats de la synthèse $\mathcal{H}_\infty$ structurée 58
Tableau 4.3 Auto-séquencement par LVE - Résultats de la synthèse $\mathcal{H}_\infty$ structurée 64
Tableau 4.4 Vérification de la synthèse $\mathcal{H}_\infty$ structurée 64
Tableau 4.5 Scénarios des tests expérimentaux 70
Tableau 6.1 Résultats d'analyse pour le premier exemple de [202] 118
Tableau 6.2 Exemple 1 121
Tableau 6.3 Analyse numérique 128
Tableau A.1 Paramètres du quadricoptère 154
Tableau A.2 Paramètres de l'Asctec Firefly 154

LISTE DES FIGURES

Figure 1.1 Quelques applications de multicoptères 1
Figure 2.1 Synthèse de contrôle avec GS classique 6
Figure 2.2 Architecture de la MRAC 8
Figure 2.3 Classification de systèmes FTC (adapté de [28]) 11
Figure 2.4 Principaux composants d'un système AFTC 15
Figure 2.5 Technique MMST (adapté de [17]) 17
Figure 2.6 Système GNC classique pour multicoptères 19
Figure 2.7 Principe de ré-allocation de contrôle (adapté de [162]) 21
Figure 3.1 Configuration "+" d'un quadricoptère (à gauche) et configuration "X6" d'un hexacoptère (à droite) 32
Figure 3.2 Géométrie d'un multicoptère : (r_i, φ_i) représente la position du i^{e} rotor dans le plan horizontal du repère $\mathcal{F}_b$; les lettres P et N définissent respectivement les couples de réaction positif et négatif autour de l'axe $\mathbf{z}_b$ qui pointe vers le haut. 37
Figure 3.3 Architecture du contrôleur nominal 43
Figure 3.4 Architecture du montage expérimental pour l'AscTec Firefly 45
Figure 3.5 AscTec Firefly (Source : http://www.asctec.de/) 46
Figure 3.6 Système VICON installé dans la volière du MRASL 47
Figure 4.1 Comparaison du filtre de Kalman et de TSKF 50
Figure 4.2 Architecture de FTC basée sur LAE utilisée à des fins de synthèse . . 52
Figure 4.3 Auto-séquencement par LAE - Surface de gains pour le premier rotor $(\gamma_4 = \gamma_6 = 0)$ 58
Figure 4.4 Auto-séquencement par LAE - Réponse du système en boucle fermée 59
Figure 4.5 Auto-séquencement par LAE - Évaluation de contraintes fréquentielles 59
Figure 4.6 Comparaison de la norme $\mathcal{H}_\infty$ $(\gamma_4 = \gamma_6 = 0)$ 60
Figure 4.7 Architecture de FTC basée sur LVE utilisée lors de la synthèse 62
Figure 4.8 Auto-séquencement par LVE - Évolution de gains du contrôleur . . . 65
Figure 4.9 Auto-séquencement par LVE - Réponses du système en boucle fermée 65
Figure 4.10 Architecture de FTC basée sur l'auto-séquencement de CA 66
Figure 4.11 Auto-séquencement de CA - Confinement des pôles en boucle fermée 69
Figure 4.12 Auto-séquencement de CA - Réponses du système en boucle fermée . 69
Figure 4.13 Cas 1 - LAE et LVE 71
Figure 4.14 Cas 1 - Amélioration de la stabilité pour $\gamma_1 = 20\%$ et $\gamma_2 = 90\%$ . . . 71

Figure 4.15 Cas 2 et 3 - Amélioration de performances 72
Figure 4.16 Cas 2 - Attitude du drone 73
Figure 4.17 Cas 2 - Vitesse de rotation des rotors 73
Figure 4.18 Cas 2 - LAE et LVE estimées 74
Figure 4.19 Cas 2 - Robustesse par rapport à l'estimation incorrecte de fautes . . 75
Figure 4.20 Simulation de la panne de rotors sur un hexacoptère NNPPNP 76
Figure 4.21 Simulation du cas 1 - Position du drone 77
Figure 4.22 Simulation du cas 1 - Commandes sur les rotors 77
Figure 4.23 Résultats expérimentaux pour l'auto-séquencement de CA 78
Figure 4.24 Vitesses de rotation des rotors pour un vol stationnaire avec $\gamma_1 = 40\%$ à $t = 12\,\mathrm{s}$ 79
Figure 4.25 LAE estimée pour un vol stationnaire avec $\gamma_1 = 40\%$ à $t = 12\,\mathrm{s}$ 79
Figure 5.1 Architecture du contrôleur FTC générique pour multicoptères 82
Figure 5.2 Loi d'adaptation 85
Figure 5.3 Loi d'adaptation avec modification robuste 88
Figure 5.4 Évolution de gains du contrôleur longitudinal par rapport à LVE . . . 90
Figure 5.5 Réponse du quadricoptère en boucle fermée au point d'opération normale 91
Figure 5.6 Réponse du quadricoptère en boucle fermée en présence de bruits . . 92
Figure 5.7 Phénomène de dérive de gains en présence de bruits 92
Figure 5.8 Réponse du quadricoptère en boucle fermée en présence de paramètres incertains 93
Figure 5.9 Évolution des paramètres adaptatifs pour une série d'échelons de référence 94
Figure 5.10 Oscillations transitoires 94
Figure 5.11 Position du quadricoptère en suivi de trajectoire 95
Figure 5.12 Attitude du quadricoptère en suivi de trajectoire 96
Figure 5.13 Commandes sur les rotors pour un vol en suivi de trajectoire 96
Figure 5.14 Évolution des paramètres adaptatifs pour un vol en suivi de trajectoire 97
Figure 5.15 Évolution des paramètres adaptatifs avec l'opérateur de projection . . 97
Figure 5.16 Performance en position de la MRAC avec l'opérateur de projection . 98
Figure 5.17 Performance en attitude de la MRAC avec l'opérateur de projection . 98
Figure 5.18 Commandes sur les rotors avec l'opérateur de projection 99
Figure 5.19 Simulation de la panne de rotors - Contrôleur générique adaptatif . . 100
Figure 5.20 Simulation de la panne de rotors - Paramètres adaptatifs 100
Figure 5.21 Contrôleur générique adaptatif - Commandes sur les rotors 100
Figure 5.22 Résultats expérimentaux pour le contrôleur générique tolérant aux fautes 102

Figure 5.23 Commandes sur les rotors 102
Figure 6.1 Région de stabilité d'intérêt 117
Figure 6.2 Stabilité d'Hurwitz de $\mathbf{A}(\boldsymbol{\delta})$. 119
Figure 6.3 Stabilité d'Hurwitz 127
Figure 6.4 Stabilité généralisée 127
Figure 6.5 Stabilité d'Hurwitz du système LTI multi-entrée 129
Figure B.1 Système multivariable en boucle fermée 157
Figure B.2 Système augmenté 158
Figure B.3 Forme standard du système bouclé 158
Figure B.4 Problème de sensibilité mixte 159
Figure C.1 Illustration de l'opérateur de projection 168
Figure D.1 Régions de stabilité d'intérêt 174

LISTE DES SIGLES ET ABRÉVIATIONS

Acronymes

AFTC	*Active Fault Tolerant Control*, commande tolérante aux fautes active
CA	*Control Allocation*, allocation de commande
CNS	Condition Nécessaire et Suffisante
DoF	*Degree of Freedom*, degré de liberté
FDD	*Fault Detection and Diagnosis*, détection et diagnostic de fautes
FRD	*Frequency-Response Data*
FTC	*Fault Tolerant Control*, commande tolérante aux fautes
GS	*Gain-Scheduling*, séquencement de gains
IMU	*Inertial Measurement Unit*, unité de mesure inertielle
LAE	*Loss of Actuator Effectiveness*, perte d'efficacité d'actionneurs
LFT	*Linear Fractional Transformation*, transformation linéaire fractionnaire
LMI	*Linear Matrix Inequality*, inégalité matricielle linéaire
LOE	*LOss of control Effectiveness*, perte d'efficacité de contrôle
LPV	*Linear Parameter Varying*, linéaire à temps variant
LQR	*Linear Quadratic Regulator*, commande linéaire quadratique
LTI	*Linear Time-Invariant*, linéaire à temps invariant
LVE	*Loss of Virtual control Effectiveness*, perte d'efficacité de commande virtuelle
MRAC	*Model Reference Adaptive Control*
MRASL	*Mobile Robotics and Autonomous Systems Laboratory*
PFTC	*Passive Fault Tolerant Control*, commande tolérante aux fautes passive
PID	Proportionnel Intégral Dérivé
ROS	*Robot Operating System*
SMC	*Sliding Mode Control*, commande par mode de glissement
TSKF	*Two-Stage Kalman Filter*, filtre de Kalman à deux étages
UAV	*Unmanned Aerial Vehicles*
VSS	Valeur singulière structurée

Notations mathématiques

$\mathbb{R}$	Corps des nombres réels
$\mathbb{C}$	Corps des nombres complexes.
$\mathbb{R}^{n\times m}$	Ensemble des matrices réelles à n lignes et m colonnes

$\mathcal{S}$	Ensemble de stabilité généralisée
$\partial\mathcal{S}$	Frontière de $\mathcal{S}$
$\overline{\mathcal{S}}$	Adhérence de $\mathcal{S}$
$\mathbf{I}_n$	Matrice identité d'ordre n
$\sigma(\mathbf{A})$	Ensemble des valeurs singulières de la matrice $\mathbf{A}$
$\lambda(\mathbf{A})$	Ensemble des valeurs propres de la matrice $\mathbf{A}$
$\otimes$	Produit de Kronecker
$\oplus$	Somme de Kronecker
$\ominus$	Soustraction de Kronecker
$\odot$	Produit bialterné
$\times$	Produit vectoriel

Modélisation de multicoptères

$\mathcal{F}_b$	Repère objet
$\mathcal{F}_i$	Repère inertiel
m	Masse du multicoptère
$\mathbf{I}_b$	Matrice d'inertie du multicoptère
d	Longueur de bras du multicoptère
k_T	Coefficient de poussée des hélices des moteurs
k_D	Coefficient de traînée des hélices des moteurs
ϕ	Angle de roulis
θ	Angle de tangage
ψ	Angle de lacet
p	Vitesse de rotation en roulis
q	Vitesse de rotation en tangage
r	Vitesse de rotation en lacet

LISTE DES ANNEXES

Annexe A	Modélisation et commande de multicoptères	152
Annexe B	Commande tolérante aux fautes	155
Annexe C	Opérateur de projection	167
Annexe D	Stabilité robuste et application gardienne	170

CHAPITRE 1 INTRODUCTION

1.1 Contexte

La dernière décennie a vu la démocratisation de l'utilisation des drones de type multicoptères aussi bien dans le domaine militaire que dans le domaine civil. Les multicoptères ou multirotors sont des aéronefs à voilure tournante de type hélicoptère avec plus de deux rotors, e.g., quadricoptères, hexacoptères ou octocoptères. Ils présentent plusieurs avantages par rapport aux drones à voilure fixe de même taille, comme leur capacité à décoller et à atterrir verticalement, et à effectuer facilement un vol stationnaire. Ces engins volants offrent de nombreuses applications telles que la surveillance de zones agricoles, forestières et maritimes, la recherche de survivants lors d'un désastre, les missions de reconnaissance ou d'agression, le transport (Fig. 1.1), etc.

(a) Livraison par drone
(Source : https://www.amazon.com/)

(b) Taxi volant
(Source : https://www.ehang.com/)

Figure 1.1 Quelques applications de multicoptères

Les défaillances de tels systèmes, surtout dans des conditions de navigation défavorables, peuvent cependant avoir des conséquences sérieuses sur leur environnement et les humains aux alentours. En effet, les drones peuvent être soumis à plusieurs types de fautes matérielles : bris des hélices, pannes des bus de communication, problèmes de batterie et d'électronique de puissance, capteurs défaillants, etc. En particulier, les fautes des actionneurs ont été identifiées comme étant des risques majeurs en raison de leur impact sur la stabilité du système, en plus de leur probabilité d'occurrence élevée. En raison des exigences d'autonomie et de sécurité, les drones devraient donc posséder une capacité de tolérance aux fautes afin de garantir leur sûreté de fonctionnement.

La tolérance aux fautes sur les drones aériens a reçu récemment une attention particulière de

la part des chercheurs. Plusieurs approches ont été développées pour assurer un vol sécuritaire du drone en présence de fautes. En général, la tolérance aux fautes consiste en deux opérations distinctes : la détection et le diagnostic de fautes (*Fault Detection and Diagnosis* - FDD), et le contrôle tolérant aux fautes (*Fault Tolerant Control* - FTC). Dans le cadre de cette thèse, on se concentre en particulier sur la partie FTC. Cependant, l'intégration des deux modules (FDD et FTC) pour réaliser un compromis entre les performances de diagnostic et de commande demeure un défi, surtout au niveau expérimental. Plusieurs travaux supposent la disponibilité d'un module de détection et d'isolation d'erreurs, et donc une estimation parfaite des fautes. Dans ce contexte, l'objectif de cette thèse est de développer un système FTC complet permettant à un multicoptère de poursuivre sa mission avec une performance acceptable en présence de fautes.

Pour l'estimation des fautes, on reprend les algorithmes classiques de la littérature, comme par exemple le filtre de Kalman. Il faut ensuite concevoir un contrôleur tolérant aux fautes qui permet de maintenir la stabilité du système ainsi qu'un bon niveau de performance malgré l'occurrence de fautes. La chaîne de développement est alors la suivante. Les lois de commande sont d'abord conçues sous MATLAB/SIMULINK, puis simulées et testées sous Gazebo, un environnement de simulation robotique *open source* développé sur l'intergiciel ROS (*Robot Operating System*). Elles sont enfin validées expérimentalement sur les multicoptères du Laboratoire de Robotique Mobile et des Systèmes Autonomes (*Mobile Robotics and Autonomous Systems Laboratory* - MRASL [1]).

1.2 Axes de recherche

La plupart des techniques actuelles pour concevoir des systèmes de commandes sont basées sur la connaissance du processus et de son environnement. Néanmoins, ces informations ne sont pas nécessairement disponibles du fait de plusieurs types d'incertitudes : paramètres inconnus ou variants, bruits de mesure, entrées perturbatrices, etc. Ces incertitudes peuvent être modélisées comme étant stochastiques ou déterministes.

En général, trois méthodes de contrôle sont envisageables pour un tel système. Premièrement, la commande stochastique est réservée aux processus dont les modèles de perturbations sont inspirés de la théorie des probabilités et de la théorie statistique. Deuxièmement, la commande robuste est utilisée pour tenir compte des incertitudes paramétriques, variant rapidement dans le temps, bornées et éventuellement dépendantes de l'état du système. Enfin, la commande adaptative est appropriée pour les processus possédant des paramètres constants

1. `http://www.polymtl.ca/robotique-mobile`

ou qui varient lentement, mais totalement inconnus. La défaillance de l'actionneur est un cas particulier d'incertitude déterministe qui peut être abordée par les deux dernières méthodes. Cela nous amène aux deux axes principaux de recherche de cette thèse.

1.2.1 Axe de recherche portant sur la synthèse robuste et autoséquencement de gains

En présence d'une perte d'efficacité de contrôle (*Loss of Control Effectiveness* - LOE), la dynamique d'un multicoptère varie rapidement et significativement. Par conséquent, un seul contrôleur linéaire tel que le contrôleur proportionnel intégral dérivé (PID) ne peut pas garantir la stabilité et les performances désirées. Dans ce cas, une des solutions privilégiées est la synthèse multimodèle qui permet de synthétiser un contrôleur en considérant plusieurs modèles simultanément de telle sorte que chaque modèle correspond à un niveau de LOE différent. Parmi les approches multimodèles, l'auto-séquencement de gains (*gain-scheduling* - GS) effectué dans le cadre de la synthèse $\mathcal{H}_\infty$ structurée présente plusieurs avantages, tels que le choix *a priori* de l'architecture de contrôle, la considération simultanée de plusieurs contraintes de robustesse et de performance, aussi bien dans le domaine temporel que dans le domaine fréquentiel, etc. Dans ce contexte, le premier axe de recherche est de concevoir des contrôleurs auto-séquencés pour le problème FTC. De plus, une analyse de robustesse sera nécessaire pour valider le système FTC. Pour ce faire, on considère en particulier la théorie des applications gardiennes et la μ-analyse.

1.2.2 Axe de recherche portant sur la commande adaptative et robuste

Le deuxième axe de recherche vise à développer un système FTC générique qui peut être mis en œuvre sur la majorité des multicoptères ; les drones partagent en effet les mêmes équations cinématiques et dynamiques. Dès lors, l'utilisation de la commande virtuelle normalisée, combinée avec un mécanisme d'adaptation, devrait permettre de synthétiser un tel système FTC générique. Plus précisément, il s'agit de concevoir dans un premier temps un contrôleur tolérant aux fautes qui génère les entrées virtuelles normalisées, i.e., les forces et les moments. Ensuite, une transformation adaptative est ajoutée à la boucle externe pour prendre en compte la configuration spécifique de chacun des multicoptères, e.g., leur masse et leur inertie. Le contrôleur fixe s'adapte alors à la dynamique réelle de la plateforme grâce à un choix approprié du modèle de référence. Enfin, la robustesse est améliorée par l'utilisation d'un opérateur de projection. Plusieurs simulations et tests expérimentaux seront effectués sur différents multicoptères en utilisant un contrôleur unique.

1.3 Plan de la thèse

À l'exception de l'introduction, cette thèse consiste en six chapitres :

Le chapitre 2 est consacré à la revue de la littérature portant sur les divers sujets abordés dans cette thèse, soit la théorie de la stabilité généralisée et de la commande, les systèmes de contrôle tolérant aux fautes ainsi que les applications pour les drones en particulier. Le chapitre se conclut sur les objectifs, la méthodologie et les contributions de la thèse.

Le chapitre 3 fournit une vue globale du problème de commande des multicoptères, soit de la modélisation complète du système vers l'implantation du contrôleur sur le montage expérimental. Nous synthétisons un correcteur tolérant aux fautes de base qui sera utilisé comme modèle de référence de notre système. Nous terminons par une description détaillée de notre montage expérimental en laboratoire.

La suite de la thèse constitue les travaux principaux portant sur les deux axes de recherche susmentionnés. Le chapitre 4 détaille d'abord la synthèse des trois contrôleurs autoséquencés dans le cadre de FTC pour multicoptères. Cette approche est étendue au chapitre 5 pour concevoir les lois de commande génériques tolérantes aux fautes. Les deux chapitres se concluent sur les résultats de simulations et de tests expérimentaux effectués au laboratoire. Ensuite, nous abordons de façon plus spécifique l'utilisation des applications gardiennes dans le chapitre 6. Cette méthode d'analyse est d'abord appliquée pour valider les designs précédents et fournir de nouvelles conditions nécessaires et suffisantes de stabilité.

Enfin, nous faisons un retour sur les contributions de cette thèse dans le chapitre 7 et nous terminons par les perspectives qui en résultent. Afin de ne pas alourdir le corps principal du manuscrit, les rappels sur les aspects théoriques concernant, entre autres, la synthèse $\mathcal{H}_\infty$ structurée et les application gardiennes, sont reportés dans les annexes.

CHAPITRE 2 REVUE DE LITTÉRATURE ET OBJECTIFS

La présente revue de littérature porte sur les différents thèmes des axes de recherche présentés dans l'introduction.

Nous abordons d'abord plusieurs techniques de synthèses de lois de commande classique et moderne, notamment la synthèse $\mathcal{H}_\infty$, l'auto-séquencement de gains et la commande adaptative. Comme mentionné précédemment, ce sont les méthodes que nous souhaitons utiliser pour la conception de nos contrôleurs. Nous effectuons ensuite un survol des systèmes FTC existants, ainsi que leurs applications aux multicoptères. Nous terminons enfin sur des rappels généraux portant sur la stabilité généralisée des systèmes linéaires stationnaires (*Linear Time-Invariant* - LTI) et les applications gardiennes. En effet, l'objectif final de la conception des lois FTC est de maintenir la stabilité ainsi que la performance du système en présence de fautes. Celles-ci incluent notamment la stabilité généralisée qui peut être garantie par la théorie des applications gardiennes. Dans l'ensemble, cette revue de littérature a pour but d'éclairer le lecteur sur la fonctionnalité d'un système FTC, et surtout de montrer l'originalité de notre travail.

Forts de cette revue de littérature, nous présenterons un tour d'horizon des principaux travaux proposés dans cette thèse, ainsi que les contributions obtenues.

2.1 Revue de littérature

Cette section résume l'état de l'art sur la synthèse de lois de commande tolérante aux fautes dans le cas des multicoptères et l'analyse de leur stabilité.

2.1.1 Synthèse de lois de commande

L'objectif principal de la thèse est de concevoir un système de commande robuste et auto-séquencée par rapport aux fautes d'actionneurs. La synthèse d'un tel contrôleur est au confluent de deux domaines : le problème du séquencement de gains dans le cadre de la commande linéaire à paramètres variants et le problème de synthèse robuste à architecture fixe. Dans la deuxième phase de la thèse, nous porterons aussi notre attention sur la commande adaptative avec modification robuste.

2.1.1.1 Commande linéaire à paramètres variants

Un système linéaire à paramètres variants (*Linear Parameter Varying* - LPV) est un système dont le modèle d'état dépend d'un vecteur de paramètres variant dans le temps $\boldsymbol{\theta}(t) \in \boldsymbol{\Theta}$, $\forall t \geq 0$ où $\boldsymbol{\Theta}$ est le domaine d'opération du système. L'obtention du modèle LPV peut par exemple se baser sur la linéarisation de la dynamique du système non linéaire en chaque point de fonctionnement dans $\boldsymbol{\Theta}$. Un contrôleur LPV doit donc assurer la stabilité et les performances du système en boucle fermée sur l'ensemble des trajectoires de $\boldsymbol{\theta}(t)$ dans $\boldsymbol{\Theta}$. Il existe de nombreuses méthodes de commande LPV dans la littérature qui permettent de synthétiser un tel contrôleur. Parmi les plus utilisées, on retient le correcteur séquencé dont les gains varient de façon continue en fonction de variables de séquencement. Cette technique, plus connue sous le terme anglais de *gain-scheduling* (GS), est largement employée dans l'industrie surtout dans le domaine aéronautique [1].

Approche classique du séquencement de gains L'approche GS classique est basée sur la discrétisation du domaine d'opération $\boldsymbol{\Theta}$ en un nombre fini de points de fonctionnement. Pour chaque modèle linéarisé obtenu, il s'agit de synthétiser un contrôleur LTI à architecture fixe permettant de satisfaire le cahier des charges. On peut donc construire un tableau de gains communément appelé *look-up table* (d'après le terme anglais). Au cours d'une manœuvre, il suffit d'interpoler les gains du correcteur précédemment synthétisés en fonction des variables de séquencement, le plus souvent contenues dans $\boldsymbol{\theta}(t)$ [1]. La figure 2.1 illustre l'architecture générale d'un système de contrôle basé sur l'approche GS classique.

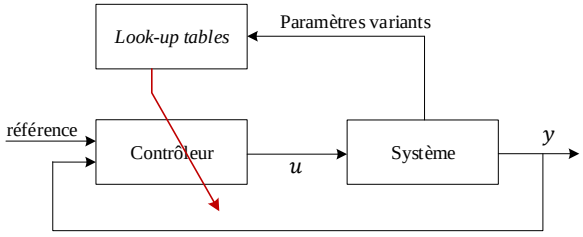


Figure 2.1 Synthèse de contrôle avec GS classique

Bien que très populaire et largement employée dans l'industrie du fait de sa simplicité, l'approche GS classique souffre de certaines limitations. En effet, le nombre de points de fonctionnement étant théoriquement infini, la construction de *look-up table* et l'interpolation des gains sont généralement coûteuses en temps de calcul. Une sélection appropriée d'un nombre limité de points de fonctionnement est donc nécessaire pour couvrir adéquatement le domaine d'opération. De plus, exception faite de la technique présentée dans [2] qui garantit

la stabilité entre deux points de synthèse, les autres approches nécessitent une vérification *a posteriori* [1, 3] pour s'assurer du bon fonctionnement du système. La dernière limitation consiste en l'implémentation elle-même du contrôleur séquencé sur le système non linéaire en présence de termes couplés cachés [1]. Afin d'éviter ces inconvénients, plusieurs techniques d'auto-séquencement de gains ont été développées.

Auto-séquencement de gains Le problème d'auto-séquencement de gains consiste à actualiser automatiquement les gains du contrôleur en fonction des paramètres variant dans le temps. À cette fin, le contrôleur est synthétisé à travers l'enveloppe de fonctionnement $\boldsymbol{\Theta}$ en utilisant une structure fixe de séquencement de gains sous la forme :

$$\mathbf{K}(\boldsymbol{\theta}) = \mathbf{K}_0 + f_1(\boldsymbol{\theta})\mathbf{K}_1 + ... + f_L(\boldsymbol{\theta})\mathbf{K}_L \tag{2.1}$$

où $\mathbf{K}_0, \ldots, \mathbf{K}_L$ sont les gains ajustables et $f_0, \ldots, f_L$ sont les fonctions de séquencement choisies *a priori*. Dans le contexte de FTC, la variable $\boldsymbol{\theta}(t)$ peut être considérée comme les amplitudes des fautes et supposée connue ou estimée en temps réel.

Parmi les méthodes LPV permettant de synthétiser un contrôleur auto-séquencé à architecture fixe, mentionnons les approches basées sur : les fonctions de Lyapunov [4–7], le placement de structures propres [8, 9], le théorème du petit gain [10] ou les applications gardiennes [11]. Dans cette thèse, on s'intéresse en particulier à la synthèse $\mathcal{H}_\infty$ structurée pour résoudre le problème d'auto-séquencement de gains [9, 12–16]. Les fondements théoriques concernant cette synthèse sont fournis en annexe B.2.

2.1.1.2 Commande adaptative

Les méthodes LPV présentées ci-dessus ne sont en général pas applicables quand les paramètres incertains sont totalement inconnus. Dans ce cas, l'une des solutions les plus efficaces est la synthèse de lois adaptatives. L'idée est de modifier en temps réel les gains du contrôleur de manière à ce que le système bouclé s'adapte rapidement à la variation de paramètres inconnus. Il s'agit de choisir une loi d'ajustement de gains, soit *indirecte* (*Self-Tuning Regulator* - STR), soit *direct* (*Model Reference Adaptive Control* - MRAC). La STR se base sur une estimation en temps réel des paramètres inconnus du système. Cette approche est appropriée pour les dynamiques suffisamment lentes. Dans le contexte de la tolérance aux fautes, l'utilisation d'un module FDD et d'un mécanisme de reconfiguration semble similaire à la STR. Ce dernier est plus convenable, car l'évolution temporelle d'une faute est souvent imprévisible et rapide, voire abrupte. On s'intéresse alors plutôt à la MRAC qui permet d'ajuster les paramètres du contrôleur en minimisant la différence entre la sortie du système bouclé et la

sortie du modèle de référence. Le principe de fonctionnement de MRAC est donné à la figure 2.2. Une boucle interne basée sur un retour d'état ou de sortie fournit le signal de commande alors qu'une boucle externe, la loi d'adaptation, ajuste les paramètres de la boucle interne.

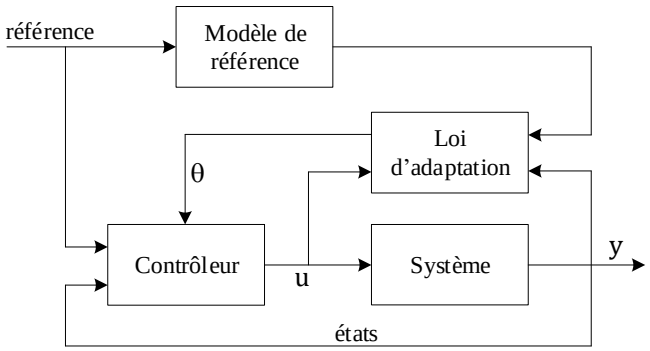


Figure 2.2 Architecture de la MRAC

Considérons le système linéaire :

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} \end{cases} \tag{2.2}$$

où $\mathbf{x} \in \mathbb{R}^n$ est le vecteur d'état, $\mathbf{u} \in \mathbb{R}^m$ le vecteur d'entrée et $\mathbf{y} \in \mathbb{R}^m$ le vecteur de sortie. Le modèle de référence est défini par :

$$\begin{cases} \dot{\mathbf{x}}_{ref} = \mathbf{A}_{ref}\mathbf{x}_{ref} + \mathbf{B}_{ref}\mathbf{r} \\ \mathbf{y}_{ref} = \mathbf{C}\mathbf{x}_{ref} \end{cases} \tag{2.3}$$

où $\mathbf{r} \in \mathbb{R}^m$ représente le vecteur de références à suivre. Les matrices $\mathbf{A}_{ref} \in \mathbb{R}^{n\times n}$ et $\mathbf{B}_{ref} \in \mathbb{R}^{n\times m}$ peuvent être choisies arbitrairement, mais $\mathbf{A}_{ref}$ doit être stable [17].

L'objectif est de forcer le vecteur d'état du système bouclé $\mathbf{x}$ à suivre le vecteur d'état du modèle de référence $\mathbf{x}_{ref}$. Si le gain statique de la matrice de fonctions de transfert $\mathbf{C}(s\mathbf{I} - \mathbf{A}_{ref})^{-1}\mathbf{B}_{ref}$ est l'identité, les sorties $\mathbf{y}_{ref}$ et $\mathbf{y}$ suivront la commande externe constante $\mathbf{r}$.

La première étape consiste à déterminer une structure de contrôle qui permet une correspondance parfaite du modèle (*perfect model matching condition*). En général, on trouve souvent un contrôleur linéaire par retour d'état :

$$\mathbf{u} = \mathbf{L}(\boldsymbol{\theta})\mathbf{x} + \mathbf{M}(\boldsymbol{\theta})\mathbf{r} + \mathbf{v}(\boldsymbol{\theta}) \tag{2.4}$$

où $\mathbf{L} \in \mathbb{R}^{m\times n}$, $\mathbf{M} \in \mathbb{R}^{m\times m}$, $\mathbf{v} \in \mathbb{R}^m$ et $\boldsymbol{\theta}$ le vecteur de paramètres à ajuster.

La deuxième étape consiste à établir l'équation de l'erreur en boucle fermée $\mathbf{e}(\boldsymbol{\theta}) = \mathbf{x} - \mathbf{x}_{ref}$ et sa dérivée $\dot{\mathbf{e}}(\boldsymbol{\theta})$. En se basant sur la dynamique de sensibilité du système $\dot{\mathbf{e}}(\boldsymbol{\theta})$, on cherche enfin une loi d'ajustement des paramètres sous la forme :

$$\dot{\boldsymbol{\theta}}(t) = \mathbf{f}\left(\mathbf{e}(t), \mathbf{x}(t), \mathbf{u}(t)\right) \tag{2.5}$$

pour assurer la convergence du système incertain (2.2) vers le modèle de référence (2.3). Un tel mécanisme d'ajustement des gains peut être obtenu de plusieurs façons, en utilisant par exemple la loi du MIT (*MIT Rule* - méthode du gradient) ou la théorie de Lyapunov (choix d'une fonction candidate). Le lecteur peut se référer aux livres [18, 19] pour une présentation plus complète de la MRAC.

2.1.1.3 Commande adaptative robuste

La robustesse du système bouclé en présence de perturbations non paramétriques mal-assorties (*unmatched disturbances*) demeure un des défis importants de la commande adaptative. En effet, ce type de perturbations cause souvent une divergence des gains adaptatifs. Ce phénomène, appelé *parameter drift* en anglais, peut conduire à des résultats non conformes, voire à l'instabilité du système bouclé. Pour contrer cela, il est nécessaire de procéder à une modification robuste supplémentaire telle que la modification de la zone morte [20], la modification-σ [21], la modification-ϵ [22] ou la modification par l'opérateur de projection [23–25]. Parmi celles-ci, on s'intéresse en particulier à la dernière méthode qui est recommandée par plusieurs auteurs [19, 26].

2.1.2 Système de contrôle tolérant aux fautes

Cette sous-section vise à résumer l'état de l'art sur la commande tolérante aux fautes.

2.1.2.1 Faute et défaillance

Avant de rentrer dans le vif du sujet, il est tout d'abord nécessaire de distinguer la faute (*fault*) de la défaillance (*failure*) [27]. Une faute est le changement inattendu d'une propriété ou d'un paramètre caractéristique du système surveillé par rapport au fonctionnement acceptable ou habituel. À l'opposé, une défaillance est une interruption permanente du service rendu par le système dans des conditions d'exploitation spécifiées. Ainsi, une faute correspond à un comportement anormal du système, pouvant éventuellement conduire à une défaillance. Dans le cas des multicoptères, la dégradation d'un moteur est donc une faute tandis que la

panne totale d'un moteur est une défaillance. Des hélices brisées ou la rupture de liaisons mécaniques sont d'autres exemples de défaillances.

2.1.2.2 Commande tolérante aux fautes

La commande tolérante aux fautes (*Fault Tolerant Control* - FTC) a pour but de maintenir la stabilité du système de commande en présence de fautes et de préserver un niveau de performance comparable à la performance nominale. Dans la littérature portant sur la FTC, on mentionne d'abord les techniques dites *passives* qui font partie de la commande robuste. Un seul contrôleur est conçu pour être robuste face aux fautes d'actionneurs ou de capteurs. L'utilisation d'une action intégrale est ainsi un moyen typique d'obtenir dans une certaine mesure une commande tolérante aux fautes passive.

Par ailleurs, on trouve également des techniques de tolérance aux fautes dites *actives* où le contrôleur est reconfiguré dynamiquement pour faire face à l'occurrence de fautes. Un contrôleur adaptatif ou un régulateur LPV sont des exemples d'un correcteur FTC actif. En général, la commande tolérante aux fautes active se décline en deux phases [28] :

— la détection et le diagnostic de fautes (*Fault Detection and Diagnosis* - FDD) qui permet de détecter, d'isoler et de déterminer la sévérité des fautes, si elles se produisent dans un système surveillé ;

— le recouvrement qui permet au système bouclé de continuer à fonctionner raisonnablement malgré l'apparition de fautes.

Au cours des dernières décennies, la conception des systèmes FTC est devenu un champ d'investigation scientifique particulièrement dynamique. Dans ce qui suit, on présente un tour d'horizon de principaux travaux reportés dans la littérature portant sur la FTC. Une vue globale des méthodes existantes est donnée à la figure 2.3.

2.1.2.3 Principes et techniques de FDD

En général, la détection et l'estimation de la sévérité de fautes ne sont pas faciles. En effet, l'évolution temporelle de fautes est normalement imprévisible, et peut donc être abrupte, progressive ou encore intermittente. De plus, une faute peut posséder une valeur indécelable ou masquée. Dès les années 1970, le problème de détection et d'isolation de fautes (*Fault detection and isolation* - FDI) a reçu une attention importante de la part des chercheurs. Plusieurs méthodes FDI ont été développées et appliquées dans le but de mettre en évidence la présence et la localisation de fautes. Une fois la faute détectée, il s'agit de chercher à la compenser par une modification adéquate de la loi de commande. En vue de la conception

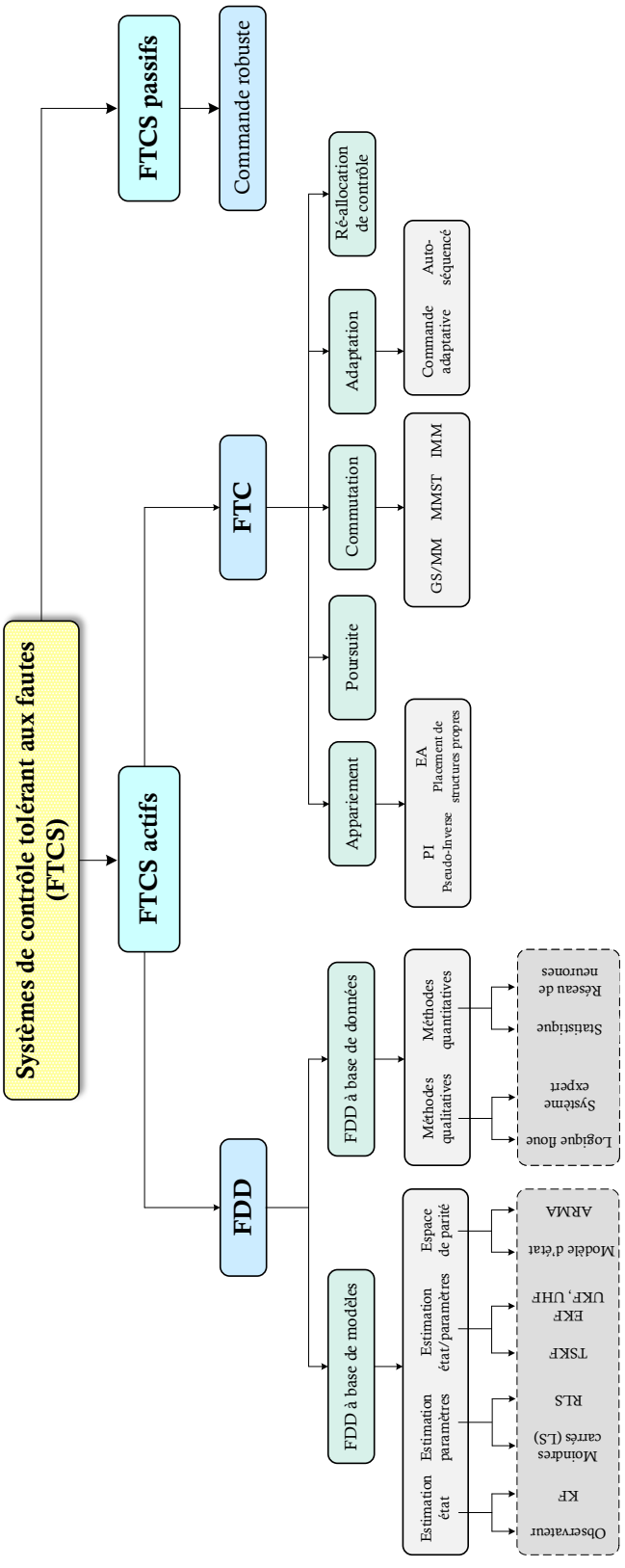


Figure 2.3 Classification de systèmes FTC (adapté de [28])

d'un tel système, le module FDI est souvent élargi en ajoutant l'identification, ce qui aboutit au FDD. Les fonctions d'un système FDD sont normalement divisées en trois tâches :

— la détection de fautes, qui permet de savoir que le système surveillé est confronté à un dysfonctionnement à un moment déterminé ;

— l'isolation de fautes, qui a pour but de déterminer la localisation des fautes ;

— l'identification de fautes, qui estime la magnitude des fautes.

Dans la littérature, le diagnostic correspond aux tâches d'isolation et d'identification.

Principe de redondance Le principe de redondance a joué un rôle important dans la conception de systèmes tolérants aux fautes. Historiquement, les premières méthodes FDD utilisaient la **redondance matérielle**, signifiant ainsi que plusieurs sous-systèmes (capteurs, actionneurs, etc.) réalisent la même fonction et peuvent être interchangés en cas de défaillance de l'un d'entre eux. Cette approche permet une détection rapide et une localisation précise de fautes. Par contre, elle engendre une augmentation du coût, du volume et de la masse ainsi que la difficulté de maintenance et la probabilité de panne. Cette méthode est donc utilisée dans les applications critiques (nucléaires, commandes de vol électriques - *fly-by-wire*, etc.). Notons que la redondance de capteurs, qui est généralement plus simple et moins coûteuse que la redondance d'actionneurs, est courante dans les systèmes modernes.

En pratique, la **redondance analytique/logicielle** est plus souvent utilisée puisqu'elle n'exige aucun matériel supplémentaire. Cette approche exploite le modèle mathématique du système ou les relations analytiques entre les grandeurs mesurées/estimées.

FDD par redondance analytique La FDD basée sur la redondance analytique a fait l'objet de nombreuses travaux dans les années récentes. Généralement, les méthodes existantes peuvent être classées en deux catégories : celles à base de modèles ou celles à base de données.

Quand aucun modèle dynamique n'est exploitable, les **méthodes à base de données** permettent de détecter et de localiser les fautes en analysant les mesures disponibles. À l'aide d'une classification des observations passées dans une base de données, le diagnostic est réalisé par reconnaissance de formes, par exemple un réseau de neurones. En l'absence même d'historique de fonctionnement, il existe des méthodes qualitatives. Dans ce cas, les seules informations disponibles sur le système surveillé sont les connaissances empiriques des experts, représentées sous forme des règles : SI <*condition*> ALORS <*conclusion*>. Un exemple typique de cette approche est la logique floue (*Fuzzy logic*). En général, les méthodes à base de

données ont des capacités d'identification assez restreintes. De plus, elles nécessitent parfois des capteurs supplémentaires. On renvoie le lecteur à [29, 30] pour plus de détails.

Contrairement aux méthodes à base de données, les **méthodes à base de modèles** reposent sur une connaissance de la dynamique du système. Le principe de fonctionnement est de comparer les mesures relevées sur le système physique et le comportement prédit par son modèle mathématique. Ces méthodes n'exigent pas en général de capteurs supplémentaires. En fait, les mesures nécessaires pour la partie commande sont souvent suffisantes pour l'algorithme FDD. Le lecteur est invité à se référer aux livres [31, 32] et aux articles de synthèse [28, 33] pour une revue détaillée de celles-ci.

FDD à base de modèles Le diagnostic à base de modèles s'effectue habituellement en combinant un générateur de résidus avec une stratégie d'analyse. La première étape consiste à calculer les résidus à partir de l'écart entre les mesures relevées sur le système et le comportement prédit par un modèle [31, 34]. Une fois les résidus déterminés, ils sont examinés en utilisant quelques seuils fixes pour détecter les fautes. La dernière étape comprend les décisions booléennes sur la présence de défauts.

Selon la technique de génération et d'évaluation de résidus, plusieurs méthodes FDD à base de modèles ont été développées. En général, ces méthodes sont classées en trois groupes :

— Estimation d'état : l'idée fondamentale est basée sur la théorie d'estimation, soit par un observateur de Luenberger dans le contexte déterministe, soit par un filtre de Kalman dans le cadre stochastique.

— Espace de parité : les équations du modèle sont projetées dans un espace particulier appelé *espace de parité* où les redondances permettent d'éliminer les inconnues.

— Estimation de paramètres : le principe est d'analyser l'impact des fautes qui se reflète sur les paramètres du système, et non pas ses variables d'état.

Les méthodes d'estimation de paramètres visent à détecter les changements structurels du système, tandis que les autres méthodes sont plus convenables quant aux fautes d'actionneurs ou de capteurs. Dans le cadre de cette thèse, on s'intéresse en particulier au filtre de Kalman à deux étages (*Two-Stage Kalman filter* - TSKF) qui permet d'estimer à la fois les états et les paramètres du système. Cette approche a été proposée par [35] et appliquée pour identifier les fautes d'actionneurs d'un avion [36, 37] ou d'un quadricoptère [38–41]. À travers [42, 43], les fautes de capteurs d'un drone ont aussi été estimées par ce filtre. En comparaison avec les autres approches telles que les filtres de Kalman classique ou étendu, le TSKF semble relativement performant [28].

2.1.2.4 Principes et techniques de FTC

Une fois les fautes détectées, il s'agit de chercher à les compenser, ou au moins à limiter leurs effets par une modification adéquate de la loi de commande. Comme mentionné précédemment, on distingue deux approches de commande tolérante aux fautes : une approche passive (PFTC) et une approche active (AFTC). Dans la première approche, le contrôleur est conçu pour être robuste vis-à-vis d'un ensemble de fautes prédéfinies. Cette approche ne nécessite aucun module de détection de défauts, ni aucune reconfiguration de la loi de commande. Cependant, sa capacité de tolérance est limitée à un nombre restreint de fautes. Dans la deuxième approche, le contrôleur est reconfiguré dynamiquement pour faire face à l'occurrence de fautes imprévues. Cette reconfiguration consiste à ajuster les gains du contrôleur ou à changer la structure du contrôleur en temps réel. Généralement, l'approche active est plus efficace que l'approche passive, mais elle nécessite un module FDD. Une comparaison détaillée des deux approches est donnée dans [44].

Commande tolérante aux fautes passive Malgré la capacité restrictive de la FTC passive, plusieurs travaux portant sur cette méthode ont été reportés dans la littérature. En pratique, un système PFTC avec un seul contrôleur fixe n'exige pas de matériel/logiciel complexe. De plus, cette approche, encore appelée *reliable control* en anglais, est plus fiable que l'approche active grâce à sa simplicité de calcul [45]. Une vue globale des méthodes PFTC peut être trouvée dans les articles [46, 47].

Parmi les FTC passives, la commande par mode de glissement (*Sliding Mode Control* - SMC) a fait l'objet de plusieurs travaux dans les années récentes. L'idée générale est de trouver une commande à structure variable forçant toute trajectoire du système à converger vers une surface de glissement en un temps fini, puis de concevoir une commande équivalente qui maintiendra toujours le système dans cette région. Le principal avantage de la SMC est la robustesse à certains types de perturbations et d'incertitudes. En effet, comme la commande est discontinue et souvent de la forme tout-ou-rien (*bang–bang control* ou *on–off control*), elle est insensible aux variations de paramètres et donc un excellent candidat pour la conception des systèmes PFTC. Le développement théorique et la mise en œuvre de la SMC dans le cadre de PFTC sont présentés dans le livre [48].

Commande tolérante aux fautes active La structure d'un système FTC actif est présentée à la figure 2.4. Typiquement, un tel système est divisé en trois sous-systèmes :

— un module FDD qui est capable de fournir aussi précisément que possible les informations concernant les fautes (temps, type et grandeur) ;

— un contrôleur qui est facilement reconfigurable/restructurable ;

— un mécanisme de reconfiguration performant qui peut rapidement réagir aux fautes en utilisant des informations issues en ligne du sous-système de diagnostic.

Différents systèmes AFTC rencontrés dans la littérature peuvent être classifiés selon [28] :

— la synthèse de la loi de commande : placement de pôles, commande adaptative, méthode de pseudo-inverse, etc.

— l'approche de conception : on peut soit sélectionner des lois de commande prédéterminées pour compenser l'impact de fautes connues (*pre-computed control laws*), soit synthétiser en ligne un nouveau contrôleur pour répondre à l'apparition de fautes (*on-line automatic redesign*).

— le mécanisme de reconfiguration : il est possible d'effectuer la reconfiguration du contrôleur par un mécanisme d'appariement (*matching*), de poursuite (*following*), de commutation (*switching*), ou d'adaptation.

— le type du système (linéaire ou non linéaire).

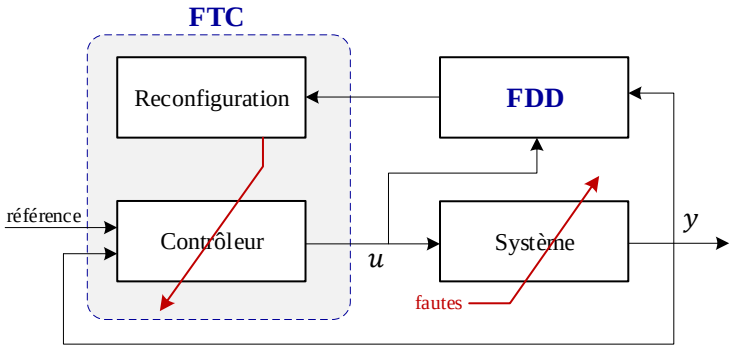


Figure 2.4 Principaux composants d'un système AFTC

Vu que l'un des problèmes critiques des systèmes AFTC est la limite de temps pour la reconfiguration du contrôleur, on mentionne en particulier les quatre mécanismes principaux de reconfiguration reportés dans la littérature.

1. Mécanisme d'appariement La reconfiguration du contrôleur par un mécanisme d'appariement est une des premières méthodes utilisées pour la tolérance aux fautes active [34,49]. Considérons le système linéaire nominal :

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \tag{2.6}$$

et la loi de commande $\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$. Le système en boucle fermée est alors :

$$\begin{cases} \dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \tag{2.7}$$

En présence de fautes, le système en boucle fermée post-défaut avec une nouvelle loi de commande par retour d'état $\mathbf{u}_f(t) = -\mathbf{K}_f\mathbf{x}_f(t)$ devient :

$$\begin{cases} \dot{\mathbf{x}}_f(t) = (\mathbf{A}_f - \mathbf{B}_f\mathbf{K}_f)\mathbf{x}_f(t) \\ \mathbf{y}_f(t) = \mathbf{C}_f\mathbf{x}_f(t) \end{cases} \tag{2.8}$$

L'idée générale est de chercher $\mathbf{K}_f$ pour minimiser $\|(\mathbf{A} - \mathbf{B}\mathbf{K}) - (\mathbf{A}_f - \mathbf{B}_f\mathbf{K}_f)\|_F$. Une des solutions de ce problème est donnée par la méthode de ***pseudo-inverse*** :

$$\mathbf{K}_f^* = \arg\min_{\mathbf{K}_f} \|(\mathbf{A} - \mathbf{B}\mathbf{K}) - (\mathbf{A}_f - \mathbf{B}_f\mathbf{K}_f)\|_F = \mathbf{B}_f^\dagger(\mathbf{A} - \mathbf{B}\mathbf{K} - \mathbf{A}_f) \tag{2.9}$$

où $\mathbf{B}_f^\dagger$ est la pseudo-inverse de la matrice $\mathbf{B}_f$. Cette méthode est convenable pour la FTC active grâce à sa simplicité et à sa possibilité de réagir aux fautes affectant toutes les matrices du modèle d'état. Par contre, la solution optimale $\mathbf{K}_f^*$ n'assure pas la stabilité du système bouclé dans certains cas. Une autre solution possible est la commande par placement de structures propres. Le principe est de placer les valeurs propres et les vecteurs propres de la matrice $\mathbf{A}_f - \mathbf{B}_f\mathbf{K}_f$ le plus proche possible de celles de la matrice nominale $\mathbf{A} - \mathbf{B}\mathbf{K}$.

2. Mécanisme de poursuite L'idée de base est de chercher une commande par retour d'état ou de sortie et une précommande pour forcer les états du système à poursuivre les états d'un modèle de référence. Cette approche a été appliquée pour la tolérance aux fautes active comme présentée dans [37, 50]. Elle est souvent utilisée en combinaison avec une autre stratégie de commande pour augmenter la capacité de tolérance [28]. Par ailleurs, elle peut fonctionner sans module FDD dans un système de contrôle tolérant aux fautes passif [49].

3. Mécanisme de commutation La reconfiguration du contrôleur par commutation est largement utilisée dans les sytèmes AFTC. La conception est basée sur un ensemble de modèles $M_i,\ i = 1, \ldots, N$ qui décrivent le système dans diverses situations de fautes en supposant que l'on puisse toutes les connaître. Une loi de commande sera synthétisée pour chaque mode de fonctionnement. Dans une telle approche, mentionnons les techniques suivantes :

— **Multimodèle** : Pour chaque modèle local M_i, un contrôleur K_i est pré-calculé (hors ligne). Quand une faute est identifiée par le module FDD, le mécanisme de commutation choisit le contrôleur du modèle correspondant le plus à la situation actuelle de fautes. L'approche classique du séquencement de gains (*gain scheduling*) est un exemple typique de cette technique de reconfiguration.

— **Multimodèle et réglage** : en anglais *multiple model switching and tuning* (MMST). L'architecture de cette technique est illustrée à la figure 2.5. La logique de commutation est effectuée via la minimisation d'un indice de performance J_i, défini pour chaque modèle M_i par [17] :

$$J_i(t) = \alpha e_i^2(t) + \beta \int_0^t \exp\left[-\lambda(t-\tau)\right] e_i^2(\tau)\mathrm{d}\tau, \quad \alpha \geq 0, \beta > 0, \lambda > 0 \tag{2.10}$$

où e_i est l'erreur entre la sortie du modèle M_i et la sortie mesurée du système, et α, β, λ sont les coefficients à choisir. Par rapport à la commutation par multimodèle, la reconfiguration par MMST est plus dynamique en utilisant l'indice de performance J_i et un banc de modèles adaptatifs.

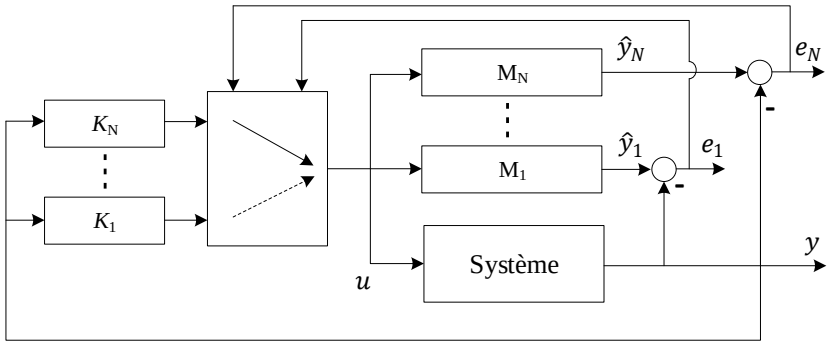


Figure 2.5 Technique MMST (adapté de [17])

— **Multimodèle interactif** : Afin d'améliorer la tolérance aux fautes imprévues/non prédéfinies, la technique multimodèle interactif (*Interacting Multiple Models* - IMM) a été développée. Dans ce cas, la commande est une combinaison convexe [49] :

$$\mathbf{u}(t) = \sum_{i=1}^{N} \mu_i \mathbf{u}_i \ , \ \sum_{i=1}^{N} \mu_i = 1 \tag{2.11}$$

où $\mathbf{u}_i$ est le signal de commande du contrôleur K_i et $0 \leq \mu_i \leq 1$ est la possibilité du modèle M_i issue du module de diagnostic.

4. Mécanisme adaptatif La reconfiguration adaptative est une des méthodes les plus efficaces pour s'accommoder des fautes dès leur apparition. Le principe est de re-synthétiser en ligne le contrôleur en utilisant les informations de fautes. Parmi les techniques de contrôle les plus utilisées, on peut trouver :

— La commande linéaire à paramètres variants où, du point de vue de la tolérance aux fautes, le vecteur de paramètres variant dans le temps $\boldsymbol{\theta}(t)$ correspond à l'amplitude des fautes estimée par le module FDD. Un contrôleur LPV lui-même fonction de $\boldsymbol{\theta}(t)$ peut donc assurer la stabilité et les performances du système bouclé en présence de fautes. Parmi les méthodes LPV les plus utilisées, mentionnons l'auto-séquencement de gains via la synthèse $\mathcal{H}_\infty$ structurée qui permet d'effectuer la synthèse de contrôleurs robustes et autoséquencés à architecture fixe (cf. Section 2.1.1).

— La commande adaptative ayant la capacité de s'adapter automatiquement au changement des paramètres du système (cf. Section 2.1.1). Si le changement provient de fautes, elle peut donc jouer le rôle d'une AFTC. Cette approche n'exige aucun module FDD, ni aucune reconfiguration du contrôleur. Néanmoins, elle présente quelques restrictions quant à la tolérance aux fautes. D'une part, elle ne permet pas de rejeter les fautes de capteurs. Dans ce cas, un contrôleur adaptatif va essayer d'augmenter la commande de manière à rendre le signal de mesure défectueux égal à la valeur souhaitée. Cela peut amener à l'instabilité du système. D'autre part, la commande adaptative est développée pour faire face aux changements graduels de paramètres du processus. Elle est particulièrement efficace pour les systèmes qui sont représentés par les modèles linéaires possédant des paramètres variant lentement. En pratique, les fautes ou les défaillances peuvent provoquer des changements abrupts que l'algorithme adaptatif ne peut pas suivre tout de suite. Cette méthode n'est ainsi appropriée qu'à la tolérance aux fautes progressives.

2.1.3 Commande classique et tolérante aux fautes pour multicoptères

Généralement, la partie mécanique d'un multicoptère est assez simple, car il repose sur la variation de la vitesse des rotors à pas fixe pour se mettre en mouvement. Cependant, la commande de cet engin volant présente une difficulté supplémentaire pour les configurations sous-actionnées. De plus, les multicoptères sont en général des systèmes fortement non linéaires, instables et qui peuvent être soumis à des perturbations importantes (e.g., le vent). De ce fait, les synthèses de commande classiques et modernes pour multicoptères ont fait l'objet de nombreuses investigations. Une revue détaillée est disponible à travers [51–53].

2.1.3.1 Système de commande de vol

La figure 2.6 illustre le système classique de guidage, de navigation et de contrôle (*Guidance, Navigation, and Control* - GNC) d'un multicoptère. L'objectif du guidage est de générer la trajectoire (*path planning*) et de déterminer la vitesse et l'accélération admissibles pour suivre cette trajectoire. Le sous-système de navigation est une combinaison de capteurs avancés (IMU, GPS, caméra, etc.) et de microcontrôleurs, destiné à mesurer et/ou à estimer les états de l'engin volant (position, vitesse, attitude, etc.). Finalement, la partie de contrôle (pilotage) agit sur les actionneurs pour suivre les références générées par le guidage tout en stabilisant le système. Comme mentionné précédemment, un multicoptère est un système sous-actionné dont les six degrés de liberté sont contrôlés par les quatre entrées virtuelles (force et moments). Les mouvements dans le plan horizontal sont contrôlés par les deux couples de roulis et de tangage. À l'heure actuelle, la plupart des multicoptères commerciaux possèdent une telle structure de contrôle hiérarchique.

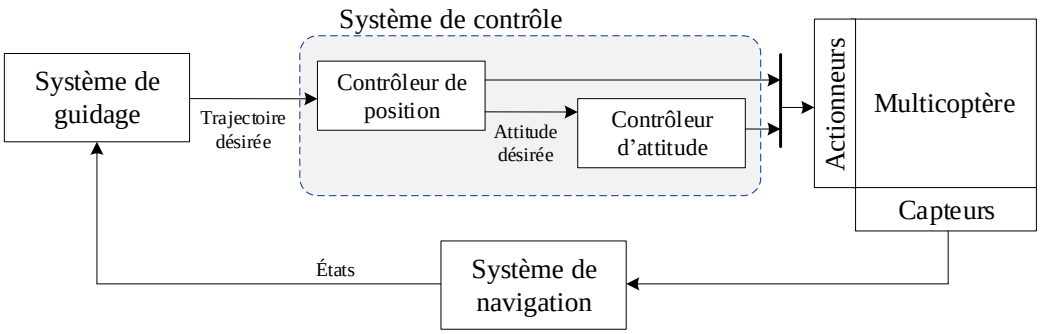


Figure 2.6 Système GNC classique pour multicoptères

Dans le cadre de ce projet, on s'intéressera plus particulièrement à la réalisation des fonctions de guidage et de contrôle. On suppose que les états du système fournis par le système de navigation sont complets, précis et rapides pour appliquer les techniques de contrôle.

2.1.3.2 Principe de la commande de multicoptères

Plusieurs techniques de contrôle classique et moderne ont été appliquées pour améliorer la stabilité et les performances en vol de différents types de multicoptères. Premièrement, les régulateurs d'attitude de type PID (Proportionnel Intégral Dérivé) ont été testés expérimentalement avec succès pour les quadricoptères [54, 55]. Dans les travaux de [56], ce type de contrôleur a été utilisé pour les deux sous-systèmes de commande du drone (position et attitude). Les résultats de simulation et expérimentaux ont démontré une bonne performance

du contrôleur PID (erreur en régime permanent faible, dépassement acceptable et temps de réponse rapide). Cependant, cette technique de contrôle basée sur la linéarisation n'assure que la stabilité du système autour du point d'équilibre. De plus, les gains du contrôleur sont choisis de façon ad-hoc dans ces travaux. Le réglage des gains demeure donc un défi.

D'autres techniques linéaires plus efficaces que le PID telles que la commande linéaire quadratique (LQR) ont été appliquées pour les multicoptères, tant en simulation que lors de vols expérimentaux. Originellement, un contrôleur d'attitude de type LQR a été testé avec succès dans [57]. En simulation, ce type de contrôleur a permis de résoudre le problème de suivi de trajectoire [58]. La commande linéaire quadratique gaussienne (LQG) a aussi été utilisée pour stabiliser un quadricoptère [59] ; cette technique de contrôle qui réunit un contrôleur LQR et un estimateur de Kalman n'exige pas une mesure totale des états du système.

Plusieurs travaux sur la commande non linéaire pour multicoptères existent dans la littérature. On mentionne d'abord la commande *backstepping* qui a été appliquée avec succès à la stabilisation et au suivi de trajectoire d'un quadricoptère [60, 61]. Dans [62], un intégrateur a été ajouté à l'algorithme de *backstepping* pour éliminer l'erreur en régime permanent du système. La commande adaptative est aussi employée pour contrôler les multicoptères en présence de paramètres variant lentement, mais totalement inconnus, tels que la masse et le moment d'inertie [63] ou le centre de gravité [64]. Afin d'améliorer la robustesse par rapport aux perturbations externes (vent), la commande adaptative $\mathcal{L}_1$ est appliquée dans [65]. D'autres techniques non linéaires telles que la commande par mode de glissement [66, 67], la commande robuste [68, 69], la commande optimale [70–72], la commande intelligente [73, 74] ont été exploitées pour le guidage et le pilotage de multicoptères.

Généralement, la plupart des techniques de contrôle présentent un compromis entre la performance et la complexité de la loi de commande. En effet, les contrôleurs linéaires sont conçus pour les vols quasi-stationnaires et ne peuvent pas fournir une bonne performance dans des conditions de vol non-nominales (grandes vitesses ou vols agressifs). D'autre part, les contrôleurs non linéaires sont souvent difficiles à régler en ligne et à mettre en œuvre sur de petits microcontrôleurs.

2.1.3.3 PFTC pour multicoptères

Parmi les méthodes passives tolérantes aux fautes, mentionnons la commande par mode de glissement (SMC) qui a été étudiée dans plusieurs travaux. Cette méthode a démontré une performance acceptable en présence d'une faute partielle faible sur une des hélices d'un quadricoptère [75]. Un autre contrôleur par mode glissant en ajoutant l'action intégrale peut compenser une perte de 15 % d'efficacité d'un rotor [76]. Dans ces travaux, la validation a

été effectuée en simulation et en tests expérimentaux sur un quadricoptère Qball-X4. Dans [77, 78], deux contrôleurs par mode glissant en cascade ont été utilisés pour tolérer les fautes d'actionneurs d'un quadricoptère AscTec Pelican . La boucle externe avec l'action intégrale contrôle l'altitude et les angles d'Euler, tandis que la boucle interne contrôle les vitesses de rotation. En simulation, un tel contrôleur peut tolérer une perte de 55% d'efficacité de deux rotors. Pour augmenter la capacité de tolérance aux fautes multiples, la commande par mode glissant du second ordre (SO-SMC) a été appliquée dans [79]. Les résultats en simulation sur un octocoptère ont démontré une très bonne performance de cette méthode. Elle peut compenser en même temps une perte de 15% d'efficacité du contrôle sur les moteurs 2, 4, 6 et une défaillance (panne totale) des moteurs 1, 3, 5 et 8.

2.1.3.4 AFTC pour multicoptères

Au cours des dix dernières années, de nombreuses techniques AFTC ont été développées pour garantir la sûreté de fonctionnement de multicoptères en présence de fautes. Le tableau 2.1 présente quelques exemples de références récentes dans ce domaine, classées en fonction de la méthode de diagnostic, de la synthèse de contrôle et de la configuration du multicoptère. Une revue de littérature sur le développement et les perspectives en commande tolérante aux fautes pour les drones est fournie dans [80]. Il est à noter que la technique d'auto-séquencement de gains via la synthèse $\mathcal{H}_\infty$ structurée n'a jamais été abordée dans le cadre de la commande tolérante aux fautes, et le sera pour la première fois dans cette thèse.

AFTC par ré-allocation de contrôle Du fait que le multicoptère est un système intégrant une redondance matérielle en actionneurs, cela permet un rétablissement du système en présence de fautes. En fait, chaque rotor joue un même rôle de manière à fournir la poussée et le moment. La technique de ré-allocation de contrôle (*control allocation* - CA) est donc une solution particulière pour l'AFTC des multicoptères. L'architecture d'un système AFTC utilisant la CA est présentée à la figure 2.7.

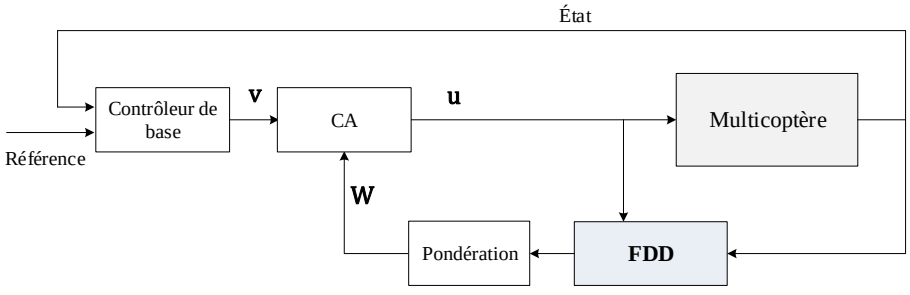


Figure 2.7 Principe de ré-allocation de contrôle (adapté de [162])

Tableau 2.1 Exemples de FTC pour les multicoptères

		Exemples de références
Technique FDD	KF	[41, 81, 82]
	KF à deux étages	[38–43, 83]
	EKF	[84, 85]
	UKF	[39, 83, 86, 87]
	Observateur	[88–99]
	Estimation de paramètres	[100–102]
	Espace de parité	[102, 103]
Technique FTC	Commande adaptative	[83, 104–118]
	Séquencement de gains	[83, 107, 108, 119–122]
	Multimodèle	[86, 123–126]
	LPV	[122, 127–129]
	Commande par mode de glissement passive	[75–79, 96, 98, 130–132]
	Commande par mode de glissement active	[94, 96, 98, 130, 131, 133–135]
	Backstepping	[98, 136, 137]
	Commande prédictive	[86, 138, 139]
	Ré-allocation de contrôle	[92, 100–102, 118, 126, 140–153]
	Logique floue (*Fuzzy*)	[120]
	Formation	[154–157]
	Autres	[158–160]
Configuration	Rotor contrarotatif	[109]
	Hexacoptère	[113, 139, 145, 150, 160]
	Octocoptère	[79, 100, 132, 143, 147]
	Octocoptère coaxial	[92–94, 110, 118, 161]
	Quadricoptère	*autres*

L'idée est de résoudre l'équation $\mathbf{v} = \mathbf{B}_{\text{CA}}[T_1 \ \dots \ T_m]^\top$ pour répartir les forces et les moments sur m moteurs via la commande virtuelle $\mathbf{v} = [T \ L \ M \ N]^\top$ (voir Éq. 3.20 pour de plus amples détails concernant ces notions). Dans ce cas, la solution donnée par la commande pseudo-inverse $\mathbf{u} \triangleq [T_1 \ \dots \ T_m]^\top = \mathbf{B}_{\text{CA}}^\dagger \mathbf{v}$ n'est pas unique. La répartition de contrôle peut être effectuée de la façon suivante :

$$\begin{aligned} &\min_{\mathbf{u}} \ \mathbf{u}^\top \mathbf{W}^{-1} \mathbf{u} \\ &\text{s.c. } \mathbf{B}_{\text{CA}} \mathbf{u} = \mathbf{v} \end{aligned} \tag{2.12}$$

où $\mathbf{W}$ est une matrice de pondération symétrique et définie positive [162]. La solution optimale de (2.12) est donnée par :

$$\mathbf{u} = \mathbf{W}\mathbf{B}_{\text{CA}}^\top \left(\mathbf{B}_{\text{CA}} \mathbf{W} \mathbf{B}_{\text{CA}}^\top \right)^{-1} \mathbf{v} \tag{2.13}$$

La matrice $\mathbf{W}$ est choisie de façon à prendre en compte les informations de fautes, les limita-

tions d'actionneurs ou certains objectifs d'optimisation. Ainsi, la CA est un atout exploitable dans le but de tolérer les fautes sans forcément reconfigurer le contrôleur, notamment en cas de perte d'actionneurs. De plus, elle présente des avantages de manière à optimiser certains objectifs ou à éviter la saturation d'actionneurs. Cependant, la dynamique et les limitations des actionneurs après une faute ne sont pas prises en compte dans la loi de commande. Cela signifie que le contrôleur essayera toujours d'obtenir la performance du système nominal même si les actionneurs ne sont pas capables de la réaliser [17]. Le lecteur peut se référer à [163, 164] pour les détails concernant la CA.

2.1.4 Stabilité généralisée

Nous terminons la revue de littérature sur des rappels généraux portant sur la stabilité généralisée des systèmes LTI et les applications gardiennes. Cette partie a pour but d'éclairer le lecteur sur les points abordés dans le chapitre 6.

2.1.4.1 Approches analytiques

L'analyse de la robustesse du système affecté par des perturbations structurées est un problème important et difficile en automatique. Ce problème est en général NP-difficile tel que mentionné dans [165–168]. Il est souvent possible de transformer le cas structuré en non structuré [169] pour l'étudier dans le domaine fréquentiel. En effet, l'analyse de la stabilité robuste basée sur le théorème du petit gain est relativement moins complexe. Néanmoins, cette transformation peut se ramener à une famille non structurée plus large que la famille originale et donc la conclusion sur l'analyse de la robustesse est plus conservative, c'est-à-dire une condition suffisante mais pas nécessaire [170].

En pratique, on le ramène souvent à l'étude de la stabilité généralisée d'un système linéaire invariant dans le temps (LTI) sous forme d'une famille de matrices ou de polynômes avec coefficients incertains [168]. Ce problème a reçu une attention importante de la part des chercheurs après le travail de Kharitonov [171] où il fournit un test de stabilité d'Hurwitz pour une famille de polynôme intervalle.[1] Une vue d'ensemble sur les approches d'analyse traditionnelles peut être trouvée dans les livres [172, 173].

Parmi les approches analytiques, mentionnons le théorème des bords (*Edge Theorem*) [174] qui permet de vérifier la stabilité généralisée d'une famille affine de polynômes avec coefficients incertains interdépendants. Ici, la stabilité généralisée, encore appelée $\mathcal{D}$-stabilité [172], est définie par le confinement des racines d'un polynôme ou des valeurs propres d'une matrice

1. Les coefficients du polynôme sont indépendants et bornées dans les intervalles spécifiées.

à l'intérieur d'une région Ω du plan complexe. Ce théorème est graphiquement démontré par le principe d'exclusion des zéros, en se basant sur la continuité des racines du polynôme par rapport aux variations de ses coefficients. On peut aussi essayer d'étudier la stabilité d'une famille de polynômes multilinéaires ou non linéaires à l'aide de ce théorème et d'un changement de variables convenable. Néanmoins, cette méthode, encore appelée *overbounding* en anglais, ne garantit pas la stabilité robuste de la famille originale dans plusieurs cas [172].

Concernant les familles de matrices incertaines, l'analyse de la stabilité robuste a été un sujet de recherche actif dans les années 80 et 90. On mentionne d'abord les études portant sur la stabilité robuste de matrices de type intervalle. Pour le cas symétrique, il suffit de tester la stabilité d'Hurwitz (ou de Schur) des sommets (*vertex matrices*) [175–177]. Cette vérification est aussi valide pour les matrices non symétriques de dimension 2×2 [178]. Quant aux matrices de type intervalle de dimension $n \times n$, la stabilité des matrices extrêmes ne peut garantir la stabilité de tout l'intervalle dans le cas général. Différentes conditions suffisantes ont été dérivées en utilisant différentes méthodes : théorie de Lyapunov [179–181], théorème de Gershgorin [182, 183], etc. En particulier, [184, 185] ont proposé des conditions nécessaires et suffisantes. [184] a montré qu'une matrice de type intervalle est robustement stable (Hurwitz/Schur) si et seulement si l'on peut diviser cet intervalle à un nombre fini de sous-intervalles qui satisfont des conditions suffisantes spécifiques. Ainsi, la charge de calcul dépend du conservatisme des conditions suffisantes appliquées. Par exemple, il est nécessaire de vérifier 11 345 sous-intervalles pour montrer la stabilité robuste d'une matrice de type intervalle de dimension 4×4 (cf. [184, Exemple 1]). L'effort de calcul excessif est aussi un défi pour la méthode d'analyse proposée par [185]. Dans ce cas, une matrice de type intervalle est robustement stable (Hurwitz/Schur) si et seulement si toutes les faces exposées à deux dimensions sont robustement stables. Pour une matrice de dimension 4×4, le nombre de faces à vérifier est de 1 966 080.

Pour les polyèdres de matrices (*matrix polytope*) [2] dont les matrices de type intervalle sont une sous-classe, [186, 187] ont présenté des conditions suffisantes de la stabilité d'Hurwitz et de Schur. Ces conditions suffisantes stipulent qu'il suffit de vérifier un nombre fini de faces du polyèdre [186, 187]. En pratique, cette méthode est efficace seulement pour les familles de matrices de faible dimension. Dans une autre recherche [188], une condition nécessaire et suffisante est établie en se basant sur une division du polyèdre de matrices en un nombre fini de sous-polyèdres. Cette approche, encore appelée *branch and bound* en anglais, présente toujours un défi concernant la surcharge de calcul.

En bref, les méthodes existantes dans la littérature impliquent d'étudier un nombre de sous-

2. Les coefficients dépendent de façon affine de paramètres incertains qui varient dans un polyèdre.

matrices qui augmente de manière exponentielle par rapport à la dimension de la famille incertaine [168]. Quelques études plus récentes [189–191] s'efforcent de réduire cette complexité de calcul pour certains cas particuliers. Par conséquent, trouver une condition nécessaire et suffisante simple pour la stabilité robuste d'une famille de matrices incertaines demeure d'une importance théorique et pratique considérable.

La plupart des approches analytiques abordées ci-dessus visent à vérifier la stabilité d'Hurwitz ou de Schur. Quant à la stabilité généralisée robuste, on mentionne particulièrement les applications gardiennes [192].

Dans cette thèse, le lien entre cette méthode d'analyse et la valeur singulière structurée (VSS) est en particulier étudié. On propose alors des nouvelles conditions nécessaires et suffisantes pour la stabilité généralisée robuste sous la forme de la VSS d'une matrice constante. Notre approche est applicable aux familles de matrices dépendant de façon affine ou rationnelle de paramètres incertains.

Applications gardiennes Développée par [192], une application gardienne est une fonction scalaire des paramètres incertains d'une famille de matrices ou de polynômes, qui s'annule dès que l'on est à la frontière du domaine de stabilité considéré Ω. Il est donc possible de déterminer les ensembles de paramètres qui assurent que la famille incertaine demeure stable par rapport à Ω. Contrairement aux autres approches qui sont souvent spécialisées pour une structure d'incertitude affine, les applications gardiennes ne se limitent pas à une structure spécifique. Ainsi, cette méthode d'analyse est assez efficace et puissante pour la stabilité généralisée [172]. Pour les cas multiparamétriques pour lesquels l'analyse graphique n'est pas faisable, les applications gardiennes permettent de ramener l'étude de stabilité généralisée à un problème classique de positivité d'un polynôme multivariable sur le domaine des paramètres. Par conséquent, il faut passer à des méthodes numériques pour obtenir l'information sur la stabilité généralisée du système.

2.1.4.2 Approches numériques

Profitant du développement de l'informatique, les approches d'analyse numériques ont été de plus en plus utilisées surtout pour les familles de matrices possédant une structure d'incertitude complexe. En effet, le problème de robustesse peut être reformulé facilement sous forme d'un problème de programmation [172]. Parmi les résultats récents, on peut mentionner les méthodes basées sur la théorie de stabilité de Lyapunov. Il s'agit de trouver une fonction de Lyapunov soit fixe et commune pour tous les membres de la famille, soit dépendant du paramètre incertain. Plusieurs auteurs ont transformé ce problème sous forme d'inégalités

matricielles linéaires (*Linear Matrix Inequalities* - LMI) et l'ont successivement résolu de multiples façons (cf. [193, 194] et leurs références). Ces méthodes garantissent une stabilité quadratique qui implique une stabilité robuste pour le cas d'incertitudes variant dans le temps. Le résultat d'analyse dispose néanmoins d'un certain conservatisme [194]. Quelques études plus récentes s'efforcent de réduire cet inconvénient, ainsi que la complexité de calcul des techniques LMI [195–197].

Parmi les méthodes non-Lyapunov, l'une des plus remarquables est la μ-analyse qui permet de traiter à la fois des incertitudes structurées et non structurées [170]. De plus, on peut aussi mentionner les approches numériques impliquant la résolution des LMI basées sur la matrice de Hermite [198], le déterminant de Hurwitz [193], ou les matrices augmentées [199]. Une comparaison détaillée de la complexité numérique est donnée dans [200].

μ-analyse La μ-analyse permet de mesurer la stabilité et la performance robuste des systèmes LTI exprimés sous la forme d'une transformation linéaire fractionnaire (*Linear Fractional Transformation* - LFT). Dans cette représentation, toutes les incertitudes structurées et non structurées sont rassemblées dans une matrice Δ. La robustesse est donc vérifiée en calculant la valeur singulière structurée μ (*Structured Singular Value* - SSV) par rapport à Δ sur une grille suffisamment dense de points le long de l'axe des fréquences [170, 201]. Bien que la μ-analyse soit un outil puissant, elle nécessite un effort de calcul excessif surtout pour les problèmes de grande dimension. En effet, le calcul de μ est lui-même un problème d'optimisation NP-difficile [168]. Sauf certains cas triviaux, seules les bornes supérieure et inférieure de μ sont calculées à l'aide d'algorithmes numériques.

Lien entre la μ-analyse et les applications gardiennes La combinaison de deux méthodes d'analyse est proposée par Chen *et al.* [202] pour le cas particulier d'une matrice de type intervalle. Dans ce travail, les auteurs ont fourni certaines conditions nécessaires et suffisantes (CNS) de la stabilité généralisée sous forme d'une seule VSS d'une matrice constante. Plus précisément, les CNS sont établies à travers l'utilisation de d'applications gardiennes construites à partir du produit de Kronecker et de sa représentation LFT. Par rapport à la μ-analyse, cette méthode permet d'éliminer la variable de balayage (*sweeping variable*) dans le domaine fréquentiel, et donc de diminuer le coût du calcul numérique.

2.2 Objectifs, méthodologie et contribution

À la lumière de la revue de littérature, les objectifs de la thèse, les méthodologies envisagées et les contributions associées seront présentés dans cette section.

L'objectif principal de ce projet de doctorat peut être résumé de la façon suivante :

Développer un système FTC actif permettant à un multicoptère de poursuivre sa mission avec une performance acceptable en présence de fautes sur les actionneurs ou de paramètres incertains.

Cet objectif de recherche est séparé en trois objectifs spécifiques qui seront détaillés ci-après.

2.2.1 Objectif 1 : développer un système FTC actif par des méthodes d'auto-séquencement de gains

Cet objectif consiste dans un premier temps à construire un sous-système de détection et de diagnostic basé sur la redondance analytique pour détecter et identifier la faute. Il s'agit de choisir une technique FDD fiable et sensible aux fautes afin de récupérer les informations de fautes le plus rapidement possible. Dans un second temps, un mécanisme de recouvrement permet au système de continuer à fonctionner normalement en dépit des fautes affectant ses différentes ressources (rotors, hélices, etc.). On cherche alors à préserver la stabilité du système et assurer la meilleure performance possible dans le cadre de la théorie de la commande linéaire. L'architecture du contrôleur doit être facilement *reconfigurable* pour faire face à l'apparition soudaine de fautes. Le défi réside dans le choix d'une stratégie de reconfiguration pour agir sur le contrôleur dès qu'une faute est identifiée par le module FDD.

Méthodologie et contribution

Le filtre de Kalman à deux étages sera d'abord utilisé comme algorithme de diagnostic. Comme mentionné précédemment, ce filtre permet d'estimer les états du système ainsi que les paramètres représentant la sévérité des fautes. En simulation, ce filtre a été employé dans le système AFTC complet d'un hélicoptère [203] et d'un drone à voilure fixe [40,41]. Cependant, une approche similaire pour multicoptères, en simulation ainsi qu'expérimentalement, n'a pas encore été rapportée dans la littérature. En notant que le filtre de Kalman à deux étages n'est pas applicable pour les systèmes sur-actionnés qui ont plus d'actionneurs que de degrés de contrôle (e.g., un hexacoptère), on envisagera par la suite un algorithme d'estimation de fautes basé sur la dynamique du multicoptère [204].

Quant au recouvrement actif, les méthodes FTC proposées dans cette thèse se basent sur la technique d'auto-séquencement de gains et la synthèse $\mathcal{H}_\infty$ structurée. Tout d'abord, les gains du contrôleur seront paramétrés sous forme de fonctions polynomiales de la perte d'efficacité de contrôle. Ensuite, les méthodes $\mathcal{H}_\infty$ structurées ayant des capacités de synthèse multi-modèles seront appliquées pour synthétiser les gains du contrôleur autoséquencé en venant

satisfaire les exigences de performance et de robustesse assignées dans le cahier des charges.

D'une part, le contrôleur autoséquencé à architecture fixe permet d'éviter les effets transitoires du mécanisme de reconfiguration dans les systèmes AFTC. La structure de séquencement de gains étant explicite, la complexité de mise en œuvre d'un tel contrôleur LPV est donc relativement faible. D'autre part, une combinaison de cette technique et de la synthèse $\mathcal{H}_\infty$ structurée permet d'améliorer la robustesse du système par rapport aux perturbations internes et externes. Bien que cette méthode de synthèse soit disponible dans plusieurs applications aéronautiques [9, 12–16], elle n'a pas encore été reportée dans la littérature.

2.2.2 Objectif 2 : Développer un système FTC actif générique par des méthodes adaptatives et robustes

Comme mentionné précédemment, la dynamique d'un multicoptère est fortement non linéaire alors que la plupart des techniques de diagnostic et de FTC dans la littérature sont souvent établies pour les modèles linéaires. Cette linéarisation peut impliquer une perte d'information à cause du changement du point d'équilibre lors de l'occurrence de fautes. Ce second objectif consiste à développer un système non linéaire AFTC pour faire face en même temps aux fautes et à la variation du point de fonctionnement. Du fait que les drones partagent les mêmes équations dynamiques, cet objectif consiste de plus à concevoir un seul contrôleur fixe et tolérant aux fautes qui peut être mis en œuvre sur la majorité des multicoptères.

Méthodologie et contribution

La méthodologie proposée pour cet objectif est basée sur la commande adaptative et robuste. Le système FTC se divise en deux boucles de contrôle : un contrôleur interne qui génère la commande virtuelle normalisée pour le problème de suivi de trajectoire en présence de fautes, et un autre adaptatif qui permet de rejeter des incertitudes du système et aussi d'harmoniser les contraintes de conception pour une plateforme spécifique. Le système de contrôle s'adapte à la faisabilité dynamique de la plateforme grâce à un choix approprié du modèle de référence. Afin d'améliorer la robustesse, la loi adaptative est augmentée par une modification robuste supplémentaire à l'aide de l'opérateur de projection. À la connaissance de l'auteur, il n'existe pas encore d'étude similaire sur la FTC des multicoptères dans la littérature.

2.2.3 Objectif 3 : analyser la robustesse du système FTC

En appliquant la synthèse $\mathcal{H}_\infty$ structurée, les méthodes FTC proposées ne garantissent pas la stabilité et les performances robustes du système bouclé hors points de synthèse. Une

validation *a posteriori* du système de commande est donc nécessaire. De plus, la robustesse du système tolérant aux fautes dépend fortement de la réponse du module FDD. En pratique, les imprécisions de diagnostic peuvent causer l'instabilité du système FTC. Une analyse de l'interaction entre les deux modules FDD et FTC intégrés dans un système AFTC complet est alors indispensable.

Méthodologie et contribution

L'analyse de stabilité robuste du système FTC sera effectuée à travers l'utilisation de la théorie des applications gardiennes. À cette fin, le lien entre cette méthode d'analyse et la valeur singulière structurée (VSS) sera étudié. Des nouvelles conditions nécessaires et suffisantes de la stabilité généralisée robuste sous forme de la VSS d'une matrice constante seront proposées. Cette approche est applicable pour les systèmes linéaires dépendant de façon affine ou rationnelle des paramètres incertains. À la connaissance de l'auteur, il n'existe pas encore d'étude similaire sur la stabilité généralisée dans la littérature.

2.2.4 Liste de publications

Les travaux effectués durant le doctorat du candidat ont été publiés ou soumis dans des journaux et des conférences [205–213]. Ces publications se divisent selon celles en lien avec le sujet de doctorat et celles effectuées en collobaration avec d'autres étudiants du laboratoire de robotique mobile et des systèmes autonomes.

Travaux principaux de doctorat

[1] D.-T. Nguyen, D. Saussié et L. Saydy, "Robust self-scheduled fault-tolerant control of a quadrotor UAV," dans *20th World Congress of the International Federation of Automatic Control, IFAC-PapersOnLine*, vol 50, no. 1, 2017, p. 5761-5767.

[2] D.-T. Nguyen, D. Saussié et L. Saydy, "Quaternion-based robust fault-tolerant control of a quadrotor UAV," dans *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2017, p. 1333-1342.

[3] D.-T. Nguyen, D. Saussié et L. Saydy, "Fault-tolerant control of a hexacopter UAV based on self-scheduled control allocation," dans *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, p. 385–393.

[4] D.-T. Nguyen, D. Saussié, et L. Saydy, "Design and Experimental Validation of Robust Self-Scheduled Fault-Tolerant Control Laws for a Multicopter UAV," *IEEE/ASME Transactions on Mechatronics*, 2020.

[5] D.-T. Nguyen, D. Saussié et L. Saydy, “Universal adaptive fault-tolerant control of a multicopter UAV,” *21th World Congress of the International Federation of Automatic Control (IFAC)*, 2020.

[6] D.-T. Nguyen, D. Saussié, et L. Saydy, “Necessary and sufficient conditions for robust stability of uncertain affine/rational systems based on guardian maps,” article de revue à soumettre, 2021.

Travaux en collaboration au sein du laboratoire

[1] A. Borowczyk, D.-T. Nguyen, A.P-V. Nguyen, D.Q. Nguyen, D. Saussié et J. Le Ny, “Autonomous landing of a quadcopter on a high-speed ground vehicle,” *Journal of Guidance, Control, and Dynamics*, p. 1–8, 2017.

[2] A. Borowczyk, D.-T. Nguyen, A.P-V. Nguyen, D.Q. Nguyen, D. Saussié et J. Le Ny, “Autonomous landing of a multirotor micro air vehicle on a hign velocity ground vehicle,” dans *20th World Congress of the International Federation of Automatic Control (IFAC)*, 2017, p. 10488-10494.

[3] C. Massé, O. Gougeon, D.-T. Nguyen et D. Saussié, “Modeling and control of a quadcopter flying in a wind field : A comparison between LQR and structured $\mathcal{H}_\infty$ control techniques,” dans *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, p. 1408–1417.

CHAPITRE 3 MODÉLISATION ET COMMANDE CLASSIQUE D'UN MULTICOPTÈRE

Ce chapitre est consacré à la modélisation du multicoptère et à la conception d'un contrôleur nominal. Dans un premier temps, nous simplifierons le modèle non linéaire d'un multicoptère standard sous certaines hypothèses simplificatrices, s'en suivra le modèle linéarisé qui nous permettra de synthétiser ultérieurement les lois de commande.

Dans un second temps, nous procèderons à la synthèse d'un régulateur linéaire quadratique (*Linear Quadratic Regulator* - LQR) par retour d'état avec une action intégrale. L'objectif est d'obtenir un contrôleur nominal qui fonctionne adéquatement dans des conditions favorables et possède une tolérance aux fautes de base. Néanmoins, la synthèse LQR s'effectue sur le modèle linéarisé qui n'est valide qu'au voisinage d'un seul point d'équilibre, celui correspondant un vol stationnaire. Nous porterons donc notre attention sur l'implantation du contrôleur linéaire sur le système non linéaire dans le but de prendre en compte le changement du point de fonctionnement durant le vol.

Nous terminerons par la description de notre montage expérimental et les aspects pratiques pour obtenir une bonne mise en œuvre des lois de commande.

3.1 Modélisation du multicoptère

Cette section rappelle les bases de l'étude du mouvement du corps rigide dans l'espace, rencontrées dans plusieurs documents [51, 213–216], avant de détailler la modélisation de deux multicoptères utilisés dans ce projet de recherche (quadricoptère et hexacoptère).

3.1.1 Préliminaires et notations

3.1.1.1 Référentiel et système de coordonnées

Pour étudier le mouvement d'un corps rigide dans l'espace, deux systèmes de coordonnées sont couramment définis : un repère inertiel fixe $\mathcal{F}_i$ dont l'origine est fixée de façon arbitraire en un point de l'espace, et un repère mobile $\mathcal{F}_b$ dont l'origine est située au centre de masse (CM) du corps (supposé confondu avec le centre de gravité).

Le repère inertiel utilisé dans cette thèse suit la convention ENU (*East-North-Up*), lié à la base $\{\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i\}$, défini à partir d'un plan tangent à la surface de la Terre. Le repère mobile $\mathcal{F}_b$ est lié à la base $\{\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b\}$ où $\mathbf{x}_b$ définit l'axe longitudinal du véhicule et pointe vers l'avant

de celui-ci, $\mathbf{y}_b$ définit l'axe latéral du véhicule et est orienté vers la gauche de celui-ci et $\mathbf{z}_b$ définit l'axe vertical du véhicule et pointe vers le haut (Fig. 3.1). Ce repère $\mathcal{F}_b$ correspond à la convention FLU (*Forward-Left-Up*) couramment utilisée en robotique.

Concernant le quadricoptère, la configuration "+" est utilisée dans cette thèse, c'est-à-dire les axes $\mathbf{x}_b$ et $\mathbf{y}_b$ sont portées par les bras. Pour l'hexacoptère, on choisit la configuration "X6" dont l'axe $\mathbf{x}_b$ est positionné entre deux bras.

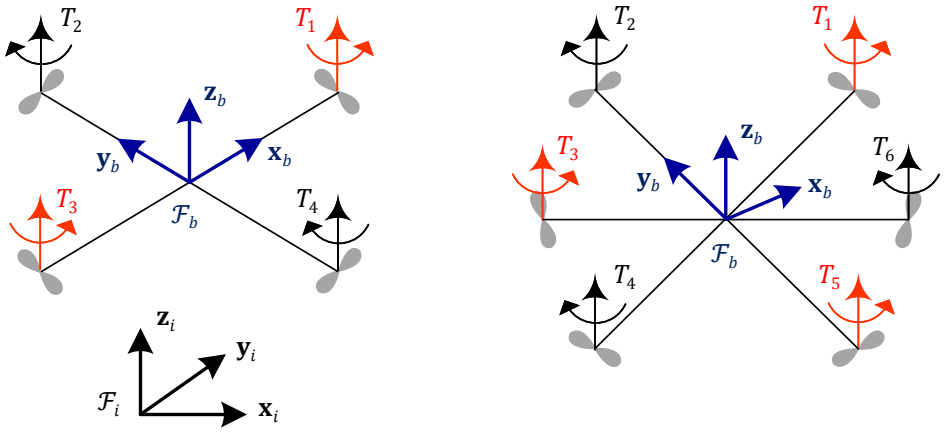


Figure 3.1 Configuration "+" d'un quadricoptère (à gauche) et configuration "X6" d'un hexacoptère (à droite)

Dans ce qui suit, l'exposant d'un vecteur physique indique le repère de référence dans lequel il est exprimé. Par exemple $\mathbf{v}_{cm/i}$ désigne le vecteur vitesse du CM par rapport au repère inertiel $\mathcal{F}_i$, et sa projection dans le repère mobile $\mathcal{F}_b$ est notée par $\mathbf{v}^b_{cm/i} \triangleq [u \;\; v \;\; w]^\top$.

Considérons un véhicule aérien possédant six degrés de liberté (*Degrees of Freedom* - DoF). Son mouvement dans l'espace à trois dimensions est déterminé par deux éléments :

— sa position $\mathbf{p}^i_{cm/i} = [x \; y \; z]^\top$, représentée par les coordonnées de l'origine du repère mobile $\mathcal{F}_b$, exprimées dans le repère de référence $\mathcal{F}_i$.

— l'attitude, ou l'orientation du repère $\mathcal{F}_b$ par rapport au repère $\mathcal{F}_i$. Cette orientation est représentée par une matrice de rotation $\mathbf{R}_{b/i}$ qui peut être paramétrisée par les trois angles d'Euler $\mathbf{\Phi} = [\phi \; \theta \; \psi]^\top$, ou le quaternion $\mathbf{q}$.

3.1.1.2 Angles d'Euler

La méthode la plus simple pour représenter l'orientation est d'utiliser les angles d'Euler $\mathbf{\Phi} = [\phi, \theta, \psi]^\top$. En aéronautique, le passage du repère inertiel $\mathcal{F}_i$ au repère mobile $\mathcal{F}_b$ s'effectue

en 3 étapes : une rotation d'angle ψ (lacet) autour de $\mathbf{z}_i$ puis une rotation d'angle θ (tangage) autour de $\mathbf{y}_1$ (axe intermédiaire obtenu après la première rotation) et enfin une rotation d'angle ϕ (roulis) autour de $\mathbf{x}_2$ (axe intermédiaire obtenu après la seconde rotation).

3.1.1.3 Matrice de rotation

Une matrice de rotation $\mathbf{R}$ appartient au sous-espace des matrices orthogonales de dimension 3, noté $SO(3)$ et défini par :

$$SO(3) = \{\mathbf{R} \,|\, \mathbf{R} \in \mathbb{R}^{3\times 3},\ \mathbf{R}^\top \mathbf{R} = \mathbf{I}_3,\ \det(\mathbf{R}) = 1\}$$

La matrice de rotation est orthogonale, donc $\mathbf{R}_{i/b} = \mathbf{R}_{b/i}^\top = \mathbf{R}_{b/i}^{-1}$ où $\mathbf{R}_{b/i}$ représente la rotation inverse, i.e., la transformation du repère mobile $\mathcal{F}_b$ au repère fixe $\mathcal{F}_i$. Elle est communément appelée en anglais *Direction Cosine Matrix* (DCM).

La paramétrisation de la matrice de rotation par les trois angles d'Euler $\mathbf{\Phi} = [\phi\ \theta\ \psi]^\top$ s'appelle couramment la matrice d'Euler et est donnée par (Annexe A.1) :

$$\mathbf{R}_{b/i} = \begin{bmatrix} \mathrm{c}_\theta \mathrm{c}_\psi & \mathrm{c}_\theta \mathrm{s}_\psi & -\mathrm{s}_\theta \\ \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{c}_\psi - \mathrm{c}_\phi \mathrm{s}_\psi & \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{s}_\psi + \mathrm{c}_\phi \mathrm{c}_\psi & \mathrm{s}_\phi \mathrm{c}_\theta \\ \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{c}_\psi + \mathrm{s}_\phi \mathrm{s}_\psi & \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{s}_\psi - \mathrm{s}_\phi \mathrm{c}_\psi & \mathrm{c}_\phi \mathrm{c}_\theta \end{bmatrix} \tag{3.1}$$

avec $\mathrm{c}_x = \cos(x)$ et $\mathrm{s}_x = \sin(x)$. On peut ainsi retrouver les angles d'Euler à partir de la matrice de rotation par les formules suivantes :

$$\phi = \arctan2(r_{23}, r_{33}),\quad \theta = -\arcsin(r_{13}) \ \text{ et } \ \psi = \arctan2(r_{12}, r_{11}) \tag{3.2}$$

où r_{ij} est l'élément de la i^{e} ligne et de la j^{e} colonne de la matrice $\mathbf{R}_{b/i}$, et arctan2 désigne la fonction arc tangente qui tient compte du quadrant.

Un inconvénient d'utiliser la paramétrisation d'Euler est l'apparition de singularité pour $\theta = \pm\pi/2$. Dans ce cas, on ne peut pas déterminer les angles ϕ et ψ à partir de la matrice de rotation. On rencontre ainsi le même problème dans l'équation cinématique d'un corps rigide (voir l'équation 3.11). Cette singularité peut être éliminée si l'orientation du corps est limitée. Par contre, la solution appropriée pour éviter ce problème est d'utiliser une représentation par quaternion.

3.1.1.4 Quaternion

Selon le théorème d'Euler [217], le repère mobile $\mathcal{F}_b$ peut être obtenu à partir du repère référentiel $\mathcal{F}_i$ par une seule rotation d'angle ϕ_a autour d'un vecteur unitaire $\mathbf{a} = [a_x \ a_y \ a_z]^\top$, exprimé dans $\mathcal{F}_i$. La matrice de rotation s'écrit alors :

$$\mathbf{R}_{b/i} = \cos\phi_a \mathbf{I}_3 + (1 - \cos\phi_a)\mathbf{a}\mathbf{a}^\top - \sin\phi_a [\mathbf{a}]_\times \tag{3.3}$$

$$= \left(2\cos^2\frac{\phi_a}{2} - 1\right)\mathbf{I}_3 + \left(2\sin^2\frac{\phi_a}{2}\right)\mathbf{a}\mathbf{a}^\top - 2\sin\frac{\phi_a}{2}\cos\frac{\phi_a}{2}[\mathbf{a}]_\times \tag{3.4}$$

où $[\mathbf{a}]_\times$ représente la matrice antisymétrique associée au vecteur $\mathbf{a} \in \mathbb{R}^3$, définie par :

$$[\mathbf{a}]_\times = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_\times \triangleq \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{3.5}$$

On définit ensuite le quaternion $\mathbf{q}$ qui se compose de deux parties, un scalaire q_0 et un vecteur $\mathbf{q}_{13} = [q_1 \ q_2 \ q_3]^\top$:

$$\mathbf{q} = \begin{bmatrix} q_0 \\ \mathbf{q}_{13} \end{bmatrix} = \begin{bmatrix} \cos(\phi_a/2) \\ \mathbf{a}\sin(\phi_a/2) \end{bmatrix} \in \mathbb{H} \tag{3.6}$$

où

$$\mathbb{H} = \{\mathbf{q} \,|\, \mathbf{q}^\top\mathbf{q} = 1, \mathbf{q} = [q_0 \ \mathbf{q}_{\mathbf{13}}^\top]^\top, q_0 \in \mathbb{R}, \mathbf{q_{13}} \in \mathbb{R}^3\} \tag{3.7}$$

La matrice de rotation est alors paramétrisée de la façon suivante [218] :

$$\mathbf{R}_{b/i} = C(\mathbf{q}) = (2q_0^2 - 1)\mathbf{I}_3 + 2\mathbf{q_{13}}\mathbf{q}_{\mathbf{13}}^\top - 2q_0[\mathbf{q_{13}}]_\times \tag{3.8}$$

Un quaternion $\mathbf{q}$ satisfaisant la condition (3.7) est appelé un *quaternion unitaire*. Un tel quaternion est donc une paramétrisation alternative de la matrice de rotation. La multiplication de deux quaternions quelconques, $\mathbf{q}_a = \left[q_{0_a} \ \mathbf{q}_{13_a}^\top\right]^\top$ et $\mathbf{q}_b = \left[q_{0_b} \ \mathbf{q}_{13_b}^\top\right]^\top$, est définie par :

$$\mathbf{q}_a * \mathbf{q}_b \triangleq \begin{bmatrix} q_{0_a} & -\mathbf{q}_{13_a}^\top \\ \mathbf{q}_{13_a} & \mathbf{I}_3 q_{0_a} + [\mathbf{q}_{13_a}]_\times \end{bmatrix} \mathbf{q}_b = \begin{bmatrix} q_{0_b} & -\mathbf{q}_{13_b}^\top \\ \mathbf{q}_{13_b} & \mathbf{I}_3 q_{0_b} - [\mathbf{q}_{13_b}]_\times \end{bmatrix} \mathbf{q}_a \tag{3.9}$$

Cette opération est non commutative, c'est-à-dire $\mathbf{q}_a * \mathbf{q}_b \neq \mathbf{q}_b * \mathbf{q}_a$. On utilise couramment le quaternion car il permet d'éviter les singularités géométriques. L'utilisation du quaternion pour effectuer des rotations en 3D présente aussi des avantages au niveau du coût des opérations mathématiques [219, 220]. Cependant, l'inconvénient de cette représentation est que le quaternion $\mathbf{q}$ et son opposé $-\mathbf{q}$ représentent la même attitude physique, c'est-à-dire

$C(\mathbf{q}) = C(-\mathbf{q})$. Pour $\mathbf{q}$, la rotation est effectuée avec un angle ϕ_a autour de $\mathbf{a}$, tandis que pour $-\mathbf{q}$, elle est effectuée avec un angle $\phi_a - 2\pi$, autour de $\mathbf{a}$ [215].

Pour résumer, l'utilisation des angles d'Euler est la plus simple et intuitive, mais son désavantage est la singularité géométrique. Les deux représentations d'attitude qui ne présentent pas de singularité sont le quaternion et la matrice de rotation. Néanmoins, ces représentations sont redondantes. La matrice de rotation a six paramètres redondants (9 éléments représentant 3 DoF) et le quaternion a une contrainte de norme unitaire (Éq. 3.7). La conversion entre ces méthodes est donnée à l'Annexe A.1.

3.1.2 Modèle dynamique d'un multicoptère

La modélisation de multicoptères est effectuée sous les hypothèses suivantes :

(**H1**) Le repère de référence $\mathcal{F}_i$ est un repère galiléen ;

(**H2**) Le multicoptère est un solide indéformable ayant une masse constante ;

(**H3**) Le multicoptère est symétrique dans les plans $\mathbf{x}_b$–$\mathbf{z}_b$ et $\mathbf{y}_b$–$\mathbf{z}_b$;

(**H4**) La dynamique des actionneurs (rotors et hélices) est suffisamment rapide pour pouvoir être négligée.

Ces hypothèses sont utilisées pour simplifier le modèle dynamique d'un multicoptère. En particulier, l'hypothèse (**H1**) permet de négliger les forces fictives. L'hypothèse (**H3**) implique que la matrice des moments d'inertie est diagonale. L'hypothèse (**H4**) permet d'utiliser directement les forces et les moments désirés comme entrées du système. Les équations générales du mouvement d'un hélicoptère peuvent alors s'énoncer comme suit.

Équations cinématiques L'équation de navigation relie la vitesse du drone dans le repère inertiel $\mathcal{F}_i$ et sa vitesse dans le repère mobile $\mathcal{F}_b$. Elle s'écrit :

$$\dot{\mathbf{p}}^i_{cm/i} = \mathbf{v}^i_{cm/i} = \mathbf{R}_{i/b}\mathbf{v}^b_{cm/i}. \tag{3.10}$$

L'équation cinématique d'Euler [218] permet quant à elle de relier la variation des angles d'Euler avec la vitesse angulaire $\boldsymbol{\omega}^b_{b/i} = [p\ q\ r]^\top$ entre les repères $\mathcal{F}_b$ et $\mathcal{F}_i$ de la façon suivante :

$$\dot{\boldsymbol{\Phi}} = \mathbf{H}(\boldsymbol{\Phi})\boldsymbol{\omega}^b_{b/i} \tag{3.11}$$

où $\mathbf{H}(\mathbf{\Phi})$ est la matrice de transformation de vitesses angulaires, donnée par :

$$\mathbf{H}(\mathbf{\Phi}) = \begin{bmatrix} 1 & \mathrm{s}_\phi \mathrm{t}_\theta & \mathrm{c}_\phi \mathrm{t}_\theta \\ 0 & \mathrm{c}_\phi & -\mathrm{s}_\phi \\ 0 & \mathrm{s}_\phi / \mathrm{c}_\theta & \mathrm{c}_\phi / \mathrm{c}_\theta \end{bmatrix} \quad \text{avec } \mathrm{t}_x := \tan x. \tag{3.12}$$

Si l'on s'intéresse à utiliser le quaternion au lieu des angles d'Euler, l'équation cinématique du quaternion est :

$$\dot{\mathbf{q}}_{b/i} = \frac{1}{2}\mathbf{q}_{b/i} * \begin{bmatrix} 0 \\ \boldsymbol{\omega}^b_{b/i} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\mathbf{q}_{13}^\top \\ \mathbf{I}_3 q_0 + [\mathbf{q}_{13}]_\times \end{bmatrix} \boldsymbol{\omega}^b_{b/i} = \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}^b_{b/i}{}^\top \\ \boldsymbol{\omega}^b_{b/i} & -[\boldsymbol{\omega}^b_{b/i}]_\times \end{bmatrix} \mathbf{q}_{b/i} \tag{3.13}$$

Enfin, l'équation cinématique de Poisson [218] est donnée par :

$$\dot{\mathbf{R}}_{b/i} = -\left[\boldsymbol{\omega}^b_{b/i}\right]_\times \mathbf{R}_{b/i} = \mathbf{R}_{b/i}\left[\boldsymbol{\omega}^i_{i/b}\right]_\times \tag{3.14}$$

où $\boldsymbol{\omega}^i_{i/b}$ est la vitesse de rotation instantanée du repère $\mathcal{F}_i$ par rapport au repère $\mathcal{F}_b$, exprimée dans $\mathcal{F}_i$, $\boldsymbol{\omega}^i_{i/b} = -\boldsymbol{\omega}^i_{b/i}$.

Équations dynamiques Il y a différentes approches pour décrire les équations dynamiques d'un objet volant, comme l'approche de Newton-Euler ou celle d'Euler-Lagrange. Dans cette thèse, l'approche de Newton-Euler est utilisée pour exploiter les concepts de force et de couple [218]. Le multicoptère est soumis à une force $\mathbf{F}^b$ et un moment $\mathbf{M}^b$, exprimés dans le repère mobile $\mathcal{F}_b$. Les équations dynamiques en translation et en rotation sont alors données par :

$$m\ddot{\mathbf{p}}^i_{cm/i} = m\mathbf{g}^i + \mathbf{R}_{i/b}\mathbf{F}^b \tag{3.15}$$

$$\mathbf{I}_b\dot{\boldsymbol{\omega}}^b_{b/i} = -\boldsymbol{\omega}^b_{b/i} \times \mathbf{I}_b\boldsymbol{\omega}^b_{b/i} + \mathbf{M}^b \tag{3.16}$$

où m est la masse du corps rigide, $\mathbf{g}^i = [0\ 0\ -g]^\top$ le vecteur d'accélération gravitationnelle, $\mathbf{I}_b \in \mathbb{R}^{3\times3}$ la matrice d'inertie du corps relativement au repère mobile $\mathcal{F}_b$ et $\times$ le produit vectoriel classique. $\mathbf{I}_b$ est symétrique et définie positive. Sous l'hypothèse (**H4**), la matrice d'inertie est de plus diagonale, c'est-à-dire $\mathbf{I}_b = \mathrm{diag}\,(I_{xx}, I_{yy}, I_{zz})$.

Il est important de noter que les forces et les moments aérodynamiques dus à la traînée en translation et en rotation de la structure, les moments gyroscopiques ainsi que les effets de sol ne seront pas considérés. Cela est dû au fait que l'on ne cherche pas un modèle parfait, mais plutôt un modèle simplifié pour les besoins de synthèse. Cette dynamique négligée peut être considérée comme une perturbation qui sera compensée par nos contrôleurs robustes et

adaptatifs. Ainsi, les vecteurs de force $\mathbf{F}^b$ et de moment $\mathbf{M}^b$ incluent seulement les forces et les moments dus à la poussée et les moments induits sur la structure. Le lecteur peut se référer à [213] pour une modélisation plus complète d'un multicoptère.

Force et moment Soit la configuration générale d'un multicoptère ayant m actionneurs (rotors et hélices) telle que montrée sur la figure 3.2.

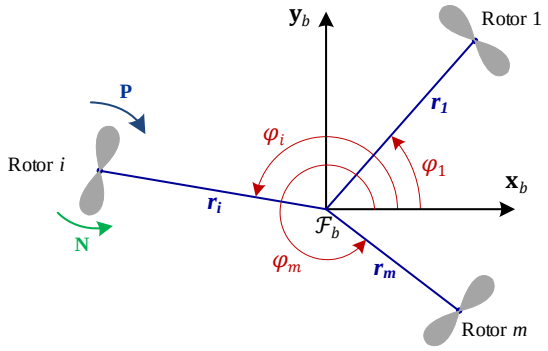


Figure 3.2 Géométrie d'un multicoptère : (r_i, φ_i) représente la position du i^{e} rotor dans le plan horizontal du repère $\mathcal{F}_b$; les lettres P et N définissent respectivement les couples de réaction positif et négatif autour de l'axe $\mathbf{z}_b$ qui pointe vers le haut.

Chaque actionneur i fournit une poussée T_i en direction de $\mathbf{z}_b$, modélisée par :

$$T_i = k_T \omega_i^2 \tag{3.17}$$

avec k_T le coefficient de poussée et ω_i la vitesse de rotation du rotor i. Le couple de réaction produit par chaque actionneur est déterminé par :

$$\tau_i = k_D \omega_i^2 \tag{3.18}$$

où k_D est le coefficient de traînée de l'hélice. Sous les hypothèses et les simplifications précédentes, les vecteurs de force et de moment dans le repère $\mathcal{F}_b$ sont définis comme :

$$\mathbf{F}^b = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}, \ \mathbf{M}^b = \begin{bmatrix} L \\ M \\ N \end{bmatrix} \tag{3.19}$$

où $T = \sum_{i=1}^{m} T_i$ représente la poussée totale des m rotors, L et M sont les moments selon les

axes $\mathbf{x}_b$ et $\mathbf{y}_b$ dus aux forces de poussée et N est le moment induit par les moteurs selon l'axe $\mathbf{z}_b$.

Sous l'hypothèse (**H4**), la relation entre les entrées virtuelles de commande $\mathbf{F}^b$ et $\mathbf{M}^b$, et les entrées internes du système (les forces propulsives des rotors) est donnée par :

$$\mathbf{v} = \mathbf{B}_{\text{CA}}\mathbf{u} \tag{3.20}$$

où $\mathbf{v} \triangleq [T\ L\ M\ N]^\top$ est le vecteur des entrées virtuelles et $\mathbf{u} \triangleq [T_1\ \ldots\ T_m]^\top$ est le vecteur des forces de propulsion des rotors. La matrice $\mathbf{B}_{\text{CA}} \in \mathbb{R}^{4\times m}$, nommée la matrice d'allocation de contrôle, est déterminée par :

$$\mathbf{B}_{\text{CA}} = \begin{bmatrix} 1 & \ldots & 1 \\ r_1 \sin(\varphi_1) & \ldots & r_m \sin(\varphi_m) \\ -r_1 \cos(\varphi_1) & \ldots & -r_m \cos(\varphi_m) \\ c\text{sign}(\omega_1) & \ldots & c\text{sign}(\omega_m) \end{bmatrix} \tag{3.21}$$

où $c = k_D/k_T$ est le facteur de changement d'échelle *force-à-moment*. La vitesse de rotation des moteurs ω_i est considérée positive dans le sens horaire selon l'axe $\mathbf{z}_b$.

Quadricoptère Pour la configuration NPNP d'un quadricoptère telle qu'illustrée à la figure 3.1, la matrice d'allocation de contrôle $\mathbf{B}_{\text{CA}} \in \mathbb{R}^{4\times 4}$ est la suivante :

$$\mathbf{B}_{\text{CA}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & d & 0 & -d \\ -d & 0 & d & 0 \\ -c & c & -c & c \end{bmatrix} \tag{3.22}$$

avec d la distance au centre de gravité (la longueur du bras).

Hexacoptère Pour la configuration NPNPNP d'un hexacoptère telle qu' illustrée à la figure 3.1, la matrice d'allocation de contrôle $\mathbf{B}_{\text{CA}} \in \mathbb{R}^{4\times 6}$ est la suivante :

$$\mathbf{B}_{\text{CA}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{d}{2} & d & \frac{d}{2} & -\frac{d}{2} & -d & -\frac{d}{2} \\ -\frac{d\sqrt{3}}{2} & 0 & \frac{d\sqrt{3}}{2} & \frac{d\sqrt{3}}{2} & 0 & -\frac{d\sqrt{3}}{2} \\ -c & c & -c & c & -c & c \end{bmatrix}. \tag{3.23}$$

3.1.2.1 Modèle non linéaire complet

À partir des équations cinématiques et dynamiques précédentes, le modèle non linéaire complet d'un multicoptère est caractérisé par :

$$\begin{cases} m\ddot{x} = T\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right) \\ m\ddot{y} = T\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right) \\ m\ddot{z} = T\cos\phi\cos\theta - mg \\ \dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \\ \dot{\theta} = q\cos\phi - r\sin\phi \\ \dot{\psi} = q\sin\phi/\cos\theta + r\cos\phi/\cos\theta \\ I_{xx}\dot{p} = (I_{yy} - I_{zz})qr + L \\ I_{yy}\dot{q} = (I_{zz} - I_{xx})rp + M \\ I_{zz}\dot{r} = (I_{xx} - I_{yy})pq + N \end{cases} \tag{3.24}$$

Les grandeurs physiques propres aux multicoptères utilisées sont donnés à l'annexe A.2.

3.1.2.2 Modèle linéarisé d'un multicoptère

Pour les besoins de synthèse, il est souvent nécessaire d'effectuer une linéarisation du modèle non linéaire (3.24) autour d'un point d'équilibre. Considérons un multicoptère en vol stationnaire ($T = mg$, $L = M = N = 0$ et $\phi = \theta = 0$) où chaque rotor délivre une poussée $T_i = T/n$ en direction de $\mathbf{z}_b$. Par simplicité, si l'on fait l'hypothèse que son CM est à l'origine de $\mathcal{F}_i$ et que l'angle de lacet ψ est nul, on obtient les équations linéarisées suivantes :

$$\begin{array}{rrr} \Delta\ddot{x} = g\Delta\theta & \Delta\dot{\phi} = \Delta p & I_{xx}\Delta\dot{p} = \Delta L \\ \Delta\ddot{y} = -g\Delta\phi & \Delta\dot{\theta} = \Delta q & I_{yy}\Delta\dot{q} = \Delta M \\ m\Delta\ddot{z} = \Delta T & \Delta\dot{\psi} = \Delta r & I_{zz}\Delta\dot{r} = \Delta N \end{array} \tag{3.25}$$

où Δ désigne la déviation d'une variable par rapport à sa valeur d'équilibre. Dès lors, le modèle d'état linéarisé est donné sous la forme :

$$\begin{cases} \boldsymbol{\Delta\dot{\mathbf{x}}} = \mathbf{A}\boldsymbol{\Delta\mathbf{x}} + \mathbf{B}\boldsymbol{\Delta\mathbf{v}} \\ \boldsymbol{\Delta\mathbf{y}} = \mathbf{C}\boldsymbol{\Delta\mathbf{x}} \end{cases} \tag{3.26}$$

où les vecteurs d'état $\mathbf{\Delta x}$, d'entrée $\mathbf{\Delta v}$ et de sortie à asservir $\mathbf{\Delta y}$ sont respectivement :

$$\begin{aligned}
\mathbf{\Delta x} &= \left[\Delta z\ \Delta \dot{z} \mid \Delta y\ \Delta \dot{y}\ \Delta \phi\ \Delta p \mid \Delta x\ \Delta \dot{x}\ \Delta \theta\ \Delta q \mid \Delta \psi\ \Delta r\right]^\top \\
\mathbf{\Delta v} &= \left[\Delta T\ \Delta L\ \Delta M\ \Delta N\right]^\top \\
\mathbf{\Delta y} &= \left[\Delta z\ \Delta y\ \Delta x\ \Delta \psi\right]^\top
\end{aligned} \tag{3.27}$$

Ce choix particulier du vecteur d'état $\mathbf{\Delta x}$ permet de rendre la matrice $\mathbf{A}$ diagonale par blocs où chacune des quatre entrées virtuelles du système affecte son propre bloc via la matrice $\mathbf{B}$:

$$\mathbf{A} = \left[\begin{array}{cc:cccc:cccc:cc}
0 & 1 & \multicolumn{4}{c:}{\multirow{2}{*}{$\mathbf{0}$}} & \multicolumn{4}{c:}{\multirow{2}{*}{$\mathbf{0}$}} & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}$}} \\
0 & 0 & \multicolumn{4}{c:}{} & \multicolumn{4}{c:}{} & \multicolumn{2}{c}{} \\
\hdashline
\multicolumn{2}{c:}{\multirow{4}{*}{$\mathbf{0}$}} & 0 & 1 & 0 & 0 & \multicolumn{4}{c:}{\multirow{4}{*}{$\mathbf{0}$}} & \multicolumn{2}{c}{\multirow{4}{*}{$\mathbf{0}$}} \\
\multicolumn{2}{c:}{} & 0 & 0 & -g & 0 & \multicolumn{4}{c:}{} & \multicolumn{2}{c}{} \\
\multicolumn{2}{c:}{} & 0 & 0 & 0 & 1 & \multicolumn{4}{c:}{} & \multicolumn{2}{c}{} \\
\multicolumn{2}{c:}{} & 0 & 0 & 0 & 0 & \multicolumn{4}{c:}{} & \multicolumn{2}{c}{} \\
\hdashline
\multicolumn{4}{c:}{\multirow{4}{*}{$\mathbf{0}$}} & 0 & 1 & 0 & 0 & \multicolumn{2}{c}{\multirow{4}{*}{$\mathbf{0}$}} \\
\multicolumn{4}{c:}{} & 0 & 0 & g & 0 & \multicolumn{2}{c}{} \\
\multicolumn{4}{c:}{} & 0 & 0 & 0 & 1 & \multicolumn{2}{c}{} \\
\multicolumn{4}{c:}{} & 0 & 0 & 0 & 0 & \multicolumn{2}{c}{} \\
\hdashline
\multicolumn{2}{c:}{\multirow{2}{*}{$\mathbf{0}$}} & \multicolumn{4}{c:}{\multirow{2}{*}{$\mathbf{0}$}} & \multicolumn{4}{c:}{\multirow{2}{*}{$\mathbf{0}$}} & 0 & 1 \\
\multicolumn{2}{c:}{} & \multicolumn{4}{c:}{} & \multicolumn{4}{c:}{} & 0 & 0
\end{array}\right] \quad
\mathbf{B} = \left[\begin{array}{cccc}
0 & 0 & 0 & 0 \\
1/m & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 1/I_{xx} & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1/I_{yy} & 0 \\
\hdashline
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1/I_{zz}
\end{array}\right]$$

où $\mathbf{0}$ représente la matrice nulle de dimensions appropriées. Cette représentation d'état met en évidence le découplage naturel du modèle linéaire. En fait, on peut le diviser facilement en quatre sous-systèmes distincts correspondant chacun à une dynamique du multicoptère : la dynamique d'altitude en z, la dynamique longitudinale en x et θ, la dynamique latérale en y et ϕ, et la dynamique de lacet en ψ. Cela permet de synthétiser séparément les lois de commande pour chaque sous-système en utilisant son modèle réduit.

De façon similaire, on peut obtenir les modèles non linéaire et linéarisé du multicoptère en utilisant le quaternion. Le lecteur peut se référer à l'article [206] pour plus de détails.

3.2 Conception et implémentation d'un contrôleur nominal

Nous allons à présent synthétiser un contrôleur nominal basé sur le modèle linéarisé (3.26), puis l'implémenter sur le système non linéaire en considérant le changement du point d'équilibre. Du fait de sa simplicité dans le design et sa performance naturelle en robustesse [57],

la synthèse LQR est choisie pour concevoir le contrôleur nominal en une seule étape.

3.2.1 Synthèse du contrôleur nominal par LQR

Avant de synthétiser la loi de commande nominale, le système linéarisé (3.26) est augmenté avec un nouvel état $\mathbf{\Delta e}_I$ qui représentent l'intégrale des erreurs de suivi, défini comme :

$$\mathbf{\Delta \dot{e}}_I = \underbrace{\mathbf{\Delta r} - \mathbf{\Delta y}}_{\mathbf{\Delta e}_I} = \mathbf{\Delta r} - \mathbf{C}\mathbf{\Delta x} \tag{3.28}$$

où $\mathbf{\Delta e}$ est le vecteur des erreurs de suivi et $\mathbf{\Delta r} = [\Delta z_r \ \Delta y_r \ \Delta x_r \ \Delta \psi_r]^\top$ est le vecteur des entrées de référence. Le modèle d'état augmenté est alors le suivant :

$$\begin{aligned} \begin{bmatrix} \mathbf{\Delta \dot{x}} \\ \mathbf{\Delta \dot{e}}_I \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0}_{12\times 4} \\ -\mathbf{C} & \mathbf{0}_4 \end{bmatrix} \begin{bmatrix} \mathbf{\Delta x} \\ \mathbf{\Delta e}_I \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0}_4 \end{bmatrix} \mathbf{\Delta v} + \begin{bmatrix} \mathbf{0}_{12\times 4} \\ \mathbf{I}_4 \end{bmatrix} \mathbf{\Delta r} \\ \mathbf{\Delta \dot{y}} &= \begin{bmatrix} \mathbf{C} & \mathbf{0}_4 \end{bmatrix} \begin{bmatrix} \mathbf{\Delta x} \\ \mathbf{\Delta e}_I \end{bmatrix}. \end{aligned} \tag{3.29}$$

La loi de commande classique par retour d'état avec action intégrale qui ramènera $\mathbf{\Delta \dot{z}}$ à zéro se met sous la forme :

$$\mathbf{\Delta v} = -\mathbf{K_x}\mathbf{\Delta x} + \mathbf{K}_{\mathbf{e}_I}\mathbf{\Delta e}_I. \tag{3.30}$$

Dans le cadre de la commande LQR à horizon infini, il s'agit de chercher les matrices de gains $\mathbf{K_x}$ et $\mathbf{K}_{\mathbf{e}_I}$ qui minimisent le critère de performance quadratique [221] :

$$\mathcal{J} = \frac{1}{2}\int_0^\infty \begin{bmatrix} \mathbf{\Delta x}^\top & \mathbf{\Delta e}_I^\top \end{bmatrix} \mathbf{Q} \begin{bmatrix} \mathbf{\Delta x} \\ \mathbf{\Delta e}_I \end{bmatrix} + \mathbf{\Delta v}^\top \mathbf{R} \mathbf{\Delta v} \ \mathrm{d}t \tag{3.31}$$

où les matrices de pondération $\mathbf{Q}$ et $\mathbf{R}$ sont respectivement semi-définie positive et définie positive, et souvent choisies diagonales. On cherche ainsi à minimiser l'énergie des états $\mathbf{\Delta x}$ sans trop solliciter la commande $\mathbf{\Delta u}$. L'amplitude relative de $\mathbf{Q}$ et de $\mathbf{R}$ permet de régler le compromis entre les états et la commande. En fait, une forte pondération sur $\mathbf{Q}$ va accélérer la dynamique et nécessiter de grandes commandes. Par contre, une forte pondération sur $\mathbf{R}$ va ralentir la dynamique et donc peu solliciter les actionneurs. Grâce aux connaissances acquises sur nos multicoptères durant les tests expérimentaux, on choisit les matrices suivantes :

$$\mathbf{Q} = \mathrm{diag}\,(50, 1, 50, 1, 500, 1, 50, 1, 500, 1, 75, 1, 150, 150, 150, 150)$$
$$\mathbf{R} = \mathrm{diag}\,(1, 1500, 1500, 1500)$$

La pondération sur la commande en moment est la plus grande pour que le drone ne soit pas trop agressif. Ainsi, la forte pondération portant sur les angles d'Euler permet d'obtenir une dynamique rapide en attitude. Cela garantit un bon fonctionnement du système de commande hiérarchique pour les multicoptères [53].

Le problème de la commande LQ a été initié et résolu en 1960 par [222]. Dans cette thèse, la fonction `lqr` de MATLAB est utilisée pour effectuer ce design préliminaire. Par une seule étape, la synthèse rend la structure de gains suivante :

$$\mathbf{K_x} = \left[\begin{array}{c|c|c|c} K_z\ K_{\dot{z}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & K_y\ K_{\dot{y}}\ K_\phi\ K_p & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & K_x\ K_{\dot{x}}\ K_\theta\ K_q & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & K_\psi\ K_r \end{array}\right] \tag{3.32}$$

$$\mathbf{K}_{\mathbf{e}_I} = \operatorname{diag}\left(K_{i,z}, K_{i,y}, K_{i,x}, K_{i,\psi}\right).$$

Les valeurs numériques des gains sont rassemblées dans l'annexe A.3.

3.2.2 Implémentation du contrôleur nominal

Le découplage naturel de la dynamique du drone (3.26) et la structure particulière des gains (3.32) permettent d'implémenter le contrôleur nominal en deux boucles tel que montré à la figure 3.3. La boucle externe commande la position du multicoptère dans le plan horizontal $\mathbf{x}_i$–$\mathbf{y}_i$ et la boucle interne commande les états plus rapides associés à l'altitude et l'attitude dans le but de maintenir la stabilité du multicoptère au point d'opération désiré. Pour une telle structure de commande hiérarchique, la boucle interne devrait fonctionner à un taux de rafraîchissement plus élevé que la boucle externe [53]. De plus, puisque la linéarisation s'effectue autour du point d'équilibre en vol stationnaire, il est nécessaire d'ajouter la commande à l'équilibre quant à la répartition des forces et des moments sur les rotors.

Tout d'abord, les lois de commande de l'altitude et du lacet sont établies sous la forme :

$$T = mg + K_{i,z}\int (z_r - z)\,\mathrm{d}t - K_z z - K_{\dot{z}}\dot{z} \tag{3.33}$$

$$N = K_{i,\psi}\int (\psi_r - \psi)\,\mathrm{d}t - K_\psi \psi - K_r r \tag{3.34}$$

Ensuite, la position du multicoptère dans le plan horizontal $\mathbf{x}_i$–$\mathbf{y}_i$ du repère inertiel est contrôlée par les commandes combinées des angles de roulis et de tangage. En notant u_x^i et u_y^i les sorties des contrôleurs de position en x et en y respectivement, les lois de commande

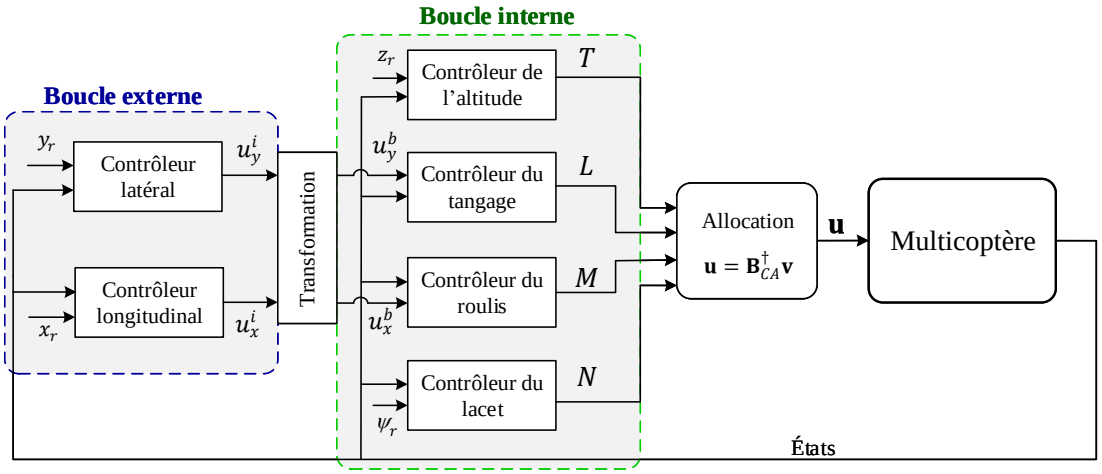


Figure 3.3 Architecture du contrôleur nominal

de la boucle externe s'écrivent :

$$\begin{aligned} u_x^i &= K_{i,x} \int (x_r - x)\,\mathrm{d}t - K_x x - K_{\dot{x}} \dot{x} \\ u_y^i &= K_{i,y} \int (y_r - y)\,\mathrm{d}t - K_y y - K_{\dot{y}} \dot{y} \end{aligned} \tag{3.35}$$

Dans les phases de linéarisation et de synthèse, on a supposé que l'angle de lacet ψ était nul. Or, en réalité, cet angle dépend de la direction prise par le drone selon les besoins pratiques. Dès lors, les signaux de commande (3.35) sont transformés au repère mobile en appliquant une rotation d'angle ψ autour de l'axe $\mathbf{z}_i$:

$$\begin{bmatrix} u_x^b \\ u_y^b \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} u_x^i \\ u_y^i \end{bmatrix} \tag{3.36}$$

Enfin, les lois de commande de la boucle interne sont données par :

$$\begin{aligned} L &= u_y^b - K_\phi \phi - K_p p \\ M &= u_x^b - K_\theta \theta - K_q q \end{aligned} \tag{3.37}$$

Une fois le vecteur des entrées virtuelles $\mathbf{v} = [T\ L\ M\ N]^\top$ déterminé, il suffit de répartir les commandes sur les rotors en utilisant :

$$\mathbf{u} = [T_1\ \ldots\ T_m]^\top = \mathbf{B}_{\mathrm{CA}}^{\dagger}\mathbf{v} \tag{3.38}$$

où $(\cdot)^\dagger$ dénote la matrice pseudo-inverse de Moore-Penrose.

3.3 Montage expérimental

3.3.1 De la synthèse à la validation

Dans le cadre de cette thèse, la synthèse des lois de commande est d'abord effectuée et simulée sous MATLAB/SIMULINK. Ensuite, nous utilisons ROTORS [1] pour simuler un banc d'essai sous Gazebo avec le code source en C++ généré à partir de SIMULINK en utilisant l'outil SIMULINK CODER. ROTORS, développé par [223], est un environnement de simulation assez complet pour les multicoptères sous la plateforme ROS (*Robot Operating System*). En effet, il tient compte de plusieurs aspects comme les effets aérodynamiques, la dynamique de rotors et toutes les composantes du drone (capteurs, hélices...). Cette phase de simulation permet de réduire des risques majeurs durant le vol, surtout la chute du multicoptère. Si le résultat est acceptable, la dernière étape consiste à valider les contrôleurs par des tests expérimentaux. Ceux-ci sont effectués dans la volière du laboratoire de robotique mobile et des systèmes autonomes (MRASL) avec le multicoptère *AscTec Firefly*. Une vue globale de notre montage expérimental est donnée à la figure 3.4.

3.3.2 AscTec Firefly

Le *AscTec Firefly* (Fig. 3.5), fabriqué par Ascending Technologies,[2] est un hexacoptère dédié à la recherche scientifique. Il est équipé d'un ordinateur de bord (le *AscTec Mastermind*, en anglais *On-Board Computer* - OBC) et d'une unité de commande de vol (le *AscTec Autopilot*, en anglais *Flight Control Unit* - FLU).

Le *AscTec Mastermind* est équipé d'un double cœur Atom de 1.66 GHz avec 4 Go de mémoire vive, et utilise le système d'exploitation Ubuntu 14.04 avec la version ROS Jade. Tous les calculs liés aux lois de commande sont effectués en ligne sur cet ordinateur de bord. Pour obtenir de bonnes performances dans un système de contrôle hiérarchique, la vitesse de fonctionnement est fixée à 1 KHz pour la boucle interne et à 200 Hz pour la boucle externe.

Le FCU *AscTec Autopilot* est équipé d'une unité de mesure inertielle (*Inertial Measurement Unit* - IMU) et de deux processeurs ARM7, un processeur de bas niveau (*Low Level Processor - LLP)* et un autre de haut niveau (*High Level Processor - HLP*). Ce FCU peut communiquer avec les différentes composantes du drone à haute vitesse en utilisant les protocoles standards SPI, I2C, UART, etc. En particulier, la communication entre le FCU et l'OBC est établie grâce à l'interface ROS [3] de HLP, qui communique ensuite avec le LLP via le protocole SPI.

1. `https://github.com/ethz-asl/rotors_simulator`
2. `http://www.asctec.de/en/asctec-research-uav`
3. `http://wiki.ros.org/asctec_mav_framework`

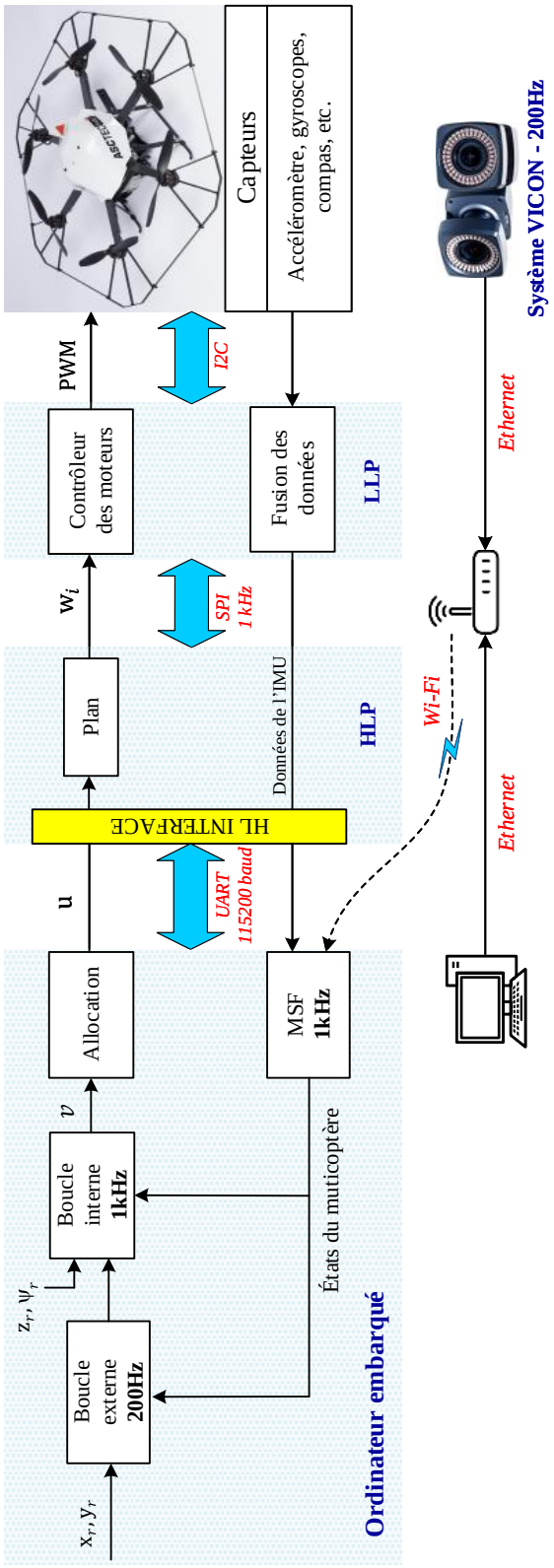


Figure 3.4 Architecture du montage expérimental pour l'AscTec Firefly

La fusion de données de différents capteurs et le contrôle des rotors s'effectuent dans le LLP. De plus, plusieurs méthodes de commande sont disponibles dans le LLP ; celle choisie est la commande de la vitesse de rotation des rotors correspondant au niveau de commande le plus bas. Plus précisément, la vitesse de rotation en rad/s générée par l'OBC sera représentée par un entier de 1 à 200 dans le HLP, puis envoyée aux contrôleurs de moteur dans le LLP.

Possédant un puissant OBC et un système de navigation inertielle (*Inertial Navigation System* - INS) de haute précision grâce aux capteurs avancés, ce multicoptère permet d'intégrer facilement les algorithmes de contrôle pour tester des systèmes de commande avec une fréquence pouvant aller jusqu'à 1.5 KHz.

Figure 3.5 AscTec Firefly (Source : http://www.asctec.de/)

3.3.3 Autres configurations

Les essais sont effectués dans un espace de vol qui est équipé d'un système de positionnement VICON (Fig. 3.6). Avec 12 caméras infrarouges, ce système mesure les positions des marqueurs attachés sur un drone en recevant la lumière réfléchie par eux. Cela permet de localiser précisément la position ainsi que l'orientation de l'engin volant. Le système VICON est en fait capable d'estimer la position avec une précision sous-millimétrique et les angles de sous-unités du degré, à une fréquence jusqu'à 250 Hz [224]. En utilisant ce système, la localisation est rarement perdue, même dans des situations extrêmes telles que des manœuvres très rapides [225]. Pour obtenir ces données, l'ordinateur de bord se connecte au serveur du VICON via un réseau périphérique de réalité virtuelle (*Virtual Reality Peripheral Network* - VRPN).[4] Dans le but d'obtenir des données plus précises et moins bruitées, les informations

4. `https://github.com/ethz-asl/ros_vrpn_client`

du VICON et de l'IMU sont combinées dans un module de fusion multi-capteurs (*MultiSensor Fusion Framework* - MSF).[5] Ce module fonctionne à un taux de rafraîchissement de 200 Hz qui est conforme à la vitesse de fonctionnement des autres processus de l'OBC.

Les autres tâches, telles que la surveillance à distance, l'initialisation des paramètres ou les commandes de haut niveau (références, décollage, atterrissage, dégradation de moteurs artificielle, etc.) sont traitées sur un autre ordinateur. Cette station de base communique avec l'OBC du multicoptère via une connexion Wifi.

Figure 3.6 Système VICON installé dans la volière du MRASL

5. https://github.com/ethz-asl/ethzasl_msf/wiki

CHAPITRE 4 COMMANDE TOLÉRANTE AUX FAUTES PAR DES MÉTHODES D'AUTO-SÉQUENCEMENT DE GAINS

Nous abordons à présent la conception de lois de commande tolérantes aux fautes par auto-séquencement de gains dans le cadre de la synthèse $\mathcal{H}_\infty$ structurée. À cette fin, trois approches de synthèse de contrôleurs séquencés sont proposées.

La première consiste à utiliser la perte d'efficacité des actionneurs (LAE) en tant que variable de séquencement. La configuration d'un hexacoptère est en particulier étudiée pour simplifier la structure de séquencement des gains et réduire le nombre de modèles à considérer dans le cadre de la synthèse multimodèle. Les critères de stabilité et de performance sont ensuite définis sous la forme de contraintes $\mathcal{H}_\infty$. La synthèse du contrôleur robuste et auto-séquencé est prise en charge par la fonction `systune` intégrée dans la *Robust Control Toolbox* de MATLAB. Dans le but de faciliter le processus de synthèse, la seconde approche consiste en l'utilisation de la perte d'efficacité de commande virtuelle (LVE). Cette méthode ne dépend pas de la configuration du multicoptère et permet de réduire significativement le nombre de paramètres ajustables lors de la synthèse. Enfin, une combinaison de la synthèse robuste et auto-séquencée avec la ré-allocation de contrôle (CA) est également envisagée pour notre troisième approche afin de tolérer les fautes des actionneurs du multicoptère sans modifier les lois de commande.

Du fait que les contrôleurs séquencés nécessitent la connaissance de l'amplitude des fautes (LAE ou LVE) pour actualiser les gains en temps réel, il s'agit d'abord de développer un module de détection et de diagnostic de fautes (FDD). Toutefois, il convient de noter que cet aspect n'est pas l'objectif principal de cette thèse. Les approches existantes sont donc empruntées pour construire un module FDD simple permettant d'estimer rapidement et précisément les fautes.

Après de brefs rappels sur les principes d'un module FDD, nous présenterons nos synthèses de FTC pour multicoptères. Nous terminerons par les résultats de simulations et expérimentaux qui ont été présentés dans trois articles de conférence [205–207], et un article de journal [209].

4.1 Estimation de fautes

La dégradation d'un rotor peut être calculée en comparant la vitesse angulaire désirée issue du contrôleur et sa vitesse réelle comme dans [226]. Les vitesses de rotation des rotors sont par exemple fournies par les encodeurs disponibles sur notre plateforme expérimentale. Or,

cette méthode ne couvre pas d'autres types de fautes telles que le bris d'hélices ou la rupture de liaisons mécaniques. Dans ce cas, on utilise deux techniques pour un module FDD : le filtre de Kalman à deux états et l'estimation de fautes basée sur la dynamique du multicoptère.

4.1.1 Représentation de fautes d'actionneurs

Soit le modèle d'état linéaire décrivant le comportement d'un système dynamique autour d'un point d'équilibre et sous les conditions de fonctionnement nominales :

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \\ \mathbf{z}(t) = \mathbf{M}\mathbf{y}(t) + \mathbf{v}(t) \end{cases} \tag{4.1}$$

avec $\mathbf{u}(t) \in \mathbb{R}^m$ le vecteur d'entrée de commande, $\mathbf{x}(t) \in \mathbb{R}^n$ le vecteur d'état, $\mathbf{y}(t) \in \mathbb{R}^p$ le vecteur des sorties régulées et $\mathbf{z}(t) \in \mathbb{R}^q$ le vecteur des sorties mesurées. La perturbation $\mathbf{w}(t) \in \mathbb{R}^n$ et l'erreur de mesure $\mathbf{v}(t) \in \mathbb{R}^q$ sont des bruits gaussiens et indépendants.

Les fautes d'actionneurs sont souvent représentées par la perte d'efficacité de contrôle (LAE) et modélisées de la façon suivante [227] :

$$\mathbf{u}_a(t) = [\mathbf{I}_m - \mathbf{\Gamma}(t)]\mathbf{u}(t) \tag{4.2}$$

où $\mathbf{u}_a(t)$ est la sortie des actionneurs et la matrice $\mathbf{\Gamma}(t) = \text{diag}(\gamma_1, \ldots, \gamma_m)$, $0 \leq \gamma_i \leq 1$, indique leur mode de fonctionnement. De cette façon, $\gamma_i = 0$ signifie que le i^{e} actionneur fonctionne normalement, tandis que $\gamma_i = 1$ représente une panne totale. En d'autres termes,

$$\mathbf{u}_a(t) = \mathbf{u}(t) - \mathbf{U}(t)\boldsymbol{\gamma}(t) \tag{4.3}$$

avec $\mathbf{U}(t) = \text{diag}\,(u_1(t),\, u_2(t),\, \ldots,\, u_m(t))$ et $\boldsymbol{\gamma}(t) = [\gamma_1(t),\, \gamma_2(t),\, \ldots,\, \gamma_m(t)]^\top$. On obtient alors le modèle d'état augmenté des fautes d'actionneurs sous la forme :

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) - \mathbf{B}\mathbf{U}(t)\boldsymbol{\gamma}(t) + \mathbf{w}(t) \\ \dot{\boldsymbol{\gamma}}(t) = \mathbf{0} \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) - \mathbf{D}\mathbf{U}(t)\boldsymbol{\gamma}(t) \\ \mathbf{z}(t) = \mathbf{M}\mathbf{y}(t) + \mathbf{v}(t) \end{cases} \tag{4.4}$$

En pratique, plusieurs études reportées dans la littérature sur l'estimation de fautes (e.g., [35–41]) utilisent le modèle discret équivalent du système linéaire (4.4).

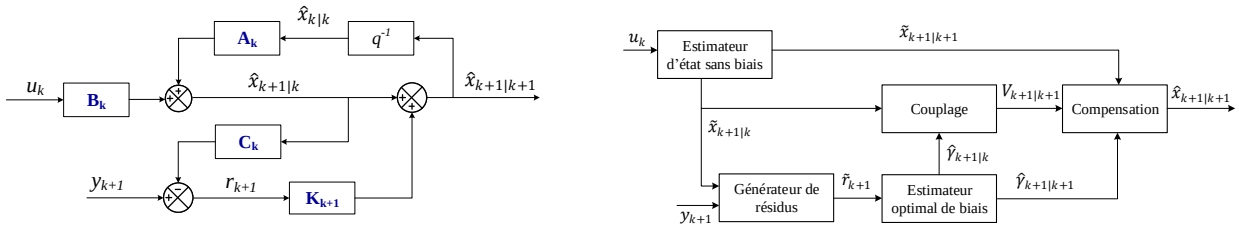
4.1.2 Filtre de Kalman à deux étages

Soit la représentation d'état à temps discret :

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k - \mathbf{B}_k\mathbf{U}_k\boldsymbol{\gamma}_k + \mathbf{w}_k^{\mathbf{x}} \\ \boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k + \mathbf{w}_k^{\gamma} \\ \mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k \end{cases} \tag{4.5}$$

avec les mêmes notations précédemment définies.

De manière à estimer $\boldsymbol{\gamma}_k$, il est possible d'utiliser un filtre de Kalman classique pour le système augmenté avec le vecteur d'état $\mathbf{x}_{ak} = [\mathbf{x}_k\ \boldsymbol{\gamma}_k]^\top$. Néanmoins, la dimension $n+m$ du filtre pourrait être un inconvénient qui limite sa mise en place, surtout pour les multicoptères possédant plusieurs actionneurs. Le filtre à deux étages (*Two-stage Kalman Filter* - TSKF) est donc proposé dans le but de réduire l'ordre du filtre de Kalman original [35]. Son principe de fonctionnement est illustré à la figure 4.1.



(a) Filtre de Kalman classique

(b) TSKF (adapté de [39])

Figure 4.1 Comparaison du filtre de Kalman et de TSKF

Selon le diagramme 4.1b, le TSKF peut être divisé en deux sous-systèmes. Le premier est un filtre de Kalman classique d'ordre n qui ne tient pas compte des fautes d'actionneurs. Le deuxième est un estimateur optimal de fautes d'ordre m. Dès lors, les états et les fautes sont simultanément estimés en combinant les deux filtres d'ordre réduit. Similairement au filtre de Kalman (Fig. 4.1a), le TSKF a aussi deux phases distinctes : une phase de prédiction et une phase de mise-à-jour. Dans la première phase identifiée par l'indice $k+1|k$, l'état estimé de l'instant précédent est utilisé pour produire une estimation de l'état courant. Ensuite, l'étape de mise-à-jour identifiée par l'indice $k+1|k+1$ utilise les mesures pour corriger l'état prédit. Dans cette phase, les équations de découplage sont ajoutées pour tenir compte de la présence des fautes. L'algorithme détaillé de ce filtre, disponible dans [35, 39], est résumé dans l'annexe B.1.

Dans les travaux reportés dans la littérature, le filtre de Kalman à deux étages a été utilisé

pour estimer les fautes des actionneurs et des capteurs d'un quadricoptère, tant en simulation que lors de vols expérimentaux (e.g., [38–43, 83]). Néanmoins, cette approche ne s'applique pas aux systèmes sur-actionnés qui ont plus d'actionneurs que de degrés de contrôle, e.g., un hexacoptère. Cela est évidemment lié au fait que le problème d'allocation de contrôle (3.38) n'a pas de solution unique. Par conséquent, le filtre de Kalman ne peut pas distinguer les actionneurs défectueux.

4.1.3 Estimation de fautes basée sur la dynamique de l'hexacoptère

Dans le cas particulier d'un hexacoptère, l'étude [204] exploite sa structure symétrique pour estimer la perte d'efficacité de chaque moteur en utilisant uniquement les informations de l'unité de mesure inertielle (IMU). À partir de (3.20) et (3.23), on obtient :

$$\mathbf{M}^b = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{d}{2} & d & \frac{d}{2} \\ -\frac{\sqrt{3}d}{2} & 0 & \frac{\sqrt{3}d}{2} \\ -c & c & -c \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} T_1 - T_4 \\ T_2 - T_5 \\ T_3 - T_6 \end{bmatrix} \tag{4.6}$$

Notons que notre multicoptère n'est pas équipé d'un capteur d'accélération angulaire. Le moment $\mathbf{M}^b$ est donc calculé à partir de (3.16) en utilisant seulement la vitesse de rotation $\boldsymbol{\omega}^b_{b/i}$ mesurée par l'IMU et l'estimation de l'accélération angulaire $\dot{\boldsymbol{\omega}}^b_{b/i}$ (présentée ci-dessous). Le résidu est alors défini comme étant :

$$\mathbf{R} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix} = \begin{bmatrix} T_{1,c} - T_{4,c} \\ T_{2,c} - T_{5,c} \\ T_{3,c} - T_{6,c} \end{bmatrix} - \mathbf{P}^{-1} \begin{bmatrix} L \\ M \\ N \end{bmatrix} \tag{4.7}$$

où $T_{i,c}$ $(i = 1, \ldots, 6)$ représente la force de poussée demandée au i^{e} rotor. En supposant que la faute ne se produise que sur les trois premiers actionneurs, la LAE est déterminée par :

$$\gamma_i = \frac{R_i}{T_{i,c}}, \quad i = 1, 2, 3 \tag{4.8}$$

Selon les résultats de simulation reportés dans [204], la réponse de cet estimateur est assez rapide et fiable. Or, à travers nos propres tests expérimentaux, nous trouvons qu'il dépend fortement de bruits de mesures et de la précision de l'estimation de l'accélération $\dot{\boldsymbol{\omega}}^b_{b/i}$. Dans cette thèse, nous l'utiliserons seulement pour vérifier la robustesse du système FTC lors d'une mauvaise estimation de fautes.

Estimation de l'accélération angulaire Au lieu de dériver directement la vitesse angulaire fournie par l'IMU, en général très bruitée, on utilise un filtre de Kalman classique pour estimer l'accélération $\dot{\boldsymbol{\omega}}^b_{b/i}$. Pour cela, on prend le modèle cinématique du second ordre tiré de [228] où le vecteur d'à-coup, plus communément appelé *jerk*, $\ddot{\boldsymbol{\omega}}(t) = \mathbf{w}(t)$ est un bruit blanc gaussien. Le modèle d'état à temps discret est donné par :

$$\begin{bmatrix} \boldsymbol{\omega}_{k+1} \\ \dot{\boldsymbol{\omega}}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & T_s\mathbf{I}_3 \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_k \\ \dot{\boldsymbol{\omega}}_k \end{bmatrix} + \begin{bmatrix} (T_s^2/2)\mathbf{I}_3 \\ \mathbf{I}_3 \end{bmatrix} \mathbf{w}_k \tag{4.9}$$

où T_s est la période d'échantillonnage. Les écart-types de bruits de processus $\mathbf{w}_k$ et de mesures sont choisis *a priori* durant un processus de réglage empirique (respectivement $5°/s^3$ et $1°/s$ lors de nos expériences).

4.2 Auto-séquencement de gains par LAE

Nous abordons dans cette section la conception du premier correcteur tolérant aux fautes par la méthode d'auto-séquencement de gains. L'architecture du système linéarisé en boucle fermée employée à des fins de design est illustrée à la figure 4.2. Contrairement à la synthèse LQR effectuée une seule fois sur la base du modèle linéarisé complet (cf. Sous-Section 3.2.1), la synthèse de FTC est constituée de quatre contrôleurs par retour d'état avec composante intégrale. Cela permet de faciliter le processus de design comme on le verra par la suite. D'autre part, la ré-allocation de contrôle n'est plus nécessaire, car la commande est directement les forces que doivent générer les rotors.

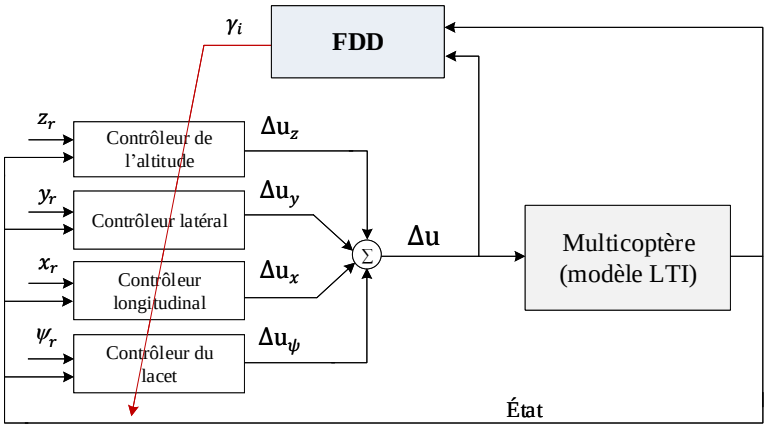


Figure 4.2 Architecture de FTC basée sur LAE utilisée à des fins de synthèse

En raison de la similarité dans le processus de synthèse, on présente seulement la conception du contrôleur longitudinal concernant la dynamique en x et θ pour un hexacoptère. Les autres

sont fournies en annexe B.3. Dans le cas d'un quadricoptère, le lecteur est invité à se référer à [205, 206] pour plus de détails.

4.2.1 Synthèse multimodèle et auto-séquencement de gains

Sur la base des équations (3.23) et (3.25), on obtient le modèle d'état réduit de la dynamique longitudinale d'un hexacoptère incluant le facteur LAE utilisé à des fins de synthèse :

$$\boldsymbol{\Delta}\dot{\mathbf{x}}_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\Delta}\mathbf{x}_x + \begin{bmatrix} \mathbf{0} \\ \\ \frac{d\sqrt{3}}{2I_{yy}}\boldsymbol{\Lambda}_x \end{bmatrix} \boldsymbol{\Delta}\mathbf{u}_x \tag{4.10}$$

où le vecteur d'état réduit est $\boldsymbol{\Delta}\mathbf{x}_x = [\Delta x \ \Delta\dot{x} \ \Delta\theta \ \Delta q]^\top$, le vecteur d'entrée de commande $\boldsymbol{\Delta}\mathbf{u}_x = [-\Delta T_1 \ \Delta T_3 \ \Delta T_4 \ -\Delta T_6]^\top$ et les fautes d'actionneurs modélisées par le vecteur ligne :

$$\boldsymbol{\Lambda}_x = \begin{bmatrix} 1-\gamma_1 & 1-\gamma_3 & 1-\gamma_4 & 1-\gamma_6 \end{bmatrix}. \tag{4.11}$$

Rappelons que γ_i représente la perte d'efficacité d'actionneur variant entre 0 (i^{e} rotor complètement sain) et 1 (panne totale). Le nombre de modèles décrivant le comportement du système est donc donné par $N = n^k$ où n est le nombre de niveaux défectueux et k est le nombre d'actionneurs défectueux. Cependant, en consultant la structure symétrique du multicoptère, on peut négliger de nombreux modèles défectueux pour simplifier le processus de design. Par exemple, $\gamma_1 = 0.5$ et $\gamma_6 = 0.5$ présentent le même effet sur la dynamique longitudinale en x et θ d'un hexacoptère. Ainsi, le nombre de modèles considérés est le nombre de k-combinaisons avec répétition d'un ensemble fini de cardinal n :

$$N = \frac{(n+k-1)!}{k!(n-1)!}. \tag{4.12}$$

Dans le cadre de la synthèse multimodèle, le domaine incertain est discrétisé comme suit : $\gamma_i \in \{0, 0.1, \ldots, 0.8\}$, $\forall i \in \{1, 3, 4, 6\}$. De cette façon, on considère $N = 495$ modèles liés aux 9 valeurs discrétisées de chaque γ_i au lieu de $9^4 = 6561$ modèles. Notons que l'on ne tient pas compte des grandes valeurs de γ_i (e.g., $\gamma_i = 0.9$ ou $\gamma_i = 1$) qui peuvent provoquer une violation des contraintes de design, des gains trop élevés du contrôleur obtenu, ou pire encore, une divergence du processus de synthèse. Bien évidemment, la robustesse du système sur l'ensemble du domaine incertain incluant la panne de moteurs sera vérifiée à la fin de la phase de synthèse.

En se basant sur la connaissance préalable du système obtenue au cours de la synthèse du contrôleur de base LQR, on choisit *a priori* la loi de commande par retour d'état avec action intégrale sous la forme :

$$\boldsymbol{\Delta}\mathbf{u}_x = -\mathbf{K}\boldsymbol{\Delta}\mathbf{x}_x + \mathbf{K}_i \int (\Delta x_r - \Delta x)\,\mathrm{d}t. \tag{4.13}$$

De manière à maintenir la stabilité et les performances acceptables du système en boucle fermée sur l'ensemble de points d'opération, les gains du contrôleur sont exprimés *a priori* sous forme de polynômes de la variable de séquencement γ_i.[1] Plus spécifiquement, on propose le séquencement de gains suivant :

$$\begin{aligned} \mathbf{K} &= \mathbf{K}_0 + \mathbf{K}_1\gamma_1 + \mathbf{K}_2\gamma_3 + \mathbf{K}_3\gamma_4 + \mathbf{K}_4\gamma_6 \\ \mathbf{K}_i &= \mathbf{K}_{i_0} + \mathbf{K}_{i_1}\gamma_1 + \mathbf{K}_{i_2}\gamma_3 + \mathbf{K}_{i_3}\gamma_4 + \mathbf{K}_{i_4}\gamma_6 \end{aligned} \tag{4.14}$$

Les coefficients des matrices $\mathbf{K}_j \in \mathbb{R}^{4\times 4}$ et $\mathbf{K}_{i_j} \in \mathbb{R}^4$, $j = 0, \ldots, 4$, sont les paramètres ajustables du contrôleur séquencé. De la sorte, il y a 100 paramètres que les méthodes numériques telles que `systune` doivent ajuster pour rencontrer le cahier des charges sur l'ensemble des modèles. Les autres structures de séquencement de gains telles que les fonctions bilinéaire ou quadratique sont admissibles. Cependant, elles sont plus coûteuses en temps de calcul lors de la synthèse et leur implémentation nécessite aussi un effort de calcul supplémentaire. De plus, la structure symétrique d'un hexacoptère nous permet d'imposer la structure de séquencement de gains suivante :

$$\begin{aligned} \mathbf{K} = &\begin{bmatrix} a_{0x} & b_{0x} & c_{0x} & d_{0x} \\ a_{0x} & b_{0x} & c_{0x} & d_{0x} \\ a_{0x} & b_{0x} & c_{0x} & d_{0x} \\ a_{0x} & b_{0x} & c_{0x} & d_{0x} \end{bmatrix} + \begin{bmatrix} a_{1x} & b_{1x} & c_{1x} & d_{1x} \\ a_{2x} & b_{2x} & c_{2x} & d_{2x} \\ a_{2x} & b_{2x} & c_{2x} & d_{2x} \\ a_{2x} & b_{2x} & c_{2x} & d_{2x} \end{bmatrix}\gamma_1 + \begin{bmatrix} a_{2x} & b_{2x} & c_{2x} & d_{2x} \\ a_{1x} & b_{1x} & c_{1x} & d_{1x} \\ a_{2x} & b_{2x} & c_{2x} & d_{2x} \\ a_{2x} & b_{2x} & c_{2x} & d_{2x} \end{bmatrix}\gamma_3 \\ &+ \begin{bmatrix} a_{2x} & b_{2x} & c_{2x} & d_{2x} \\ a_{2x} & b_{2x} & c_{2x} & d_{2x} \\ a_{1x} & b_{1x} & c_{1x} & d_{1x} \\ a_{2x} & b_{2x} & c_{2x} & d_{2x} \end{bmatrix}\gamma_4 + \begin{bmatrix} a_{2x} & b_{2x} & c_{2x} & d_{2x} \\ a_{2x} & b_{2x} & c_{2x} & d_{2x} \\ a_{2x} & b_{2x} & c_{2x} & d_{2x} \\ a_{1x} & b_{1x} & c_{1x} & d_{1x} \end{bmatrix}\gamma_6 \end{aligned} \tag{4.15}$$

et

$$\begin{aligned} \mathbf{K}_i = &\begin{bmatrix} i_{0x} & i_{0x} & i_{0x} & i_{0x} \end{bmatrix}^\top + \begin{bmatrix} i_{1x} & i_{2x} & i_{2x} & i_{2x} \end{bmatrix}^\top \gamma_1 + \begin{bmatrix} i_{2x} & i_{1x} & i_{2x} & i_{2x} \end{bmatrix}^\top \gamma_3 \\ &+ \begin{bmatrix} i_{2x} & i_{2x} & i_{1x} & i_{2x} \end{bmatrix}^\top \gamma_4 + \begin{bmatrix} i_{2x} & i_{2x} & i_{2x} & i_{1x} \end{bmatrix}^\top \gamma_6 \end{aligned} \tag{4.16}$$

1. Théoriquement, tout autre type de fonction est admissible.

De cette façon, si la faute apparaît sur le premier rotor, sa commande individuelle est actualisée par $[a_{1x} \;\; b_{1x} \;\; c_{1x} \;\; d_{1x}]$ et i_{1x}. La commande des autres actionneurs est par ailleurs ajustée via les paramètres $[a_{2x} \;\; b_{2x} \;\; c_{2x} \;\; d_{2x}]$ et i_{2x}. Il en résulte que le nombre de paramètres ajustables est réduit de 100 à 15.

Parmi les méthodes numériques, on utilise la fonction MATLAB `systune` [13] pour optimiser le séquencement de gains en fonction des contraintes dans le cadre de la synthèse $\mathcal{H}_\infty$ structurée. Les détails concernant cette phase de synthèse sont présentés dans les sous-sections suivantes.

4.2.2 Synthèse $\mathcal{H}_\infty$ structurée

La capacité multimodèle de la synthèse $\mathcal{H}_\infty$ structurée permet de procéder à l'auto-séquencement de gains. En se basant sur les récents travaux tirant profit de cette méthode en aéronautique [9, 12–16], on propose de l'appliquer pour résoudre le problème FTC. On présente ici le processus utilisé pour synthétiser et ajuster les lois de commande. Pour plus de détails sur le concept théorique et sur la mise en œuvre de la synthèse robuste à architecture fixe, le lecteur peut se référer à l'annexe B.2.

Dans le cadre de la synthèse $\mathcal{H}_\infty$ structurée, les contraintes imposées par le cahier des charges sont reformulées dans le formalisme $\mathcal{H}_\infty$ comme :

$$\left\| WT \right\|_\infty < 1 \tag{4.17}$$

où T est la fonction de transfert considérée et W est la fonction de pondération spécifiant la contrainte. En pratique, MATLAB permet de définir les pondérations de manière visuelle via le modèle appelé FRD (*Frequency Response Data*) à l'aide de la fonction `frd`. Par exemple, les perturbations en entrée du système étant un phénomène lié aux basses fréquences, on devrait pondérer la fonction de transfert correspondante T_1 par un filtre W_1 de type passe-bas. De la sorte, le modèle FRD créé par `frd`([0.1 0.1 10], [0 0.1 1]) peut être employé pour définir la contrainte de rejet de perturbations. Cela signifie que le gain statique de la fonction de transfert entre l'entrée perturbée et la sortie régulée est inférieur à 0.1 (c'est-à-dire $-20\,\mathrm{dB}$) dans la basse fréquence $[0, 0.1]\,\mathrm{rad/s}$. La fonction de pondération correspondante est donc récupérée par la fonction MATLAB `getWeight` :

$$W_1 = \frac{0.1}{(s + 0.1)^2}. \tag{4.18}$$

On cherche également à minimiser l'impact du bruit de mesure sur les sorties, principalement aux hautes fréquences. De façon similaire, on peut utiliser le modèle FRD défini par

`frd`([1 1 0.001 0.0001], [0 10 100 1000]) pour spécifier ce critère de performance. La pondération est donc un filtre passe-haut :

$$W_2 = \frac{1250000(s+10)^3}{(s+125000)(s+100)^2}. \tag{4.19}$$

Les autres contraintes telles que les objectifs de poursuite, les marges de stabilité ou l'effort de commande sont présentées dans le tableau 4.1.

Tableau 4.1 Cahier des charges

	Critère	**Exigence**
1	Suivi de trajectoire en x, y, z	Temps de réponse 3.25 s
		Erreur en régime permanent $< 0.01\%$
	Suivi de trajectoire en ψ	Temps de réponse 4 s
		Erreur en régime permanent $< 0.01\%$
2	Dépassement	2%
3	Pôles en boucle fermée	Amortissement > 0.6
		Pulsation naturelle > 1
4	Marges de stabilité	Marge de gain > 10 dB
		Marge de phase $> 45°$
5	Saturations d'actionneurs	`frd`([0.1 1 10], [0 1 10])
6	Rejet de perturbations	`frd`([0.1 0.1 10], [0 0.1 1])
7	Rejet de bruits de mesure	`frd`([1 1 10^{-3} 10^{-4}], [0 10 100 1000])

4.2.2.1 Considérations pratiques

La méthode de synthèse proposée n'offre aucune garantie quant à la convergence vers une solution stabilisant et satisfaisant les contraintes. Cette non garantie provient de la non convexité du problème de synthèse $\mathcal{H}_\infty$ structurée [9] et du choix *a priori* de l'architecture du contrôleur, soit la loi de commande (4.13) et la structure de séquencement (4.14). Nous portons donc notre attention sur les aspects déterminant les possibilités de succès ou d'échec de la synthèse du contrôleur :

1. Le problème d'optimisation de la synthèse $\mathcal{H}_\infty$ structurée étant non lisse et non convexe, une bonne initialisation de la méthode numérique est nécessaire. Pour cela, on tire profit des gains LQR permettant de stabiliser le système nominal. Le séquencement de gains est donc optimisé par `systune` autour de cette condition initiale pour satisfaire au mieux les contraintes.
2. Un choix judicieux de pondérations basé sur une bonne connaissance du système à contrôler est également indispensable. En effet, des contraintes trop sévères peuvent

causer un échec de l'algorithme `systune` tandis que des contraintes trop faibles ne permettent pas d'obtenir des résultats adéquats. En d'autres termes, les critères de performance listés dans le tableau 4.1 sont posées de façon à obtenir le meilleur compromis possible entre les spécifications usuellement requises par les drones, la convergence de l'algorithme `systune` et la performance expérimentale du système. Plus particulièrement, étant donné que l'on ne cherche pas un suivi de trajectoire agressif, il est possible de relaxer la contrainte de poursuite. Pour cela, il suffit de la limiter dans la plage des basses fréquences $[0, 0.1]$ rad/s. En bref, la transposition du cahier des charges en contraintes $\mathcal{H}_\infty$ est une phase primordiale et délicate lors de la synthèse.

3. La discrétisation du domaine incertain est aussi importante. En effet, les fautes critiques (e.g., $\gamma_i = 0.9$ ou $\gamma_i = 1$) peuvent causer une divergence de l'algorithme `systune` ou des gains trop grands qui peuvent être risqués expérimentalement ; ces gains trop forts peuvent dans le meilleur des cas causer de grands déplacements lors de l'apparition de la faute ou dans le pire des cas, un crash. À l'inverse, une plage de fautes trop faibles ne couvre pas efficacement le domaine incertain. En effectuant plusieurs fois la synthèse via la fonction `systune`, on choisit la plage $[0, 0.8]$ rendant possible la résolution numérique du problème FTC.

Les détails de la mise en œuvre de `systune` en ligne de commande MATLAB sont reportés dans l'annexe B.3.3.

4.2.3 Résultats de synthèse

L'algorithme d'optimisation `systune` est exécuté sur un ordinateur présentant un *quad-core* Intel i5-8250U cadencé à 1.6 GHz avec 16 GB RAM et MATLAB R2018a. Après 42 itérations en 4 min 40 s, la synthèse renvoie les paramètres suivants :

$$\begin{aligned}
[a_{0x}\ b_{0x}\ c_{0x}\ d_{0x}] &= [1.5550 \quad 1.1318 \quad 3.8617 \quad 0.6825] \\
[a_{1x}\ b_{1x}\ c_{1x}\ d_{1x}] &= [-0.1837 \ -0.0752 \quad 0.9880\ 0.6589] \\
[a_{2x}\ b_{2x}\ c_{2x}\ d_{2x}] &= [0.8541 \quad 0.6792 \quad 2.8027 \quad 0.5632] \\
[i_{0x}\ i_{1x}\ i_{2x}] &= [0.9816 \ -0.1667 \quad 0.5578].
\end{aligned} \tag{4.20}$$

La mesure de satisfaction de chaque contrainte est donnée dans le tableau 4.2. La valeur maximale des normes $\mathcal{H}_\infty$ finales est exactement 1.0, ce qui signifie que le cahier des charges est entièrement respecté pour $\gamma_i \in \{0, 0.1, \ldots, 0.8\}$, $i = 1, 3, 4, 6$.

La figure 4.3 montre l'évolution des gains du contrôleur en fonction des LAE γ_1 et γ_3 pour le premier rotor, soit la première ligne de $\mathbf{K}$ dans l'équation (4.15). On obtient donc une surface

Tableau 4.2 Auto-séquencement par LAE - Résultats de la synthèse $\mathcal{H}_\infty$ structurée

Contrainte	1	2	3	4	5	6	7
Norme $\mathcal{H}_\infty$ initiale (LQR)	1.07	0.93	16.32	∞	0.84	0.78	1.05
Norme $\mathcal{H}_\infty$ finale	0.99	0.93	1.00	0.96	0.29	0.25	1.00

des gains antisymétrique due aux différentes valeurs de $[a_{1x}\ b_{1x}\ c_{1x}\ d_{1x}]$ et $[a_{2x}\ b_{2x}\ c_{2x}\ d_{2x}]$.

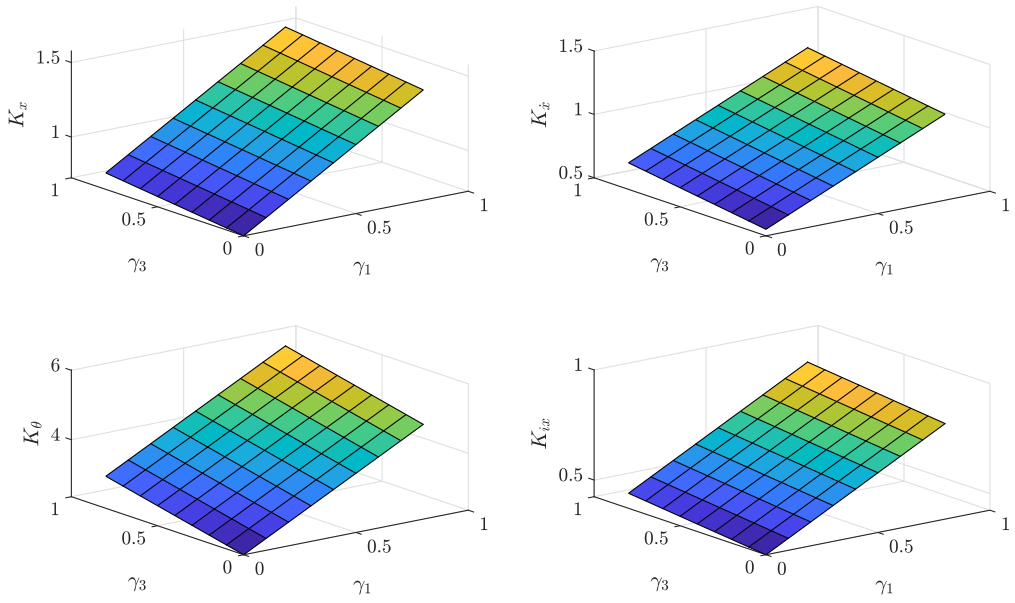
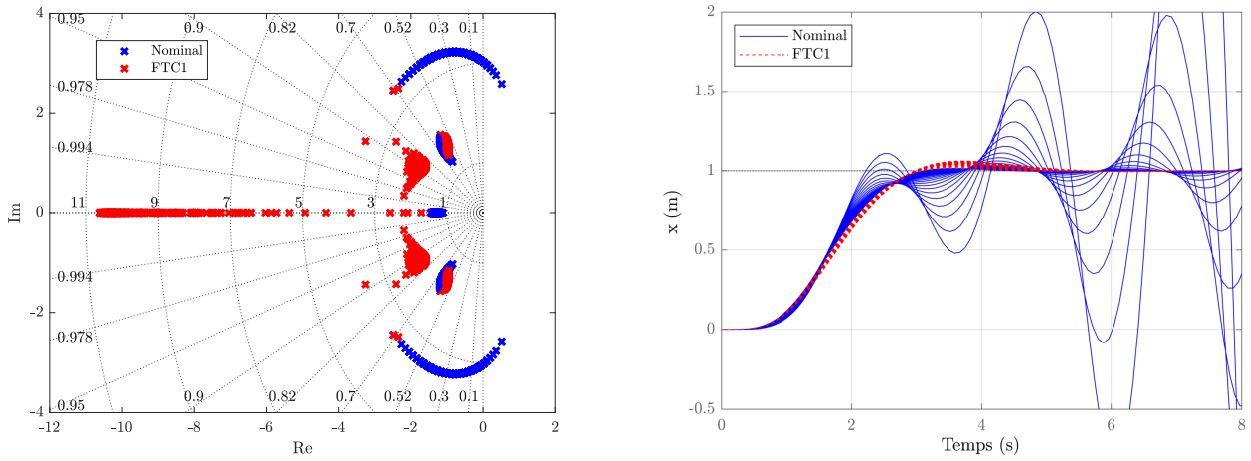


Figure 4.3 Auto-séquencement par LAE - Surface de gains pour le premier rotor ($\gamma_4 = \gamma_6 = 0$)

Si les valeurs de LAE (γ_i) sont bien estimées par le module FDD, les gains du contrôleur auto-séquencé peuvent être actualisés en temps réel pour maintenir un niveau de performance adéquat. En effet, pour 495 modèles considérés, les cinq pôles du système bouclé se concentrent autour de leurs valeurs nominales (Fig. 4.4a). Il en résulte des bonnes réponses à l'échelon unité tel qu'illustré à la figure 4.4b. On remarque que le contrôleur nominal ne permet pas de stabiliser le système dans tous les cas.

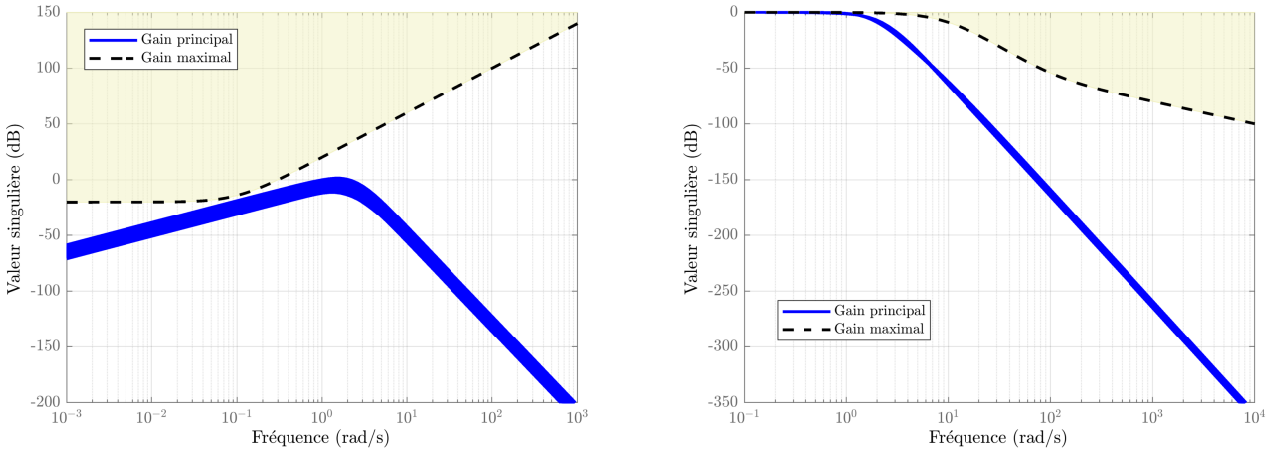
Les réponses fréquentielles du système bouclé avec le séquencement de gains sont tracées sur la figure 4.5. La région colorée limitée par la ligne pointillée représente la région de violation de l'objectif imposé. Comme on peut l'observer, les réponses fréquentielles des fonctions de sensibilité restent robustement en dehors de cette région. Cela signifie que les critères de performance sont parfaitement rencontrés sur l'ensemble des points de synthèse choisis.



(a) Confinement des pôles en boucle fermée

(b) Réponse à l'échelon unité

Figure 4.4 Auto-séquencement par LAE - Réponse du système en boucle fermée



(a) Contrainte de rejet de perturbations

(b) Contrainte de rejet de bruits de mesures

Figure 4.5 Auto-séquencement par LAE - Évaluation de contraintes fréquentielles

De manière à vérifier la stabilité et la performance du système bouclé hors points de synthèse (e.g., $\gamma_i > 0.8$), on calcule la valeur maximale des normes $\mathcal{H}_\infty$ des contraintes. La figure 4.6 trace cet indice en fonction des paramètres incertains γ_1 et γ_3 en supposant que $\gamma_4 = \gamma_6 = 0$. Il est clair que le contrôleur auto-séquencé est plus robuste aux fautes d'actionneurs que le contrôleur de base. On note également que l'indice de satisfaction des contraintes est 1.1 pour le cas $\gamma_1 = \gamma_3 = 1$. Les contraintes imposées sont ainsi majoritairement rencontrées.

Comme il l'a été mentionné auparavant, la présente méthode ne garantit pas *a priori* la stabilité du système bouclé entre deux points de synthèse. Une analyse de robustesse *a posteriori* reposant sur les applications gardiennes sera discutée dans le chapitre 6.

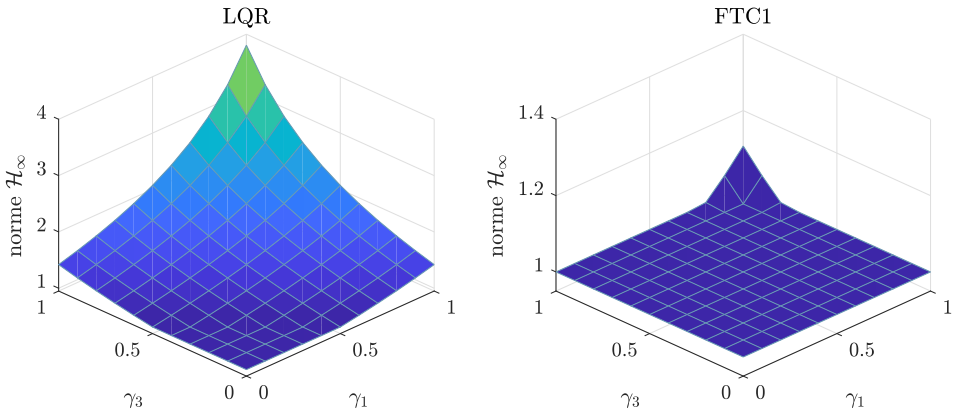


Figure 4.6 Comparaison de la norme $\mathcal{H}_\infty$ ($\gamma_4 = \gamma_6 = 0$)

4.2.4 Implémentation du contrôleur et technique d'ajustement

4.2.4.1 Implémentation du contrôleur auto-séquencé sur le système non linéaire

La synthèse du contrôleur auto-séquencé étant à présent complétée, il reste à l'implémenter sur le système non linéaire d'origine. Pour ce faire, on recourt à l'implémentation du contrôleur nominal (Sous-Section 3.2.2) où l'angle de lacet ψ est pris en compte. Cette implémentation présente néanmoins un problème qui réside dans la matrice d'allocation de contrôle $\mathbf{B}_{\text{CA}}$ (Eq. (3.23)). En effet, selon la loi $\Delta\mathbf{u} = \mathbf{B}_{\text{CA}}^{\dagger}\Delta\mathbf{v}$, la répartition de contrôle prend la forme :

$$\frac{d\sqrt{3}}{2}|\Delta T_i| = \frac{\Delta M}{4} \Leftrightarrow |\Delta T_i| = \frac{\Delta M}{2d\sqrt{3}},\ i = 1, 3, 4, 6.$$

Cela enlève l'avantage de la structure originale de séquencement de gains (Eqs. (4.15) et (4.16)). Pour pallier ce problème, l'implémentation du contrôleur auto-séquencé est divisée en deux parties. La première correspond aux gains du contrôleur en mode de fonctionnement

nominal, c'est-à-dire les coefficients $\mathbf{K}_0$ et $\mathbf{K}_{i_0}$ dans l'équation (4.14). Après la synthèse, ces valeurs sont refondues dans la structure (3.32) en utilisant la relation (3.20). De cette façon, on obtient :

$$[K_x \ K_{\dot{x}} \ K_\theta \ K_q \ K_{i,x}] = 2d\sqrt{3}[a_{0x} \ b_{0x} \ c_{0x} \ d_{0x} \ i_{0x}] \tag{4.21}$$

Cette partie est ensuite implémentée sur le système non linéaire comme dans la Sous-Section 3.2.2 en tenant compte de l'angle de cap ψ durant le vol.

La deuxième partie vise à tolérer les fautes via les coefficients $\mathbf{K}_j$ et $\mathbf{K}_{i_j}$, $j = 1, \ldots, 4$ dans l'équation (4.14). La loi séquencée s'écrit :

$$\mathbf{\Delta u}_x^{\text{gs}} = -\mathbf{K}^{\text{gs}}\mathbf{\Delta x}_x + \mathbf{K}_i^{\text{gs}} \int (\Delta x_r - \Delta x)\, \mathrm{d}t \tag{4.22}$$

où $\mathbf{\Delta u}_x^{\text{gs}} = [-\Delta T_{1,x}^{\text{gs}} \ \Delta T_{3,x}^{\text{gs}} \ \Delta T_{4,x}^{\text{gs}} \ -\Delta T_{6,x}^{\text{gs}}]^\top$, $\mathbf{K}^{\text{gs}} = \mathbf{K} - \mathbf{K}_0$ et $\mathbf{K}_i^{\text{gs}} = \mathbf{K}_i - \mathbf{K}_{i_0}$.

Dès lors, la variation de la force ΔT_i^{gs} imposée au i^{e} rotor est obtenue en combinant les sorties de quatre sous-contrôleurs (Fig. 4.2) comme étant :

$$\Delta T_i^{\text{gs}} = \Delta T_{i,x}^{\text{gs}} + \Delta T_{i,y}^{\text{gs}} + \Delta T_{i,z}^{\text{gs}} + \Delta T_{i,\psi}^{\text{gs}}. \tag{4.23}$$

Finalement, la poussée totale demandée au i^{e} rotor se met sous la forme :

$$T_i = T_i^n + \Delta T_i^{\text{gs}} \tag{4.24}$$

où T_i^n est la poussée nominale générée par la première partie.

4.2.4.2 Technique d'ajustement

Comme mentionné auparavant, la loi de commande par retour d'état avec composante intégrale (4.13) et la structure de séquencement (4.14) permettent de synthétiser avec succès le contrôleur tolérant aux fautes en simulation. Il existe de plus un processus itératif d'ajustement des gains du contrôleur pour obtenir de meilleurs résultats lors de vols expérimentaux. En effet, l'environnement en simulation est toujours idéalisé par rapport aux tests expérimentaux. La procédure de réglage est la suivante :

1. Établissement et/ou ajustement des contraintes ;
2. Synthèse du contrôleur robuste et auto-séquencé à l'aide de `systune` ;
3. Simulation dans MATLAB et ROS/Gazebo ;
4. Implémentation du contrôleur sur l'hexacoptère Asctec Firefly et tests en laboratoire.

Si le résultat d'une étape n'est pas adéquat, un retour en arrière est effectué pour l'améliorer.

4.3 Auto-séquencement de gains par LVE

En utilisant la LAE pour ajuster les gains du contrôleur, la méthode précédente présente une capacité de tolérance aux fautes efficace et robuste. Cependant, la complexité du problème d'optimisation de la synthèse est fortement liée à la configuration du multicoptère. En effet, la méthode peut être coûteuse en temps de calcul pour les drones possédant plusieurs rotors à cause du grand nombre de modèles à considérer (Eq. (4.12)). De même, la structure de séquencement de gains dépend de la géométrie du multicoptère (Eq. (4.15)). La simplification proposée tirant profit de la configuration spéciale d'un hexacoptère ne s'applique pas en général à d'autres types de véhicules. De plus, la mise en œuvre d'un tel système FTC sur les petits drones pourrait être un autre défi.

De manière à éviter ces obstacles, on propose la deuxième méthode FTC basée sur la LVE. L'architecture du système bouclé utilisée à des fins de synthèse est montrée à la figure 4.7. Le vecteur d'entrée étant la commande virtuelle $\mathbf{\Delta v} = [\Delta T\ \Delta L\ \Delta M\ \Delta N]^\top$, la ré-allocation de contrôle est donc indispensable. Comme précédemment, seule la conception du contrôleur longitudinal concernant le vecteur d'état $\mathbf{\Delta x}_x = [\Delta x\ \Delta \dot{x}\ \Delta \theta\ \Delta q]^\top$ est présentée ici.

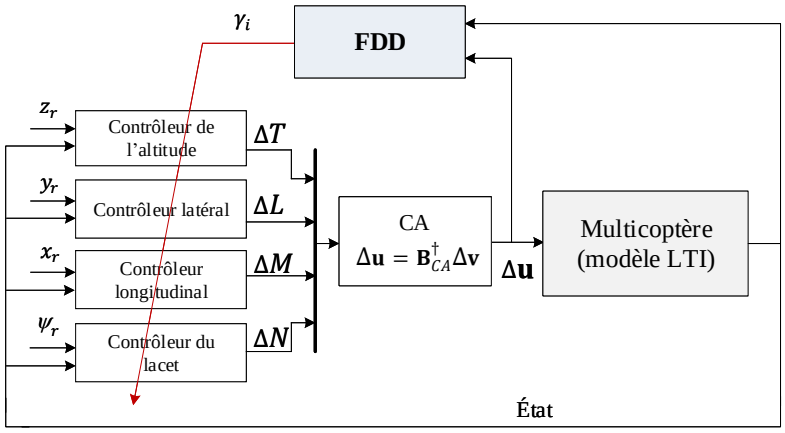


Figure 4.7 Architecture de FTC basée sur LVE utilisée lors de la synthèse

Tout d'abord, il est important d'être clair quant à notre définition de LVE.

Définition 4.3.1. *L'efficacité de la commande virtuelle est la possibilité de générer la poussée totale des rotors, les moments selon les axes* $\mathbf{x}_b$ *et* $\mathbf{y}_b$ *dus aux forces de poussée et le moment induit selon l'axe* $\mathbf{z}_b$ *du véhicule.*

À partir de cette définition, la LVE d'un hexacoptère est liée à la LAE par l'équation suivante :

$$\begin{bmatrix} \gamma_T \\ \gamma_L \\ \gamma_M \\ \gamma_N \end{bmatrix} \triangleq \begin{bmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/8 & 1/4 & 1/8 & 1/8 & 1/4 & 1/8 \\ 1/4 & 0 & 1/4 & 1/4 & 0 & 1/4 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_6 \end{bmatrix} \tag{4.25}$$

où $\gamma_{T/L/M/N} \in [0, 1]$ représentent respectivement la perte d'efficacité de la poussée totale et des moments de roulis, de tangage et de lacet.

Le modèle d'état linéarisé de la dynamique longitudinale est donné par :

$$\boldsymbol{\Delta}\dot{\mathbf{x}}_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\Delta}\mathbf{x}_x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1-\gamma_M}{I_{yy}} \end{bmatrix} \Delta M \tag{4.26}$$

Comme précédemment, on choisit la loi de commande par retour d'état avec action intégrale :

$$\Delta M = -\mathbf{K}_x \boldsymbol{\Delta}\mathbf{x}_x + K_i \int (\Delta x_r - \Delta x)\, \mathrm{d}t \tag{4.27}$$

où $\mathbf{K}_x \in \mathbb{R}^{1\times 4}$ et $K_i \in \mathbb{R}$. Tout d'abord, on définit *a priori* le séquencement de gains sous la forme de fonctions quadratiques de γ_M. L'évolution de gain obtenue est finalement presque linéaire, et par simplicité, on choisit une nouvelle fois un séquencement de gains affine :

$$\mathbf{K}_x = \mathbf{K}_{x_0} + \mathbf{K}_{x_1}\gamma_M \;\text{ et }\; K_{ix} = K_{ix_0} + K_{ix_1}\gamma_M. \tag{4.28}$$

Il y a donc seulement 10 paramètres ajustables à synthétiser. Cette formule de séquencement ne dépend pas de la configuration du multicoptère et l'on peut également fixer les coefficients $\mathbf{K}_{x_0}$ et K_{ix_0} aux valeurs des gains nominaux pour simplifier la synthèse.

En considérant 17 modèles défectueux ($\gamma_M \in \{0, 0.05, \ldots, 0.8\}$), on applique la même procédure de design que celle présentée à la Section 4.2. Puisque les nouveaux points d'intérêt correspondent à l'entrée de commande virtuelle, les critères de performances formulés dans le tableau 4.1 sont légèrement modifiés. Par exemple, la contrainte de saturation d'actionneurs est maintenant caractérisée par le modèle `frd`([0.001 0.01 0.1], [0 0.1 1]). La synthèse est une nouvelle fois prise en charge par la fonction MATLAB `systune`. Après 37 itérations et

15 secondes d'exécution, le séquencement de gains obtenu est le suivant :

$$\begin{aligned} \mathbf{K}_x &= [1.34 \;\; 0.80 \;\; 2.55 \; 0.45] + [1.20 \;\; 1.35 \;\; 6.58 \; 1.70]\gamma_M \\ K_{ix} &= 1.09 + 0.16\gamma_M \end{aligned} \tag{4.29}$$

Le gain d'optimisation de la synthèse est 1.20 (Tab. 4.3), ce qui indique que le cahier des charges est majoritairement satisfait. Ainsi, ce choix de séquencement a légèrement dégradé les performances du contrôleur par rapport à celui de la méthode précédente. Cependant, il présente l'avantage d'une plus grande simplicité et d'une réduction du temps de calcul.

Tableau 4.3 Auto-séquencement par LVE - Résultats de la synthèse $\mathcal{H}_\infty$ structurée

Contrainte	1	2	3	4	5	6	7
Norme $\mathcal{H}_\infty$ finale	1.17	0.93	1.20	0.52	0.46	1.00	0.86

La figure 4.8 illustre l'évolution de gains du contrôleur séquencé en fonction de γ_M. Grâce à cet ajustement de gains, les pôles en boucle fermée demeurent dans la zone d'amortissement désirée (Fig. 4.9a). Par conséquent, la stabilité et les performances du système bouclé sont garanties à travers l'ensemble des modèles considérés (Fig. 4.9b).

Pour compléter le design, on calcule la norme $\mathcal{H}_\infty$ finale du système bouclé en présence des fautes ($0 \leq \gamma_M \leq 1$). Le résultat obtenu est résumé dans le tableau 4.4. On note en particulier que le contrôleur nominal ne peut pas stabiliser le système pour une valeur de $\gamma_M > 0.6$, alors que le contrôleur FTC le permet jusqu'à $\gamma_M = 0.9$. On le retrouvera ultérieurement à la section 6.2.1.

Tableau 4.4 Vérification de la synthèse $\mathcal{H}_\infty$ structurée

γ_M	0	0.1	0.2	0.4	0.6	0.8	0.9	1
Nominal	1.19	1.43	1.80	3.71	∞	∞	∞	∞
FTC	1.19	1.16	1.14	1.18	1.11	1.19	3.69	∞

Finalement, le contrôleur est implémenté sur le système non linéaire en deux boucles de contrôle séparées comme dans le cas du contrôleur nominal (cf. Sous-Section 3.2.2).

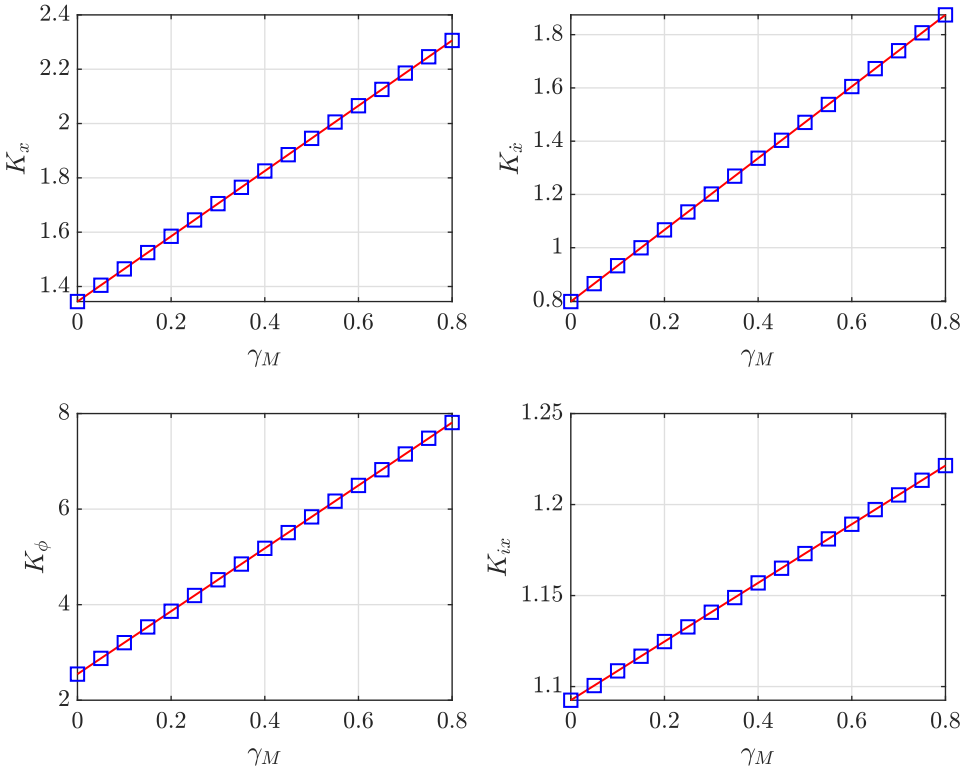
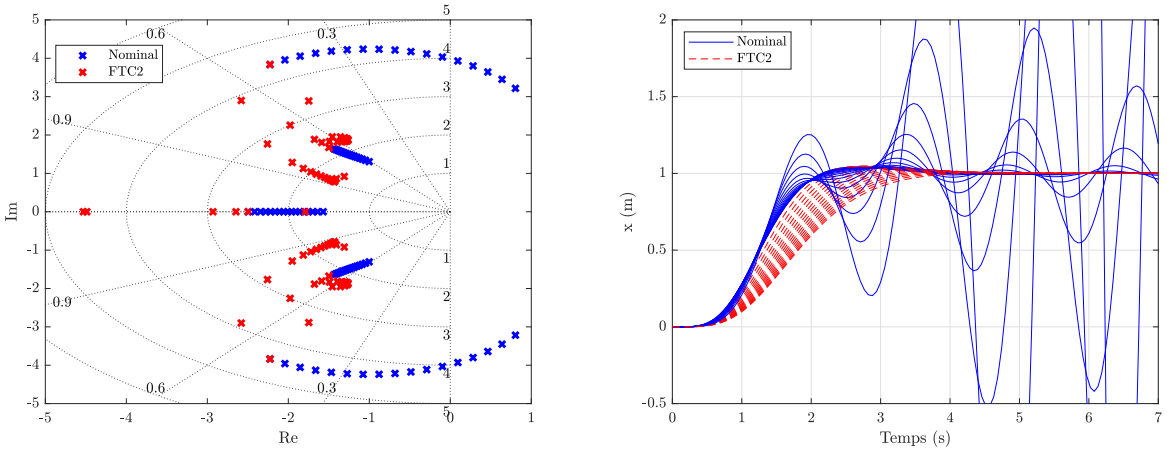


Figure 4.8 Auto-séquencement par LVE - Évolution de gains du contrôleur

.



(a) Confinement des pôles en boucle fermée

(b) Réponse à l'échelon unité

Figure 4.9 Auto-séquencement par LVE - Réponses du système en boucle fermée

4.4 Auto-séquencement de CA

Motivé par l'auto-séquencement de gains et la synthèse $\mathcal{H}_\infty$ structurée, on propose une extension de cette approche au problème d'allocation de contrôle. Plus précisément, on s'intéresse à l'architecture de contrôle illustrée à la figure 4.10. Dans ce cas, la tolérance aux fautes est prise en charge par la tâche CA ; le contrôleur n'a plus besoin d'être reconfiguré. On reprend ici le contrôleur nominal de type LQR (voir Sous-Section 3.2.1).

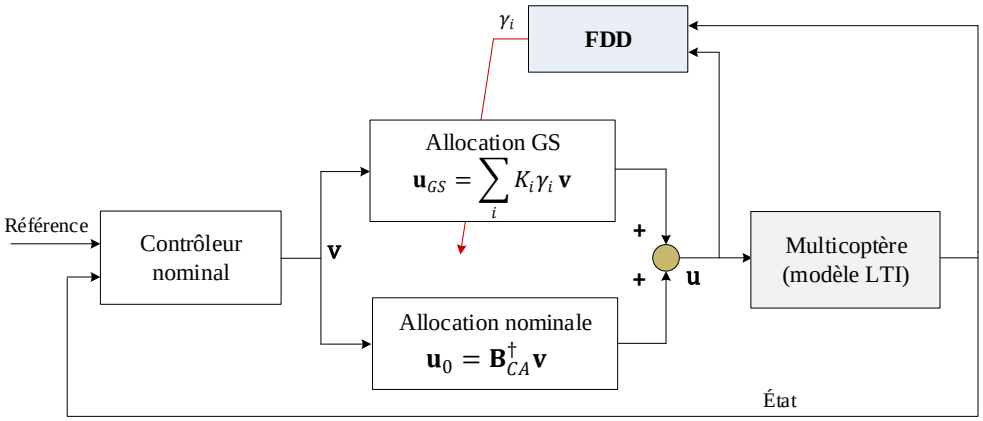


Figure 4.10 Architecture de FTC basée sur l'auto-séquencement de CA

Formulation du problème d'allocation de contrôle On considère le problème de CA pour les systèmes sur-actionnés ayant plus d'actionneurs que de degrés de contrôle ($m > 4$ pour les multicoptères classiques) :

$$\begin{aligned} \mathbf{B}_{\mathrm{CA}}\mathbf{\Lambda}\mathbf{u} &= \mathbf{v} \\ \mathbf{\Lambda} \triangleq \mathbf{I} - \mathbf{\Gamma} &= \mathrm{diag}\,(\lambda_i) \;\; ; \;\; \underline{\mathbf{u}} \leq \mathbf{u} \leq \overline{\mathbf{u}} \end{aligned} \tag{4.30}$$

où $\mathbf{\Lambda} \in \mathbb{R}^{m\times m}$ est la matrice d'efficacité de contrôle, $\mathbf{\Gamma} \in \mathbb{R}^{m\times m}$ est définie dans l'équation (4.2), $\underline{\mathbf{u}}$ et $\overline{\mathbf{u}}$ représentent les limitations de position et de taux de variation des actionneurs [163]. Ainsi, $\lambda_i = 1$ représente le point de fonctionnement régulier et $\lambda_i = 0$ indique que le i^{e} actionneur est complètement défectueux. La matrice d'allocation de contrôle $\mathbf{B}_{\mathrm{CA}} \in \mathbb{R}^{4\times m}$ est de ligne rang plein, car le système est sur-actionné.

L'objectif de CA est de répartir la commande virtuelle $\mathbf{v}$ sur l'ensemble des actionneurs, c'est-à-dire de trouver une solution $\mathbf{u}$ satisfaisant l'équation (4.30). De manière à minimiser l'effort de commande, i.e., la norme $\|\mathbf{u}\|_2$, la matrice pseudo-inverse de Moore-Penrose est

souvent utilisée dans la littérature :

$$\mathbf{u} = \mathbf{B}_{\mathrm{CA}}^{\dagger}\mathbf{v} = \mathbf{B}_{\mathrm{CA}}^{\top}\left(\mathbf{B}_{\mathrm{CA}}\mathbf{B}_{\mathrm{CA}}^{\top}\right)^{-1}\mathbf{v}. \tag{4.31}$$

Une fois la matrice d'efficacité de contrôle $\mathbf{\Lambda}$ estimée, il est possible de prendre en compte directement cette matrice comme dans [153] :

$$\mathbf{u} = \mathbf{\Lambda}^{-1}\mathbf{B}_{\mathrm{CA}}^{\dagger}\mathbf{v} \tag{4.32}$$

Cependant, la solution dépend fortement de la valeur estimée de $\mathbf{\Lambda}$ par le module FDD. De plus, lorsque $\lambda_i \to 0$, la commande correspondante approche l'infini. Pour éviter ce problème, une solution possible est la ré-allocation par optimisation (voir Sous-Section 2.1.3).[2] Une autre solution est l'allocation de contrôle adaptative qui sera abordée dans le Chapitre 5.

Auto-séquencement de l'allocation de contrôle On développe à présent la méthode d'allocation de contrôle auto-séquencée basée sur la synthèse $\mathcal{H}_\infty$ structurée. Pour cela, la loi CA classique est augmentée par :

$$\mathbf{u} = \mathbf{K}\mathbf{v} = \left(\mathbf{K}_0 + \sum_i \mathbf{K}_i\gamma_i\right)\mathbf{v} \tag{4.33}$$

où les coefficients des matrices $\mathbf{K}_0$ et $\mathbf{K}_i$ sont les paramètres ajustables à synthétiser. Dans le cas d'un hexacoptère, $\mathbf{K}_0 \in \mathbb{R}^{6\times4}$ et $\mathbf{K}_i \in \mathbb{R}^{6\times4}$. Par simplicité, on opte ici pour un séquencement sous forme de fonction affine des LAE γ_i. Ainsi, le nombre de paramètres est $4m(m+1) = 168$ avec $m = 6$, le nombre de rotors de l'hexacoptère. Bien évidemment, un auto-séquencement plus complexe utilisant des fonctions bilinéaires ou quadratiques pourrait être choisi pour satisfaire au mieux les spécifications imposées.

On cherche maintenant à réduire le nombre des paramètres ajustables dans le but de faciliter le processus de réglage. Pour ce faire, la loi CA (4.33) est modifiée comme suit :

$$\mathbf{u} = \mathbf{K}\mathbf{v} = \left(\mathbf{B}_{\mathrm{CA}}^{\dagger} + \sum_i \mathbf{K}_i\gamma_i\right)\mathbf{v} = \mathbf{u}_0 + \mathbf{u}_{\mathrm{gs}}. \tag{4.34}$$

Ainsi, l'impact des fautes est atténué par la composante adaptative $\mathbf{u}_{\mathrm{gs}}$ tout en conservant une ré-allocation de contrôle nominale $\mathbf{u}_0$. Utiliser les LAE γ_i au lieu de λ_i assure que le comportement du système bouclé dans la situation normale ($\gamma_i = 0$) n'est pas changé par la

2. La matrice de pondération (voir Eq. (2.12)) peut être simplement choisie comme étant $\mathbf{W} = \mathbf{\Lambda}$. De cette façon, une grande pondération sera appliquée sur l'actionneur défectueux pour minimiser son utilisation.

CA. De cette façon, l'auto-séquencement de CA peut être couplé à n'importe quel contrôleur de base pour augmenter la capacité de tolérance aux fautes. On peut également simplifier la synthèse de CA en tirant profit de la configuration particulière du multicoptère. Par exemple, le deuxième rotor d'un hexacoptère (Fig. 3.1) ne produit aucun moment selon l'axe $\mathbf{y}_b$. Il en résulte des composantes nulles imposées dans les matrices $\mathbf{K}_i$. En bref, le nombre des paramètres ajustables est réduit de 168 à 132 dans le cas d'un hexacoptère.

En considérant trois valeurs de $\gamma_i \in \{0, 0.35, 0.7\}$, soit 28 modèles défectueux selon l'équation (4.12), on applique la même procédure de design que celle présentée à la section 4.2. Étant donné que le modèle linéaire complet du système bouclé est considéré durant la phase de synthèse, on peut imposer une contrainte supplémentaire sur le découplage [3] en plus des contraintes listées dans le tableau 4.1. Cette contrainte est spécifiée par le modèle `frd`[0.1 0.1 1], [0 0.1 1]. Une nouvelle fois, on utilise la fonction MATLAB `systune` pour ajuster les coefficients de la matrice CA. Après 143 itérations en 147.8 secondes, on obtient un gain d'optimisation de la synthèse de 1.29, ce qui indique que les contraintes imposées sont en grande partie rencontrées. Plus précisément, seule la contrainte sur les pôles en boucle fermée est violée. Cela découle du fait que la CA n'est pas une méthode efficace pour replacer les pôles du système. Néanmoins, on peut observer une amélioration assez importante sur la figure 4.11 au niveau de la position des pôles en boucle fermée. En effet, l'auto-séquencement de CA permet au moins de stabiliser le système en présence des différentes amplitudes de fautes. Les réponses temporelles du système sont illustrées à la figure 4.12. On mentionne en particulier que le découplage entre les différentes entrées de référence et les sorties régulées est bien amélioré.

Pour conclure, cette méthode présente l'avantage de ne pas nécessiter de résoudre un problème d'optimisation durant le vol. La loi CA séquencée étant explicite, la reconfiguration est rapide en temps réel. De plus, elle ne se limite pas aux multicoptères, mais peut être appliquée aux autres systèmes sur-actionnés pour tolérer les fautes d'actionneurs, de capteurs, etc.

3. L'influence d'une entrée de référence sur les autres sorties régulées.

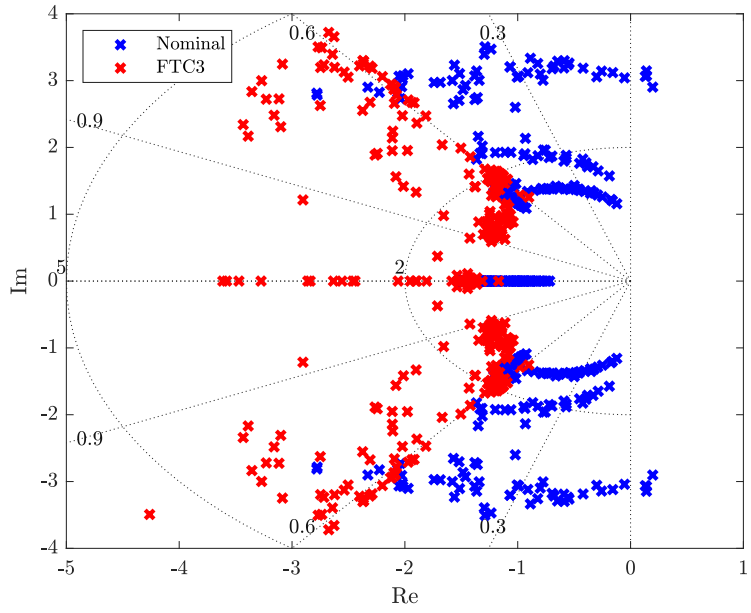


Figure 4.11 Auto-séquencement de CA - Confinement des pôles en boucle fermée

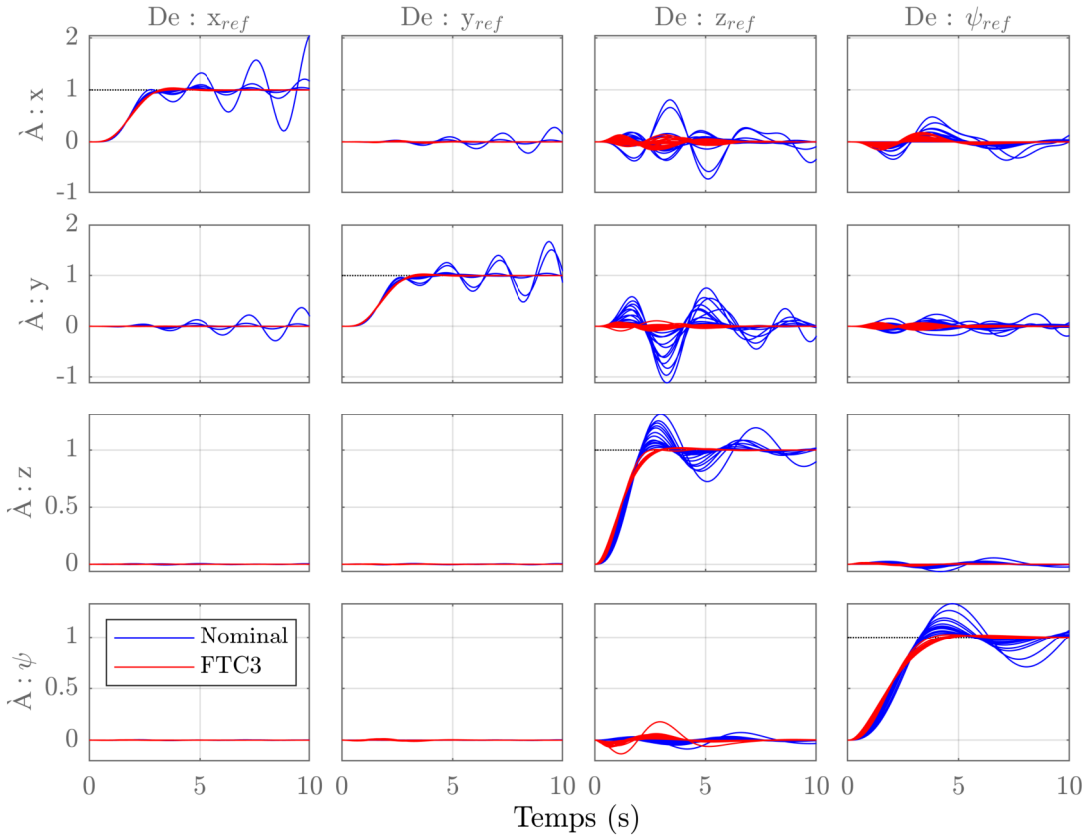


Figure 4.12 Auto-séquencement de CA - Réponses du système en boucle fermée

4.5 Simulation et tests expérimentaux

Différents essais en vol stationnaire et en suivi de trajectoire ont été effectués pour évaluer les performances du système bouclé en présence de fautes ; le tableau 4.5 présente les trois cas de niveaux de fautes d'actionneurs considérés dans ce travail.

Tableau 4.5 Scénarios des tests expérimentaux

Cas	γ_1	γ_2	γ_3	γ_T	γ_L	γ_M	γ_N
1	20%	90%	0	18.3%	25%	5%	18.3%
2	70%	50%	0	20%	21.25%	17.5%	20%
3	50%	40%	30%	20%	20%	20%	20%

Les LAE sont injectées de manière artificielle en introduisant un poids sur les commandes envoyées aux rotors. Par exemple, dans le cas 1, la multiplication de la commande RPM (*Rounds Per Minute*) du 2^{e} rotor par 0.315 durant le vol implique une dégradation de la vitesse de rotation de 68.5% ou une perte de la poussée de 90%, soit $\gamma_2 = 90\%$.

Les trois approches d'auto-séquencement ont été testées en simulation et expérimentalement sur l'hexacoptère Firefly. Du fait de contraintes de temps et de disponibilité du matériel, elles n'ont cependant pas pu être toutes testées pour les situations présentées ci-après. De plus, on ne peut pas vérifier expérimentalement une panne totale de rotor en raison de la configuration imposée NPNPNP du Firefly,[4] de l'altitude limitée de l'espace de vol intérieur et de la sécurité des essais en vol à basse altitude. On peut néanmoins considérer que le premier cas avec deux LAE de 20% et de 90% (Tab. 4.5) se rapproche d'une panne d'un rotor. Enfin, il sera possible de tester deux pannes successives de rotors pour un hexacoptère NNPPNP dans l'environnement de simulation ROS/Gazebo.

Les résultats présentés ci-après ont pour la plupart été publiés dans les articles [207, 209] et sont visualisables dans la vidéo des essais expérimentaux, disponible à l'adresse `https://youtu.be/8BS9n7uLMtI`. Des résultats préliminaires pour la première approche d'auto-séquencement par LAE ont été obtenus sur un quadricoptère et ont été présentés dans les articles de conférence [205, 206] ; ils ne sont pas reportés ici par souci de concision.

4.5.1 Amélioration de la stabilité

Pour le premier test expérimental, deux LAE de 20% et 90% sont respectivement rendues dans les deux premiers rotors à la fois (cas 1) . Sur la base de la figure 4.13, on trouve que

4. En effet, l'hexacoptère NPNPNP n'est pas commandable dès qu'un seul rotor tombe en panne [145].

les valeurs de LAE sont bien calculées en comparant les vitesses angulaires désirées et les vitesses réelles des rotors fournies par les encodeurs. En raison des bruits des capteurs, de légers sauts dans les données sont également observés.

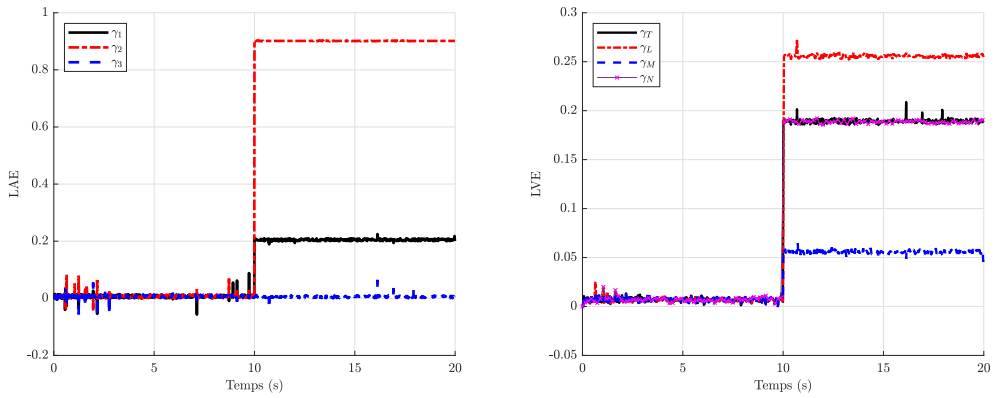
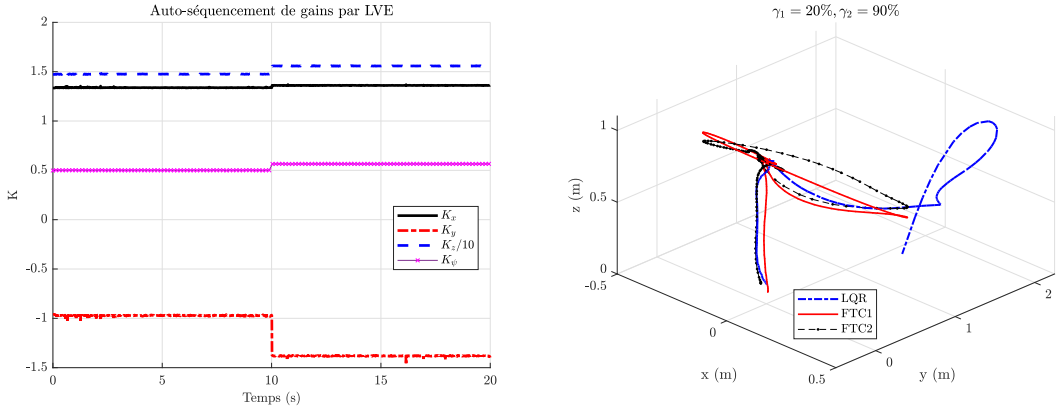


Figure 4.13 Cas 1 - LAE et LVE

L'évolution temporelle des gains du contrôleur séquencé par rapport à LVE est illustrée à la figure 4.14a. On peut observer que la stabilité du multicoptère est garantie par les contrôleurs auto-séquencés par LAE (FTC1) et par LVE (FTC2). En effet, le drone revient rapidement à sa position en vol stationnaire après l'apparition soudaine des fautes (Fig. 4.14b). L'instabilité provient de la perte d'efficacité du moment de roulis ($\gamma_L = 25\%$). Dans ce cas, un changement significatif des gains du contrôleur latéral auto-séquencé, (gain K_y dans Fig. 4.14a), permet d'assurer la stabilité en générant un moment approprié agissant sur le drone. Quant au contrôleur LQR, il ne parvient pas à stabiliser le système après les fautes. Le niveau de fautes le plus haut qu'il peut compenser expérimentalement est $\gamma_1 = 20\%$ et $\gamma_2 = 65\%$.



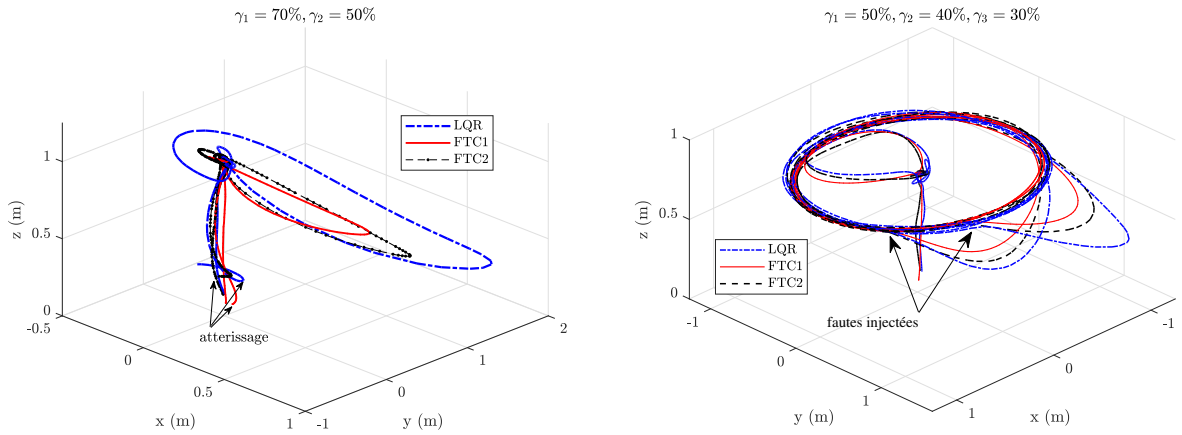
(a) Évolution de gains

(b) Position du drone

Figure 4.14 Cas 1 - Amélioration de la stabilité pour $\gamma_1 = 20\%$ et $\gamma_2 = 90\%$

4.5.2 Évaluation des performances

Vol stationnaire Pour analyser les performances du système bouclé en vol stationnaire, on considère les pertes d'efficacité $\gamma_1 = 70\%$ et $\gamma_2 = 50\%$ (cas 2). Tel que montré sur la figure 4.15a, l'auto-séquencement de gains par LAE (FTC1) réagit plus rapidement à l'apparition des fautes que le contrôleur nominal LQR et le contrôleur auto-séquencé par LVE (FTC2), surtout dans le plan $\mathbf{x}_i$-$\mathbf{y}_i$.



(a) Vol stationnaire

(b) Vol en suivi de trajectoire

Figure 4.15 Cas 2 et 3 - Amélioration de performances

Les angles d'Euler donnés à la figure 4.16 permettent de mieux visualiser l'attitude du drone. Dans ce cas, la dynamique latérale en y et ϕ est plus affectée par les fautes que la dynamique longitudinale en x et θ ($\gamma_L = 21.25\%$ par rapport à $\gamma_M = 17.5\%$). Pour le contrôleur nominal seul, l'angle de roulis atteint alors environ 35° et le drone évite l'écrasement de justesse.[5] Ainsi, les contrôleurs tolérants aux fautes sont largement plus performants en maintien de position. On peut également observer que la précision d'atterrissage après les fautes est aussi améliorée avec les lois d'auto-séquencement de gains proposées (Fig. 4.15a).

Enfin, il est intéressant de constater que les vitesses de rotation commandées aux rotors par le LQR et par le contrôleur auto-séquencé par LVE (FTC2) ne sont pas très différentes (Fig. 4.17). De plus, ces commandes avant les fautes ne sont pas strictement identiques pour les 6 rotors (contrairement à ce qui est attendu), indiquant que le drone n'est pas parfaitement symétrique.

5. À 1:01 dans la vidéo disponible à l'adresse `https://youtu.be/8BS9n7uLMtI?t=61`.

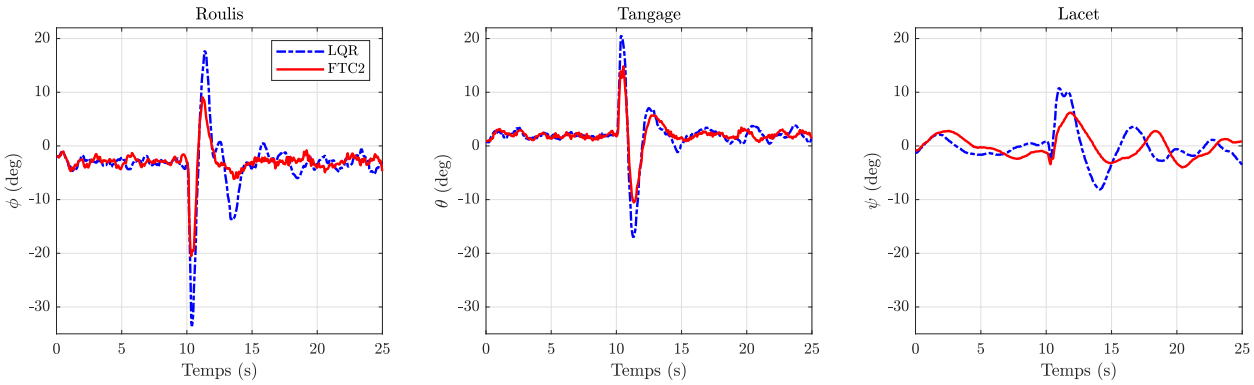


Figure 4.16 Cas 2 - Attitude du drone

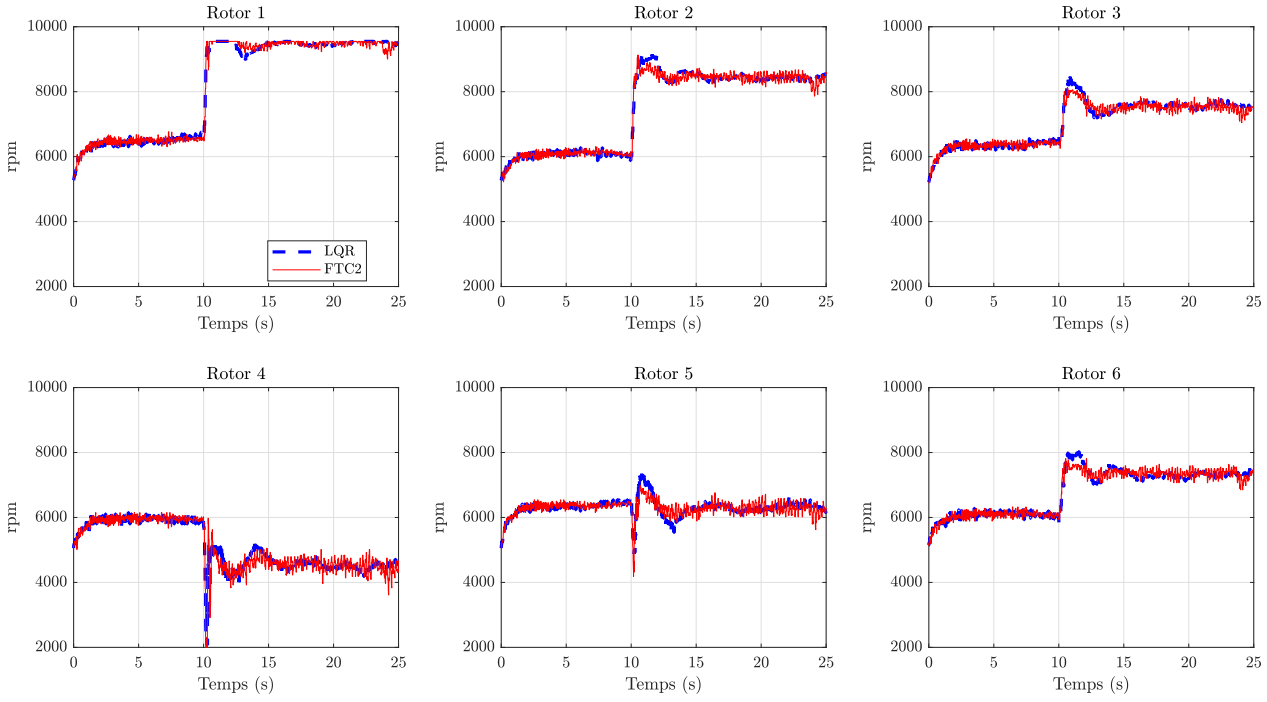


Figure 4.17 Cas 2 - Vitesse de rotation des rotors

Vol en suivi de trajectoire Le suivi de trajectoire à une altitude de 0.75 m et un lacet nul est testé pour valider les contrôleurs dans un scénario plus réaliste. Puisque l'on n'a pas intégré explicitement des algorithmes supplémentaires de suivi de trajectoire, la référence consiste ici en un cercle de rayon 1 m et une vitesse constante de 1 m/s. Plus précisément, la trajectoire désirée prend la forme :

$$[x_r \quad y_r \quad z_r \quad \psi_r]^\top = [\cos(t) \quad \sin(t) \quad 1 \quad 0]^\top \tag{4.35}$$

Une fois que l'hexacoptère a bien suivi cette trajectoire, il est successivement soumis à trois LAE : $\gamma_1 = 50\%$ à $t = 27\,\text{s}$, $\gamma_2 = 40\%$ et $\gamma_3 = 30\%$ à $t = 34\,\text{s}$ (cas 3). La performance du système bouclé est évaluée à la figure 4.15b. Une nouvelle fois, on observe que les méthodes d'auto-séquencement (FTC1 et FTC2) permettent d'obtenir des résultats satisfaisants en suivi de trajectoire et en tolérance aux fautes.[6]

4.5.3 Robustesse à l'estimation incorrecte de fautes

Le deuxième cas de test est reproduit pour vérifier le fonctionnement du système FTC bouclé lorsque les fautes ne sont pas bien estimées (cf. Sous-Section 4.1.3). Comme indiqué à la figure 4.18, les valeurs de LAE et de LVE fournies par le module FDD varient de manière significative autour de leurs valeurs exactes à cause des bruits de mesure. Il en résulte les mêmes effets sur les gains du contrôleur séquencé (Fig. 4.19a). En conséquence, la figure 4.19b montre une légère dégradation des performances par rapport aux résultats précédents (Fig. 4.15a). Comparativement au contrôleur nominal, les contrôleurs séquencés demeurent encore plus performants. Cela vérifie expérimentalement la robustesse du système FTC lors d'une mauvaise estimation de fautes.

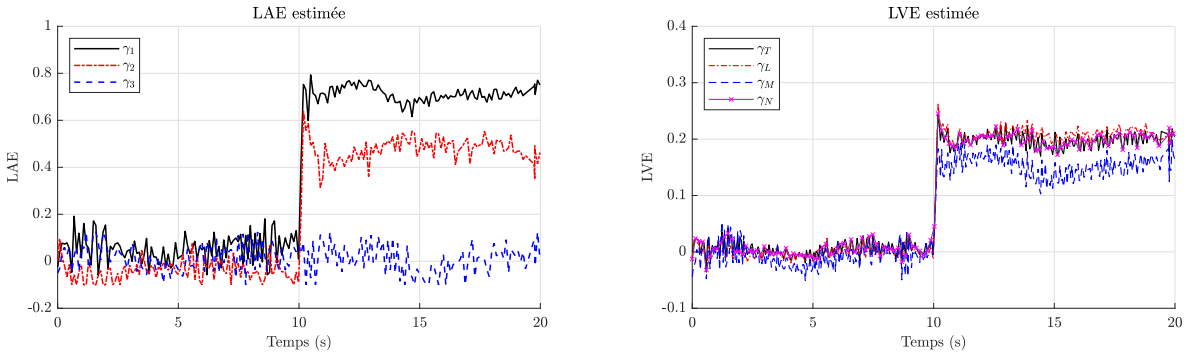
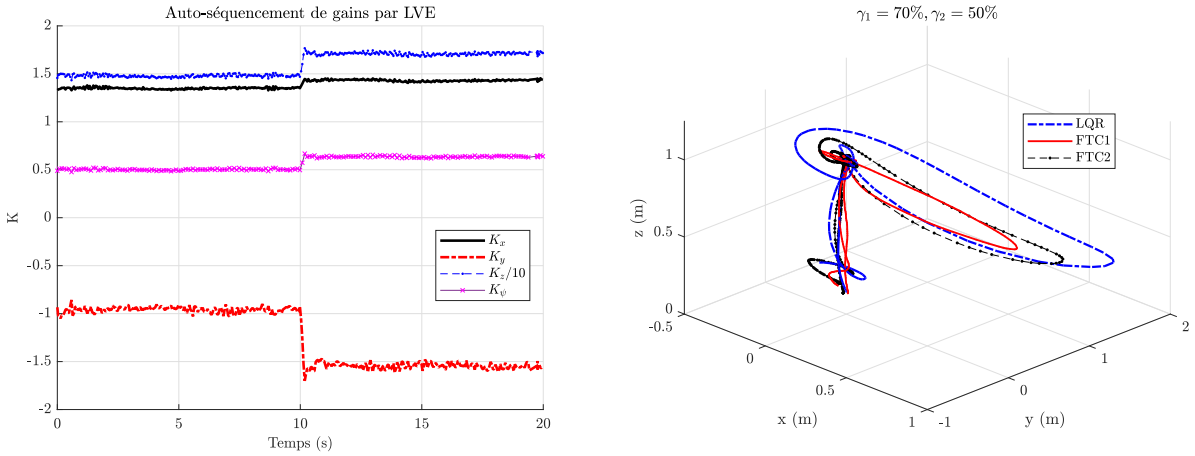


Figure 4.18 Cas 2 - LAE et LVE estimées

6. À 1:35 dans la vidéo disponible à l'adresse `https://youtu.be/8BS9n7uLMtI?t=95`.



(a) Évolution de gains

(b) Position du drone

Figure 4.19 Cas 2 - Robustesse par rapport à l'estimation incorrecte de fautes

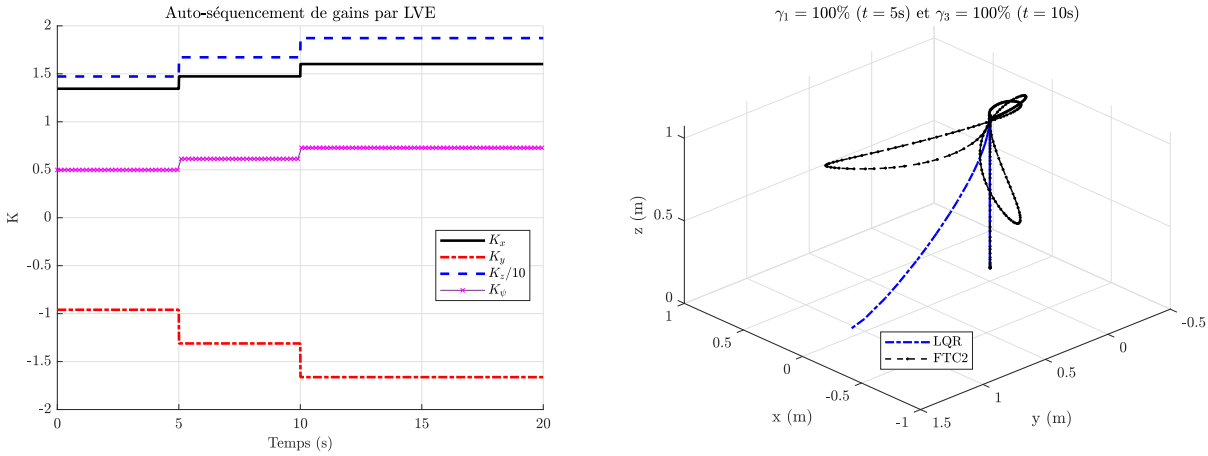
4.5.4 Panne de rotors

Comme nous ne pouvons modifier la configuration de notre hexacoptère Firefly, nous utilisons RotorS [7] pour simuler un hexacoptères NNPPNP sous ROS/Gazebo. Le but est de vérifier le comportement du système en présence d'une perte d'efficacité de 100% dans le premier rotor à $t = 5\,\mathrm{s}$ et dans le troisième rotor à $t = 10\,\mathrm{s}$. Notons que la configuration NNPPNP permet de maintenir la commandabilité d'un hexacoptère jusqu'à deux pannes de rotors [145]. On implémente ici le contrôleur auto-séquencé par LVE (FTC2) du fait de la simplicité de sa mise en œuvre. La loi de commande n'est pas modifiée et seule la dernière ligne de la matrice d'allocation de contrôle $\mathbf{B}_{\mathrm{CA}}$ doit être changée pour tenir compte de la configuration NNPPNP, soit :

$$\mathbf{B}_{\mathrm{CA}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{d}{2} & d & \frac{d}{2} & -\frac{d}{2} & -d & -\frac{d}{2} \\ -\frac{d\sqrt{3}}{2} & 0 & \frac{d\sqrt{3}}{2} & \frac{d\sqrt{3}}{2} & 0 & -\frac{d\sqrt{3}}{2} \\ -c & -c & c & c & -c & c \end{bmatrix}. \tag{4.36}$$

Le profil de gains et la réponse du système bouclé sont illustrés à la figure 4.20. On remarque d'abord que le LQR avec composante intégrale ne peut pas stabiliser le système même s'il y a une seule panne de rotor. À l'opposé, le contrôleur séquencé possédant un mécanisme d'ajustement des gains en temps réel permet de maintenir la stabilité du système face à deux pannes totales.[8]

7. Lien `https://github.com/ethz-asl/rotors_simulator`.
8. À 2:53 dans la vidéo disponible à l'adresse `https://youtu.be/8BS9n7uLMtI?t=95`.



(a) Évolution de gains

(b) Position du drone

Figure 4.20 Simulation de la panne de rotors sur un hexacoptère NNPPNP

4.5.5 Auto-séquencement de CA

Dans cette sous-section, on présente les résultats de simulation et de tests expérimentaux pour l'approche d'auto-séquencement de CA.

Simulation D'abord, le premier cas de test décrit dans le tableau 4.5 est simulé sous MATLAB/SIMULINK en utilisant le modèle non linéaire (3.24). La figure 4.21 montre la position du drone en vol stationnaire suite à deux pertes d'efficacité simultanées $\gamma_1 = 20\%$ et $\gamma_2 = 90\%$. On rappelle que ces fautes provoquent l'écrasement de l'appareil contrôlé par le LQR lors du vol expérimental (Sous-Section 4.5.1). En simulation, le drone demeure stable avec une grande oscillation en position (Fig. 4.21). Dans ce cas, l'auto-séquencement de CA (FTC3) permet d'améliorer la capacité de tolérance aux fautes du contrôleur nominal (LQR) : l'erreur maximale sur la position est réduite de 3.4613 m (LQR) à 2.0848 m (FTC3). Les temps pour que le drone reprenne sa position d'équilibre sont respectivement environ 9 s et 6 s.

La figure 4.21 démontre encore que le contrôleur auto-séquencé par LVE (FTC2) est le plus performant avec un déplacement maximal de 1.2727 m.[9] Pour bien évaluer les deux approches d'auto-séquencement (FTC2 et FTC3), les évolutions de la commande envoyée aux rotors sont données à la figure 4.22. On peut y voir que la loi CA auto-séquencée par LAE cause une saturation du signal de commande sur le 2e rotor,[10] tandis que le séquencement de gains

9. Similairement à la figure 4.14b, le résultat obtenu avec l'approche d'auto-séquencement par LAE (FTC1) est assez proche à celui de FTC2. Il n'est pas donc reporté dans cette sous-section.

10. $T_{i,max} = 9.3746$ N, cf. Annexe A.2.

par LVE peut l'éviter. De ce fait, on ne vérifie pas expérimentalement ce scénario pour éviter les risques possibles.

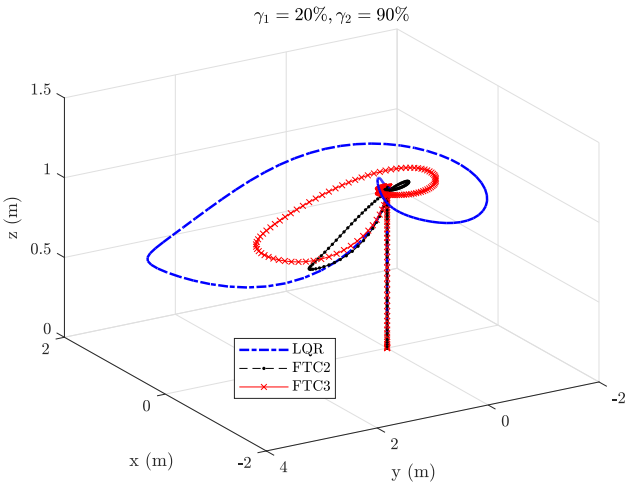


Figure 4.21 Simulation du cas 1 - Position du drone

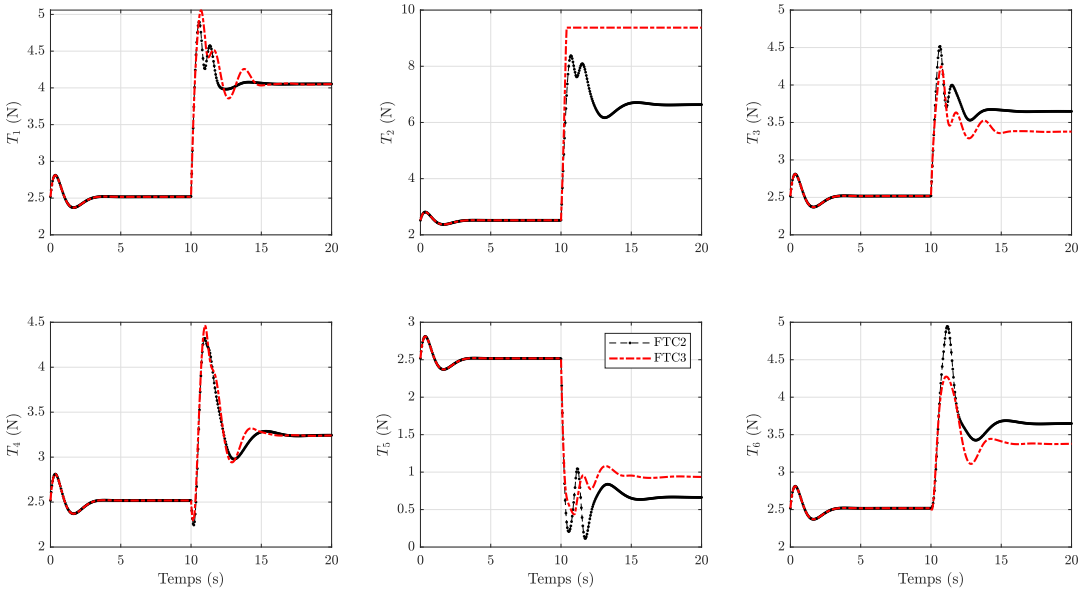
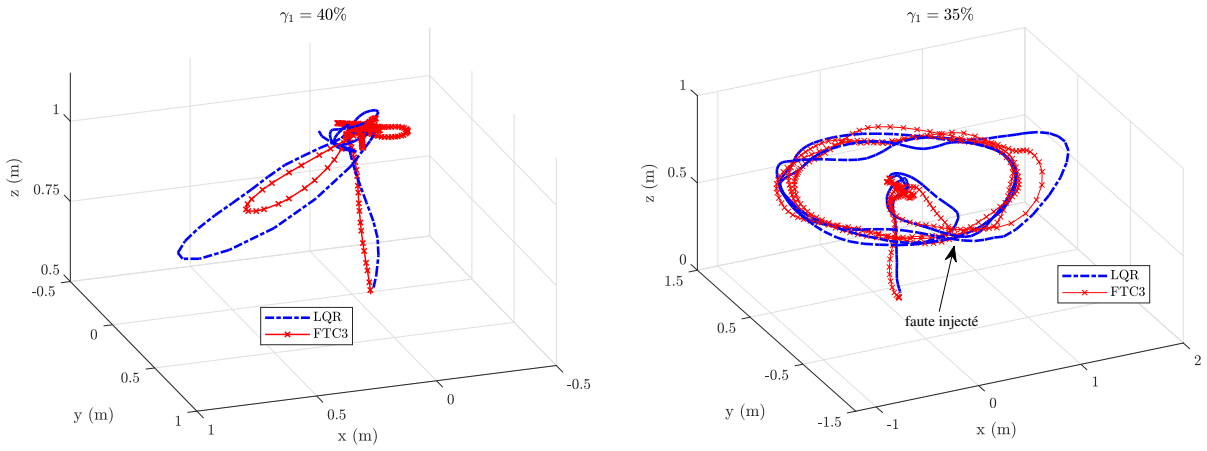


Figure 4.22 Simulation du cas 1 - Commandes sur les rotors

Résultats expérimentaux L'approche d'auto-séquencement de CA est ensuite vérifiée expérimentalement sur l'hexacoptère Firefly. Vu le résultat de simulation et pour des raisons de sécurité, les niveaux de fautes testés sont moins sévères que ceux listés dans le tableau 4.5. Plus précisément, on introduit artificiellement $\gamma_1 = 40\%$ à $t = 10\,\text{s}$ pour le vol stationnaire et $\gamma_1 = 35\%$ à $t = 15\,\text{s}$ pour le vol en suivi de trajectoire. On reprend ici la trajectoire désirée définie dans l'équation (4.35).

La loi CA proposée (FTC3) montre une amélioration considérable par rapport au contrôleur nominal (LQR) en maintien de position (Fig. 4.23a) et en précision de suivi de trajectoire (Fig. 4.23b). Les vitesses de rotation demandées aux rotors en vol stationnaire sont illustrées à la figure 4.24. Avant la faute, on peut déjà observer une différence entre les commandes du contrôleur nominal et celles du contrôleur augmenté par le séquencement de CA. Cette différence provient de l'erreur d'estimation de fautes telle que montrée à la figure 4.25. Après l'apparition de la faute ($\gamma_1 = 40\%$), le contrôleur auto-séquencé est capable de la compenser rapidement grâce à une nouvelle répartition de contrôle sur les rotors.

On remarque ainsi que les réponses temporelles en position ne sont pas aussi lisses que celles présentées précédemment (Fig. 4.15). Cela est dû au fait que le séquencement de CA est effectué en utilisant les valeurs LAE estimées en temps réel (Fig. 4.25). Similairement à la sous-section 4.5.3, on peut constater que l'estimation de fautes basée sur la dynamique de l'hexacoptère (Sous-Section 4.1.3) n'est pas précise à cause des bruits de mesures. En revanche, le résultat obtenu permet de démontrer que la CA séquencée fonctionne bien malgré cet inconvénient.



(a) Vol stationnaire

(b) Vol en suivi de trajectoire

Figure 4.23 Résultats expérimentaux pour l'auto-séquencement de CA

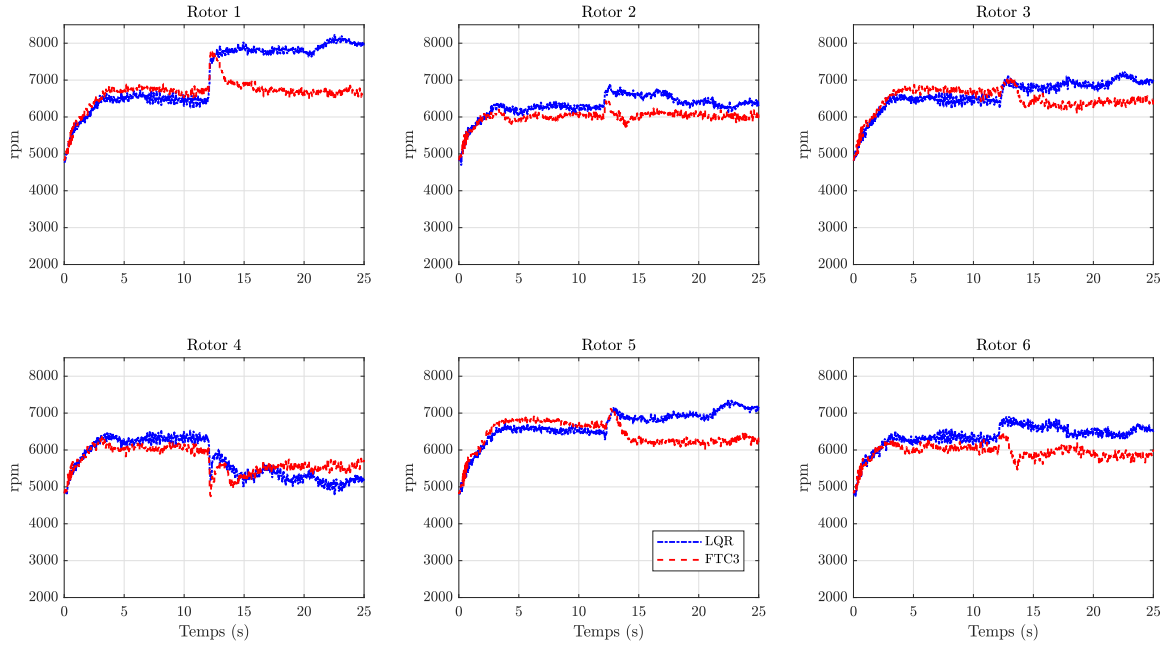


Figure 4.24 Vitesses de rotation des rotors pour un vol stationnaire avec $\gamma_1 = 40\%$ à $t = 12\,\mathrm{s}$

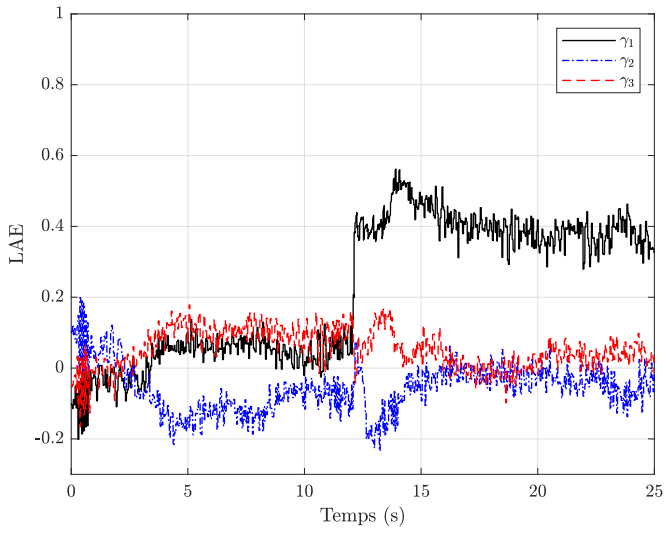


Figure 4.25 LAE estimée pour un vol stationnaire avec $\gamma_1 = 40\%$ à $t = 12\,\mathrm{s}$

4.6 Conclusion du chapitre

Dans ce chapitre, on a proposé trois approches FTC pour multicoptères basées sur l'auto-séquencement de gains et la synthèse $\mathcal{H}_\infty$ structurée. Les résultats de simulation et expérimentaux ont démontré que les contrôleurs auto-séquencés sont effectivement capables de compenser certains niveaux de fautes d'actionneurs. De plus, le bon fonctionnement du système FTC est aussi assuré lors d'une mauvaise estimation de fautes.

En examinant les résultats obtenus, on a constaté que l'auto-séquencement de gains par LAE est le plus performant au niveau de tolérance aux fautes. Cependant cette méthode présente plusieurs défis : un grand nombre de modèles à considérer durant la synthèse, une structure de séquencement de gains dépendant de la géométrie du multicoptère, une implémentation compliquée. La méthode d'auto-séquencement de gains par LVE permet de lever cette limitation. De plus, elle a présenté une performance assez proche à celle de la première méthode.

En pratique, la plupart des multicoptères commerciaux possèdent un contrôleur de base qui est synthétisé pour fonctionner dans des conditions nominales. Dans le but de tolérer les fautes sans modifier la loi de commande nominale d'un tel système, l'approche d'auto-séquencement de CA a été développée. Lors de vols expérimentaux, on a aussi obtenu une bonne performance du système en boucle fermée.

CHAPITRE 5 COMMANDE GÉNÉRIQUE TOLÉRANTE AUX FAUTES PAR DES MÉTHODES ADAPTATIVES ET ROBUSTES

Le chapitre 4 a montré que l'utilisation conjointe de l'auto-séquencement de gains et de la synthèse $\mathcal{H}_\infty$ structurée est une méthode pertinente dans le cadre de la tolérance aux fautes pour multicoptères. Néanmoins, cette approche nécessite souvent plusieurs itérations en simulation et en vérification expérimentale afin d'obtenir des résultats conformes aux attentes. De plus, les paramètres du multicoptère étant pris en compte durant la phase de synthèse, il faut ainsi répéter cette phase délicate pour chaque drone. De manière à éviter cet écueil, on aborde dans ce chapitre une stratégie de contrôle généralisée.

L'idée initiale est d'utiliser un vecteur d'entrée de commande normalisé. Ceci provient du fait que les drones possèdent le même modèle dynamique et que leur propre configuration est considérée dans la matrice d'allocation de contrôle. Il s'agit de tenir compte de la limitation physique de chaque drone lors de la synthèse. À cette fin, on propose une procédure de design en deux étapes. Dans un premier temps, on cherche à synthétiser un contrôleur tolérant aux fautes reposant sur la commande virtuelle normalisée. Dans un second temps, une transformation adaptative de type MRAC est ajoutée pour prendre en compte les incertitudes paramétriques ainsi que les limitations physiques du multicoptère. Pour pallier le phénomène de dérive des gains (*parameter drift*) en présence de perturbations non paramétriques mal-assorties (*unmatched uncertainties*), on modifie la loi adaptative par l'opérateur de projection. Des simulations et des vols expérimentaux ont été effectués sur différents multicoptères en utilisant un seul contrôleur avec des paramètres constants.

5.1 Approche adaptative

Dans cette section, on introduit la synthèse FTC générique pour multicoptères basée sur la MRAC et l'auto-séquencement de gains. L'architecture adoptée ici est illustrée à la figure 5.1. Utiliser la commande virtuelle normalisée et un mécanisme adaptatif permet de l'implémenter sur la plupart des multicoptères traditionnels. Ce travail a été reporté dans l'article de conférence [208].

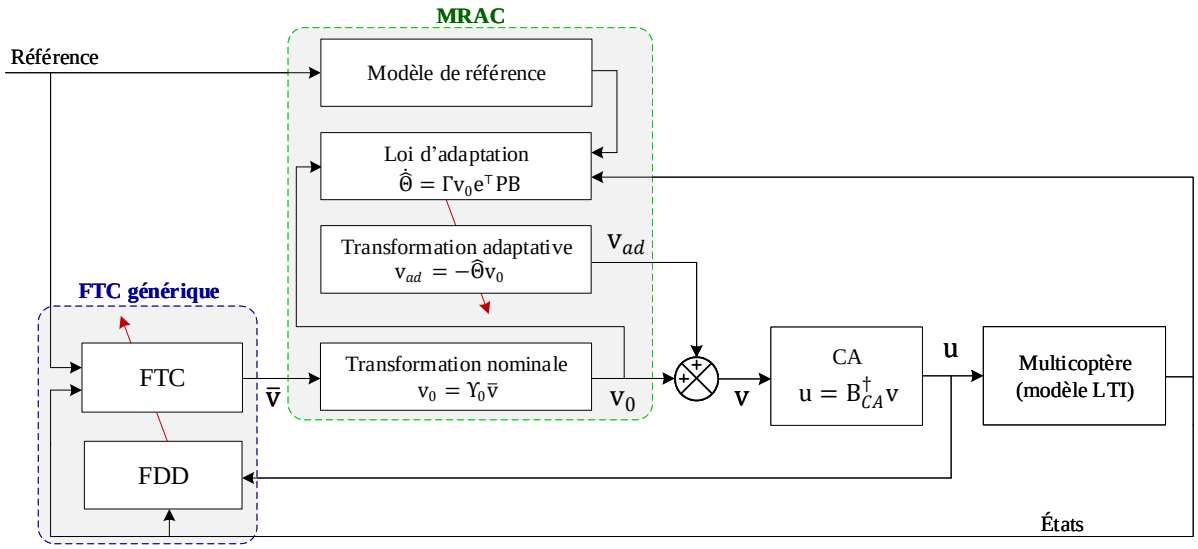


Figure 5.1 Architecture du contrôleur FTC générique pour multicoptères

5.1.1 Tolérance aux fautes générique

Avant de procéder à la synthèse du contrôleur, il est nécessaire de clarifier la définition de la commande virtuelle normalisée, notée $\bar{\mathbf{v}}$, de la façon suivante :

$$\bar{\mathbf{v}} \triangleq \begin{bmatrix} \bar{T} & \bar{L} & \bar{M} & \bar{N} \end{bmatrix}^\top = \begin{bmatrix} T/m \\ \mathbf{I}_b^{-1}\mathbf{M}^b \end{bmatrix} \tag{5.1}$$

où $\bar{T}$, $\bar{L}$, $\bar{M}$ et $\bar{N}$ dénotent respectivement la poussée totale normalisée et les moments normalisés selon les axes $\mathbf{x}_b$, $\mathbf{y}_b$ et $\mathbf{z}_b$. Dès lors, on s'intéresse à la conception d'un contrôleur fixe produisant ce nouveau vecteur d'entrée. En tirant une dernière fois profit de l'idée d'auto-séquencement de gains dans le cadre de la synthèse $\mathcal{H}_\infty$ structurée, on propose un contrôleur séquencé dont le design est assez identique à celui présenté dans la section 4.3. [1] La seule différence réside dans une modification du modèle linéarisé, soit :

$$\boldsymbol{\Delta}\dot{\mathbf{x}}_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\Delta}\mathbf{x}_x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1-\gamma_M \end{bmatrix} \Delta\bar{M}. \tag{5.2}$$

1. Bien évidemment, le concepteur peut choisir n'importe quelle stratégie de tolérance aux fautes.

La loi de commande prend donc la forme suivante :

$$\Delta\bar{M} = -\mathbf{K}\boldsymbol{\Delta}\mathbf{x}_x + K_i \int (\Delta x_r - \Delta x)\,\mathrm{d}t. \tag{5.3}$$

La synthèse du contrôleur est similaire à celle effectuée dans la sous-section 4.3 et n'est pas reportée ici. Le lecteur est renvoyé à [208] pour la démarche complète de design.

Pour ce qui est de l'implémentation du contrôleur, il suffit de récupérer la commande virtuelle en appliquant une transformation supplémentaire sous la forme :

$$\mathbf{v} = \boldsymbol{\Upsilon}\bar{\mathbf{v}} = -\boldsymbol{\Upsilon}\mathbf{K}\mathbf{x} \tag{5.4}$$

où $\boldsymbol{\Upsilon} = \mathrm{diag}\,(m, \mathbf{I}_b)$ et $\mathbf{K}$ est le gain séquencé de retour d'état. Le reste demeure similaire à celui effectué dans la sous-section 3.2.2.

5.1.2 Augmentation par MRAC

Le contrôleur précédemment synthétisé présente l'avantage d'une grande simplicité dans son design et sa capacité *universelle*. Or, les paramètres du multicoptère (masse m et inertie $\mathbf{I}_b$) n'étant pas considérés dans la phase de synthèse, un mécanisme adaptatif est nécessaire. Sur la base du système linéarisé (3.25), on voit que seule la dynamique de la boucle interne (Fig. 3.3) est directement influencée par ces paramètres du système. De manière à simplifier la synthèse sans oublier l'angle de cap ψ dans l'implémentation (Eq. (3.36)), le modèle d'état suivant est utilisé :

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\Lambda}\mathbf{v} + \mathbf{B}_r\mathbf{r} \tag{5.5}$$

où $\mathbf{x} = [z \;\; \dot{z} \;\; e_{zI} \;\; \phi \;\; p \;\; \theta \;\; q \;\; \psi \;\; r \;\; e_{\psi I}]^\top$ est le vecteur d'état augmenté avec les états des intégrateurs sur les erreurs de suivi e_{zI} et $e_{\psi I}$, $\mathbf{v}$ le vecteur d'entrée virtuelle (3.27) et $\mathbf{r} = \left[z_r \;\; u_y^b \;\; u_x^b \;\; \psi_r\right]^\top$ le vecteur de commande externe borné.[2] Les matrices diagonales par blocs $\mathbf{A}$, $\mathbf{B}$ et $\mathbf{B}_r$ sont connues et peuvent être obtenues facilement à partir de l'équation (3.25). La matrice inconnue $\boldsymbol{\Lambda} \in \mathbb{R}^{4\times4}$ représente les incertitudes paramétriques affectant le vecteur d'entrée, soit m, I_{xx}, I_{yy} et I_{zz} dans le cas du multicoptère. On suppose ici que $\boldsymbol{\Lambda}$ est diagonale définie positive, ce qui implique que la paire $(\mathbf{A}, \mathbf{B}\boldsymbol{\Lambda})$ est commandable.

De manière à maintenir une performance adéquate du système FTC en dépit de la matrice inconnue $\boldsymbol{\Lambda}$, la loi de transformation (5.4) est augmentée avec une composante adaptative :

$$\mathbf{v} = \mathbf{v}_0 + \mathbf{v}_{ad} \tag{5.6}$$

2. Par simplicité, la notation de déviation Δ est omise dans cette section.

où $\mathbf{v}_0 = \boldsymbol{\Upsilon}_0\bar{\mathbf{v}} = -\boldsymbol{\Upsilon}_0\mathbf{K}\mathbf{x}$ est la sortie du contrôleur par retour d'état précédemment synthétisé. La matrice diagonale $\boldsymbol{\Upsilon}_0$ représente la transformation nominale. En substituant $\mathbf{v}$ dans (5.5) par (5.6), on obtient :

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\Lambda}\left(\mathbf{v}_0 + \mathbf{v}_{ad}\right) + \mathbf{B}_r\mathbf{r} \\ &= \underbrace{\left(\mathbf{A} - \mathbf{B}\boldsymbol{\Upsilon}_0\mathbf{K}\right)}_{\mathbf{A}_{ref}}\mathbf{x} + \mathbf{B}\boldsymbol{\Lambda}\left[\boldsymbol{\Theta}\mathbf{v}_0 + \mathbf{v}_{ad}\right] + \mathbf{B}_r\mathbf{r}\end{aligned} \tag{5.7}$$

où $\boldsymbol{\Theta} \triangleq \operatorname{diag}\left(\theta_1, \ldots, \theta_4\right) = \mathbf{I}_4 - \boldsymbol{\Lambda}^{-1}$ est diagonale et inconnue.

On considère le modèle de référence suivant :

$$\dot{\mathbf{x}}_{ref} = \mathbf{A}_{ref}\mathbf{x}_{ref} + \mathbf{B}_r\mathbf{r} \tag{5.8}$$

où $\mathbf{A}_{ref} = (\mathbf{A} - \mathbf{B}\boldsymbol{\Upsilon}_\mathbf{0}\mathbf{K})$ est supposée Hurwitz par design et $\mathbf{x}_{ref}(t)$ représente le comportement du système dans le cas nominal, c'est-à-dire $\boldsymbol{\Lambda} = \mathbf{I}_4$. L'objectif de la commande adaptative est de forcer l'état du système $\mathbf{x}(t)$ à suivre globalement asymptotiquement l'état du modèle de référence $\mathbf{x}_{ref}(t)$. Autrement dit, l'erreur de suivi :

$$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{x}_{ref}(t) \tag{5.9}$$

tend vers l'origine globalement et asymptotiquement. En comparant la dynamique du système bouclé (5.7) et la dynamique désirée (5.8), on propose la transformation adaptative suivante :

$$\mathbf{v}_{ad} = -\hat{\boldsymbol{\Theta}}\mathbf{v}_0 \tag{5.10}$$

où $\hat{\boldsymbol{\Theta}}$ est l'estimation de la matrice inconnue $\boldsymbol{\Theta}$. Notons que $\hat{\boldsymbol{\Theta}} \in \mathbb{R}^{4\times4}$ est aussi diagonale. Si l'estimation est parfaite, soit $\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}$, on retrouvera donc le modèle de référence ((5.8)). Cependant, un tel gain idéal n'existe pas en pratique à cause du paramètre inconnu $\boldsymbol{\Lambda}$. Pour définir la dynamique du gain adaptatif, on choisit la loi d'adaptation classique reposant sur la théorie de Lyapunov :

$$\dot{\hat{\boldsymbol{\Theta}}} = \boldsymbol{\Gamma}\mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B} \tag{5.11}$$

où les matrices $\boldsymbol{\Gamma}$, $\mathbf{P}$ et $\mathbf{Q}$ sont symétriques définies positives de dimensions appropriées. De plus, $\boldsymbol{\Gamma}$ est diagonale et représente le taux d'adaptation. On note $\mathbf{P} = \mathbf{P}^\top$ la solution unique de l'équation algébrique de Lyapunov :

$$\mathbf{P}\mathbf{A}_{ref} + \mathbf{A}_{ref}^\top\mathbf{P} = -\mathbf{Q} \tag{5.12}$$

pour un choix adéquat de $\mathbf{Q} = \mathbf{Q}^\top > 0$.

La figure 5.2 montre le schéma bloc détaillé de la loi adaptative proposée.

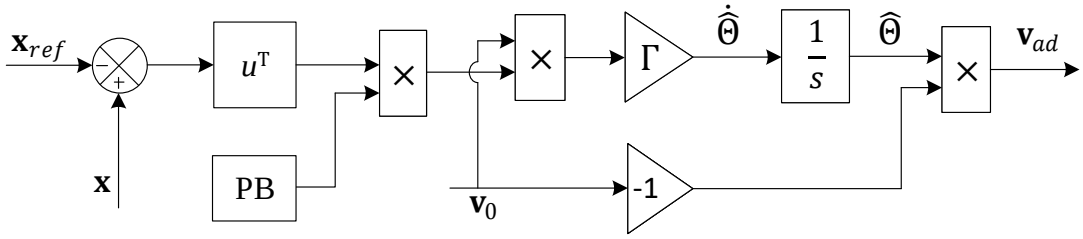


Figure 5.2 Loi d'adaptation

5.1.3 Analyse de la stabilité

Afin de démontrer la stabilité du système (5.5) en boucle fermée avec le contrôleur de type MRAC (Eqs. (5.6) et (5.10)) et la loi d'adaptation (5.11), on établit d'abord la dynamique d'erreur de suivi à partir des équations (5.7), (5.8) et (5.10) :

$$\begin{aligned}\dot{\mathbf{e}} &= \dot{\mathbf{x}} - \dot{\mathbf{x}}_{ref} \\ &= (\mathbf{A}_{ref}\mathbf{x} + \mathbf{B}\boldsymbol{\Lambda}\left[\boldsymbol{\Theta}\mathbf{v}_0 + \mathbf{v}_{ad}\right] + \mathbf{B}_r\mathbf{r}) - (\mathbf{A}_{ref}\mathbf{x}_{ref} + \mathbf{B}_r\mathbf{r}) \\ &= \mathbf{A}_{ref}\mathbf{e} - \mathbf{B}\boldsymbol{\Lambda}\boldsymbol{\Delta\Theta}\mathbf{v}_0\end{aligned} \tag{5.13}$$

où $\boldsymbol{\Delta\Theta} := \hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}$ est l'erreur d'estimation des paramètres adaptatifs. On choisit la fonction candidate de Lyapunov radialement non bornée :

$$V = \mathbf{e}^\top\mathbf{P}\mathbf{e} + \operatorname{trace}\left(\boldsymbol{\Delta\Theta}\boldsymbol{\Gamma}^{-1}\boldsymbol{\Delta\Theta}^\top\boldsymbol{\Lambda}\right) \tag{5.14}$$

La dérivée temporelle de V le long de la trajectoire du modèle (5.13) est donnée par :

$$\begin{aligned}\dot{V} &= 2\mathbf{e}^\top\mathbf{P}\dot{\mathbf{e}} + 2\operatorname{trace}\left(\boldsymbol{\Delta\Theta}\boldsymbol{\Gamma}^{-1}\dot{\hat{\boldsymbol{\Theta}}}^\top\boldsymbol{\Lambda}\right) \\ &= 2\mathbf{e}^\top\mathbf{P}\mathbf{A}_{ref}\mathbf{e} - 2\mathbf{e}^\top\mathbf{P}\mathbf{B}\boldsymbol{\Lambda}\boldsymbol{\Delta\Theta}\mathbf{v}_0 + 2\operatorname{trace}\left(\boldsymbol{\Delta\Theta}\boldsymbol{\Gamma}^{-1}\dot{\hat{\boldsymbol{\Theta}}}^\top\boldsymbol{\Lambda}\right).\end{aligned} \tag{5.15}$$

Le premier terme est simplifié grâce à l'équation de Lyapunov (5.12) :

$$2\mathbf{e}^\top\mathbf{P}\mathbf{A}_{ref}\mathbf{e} = \mathbf{e}^\top\mathbf{P}\mathbf{A}_{ref}\mathbf{e} + \mathbf{e}^\top\mathbf{P}\mathbf{A}_{ref}\mathbf{e} = \mathbf{e}^\top\mathbf{P}\mathbf{A}_{ref}\mathbf{e} + \mathbf{e}^\top\mathbf{A}_{ref}^\top\mathbf{P}\mathbf{e} = -\mathbf{e}^\top\mathbf{Q}\mathbf{e}. \tag{5.16}$$

En utilisant l'identité $\mathbf{a}^\top\mathbf{b} = \text{trace}\left(\mathbf{b}\mathbf{a}^\top\right)$ qui est valide pour deux vecteurs de même dimension, on obtient :

$$\underbrace{\mathbf{e}^\top\mathbf{P}\mathbf{B}\boldsymbol{\Lambda}}_{\mathbf{a}^\top}\underbrace{\boldsymbol{\Delta}\boldsymbol{\Theta}\mathbf{v}_0}_{\mathbf{b}} = \text{trace}\left(\boldsymbol{\Delta}\boldsymbol{\Theta}\mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\boldsymbol{\Lambda}\right). \tag{5.17}$$

L'équation (5.15) est donc reformulée comme suit :

$$\dot{V} = -\mathbf{e}^\top\mathbf{Q}\mathbf{e} + 2\,\text{trace}\left[\boldsymbol{\Delta}\boldsymbol{\Theta}\left(\boldsymbol{\Gamma}^{-1}\dot{\hat{\boldsymbol{\Theta}}} - \mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\right)\boldsymbol{\Lambda}\right]. \tag{5.18}$$

En imposant la loi adaptative (5.11), on a :

$$\dot{V} = -\mathbf{e}^\top\mathbf{Q}\mathbf{e} \leq 0. \tag{5.19}$$

L'origine du système (5.13) est donc (localement) stable au sens de Lyapunov. En appliquant le lemme de Barbalat [229, Chapitre 8], on a de plus $\lim_{t\to\infty}\dot{V} = 0$. Par conséquent, $\lim_{t\to\infty}\mathbf{e}(t) = 0$, ce qui implique que l'origine est asymptotiquement stable. Finalement, la fonction de Lyapunov V étant radialement non bornée, on conclut que $\mathbf{e}(t)$ tend vers l'origine globalement et asymptotiquement lorsque $t \to \infty$, ce qui complète la preuve de stabilité. En résumé, le système incertain (5.5) est garanti de converger vers le modèle de référence (5.8).

Le lecteur intéressé est invité à se référer à [229, Chapitre 4] pour plus de détails concernant les notions de stabilité de Lyapunov et à [19] pour une démonstration plus détaillée de la stabilité du système MRAC.

Remarque 5.1.1. En ce qui concerne le modèle de référence, on peut aussi choisir :

$$\mathbf{A}_{ref} = (\mathbf{A} - \mathbf{B}\boldsymbol{\Upsilon}_\mathbf{0}\mathbf{K}_\mathbf{0}) \tag{5.20}$$

où $\mathbf{K}_0$ est la valeur du gain séquencé $\mathbf{K}$ au point d'opération normale. Ce choix garantit que le modèle de référence n'est pas changé durant le vol. De plus, de manière à ce que la stratégie de contrôle s'adapte aux critères spécifiques du multicoptère, le concepteur peut imposer n'importe quel modèle de référence approprié dont la structure diagonale par blocs doit être respectée.

Remarque 5.1.2. Non seulement la matrice inconnue $\boldsymbol{\Lambda}$ introduit l'erreur de modélisation, mais elle peut représenter les pertes d'efficacité de contrôle. Cela signifie que l'algorithme adaptatif proposé a également une capacité de tolérance aux fautes supplémentaire. Comme on peut le voir ultérieurement, cette augmentation par MRAC permet de maintenir au mieux le niveau de performance original en présence de fautes.

5.2 Approche adaptative et robuste

5.2.1 Modification robuste par opérateur de projection

Considérons maintenant le modèle d'état (5.5) en introduisant la perturbation non paramétrique mal-assortie $\boldsymbol{\xi}(t) \in \mathbb{R}^n$:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\Lambda}\mathbf{v} + \mathbf{B}_r\mathbf{r} + \boldsymbol{\xi}. \tag{5.21}$$

On suppose que $\boldsymbol{\xi}(t)$ est uniformément bornée dans le temps, soit $\|\boldsymbol{\xi}(t)\| \leq \xi_{max}$. La dynamique de l'erreur de suivi est donnée par :

$$\dot{\mathbf{e}} = \dot{\mathbf{x}} - \dot{\mathbf{x}}_{ref} = \mathbf{A}_{ref}\mathbf{e} - \mathbf{B}\boldsymbol{\Lambda}\boldsymbol{\Delta\Theta}\mathbf{v}_0 + \boldsymbol{\xi}. \tag{5.22}$$

En utilisant la même fonction candidate de Lyapunov (5.14), on a :

$$\dot{V}(\mathbf{e}, \boldsymbol{\Delta\Theta}) = -\mathbf{e}^\top\mathbf{Q}\mathbf{e} + 2\,\mathrm{trace}\left[\boldsymbol{\Delta\Theta}\left(\boldsymbol{\Gamma}^{-1}\dot{\hat{\boldsymbol{\Theta}}} - \mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\right)\boldsymbol{\Lambda}\right] + 2\mathbf{e}^\top\mathbf{P}\boldsymbol{\xi}. \tag{5.23}$$

En imposant la loi adaptative (5.11), on a :

$$\dot{V}(\mathbf{e}, \boldsymbol{\Delta\Theta}) = -\mathbf{e}^\top\mathbf{Q}\mathbf{e} + 2\mathbf{e}^\top\mathbf{P}\boldsymbol{\xi} \leq -\lambda_{min}(\mathbf{Q})\|\mathbf{e}\|^2 + 2\|\mathbf{e}\|\lambda_{max}(\mathbf{P})\xi_{max}. \tag{5.24}$$

Ainsi, nous avons $\dot{V}(\mathbf{e}, \boldsymbol{\Delta\Theta}) \leq 0$ à l'extérieur de l'ensemble :

$$E_0 = \left\{ (\mathbf{e}, \boldsymbol{\Delta\Theta}) : \|\mathbf{e}\| \leq 2\frac{\lambda_{max}(\mathbf{P})}{\lambda_{min}(\mathbf{Q})}\xi_{max} =: e_0 \right\} \tag{5.25}$$

Selon [229, Théorème 4.18], la trajectoire $\mathbf{e}(t)$ rentre dans un ensemble compact ($\Omega_0 \supset E_0$) $\subset \mathbb{R}^n$ dans un temps fini et ne quittera jamais cet ensemble. Cependant, le paramètre adaptatif $\boldsymbol{\Theta}$ peut être non borné, car $\dot{V}(\mathbf{e}, \boldsymbol{\Delta\Theta})$ peut être positif dans Ω_0. Ce phénomène de dérive des gains, souvent appelé *parameter drift* en anglais, peut causer des problèmes de stabilité et éventuellement mener à l'écrasement de l'appareil.

Pour borner le gain adaptatif $\boldsymbol{\Theta}$, on reformule la loi d'adaptation (5.11) avec l'opérateur de projection (cf. [230]) comme suit :

$$\dot{\hat{\boldsymbol{\Theta}}} = \boldsymbol{\Gamma}\mathrm{Proj}\left(\hat{\boldsymbol{\Theta}}, \mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\right) \tag{5.26}$$

où Proj dénote l'opérateur de projection. Les notions élémentaires concernant cet opérateur sont rassemblées dans l'annexe C. Plus précisément, on utilise la définition C.2.2, puisque

$\hat{\boldsymbol{\Theta}} = \left[\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_4\right]$ est une matrice de dimension 4×4 dans ce cas particulier. La fonction convexe utilisée pour construire l'opérateur de projection est la suivante :

$$f_i(\hat{\boldsymbol{\theta}}_i) = \frac{(1+\epsilon_i)\,\hat{\boldsymbol{\theta}}_i^{\top}\hat{\boldsymbol{\theta}}_i - \theta_{i,max}^2}{\epsilon_i \theta_{i,max}^2},\ i = 1, \ldots, 4 \tag{5.27}$$

où $\epsilon_i > 0$ est la tolérance de l'opérateur et $\theta_{i,max}$ est la borne imposée à $\|\hat{\boldsymbol{\theta}}_i\|$. Le gradient de la fonction $f_i(\hat{\boldsymbol{\theta}}_i)$ est donc donné par :

$$\nabla f_i(\hat{\boldsymbol{\theta}}_i) = 2\frac{1+\epsilon_i}{\epsilon_i \theta_{i,max}^2}\hat{\boldsymbol{\theta}}_i. \tag{5.28}$$

La figure 5.3 montre le schéma bloc détaillé de la loi de transformation augmentée avec l'opérateur de projection (Eqs. (5.10) et (5.26)).

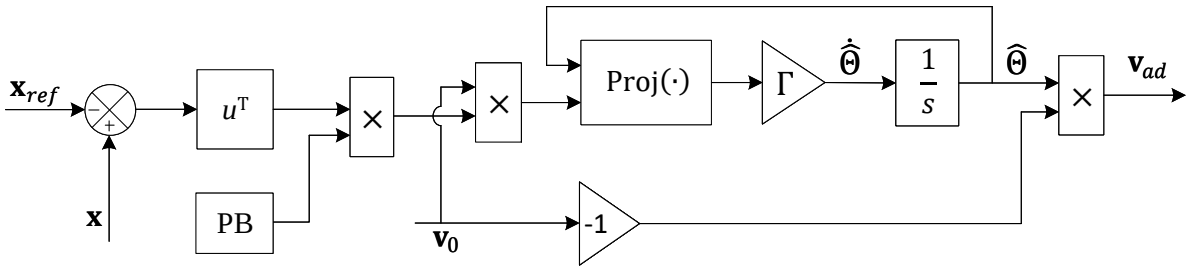


Figure 5.3 Loi d'adaptation avec modification robuste

5.2.2 Analyse de la stabilité

Avant de procéder à l'analyse de stabilité, on commence par les hypothèses :

$$\boldsymbol{\theta}_i \in \Omega_{i,0} \triangleq \left\{\boldsymbol{\theta}_i \in \mathbb{R}^4 \middle|\ f_i(\boldsymbol{\theta}_i) \leq 0\right\} = \left\{\boldsymbol{\theta}_i \in \mathbb{R}^4 \middle|\ \|\boldsymbol{\theta}_i\| \leq \frac{\theta_{i,max}}{\sqrt{1+\epsilon_i}}\right\},\ i = 1, \ldots, 4 \tag{5.29}$$

$$\hat{\boldsymbol{\theta}}_i(0) \in \Omega_{i,1} \triangleq \left\{\hat{\boldsymbol{\theta}}_i \in \mathbb{R}^4 \middle|\ f_i(\hat{\boldsymbol{\theta}}_i) \leq 1\right\} = \left\{\hat{\boldsymbol{\theta}}_i \in \mathbb{R}^4 \middle|\ \|\hat{\boldsymbol{\theta}}_i\| \leq \theta_{i,max}\right\},\ i = 1, \ldots, 4 \tag{5.30}$$

L'hypothèse (5.29) dépend du choix des paramètres ($\theta_{i,max}$ et ϵ_i) et les caractéristiques physiques du système (matrice inconnue $\boldsymbol{\Lambda}$). L'hypothèse (5.30) est toujours vérifiée si le gain adaptatif commence son évolution à partir de l'origine ($f_i(\mathbf{0}) = -1/\epsilon_i < 0$).

La dérivée temporelle de la fonction candidate de Lyapunov (5.14) avec la loi d'adaptation

(5.26) s'écrit :

$$\dot{V}(\mathbf{e}, \boldsymbol{\Delta\Theta}) = -\mathbf{e}^\top \mathbf{Q}\mathbf{e} + 2\,\mathrm{trace}\left[\boldsymbol{\Delta\Theta}\left(\mathrm{Proj}\left(\hat{\boldsymbol{\Theta}}, \mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\right) - \mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\right)\boldsymbol{\Lambda}\right] + 2\mathbf{e}^\top\mathbf{P}\boldsymbol{\xi}. \quad (5.31)$$

Sous l'hypothèse (5.29) et en notant que $\boldsymbol{\Delta\Theta} = \hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}$ est diagonale, on peut utiliser le lemme C.2.3 pour montrer que :

$$\mathrm{trace}\left[\boldsymbol{\Delta\Theta}\left(\mathrm{Proj}\left(\hat{\boldsymbol{\Theta}}, \mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\right) - \mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\right)\right] \leq 0. \quad (5.32)$$

De plus, comme la matrice $\boldsymbol{\Lambda}$ est diagonale définie positive, on obtient :

$$\mathrm{trace}\left[\boldsymbol{\Delta\Theta}\left(\mathrm{Proj}\left(\hat{\boldsymbol{\Theta}}, \mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\right) - \mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\right)\boldsymbol{\Lambda}\right] \leq 0. \quad (5.33)$$

On retrouve donc l'inégalité (5.24). Dans ce cas, sous l'hypothèse (5.30), le lemme C.2.1 garantit que $\hat{\boldsymbol{\theta}}_i(t) \in \Omega_{i,1}, \forall t \geq 0$, soit $\|\hat{\boldsymbol{\theta}}_i\| \leq \theta_{i,max}$. Ainsi, à l'extérieur de l'ensemble

$$\Omega \triangleq \left\{ (\mathbf{e}, \boldsymbol{\Delta\Theta}) : \|\mathbf{e}\| \leq e_0 \cap \|\Delta\boldsymbol{\theta}_1\| \leq 2\theta_{1,max} \cap \ldots \cap \|\Delta\boldsymbol{\theta}_4\| \leq 2\theta_{4,max} \right\} \quad (5.34)$$

nous avons $\dot{V} \leq 0$, ce qui indique que l'erreur de suivi $\mathbf{e}$ et l'erreur d'estimation des paramètres adaptatifs $\boldsymbol{\Delta\Theta}$ entreront dans un ensemble compact recouvrant Ω dans un temps fini et n'en ressortiront plus. Le paramètre ϵ_0 (cf. Eq. (5.25)) représente une tolérance sur l'erreur minimale de suivi. Dans le cas $\|\mathbf{e}\| \leq \epsilon_0$ et si $\dot{V} > 0$, alors la dynamique du système quittera Ω tout en conservant $\|\hat{\boldsymbol{\theta}}_i\| \leq \theta_{i,max}$. Une fois à l'extérieur de Ω, le système se stabilisera à la frontière de Ω ($\dot{V} \leq 0$).

Ainsi, l'opérateur de projection permet de borner tous les signaux du système. En d'autres termes, on passe de la stabilité asymptotique globale vers la stabilité ultimement uniformément bornée (UUB) lorsqu'il y a des perturbations non paramétriques bornées.

5.3 Simulation et tests expérimentaux

Plusieurs simulations et vols expérimentaux ont été effectués pour vérifier la stratégie de contrôle proposée. Dans le but de démontrer la portabilité du système FTC générique, les contrôleurs ont été implémentés sur deux plates-formes différentes : un quadricoptère de petite taille et l'hexacoptère Firefly que l'on peut considérer de taille moyenne ; les paramètres sont fournis en annexe A.2.

On utilise le séquencement de gains suivant pour les deux multicoptères :

$$\begin{aligned}
\mathbf{K}_x &= \begin{bmatrix}30 & 17.7 & 56.3 & 9.9\end{bmatrix} + \begin{bmatrix}11.4 & 17.4 & 54.8 & 12.2\end{bmatrix}\gamma_M + \begin{bmatrix}-6.6 & -5.3 & 46.3 & 22.9\end{bmatrix}\gamma_M^2 \\
\mathbf{K}_y &= \begin{bmatrix}-27.6 & -15.4 & 51 & 9.3\end{bmatrix} + \begin{bmatrix}-80.8 & -50.4 & 156 & 29\end{bmatrix}\gamma_L \\
\mathbf{K}_z &= \begin{bmatrix}9.6 & 4.4\end{bmatrix} + \begin{bmatrix}7.8 & 8.2\end{bmatrix}\gamma_T \\
\mathbf{K}_\psi &= \begin{bmatrix}5.1 & 3.2\end{bmatrix} + \begin{bmatrix}7.1 & 5.7\end{bmatrix}\gamma_N \\
K_{i,x} &= 24.5 - 3\gamma_M - 3.3\gamma_M^2,\ K_{i,y} = -21.4 - 61.6\gamma_L,\ K_{i,z} = 8 + 2.8\gamma_T,\ K_{i,\psi} = 3.2 + 4.5\gamma_N.
\end{aligned} \tag{5.35}$$

L'évolution des gains du contrôleur longitudinal par rapport à la LVE est presque linéaire (Fig. 5.4). Par simplicité, on choisit des séquencements de gains affines pour les autres contrôleurs.

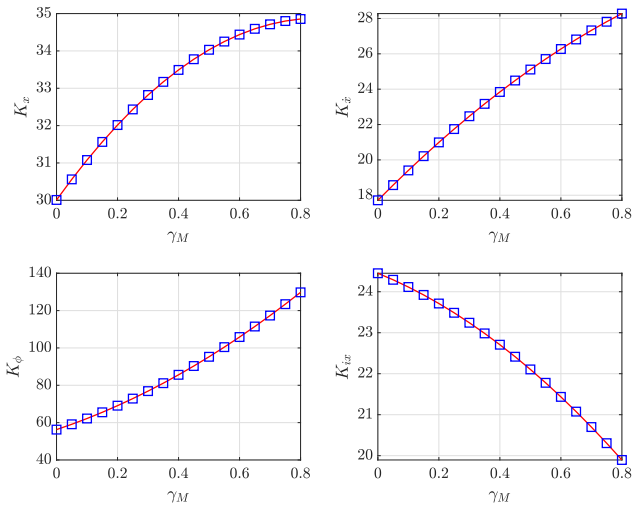


Figure 5.4 Évolution de gains du contrôleur longitudinal par rapport à LVE

La loi d'adaptation est obtenue en utilisant :

$$\mathbf{\Gamma} = \mathrm{diag}\,(0.1, 1, 1, 1) \quad \text{et} \quad \mathbf{Q} = \mathrm{diag}\,(\mathbf{I}_3, \mathbf{I}_2, \mathbf{I}_2, 10\mathbf{I}_3)\,. \tag{5.36}$$

L'opérateur de projection est construit avec :

$$\epsilon_i = 0.001 \ \text{ et } \ \theta_{i,max} = 1,\ i = 1, \ldots, 4. \tag{5.37}$$

Ces valeurs sont choisies après plusieurs itérations en simulation pour obtenir une performance adéquate, une commande réalisable et le moins possible d'oscillations transitoires. On note également que tous les paramètres adaptatifs sont initialisés à zéro.

5.3.1 Simulation sur un quadricoptère

On simule les réponses temporelles d'un quadricoptère pour deux lois d'adaptation précédemment synthétisées. L'objectif de ces simulations est de vérifier le fonctionnement du contrôleur générique tolérant aux fautes, l'adaptation aux paramètres incertains de l'augmentation MRAC et l'effet de la modification robuste par opérateur de projection.

Condition nominale On vérifie d'abord le comportement du système en boucle fermée dans les conditions nominales (sans incertitudes). La figure 5.5 montre les résultats obtenus pour une série d'échelons de commande en z et en ψ avec un angle de lacet initial de $\pi/4$ rad. Les réponses temporelles du drone et celles du modèle de référence sont identiques. On rappelle que le modèle de référence (Eq. (5.20)) représente la performance du système bouclé avec le contrôleur générique de base (cf. Sous-Section 5.1.1) dans les conditions idéales.

La figure 5.5 montre également que les commandes envoyées aux rotors résident dans les limites réalisables et raisonnables.[3]

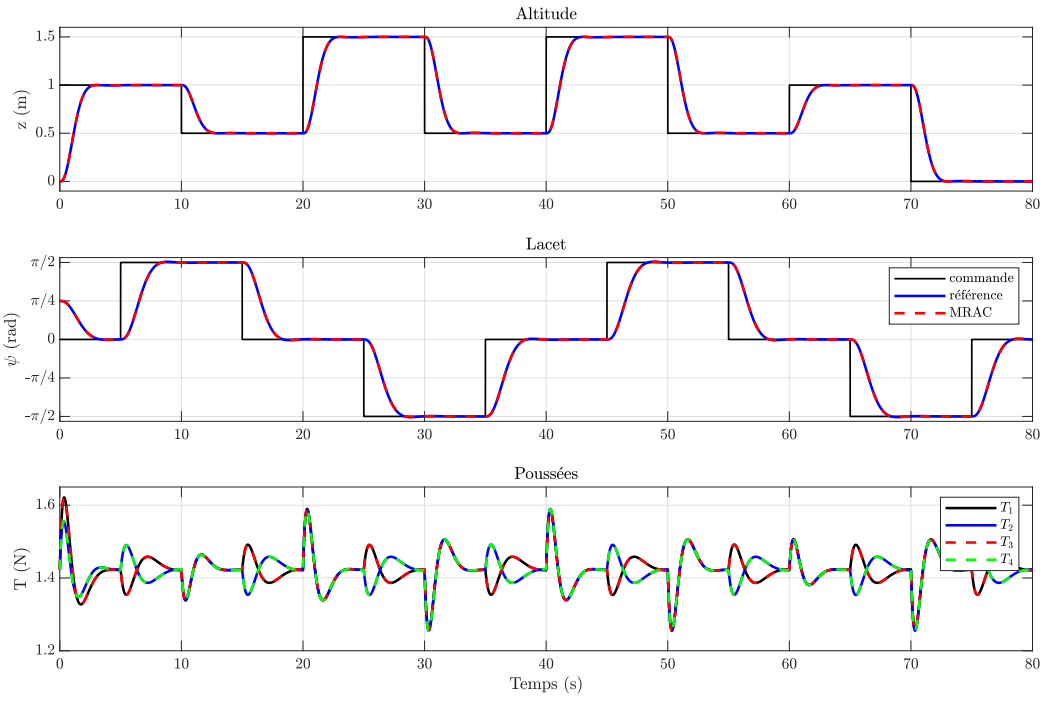


Figure 5.5 Réponse du quadricoptère en boucle fermée au point d'opération normale

3. $T_{i,max} = 3.75\,\text{N}$, cf. Annexe A.2.

Phénomène de dérive de gains On simule ensuite le système en introduisant les perturbations non paramétriques mal-assorties, soit $\boldsymbol{\xi}(t)$ dans l'équation (5.21). Plus précisément, les états $\dot{\phi}, \dot{\theta}, \dot{\psi}$ et $\dot{z}$ sont perturbés par des bruits blancs possédant une densité spectrale de 0.05 et une fréquence de 10 Hz. Les réponses temporelles sont encore acceptables tel que montré sur la figure 5.6. Cependant, les paramètres adaptatifs présentent le problème de dérive indésirable (Fig. 5.7). Le paramètre le plus affecté par ce phénomène est $\hat{\theta}_4$. On remarque également que le seul paramètre $\hat{\theta}_1$ ne diverge pas et reste autour de sa vraie valeur.

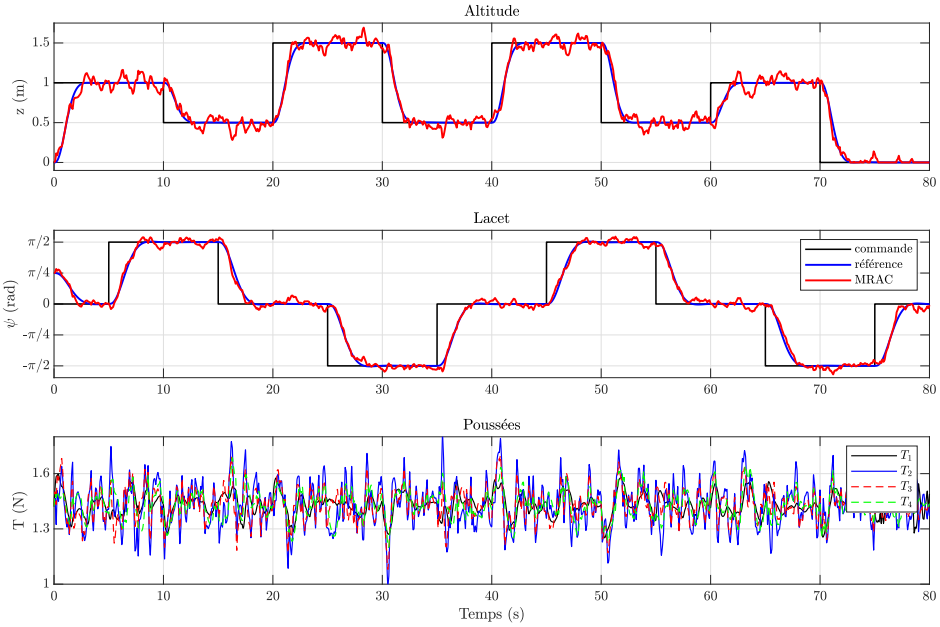


Figure 5.6 Réponse du quadricoptère en boucle fermée en présence de bruits

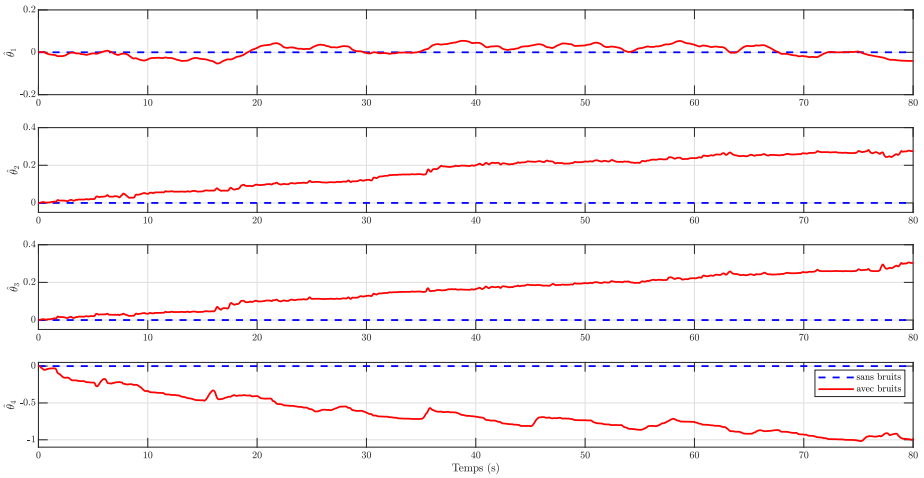


Figure 5.7 Phénomène de dérive de gains en présence de bruits

Adaptation aux paramètres incertains On vérifie maintenant le fonctionnement du contrôleur générique adaptatif en présence de paramètres incertains. À des fins de simulation, on suppose que les paramètres inconnus du quadricoptère sont :

$$m = 1.5m_0 \text{ et } \mathbf{I}_b = 2\mathbf{I}_{b,0} \tag{5.38}$$

où m_0 et $\mathbf{I}_{b,0}$ sont les valeurs nominales données en Annexe A.2. En observant le modèle d'état (5.5), la valeur idéale de la matrice des paramètres adaptatifs $\hat{\mathbf{\Theta}}$ est donnée par :

$$\mathbf{\Theta} = \text{diag}\left(\theta_1, \ldots, \theta_4\right) = \mathbf{I}_4 - \mathbf{\Lambda}^{-1} = \text{diag}\left(-0.5, -1, -1, -1\right). \tag{5.39}$$

La figure 5.8 donne les résultats de cette simulation. Les réponses temporelles du système sont comparables à celles présentées à la figure 5.5, bien que le décollage du drone soit ralenti à cause de la masse additionnelle. Il est intéressant d'observer que les paramètres adaptatifs $\hat{\theta}_1$ et $\hat{\theta}_4$ convergent bien vers leurs valeurs idéales (Fig. 5.9). On rappelle que la convergence de paramètres n'est pas garantie par la MRAC. En effet, $\hat{\theta}_2$ et $\hat{\theta}_3$ ne varient pas, car le drone ne se déplace pas dans le plan $\mathbf{x}_i$–$\mathbf{y}_i$.

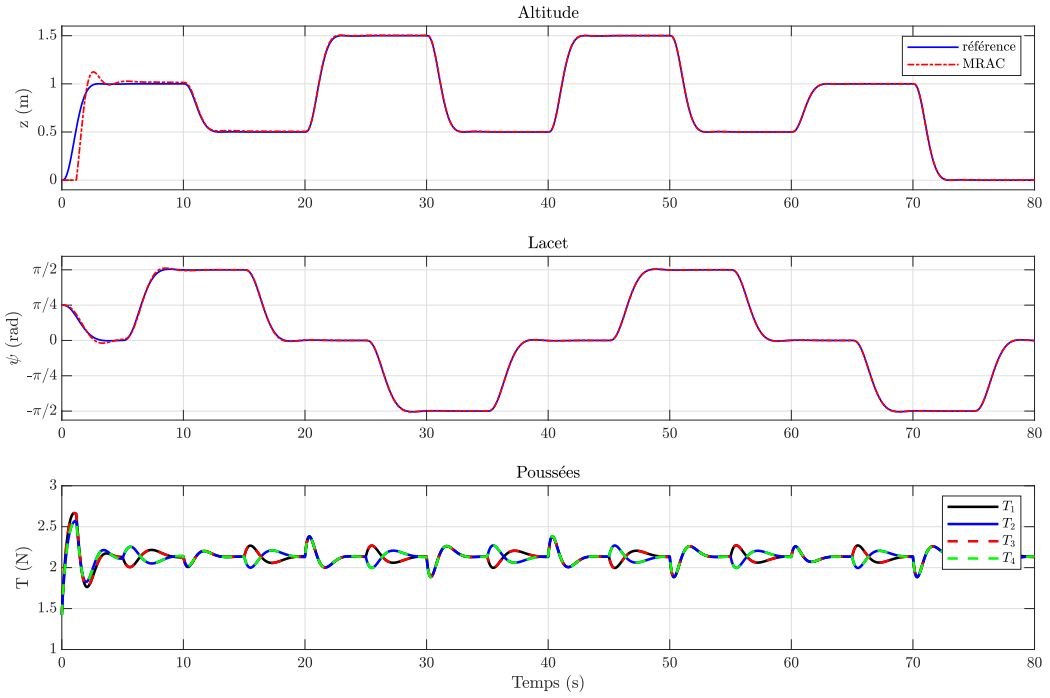


Figure 5.8 Réponse du quadricoptère en boucle fermée en présence de paramètres incertains

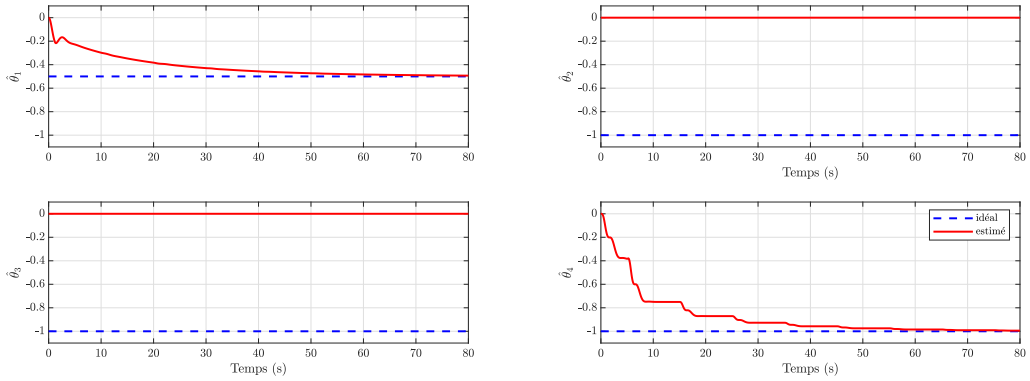


Figure 5.9 Évolution des paramètres adaptatifs pour une série d'échelons de référence

Oscillations transitoires Les matrices $\mathbf{\Gamma}$ et $\mathbf{Q}$ (Eq. (5.36)) sont choisies de façon empirique pour réduire le plus possible les oscillations transitoires souvent rencontrées dans les systèmes MRAC. Pour montrer ce phénomène, on augmente le taux de convergence du paramètre $\hat{\theta}_1$ en utilisant :

$$\mathbf{\Gamma} = \mathbf{I}_4 \quad \text{et} \quad \mathbf{Q} = \text{diag}\left(10\mathbf{I}_3, \mathbf{I}_2, \mathbf{I}_2, 10\mathbf{I}_3\right). \tag{5.40}$$

Les résultats obtenus sont donnés à la figure 5.10. On observe que la réponse temporelle en z est améliorée par rapport à celle de la figure 5.8. Cependant, les oscillations transitoires sur la commande et le paramètre adaptatif $\hat{\theta}_1$ apparaissent. On note également que $\hat{\theta}_1$ converge rapidement vers sa valeur idéale tel qu'attendu.

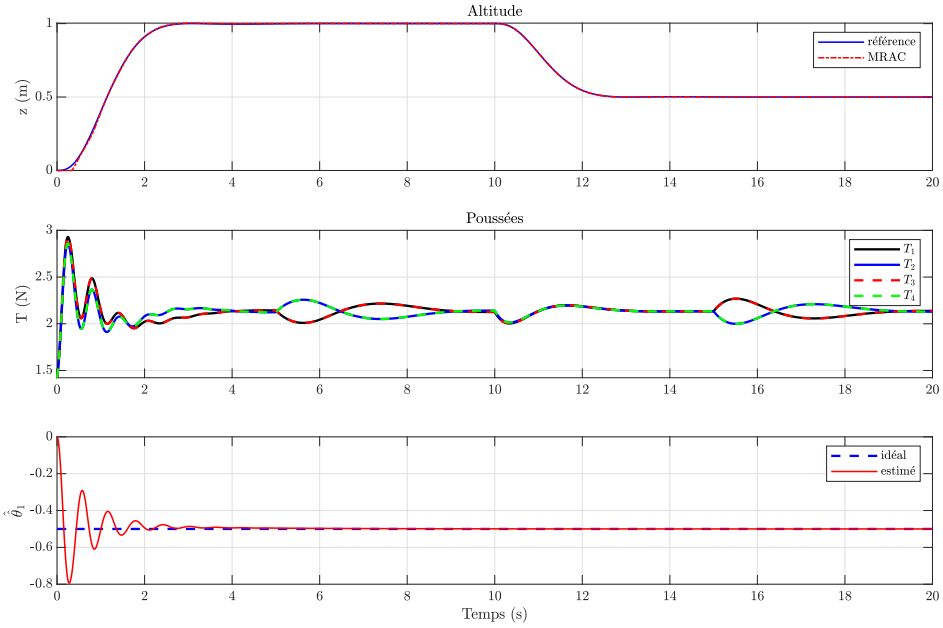


Figure 5.10 Oscillations transitoires

Tolérance aux fautes On vérifie maintenant la capacité à compenser les fautes d'actionneurs du contrôleur générique adaptatif. On reprend ici la trajectoire désirée définie dans l'équation (4.35). Lors de simulation, une perte d'efficacité de 35% est injectée au premier rotor à $t = 30\,\text{s}$. Les paramètres incertains (5.38) sont aussi considérés dans cette simulation. Les réponses spatiales et temporelles du drone sont données aux figures 5.11 et 5.12.

On observe d'abord que le contrôleur nominal (sans auto-séquencement de gains, ni augmentation par MRAC) ne réussit pas à maintenir la stabilité du système en présence de plusieurs incertitudes. Grâce au séquencement de gains par LVE, le contrôleur générique de base (sans augmentation par MRAC) peut conserver la stabilité du drone et rejoindre la trajectoire désirée. De plus, les performances du système sont considérablement améliorées par l'augmentation adaptative. Son efficacité se manifeste par un angle ψ plus petit, une récupération plus rapide des fautes ainsi qu'un meilleur comportement. Par exemple, le cap du drone est dévié de 17° avec le contrôleur générique adaptatif alors que la déviation atteint 36.4° avec le contrôleur générique de base (Fig. 5.12).

La figure 5.13 montre que les signaux de commande du contrôleur générique adaptatif sont encore réalisables et raisonnables. Le profil des paramètres adaptatifs est tracé à la figure 5.14. On mentionne en particulier leur dérive rapide lors de l'apparition des fautes. Par la suite, on cherche à borner ces paramètres avec l'opérateur de projection.

En résumé, le contrôleur générique adaptatif est capable de tolérer les fautes d'actionneurs tout en s'adaptant aux paramètres incertains.

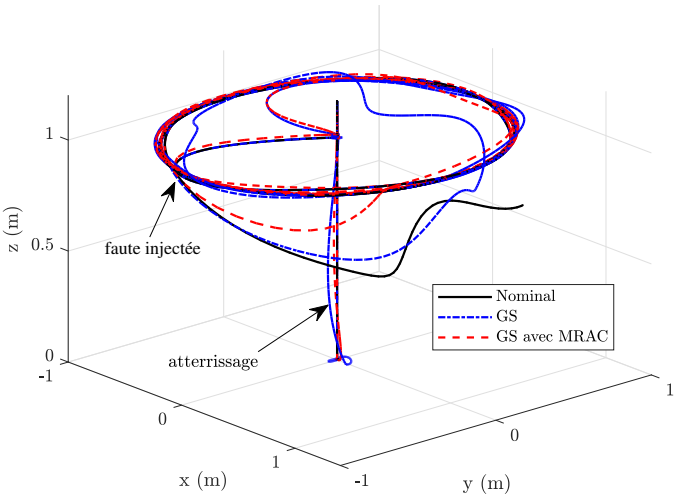


Figure 5.11 Position du quadricoptère en suivi de trajectoire

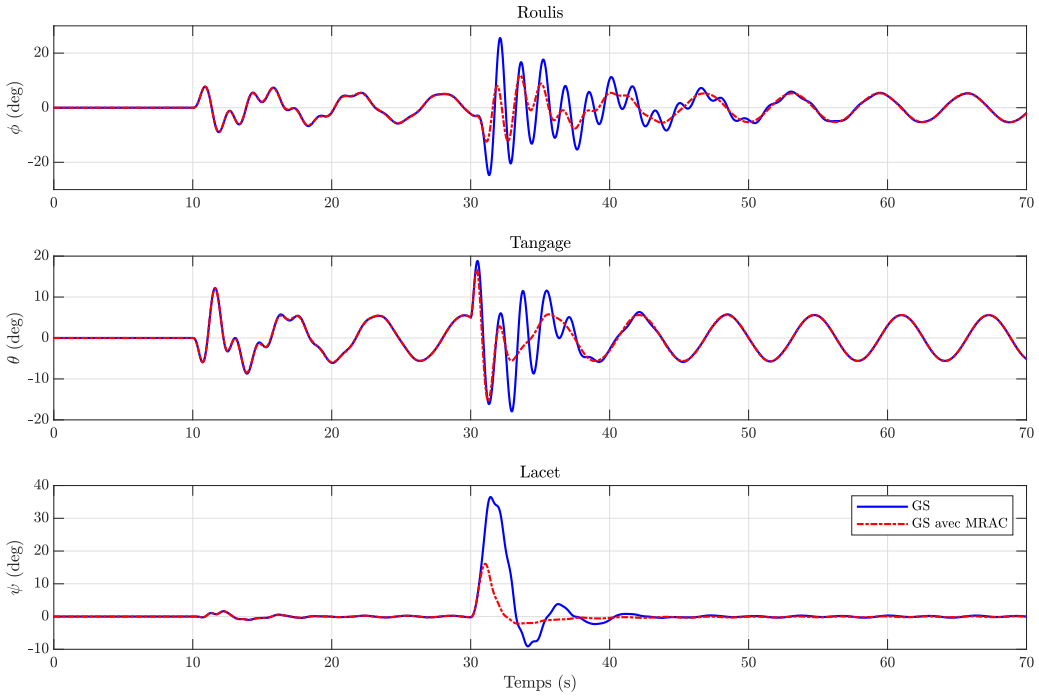


Figure 5.12 Attitude du quadricoptère en suivi de trajectoire

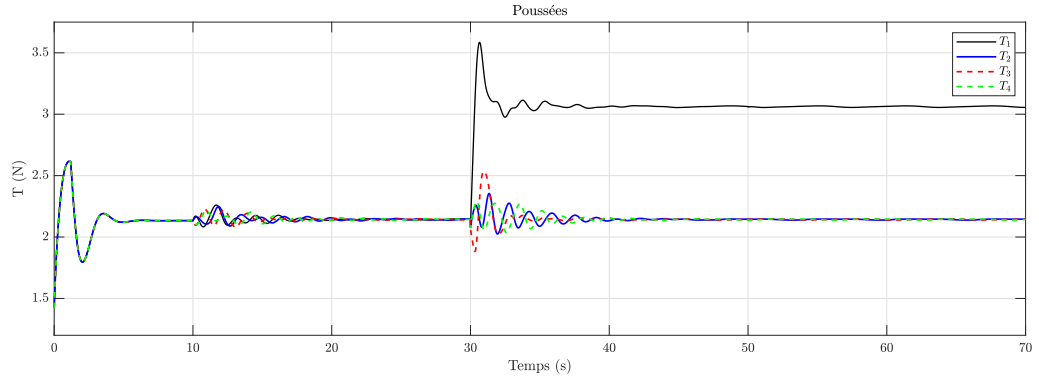


Figure 5.13 Commandes sur les rotors pour un vol en suivi de trajectoire

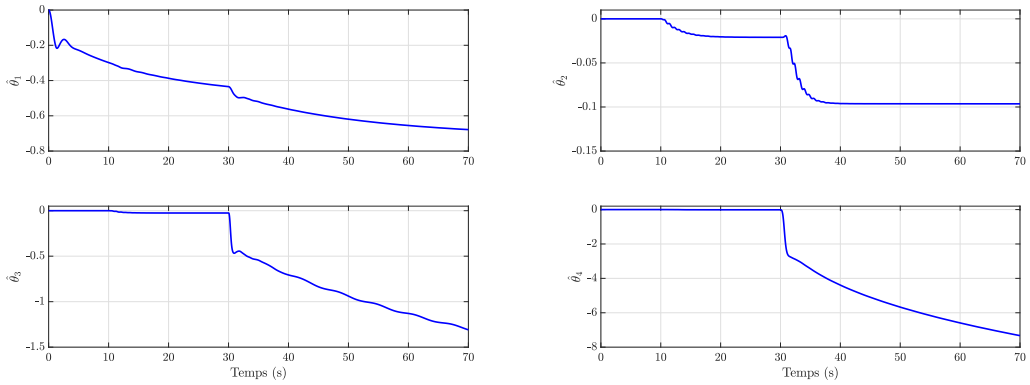


Figure 5.14 Évolution des paramètres adaptatifs pour un vol en suivi de trajectoire

Opérateur de projection On reproduit le même scénario de vol en suivi de trajectoire ci-dessus en appliquant une modification robuste par opérateur de projection. Les bornes maximales pour les paramètres adaptatifs sont données par (5.37). Dans cette simulation, les paramètres $\hat{\theta}_1$ et $\hat{\theta}_2$ n'atteignent jamais leurs limites (Fig. 5.15). Les paramètres $\hat{\theta}_3$ et $\hat{\theta}_4$ sont par ailleurs bien bornés grâce à l'opérateur de projection.

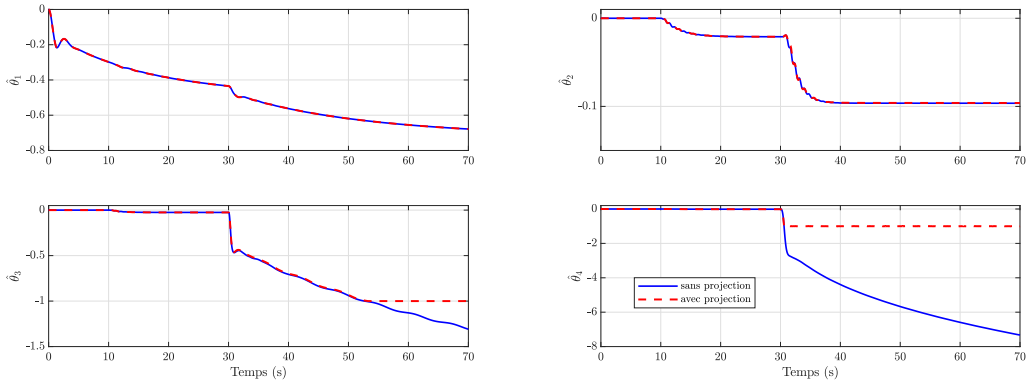


Figure 5.15 Évolution des paramètres adaptatifs avec l'opérateur de projection

Les figures 5.16 et 5.17 démontrent une légère dégradation des performances du quadricoptère en position et en attitude à cause de la saturation des paramètres adaptatifs. Cependant, l'effort de commande (Fig. 5.18) est assez similaire à celui de la simulation précédente (Fig. 5.13). L'opérateur de projection permet ainsi de borner tous les signaux du système en conservant un niveau de performance comparable.

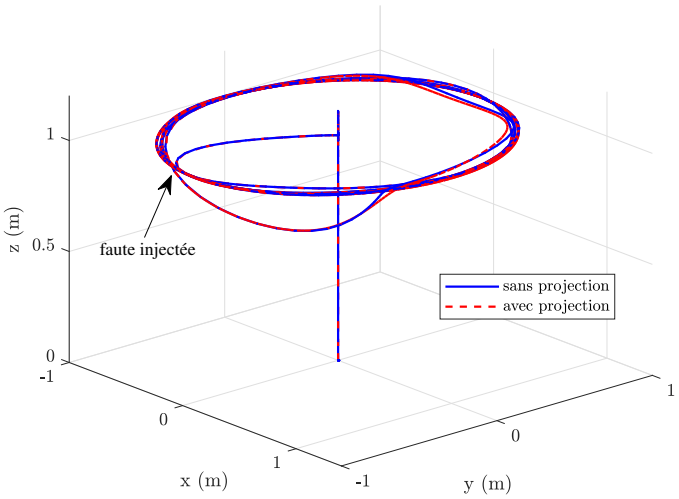


Figure 5.16 Performance en position de la MRAC avec l'opérateur de projection

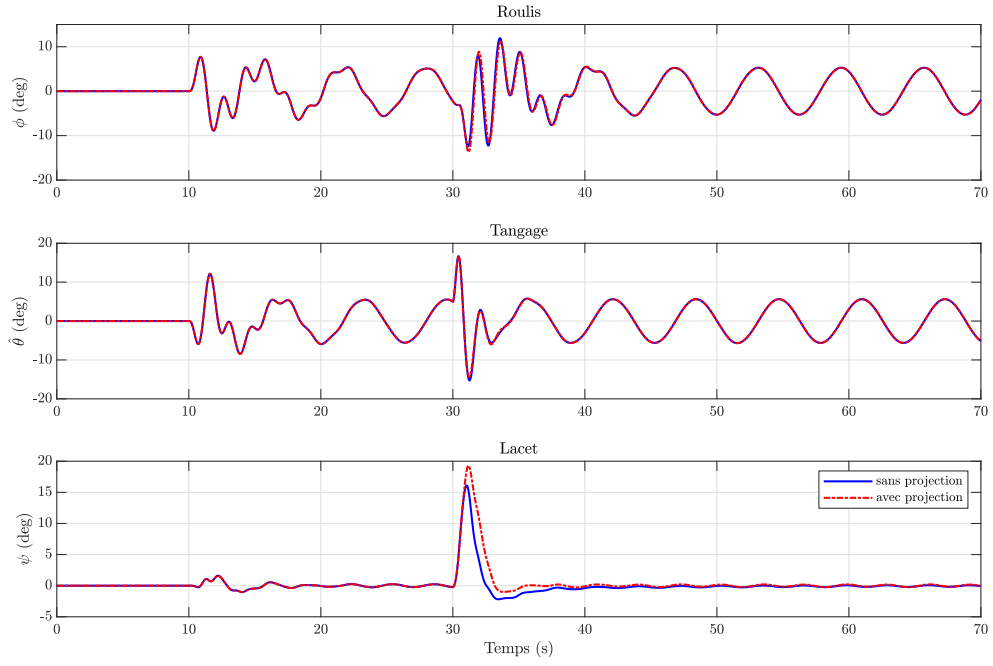


Figure 5.17 Performance en attitude de la MRAC avec l'opérateur de projection

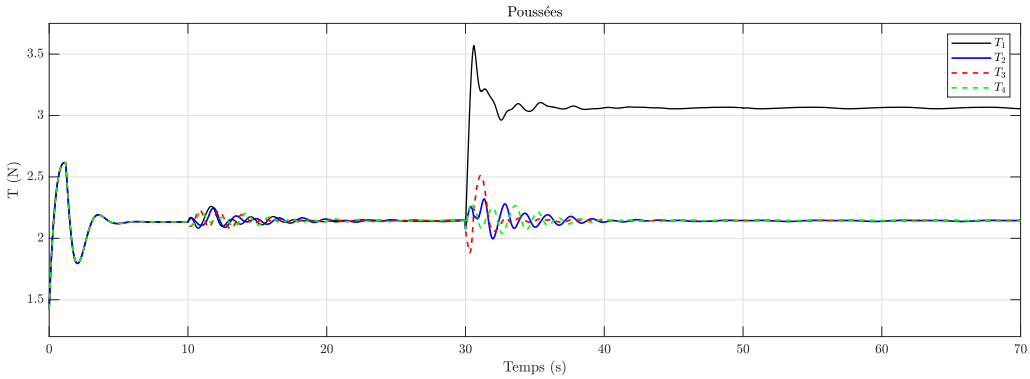


Figure 5.18 Commandes sur les rotors avec l'opérateur de projection

5.3.2 Simulation et tests expérimentaux sur un hexacoptère

On déploie le contrôleur générique adaptatif sur l'hexacoptère Firefly. L'objectif est de valider ce contrôleur dans la mesure où il peut tolérer les fautes sur différents multicoptères.

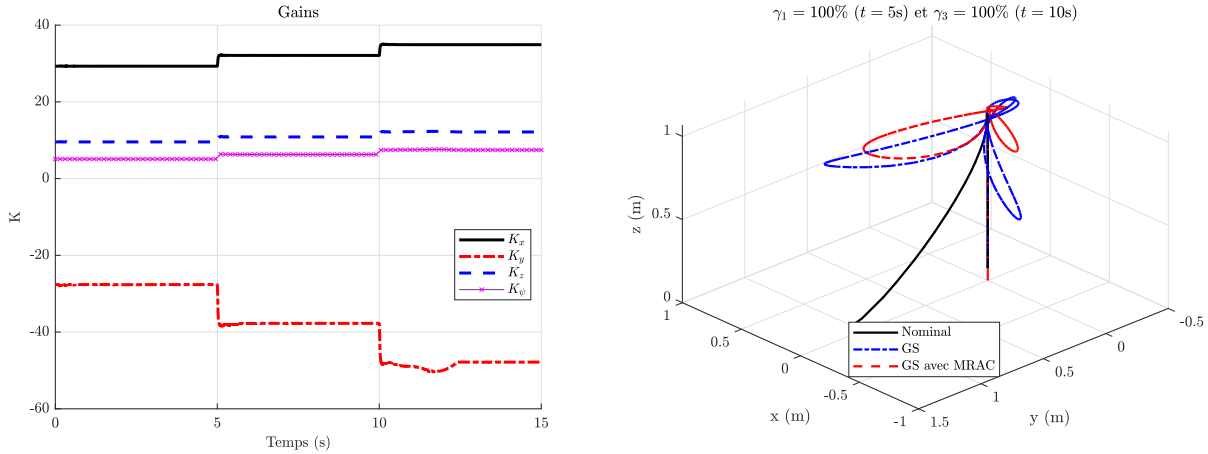
Simulation de pannes de rotors

On utilise RotorS pour simuler l'hexacoptère Firefly (NNPPNP) en maintien de position lorsqu'il y a deux pannes successives de rotors, soit $\gamma_1 = 100\%$ à $t = 5\,\mathrm{s}$ et $\gamma_3 = 100\%$ à $t = 10\,\mathrm{s}$. La figure 5.19a illustre l'évolution des gains du contrôleur générique tolérant aux fautes (auto-séquencement par LVE). Ce profil de gains permet de maintenir la position du drone malgré les défaillances de rotors (Fig. 5.19b, ligne bleue). Ce résultat est assez similaire à celui présenté à la sous-section 4.5.4, car on ne considère pas les paramètres incertains.

La figure 5.19b montre également que la tolérance aux fautes du contrôleur générique est améliorée par l'augmentation MRAC. Puisqu'on ne simule pas les incertitudes non paramétriques mal-assorties $\boldsymbol{\xi}(t)$,[4] les paramètres adaptatifs sont bornés (Fig. 5.20). De plus, on remarque que la composante adaptative n'est active qu'après l'occurrence de fautes. Elle fournit donc une commande supplémentaire au contrôleur générique de base. Par conséquent, le drone revient plus rapidement à sa position en vol stationnaire (Fig. 5.19b).

Il est intéressant de constater que l'effort de commande n'est pas saturé dans ce cas (Fig. 5.21). Les vitesses de rotation désirées sont raisonnables et réalisables par les rotors.[5]

4. Les fautes d'actionneurs peuvent être modélisées dans la matrice inconnue $\boldsymbol{\Lambda}$ par l'équation (4.25).
5. La vitesse de rotation maximale des rotors de l'hexacoptère Firefly est 10000 rpm, cf. Annexe A.2.



(a) Évolution de gains

(b) Position du drone

Figure 5.19 Simulation de la panne de rotors - Contrôleur générique adaptatif

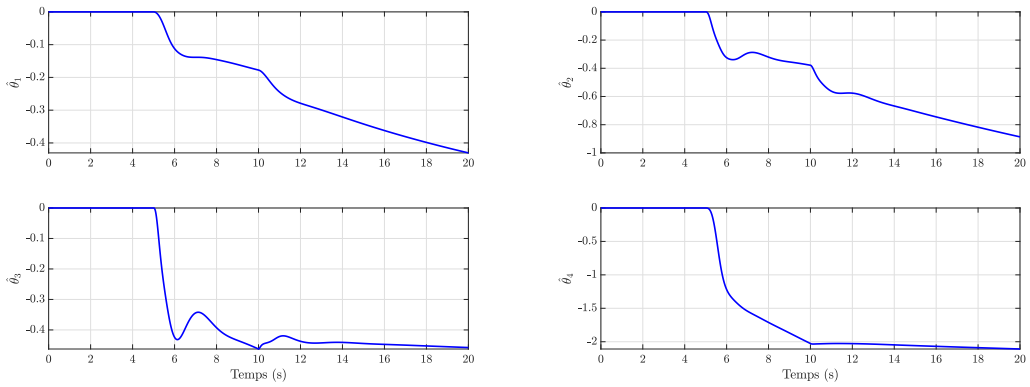


Figure 5.20 Simulation de la panne de rotors - Paramètres adaptatifs

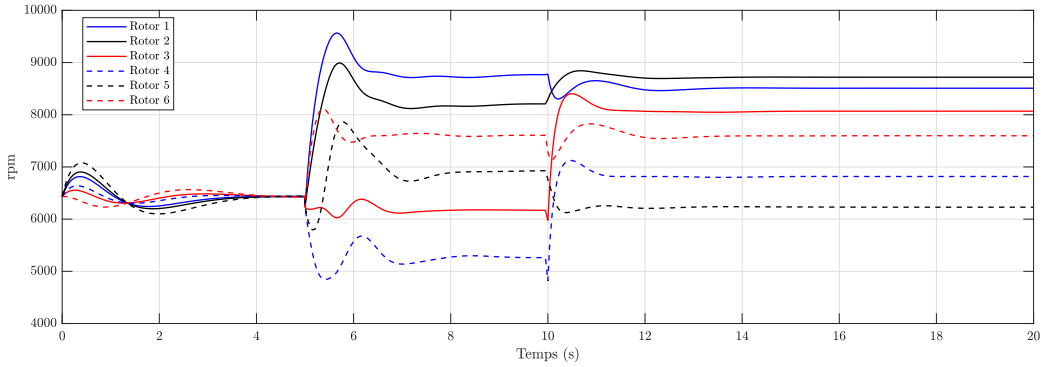


Figure 5.21 Contrôleur générique adaptatif - Commandes sur les rotors

Résultats expérimentaux

La FTC générique par auto-séquencement de gains combinée avec la MRAC étant atteinte en simulation, il reste à la valider expérimentalement. Pour assurer la sécurité de vol [6] et du fait de contraintes de temps, on vérifie seulement la tolérance aux fautes du contrôleur générique sur le Firefly.

On reprend ici le premier scénario de test décrit dans le tableau 4.5 pour un vol stationnaire. Dans ce cas, les valeurs de LAE ($\gamma_1 = 20\%$ et $\gamma_2 = 90\%$) sont précisément calculées à l'aide des mesures fournies par les encodeurs. Par conséquent, on obtient un profil de gains assez lisse (Fig. 5.22a). La figure 5.22b montre qu'il y a un écrasement du drone avec le contrôleur nominal (sans séquencement de gains). Au contraire, le drone est capable de retourner à son point d'équilibre avec le contrôleur générique (GS). Puisque tous les paramètres sont connus, les résultats obtenus sont assez similaires à ceux présentés dans la sous-section 4.5.1.

Les vitesses de rotation envoyées aux contrôleurs de moteur sont données à la figure 5.23. En pratique, on impose une borne de 95% de la vitesse maximale des rotors, soit 9500 rpm. On peut observer que le signal de commande sur le 2^{e} rotor est saturé pendant quelques instants. Cependant, l'influence de cette saturation n'est pas importante puisque le 2^{e} rotor est déjà à la frontière d'une panne totale.

En résumé, les résultats expérimentaux ont permis de montrer que le contrôleur générique fonctionne adéquatement sur l'hexacoptère Firefly dans des conditions idéales (sans incertitudes). De plus, il présente une capacité de tolérance aux fautes considérable.

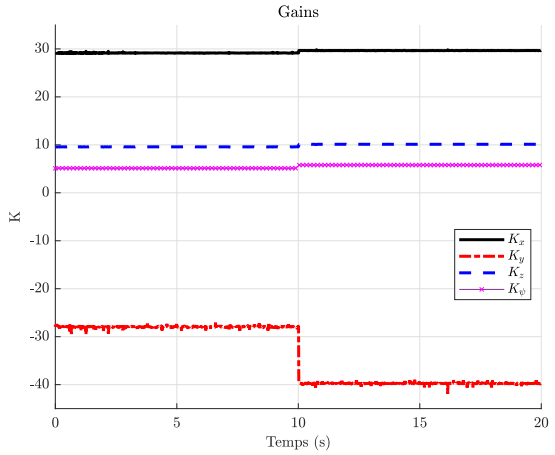
5.4 Conclusion du chapitre

On a d'abord procédé à la synthèse d'un contrôleur auto-séquencé reposant sur la commande virtuelle normalisée. L'objectif est d'obtenir un contrôleur tolérant aux fautes de base qui fonctionne adéquatement sur différents multicoptères dans des conditions idéales. Ensuite, une transformation adaptative de type MRAC est ajoutée pour prendre en compte les incertitudes paramétriques. Enfin, on modifie la loi adaptative par l'opérateur de projection dans le but de borner les gains adaptatifs tout en conservant un niveau de performance acceptable.

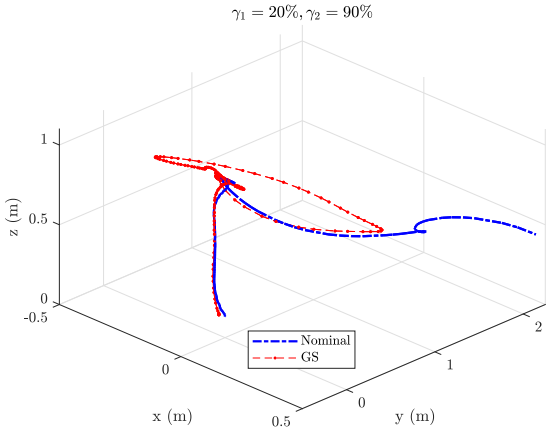
Un contrôleur unique avec des paramètres constants a été implémenté sur deux différents multicoptères. Les résultats obtenus, autant en simulation qu'expérimentalement, ont démontré le bon fonctionnement du système en boucle fermée. Non seulement le contrôleur

6. L'espace de vol dans la volière du MRASL ne permet pas d'effectuer les tests avec plusieurs incertitudes en même temps (fautes et paramètres incertains).

s'adapte aux différentes configurations (avec paramètres inconnus), mais il permet d'améliorer la performance du système en présence de fautes.



(a) Évolution de gains



(b) Position du drone

Figure 5.22 Résultats expérimentaux pour le contrôleur générique tolérant aux fautes

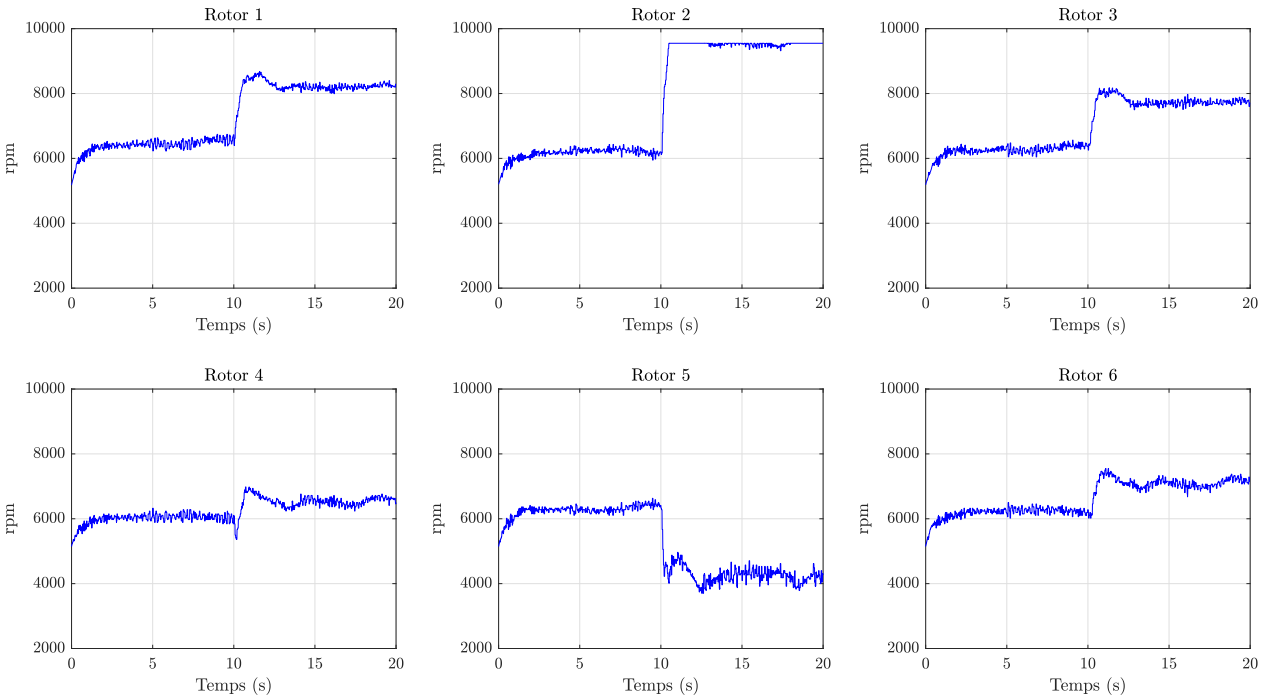


Figure 5.23 Commandes sur les rotors

CHAPITRE 6 ANALYSE DE LA STABILITÉ GÉNÉRALISÉE ROBUSTE DU SYSTÈME FTC PAR APPLICATIONS GARDIENNES

Les chapitres 4 et 5 ont permis de synthétiser avec succès les lois de commande tolérantes aux fautes pour multicoptères. Nous avons ensuite vérifié les résultats de synthèse sur un ensemble de points du domaine de fonctionnement. Toutefois, aucune garantie n'est offerte pour assurer le bon comportement du système bouclé entre deux points de synthèse. Dans le but de valider nos designs, nous allons maintenant étudier la stabilité généralisée robuste du système FTC complet sur l'ensemble des points du domaine incertain.

Parmi les outils d'étude de la stabilité, nous nous intéressons au lien entre les applications gardienne et la VSS proposé par Chen *et al.* [202]. Dans ce travail, les auteurs ont établi les CNS de la stabilité généralisée robuste sous forme de la VSS d'une seule matrice constante. Cela permet d'éviter la complexité du problème d'optimisation de la μ-analyse. Néanmoins, cette méthode présente également quelques limitations, la principale résidant dans le fait qu'elle est applicable seulement pour les matrices à intervalle. De plus, la dimension élevée du problème due à l'utilisation du produit de Kronecker peut être un autre inconvénient. De manière à s'affranchir de ces limitations, nous proposons d'abord une solution simplifiée applicable pour les systèmes incertains possédant une structure d'incertitude affine. Ensuite, cette méthode est étendue au cas d'incertitudes rationnelles.

Pour éviter une charge de calcul élevée due au produit de Kronecker, nous proposons une solution alternative via le recours au produit bialterné. En résumé, la méthode d'analyse proposée présente l'avantage d'être plus simple et plus générale que celle de [202]. Elle rend ainsi possible la résolution numérique des applications gardiennes dans le cas d'une famille incertaine multiparamétrique. Il convient cependant de noter que l'ensemble des incertitudes constitue un hyperrectangle dans notre cas présent. [1]

En nous basant sur les applications gardiennes, nous analyserons la robustesse du système FTC vis-à-vis de la perte d'efficacité de contrôle (LOE),[2] ainsi que de l'erreur d'estimation de fautes sortant du module FDD. Cette phase validera la stratégie de commande tolérante aux fautes par séquencement de gains.

1. On rencontre aussi le cas des polytopes dans la littérature.
2. LOE désigne ici LAE ou LVE.

6.1 Stabilité généralisée robuste du système incertain

Nous commençons cette section par un bref rappel sur la représentation d'un système linéaire incertain et les applications gardiennes. Les autres préliminaires mathématiques tels que la forme LFT, le produit de Kronecker ou le produit bialterné sont disponibles dans l'annexe D. Nous présentons ensuite le lien entre les applications gardiennes et la VSS dans le but de faciliter l'analyse de la robustesse face aux incertitudes structurées bornées dans un hyperrectangle. Les conditions nécessaires et suffisantes de stabilité robuste sous la forme de la VSS d'une matrice constante sont alors proposées. Nous terminons sur des exemples d'application pour montrer l'avantage de cette méthode d'analyse comparativement à celles existantes dans la littérature.

6.1.1 Préliminaires et notations

6.1.1.1 Incertitudes paramétriques

Soit la représentation d'état d'une famille de systèmes linéaires soumis aux perturbations structurées :

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{r})\mathbf{x}(t), \quad \mathbf{r} \in \mathbb{U} \tag{6.1}$$

où $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{r} = [r_1, \ldots, r_k]^\top \in \mathbb{U} \subset \mathbb{R}^k$ est le vecteur de paramètres incertains et $\mathbb{U}$ est un sous-ensemble connexe par arcs (*pathwise connected*) et borné [3] de $\mathbb{R}^k$. La matrice incertaine $\mathbf{A}(\mathbf{r}) \in \mathbb{R}^{n\times n}$ dépend de façon continue du vecteur paramétrique $\mathbf{r}$ et peut être représentée sous la forme suivante :

$$\mathbf{A}(\mathbf{r}) = \mathbf{A}_0 + \sum_{i=1}^{l} h_i(\mathbf{r})\mathbf{A}_i, \quad h_i \neq h_j, \ \forall i \neq j, \ 1 \leq i, j \leq l \tag{6.2}$$

où les $\mathbf{A}_i \in \mathbb{R}^{n\times n}$ sont des matrices constantes et les $h_i : \mathbb{U} \to \mathbb{R}$ sont des fonctions continues de $\mathbf{r}$. Dans cette section, on va étudier en particulier l'hyperrectangle $\mathbb{U} \subset \mathbb{R}^k$ défini par :

$$\mathbb{U} = \left\{ \mathbf{r} \in \mathbb{R}^k : \underline{r_i} \leq r_i \leq \overline{r_i}, 1 \leq i \leq k \right\} \tag{6.3}$$

Définition 6.1.1 ([172])**.** *On dit que la matrice paramétrée* $\mathbf{A}(\mathbf{r})$ *a une structure d'incertitude linéaire, affine, multi-linéaire, polynomiale ou rationnelle si chaque fonction* $h_i : \mathbb{R}^k \to \mathbb{R}$ *est respectivement linéaire, affine, multi-linéaire, polynomiale ou rationnelle de* $\mathbf{r}$*.*

3. Pour l'analyse de la stabilité robuste, on suppose souvent une borne *a priori* pour le vecteur de paramètres incertains $\mathbf{r}$ [172]. Cette condition n'est pas essentielle pour les applications gardiennes, mais nous permet de les connecter avec la VSS.

Notons qu'une matrice à intervalle $\mathbf{A}_I = \left[\underline{\mathbf{A}} \quad \overline{\mathbf{A}}\right]$ est un cas spécial de $\mathbf{A}(\mathbf{r})$ avec une structure d'incertitude affine. Pour obtenir la formulation (6.2), on définit d'abord :

$$\mathbf{A}_0 = \frac{1}{2}(\underline{\mathbf{A}} + \overline{\mathbf{A}}), \quad \hat{\mathbf{A}} = \frac{1}{2}(\overline{\mathbf{A}} - \underline{\mathbf{A}}). \tag{6.4}$$

Soit l le nombre des valeurs non nulles dans $\hat{\mathbf{A}}$, donc $l \leq n^2$. On peut ensuite représenter la matrice $\hat{\mathbf{A}}$ comme $\hat{\mathbf{A}} = \sum_{i=1}^{l} \mathbf{A}_i$ où chaque matrice $\mathbf{A}_i$ a un seul élément non nul et $\mathbf{A}_i \neq \mathbf{A}_j, \forall i \neq j$. Cela permet d'exprimer la matrice à intervalle de la façon suivante :

$$\mathbf{A}_I = \left[\underline{\mathbf{A}} \quad \overline{\mathbf{A}}\right] = \mathbf{A}_0 + \sum_{i=1}^{l} r_i \mathbf{A}_i, \quad -1 \leq r_i \leq 1, \ \forall i = 1, \ \ldots, l. \tag{6.5}$$

On remarque que la définition 6.1.1 est aussi disponible pour les familles de polynômes représentées par $p(s, \mathbf{r}) = p_0(s) + \sum_{i=1}^{l} h_i(\mathbf{r}) p_i(s)$ avec $h_i \neq h_j, \ \forall i \neq j, \ 1 \leq i, j \leq l$.

6.1.1.2 Applications gardiennes

Avant de définir les applications gardiennes, il est nécessaire d'abord de clarifier la définition de la stabilité généralisée robuste [192], aussi appelée la $\mathcal{D}$-stabilité robuste [172].

Définition 6.1.2 ([192]). *Soit Ω un domaine ouvert du plan complexe correspondant à l'ensemble de stabilité désirée. La famille paramétrée de matrices $\mathcal{A} := \left\{\mathbf{A}(\mathbf{r}) : \mathbf{r} \in \mathbb{U} \subset \mathbb{R}^k\right\}$ ou de polynômes $\mathcal{P} := \left\{p(\cdot, \mathbf{r}) : \mathbf{r} \in \mathbb{U} \subset \mathbb{R}^k\right\}$ est robustement Ω-stable si les valeurs propres des matrices $\mathbf{A}(\mathbf{r}) \in \mathcal{A}$ ou les racines des polynômes $p(s, \mathbf{r}) \in \mathcal{P}$ sont toutes à l'intérieur de Ω.*

En particulier, la famille est dite nominalement Ω-stable si $\mathbf{A}(\mathbf{r}_0)$ ou $p(s, \mathbf{r}_0)$ est stable par rapport à Ω pour un certain $\mathbf{r}_0 \in \mathbb{U}$.

On note $\mathcal{S}(\Omega)$, l'ensemble de stabilité généralisée relativement à une région Ω du plan complexe, définie par :

$$\mathcal{S}(\Omega) = \left\{\mathbf{A} \in \mathbb{R}^{n \times n} : \sigma(\mathbf{A}) \subset \Omega\right\} \tag{6.6}$$

pour les matrices, ou

$$\mathcal{S}(\Omega) = \{p \in \mathcal{P}_n : \mathcal{Z}(p) \subset \Omega\} \tag{6.7}$$

pour les polynômes. Ici, $\sigma(\mathbf{A})$ dénote l'ensemble des valeurs propres de $\mathbf{A}$ et $\mathcal{Z}(p)$ dénote l'ensemble des racines de p.

Définition 6.1.3 ([192]). *Soit $\mathcal{X}$ l'ensemble des matrices carrées réelles de dimension n ou l'ensemble des polynômes de degré n à coefficients réels, et $\mathcal{S}$ un sous-ensemble ouvert de $\mathcal{X}$.*

Soit $\nu : \mathcal{X} \to \mathbb{C}$. On dit que ν garde $\mathcal{S}$ si :

$$\forall \mathbf{x} \in \overline{\mathcal{S}} : \nu(\mathbf{x}) = 0 \Longleftrightarrow \mathbf{x} \in \partial\mathcal{S} \tag{6.8}$$

où $\overline{\mathcal{S}}$ est l'adhérence de $\mathcal{S}$ et $\partial\mathcal{S}$ la frontière de $\mathcal{S}$. L'application $\nu(\mathbf{x})$ est dite polynomique si elle est une fonction polynomiale des coefficients de $\mathbf{x}$.

On rappelle que l'adhérence est le plus petit ensemble fermé contenant $\mathcal{S}$. Il existe plusieurs applications gardiennes permettant de garder différentes régions du plan complexe (cf. [192]). Le lecteur peut trouver les applications gardiennes usuelles dans l'annexe D.4.

Définition 6.1.4 ([172])**.** *On dit que l'application $\nu : \mathbb{R}^{n\times n} \to \mathbb{C}$ admet une représentation déterminante polynomiale si :*

$$\nu(\mathbf{A}) = \det f(\mathbf{A}) \tag{6.9}$$

où $f : \mathbb{R}^{n\times n} \to \mathbb{R}^{p\times p}$ telle que $f(\mathbf{A})$ dépend de façon polynomiale des coefficients de $\mathbf{A}$.

Ainsi, chacune des applications gardiennes fournies en annexe D.4 admet une représentation déterminante polynomiale.

Le théorème suivant permet d'établir une CNS de la robustesse en termes de stabilité généralisée à travers l'utilisation des applications gardiennes.

Théorème 6.1.1 ([192])**.** *Soit l'ensemble $\mathcal{S}(\Omega)$ gardé par l'application gardienne ν_Ω et supposons que $\mathbf{x}(\mathbf{r}_0) \in \mathcal{S}$ pour un $\mathbf{r}_0 \in \mathbb{U}$. Alors*

$$\begin{aligned} \mathbf{x}(\mathbf{r}) \in \mathcal{S},\ \forall \mathbf{r} \in \mathbb{U} &\Longleftrightarrow \nu_\Omega\left(\mathbf{x}(\mathbf{r})\right) \neq 0,\ \forall \mathbf{r} \in \mathbb{U} \\ &\Longleftrightarrow \nu_\Omega\left(\mathbf{x}(\mathbf{r}_0)\right)\nu_\Omega\left(\mathbf{x}(\mathbf{r})\right) > 0,\ \forall \mathbf{r} \in \mathbb{U}. \end{aligned} \tag{6.10}$$

On se reportera à [11, 192] pour des exemples d'application du théorème 6.1.1.

Corollaire 6.1.1 ([11])**.** *Soit l'ensemble $\mathcal{S}(\Omega)$ gardé par l'application gardienne ν_Ω. Soit $\mathcal{C} \triangleq \{\mathbf{r} \in \mathbb{U} : \nu_\Omega\left(\mathbf{x}(\mathbf{r})\right) = 0\}$. Alors, l'ensemble $\mathcal{C}$ divise l'espace des paramètres $\mathbb{U} \subset \mathbb{R}^k$ en composantes $\mathcal{C}_i$ qui sont soit stables ou instables par rapport à Ω. Pour savoir si $\mathbf{x}(\mathbf{r})$ est Ω-stable sur $\mathcal{C}_i$, il suffit de tester $\mathbf{x}(\mathbf{r}^i)$ pour un vecteur quelconque $\mathbf{r}^i \in \mathcal{C}_i$.*

Le corollaire des composantes 6.1.1 fournit ainsi une méthode d'analyse graphique qui est utilisable pour les familles monoparamétriques ou biparamétriques.

6.1.1.3 Stabilité généralisée robuste par l'application gardienne et la VSS

La μ-analyse permet d'étudier la stabilité robuste du système vis-à-vis des incertitudes structurées et non structurées qui sont toutes rassemblées dans une matrice Δ. Rappelons que la VSS de la matrice complexe $\mathbf{M} \in \mathbb{C}^{n\times n}$ est définie comme étant [231] :

$$\mu_{\mathbf{\Delta}}(\mathbf{M}) := \frac{1}{\min\limits_{\Delta\in\mathbf{\Delta}} \{\bar{\sigma}(\Delta) : \det(\mathbf{I} - \mathbf{M}\Delta) = 0\}} \tag{6.11}$$

où $\bar{\sigma}(\Delta)$ est la valeur singulière maximum de $\Delta \in \mathbf{\Delta} \subset \mathbb{C}^{n\times n}$. Si $\mathbf{I} - \mathbf{M}\Delta$ n'est pas singulière sur $\mathbf{\Delta}$, alors $\mu_{\mathbf{\Delta}}(\mathbf{M}) := 0$.

La CNS de stabilité robuste établie par la μ-analyse prend la forme :

$$\sup_{\omega\in\mathbb{R}} \mu_{\mathbf{\Delta}}(\mathbf{G}(j\omega)) < \frac{1}{\gamma} \tag{6.12}$$

pour tout $\Delta \in \mathbf{\Delta}$ avec $\|\Delta\|_\infty \leq \gamma$ (cf. [170, 202]). Ainsi, la μ-analyse ramène le problème de stabilité au problème d'optimisation de la VSS d'une matrice de fonctions fréquentielles. Comme il a été mentionné auparavant, la μ-analyse nécessite un effort de calcul conséquent surtout pour les systèmes de grandes dimensions. Afin de pallier ce problème, Chen *et al.* [202] ont développé une méthode alternative basée sur les applications gardiennes et la VSS pour le cas particulier de matrices par intervalles..

Théorème 6.1.2 ([202])**.** *Supposons que la matrice centrale $\mathbf{A}_0 \in \mathbb{R}^{n\times n}$ est stable au sens de Hurwitz, alors le système à intervalle $\dot{\mathbf{x}}(t) = \mathbf{A}_I\mathbf{x}(t) = \mathcal{F}_l(\mathbf{M}, \Delta)\mathbf{x}(t)$ est robustement stable si et seulement si*

$$\mu_{\mathbf{\Delta}_P}(\mathbf{M}_C) < 1 \tag{6.13}$$

où $\mathbf{\Delta}_P = \left\{ \operatorname{diag}(\lambda_1\mathbf{I}_{2n}, \ldots, \lambda_q\mathbf{I}_{2n}) : \lambda_i \in [-1, 1] \right\}$ et $\mathbf{M}_C = \mathbf{P}\hat{\mathbf{M}}_{\mathbf{21}}\hat{\mathbf{M}}_{\mathbf{11}}^{-\mathbf{1}}\hat{\mathbf{M}}_{\mathbf{12}}\mathbf{P}^\top$ est construite en se basant sur une représentation LFT de $\mathbf{A}_I$, l'application gardienne définie avec le produit de Kronecker (D.10), le lemme D.3.2 et une matrice de permutation appropriée $\mathbf{P}$. Pour plus de détails sur la construction de $\mathbf{M}_C$, le lecteur peut se référer à [202].

L'avantage du théorème 6.1.2 est de ne pas nécessiter de recouvrir une plage de fréquence comme dans la condition (6.12). La variable de balayage ω étant éliminée, il faut calculer la VSS d'une seule matrice constante. Cette méthode présente ainsi un atout majeur dans son coût de calcul numérique par rapport à la μ-analyse. Néanmoins, comme mentionnée en introduction, elle a quelques limitations dans sa mise en œuvre.

Dans les sous-sections suivantes, nous présentons une version simplifiée et plus générale du théorème 6.1.2. En analysant la preuve de ce théorème, le point clé de la démonstration

réside dans la représentation LFT de la fonction matricielle $f(\mathbf{A})$ de l'application gardienne choisie (cf. Eq. (6.9)). Nous proposons donc une solution simplifiée quant à la structure d'incertitude affine. Cette solution ne nécessite ni une forme LFT du système paramétré ni une matrice de permutation. Ensuite, nous utilisons les applications gardiennes construites à partir du produit bialterné pour réduire l'effort de calcul. Enfin, nous cherchons à étendre cette méthode dans le cas de la structure d'incertitude rationnelle.

6.1.2 Structure d'incertitude affine

Soit $\mathcal{A} = \left\{\mathbf{A}(\mathbf{r}) : \mathbf{r} \in \mathbb{U} \subset \mathbb{R}^k\right\}$ une famille de matrices réelles de dimension $n \times n$ avec une structure d'incertitude affine, i.e., chaque coefficient de $\mathbf{A}(\mathbf{r})$ dépend de façon affine du vecteur paramétrique $\mathbf{r}$. La matrice paramétrée $\mathbf{A}(\mathbf{r})$ s'écrit alors :

$$\mathbf{A}(\mathbf{r}) = \mathbf{A}_0 + \sum_{i=1}^{k} r_i \mathbf{A}_i \tag{6.14}$$

avec $\mathbf{A}_i$ matrices constantes données. Le paramètre incertain $\mathbf{r}$ est normalisé et varie dans un hyperrectangle $\mathbb{U} \subset \mathbb{R}^k$ défini par :

$$\mathbb{U} = \left\{\mathbf{r} \in \mathbb{R}^k : |r_i| \leq 1, 1 \leq i \leq k\right\}. \tag{6.15}$$

Sans perte de généralité, on suppose que $\mathbf{A}_0$ est stable par rapport à la région d'intérêt Ω.

6.1.2.1 Stabilité d'Hurwitz avec produit de Kronecker

Soit $\Omega = \mathbb{C}^o_-$ le demi-plan gauche complexe ouvert. L'ensemble $\mathcal{S}(\Omega)$ est gardé par l'application gardienne :

$$\nu_H(\mathbf{A}) = \det f(\mathbf{A}) \text{ avec } f(\mathbf{A}) = \mathbf{A} \oplus \mathbf{A}. \tag{6.16}$$

En appliquant les propriétés fondamentales du produit de Kronecker (cf. Annexe D.1) et après manipulations, on obtient :

$$f\left(\mathbf{A}(\mathbf{r})\right) = \mathbf{A}(\mathbf{r}) \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{A}(\mathbf{r}) = \mathbf{A}_0 \oplus \mathbf{A}_0 + \sum_{i=1}^{k} r_i \mathbf{A}_i \oplus \mathbf{A}_i. \tag{6.17}$$

On constate donc que $f\left(\mathbf{A}(\mathbf{r})\right)$ est aussi une matrice paramétrée possédant une structure d'incertitude affine. Dès lors, la technique de Morton [232] peut être appliquée pour obtenir une réalisation LFT d'ordre minimal de $f\left(\mathbf{A}(\mathbf{r})\right)$. En notant $\hat{\mathbf{A}}_i = \mathbf{A}_i \oplus \mathbf{A}_i, i = 1 \ldots k$, on

effectue ensuite une décomposition à rang minimum pour $\hat{\mathbf{A}}_i$ comme suit :

$$\hat{\mathbf{A}}_i = \hat{\mathbf{A}}_{il}\hat{\mathbf{A}}_{ir}$$

où $\hat{\mathbf{A}}_{il} \in \mathbb{R}^{n^2 \times q_i}$ et $\hat{\mathbf{A}}_{ir} \in \mathbb{R}^{q_i \times n^2}$ avec $q_i = \operatorname{rank} \hat{\mathbf{A}}_i$. L'équation (6.17) devient :

$$f\left(\mathbf{A}(\mathbf{r})\right) = \underbrace{\mathbf{A}_0 \oplus \mathbf{A}_0}_{\mathbf{N}_{11}} + \underbrace{\begin{bmatrix}\hat{\mathbf{A}}_{1l} & \dots & \hat{\mathbf{A}}_{kl}\end{bmatrix}}_{\mathbf{N}_{12}} \underbrace{\begin{bmatrix} r_1\mathbf{I}_{q_1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & r_k\mathbf{I}_{q_k}\end{bmatrix}}_{\Delta(\mathbf{r})} \underbrace{\begin{bmatrix}\hat{\mathbf{A}}_{1r} \\ \vdots \\ \hat{\mathbf{A}}_{kr}\end{bmatrix}}_{\mathbf{N}_{21}} =: \mathcal{F}_l\left(\mathbf{N}, \Delta(\mathbf{r})\right). \quad (6.18)$$

avec $\mathbf{N} := \begin{bmatrix}\mathbf{N}_{11} & \mathbf{N}_{12} \\ \mathbf{N}_{21} & \mathbf{0}\end{bmatrix}$.

Corollaire 6.1.2. *La famille de matrices (6.14)-(6.15) est robustement stable au sens de Hurwitz si et seulement si elle est nominalement stable et*

$$\mu_{\mathbf{\Delta}}\left(\mathbf{N}_{21}\mathbf{N}_{11}^{-1}\mathbf{N}_{12}\right) < 1 \quad (6.19)$$

où $\mathbf{\Delta} = \left\{\Delta = \operatorname{diag}\left(r_1\mathbf{I}_{q_1}, \dots, r_k\mathbf{I}_{q_k}\right) \in \mathbb{R}^{q\times q},\ q = q_1 + \dots + q_k\right\}$.

Démonstration. La preuve du corollaire 6.1.2 suit les mêmes lignes de celle du théorème 3 dans [202]. Puisque $\mathbf{A}_0 \in \mathcal{S}(\Omega)$ et ν_H garde $\mathcal{S}(\Omega)$, on a $\nu_H(\mathbf{A}_0) = \det f(\mathbf{A}_0) = \det \mathbf{N}_{11} \neq 0$, c'est-à-dire $\mathbf{N}_{11}$ est inversible. Alors, le théorème 6.1.1 implique que :

$$\begin{aligned}
\mathbf{A}(\mathbf{r}) \in \mathcal{S}(\mathbf{\Omega}),\ \forall \mathbf{r} \in \mathbb{U} &\Leftrightarrow \nu_H\left(\mathbf{A}(\mathbf{r})\right) \neq 0,\ \forall \mathbf{r} \in \mathbb{U} \\
&\Leftrightarrow \det\left[\mathbf{N}_{11} + \mathbf{N}_{12}\Delta\left(\mathbf{I}_q - \mathbf{N}_{22}\Delta\right)^{-1}\mathbf{N}_{21}\right] \neq 0,\ \forall \mathbf{r} \in \mathbb{U} \\
&\Leftrightarrow \det\left[\mathbf{I}_{n^2} + \mathbf{N}_{11}^{-1}\mathbf{N}_{12}\Delta\left(\mathbf{I}_q - \mathbf{N}_{22}\Delta\right)^{-1}\mathbf{N}_{21}\right] \neq 0,\ \forall \mathbf{r} \in \mathbb{U} \\
&\Leftrightarrow \det\left[\mathbf{I}_q + \left(\mathbf{I}_q - \mathbf{N}_{22}\Delta\right)^{-1}\mathbf{N}_{21}\mathbf{N}_{11}^{-1}\mathbf{N}_{12}\Delta\right] \neq 0,\ \forall \mathbf{r} \in \mathbb{U} \\
&\Leftrightarrow \det\left[\mathbf{I}_q - \mathbf{N}_{22}\Delta + \mathbf{N}_{21}\mathbf{N}_{11}^{-1}\mathbf{N}_{12}\Delta\right] \neq 0,\ \forall \mathbf{r} \in \mathbb{U} \\
&\Leftrightarrow \min_{\Delta\in\mathbf{\Delta}}\left\{\overline{\sigma}\left(\Delta\right) : \det\left[\mathbf{I}_q - \left(\mathbf{N}_{22} - \mathbf{N}_{21}\mathbf{N}_{11}^{-1}\mathbf{N}_{12}\right)\Delta\right] = 0\right\} > 1 \\
&\Leftrightarrow \mu_{\mathbf{\Delta}}\left(\mathbf{Q}\right) < 1
\end{aligned}$$

avec $\mathbf{Q} := \mathbf{N}_{22} - \mathbf{N}_{21}\mathbf{N}_{11}^{-1}\mathbf{N}_{12}$. La démonstration est complétée en notant que dans ce cas $\mathbf{N}_{22} = \mathbf{0}$ et que $\mu_{\mathbf{\Delta}}\left(\mathbf{Q}\right) = \mu_{\mathbf{\Delta}}\left(-\mathbf{Q}\right)$.

L'avantage de la formulation du corollaire 6.1.2 comparativement à la formulation du théorème

6.1.2 réside dans le fait qu'il ne faut pas utiliser la représentation LFT de la matrice incertaine $\mathbf{A}(\mathbf{r})$ pour réaliser $f\left(\mathbf{A}(\mathbf{r})\right) = \mathcal{F}_l\left(\mathbf{N}, \Delta(\mathbf{r})\right)$. De plus, on peut assurer que la forme LFT (6.18) dispose d'un ordre minimal donné par :

$$q := \sum_{i=1}^{k} q_i = \sum_{i=1}^{k} \operatorname{rank}\left(\mathbf{A}_i \oplus \mathbf{A}_i\right) \leq kn^2. \tag{6.20}$$

On remarque particulièrement que la structure du bloc d'incertitude Δ (cf. Eq. 6.18) est bien définie pour le calcul de μ. Par conséquent, on n'a besoin d'aucune matrice de permutation comme dans le théorème 6.1.2.

6.1.2.2 Stabilité d'Hurwitz avec produit bialterné

On s'intéresse maintenant à réduire la dimension de $f\left(\mathbf{A}(\mathbf{r})\right) = \mathcal{F}_l\left(\mathbf{N}, \Delta(\mathbf{r})\right)$ via le produit bialterné pour atténuer l'effort de calcul. Considérons alors l'application gardienne

$$\nu_H(\mathbf{A}) = \det(\mathbf{A}) \det(\mathbf{A} \odot \mathbf{I}_n) \tag{6.21}$$

qui protège aussi le demi-plan complexe gauche ouvert (cf. Annexe D.4).

Soient $f_1\left(\mathbf{A}(\mathbf{r})\right) := \mathbf{A}(\mathbf{r})$ et $f_2\left(\mathbf{A}(\mathbf{r})\right) := \mathbf{A}(\mathbf{r}) \odot \mathbf{I}_n$. Il s'agit d'abord de reformuler $f_1\left(\mathbf{A}(\mathbf{r})\right)$ sous la forme d'une LFT inférieure. À cet égard, on applique une nouvelle fois la décomposition à rang minimum pour les matrices constantes $\mathbf{A}_i$ dans l'équation (6.14).

Soit $\mathbf{A}_i = \mathbf{A}_{il}\mathbf{A}_{ir}$, où $\mathbf{A}_{il} \in \mathbb{R}^{n \times m_i}$ et $\mathbf{A}_{ir} \in \mathbb{R}^{m_i \times n}$ avec $m_i = \operatorname{rank} \mathbf{A}_i$. On peut donc exprimer la matrice incertaine $\mathbf{A}(\mathbf{r})$ comme étant :

$$\begin{aligned} f_1\left(\mathbf{A}(\mathbf{r})\right) = \mathbf{A}(\mathbf{r}) &= \mathbf{A}_0 + \sum_{i=1}^{k} r_i \mathbf{A}_i = \mathbf{A}_0 + \sum_{i=1}^{k} r_i \mathbf{A}_{il}\mathbf{A}_{ir} \\ &= \underbrace{\mathbf{A}_0}_{\mathbf{M}_{11}} + \underbrace{\begin{bmatrix} \mathbf{A}_{1l} & \dots & \mathbf{A}_{kl} \end{bmatrix}}_{\mathbf{M}_{12}} \underbrace{\begin{bmatrix} r_1\mathbf{I}_{m_1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & r_k\mathbf{I}_{m_k} \end{bmatrix}}_{\Delta_A(\mathbf{r})} \underbrace{\begin{bmatrix} \mathbf{A}_{1r} \\ \vdots \\ \mathbf{A}_{kr} \end{bmatrix}}_{\mathbf{M}_{21}} =: \mathcal{F}_l\left(\mathbf{M}, \Delta_A(\mathbf{r})\right). \end{aligned} \tag{6.22}$$

On cherche dans un second temps une représentation LFT de $f_2\left(\mathbf{A}(\mathbf{r})\right)$. En exploitant les propriétés de base du produit bialternaté (cf. Annexe D.2), on obtient :

$$f_2\left(\mathbf{A}(\mathbf{r})\right) = \mathbf{A}(\mathbf{r}) \odot \mathbf{I}_n = \mathbf{A}_0 \odot \mathbf{I}_n + \sum_{i=1}^{k} r_i \underbrace{\mathbf{A}_i \odot \mathbf{I}_n}_{\hat{\mathbf{A}}_i}. \tag{6.23}$$

De même, par décomposition à rang minimum, $f_2(\mathbf{A}(\mathbf{r}))$ prend la forme :

$$f_2(\mathbf{A}(\mathbf{r})) = \underbrace{\mathbf{A}_0 \odot \mathbf{I}_n}_{\mathbf{N}_{11}} + \underbrace{\begin{bmatrix}\hat{\mathbf{A}}_{1l} & \dots & \hat{\mathbf{A}}_{kl}\end{bmatrix}}_{\mathbf{N}_{12}} \underbrace{\begin{bmatrix} r_1\mathbf{I}_{q_1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & r_k\mathbf{I}_{q_k}\end{bmatrix}}_{\Delta(\mathbf{r})} \underbrace{\begin{bmatrix}\hat{\mathbf{A}}_{1r} \\ \vdots \\ \hat{\mathbf{A}}_{kr}\end{bmatrix}}_{\mathbf{N}_{21}} =: \mathcal{F}_l(\mathbf{N}, \Delta(\mathbf{r})) \tag{6.24}$$

où $\hat{\mathbf{A}}_i := \mathbf{A}_i \odot \mathbf{I}_n = \hat{\mathbf{A}}_{il}\hat{\mathbf{A}}_{ir}$, $\hat{\mathbf{A}}_{il} \in \mathbb{R}^{\frac{n(n-1)}{2} \times q_i}$, $\hat{\mathbf{A}}_{ir} \in \mathbb{R}^{q_i \times \frac{n(n-1)}{2}}$ avec $q_i = \operatorname{rank} \hat{\mathbf{A}}_i$. L'ordre de cette représentation LFT est minimal et donné par :

$$q := \sum_{i=1}^{k} q_i = \sum_{i=1}^{k} \operatorname{rank}(\mathbf{A}_i \odot \mathbf{I}_n) \leq k\frac{n(n-1)}{2}. \tag{6.25}$$

Sur la base des développements précédents, on en déduit le corollaire suivant.

Corollaire 6.1.3. *La famille de matrices (6.14)-(6.15) est robustement stable (Hurwitz) si et seulement si elle est nominalement stable et*

$$\mu_{\mathbf{\Delta}_A}\left(\mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12}\right) < 1 \quad et \quad \mu_{\mathbf{\Delta}}\left(\mathbf{N}_{21}\mathbf{N}_{11}^{-1}\mathbf{N}_{12}\right) < 1. \tag{6.26}$$

La démonstration du corollaire 6.1.3 est similaire à celle du corollaire 6.1.2.

Remarque 6.1.1. En comparant les équations (6.20) et (6.25), on observe que la dimension du bloc d'incertitude Δ résultant du produit bialterné est inférieure à celle obtenue avec le produit de Kronecker. On note également que les dimensions des matrices carrées $f(\mathbf{A})$, $f_1(\mathbf{A})$ et $f_2(\mathbf{A})$ dans les équations (6.17), (6.22) et (6.24) sont respectivement n^2, n et $n(n-1)/2$. Ainsi, la deuxième méthode basée sur le produit bialterné est préférable en termes de temps de calcul, surtout dans le cas d'un système d'ordre élevé.

Remarque 6.1.2. On peut également adopter une démarche similaire pour les applications gardiennes (D.12) et (D.13) concernant la stabilité généralisée. Notons que les autres applications (D.14-D.16) ne conviennent pas à cause de la non linéarité de la fonction $f(\mathbf{A})$ dans leur formulation. On le reporte à la sous-section suivante.

6.1.3 Structure d'incertitude rationnelle

Plus généralement, nous abordons la stabilité généralisée robuste d'une famille de matrices $\mathcal{A} = \left\{\mathbf{A}(\mathbf{r}) \in \mathbb{R}^{n\times n} : \mathbf{r} \in \mathbb{U} \subset \mathbb{R}^k\right\}$ qui dépend de façon rationnelle du vecteur paramétrique $\mathbf{r} \in \mathbb{U}$, où $\mathbb{U}$ est un hyperrectangle défini par l'équation (6.15). Rappelons qu'il existe toujours

une représentation LFT d'une telle matrice $\mathbf{A}(\mathbf{r})$ portant sur le lemme D.3.3 énoncé en annexe D.3. Soit :

$$\mathbf{A}(\mathbf{r}) = \mathcal{F}_l\left(\mathbf{M}, \Delta_A(\mathbf{r})\right) = \mathcal{F}_l\left(\begin{bmatrix}\mathbf{M}_{11} & \mathbf{M}_{12}\\ \mathbf{M}_{21} & \mathbf{M}_{22}\end{bmatrix}, \Delta_A(\mathbf{r})\right) \tag{6.27}$$

où $\mathbf{M}_{11} \in \mathbb{R}^{n\times n}$, $\mathbf{M}_{12} \in \mathbb{R}^{n\times m}$, $\mathbf{M}_{21} \in \mathbb{R}^{m\times n}$ et $\mathbf{M}_{22} \in \mathbb{R}^{m\times m}$. Le bloc d'incertitude Δ_A appartient à l'ensemble structuré $\mathbf{\Delta}_A$ défini par :

$$\mathbf{\Delta}_A \triangleq \left\{ \operatorname{diag}\left(r_1\mathbf{I}_{m_1}, \ldots, r_k\mathbf{I}_{m_k}\right) : r_i \in [-1, 1]\right\} \subset \mathbb{R}^{m\times m} \tag{6.28}$$

avec $m = \sum_{i=1}^{k} m_i$ qui représente l'ordre de la LFT (6.27).

6.1.3.1 Cas général

Soit Ω un sous-ensemble ouvert du plan complexe tel que $\mathcal{S}(\Omega)$ soit gardé par une application gardienne $\nu_\Omega(\mathbf{A}) = \det f(\mathbf{A})$. Supposons que $\nu_\Omega(\mathbf{A})$ admette une représentation déterminante polynomiale (cf. Définition 6.1.4). Alors, $f(\mathbf{A})$ est aussi une fonction matricielle rationnelle du paramètre incertain $\mathbf{r}$. Il en résulte :

$$f\left(\mathbf{A}(\mathbf{r})\right) = \mathcal{F}_l\left(\mathbf{N}, \Delta(\mathbf{r})\right) = \mathcal{F}_l\left(\begin{bmatrix}\mathbf{N}_{11} & \mathbf{N}_{12}\\ \mathbf{N}_{21} & \mathbf{N}_{22}\end{bmatrix}, \Delta(\mathbf{r})\right) \tag{6.29}$$

où $\mathbf{N}_{11} \in \mathbb{R}^{p\times p}$, $\mathbf{N}_{12} \in \mathbb{R}^{p\times q}$, $\mathbf{N}_{21} \in \mathbb{R}^{q\times p}$ et $\mathbf{N}_{22} \in \mathbb{R}^{q\times q}$. Le bloc d'incertitude $\Delta \in \mathbf{\Delta}$ où

$$\mathbf{\Delta} \triangleq \left\{ \operatorname{diag}\left(r_1\mathbf{I}_{q_1}, \ldots, r_k\mathbf{I}_{q_k}\right) : r_i \in [-1, 1]\right\} \subset \mathbb{R}^{q\times q} \tag{6.30}$$

avec $q = \sum_{i=1}^{k} q_i$ qui représente l'ordre de la LFT (6.29).

Le théorème suivant fournit une CNS de base pour la stabilité généralisée robuste du système linéaire dépendant rationnellement du paramètre incertain réel.

Théorème 6.1.3. *Soient la famille de matrices $\mathcal{A} = \{\mathbf{A}(\mathbf{r}) : \mathbf{r} \in \mathbb{U}\}$ définie par (6.27)-(6.28), et l'ensemble $\mathcal{S}(\Omega)$ gardé par l'application gardienne $\nu_\Omega(\mathbf{A}) = \det f(\mathbf{A})$ où $f(\mathbf{A})$ est donnée par (6.29). Alors, la famille paramétrée $\mathcal{A}$ est robustement Ω-stable si et seulement si elle est nominalement Ω-stable et*

$$\mu_{\mathbf{\Delta}}\left(\mathbf{N_{22}} - \mathbf{N_{21}}\mathbf{N_{11}^{-1}}\mathbf{N_{12}}\right) < 1. \tag{6.31}$$

La démonstration du théorème 6.1.3 est similaire à celle du corollaire 6.1.2.

Remarque 6.1.3. L'ordre de $f\left(\mathbf{A}(\mathbf{r})\right) = \mathcal{F}_l\left(\mathbf{N}, \Delta(\mathbf{r})\right)$ n'est pas essentiellement minimal.

Or, en utilisant la VSS comme indice de stabilité, on devrait réduire autant que possible la dimension de $\mathbf{N}$ et $\Delta(\mathbf{r})$. Cela permet d'obtenir précisément et rapidement la VSS par des outils numériques tels que la commande `mussv` de MATLAB. De la sorte, le produit bialterné est préféré pour la construction d'applications gardiennes plutôt que le produit de Kronecker.

Dans ce qui suit, on va appliquer le théorème 6.1.3 pour différentes régions de stabilité d'intérêt Ω. Il s'agit de trouver l'expression explicite de $f(\mathbf{A}(\mathbf{r}))$ en termes de LFT avec l'ordre le plus bas possible. L'idée est d'utiliser $\mathbf{A}(\mathbf{r}) = \mathcal{F}_l(\mathbf{M}, \Delta_A(\mathbf{r}))$ pour réaliser $f(\mathbf{A}(\mathbf{r})) = \mathcal{F}_l(\mathbf{N}, \Delta(\mathbf{r}))$. Cela découle du fait que l'interconnexion de LFT est aussi une LFT [201].

6.1.3.2 Stabilité d'Hurwitz avec produit de Kronecker

On commence habituellement avec l'application gardienne $\nu_H(\mathbf{A}) = \det(\mathbf{A} \oplus \mathbf{A})$ dans le but d'étudier la stabilité d'Hurwitz. En appliquant le lemme D.3.2, on obtient :

$$f(\mathbf{A}(\mathbf{r})) = \mathbf{A}(\mathbf{r}) \oplus \mathbf{A}(\mathbf{r}) = \mathcal{F}_l(\mathbf{M}, \Delta_A(\mathbf{r})) \oplus \mathcal{F}_l(\mathbf{M}, \Delta_A(\mathbf{r})) =: \mathcal{F}_l(\hat{\mathbf{M}}, \hat{\Delta}(\mathbf{r})) \tag{6.32}$$

où

$$\begin{aligned} \hat{\mathbf{M}} &= \left[\begin{array}{c|cc} \mathbf{M}_{11} \oplus \mathbf{M}_{11} & \mathbf{M}_{12} \otimes \mathbf{I}_n & \mathbf{I}_n \otimes \mathbf{M}_{12} \\ \hline \mathbf{M}_{21} \otimes \mathbf{I}_n & \mathbf{M}_{22} \otimes \mathbf{I}_n & \mathbf{0} \\ \mathbf{I}_n \otimes \mathbf{M}_{21} & \mathbf{0} & \mathbf{I}_n \otimes \mathbf{M}_{22} \end{array}\right] =: \left[\begin{array}{c|c} \hat{\mathbf{M}}_{11} & \hat{\mathbf{M}}_{12} \\ \hline \hat{\mathbf{M}}_{21} & \hat{\mathbf{M}}_{22} \end{array}\right] \\ \hat{\Delta}(\mathbf{r}) &= \begin{bmatrix} \Delta_A(\mathbf{r}) \otimes \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \otimes \Delta_A(\mathbf{r}) \end{bmatrix}. \end{aligned} \tag{6.33}$$

En regardant la structure du bloc d'incertitude $\hat{\Delta}(\mathbf{r})$, on peut trouver que les paramètres incertains ne sont pas bien placés pour calculer la VSS. Par exemple, si $n = k = 2$ et $m = 2$, soit $m_1 = m_2 = 1$ (cf. Eq. (6.28)), on obtient $\hat{\Delta} = \operatorname{diag}(r_1,\ r_1,\ r_2,\ r_2,\ r_1,\ r_2,\ r_1,\ r_2)$ via l'équation (6.33), tandis que la structure requise est $\Delta = \operatorname{diag}(r_1\mathbf{I}_4,\ r_2\mathbf{I}_4)$. On a alors besoin d'une matrice de permutation $\mathbf{P}$ de dimension $2mn \times 2mn$ tel que :

$$\Delta(\mathbf{r}) = \mathbf{P}\hat{\Delta}(\mathbf{r})\mathbf{P}^\top = \operatorname{diag}(r_1\mathbf{I}_{2m_1n}, \ldots, r_k\mathbf{I}_{2m_kn}). \tag{6.34}$$

Par exemple, dans le cas de $n = k = m = 2$, la matrice de permutation $\mathbf{P}$ est donnée par :

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{e}_1\mathbf{e}_1^\top & \mathbf{e}_2\mathbf{e}_1^\top \\ \mathbf{0}_2 & \mathbf{I}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{e}_1\mathbf{e}_2^\top & \mathbf{e}_2\mathbf{e}_2^\top \end{bmatrix} \tag{6.35}$$

où $\{\mathbf{e}_1, \mathbf{e}_2\}$ est la base canonique de $\mathbb{R}^2$.

En introduisant une telle matrice de permutation, on peut reformuler $f(\mathbf{A}(\mathbf{r}))$ comme suit :

$$f(\mathbf{A}(\mathbf{r})) = \mathcal{F}_l(\hat{\mathbf{M}}, \hat{\Delta}(\mathbf{r})) = \mathcal{F}_l\left(\begin{bmatrix} \hat{\mathbf{M}}_{11} & \hat{\mathbf{M}}_{12}\mathbf{P}^\top \\ \mathbf{P}\hat{\mathbf{M}}_{21} & \mathbf{P}\hat{\mathbf{M}}_{22}\mathbf{P}^\top \end{bmatrix}, \Delta(\mathbf{r})\right) =: \mathcal{F}_l(\mathbf{N}, \Delta(\mathbf{r})). \tag{6.36}$$

L'égalité (6.36) est vérifiée en regardant la propriété d'une matrice de permutation $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$. On propose ensuite le corollaire (6.1.4) qui est une conséquence directe du théorème 6.1.3.

Corollaire 6.1.4. *La famille de matrices (6.27)-(6.28) est robustement stable (Hurwitz) si et seulement si elle est nominalement stable et*

$$\mu_{\mathbf{\Delta}}\left(\mathbf{P}\left(\hat{\mathbf{M}}_{22} - \hat{\mathbf{M}}_{21}\hat{\mathbf{M}}_{11}^{-1}\hat{\mathbf{M}}_{12}\right)\mathbf{P}^\top\right) < 1. \tag{6.37}$$

Nous porterons notre attention sur les dimensions de la matrice $f(\mathbf{A}(\mathbf{r}))$ et du bloc Δ qui sont respectivement $p \times p = n^2 \times n^2$ et $q \times q = 2mn \times 2mn$ dans ce cas.

6.1.3.3 Stabilité d'Hurwitz avec produit bialterné

Comme précédemment, on cherche toujours à réduire la dimension de $f(\mathbf{A}(\mathbf{r})) = \mathcal{F}_l(\mathbf{N}, \Delta(\mathbf{r}))$ dans la mesure du possible. Pour cela, on propose d'abord le lemme ci-dessous. On introduit les notations suivantes.

Soient $\{\mathbf{e}_i \in \mathbb{R}^n : 1 \leq i \leq n\}$ la base canonique de $\mathbb{R}^n$ et $\mathcal{Q}_{2,n}$ l'ensemble des paires d'entiers (i, j), $i = 1, \ldots, n-1,\ j = i+1, \ldots, n$, listé lexicalement, c'est-à-dire :

$$\mathcal{Q}_{2,n} = \{(1,2), (1,3), \ldots, (1,n), (2,3), \ldots, (2,n), \ldots, (n-1,n)\}. \tag{6.38}$$

Soit $\mathbf{V}$ une matrice $n^2 \times \frac{n(n-1)}{2}$ dont les colonnes sont données par $\frac{1}{\sqrt{2}}(\mathbf{e}_i \otimes \mathbf{e}_j - \mathbf{e}_j \otimes \mathbf{e}_i)$ et l'ordre des colonnes correspond à l'ordre lexical des paires (i, j) dans $\mathcal{Q}_{2,n}$, soit :

$$\mathbf{V}(:, k) = \frac{1}{\sqrt{2}}(\mathbf{e}_i \otimes \mathbf{e}_j - \mathbf{e}_j \otimes \mathbf{e}_i),\ (i,j) = \mathcal{Q}_{2,n}(k), k = 1, \ldots, \frac{n(n-1)}{2}. \tag{6.39}$$

Lemme 6.1.4. *Soient $\Delta \in \mathbb{C}^{m\times m}$ et $\mathbf{M}$ une matrice complexe de dimension $(n+m) \times (n+m)$, partitionnée avec $\mathbf{M}_{11} \in \mathbb{C}^{n\times n}$, $\mathbf{M}_{22} \in \mathbb{C}^{m\times m}$. Alors,*

$$2\mathcal{F}_l(\mathbf{M}, \Delta) \odot \mathbf{I}_n =: \mathcal{F}_l(\hat{\mathbf{M}}, \hat{\Delta}) \tag{6.40}$$

où

$$\hat{\mathbf{M}} = \left[\begin{array}{c:cc} \mathbf{V}^\top(\mathbf{M}_{11} \oplus \mathbf{M}_{11})\,\mathbf{V} & \mathbf{V}^\top(\mathbf{M}_{12} \otimes \mathbf{I}_n) & \mathbf{V}^\top(\mathbf{I}_n \otimes \mathbf{M}_{12}) \\ \hdashline (\mathbf{M}_{21} \otimes \mathbf{I}_n)\,\mathbf{V} & \mathbf{M}_{22} \otimes \mathbf{I}_n & \mathbf{0} \\ (\mathbf{I}_n \otimes \mathbf{M}_{21})\,\mathbf{V} & \mathbf{0} & \mathbf{I}_n \otimes \mathbf{M}_{22} \end{array}\right] = \left[\begin{array}{c:c} \hat{\mathbf{M}}_{11} & \hat{\mathbf{M}}_{12} \\ \hdashline \hat{\mathbf{M}}_{21} & \hat{\mathbf{M}}_{22} \end{array}\right], \tag{6.41}$$

$$\hat{\Delta} = \left[\begin{array}{cc} \Delta \otimes \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \otimes \Delta \end{array}\right]$$

avec $\mathbf{V}$ *donné par (6.39).*

Démonstration. Pour établir (6.41), on utilise le fait que :

$$2\,(\mathbf{A} \odot \mathbf{B}) = \mathbf{V}^\top\,(\mathbf{A} \otimes \mathbf{B} + \mathbf{B} \otimes \mathbf{A})\,\mathbf{V} \tag{6.42}$$

où $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n\times n}$ et $\mathbf{V} \in \mathbb{R}^{n^2 \times \frac{n(n-1)}{2}}$ définie ci-dessus. On se reportera à [233, Théorème 4.2] pour une démonstration de cette relation entre le produit bialterné et le produit de Kronecker. Avec cette propriété, on en déduit :

$$2\mathcal{F}_l(\mathbf{M}, \Delta) \odot \mathbf{I}_n = \mathbf{V}^\top\,[\mathcal{F}_l(\mathbf{M}, \Delta) \otimes \mathbf{I}_n]\,\mathbf{V} + \mathbf{V}^\top\,[\mathbf{I}_n \otimes \mathcal{F}_l(\mathbf{M}, \Delta)]\,\mathbf{V}. \tag{6.43}$$

En se basant sur les propriétés fondamentales du produit de Kronecker (cf. Annexe D.1), soient $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$ et $(\mathbf{AB}) \otimes (\mathbf{CD}) = (\mathbf{A} \otimes \mathbf{C})\,(\mathbf{B} \otimes \mathbf{D})$, et après quelques manipulations, on dispose du résultat suivant :

$$\begin{aligned} \mathbf{V}^\top\,[\mathcal{F}_l(\mathbf{M}, \Delta) \otimes \mathbf{I}_n]\,\mathbf{V} &= \mathcal{F}_l\left(\begin{bmatrix} \mathbf{V}^\top(\mathbf{M}_{11} \otimes \mathbf{I}_n)\mathbf{V} & \mathbf{V}^\top(\mathbf{M}_{12} \otimes \mathbf{I}_n) \\ (\mathbf{M}_{21} \otimes \mathbf{I}_n)\mathbf{V} & \mathbf{M}_{22} \otimes \mathbf{I}_n \end{bmatrix}, \Delta \otimes \mathbf{I}_n\right), \\ \mathbf{V}^\top\,[\mathbf{I}_n \otimes \mathcal{F}_l(\mathbf{M}, \Delta)]\,\mathbf{V} &= \mathcal{F}_l\left(\begin{bmatrix} \mathbf{V}^\top(\mathbf{I}_n \otimes \mathbf{M}_{11})\mathbf{V} & \mathbf{V}^\top(\mathbf{I}_n \otimes \mathbf{M}_{12}) \\ (\mathbf{I}_n \otimes \mathbf{M}_{21})\mathbf{V} & \mathbf{I}_n \otimes \mathbf{M}_{22} \end{bmatrix}, \mathbf{I}_n \otimes \Delta\right). \end{aligned} \tag{6.44}$$

La preuve du lemme 6.1.4 est complète en notant que $2\mathcal{F}_l(\mathbf{M}, \Delta) \odot \mathbf{I}_n$ est à présent la somme de deux LFT (cf. [201, 234]).

Sur la base de ce lemme, on procède maintenant à l'analyse de robustesse via l'application gardienne suivante :

$$\nu_H(\mathbf{A}(\mathbf{r})) = \det(\mathbf{A})\det(2\mathbf{A} \odot \mathbf{I}_n) = \det\left(\mathcal{F}_l(\mathbf{M}, \Delta_A(\mathbf{r}))\right)\det\left(\mathcal{F}_l(\hat{\mathbf{M}}, \hat{\Delta}(\mathbf{r}))\right) \tag{6.45}$$

qui garde aussi la stabilité d'Hurwitz. En considérant la structure du bloc $\hat{\Delta}(\mathbf{r})$ (Eq. (6.41)), on doit utiliser une matrice de permutation $\mathbf{P}$ similaire à celle dans (6.34) pour replacer les

paramètres incertains à la position désirée. Enfin, on a le résultat suivant.

Corollaire 6.1.5. *La famille de matrices (6.27)-(6.28) est robustement stable (Hurwitz) si et seulement si elle est nominalement stable et*

$$\mu_{\mathbf{\Delta}_A}\left(\mathbf{M}_{22}-\mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12}\right)<1 \quad et \quad \mu_{\mathbf{\Delta}}\left(\mathbf{P}\left(\hat{\mathbf{M}}_{22}-\hat{\mathbf{M}}_{21}\hat{\mathbf{M}}_{11}^{-1}\hat{\mathbf{M}}_{12}\right)\mathbf{P}^{\top}\right)<1. \tag{6.46}$$

Il est en particulier à noter que la dimension de la matrice carrée $\mathcal{F}_l(\hat{\mathbf{M}},\hat{\Delta}(\mathbf{r}))$ dans (6.45) est $n(n-1)/2$, c'est-à-dire une diminution remarquable par rapport à n^2 dans le cas du corollaire 6.1.4 (plus de 50%).

6.1.3.4 Système discret

Soient Ω un disque ouvert de rayon ω et l'ensemble $\mathcal{S}(\Omega)$ gardé par (cf. Annexe D.4) :

$$\nu_\omega(\mathbf{A})=\det f(\mathbf{A})=\det\left(\mathbf{A}\otimes\mathbf{A}-\omega^2\mathbf{I}_{n^2}\right). \tag{6.47}$$

Si $\omega=1$, on retrouve le cas de la stabilité de Schur pour les systèmes discrets. En prenant $\mathbf{A}(\mathbf{r})=\mathcal{F}_l\left(\mathbf{M},\Delta_A(\mathbf{r})\right)$, on peut décomposer $f\left(\mathbf{A}(\mathbf{r})\right)$ sous la forme LFT via le lemme D.3.1 en annexe D.3. On obtient donc :

$$f\left(\mathbf{A}(\mathbf{r})\right)=\mathbf{A}(\mathbf{r})\otimes\mathbf{A}(\mathbf{r})-\omega^2\mathbf{I}_{n^2}=\mathcal{F}_l\left(\hat{\mathbf{M}}_s,\hat{\Delta}_s(\mathbf{r})\right) \tag{6.48}$$

où

$$\begin{aligned}\hat{\mathbf{M}}_s&=\left[\begin{array}{c|cc}\mathbf{M}_{11}\otimes\mathbf{M}_{11}-\omega^2\mathbf{I}_{n^2} & \mathbf{M}_{12}\otimes\mathbf{I}_n & \mathbf{M}_{11}\otimes\mathbf{M}_{12}\\ \hline \mathbf{M}_{21}\otimes\mathbf{M}_{11} & \mathbf{M}_{22}\otimes\mathbf{I}_n & \mathbf{M}_{21}\otimes\mathbf{M}_{12}\\ \mathbf{I}_n\otimes\mathbf{M}_{21} & \mathbf{0} & \mathbf{I}_n\otimes\mathbf{M}_{22}\end{array}\right],\\ \hat{\Delta}_s(\mathbf{r})&=\left[\begin{array}{cc}\Delta_A(\mathbf{r})\otimes\mathbf{I}_n & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_n\otimes\Delta_A(\mathbf{r})\end{array}\right].\end{aligned} \tag{6.49}$$

Dès lors, une démarche similaire à celle du corollaire 6.1.4 sera effectuée pour vérifier la stabilité robuste de $\mathbf{A}(\mathbf{r})$.

6.1.3.5 Stabilité généralisée

En général, il est possible de combiner les applications gardiennes pour protéger un ensemble composé de plusieurs frontières distinctes. [4] Considérons par exemple la région Ω illustrée à

4. cf. Lemme D.4.1 en annexe D.4

la figure 6.1 qui est l'intersection entre le demi-plan gauche ouvert décalé de α et le cône d'amortissement de $\xi = \cos\theta$.

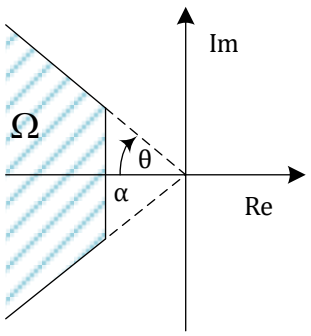


Figure 6.1 Région de stabilité d'intérêt

Une application gardienne correspondante à Ω est donnée par :

$$\nu_\Omega(\mathbf{A}) = \det f_1(\mathbf{A}) \det f_2(\mathbf{A}) \det f_3(\mathbf{A}) \tag{6.50}$$

où

$$\begin{aligned} f_1(\mathbf{A}) &= \mathbf{A} - \alpha\mathbf{I}_n \\ f_2(\mathbf{A}) &= (\mathbf{A} - \alpha\mathbf{I}_n) \odot \mathbf{I}_n = f_1(\mathbf{A}) \odot \mathbf{I}_n, \\ f_3(\mathbf{A}) &= e^{j\theta}\mathbf{A} \ominus e^{-j\theta}\mathbf{A}. \end{aligned} \tag{6.51}$$

Rappelons que $\mathbf{A}(\mathbf{r}) = \mathcal{F}_l(\mathbf{M}, \Delta_A(\mathbf{r}))$. Similairement aux développements précédents, on peut ainsi reformuler $f_1(\mathbf{A}(\mathbf{r})) = \mathcal{F}_l\left(\hat{\mathbf{M}}_1, \hat{\Delta}_1(\mathbf{r})\right)$ avec

$$\hat{\mathbf{M}}_1 = \left[\begin{array}{c|c} \mathbf{M}_{11} - \alpha\mathbf{I}_n & \mathbf{M}_{12} \\ \hline \mathbf{M}_{21} & \mathbf{M}_{22} \end{array}\right], \quad \hat{\Delta}_1(\mathbf{r}) = \Delta_A(\mathbf{r}) \tag{6.52}$$

et $f_3(\mathbf{A}(\mathbf{r})) = \mathcal{F}_l(\mathbf{M}_3, \Delta_3(\mathbf{r}))$ avec

$$\begin{aligned} \hat{\mathbf{M}}_3 &= \left[\begin{array}{c|cc} e^{j\theta}\mathbf{M}_{11} \ominus e^{-j\theta}\mathbf{M}_{11} & e^{j\theta}\mathbf{M}_{12} \otimes \mathbf{I}_n & -e^{-j\theta}\mathbf{I}_n \otimes \mathbf{M}_{12} \\ \hline \mathbf{M}_{21} \otimes \mathbf{I}_n & \mathbf{M}_{22} \otimes \mathbf{I}_n & \mathbf{0} \\ \mathbf{I}_n \otimes \mathbf{M}_{21} & \mathbf{0} & \mathbf{I}_n \otimes \mathbf{M}_{22} \end{array}\right], \\ \hat{\Delta}_3(\mathbf{r}) &= \left[\begin{array}{cc} \Delta_A(\mathbf{r}) \otimes \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \otimes \Delta_A(\mathbf{r}) \end{array}\right]. \end{aligned} \tag{6.53}$$

La représentation $\mathcal{F}_l\left(\hat{\mathbf{M}}_2, \hat{\Delta}_2(\mathbf{r})\right)$ de $f_2(\mathbf{A}(\mathbf{r}))$ est fournie par le lemme 6.1.4. On peut donc

étudier la stabilité robuste de $\mathbf{A}(\mathbf{r})$ par rapport à Ω de manière similaire à celle présentée précédemment en utilisant les matrices de permutation appropriées pour $f_2(\mathbf{A})$ et $f_3(\mathbf{A})$.

Remarque 6.1.4. Il est plus difficile d'obtenir la forme LFT de $f(\mathbf{A}(\mathbf{r}))$ par rapport aux applications gardiennes (D.14) et (D.16) en annexe D.4. On note également que les représentations LFT (6.32) et (6.40) ne garantissent pas en général un ordre minimal. Néanmoins, ces formulations sont explicites et faciles à obtenir. En pratique, peu importe l'application gardienne utilisée, le but est toujours de trouver une réalisation LFT d'ordre le plus petit possible et sans approximation. Parmi les outils numériques disponibles, on utilise la LFR Toolbox [234] pour la mise sous forme LFT et la réduction d'ordre de façon exacte.

6.1.4 Complexité numérique

Cette sous-section vise à comparer les solutions proposées ci-dessus et la solution initiale reportée dans [202] en termes de complexité de calcul. À cette fin, on reproduit le premier exemple de [202] concernant la stabilité d'Hurwitz d'une matrice à intervalle 2×2.[5] De manière à pouvoir appliquer la méthode d'analyse proposée, la matrice incertaine est exprimée sous la forme (6.14) avec $n = 2$ et $k = 4$. Les résultats d'analyse sont résumés dans le tableau 6.1. Notons que seul le temps de calcul de la VSS via la fonction MATLAB `mussv` est reporté.

Tableau 6.1 Résultats d'analyse pour le premier exemple de [202]

Méthode		Dimension de $f(\mathbf{A})$	Dimension de Δ (ordre de LFT)	μ	Temps de calcul
Théorème 6.1.2 [202]		$n^2 = 4$	$kn^2 = 16$	0.9524	0.129 s
Corollaire 6.1.2		$n^2 = 4$	$\sum_{i=1}^{k} \operatorname{rank}(\mathbf{A}_i \oplus \mathbf{A}_i) = 10$	0.8919	0.071 s
Corollaire 6.1.3	$f_1(\mathbf{A})$	$n = 2$	$\sum_{i=1}^{k} \operatorname{rank} \mathbf{A}_i = 4$	0.9815	0.013 s
	$f_2(\mathbf{A})$	$n(n-1)/2 = 1$	$\sum_{i=1}^{k} \operatorname{rank}(\mathbf{A}_i \odot \mathbf{I}_n) = 2$	0.5765	0.007 s

Toutes les méthodes concluent que la matrice à intervalle étudiée est robustement stable (Hurwitz). Du fait des dimensions élevées de $f(\mathbf{A}(\mathbf{r}))$ et de Δ, la première méthode proposée par [202] nécessite plus de calculs que les autres. À l'inverse, le corollaire 6.1.3 est sans doute le moins coûteux en temps de calcul. Cela s'explique par l'avantage du produit bialterné comparativement au produit de Kronecker. Par exemple, si $n = 4$, on mentionne une réduction importante de $n^2 = 16$ (produit de Kronecker) à $n(n-1)/2 = 3$ (produit bialterné). Pour des grandes valeurs de n, le produit bialterné permet de réduire au moins de 50 % le temps de

5. Les calculs sont effectués sur un ordinateur équipé d'un quad-core Intel i5-8250U cadencé à 1.60GHz avec 16 GB RAM et MATLAB R2019b.

calcul. Éventuellement, cette opération peut amener à un meilleur résultat d'analyse comme on le verra dans l'exemple 6.1.5.1. De même, dans le cas de structure d'incertitude rationnelle, le corollaire 6.1.5 demande moins de temps de calcul que le corollaire 6.1.4.

6.1.5 Exemples illustratifs

Dans cette sous-section, quelques exemples déjà étudiés dans la littérature sont reproduits pour montrer l'efficacité et surtout la simplicité de notre approche.

6.1.5.1 Exemple 1

Considérons la famille de matrices qui dépend de façon affine du vecteur incertain $\boldsymbol{\delta} = [\delta_1 \;\; \delta_2]^\top \in \mathbb{R}^2$:

$$\mathbf{A}(\boldsymbol{\delta}) = \begin{bmatrix} -2+\delta_1 & 0 & -1+\delta_1 \\ 0 & -3+\delta_2 & 0 \\ -1+\delta_1 & -1+\delta_2 & -4+\delta_1 \end{bmatrix} \tag{6.54}$$

La stabilité d'Hurwitz de cette famille paramétrée a été étudiée dans plusieurs articles tels que [7,198,235,236]. Le vrai domaine de stabilité calculé algébriquement est donné par ($\delta_1 < 1.75, \delta_2 < 3$). On peut vérifier graphiquement en regardant les courbes de $\nu_1 = \det \mathbf{A}(\boldsymbol{\delta}) = (7-4\delta_1)(\delta_2-3) = 0$ et de $\nu_2 = \det \mathbf{A}(\boldsymbol{\delta}) \odot \mathbf{I} = (\delta_1-3)(2\delta_1\delta_2 - 10\delta_1 - 12\delta_2 + \delta_2^2 + 34) = 0$ données à la figure 6.2 (cf. Corollaire 6.1.1). L'objectif de cet exemple est de montrer les démarches de notre approche d'analyse pour obtenir le même résultat.

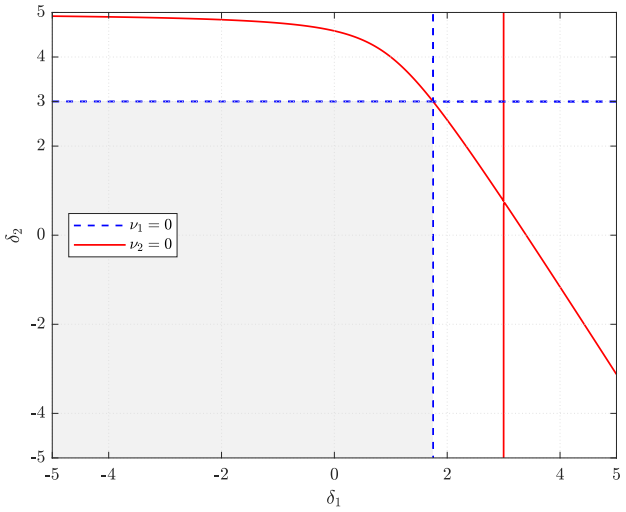


Figure 6.2 Stabilité d'Hurwitz de $\mathbf{A}(\boldsymbol{\delta})$.

Tout d'abord, on vérifie la stabilité d'Hurwitz de $\mathbf{A}(\boldsymbol{\delta})$ pour $(\delta_1, \delta_2) \in \mathbb{U} = [-7, 1] \times [-7, 1]$.

Notons que $\mathbf{A}$ est nominalement stable sur ce rectangle. En appliquant une normalisation standard,[6] on obtient :

$$\mathbf{A}(\mathbf{r}) = \begin{bmatrix} -5 & 0 & -4 \\ 0 & -6 & 0 \\ -4 & -4 & -7 \end{bmatrix} + r_1 \begin{bmatrix} 4 & 0 & 4 \\ 0 & 0 & 0 \\ 4 & 0 & 4 \end{bmatrix} + r_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 4 & 0 \end{bmatrix} \tag{6.55}$$

où $\mathbf{r} = [r_1 \;\; r_2]^\top$ est le nouveau paramètre incertain normalisé variant dans $[-1, 1] \times [-1, 1]$. Considérons l'application gardienne $\nu_H(\mathbf{A}) = \det f(\mathbf{A}) = \det(\mathbf{A} \oplus \mathbf{A})$. La mise sous forme LFT de $f(\mathbf{A}) = \mathcal{F}_l(\mathbf{N}, \Delta)$ est réalisée par l'équation (6.18) où les dimensions de $\mathbf{N}_{11}$ et Δ sont respectivement $n^2 \times n^2 = 9 \times 9$ et $q \times q = 10 \times 10$.[7] Ensuite, la VSS d'intérêt est calculée à l'aide de la fonction MATLAB `mussv` :

$$\mu_{\mathbf{\Delta}}\left(\mathbf{N}_{21}\mathbf{N}_{11}^{-1}\mathbf{N}_{12}\right) \leq 0.8421 < 1.$$

Sur la base du théorème 6.1.3, en notant que $\mathbf{A}(\mathbf{0})$ est Hurwitz, on en déduit que $\mathbf{A}(\boldsymbol{\delta})$ est aussi Hurwitz pour tout $r_1, r_2 \in [-1, 1]$, soit $\delta_1, \delta_2 \in [-7, 1]$.

On utilise maintenant l'application gardienne $\nu_H(\mathbf{A}) = \det(\mathbf{A}) \det(\mathbf{A} \odot \mathbf{I}_n)$ pour résoudre le même problème. Dans ce cas, une réalisation LFT possible de $f_1(\mathbf{A}) = \mathbf{A}$ est donnée par :

$$\mathbf{A} = \mathcal{F}_l(\mathbf{M}, \Delta_A) = \mathcal{F}_l\left(\left[\begin{array}{ccc:cc} -5 & 0 & -4 & 2 & 0 \\ 0 & -6 & 0 & 0 & 2 \\ -4 & -4 & -7 & 2 & 2 \\ \hdashline 2 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \end{array}\right], \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}\right) \tag{6.56}$$

où $r_1, r_2 \in [-1, 1]$. Grâce à (6.24), on obtient ensuite $f_2(\mathbf{A}) = \mathbf{A} \odot \mathbf{I}_n$ sous la forme :

$$\mathbf{A} \odot \mathbf{I}_n = \mathcal{F}_l(\mathbf{N}, \Delta) = \mathcal{F}_l\left(\left[\begin{array}{ccc:cccc} -5.5 & 0 & 2 & -\sqrt{2} & 0 & 2 & 0 \\ -2 & -6 & 0 & 0 & -2 & 2 & 0 \\ 2 & 0 & -6.5 & \sqrt{2} & 0 & 0 & 2 \\ \hdashline -\sqrt{2} & 0 & \sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right], \begin{bmatrix} r_1\mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & r_2\mathbf{I}_2 \end{bmatrix}\right) \tag{6.57}$$

6. La valeur nominale est au centre de l'intervalle.

7. En pratique, on peut également utiliser la LFRT Toolbox [234] ou la fonction `lftdata` de MATLAB pour obtenir automatiquement la même formulation.

Ceci nous permet de calculer les indices de stabilité suivante :

$$\mu_{\boldsymbol{\Delta}_A}\left(\mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12}\right) \leq 0.8421 \quad \text{et} \quad \mu_{\boldsymbol{\Delta}}\left(\mathbf{N}_{21}\mathbf{N}_{11}^{-1}\mathbf{N}_{12}\right) \leq 0.7672.$$

Le corollaire 6.1.3 implique donc la même conclusion sur la stabilité d'Hurwiz de $\mathbf{A}(\boldsymbol{\delta})$. L'avantage ici réside dans la diminution de dimension des LFTs de $(9+10)\times(9+10)$ à $(3+2)\times(3+2)$ et $(3+4)\times(3+4)$.

Les résultats d'autres cas de test sont résumés dans le tableau 6.2.

— L'instabilité de $\mathbf{A}(\boldsymbol{\delta})$ sur $\mathbb{U} = [-2,2]\times[-2,2]$ est démontrée en considérant la borne inférieure de la VSS.

— Dans le cas de $\mathbb{U} = [-10^{-3}, 1.749]\times[-10^{-3}, 2.999]$, la stabilité robuste est vérifiée par $\nu_H(\mathbf{A}) = \det(\mathbf{A})\det(\mathbf{A}\odot\mathbf{I}_n)$, tandis que l'on n'a pas de conclusion en utilisant le produit de Kronecker à cause de la précision de l'algorithme numérique.

— Le rectangle de stabilité le plus large que l'on a obtenu avec le produit bialterné est $[-10^{10}, 1.7499]\times[-10^{6}, 2.9999]$. Dans ce cas, on a effectué une normalisation non standard (cf. [234]) où la valeur nominale du paramètre incertain $\boldsymbol{\delta}$ n'est pas au centre du rectangle mais fixée à l'origine.

Tableau 6.2 Exemple 1

$\mathbb{U}$	Kronecker	Bialterné
$[-2,2]\times[-2,2]$	$1.1144 \leq \mu_{\boldsymbol{\Delta}} \leq 1.1429$ Instable	$1.1429 \leq \mu_{\boldsymbol{\Delta}_A} \leq 1.1429$ $0.8889 \leq \mu_{\boldsymbol{\Delta}} \leq 0.9035$ Instable
$[-10^3, 1.749]\times[-10^3, 2.999]$	$0.9975 \leq \mu_{\boldsymbol{\Delta}} \leq 1.0001$ S.O.	$\mu_{\boldsymbol{\Delta}_A} \leq 0.9999985 < 1$ $\mu_{\boldsymbol{\Delta}} \leq 0.9999987 < 1$ Stable

6.1.5.2 Exemple 2

Considérons la famille de matrices ayant une structure d'incertitude rationnelle [237] :

$$\mathbf{A}(\boldsymbol{\delta}) = \begin{bmatrix} -1 & 2\delta_1 & 2 \\ \delta_2 & -2 & 1 \\ 3 & -1 & \dfrac{\delta_3 - 10}{\delta_1 + 1} \end{bmatrix}$$

où $\delta_1 \in [-0.5, 1]$, $\delta_2 \in [-2, 1]$ et $\delta_3 \in [-0.5, 2]$. Comme reporté dans [237], la nouvelle famille de matrices de type affine obtenue en remplaçant $\delta_4 = (\delta_3 - 10)/(\delta_1 + 1) + 12$ n'est pas stable

(Hurwitz) par la méthode de Lyapunov.[8] On va donc montrer l'avantage des corollaires 6.1.4 et 6.1.5 dans ce cas où la sur-paramétrisation (*overbounding*) a échoué.

La matrice $\mathbf{A}(\mathbf{0})$ est Hurwitz. On commence donc par la LFT de $\mathbf{A}(\boldsymbol{\delta})$:

$$\mathbf{A} = \mathcal{F}_l\left(\mathbf{M}, \Delta_A\right) = \mathcal{F}_l\left(\left[\begin{array}{ccc:ccc:c} -1 & 0.5 & 2 & -2 & 0 & 0 & 0 \\ -0.5 & -2 & 1 & 0 & 0 & -1 & 0 \\ 3 & -1 & -7.4 & 0 & -7.4 & 0 & -1 \\ \hdashline 0 & -0.75 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.6 & 0 & -0.6 & 0 & 0 \\ -1.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 \end{array}\right], \begin{bmatrix} r_1 & 0 & 0 & 0 \\ 0 & r_1 & 0 & 0 \\ 0 & 0 & r_2 & 0 \\ 0 & 0 & 0 & r_3 \end{bmatrix}\right)$$

qui est réalisée, normalisée ($r_1, r_2, r_3 \in [-1, 1]$) et réduite avec la LFR Toolbox. La première VSS d'intérêt est :

$$\mu_{\boldsymbol{\Delta}_A}\left(\mathbf{M}_{22} - \mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12}\right) \geq 4.7360.$$

Ainsi, l'instabilité de $\mathbf{A}(\boldsymbol{\delta})$ est vérifiée sur cet hyperrectangle via le corollaire 6.1.5. En vérifiant la condition (6.37) du corollaire 6.1.4, on obtient la même conclusion. En effet,

$$\mu_{\boldsymbol{\Delta}}\left(\mathbf{P}\left(\hat{\mathbf{M}}_{22} - \hat{\mathbf{M}}_{21}\hat{\mathbf{M}}_{11}^{-1}\hat{\mathbf{M}}_{12}\right)\mathbf{P}^\top\right) \geq 4.7599.$$

De plus, on trouve que $\mathbf{A}(\boldsymbol{\delta})$ est stable sur $\mathbb{U} = [-0.25, 0.25] \times [-1, 0.5] \times [-10, 1]$ puisque :

$$0.9065 \leq \mu_{\boldsymbol{\Delta}}\left(\mathbf{P}\left(\hat{\mathbf{M}}_{22} - \hat{\mathbf{M}}_{21}\hat{\mathbf{M}}_{11}^{-1}\hat{\mathbf{M}}_{12}\right)\mathbf{P}^\top\right) \leq 0.9301.$$

On note également que le temps de calcul de CPU est 0.5 s dans le cas du corollaire 6.1.4 et moins de 0.1 s dans le cas du corollaire 6.1.5.

6.1.5.3 Exemple 3

Considérons la famille de matrices étudiée dans [192] :

$$\mathbf{A}(\delta) = \begin{bmatrix} \delta - 3 & 1 & 2\delta + 1 \\ \delta & -1 & -1 \\ 1 & \delta + 1 & -3 \end{bmatrix}, \quad \delta \in [-0.5, 0.5].$$

8. Via la fonction `quadstab` intégrée dans la LMI Toolbox de MATLAB.

Dans cet exemple, on vérifie la stabilité robuste de $\mathbf{A}(\delta)$ relativement à Ω définie par $\alpha = 1$ et $\theta = \pi/4$ (cf. Fig. 6.1) en utilisant l'application gardienne (6.50). Tout d'abord, en vérifiant les valeurs propres de $\mathbf{A}(0)$, on trouve que $\mathbf{A}$ est nominalement Ω-stable. Ensuite, on réalise et normalise les fonctions matricielles $f_i(\mathbf{A}) = \mathcal{F}_l\left(\mathbf{N}_i, \mathbf{\Delta}_i\right), i \in \{1, 2, 3\}$ dans (6.51) via la LFR Toolbox. Notons que pour cette famille monoparamétrique, aucune matrice de permutation n'est nécessaire si on veut la réaliser manuellement. La famille paramétrée est robustement Ω-stable puisqu'elle est nominalement Ω-stable et :

$$\begin{aligned}
\mu_{\mathbf{\Delta}_1}\left(\mathbf{N}_{1,22} - \mathbf{N}_{1,21}\mathbf{N}_{1,11}^{-1}\mathbf{N}_{1,12}\right) &\leq 0.8447 < 1 \\
\mu_{\mathbf{\Delta}_2}\left(\mathbf{N}_{2,22} - \mathbf{N}_{2,21}\mathbf{N}_{2,11}^{-1}\mathbf{N}_{2,12}\right) &\leq 0.0868 < 1 \\
\mu_{\mathbf{\Delta}_3}\left(\mathbf{N}_{3,22} - \mathbf{N}_{3,21}\mathbf{N}_{3,11}^{-1}\mathbf{N}_{3,12}\right) &\leq 0.5256 < 1.
\end{aligned}$$

On peut également trouver que cette famille n'est pas robustement stable par rapport à $\delta \in [-0.59195, 0.59195]$ à cause de $\mu_{\mathbf{\Delta}_1}\left(\mathbf{N}_{1,22} - \mathbf{N}_{1,21}\mathbf{N}_{1,11}^{-1}\mathbf{N}_{1,12}\right) > 1$. Ce résultat coïncide avec celui reporté dans [192].

6.2 Validation du système FTC par applications gardiennes

Sur la base des applications gardiennes et des développements précédents, on procède enfin à l'analyse de la stabilité généralisée du système FTC proposé dans le chapitre 4.

6.2.1 Système LTI mono-entrée en présence de LOE

Considérons le modèle d'état du système LTI :

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + (1 - \gamma)\mathbf{B}u \tag{6.58}$$

soumis à une perte d'efficacité de contrôle $\gamma \in [0, 1]$. Si l'on applique la loi de commande auto-séquencée par retour d'état, soit $u = -\mathbf{K}(\gamma)\mathbf{x}$, la dynamique du système en boucle fermée est donc caractérisée par $\mathbf{A}_{BF}(\gamma) = \mathbf{A} - (1 - \gamma)\mathbf{B}\mathbf{K}(\gamma)$.

Corollaire 6.2.1. *Soit l'ensemble $\mathcal{S}(\Omega)$ gardé par l'application ν_Ω et supposons que $\mathbf{A}_{BF}(0)$ est stable par rapport à Ω, c'est-à-dire $\mathbf{A}_{BF}(0) \in \mathcal{S}(\Omega)$. Alors, $\mathbf{A}_{BF}(\gamma)$ est stable par rapport à Ω pour tout $\gamma \in [0, \gamma^+)$, où*

$$\gamma^+ \triangleq \min\left\{\gamma \in [0, 1] : \nu_\Omega\left[\mathbf{A}_{BF}(\gamma)\right] = 0\right\} \tag{6.59}$$

ou bien pour tout $\gamma \in [0, 1]$ si γ^+ n'existe pas.

Exemple 6.2.1. Revenons sur le séquencement de gains par LVE abordé dans la section 4.3. En appliquant la loi de commande (4.27)-(4.29) pour le système (4.26), on obtient :

$$\mathbf{A}_{BF}(\gamma_M) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & g & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \alpha(1.34{+}1.2\gamma_M) & \alpha(0.8{+}1.35\gamma_M) & \alpha(2.55{+}6.58\gamma_M) & \alpha(0.45{+}1.7\gamma_M) & \alpha(\text{–}1.09{-}0.16\gamma_M) \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.60}$$

avec $\alpha = (\gamma_M - 1)/I_{yy}$, $g = 9.81$ m/s^2 et $I_{yy} = 0.0459$ kg·m^2 (cf. Annexe A.2). Les valeurs propres de $\mathbf{A}_{BF}(0)$ sont $\{-2.55, -2.22 \pm 3.88j, -1.4 \pm 1.61j\}$. On étudie tout d'abord la stabilité de Hurwitz grâce à l'application gardienne suivante :

$$\begin{aligned} \nu_H &= \det(\mathbf{A}_{BF}) \det(\mathbf{A}_{BF} \odot \mathbf{I}_5) \\ &\propto (\gamma_M - 1)^3(\gamma_M - 1.0009)(\gamma_M - 0.9298)(\gamma_M + 0.1534)(\gamma_M + 0.6562)(\gamma_M + 6.8125) \\ &\quad (\gamma_M^2 + 0.8351\gamma_M + 0.2636) \end{aligned}$$

En prenant $\nu_H = 0$ et en notant que $\mathbf{A}_{BF}(0)$ est Hurwitz, le système bouclé est donc stable pour tout $\gamma_M \in [0, 0.9298)$ selon le corollaire 6.2.1. Dans le cas du contrôleur nominal, soit $\mathbf{K}_x = \mathbf{K}_{x_0}$ et $K_{ix} = K_{ix_0}$ (cf. Eq. (4.28)), l'intervalle de stabilité est réduit à $[0, 0.5758)$ du fait que $\nu_H \propto (\gamma_M - 1)^3(\gamma_M - 1.0109)(\gamma_M - 0.5758)$.

On considère ensuite le domaine Ω du plan complexe tel que montré sur la figure 6.1 avec $\alpha = -0.5$ et $\xi = \cos(\theta) = 0.4$. Il est possible de vérifier que les valeurs propres de $\mathbf{A}_{BF}(0)$ demeurent à l'intérieur de Ω. Une application gardienne correspondante à Ω est obtenue par la multiplication des deux applications (D.12) et (D.14), soit $\nu_\Omega = \nu_\alpha \nu_\xi$ où :

$$\begin{aligned} \nu_\alpha &= \det(\mathbf{A}_{BF} - \alpha\mathbf{I}) \det\left(\mathbf{A}_{BF} \odot \mathbf{I} - \alpha\mathbf{I} \odot \mathbf{I}\right) \\ &\propto (\gamma_M - 3.3641)(\gamma_M - 1.0420)(\gamma_M - 1.0007)(\gamma_M - 0.9996)(\gamma_M - 0.9962)(\gamma_M - 0.8974) \\ &\quad (\gamma_M + 0.1250)(\gamma_M + 1.3364)(\gamma_M^2 + 0.3846\gamma_M + 0.2815), \\ \nu_\xi &= \det(\mathbf{A}_{BF}) \det\left(\mathbf{A}_{BF}^2 \odot \mathbf{I} + (1 - 2\xi^2)\mathbf{A}_{BF} \odot \mathbf{A}_{BF}\right) \\ &\propto (\gamma_M - 1)^6(\gamma_M - 1.0943)(\gamma_M - 0.9866)(\gamma_M - 0.8332)(\gamma_M + 0.0308)(\gamma_M + 0.2667)(\gamma_M + \\ &\quad 6.8125)(\gamma_M^2 + 1.2132\gamma_M + 0.3693)(\gamma_M^2 + 0.9305\gamma_M + 0.3334)(\gamma_M^2 - 0.0406\gamma_M + 0.1199) \end{aligned}$$

En prenant $\nu_\Omega = 0$, on en déduit que le système en boucle fermée avec l'auto-séquencement

de gains (4.29) est stable par rapport à Ω pour tout $\gamma_M \in [0, 0.8332)$. On note également que le contrôleur nominal ne peut maintenir le système Ω-stable que pour $\gamma_M \in [0, 0.1151)$.

Vu le degré de l'application gardienne polynomique dans l'exemple ci-dessus qui est relativement élevé, on propose le corollaire suivant qui ne nécessite pas un tel développement polynomial. Pour la suite, $\mathcal{F}(\cdot)$ dénote la somme de Kronecker, soit $\mathcal{F}(\mathbf{A}) = \mathbf{A} \oplus \mathbf{A}$.

Corollaire 6.2.2. *Soit le système LTI (6.58) et la loi de commande par séquencement de gains* $u = -(\mathbf{K}_0 + \mathbf{K}_1\gamma)\mathbf{x}$. *Supposons que* $\mathbf{A}_{BF}(0) = \mathbf{A} - \mathbf{B}\mathbf{K}_0$ *est Hurwitz. Alors,* $\mathbf{A}_{BF}(\gamma)$ *est robustement stable (Hurwitz) pour* $\gamma \in [0, \gamma^+)$, *où*

$$\gamma^+ \triangleq \min \left\{ \frac{1}{\lambda(\mathbf{M})} : \lambda(\mathbf{M}) \geq 1 \right\} \tag{6.61}$$

ou bien pour tout $\gamma \in [0, 1]$ *si* γ^+ *n'existe pas. Ici,* $\lambda(\mathbf{M})$ *désigne les valeurs propres réelles de la matrice* $\mathbf{M}$ *définie par :*

$$\mathbf{M} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{F}_0^{-1}\mathbf{F}_2 & -\mathbf{F}_0^{-1}\mathbf{F}_1 \end{bmatrix} \tag{6.62}$$

où $\mathbf{F}_0 := \mathcal{F}(\mathbf{A} - \mathbf{B}\mathbf{K}_0)$, $\mathbf{F}_1 := \mathcal{F}\left(\mathbf{B}(\mathbf{K}_0 - \mathbf{K}_1)\right)$ *et* $\mathbf{F}_2 := \mathcal{F}(\mathbf{B}\mathbf{K}_1)$.

Ce résultat est une conséquence directe de [192, Théorème 1]. La démonstration est disponible dans l'annexe D.5.

Exemple 6.2.2. On reprend ici le même exemple étudié ci-dessus. L'ensemble des valeurs propres réelles de la matrice $\mathbf{M}$ est $\{-6.5181, -1.5239, -0.1468, 0, 0.9991, 1, 1.0755\}$. Via le corollaire 6.2.2, on obtient $\gamma^+ = 1/1.0755 = 0.9298$. Notons que si $\mathbf{K}_1 = \mathbf{0}$, on peut simplement choisir $\mathbf{M} = -\mathbf{F}_0^{-1}\mathbf{F}_1$. Dans ce cas, les valeurs propres réelles de $\mathbf{M}$ sont $\{0, 0.9892, 1, 1.7367\}$. On a donc $\gamma^+ = 1/1.7367 = 0.5758$. Ces résultats sont identiques à ceux obtenus pour l'exemple 6.2.1.

6.2.2 Biais de FDD

En pratique, puisque le retour d'état se fait sur la valeur estimée de LOE, le biais du module FDD peut éventuellement dégrader la robustesse du système en termes de stabilité ou de performance. Soit la dynamique longitudinale du multicoptère étudiée dans l'exemple 6.2.1. La loi de commande par retour d'état est :

$$u = -\mathbf{K}(\hat{\gamma}_M)\mathbf{x} = -\mathbf{K}(\gamma_M + \Delta\gamma_M)\mathbf{x} \tag{6.63}$$

où $\hat{\gamma}_M$ est la valeur estimée de LOE et $\Delta\gamma_M$ représente le biais de FDD. La structure de séquencement de gains $\mathbf{K}(\hat{\gamma}_M)$ est donnée à (4.28)-(4.29). Alors, la dynamique du système en boucle fermée est caractérisée par :

$$\mathbf{A}_{BF}(\gamma_M, \hat{\gamma}_M) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & g & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \alpha(1.34{+}1.2\hat{\gamma}_M) & \alpha(0.8{+}1.35\hat{\gamma}_M) & \alpha(2.55{+}6.58\hat{\gamma}_M) & \alpha(0.45{+}1.7\hat{\gamma}_M) & \alpha(-1.09{-}0.16\hat{\gamma}_M) \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.64}$$

où $\alpha = (\gamma_M - 1)/I_{yy}$.

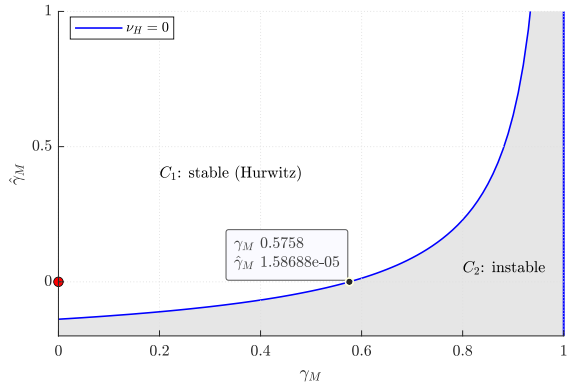
6.2.2.1 Analyse graphique

On vérifie tout d'abord la stabilité d'Hurwitz du système bouclé avec l'application gardienne $\nu_H(\gamma_M, \hat{\gamma}_M) = \det(\mathbf{A}_{BF}) \det(\mathbf{A}_{BF} \odot \mathbf{I}_5)$. Sur la base du corollaire des composantes 6.1.1, les points d'annulation de ν_H divisent l'espace des paramètres $\mathbb{U} = [0, 1] \times [0, 1]$ en composantes stables ou instables. Pour visualiser les courbes qui annulent l'application gardienne, on utilise la fonction MATLAB `fimplicit` (Fig. 6.3a). La région de stabilité est donc $\mathcal{C}_1$ qui contient l'origine. De même, si l'on s'intéresse à l'influence du biais de FDD, soit $\Delta\gamma_M$, la figure 6.3b illustre les courbes où $\nu_H(\gamma_M, \Delta\gamma_M)$ s'annule.

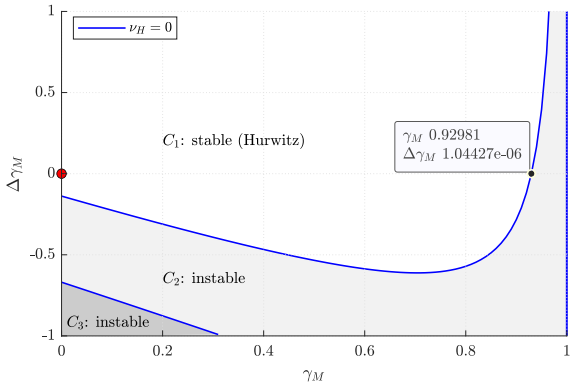
En prenant $\hat{\gamma}_M = 0$, on retrouve la valeur limite $\gamma_M^+ = 0.5758$ avec laquelle le contrôleur nominal peut maintenir le système stable (Fig. 6.3a). En prenant $\Delta\gamma_M = 0$, on observe que le contrôleur séquencé sans biais de FDD peut garder la stabilité du système jusqu'à $\gamma_M^+ = 0.9298$ (Fig. 6.3b). Ces résultats sont identiques à ceux obtenus précédemment.

De façon similaire, on procède à une analyse graphique de la stabilité généralisée par rapport à la région Ω définie à l'exemple 6.2.1. On recourt une nouvelle fois à l'application gardienne construite par la multiplication des deux autres (Eqs. (D.12) et (D.14)), soit $\nu_\Omega = \nu_\alpha \nu_\xi$.

Comme illustré à la figure 6.4a, la fonction `fimplicit` n'est pas capable de détecter les lignes de niveau 0 de ν_ξ dans la région $[0, 1] \times [0, 1]$. Cela s'explique par la complexité relative de la fonction polynomiale ν_ξ (cf. Eq. (D.14)) et la résolution numérique de `fimplicit`. Dans ce cas, on effectue manuellement une analyse des racines du polynôme ν_ξ. La figure 6.4b illustre graphiquement le résultat obtenu où l'on trouve la région $\mathcal{C}_1$ qui satisfait les deux critères à la fois. On peut ainsi voir que le système est stable par rapport au demi-plan gauche décalé de $\alpha = -0.5$ pour tout $(\gamma_M, \hat{\gamma}_M)$ qui résident dans l'union de $\mathcal{C}_1$ et $\mathcal{C}_2$.

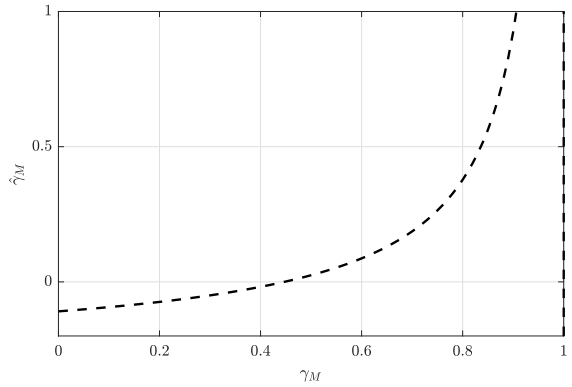


(a) $\nu_H(\gamma_M, \hat{\gamma}_M) = 0$ détecté par `fimplicit`

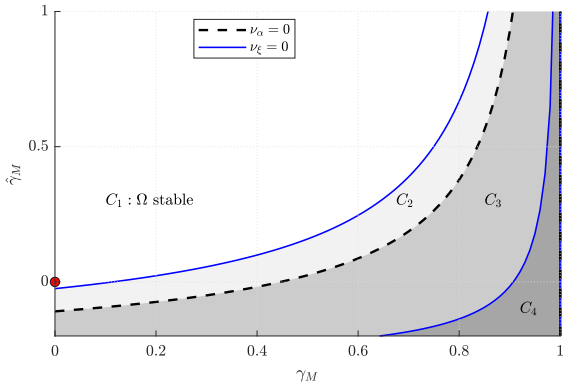


(b) $\nu_H(\gamma_M, \Delta\gamma_M) = 0$ détecté par `fimplicit`

Figure 6.3 Stabilité d'Hurwitz



(a) $\nu_\Omega = 0$ détecté par `fimplicit`



(b) $\nu_\Omega = 0$ détecté manuellement

Figure 6.4 Stabilité généralisée

6.2.2.2 Analyse numérique

En général, l'analyse graphique n'est pas aisée pour les systèmes d'ordre élevé ou les familles dépendant de plusieurs paramètres incertains. On passe donc à la méthode numérique présentée dans la section 6.1. Quelques résultats d'analyse sont résumés dans le tableau 6.3. Pour tous les cas, le résultat obtenu coïncide avec celui vérifié graphiquement ci-dessus.

On mentionne en particulier que, pour $(\gamma_M, \hat{\gamma}_M) \in [0, 0.2] \times [0, 1]$, la méthode numérique confirme que le système bouclé n'est pas robustement Ω-stable. Ceci n'est pas détecté par la méthode graphique via la fonction MATLAB `fimplicit` (Fig. 6.4a).

Tableau 6.3 Analyse numérique

	$(\gamma_M, \hat{\gamma}_M) \in \mathbb{U}$	VSS
Stabilité d'Hurwitz	$[0, 0.4] \times [0, 1]$	$\mu_{\mathbf{\Delta}_A} < 0.25$ et $\mu_{\mathbf{\Delta}} < 0.9$: stable
(cf. Corollaire 6.1.5)	$[0, 0.8] \times [0, 1]$	$\mu_{\mathbf{\Delta}_A} < 0.67$ et $\mu_{\mathbf{\Delta}} > 1.23$: instable
Stabilité généralisée	$[0, 0.2] \times [0, 1]$	$\mu_{\mathbf{\Delta}_3} > 1.02$: Ω-instable
(cf. Éq. (6.50))	$[0.2, 0.4] \times [0.2, 1]$	$\mu_{\mathbf{\Delta}_1} < 0.16, \mu_{\mathbf{\Delta}_2} < 0.68, \mu_{\mathbf{\Delta}_3} < 0.85$: Ω-stable

6.2.3 Système LTI multi-entrée en présence de LOE

On considère le séquencement de gains par LAE synthétisé dans la section 4.2. La dynamique longitudinale du multicoptère en boucle fermée est représentée par :

$$\mathbf{A}_{BF} = \mathbf{A}_a - \mathbf{B}_a \mathbf{\Lambda} \mathbf{K}_a(\gamma_1, \gamma_3, \gamma_4, \gamma_6) \tag{6.65}$$

où $(\mathbf{A}_a, \mathbf{B}_a)$ est le modèle d'état de la boucle ouverte augmenté avec l'erreur de suivi en x, $\mathbf{\Lambda} = \text{diag}\,(1-\gamma_1, 1-\gamma_3, 1-\gamma_4, 1-\gamma_6)$ et $\mathbf{K}_a = [\mathbf{K} \ -\mathbf{K}_i]$ donné par (4.14)-(4.16) avec les valeurs dans (4.20).

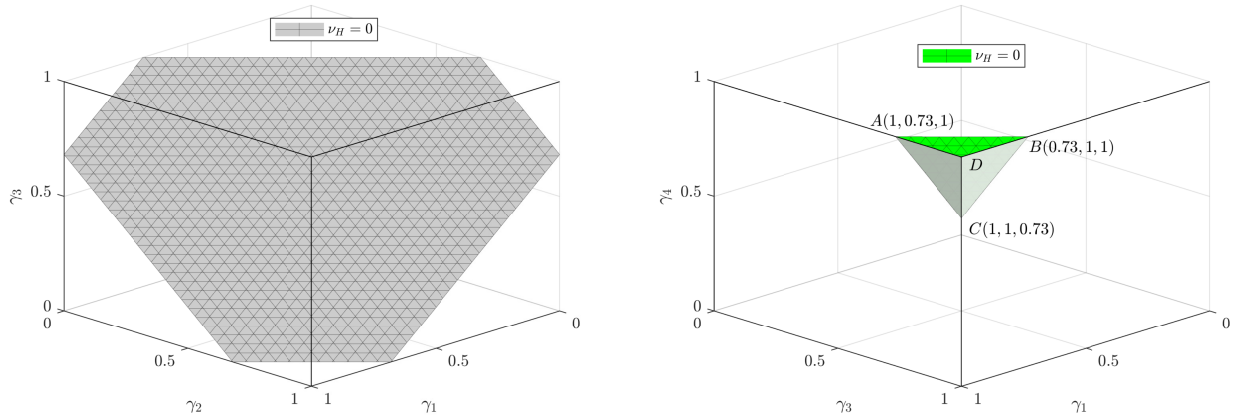
Pour ce qui est de la stabilité d'Hurwitz, l'application gardienne $\nu_H = \det(\mathbf{A}_{BF}) \det(\mathbf{A}_{BF} \odot \mathbf{I}_5)$ est un polynôme multivariable d'ordre élevé qui rend l'analyse graphique impossible. Cependant, en fixant un paramètre, il devient possible de l'étudier graphiquement. Par exemple, en fixant $\gamma_6 = 1$, on peut tracer les surfaces de niveau 0 de ν_H via la fonction `fimplicit3` de MATLAB. La figure 6.5 montre les graphiques obtenus où l'on peut distinguer les ensembles stables ou instables via le corollaire 6.1.1. En comparant les résultats, on voit que la stabilité du système bouclé est bien augmentée avec le contrôleur séquencé. Dans ce cas, la région instable est la pyramide $ABCD$ (Fig. 6.5b).

Comme préalablement mentionné, il est en général difficile de déterminer analytiquement les points d'annulation de l'application gardienne surtout dans le cas multiparamétrique. La méthode numérique développée précédemment est donc une solution pertinente.

Soit $\gamma_1, \gamma_3, \gamma_6 \in [0, 1]$ et $\gamma_4 \in [0, 0.7]$. Via le corollaire 6.1.5, on obtient les VSS suivantes :

$$\mu_{\mathbf{\Delta}_A}\left(\mathbf{M}_{22} - \mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12}\right) < 0.8513 \quad \text{et} \quad \mu_{\mathbf{\Delta}}\left(\mathbf{P}\left(\hat{\mathbf{M}}_{22} - \hat{\mathbf{M}}_{21}\hat{\mathbf{M}}_{11}^{-1}\hat{\mathbf{M}}_{12}\right)\mathbf{P}^\top\right) < 0.9856.$$

Cela indique que le système bouclé est stable. Le même cas de test mais avec $\gamma_4 \in [0, 0.8]$



(a) Contrôleur nominal (b) Contrôleur auto-séquencé

Figure 6.5 Stabilité d'Hurwitz du système LTI multi-entrée

nous donne :

$$\mu_{\mathbf{\Delta}_A}\left(\mathbf{M}_{22} - \mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12}\right) < 0.9006 \quad \text{et} \quad \mu_{\mathbf{\Delta}}\left(\mathbf{P}\left(\hat{\mathbf{M}}_{22} - \hat{\mathbf{M}}_{21}\hat{\mathbf{M}}_{11}^{-1}\hat{\mathbf{M}}_{12}\right)\mathbf{P}^{\top}\right) > 1.0132.$$

Ainsi, le système bouclé est instable ; ceci est en accord avec le résultat illustré à la figure 6.5b.

6.3 Conclusion du chapitre

Dans la première partie du chapitre, nous avons étudié la stabilité généralisée d'un système linéaire qui dépend de façon affine ou rationnelle de paramètres incertains. Les nouvelles conditions nécessaires et suffisantes de stabilité robuste ont été proposées sous la forme de la VSS d'une matrice constante. Cette méthode est basée sur les applications gardiennes, la valeur singulière structurée et la représentation LFT du système paramétré.

Dans la deuxième partie du chapitre, nous avons procédé à l'analyse de robustesse du système FTC contrôlé par les séquencements de gains. En nous basant sur les applications gardiennes et les résultats de la première partie du chapitre, nous avons étudié la robustesse du système FTC vis-à-vis de la perte d'efficacité de contrôle ainsi que de l'erreur d'estimation de fautes sortant du module FDD.

CHAPITRE 7 CONCLUSION

7.1 Contributions

La pierre d'angle de la thèse était de synthétiser une loi de commande tolérante aux fautes pour multicoptères dans le but de préserver la stabilité et la performance admissible du système bouclé malgré différents types d'incertitudes. Pour répondre à cette question, deux approches FTC ont été proposées : la première portant sur le séquencement de gains et la deuxième portant sur la commande générique par des méthodes adaptatives et robustes. Afin de valider les designs, une analyse approfondie de la stabilité généralisée robuste quant aux incertitudes paramétriques a été effectuée.

Après une revue de littérature exhaustive, nous avons dans le chapitre 3 abordé la modélisation générale de la dynamique des multicoptères ainsi que la conception d'un contrôleur nominal. Nous avons ensuite détaillé l'implémentation du contrôleur linéaire sur le montage expérimental et les aspects pratiques pour assurer un bon fonctionnement du drone.

Le chapitre 4 était consacré à la première approche FTC pour multicoptères basée sur l'autoséquencement de gains. À cette fin, nous avons conçu trois contrôleurs séquencés à architecture fixe dans le cadre de la synthèse $\mathcal{H}_\infty$ structurée. Tout d'abord, nous avons cherché à séquencer les gains du premier contrôleur en fonction de la perte d'efficacité des actionneurs. En tirant profit de la configuration spécifique d'un hexacoptère, la structure de séquencement de gains a été simplifiée de manière significative. Pour le rendre implantable sur les autres multicoptères, la perte d'efficacité de contrôle virtuelle a été utilisée en tant que variable de séquencement dans le deuxième contrôleur. Enfin, on s'est proposé d'étendre cette approche pour le problème de ré-allocation de contrôle. À travers les expérimentations réelles, tous les contrôleurs ont démontré leur bon niveau de performance dans la mesure où ils ont permis de compenser les fautes d'actionneurs.

Le chapitre 5 s'est concentré sur la conception du contrôleur générique tolérant aux fautes. La synthèse du contrôleur s'est basée sur le recours à la commande adaptative et à l'utilisation des entrées normalisées. De la sorte, un contrôleur unique avec des paramètres constants a été implémenté sur différents multicoptères. Pour le rendre robuste face aux perturbations non paramétriques, une modification par opérateur de projection a été appliquée. Les résultats ainsi obtenus, autant en simulation qu'expérimentalement, ont permis de démontrer un bon comportement du système bouclé pour tous les indices de performance assignés.

Dans le dernier chapitre, nous avons fourni une façon systématique pour analyser la stabilité

généralisée d'un système linéaire qui dépend de façon affine ou rationnelle de paramètres incertains. Cette méthode est basée sur les applications gardiennes, la valeur singulière structurée et la représentation LFT du système paramétré. Via cette étude, nous avons enfin procédé à l'analyse de robustesse du système FTC contrôlé par les séquencements de gains précédemment synthétisés.

En résumé, les stratégies FTC proposées ont démontré de nombreux avantages parmi lesquels : une architecture fixe et simple avec séquencement de gains, une robustesse par rapport aux incertitudes paramétriques et non paramétriques, et une tolérance aux fautes efficace. Puisque les dégradations ou les pannes de rotors peuvent causer l'instabilité et mener à l'écrasement de l'appareil, l'utilisation d'un tel système FTC est une solution à considérer pour atténuer ce phénomène. Cela permet ainsi d'augmenter la durée de vie des multicoptères et surtout d'éviter des accidents importants sur l'environnement et les humains aux alentours.

7.2 Perspectives

Au regard de la commande générique tolérante aux fautes, il serait intéressant de l'adapter pour les véhicules à voilure fixe. Le fondement de cette approche réside dans le fait que les actionneurs d'un tel UAV agissent sur le fonctionnement du système de manière similaire aux rotors d'un multicoptère. En effet, ils fournissent une force le long d'un axe fixe du repère objet [1] et un moment autour de l'axe de rotation.[2] Bien évidemment, les forces aérodynamiques de portance et de traînée ne seront plus négligées dans ce cas.

Dans la littérature, il existe très peu de recherches concernant une telle stratégie de contrôle universelle pour UAV. On mentionne ici le travail de [238], où un seul contrôleur non linéaire est implémenté sur un quadricoptère et un petit aéronef de type aile volante. L'approche adoptée consiste à utiliser un *mixer* pour générer le signal de commande de l'actionneur. Cependant, les résultats préliminaires en simulations n'étaient pas vraiment convaincants en termes de performance en suivi de trajectoire. Combiner cette approche avec notre algorithme FTC adaptatif et robuste pourrait être une des pistes de travail possibles.

En ce qui concerne les études reposant sur les applications gardiennes, on n'a traité que les incertitudes structurées bornées dans un hyperrectangle. Il reste à étudier le cas de polytopes que l'on rencontre souvent dans la littérature. On pourrait aussi exploiter ces résultats pour trouver une approximation de la région de stabilité la plus large dans l'espace des paramètres incertains via la VSS. Vu la complexité du problème, les méthodes d'optimisation sans dérivée seraient considérées pour le résoudre sous forme d'une boîte noire. De plus, on pourrait

1. soit $\mathbf{z}_b$ d'un multicoptère soit $\mathbf{x}_b$ d'un véhicule à voilure fixe
2. soit le vecteur $\mathbf{a}$ du quaternion (3.6)

améliorer la méthode du séquencement de gains par les applications gardiennes [11] dans le cadre de la commande robuste. En effet, l'approche développée dans cette thèse a fourni un test simple de stabilité sous forme de la VSS d'une matrice constante. De cette façon, il serait possible de simplifier les algorithmes proposés dans [11]. En particulier, on pourrait étendre cette approche au cas multiparamétrique.

RÉFÉRENCES

[1] W. J. Rugh et J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, n°. 10, p. 1401–1425, 2000.

[2] D. J. Stilwell et W. J. Rugh, "Stability preserving interpolation methods for the synthesis of gain scheduled controllers," *Automatica*, vol. 36, n°. 5, p. 665–671, 2000.

[3] S. Bhattacharyya *et al.*, "Certification considerations for adaptive systems," dans *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. IEEE, 2015, p. 270–279.

[4] S. Boyd *et al.*, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[5] P. Apkarian, P. Gahinet et G. Becker, "Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems : a design example," *Automatica*, vol. 31, n°. 9, p. 1251–1261, 1995.

[6] J.-M. Biannic et P. Apkarian, "Missile autopilot design via a modified LPV synthesis technique," *Aerospace Science and Technology*, vol. 3, n°. 3, p. 153 – 160, 1999.

[7] P. Apkarian et H. D. Tuan, "Parameterized LMIs in control theory," *SIAM journal on control and optimization*, vol. 38, n°. 4, p. 1241–1264, 2000.

[8] J.-F. Magni, Y. Le Gorrec et C. Chiappa, "A multimodel-based approach to robust and self-scheduled control design," dans *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, vol. 3. IEEE, 1998, p. 3009–3014.

[9] H. Lhachemi, "Commande des systèmes aérospatiaux non linéaires à dimensions finie et infinie," Thèse de doctorat, École Polytechnique de Montréal, 2017.

[10] A. Packard, "Gain scheduling via linear fractional transformations," *Systems & control letters*, vol. 22, n°. 2, p. 79–92, 1994.

[11] D. Saussié *et al.*, "Gain scheduling with guardian maps for longitudinal flight control," *Journal of Guidance, Control, and Dynamics*, vol. 34, n°. 4, p. 1045–1059, 2011.

[12] D. Saussié, Q. Barbès et C. Bérard, "Self-scheduled and structured $\mathcal{H}_\infty$ synthesis : A launch vehicle application," dans *2013 American Control Conference*, 2013, p. 1590–1595.

[13] P. Apkarian, P. Gahinet et C. Buhr, "Multi-model, multi-objective tuning of fixed-structure controllers," dans *European Control Conference (ECC)*, 2014, p. 856–861.

[14] H. Lhachemi, D. Saussie et G. Zhu, "A robust and self-scheduled longitudinal flight control system : a multi-model and structured $\mathcal{H}_\infty$ approach," dans *AIAA Guidance, Navigation, and Control Conference*. AIAA Scitech, 2014.

[15] ——, "A structured $\mathcal{H}_\infty$-based optimization approach for integrated plant and self-scheduled flight control system design," *Aerospace Science and Technology*, vol. 45, p. 30–38, 2015.

[16] H. Lhachemi, D. Saussié et G. Zhu, "Gain-scheduling control design in the presence of hidden coupling terms," *Journal of Guidance, Control, and Dynamics*, vol. 39, n°. 8, p. 1872–1880, 2016.

[17] G. Ducard, *Fault-tolerant Flight Control and Guidance Systems : Practical Methods for Small Unmanned Aerial Vehicles*. Springer London, 2010.

[18] K. J. Aström et B. Wittenmark, *Adaptive control*, 2e éd. Addison-Wesley Longman, 1995.

[19] E. Lavretsky et K. Wise, *Robust and Adaptive Control : With Aerospace Applications*. Springer London, 2012.

[20] B. Peterson et K. Narendra, "Bounded error adaptive control," *IEEE Transactions on Automatic Control*, vol. 27, n°. 6, p. 1161–1168, 1982.

[21] P. A. Ioannou et P. V. Kokotović, *Adaptive systems with reduced models*. Springer-Verlag New York, 1983, vol. 68.

[22] K. Narendra et A. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Transactions on Automatic control*, vol. 32, n°. 2, p. 134–145, 1987.

[23] P. Ioannou et B. Fidan, *Adaptive control tutorial*. SIAM, 2006.

[24] G. Larchev, S. Campbell et J. Kaneshige, "Projection operator : A step toward certification of adaptive controllers," dans *AIAA Infotech@ Aerospace 2010*, 2010, p. 3366–3379.

[25] N. Hovakimyan et C. Cao, $\mathcal{L}_1$ *adaptive control theory : guaranteed robustness with fast adaptation*. Philadelphia, Pa. : Society for Industrial and Applied Mathematics, 2010.

[26] J. Pilon, "Commande adaptative robuste par modèle de référence d'un quadricoptère," Thesis, 2019.

[27] R. Isermann, *Fault-Diagnosis Systems : An Introduction from Fault Detection to Fault Tolerance*. Springer Berlin Heidelberg, 2006.

[28] Y. M. Zhang et J. Jiang, "Bibliographical review on reconfigurable Fault-Tolerant Control Systems," *Annual Reviews in Control*, vol. 32, n°. 2, p. 229–252, 2008.

[29] V. Venkatasubramanian *et al.*, "A review of process fault detection and diagnosis : Part iii : Process history based methods," *Computers & Chemical Engineering*, vol. 27, n°. 3, p. 327–346, 2003.

[30] J. Marzat, "Diagnostic des systèmes aéronautiques et réglage automatique pour la comparaison de méthodes," Thèse de doctorat, Université Paris Sud-Paris XI, 2011.

[31] S. Simani, C. Fantuzzi et R. Patton, *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. Springer London, 2013.

[32] J. Chen et R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Springer US, 2012.

[33] V. Venkatasubramanian *et al.*, "A review of process fault detection and diagnosis : Part i : Quantitative model-based methods," *Computers & Chemical Engineering*, vol. 27, n°. 3, p. 293 – 311, 2003.

[34] M. Blanke *et al.*, *Diagnosis and Fault-Tolerant Control*, 3e éd. Springer Berlin Heidelberg, 2015.

[35] J. Y. Keller et M. Darouach, "Optimal Two-Stage Kalman filter in the presence of random bias," *Automatica*, vol. 33, n°. 9, p. 1745–1748, 1997.

[36] N. E. Wu, Y. M. Zhang et K. M. Zhou, "Detection, estimation, and accommodation of loss of control effectiveness," *International Journal of Adaptive Control and Signal Processing*, vol. 14, n°. 7, p. 775–795, 2000.

[37] Y. Zhang et J. Jiang, "Active fault-tolerant control system against partial actuator failures," *IEE proceedings-Control Theory and applications*, vol. 149, n°. 1, p. 95–104, 2002.

[38] M. H. Amoozgar, A. Chamseddine et Y. M. Zhang, "Experimental test of a Two-Stage Kalman filter for actuator fault detection and diagnosis of an unmanned quadrotor helicopter," *Journal of Intelligent & Robotic Systems*, vol. 70, n°. 1-4, p. 107–117, 2013.

[39] A. Chamseddine, M. H. Amoozgar et Y. M. Zhang, *Experimental Validation of Fault Detection and Diagnosis for Unmanned Aerial Vehicles*. Dordrecht : Springer Netherlands, 2015, p. 1123–1155.

[40] F. Caliskan et C. Hajiyev, "Reconfigurable control of an UAV against sensor/actuator failures," dans *1st IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles (ACNAAV)*, vol. 28. IFAC Secretariat, 2015, p. 7–12.

[41] ——, "Active Fault-Tolerant Control of UAV dynamics against sensor-actuator failures," *Journal of Aerospace Engineering*, vol. 29, n°. 4, 2016.

[42] F. Caliskan et C. Hacizade, "Sensor and actuator FDI applied to an UAV dynamic model," dans *19th World Congress on International Federation of Automatic Control (IFAC)*, 2014.

[43] M. Moghadam et F. Caliskan, "Actuator and sensor fault detection and diagnosis of quadrotor based on Two-Stage Kalman filter," dans *5th Australian Control Conference (AUCC)*. IEEE Computer Society, 2015, p. 182–187.

[44] J. Jiang et X. Yu, "Fault-Tolerant Control systems : A comparative study between active and passive approaches," *Annual Reviews in Control*, vol. 36, n°. 1, p. 60–72, 2012.

[45] J. Stoustrup et V. D. Blondel, "Fault tolerant control : a simultaneous stabilization result," *IEEE Transactions on Automatic Control*, vol. 49, n°. 2, p. 305–310, Feb 2004.

[46] R. J. Veillette, J. B. Medanic et W. R. Perkins, "Design of reliable control systems," *IEEE Transactions on Automatic Control*, vol. 37, n°. 3, p. 290–304, Mar 1992.

[47] M. Benosman, *Passive Fault Tolerant Control*. InTech, 2011, p. 283 – 308.

[48] H. Alwi, C. Edwards et C. Tan, *Fault Detection and Fault-Tolerant Control Using Sliding Modes*. Springer London, 2011.

[49] C. Edwards, T. Lombaerts et H. Smaili, *Fault Tolerant Flight Control : A Benchmark Challenge*. Springer Berlin Heidelberg, 2010.

[50] Z. Gao et P. J. ANTSAKLIS, "Reconfigurable control system design via perfect model following," *International Journal of Control*, vol. 56, n°. 4, p. 783–798, 1992.

[51] R. Mahony, R. W. Beard et V. Kumar, *Modeling and Control of Aerial Robots*. Springer International Publishing, 2016, p. 1307–1334.

[52] A. Zulu et S. John, "A review of control algorithms for autonomous quadrotors," *Open Journal of Applied Sciences*, vol. 04, n°. 14, p. 547–556, 2014.

[53] T. P. Nascimento et M. Saska, "Position and attitude control of multi-rotor aerial vehicles : A survey," *Annual Reviews in Control*, vol. 48, p. 129 – 146, 2019. [En ligne]. Disponible : http://www.sciencedirect.com/science/article/pii/S1367578819300483

[54] N. Michael *et al.*, "The grasp multiple micro-uav testbed," *IEEE Robotics & Automation Magazine*, vol. 17, n°. 3, p. 56–65, 2010.

[55] K. U. Lee *et al.*, "Hovering control of a quadrotor," dans *Control, Automation and Systems (ICCAS), 2012 12th International Conference on*. IEEE, 2012, p. 162–167.

[56] J. Li et Y. Li, "Dynamic analysis and pid control for a quadrotor," dans *2011 IEEE International Conference on Mechatronics and Automation*, Aug 2011, p. 573–578.

[57] S. Bouabdallah, A. Noth et R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," dans *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, Sept 2004, p. 2451–2456 vol.3.

[58] I. D. Cowling *et al.*, “A prototype of an autonomous controller for a quadrotor UAV,” dans *2007 European Control Conference (ECC)*, July 2007, p. 4001–4008.

[59] L. D. Minh et C. Ha, “Modeling and control of quadrotor MAV using vision-based measurement,” dans *International Forum on Strategic Technology 2010*, Oct 2010, p. 70–75.

[60] T. Madani et A. Benallegue, “Backstepping control for a quadrotor helicopter,” dans *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006.

[61] S. Bouabdallah et R. Siegwart, “Backstepping and sliding-mode techniques applied to an indoor micro quadrotor,” dans *IEEE International Conference on Robotics and Automation*, 2005, p. 2247–2252.

[62] Z. Fang et W. Gao, “Adaptive integral backstepping control of a micro-quadrotor,” dans *Intelligent Control and Information Processing (ICICIP), 2011 2nd International Conference on*, vol. 2. IEEE, 2011, p. 910–915.

[63] C. Diao *et al.*, “A nonlinear adaptive control approach for quadrotor uavs,” dans *Control Conference (ASCC), 2011 8th Asian*. IEEE, 2011, p. 223–228.

[64] I. Palunko et R. Fierro, “Adaptive control of a quadrotor with dynamic changes in the center of gravity,” *IFAC Proceedings Volumes*, vol. 44, n°. 1, p. 2626–2631, 2011.

[65] P. De Monte et B. Lohmann, “Position trajectory tracking of a quadrotor helicopter based on l1 adaptive control,” dans *2013 European Control Conference (ECC)*, 2013, p. 3346–3353.

[66] K. Runcharoon et V. Srichatrapimuk, “Sliding mode control of quadrotor,” dans *2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*, May 2013, p. 552–557.

[67] R. Xu et U. Ozguner, “Sliding mode control of a quadrotor helicopter,” dans *Proceedings of the 45th IEEE Conference on Decision and Control*, Dec 2006, p. 4957–4962.

[68] Y. Bai *et al.*, “Robust control of quadrotor unmanned air vehicles,” dans *Control Conference (CCC), 2012 31st Chinese*. IEEE, 2012, p. 4462–4467.

[69] C. T. Ton et W. Mackunis, “Robust attitude tracking control of a quadrotor helicopter in the presence of uncertainty,” dans *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, p. 937–942.

[70] A. C. Satici, H. Poonawala et M. W. Spong, “Robust optimal control of quadrotor uavs,” *IEEE Access*, vol. 1, p. 79–93, 2013.

[71] A. Roza et M. Maggiore, “Path following controller for a quadrotor helicopter.” dans *ACC*, 2012, p. 4655–4660.

[72] O. Falkenberg *et al.*, "Model identification and hinf attitude control for quadrotor mav's," *Intelligent Robotics and Applications*, p. 460–471, 2012.

[73] M. Santos, V. Lopez et F. Morata, "Intelligent fuzzy controller of a quadrotor," dans *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on.* IEEE, 2010, p. 141–146.

[74] C. Nicol, C. Macnab et A. Ramirez-Serrano, "Robust neural network control of a quadrotor helicopter," dans *Canadian Conference on Electrical and Computer Engineering*, 2008, p. 1233–1237.

[75] Y. Zhang et A. Chamseddine, *Fault Tolerant Flight Control Techniques with Application to a Quadrotor UAV Testbed.* InTech, 2012, p. 119 – 150.

[76] T. Li, Y. Zhang et B. W. Gordon, "Passive and active nonlinear fault-tolerant control of a quadrotor unmanned aerial vehicle based on the sliding mode control technique," *Proceedings of the Institution of Mechanical Engineers, Part I : Journal of Systems and Control Engineering*, vol. 227, n°. 1, p. 12–23, 2013.

[77] A.-R. Merheb, H. Noura et F. Bateman, "Passive Fault Tolerant Control of quadrotor UAV using regular and cascaded sliding mode control," dans *Conference on Control and Fault-Tolerant Systems, SysTol.* IEEE Computer Society, 2013, p. 330–335.

[78] ——, "Design of passive fault–tolerant controllers of a quadrotor based on sliding mode theory," *International Journal of Applied Mathematics and Computer Science*, vol. 25, n°. 3, p. 561–576, 2015.

[79] A.-R. Merheb, F. Bateman et H. Noura, "Passive and Active Fault Tolerant Control of octorotor UAV using Second Order Sliding Mode control," dans *IEEE Conference on Control and Applications (CCA).* IEEE Computer Society, 2015, p. 1907–1912.

[80] S. Iman et Z. Youmin, "A review on Fault-Tolerant Control for unmanned aerial vehicles (UAVs)," dans *Infotech@Aerospace 2011.* AIAA Inc., 2011.

[81] M. H. Amoozgar, A. Chamseddine et Y. M. Zhang, "Experimental test of an interacting multiple model filtering algorithm for actuator fault detection and diagnosis of an unmanned quadrotor helicopter," dans *5th International Conference on Intelligent Robotics and Applications (ICIRA)*, vol. 7506 LNAI. Springer Verlag, 2012, p. 473–482.

[82] B. Yu *et al.*, "Fault detection for partial loss of effectiveness faults of actuators in a quadrotor unmanned helicopter," dans *11th World Congress on Intelligent Control and Automation (WCICA).* IEEE Computer Society, 2015, p. 3204–3209.

[83] Y. M. Zhang *et al.*, "Development of advanced FDD and FTC techniques with application to an unmanned quadrotor helicopter testbed," *Journal of the Franklin Institute*, vol. 350, n°. 9, p. 2396–2422, 2013.

[84] G. Ducard, *Actuator Fault Detection in UAVs*. Dordrecht : Springer Netherlands, 2015, p. 1071–1122.

[85] G. J. J. Ducard, "SMAC-FDI : A single model active fault detection and isolation system for unmanned aircraft," *International Journal of Applied Mathematics and Computer Science*, vol. 25, n°. 1, p. 189–201, 2015.

[86] A. De Souza Cndido, R. K. Harrop Galvao et T. Yoneyama, "Actuator fault diagnosis and control of a quadrotor," dans *12th IEEE International Conference on Industrial Informatics (INDIN)*. IEEE Computer Society, 2014, p. 310–315.

[87] A. Chamseddine *et al.*, "Active Fault-Tolerant Control System design with trajectory re-planning against actuator faults and saturation : Application to a Quadrotor Unmanned Aerial Vehicle," *International Journal of Adaptive Control and Signal Processing*, vol. 29, n°. 1, p. 1–23, 2015.

[88] A. Freddi, S. Longhi et A. Monteriu, "Actuator fault detection system for a mini-quadrotor," dans *IEEE International Symposium on Industrial Electronics (ISIE)*. IEEE Computer Society, 2010, p. 2055–2060.

[89] A. Freddi, S. Longhi et A. Monteriù, "A diagnostic thau observer for a class of unmanned vehicles," *Journal of Intelligent and Robotic Systems*, vol. 67, n°. 1, p. 61–73, 2012.

[90] Y. Al Younes *et al.*, "Sensor fault detection and isolation in the quadrotor vehicle using nonlinear identity observer approach," dans *2nd International Conference on Control and Fault-Tolerant Systems (SysTol)*. IEEE Computer Society, 2013, p. 486–491.

[91] K. P. B. Chandra, H. Alwi et C. Edwards, "Fault reconstruction for a quadrotor using an lpv sliding mode observer," dans *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, vol. 28. IFAC Secretariat, 2015, p. 374–379.

[92] M. Saied *et al.*, "Fault Tolerant Control for multiple successive failures in an octorotor : Architecture and experiments," dans *IEEE/Rsj International Conference on Intelligent Robots and Systems (IROS)*, 2015, p. 40–45.

[93] I. F. Majd Saied, B. L. Clovis Francis et H. Shraim, "Détection et diagnostic d'une défaillance d'actionneur dans un octorotor," dans 11^{e} *Congrès International Pluridisciplinaire en Qualité, Sûreté de Fonctionnement et Développement Durable (QUALITA)*, 2015.

[94] M. Saied *et al.*, "Actuator fault diagnosis in an octorotor UAV using sliding modes technique : Theory and experimentation," dans *European Control Conference (ECC)*. IEEE Computer Society, 2015, p. 1639–1644.

[95] F. R. Lopez-Estrada *et al.*, "LPV model-based tracking control and robust sensor fault diagnosis for a quadrotor UAV," *Journal of Intelligent and Robotic Systems : Theory and Applications*, 2015.

[96] A.-R. Merheb *et al.*, "Fault severity based integrated Fault Tolerant Controller for quadrotor UAVs," dans *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE Computer Society, 2015, p. 660–668.

[97] G. P. Falconi, C. D. Heise et F. Holzapfel, "Fault-tolerant position tracking of a hexacopter using an extended state observer," dans *6th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE Computer Society, 2015, p. 550–556.

[98] F. Chen *et al.*, "Robust backstepping sliding-mode control and observer-based fault estimation for a quadrotor UAV," *IEEE Transactions on Industrial Electronics*, vol. 63, n°. 8, p. 5044–5056, 2016.

[99] L. Qin *et al.*, "Active fault-tolerant control for a quadrotor with sensor faults," *Journal of Intelligent & Robotic Systems*, p. 1–19, 2017.

[100] F. Michael, S. Johannes et F. Walter, "Fast actuator fault detection and reconfiguration for multicopters," dans *AIAA Guidance, Navigation, and Control Conference*. AIAA Inc., 2015.

[101] V. Stepanyan, K. S. Krishnakumar et A. Bencomo, "Identification and reconfigurable control of impaired multi-rotor drones," dans *AIAA Guidance, Navigation, and Control Conference*, 2016.

[102] X. He *et al.*, "Fault Tolerant Control of a quadrotor based on parameter estimation techniques and use of a reconfigurable PID controller," dans *24th Mediterranean Conference on Control and Automation*. IEEE Computer Society, 2016, p. 188–193.

[103] W. Han, Z. Wang et Y. Shen, "Fault estimation for a quadrotor unmanned aerial vehicle by integrating the parity space approach with recursive least squares," *Proceedings of the Institution of Mechanical Engineers, Part G : Journal of Aerospace Engineering*, 2017.

[104] D. Zachary, A. Anuradha et L. Eugene, "Adaptive control of quadrotor UAVs in the presence of actuator uncertainties," dans *AIAA Infotech@Aerospace 2010*. AIAA Inc., 2010.

[105] X. Zhang et Y. Zhang, "Fault Tolerant Control for quad-rotor UAV by employing Lyapunov-based adaptive control approach," dans *AIAA Guidance, Navigation, and Control Conference*. AIAA Inc., 2010.

[106] A. Chamseddine *et al.*, "Model Reference Adaptive Fault Tolerant Control of a quadrotor UAV," dans *AIAA Infotech at Aerospace Conference and Exhibit*. AIAA Inc., 2011.

[107] I. Sadeghzadeh *et al.*, "Fault-tolerant trajectory tracking control of a quadrotor helicopter using Gain-Scheduled PID and Model Reference Adaptive Control," dans *Annual Conference of the Prognostics and Health Management Society*, 2011.

[108] I. Sadeghzadeh, A. Mehta et Y. Zhang, "Fault/damage tolerant control of a quadrotor helicopter UAV using Model Reference Adaptive Control and Gain-Scheduled PID," dans *AIAA Guidance, Navigation, and Control Conference 2011*. AIAA Inc., 2011.

[109] D. Kapoor *et al.*, "Adaptive failure compensation for coaxial rotor helicopter under propeller failure," dans *American Control Conference*. IEEE Computer Society, 2012, p. 2539–2544.

[110] A. Chamseddine *et al.*, "Optimal reliability design for over-actuated systems based on the MIT rule : Application to an octocopter helicopter testbed," *Reliability Engineering & System Safety*, vol. 132, p. 196–206, 2014.

[111] Q. Shen *et al.*, "Fault-tolerant attitude tracking control for a quadrotor aircraft," dans *IEEE 53rd Annual Conference on Decision and Control (CDC)*, 2014, p. 6129–6134.

[112] F. Chen *et al.*, "A reconfiguration scheme for quadrotor helicopter via simple adaptive control and quantum logic," *IEEE Transactions on Industrial Electronics*, vol. 62, n°. 7, p. 4328–4335, 2015.

[113] M. Muhlegg *et al.*, "$\mathcal{L}_1$ fault tolerant adaptive control of a hexacopter with control degradation," dans *IEEE Conference on Control and Applications (CCA)*, 2015, p. 750–755.

[114] R. C. Avram *et al.*, "Nonlinear adaptive control design and controller integrity monitoring for quadrotor UAVs," dans *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE Computer Society, 2016, p. 301–309.

[115] X. Dan, J. F. Whidborne et A. Cooke, "Fault Tolerant Control of a quadrotor using $\mathcal{L}_1$ adaptive control," *International Journal of Intelligent Unmanned Systems*, vol. 4, n°. 1, p. 43–66, 2016.

[116] A. A. Ghaffar et T. Richardson, "Position tracking of an underactuated quadrotor using Model Reference Adaptive control," dans *AIAA Guidance, Navigation, and Control Conference*. AIAA Inc., 2016.

[117] J. Wang *et al.*, “Direct adaptive control scheme for a quadrotor helicopter with actuator failures via quantum logic,” *Journal of Aerospace Engineering*, vol. 29, n°. 5, 2016.

[118] H.-J. Yoon, V. Cichella et N. Hovakimyanx, “Robust adaptive control allocation for an octocopter under actuator faults,” dans *AIAA Guidance, Navigation, and Control Conference*. AIAA Inc., 2016.

[119] A. B. Milhim, Y. Zhang et C.-A. Rabbath, “Gain scheduling based PID controller for Fault Tolerant Control of a quad-rotor UAV,” dans *AIAA Infotech at Aerospace*. AIAA Inc., 2010.

[120] M. H. Amoozgar, A. Chamseddine et Y. Zhang, “Fault-tolerant Fuzzy Gain-Scheduled PID for a quadrotor helicopter testbed in the presence of actuator faults,” dans *2nd IFAC Conference on Advances in PID Control*, vol. 2. IFAC Secretariat, 2012, p. 282–287.

[121] I. Sadeghzadeh *et al.*, “Active Fault Tolerant Control of a quadrotor UAV based on gain scheduled PID control,” dans *25th IEEE Canadian Conference on Electrical and Computer Engineering : Vision for a Greener Future (CCECE)*. IEEE Computer Society, 2012.

[122] ——, “Linear parameter varying control synthesis : State feedback versus $\mathcal{H}_\infty$ technique with application to quadrotor UAV,” dans *International Conference on Unmanned Aircraft Systems (ICUAS 2014)*. IEEE Computer Society, 2014, p. 1099–1104.

[123] F. Chen *et al.*, “A reconfiguration control scheme for a quadrotor helicopter via combined multiple models,” *International Journal of Advanced Robotic Systems*, vol. 11, n°. 1, 2014.

[124] B. Yu, Y. Zhang et Y. Qu, “Fault Tolerant Control using PID structured optimal technique against actuator faults in a quadrotor UAV,” dans *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE Computer Society, 2014, p. 167–174.

[125] H. Basak et E. Prempain, “Switching recovery control of a quadcopter UAV,” dans *European Control Conference (ECC)*. IEEE Computer Society, 2015, p. 3641–3646.

[126] P. Lu et E.-J. Van Kampen, “Active Fault-Tolerant Control for quadrotors subjected to a complete rotor failure,” dans *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-December. IEEE Computer Society, 2015, p. 4698–4703.

[127] D. Rotondo *et al.*, “Fault Tolerant Control design for polytopic uncertain LPV systems : Application to a quadrotor,” dans *Conference on Control and Fault-Tolerant Systems, SysTol*. IEEE Computer Society, 2013, p. 643–648.

[128] F. R. Lopez-Estrada *et al.*, "Robust sensor fault diagnosis and tracking controller for a UAV modelled as LPV system," dans *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE Computer Society, 2014, p. 1311–1316.

[129] Z. Liu, C. Yuan et Y. Zhang, "Adaptive Fault-Tolerant Control of unmanned quadrotor helicopter using linear parameter varying control technique," *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016.

[130] T. Li, Y. Zhang et B. Gordon, "Fault Tolerant Control applied to a quadrotor unmanned helicopter," dans *ASME Design Engineering Technical Conference*, vol. 3. American Society of Mechanical Engineers, 2011, p. 1013–1022.

[131] T. Li, Y. Zhang et B. W. Gordon, "Nonlinear Fault-Tolerant Control of a quadrotor UAV based on sliding mode control technique," dans *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, vol. 8. IFAC Secretariat, 2012, p. 1317–1322.

[132] H. Alwi et C. Edwards, "LPV sliding mode fault tolerant control of an octorotor using fixed control allocation," dans *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, 2013, p. 772–777.

[133] F. Sharifi *et al.*, "Fault Tolerant Control of a quadrotor UAV using sliding mode control," dans *Conference on Control and Fault-Tolerant Systems, SysTol'10*. IEEE Computer Society, 2010, p. 239–244.

[134] A.-R. Merheb, H. Noura et F. Bateman, "A novel emergency controller for quadrotor UAVs," dans *IEEE Conference on Control Applications (CCA)*. Institute of Electrical and Electronics Engineers Inc., 2014, p. 747–752.

[135] A. R. Merheb, H. Noura et F. Bateman, "Active fault tolerant control of quadrotor UAV using sliding mode control," dans *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, p. 156–166.

[136] H. Khebbache, "Robust control algorithm considering the actuator faults for attitude tracking of an UAV quadrotor aircraft," *International Journal of Control and Automation*, vol. 5, n°. 4, p. 55–66, 2012.

[137] V. Lippiello, F. Ruggiero et D. Serra, "Emergency landing for a quadrotor in case of a propeller failure : A backstepping approach," dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, p. 4782–4788.

[138] B. Yu *et al.*, "Fault-Tolerant Control with linear quadratic and model predictive control techniques against actuator faults in a quadrotor UAV," dans *Conference on Control and Fault-Tolerant Systems, SysTol*. IEEE Computer Society, 2013, p. 661–666.

[139] M. Kamel *et al.*, “Fast nonlinear model predictive control for multicopter attitude tracking on so(3),” dans *IEEE Conference on Control and Applications (CCA)*, 2015, p. 1160–1166.

[140] Q.-L. Zhou *et al.*, “Design of feedback linearization control and reconfigurable control allocation with application to a quadrotor UAV,” dans *Conference on Control and Fault-Tolerant Systems, SysTol'10 - Final Program and Book of Abstracts*. IEEE Computer Society, 2010, p. 371–376.

[141] ——, “Reconfigurable control allocation technology using weighted least squares for nonlinear system in unmanned aerial vehicle,” dans *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010, p. 1138.

[142] C. H. Lo *et al.*, “Modeling and simulation of fault tolerant strategies for a quad rotor UAV,” dans *AIAA Modeling and Simulation Technologies Conference 2012*. AIAA Inc., 2012.

[143] A. Marks, J. F. Whidborne et I. Yamamoto, “Control allocation for fault tolerant control of a VTOL octorotor,” dans *2012 UKACC International Conference on Control*, 2012, p. 357–362.

[144] I. Sadeghzadeh *et al.*, “Control allocation and re-allocation for a modified quadrotor helicopter against actuator faults,” *IFAC Proceedings Volumes*, vol. 45, n°. 20, p. 247–252, 2012.

[145] T. Schneider *et al.*, “Fault-Tolerant Control Allocation for multirotor helicopters using parametric programming,” dans *International Micro Air Vehicle Conference and Flight Competition (IMAV)*, 2012.

[146] A. Chamseddine *et al.*, *Control Allocation for a Modified Quadrotor Helicopter Based on Reliability Analysis*, ser. Infotech@Aerospace Conferences. AIAA Inc., 2012.

[147] A. Merheb et F. Bateman, “Active fault tolerant control of octorotor UAV using dynamic control allocation,” dans *International Conference on Intelligent Unmanned Systems*, 2014.

[148] I. Sadeghzadeh, “Fault tolerant flight control of unmanned aerial vehicles,” Thesis, Concordia University, 2015.

[149] J. C. Monteiro, F. Lizarralde et H. Liu, “Optimal control allocation of quadrotor UAVs subject to actuator constraints,” dans *American Control Conference (ACC)*, 2016, p. 500–505.

[150] G. P. Falconi et F. Holzapfel, “Adaptive Fault Tolerant Control Allocation for a hexacopter system,” dans *American Control Conference (ACC)*, vol. 2016-July. Institute of Electrical and Electronics Engineers Inc., 2016, p. 6760–6766.

[151] ——, "Adaptive fault tolerant control allocation for a hexacopter system," dans *American Control Conference (ACC)*, 2016.

[152] S. Tohidi, Y. Yildiz et I. Kolmanovsky, "Adaptive control allocation for over-actuated systems with actuator saturation," dans *20th World Congress of the International Federation of Automatic Control (IFAC)*, 2017.

[153] V. S. A. Holzapfel, G. P. Falconi et F., "Fault tolerant incremental attitude control using online parameter estimation for a multicopter system," dans *25th Mediterranean Conference on Control and Automation (MED)*, 2017.

[154] A. Chamseddine, Y. Zhang et C. A. Rabbath, "Trajectory planning and re-planning for fault tolerant formation flight control of quadrotor unmanned aerial vehicles," dans *American Control Conference*. Institute of Electrical and Electronics Engineers Inc., 2012, p. 3291–3296.

[155] M. A. Kamel, K. A. Ghamry et Y. Zhang, "Fault tolerant cooperative control of multiple UAVs-UGVs under actuator faults," dans *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE Computer Society, 2015, p. 644–649.

[156] H. Yang *et al.*, "Cooperative control reconfiguration in multiple quadrotor systems with actuator faults," dans *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, vol. 28. IFAC Secretariat, 2015, p. 386–391.

[157] M. A. Kamel, K. A. Ghamry et Y. Zhang, "Real-time fault-tolerant cooperative control of multiple UAVs-UGVs in the presence of actuator faults," dans *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE Computer Society, 2016, p. 1267–1272.

[158] M. W. Mueller et R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," dans *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Computer Society, 2014, p. 45–52.

[159] V. Lippiello, F. Ruggiero et D. Serra, "Emergency landing for a quadrotor in case of a propeller failure : A PID based approach," dans *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2014, p. 1–7.

[160] G. P. Falconi, S. P. Schatz et F. Holzapfel, "Fault Tolerant Control of a hexarotor using a command governor augmentation," dans *24th Mediterranean Conference on Control and Automation*. IEEE Computer Society, 2016, p. 182–187.

[161] J. Lee, H. S. Choi et H. Shim, "Fault Tolerant Control of hexacopter for actuator faults using time delay control method," *International Journal of Aeronautical and Space Sciences*, vol. 17, n°. 1, p. 54–63, 2016.

[162] H. Alwi, C. Edwards et C. P. Tan, "Fault tolerant control with online control allocation," dans *Fault Detection and Fault-Tolerant Control Using Sliding Modes*. Springer, 2011, p. 225–246.

[163] M. W. Oppenheimer, D. B. Doman et M. A. Bolender, "Control allocation for overactuated systems," dans *14th Mediterranean Conference on Control and Automation (MED'06)*. IEEE, 2006, p. 1–6.

[164] T. A. Johansen et T. I. Fossen, "Control allocation—a survey," *Automatica*, vol. 49, n°. 5, p. 1087–1103, 2013.

[165] A. Nemirovskii, "Several np-hard problems arising in robust stability analysis," *Mathematics of Control, Signals and Systems*, vol. 6, n°. 2, p. 99–105, 1993.

[166] S. Poljak et J. Rohn, "Checking robust nonsingularity is np-hard," *Mathematics of Control, Signals and Systems*, vol. 6, n°. 1, p. 1–9, 1993.

[167] O. Toker, "On the complexity of the robust stability problem for linear parameter varying systems," *IFAC Proceedings Volumes*, vol. 29, n°. 1, p. 3450–3454, 1996.

[168] V. D. Blondel et J. N. Tsitsiklis, "A survey of computational complexity results in systems and control," *Automatica*, vol. 36, n°. 9, p. 1249–1274, 2000.

[169] J. Doyle, B. Francis et A. Tannenbaum, *Feedback Control Theory*. Dover Publications, 2013.

[170] K. Zhou et J. Doyle, *Essentials of Robust Control*. Prentice Hall, 1998.

[171] V. L. Kharitonov, "Asymptotic stability of an equilibrium position of a family of systems of linear differential equations," *Differential'nye Uravneniya*, vol. 14, p. 2086–2088, 1978.

[172] B. Barmish, *New Tools for Robustness of Linear Systems*. MacMillan, 1994.

[173] S. P. Bhattacharyya et L. H. Keel, *Robust control : the parametric approach*. Elsevier, 1995, p. 49–52.

[174] A. C. Bartlett, C. V. Hollot et H. Lin, "Root locations of an entire polytope of polynomials : It suffices to check the edges," *Mathematics of Control, Signals and Systems*, vol. 1, n°. 1, p. 61–71, 1988.

[175] C. B. Soh, "Necessary and sufficient conditions for stability of symmetric interval matrices," *International Journal of Control*, vol. 51, n°. 1, p. 243–248, 1990.

[176] D. Hertz, "The extreme eigenvalues and stability of real symmetric interval matrices," *IEEE Transactions on Automatic Control*, vol. 37, n°. 4, p. 532–535, 1992.

[177] J. Rohn, "Positive definiteness and stability of interval matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 15, n°. 1, p. 175–184, 1994.

[178] M. Mansour, "Robust stability of interval matrices," dans *Proceedings of the 28th IEEE Conference on Decision and Control*, p. 46–51 vol.1.

[179] C.-L. Jiang, "Sufficient condition for the asymptotic stability of interval matrices," *International Journal of Control*, vol. 46, n°. 5, p. 1803–1810, 1987.

[180] M. Mansour, "Simplified sufficient conditions for the asymptotic stability of interval matrices," *International Journal of Control*, vol. 50, n°. 1, p. 443–444, 1989.

[181] W. Kaining et A. N. Michel, "On sufficient conditions for the stability of interval matrices," *Systems & Control Letters*, vol. 20, n°. 5, p. 345–351, 1993.

[182] J. Chen, "Sufficient conditions on stability of interval matrices : connections and new results," *IEEE Transactions on Automatic Control*, vol. 37, n°. 4, p. 541–544, 1992.

[183] M. E. Sezer et D. D. Siljak, "On stability of interval matrices," *IEEE Transactions on Automatic Control*, vol. 39, n°. 2, p. 368–371, 1994.

[184] W. Kaining, A. N. Michel et L. Derong, "Necessary and sufficient conditions for the hurwitz and schur stability of interval matrices," *IEEE Transactions on Automatic Control*, vol. 39, n°. 6, p. 1251–1255, 1994.

[185] X. Yang et R. Unbehauen, "Robust hurwitz and schur stability test for interval matrices," dans *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, vol. 5, p. 4209–4214 vol.5.

[186] J. D. Cobb et C. L. Demarco, "The minimal dimension of stable faces required to guarantee stability of a matrix polytope," *IEEE Transactions on Automatic Control*, vol. 34, n°. 9, p. 990–992, 1989.

[187] J. D. Cobb, "The minimal dimension of stable faces required to guarantee stability of a matrix polytope : D-stability," *IEEE Transactions on Automatic Control*, vol. 35, n°. 4, p. 469–473, 1990.

[188] C. L. DeMarco, V. Balakrishnan et S. Boyd, "A branch and bound methodology for matrix polytope stability problems arising in power systems," dans *29th IEEE Conference on Decision and Control*, p. 3022–3027 vol.6.

[189] J. Rohn, "An algorithm for checking stability of symmetric interval matrices," *IEEE Transactions on Automatic Control*, vol. 41, n°. 1, p. 133–136, 1996.

[190] T. Alamo *et al.*, "A new vertex result for robustness problems with interval matrix uncertainty," *Systems & Control Letters*, vol. 57, n°. 6, p. 474–481, 2008.

[191] R. Yedavalli et N. Devarakonda, "Sufficiency of vertex matrix check for robust stability of interval matrices via the concept of qualitative robustness," dans *2013 European Control Conference*, 2013, p. 2351–2356.

[192] L. Saydy, A. Tits et E. Abed, "Guardian maps and the generalized stability of parametrized families of matrices and polynomials," *Mathematics of Control Signals and Systems*, vol. 3, p. 345–371, 1990.

[193] G. Chesi *et al.*, *Homogeneous polynomial forms for robustness analysis of uncertain systems*. Springer Science & Business Media, 2009, vol. 390.

[194] G. Chesi, "LMI techniques for optimization over polynomials in control : a survey," *IEEE Transactions on Automatic Control*, vol. 55, nº. 11, p. 2500–2510, 2010.

[195] ——, "Sufficient and necessary LMI conditions for robust stability of rationally time-varying uncertain systems," *IEEE Transactions on Automatic Control*, vol. 58, nº. 6, p. 1546–1551, 2013.

[196] W. Xiang, "Parameter-memorized Lyapunov functions for discrete-time systems with time-varying parametric uncertainties," *Automatica*, vol. 87, p. 450–454, 2018.

[197] P. B. Cox, S. Weiland et R. Tóth, "Affine parameter-dependent Lyapunov functions for LPV systems with affine dependence," *IEEE Transactions on Automatic Control*, vol. 63, nº. 11, p. 3865–3872, 2018.

[198] D. Henrion *et al.*, "On parameter-dependent Lyapunov functions for robust stability of linear systems," dans *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 1. IEEE, p. 887–892.

[199] G. Chesi, "Instability analysis of uncertain systems via determinants and LMIs," *IEEE Transactions on Automatic Control*, vol. 60, nº. 9, p. 2458–2463, 2015.

[200] ——, "On the complexity of robust stability analysis of polytopic LTI systems," dans *2018 IEEE Workshop on Complexity in Engineering (COMPENG)*, p. 1–5.

[201] J. Doyle, A. Packard et K. Zhou, "Review of LFTs, LMIs, and μ," 1991.

[202] S.-J. Chen, S.-P. Yang et L.-G. Shiau, "Guardian map approach to robust stability of interval systems," *International Journal of Systems Science*, vol. 43, nº. 12, p. 2193–2201, 2012.

[203] B. Jiang et F. N. Chowdhury, "Fault estimation and accommodation for linear MIMO discrete-time systems," *IEEE Transactions on Control Systems Technology*, vol. 13, nº. 3, p. 493–499, May 2005.

[204] C. H. Lee et M. K. Park, "Actuator fault estimation method using hexacopter symmetry," *Journal of Institute of Control, Robotics and Systems*, vol. 22, p. 519–523, 2016.

[205] D.-T. Nguyen, D. Saussié et L. Saydy, "Robust self-scheduled fault-tolerant control of a quadrotor UAV," dans *20th World Congress of the International Federation of Automatic Control (IFAC)*, 2017, p. 5761 – 5767.

[206] ——, "Quaternion-based robust fault-tolerant control of a quadrotor UAV," dans *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, p. 1333–1342.

[207] ——, "Fault-tolerant control of a hexacopter UAV based on self-scheduled control allocation," dans *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, p. 385–393.

[208] ——, "Universal adaptive fault-tolerant control of a multicopter UAV," dans *21th World Congress of the International Federation of Automatic Control (IFAC)*, 2020.

[209] ——, "Design and experimental validation of robust self-scheduled fault-tolerant control laws for a multicopter UAV," *IEEE/ASME Transactions on Mechatronics*, 2020.

[210] ——, "Necessary and sufficient conditions for robust stability of uncertain affine/rational systems based on guardian maps," article de revue à soumettre, 2021.

[211] A. Borowczyk *et al.*, "Autonomous landing of a multirotor micro air vehicle on a hign velocity ground vehicle," dans *20th World Congress of the International Federation of Automatic Control (IFAC)*, 2017, p. 10 488 – 10 494.

[212] ——, "Autonomous landing of a quadcopter on a high-speed ground vehicle," *Journal of Guidance, Control, and Dynamics*, p. 1–8, 2017.

[213] C. Massé *et al.*, "Modeling and control of a quadcopter flying in a wind field : A comparison between lqr and structured $\mathcal{H}_\infty$ control techniques," dans *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, p. 1408–1417.

[214] J. Farrell, *Aided Navigation : GPS with High Rate Sensors*. McGraw-Hill Education, 2008.

[215] A. de Ruiter, C. Damaren et J. Forbes, *Spacecraft Dynamics and Control : An Introduction*. Wiley, 2012.

[216] B. Stevens, F. Lewis et E. Johnson, *Aircraft Control and Simulation : Dynamics, Controls Design, and Autonomous Systems*. Wiley, 2015.

[217] R. M. Murray, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[218] B. Stevens, F. Lewis et E. Johnson, *Aircraft Control and Simulation : Dynamics, Controls Design, and Autonomous Systems*, 3$^{\text{e}}$ éd. Wiley, 2015.

[219] D. Eberly, "Rotation representations and performance issues," *Magic Software, Inc., Chapel Hill, NC*, 2002.

[220] V. Artale, C. L. R. Milazzo et A. Ricciardello, "An example of quaternion parameterization for dynamical simulations," dans *Journal of Physics : Conference Series*, vol. 490. Institute of Physics Publishing, 2014.

[221] B. D. O. Anderson et J. B. Moore, *Optimal control*. Englewood, NJ : Prentice-Hall, 1990.

[222] R. E. Kálmán *et al.*, "Contributions to the theory of optimal control," *Bol. Soc. Mat. Mexicana*, vol. 5, n°. 2, p. 102–119, 1960.

[223] F. Furrer *et al.*, *Robot Operating System (ROS) : The Complete Reference (Volume 1)*. Cham : Springer International Publishing, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework, p. 595–625.

[224] Motion capture systems from Vicon. [En ligne]. Disponible : https://www.vicon.com/products/archived-products/bonita.

[225] C. Powers, D. Mellinger et V. Kumar, *Quadrotor Kinematics and Dynamics*. Dordrecht : Springer Netherlands, 2015, p. 307–328.

[226] C. T. Raabe, "Adaptive, failure-tolerant control for hexacopters," dans *AIAA Infotech@Aerospace (I@A) Conference*, 2013.

[227] Y. M. Zhang, "Lecture notes on fault diagnosis and fault tolerant control systems," Jan 2017.

[228] P. Zarchan et H. Musoff, *Fundamentals of Kalman Filtering : A Practical Approach*. AIAA, Inc., 2000.

[229] H. Khalil, *Nonlinear Systems*, 3e éd. Prentice Hall, 2002.

[230] E. Lavretsky et T. E. Gibson, "Projection operator in adaptive systems," *arXiv preprint arXiv :1112.4232*, 2011.

[231] J. Doyle, "Analysis of feedback systems with structured uncertainties," dans *IEE Proceedings D-Control Theory and Applications*, vol. 129, n°. 6. IET, 1982, p. 242–250.

[232] B. G. Morton, "New applications of mu to real-parameter variation problems," dans *1985 24th IEEE Conference on Decision and Control*, 1985, p. 233–238.

[233] L. Elsner et V. Monov, "The bialternate matrix product revisited," *Linear algebra and its applications*, vol. 434, n°. 4, p. 1058–1066, 2011.

[234] J.-F. Magni, "Linear fractional representation toolbox for use with MATLAB," Report, 2006. [En ligne]. Disponible : http://w3.onera.fr/smac/lfrt

[235] D. D. Siljak, "Parameter space methods for robust control design : a guided tour," dans *1988 American Control Conference*. IEEE, p. 783–783.

[236] R. Shuoh, P. T. Kabamba et D. S. Bernstein, "Guardian map approach to robust stability of linear systems with constant real parameter uncertainty," *IEEE Transactions on Automatic Control*, vol. 39, n°. 1, p. 162–164, 1994.

[237] C. Scherer et S. Weiland, "Linear matrix inequalities in control," 2000.

[238] E. Bulka et M. Nahon, "A universal controller for unmanned aerial vehicles," dans *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 4171–4176.

[239] J. Doyle *et al.*, "State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems," dans *1988 American Control Conference*. IEEE, 1988, p. 1691–1696.

[240] P. Gahinet et P. Apkarian, "A linear matrix inequality approach to $\mathcal{H}_\infty$ control," *International journal of robust and nonlinear control*, vol. 4, n°. 4, p. 421–448, 1994.

[241] ——, "Decentralized and fixed-structure $\mathcal{H}_\infty$ control in MATLAB," *50th IEEE Conference on Decision and Control and European Control Conference (Cdc-Ecc)*, p. 8205–8210, 2011.

[242] ——, "Structured $\mathcal{H}_\infty$ synthesis in MATLAB," *IFAC Proceedings Volumes*, vol. 44, n°. 1, p. 1435–1440, 2011.

[243] M. G. Safonov, K.-C. Goh et J. Ly, "Control system synthesis via bilinear matrix inequalities," dans *Proceedings of 1994 American Control Conference-ACC'94*, vol. 1. IEEE, 1994, p. 45–49.

[244] S. Boyd, S. P. Boyd et L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[245] J.-L. Lin et S.-J. Chen, "LFT approach to robust D-stability bounds of uncertain linear singular systems," *IEE Proceedings-Control Theory and Applications*, vol. 145, n°. 2, p. 127–134, 1998.

[246] L. El Ghaoui et G. Scorletti, "Control of rational systems using linear-fractional representations and linear matrix inequalities," *Automatica*, vol. 32, n°. 9, p. 1273–1284, 1996.

ANNEXE A MODÉLISATION ET COMMANDE DE MULTICOPTÈRES

A.1 Conversion entre les différentes représentations de rotation

A.1.1 Angles d'Euler - Matrice de rotation

Les matrices de rotation principales sont données par :

$$\mathbf{R}_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, \quad \mathbf{R}_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, \quad \mathbf{R}_{z,\psi} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

En conséquence, la matrice de rotation correspondant aux trois angles d'Euler $\mathbf{\Phi} = [\phi, \theta, \psi]^\top$ en ordre z-y-x est donnée par :

$$\mathbf{R}_{b/i} = \mathbf{R}_{x,\phi}\mathbf{R}_{y,\theta}\mathbf{R}_{z,\psi} = \begin{bmatrix} \mathrm{c}_\theta\mathrm{c}_\psi & \mathrm{c}_\theta\mathrm{s}_\psi & -\mathrm{s}_\theta \\ \mathrm{s}_\phi\mathrm{s}_\theta\mathrm{c}_\psi - \mathrm{c}_\phi\mathrm{s}_\psi & \mathrm{s}_\phi\mathrm{s}_\theta\mathrm{s}_\psi + \mathrm{c}_\phi\mathrm{c}_\psi & \mathrm{s}_\phi\mathrm{c}_\theta \\ \mathrm{c}_\phi\mathrm{s}_\theta\mathrm{c}_\psi + \mathrm{s}_\phi\mathrm{s}_\psi & \mathrm{c}_\phi\mathrm{s}_\theta\mathrm{s}_\psi - \mathrm{s}_\phi\mathrm{c}_\psi & \mathrm{c}_\phi\mathrm{c}_\theta \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{23} & r_{33} \end{bmatrix} \tag{A.1}$$

où $\mathrm{c}_x = \cos(x)$ et $\mathrm{s}_x = \sin(x)$. Ainsi, on peut retrouver les angles d'Euler à partir de la matrice de rotation :

$$\phi = \arctan 2(r_{23}, r_{33}), \quad \theta = -\arcsin(r_{13}) \quad \text{et} \quad \psi = \arctan 2(r_{12}, r_{11}) \tag{A.2}$$

Notons que cette solution, par convention, est obtenue en supposant que $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$. Comme dit précédemment, la paramétrisation d'Euler présente une singularité en $\theta = \pm\dfrac{\pi}{2}$.

A.1.2 Quaternion - Matrice de rotation

La matrice de rotation correspondant au quaternion $\mathbf{q}$ est :

$$\mathbf{R}_{b/i} = C(\mathbf{q}) = (2q_0^2 - 1)\mathbf{I}_3 + 2\mathbf{q_{13}}\mathbf{q_{13}}^\top - 2q_0[\mathbf{q_{13}}]_\times \tag{A.3}$$

$$= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{A.4}$$

On peut également retrouver le quaternion à partir de la matrice de rotation :

$$q_0 = \pm\frac{\sqrt{r_{11}+r_{22}+r_{33}+1}}{2} = \pm\frac{\sqrt{\text{trace}(\mathbf{R}_{b/i})+1}}{2} \tag{A.5}$$

$$\text{si } q_0 \neq 0 \ : \quad q_1 = \frac{(r_{23}-r_{32})}{4q_0},\ q_2 = \frac{(r_{31}-r_{13})}{4q_0},\ q_3 = \frac{(r_{12}-r_{21})}{4q_0} \tag{A.6}$$

Notons que $q_0 = 0$ correspond à une rotation d'angle $\phi_a = \pm\pi$ (voir la définition de quaternion en équation 3.6). Dans ce cas, la matrice de rotation devient :

$$\mathbf{R}_{b/i} = -\mathbf{I}_3 + 2\mathbf{q_{13}}\mathbf{q_{13}}^\top$$

La partie vectorielle du quaternion $\mathbf{q}$ est alors déterminée par :

$$|q_1| = \sqrt{\frac{r_{11}+1}{2}},\ \ |q_2| = \sqrt{\frac{r_{22}+1}{2}},\ \ |q_3| = \sqrt{\frac{r_{33}+1}{2}} \tag{A.7}$$

A.1.3 Angles d'Euler - Quaternion

Les quaternions unitaires qui représentent les rotations principales autour d'un axe sont donnés par :

$$\mathbf{q}_{x,\phi} = \begin{bmatrix} \cos(\phi/2) \\ \sin(\phi/2) \\ 0 \\ 0 \end{bmatrix},\ \ \mathbf{q}_{y,\theta} = \begin{bmatrix} \cos(\theta/2) \\ 0 \\ \sin(\theta/2) \\ 0 \end{bmatrix},\ \ \mathbf{q}_{z,\psi} = \begin{bmatrix} \cos(\psi/2) \\ 0 \\ 0 \\ \sin(\psi/2) \end{bmatrix}$$

Le quaternion correspondant aux angles d'Euler $\mathbf{\Phi} = [\phi, \theta, \psi]^\top$ en séquence z-y-x est donc :

$$\mathbf{q}_{b/i} = \mathbf{q}_{z,\psi} * \mathbf{q}_{y,\theta} * \mathbf{q}_{x,\phi} = \pm \begin{bmatrix} c\frac{\phi}{2}c\frac{\theta}{2}c\frac{\psi}{2} + s\frac{\phi}{2}s\frac{\theta}{2}s\frac{\psi}{2} \\ s\frac{\phi}{2}c\frac{\theta}{2}c\frac{\psi}{2} - c\frac{\phi}{2}s\frac{\theta}{2}s\frac{\psi}{2} \\ c\frac{\phi}{2}s\frac{\theta}{2}c\frac{\psi}{2} + s\frac{\phi}{2}c\frac{\theta}{2}s\frac{\psi}{2} \\ c\frac{\phi}{2}c\frac{\theta}{2}s\frac{\psi}{2} - s\frac{\phi}{2}s\frac{\theta}{2}c\frac{\psi}{2} \end{bmatrix} \tag{A.8}$$

Notons que contrairement aux matrices de rotations, on utilise l'ordre inverse lors de la multiplication des quaternions pour une séquence de rotations ([216]).

Finalement, les angles d'Euler peuvent être retrouvés via la matrice de rotation $\mathbf{R}_{b/i} = C(\mathbf{q})$ (voir équations A.2 et A.3).

A.2 Paramètres de multicoptères

A.2.1 Quadricoptère

Tableau A.1 Paramètres du quadricoptère

Paramètres	Description	Valeur
m	Masse	$0.58\,\mathrm{kg}$
d	Longueur du bras	$0.25\,\mathrm{m}$
(I_{xx}, I_{yy}, I_{zz})	Moments d'inertie	$(0.01,\ 0.01,\ 0.01)\,\mathrm{kg{\cdot}m^2}$
k_T	Coefficient de poussée	$7.72\text{e-}06\,\mathrm{N{\cdot}s^2/rad^2}$
k_D	Coefficient de traînée	$2.82\text{e-}07\,\mathrm{N{\cdot}s^2/rad^2}$
$T_{i,max}$	Poussée maximale	$3.75\,\mathrm{N}$

A.2.2 Hexacoptère

Tableau A.2 Paramètres de l'Asctec Firefly

Paramètres	Description	Valeur
m	Masse	$1.54\,\mathrm{kg}$
d	Longueur du bras	$0.215\,\mathrm{m}$
(I_{xx}, I_{yy}, I_{zz})	Moments d'inertie	$(0.0348,\ 0.0459,\ 0.0977)\,\mathrm{kg{\cdot}m^2}$
k_T	Coefficient de poussée	$8.5486\text{e-}06\,\mathrm{N{\cdot}s^2/rad^2}$
k_D	Coefficient de traînée	$3.1202\text{e-}07\,\mathrm{N{\cdot}s^2/rad^2}$
$T_{i,max}$	Poussée maximale	$9.3746\,\mathrm{N}$
$\omega_{i,max}$	Vitesse maximale	$10000\,\mathrm{rpm}$

A.3 Contrôleur nominal pour l'Astec Firefly

En appliquant la synthèse LQR, les gains du contrôleur nominal obtenus sont :

$$\mathbf{K_x} = \begin{bmatrix} 14.72 & 6.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.53 & -0.39 & 1.58 & 0.33 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.54 & 0.41 & 1.76 & 0.40 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.50 & 0.31 \end{bmatrix} \tag{A.9}$$

$$\mathbf{K_z} = \mathrm{diag}\,(12.2474,\ -0.3162,\ 0.3162,\ 0.3162)\,.$$

ANNEXE B COMMANDE TOLÉRANTE AUX FAUTES

B.1 Filtre de Kalman à deux étages

Soit le système LTI discret

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k - \mathbf{B}_k\mathbf{U}_k\boldsymbol{\gamma}_k + \mathbf{w}_k^{\mathbf{x}} \\ \boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k + \mathbf{w}_k^{\gamma} \\ \mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k \end{cases}$$

où $\mathbf{u}_k \in \mathbb{R}^m$, $\mathbf{x}_k \in \mathbb{R}^n$ et $\mathbf{y}_k \in \mathbb{R}^p$ représentent respectivement les vecteurs d'entrée, d'état et de sortie mesurable du système. Les fautes des actionneurs sont modélisées par le terme $-\mathbf{B}_k\mathbf{U}_k\boldsymbol{\gamma}_k$ tel que mentionné dans la sous-section 4.1.1. Les perturbations $\mathbf{w}_k^{\mathbf{x}} \in \mathbb{R}^n, \mathbf{w}_k^{\gamma} \in \mathbb{R}^m$ et l'erreur de mesure $\mathbf{v}_k \in \mathbb{R}^p$ sont des bruits gaussiens et indépendants. Les équations du filtre de Kalman à deux étages [35] sont regroupées de la façon suivante :

Estimation d'état

$$\begin{cases} \tilde{\mathbf{x}}_{k+1|k} = \mathbf{A}_k\tilde{\mathbf{x}}_{k|k} + \mathbf{B}_k\mathbf{u}_k + \mathbf{W}_k\tilde{\boldsymbol{\gamma}}_{k|k} - \mathbf{V}_{k+1|k}\boldsymbol{\gamma}_{k|k} \\ \tilde{\mathbf{P}}^{\mathbf{x}}_{k+1|k} = \mathbf{A}_k\tilde{\mathbf{P}}^{\mathbf{x}}_{k|k}\mathbf{A}_k^\top + \mathbf{Q}_k^{\mathbf{x}} + \mathbf{W}_k\mathbf{P}^{\gamma}_{k|k}\mathbf{W}_k^\top - \mathbf{V}_{k+1|k}\mathbf{P}^{\gamma}_{k+1|k}\mathbf{V}_{k+1|k}^\top \\ \tilde{\mathbf{x}}_{k+1|k+1} = \tilde{\mathbf{x}}_{k+1|k} + \tilde{\mathbf{K}}^{\mathbf{x}}_{k+1}\left(\mathbf{y}_{k+1} - \mathbf{C}_{k+1}\tilde{\mathbf{x}}_{k+1|k}\right) \\ \tilde{\mathbf{K}}^{\mathbf{x}}_{k+1} = \tilde{\mathbf{P}}^{\mathbf{x}}_{k+1|k}\mathbf{C}_{k+1}^\top\left(\mathbf{C}_{k+1}\tilde{\mathbf{P}}^{\mathbf{x}}_{k+1|k}\mathbf{C}_{k+1}^\top + \mathbf{R}_{k+1}\right)^{-1} \\ \tilde{\mathbf{P}}^{\mathbf{x}}_{k+1|k+1} = \left(\mathbf{I} - \tilde{\mathbf{K}}^{\mathbf{x}}_{k+1}\mathbf{C}_{k+1}\right)\tilde{\mathbf{P}}^{\mathbf{x}}_{k+1|k} \end{cases} \tag{B.1}$$

Génération de résidus

$$\begin{cases} \tilde{\mathbf{r}}_{k+1} = \mathbf{y}_{k+1} - \mathbf{C}_{k+1}\tilde{\mathbf{x}}_{k+1|k} \\ \tilde{\mathbf{S}}_{k+1} = \mathbf{C}_{k+1}\tilde{\mathbf{P}}^{\mathbf{x}}_{k+1|k}\mathbf{C}_{k+1}^\top + \mathbf{R}_{k+1} \end{cases} \tag{B.2}$$

Estimation de biais

$$\begin{cases} \hat{\boldsymbol{\gamma}}_{k+1|k} = \hat{\boldsymbol{\gamma}}_{k|k} \\ \mathbf{P}^{\gamma}_{k+1|k} = \mathbf{P}^{\gamma}_{k|k} + \mathbf{Q}_k^{\gamma} \\ \hat{\boldsymbol{\gamma}}_{k+1|k+1} = \hat{\boldsymbol{\gamma}}_{k+1|k} + \mathbf{K}^{\gamma}_{k+1}\left(\tilde{\mathbf{r}}_{k+1} - \mathbf{H}_{k+1|k}\hat{\boldsymbol{\gamma}}_{k+1|k}\right) \\ \mathbf{K}^{\gamma}_{k+1} = \mathbf{P}^{\gamma}_{k+1|k}\mathbf{H}_{k+1|k}^\top\left(\mathbf{H}_{k+1|k}\mathbf{P}^{\gamma}_{k+1|k}\mathbf{H}_{k+1|k}^\top + \tilde{\mathbf{S}}_{k+1}\right)^{-1} \\ \mathbf{P}^{\gamma}_{k+1|k+1} = \left(\mathbf{I} - \mathbf{K}^{\gamma}_{k+1}\mathbf{H}_{k+1|k}\right)\mathbf{P}^{\gamma}_{k+1|k} \end{cases} \tag{B.3}$$

Couplage

$$\begin{cases} \mathbf{W}_k = \mathbf{A}_k \mathbf{V}_{k|k} - \mathbf{B}_k \mathbf{U}_k \\ \mathbf{V}_{k+1|k} = \mathbf{W}_k \mathbf{P}^{\gamma}_{k|k} \left(\mathbf{P}^{\gamma}_{k+1|k} \right)^{-1} \\ \mathbf{H}_{k+1|k} = \mathbf{C}_{k+1} \mathbf{V}_{k+1|k} \\ \mathbf{V}_{k+1|k+1} = \mathbf{V}_{k+1|k} - \tilde{\mathbf{K}}^{\mathbf{x}}_{k+1} \mathbf{H}_{k+1|k} \end{cases} \tag{B.4}$$

Compensation de biais

$$\begin{cases} \hat{\mathbf{x}}_{k+1|k+1} = \mathbf{x}_{k+\tilde{1}|k+1} + \mathbf{V}_{k+1|k+1} \hat{\boldsymbol{\gamma}}_{k+1|k+1} \\ \mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+\tilde{1}|k+1} + \mathbf{V}_{k+1|k+1} \mathbf{P}^{\gamma}_{k+1|k+1} \mathbf{V}_{k+1|k+1} \end{cases} \tag{B.5}$$

Similairement au filtre de Kalman, l'indice $k+1|k$ indique l'étape de prédiction, alors que l'indice $k+1|k+1$ indique la phase de mise-à-jour.

B.2 Rappels sur la synthèse $\mathcal{H}_\infty$

Cette section est consacrée à des rappels théoriques sur la synthèse $\mathcal{H}_\infty$ classique et structurée ainsi que les outils utilisés pour effectuer la synthèse de contrôleurs robustes.

B.2.1 Contraintes $\mathcal{H}_\infty$ pour asservissements

La norme $\mathcal{H}_\infty$ d'une matrice $\mathbf{G}(s)$ de fonctions de transferts rationnels réels stables et propres entre l'entrée $\mathbf{u}$ d'énergie finie et la sortie $\mathbf{y}$ est définie par :

$$\|\mathbf{G}\|_\infty = \sup_{\omega \in \mathbb{R}} \bar{\sigma} \left(\mathbf{G}(j\omega) \right) = \sup_{0 < \|u\|_2 < \infty} \frac{\|\mathbf{y}\|_2}{\|\mathbf{u}\|_2} \tag{B.6}$$

où $\bar{\sigma} \left(\mathbf{G}(j\omega) \right)$ désigne la valeur singulière maximale de $\mathbf{G}(j\omega)$ et $\|\cdot\|_2$ représente la norme $\mathcal{L}_2$ des signaux. Ainsi, la norme $\mathcal{H}_\infty$ correspond à la valeur maximale du rapport entre l'énergie du signal de sortie et celle du signal d'entrée.

Dans le cadre de la synthèse $\mathcal{H}_\infty$, il s'agit d'abord de transposer les contraintes imposées par le cahier des charges dans le formalisme $\mathcal{H}_\infty$. Considérons le système multivariable en boucle fermée comme indiqué à la figure B.1. Les fonctions de transfert d'intérêt sont les suivantes :

- Fonctions de sensibilité en sortie : $\mathbf{S} = (\mathbf{I} + \mathbf{GK})^{-1}$;
- Fonctions de sensibilité complémentaire en sortie : $\mathbf{T} = \mathbf{GK} (\mathbf{I} + \mathbf{GK})^{-1}$.

En notant que $\mathbf{e} = \mathbf{Sr}$, $\mathbf{u} = \mathbf{KSr}$ (si $\mathbf{w}_i = \mathbf{w}_o = \mathbf{n} = 0$) et que $\mathbf{y} = -\mathbf{Tn}$ (si $\mathbf{r} = \mathbf{w}_i = \mathbf{w}_o = 0$), on peut donc traduire les spécifications de design en contraintes $\mathcal{H}_\infty$ comme :

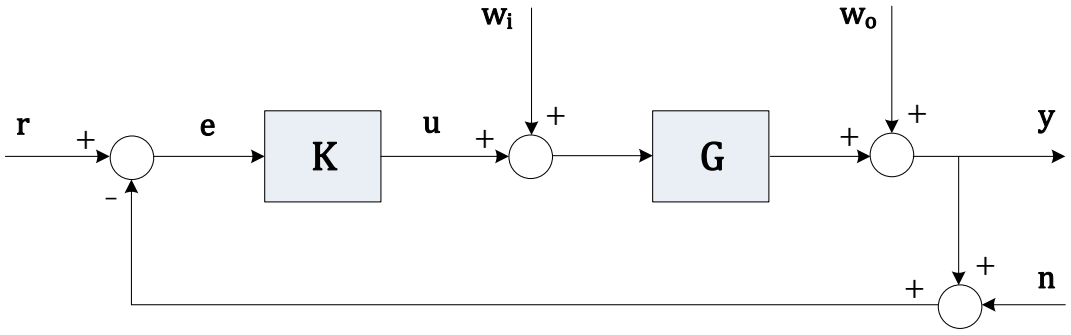


Figure B.1 Système multivariable en boucle fermée

— Performance en suivi : minimiser $\|\mathbf{S}\|_\infty$ pour obtenir une erreur de suivi $\mathbf{e}$ faible, principalement aux basses fréquences ;

— Effort de commande : minimiser $\|\mathbf{KS}\|_\infty$ pour éviter la sollicitation aux hautes fréquences et aussi la saturation de l'actionneur ;

— Robustesse quant au bruit de mesure : minimiser $\|\mathbf{T}\|_\infty$ pour atténuer l'influence du bruit $\mathbf{n}$ sur la sortie $\mathbf{y}$, principalement aux hautes fréquences.

Il est bien évidemment possible de définir d'autres critères (rejet de perturbations, marge de stabilité) en contraintes $\mathcal{H}_\infty$. On renvoie le lecteur pour plus de détails à [170].

De manière à adopter les caractéristiques physiques de différents systèmes, on peut utiliser les fonctions de pondération. Les contraintes normalisées se mettent sous la forme :

$$\|\mathbf{W}_1\mathbf{S}\|_\infty < 1, \quad \|\mathbf{W}_2\mathbf{KS}\|_\infty < 1 \text{ et } \quad \|\mathbf{W}_3\mathbf{T}\|_\infty < 1, \tag{B.7}$$

où $\mathbf{W}_1$ est un filtre de type passe-bas et $\mathbf{W}_2, \mathbf{W}_3$ des filtres de type passe-haut.

En notant que $\mathbf{S} + \mathbf{T} = \mathbf{I}$, on cherche de plus à trouver le meilleur compromis entre la performance et la robustesse. Il en résulte une procédure itérative pour ajuster au mieux les pondérations fréquentielles.

B.2.2 Synthèse $\mathcal{H}_\infty$ classique

B.2.2.1 Problème $\mathcal{H}_\infty$ standard

Soit le système augmenté $\mathbf{P}(s)$ avec l'entrée de commande $\mathbf{u}$, les entrées exogènes $\mathbf{w}$ (consignes ou perturbations), les sorties mesurées $\mathbf{y}$ et les sorties régulées $\mathbf{z}$ (Fig. B.2). $\mathbf{P}(s)$ est décom-

posé de la façon suivante :

$$\begin{bmatrix} \mathbf{Z}(s) \\ \mathbf{Y}(\mathbf{s}) \end{bmatrix} = \mathbf{P}(s) \begin{bmatrix} \mathbf{W}(s) \\ \mathbf{U}(\mathbf{s}) \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11}(s) & \mathbf{P}_{12}(s) \\ \mathbf{P}_{21}(s) & \mathbf{P}_{22}(s) \end{bmatrix} \begin{bmatrix} \mathbf{W}(s) \\ \mathbf{U}(\mathbf{s}) \end{bmatrix} \tag{B.8}$$

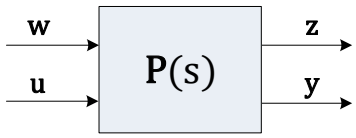


Figure B.2 Système augmenté

On reboucle le système augmenté par la loi de commande $\mathbf{U}(s) = \mathbf{K}(s)\mathbf{Y}(s)$ (Fig. B.3). La fonction de transfert entre les entrées exogènes $\mathbf{w}$ et les sorties régulées $\mathbf{z}$ est alors donnée sous la forme d'une LFT inférieure :

$$\mathbf{T}_{\mathbf{w}\to\mathbf{z}}(s) = \mathbf{T}_{\mathbf{zw}}(s) = \mathcal{F}_l(\mathbf{P}, \mathbf{K}) = \mathbf{P}_{11}(s) + \mathbf{P}_{12}(s)\mathbf{K}(s)(\mathbf{I} - \mathbf{P}_{22}(s)\mathbf{K}(s))^{-1}\mathbf{P}_{21}(s) \tag{B.9}$$

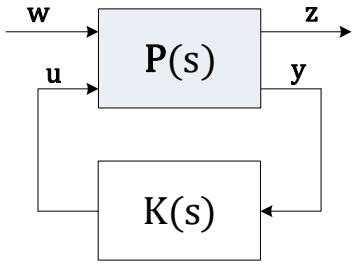


Figure B.3 Forme standard du système bouclé

De façon générale, l'objectif du problème de synthèse $\mathcal{H}_\infty$ [231] est de trouver un contrôleur $\mathbf{K}(s)$ stabilisant internement le système et minimisant l'impact des perturbations $\mathbf{w}$ sur les sorties $\mathbf{y}$, soit la norme $\mathcal{H}_\infty$ de la fonction de transfert $\mathbf{T}_{\mathbf{zw}}(s) = \mathcal{F}_l(\mathbf{P}(s), \mathbf{K}(s))$.

Problème B.2.1. *Le problème $\mathcal{H}_\infty$ optimal consiste à trouver un contrôleur stabilisant $\mathbf{K}(s)$ tel que $\|\mathbf{T}_{\mathbf{zw}}(s)\|_\infty = \|\mathcal{F}_l(\mathbf{P}(s), \mathbf{K}(s))\|_\infty$ soit minimale.*

La norme $\mathcal{H}_\infty$ optimale, notée γ_{opt}, est une mesure de la limite de performance atteignable du

système bouclé. En pratique, on s'intéresse généralement à réduire la norme $\mathcal{H}_\infty$ en dessous d'un seuil positif $\gamma > \gamma_{opt}$.

Problème B.2.2. *Le problème $\mathcal{H}_\infty$ sous-optimal consiste à trouver un contrôleur stabilisant $\mathbf{K}(s)$ tel que $\|\mathbf{T_{zw}}(s)\|_\infty = \|\mathcal{F}_l(\mathbf{P}(s), \mathbf{K}(s))\|_\infty < \gamma$ avec $\gamma > \gamma_{opt}$.*

B.2.2.2 Mise en forme du problème $\mathcal{H}_\infty$ standard

On présente ici la reformulation sous forme standard du problème de sensibilité mixte souvent rencontré dans le cadre de la synthèse robuste. Considérons le schéma bloc de la figure B.4 qui représente à la fois le système bouclé et le cahier des charges formulé via les fonctions de pondération précédemment introduites (cf. Eq. (B.7)).

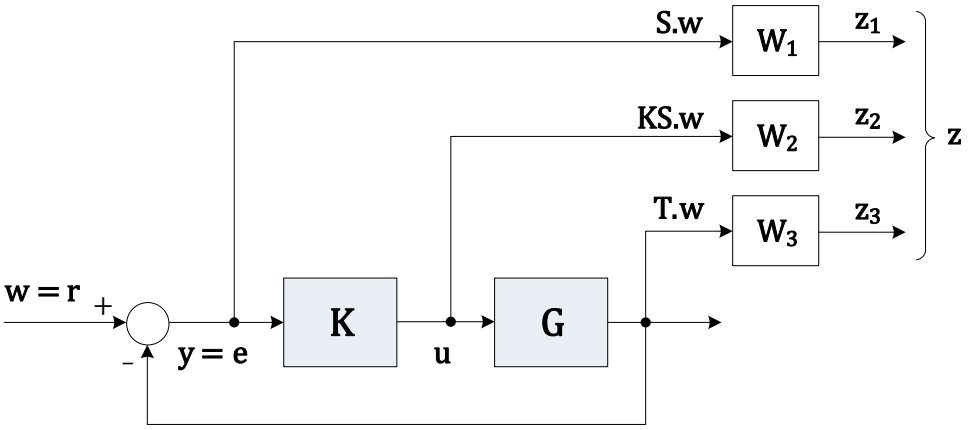


Figure B.4 Problème de sensibilité mixte

En définissant le vecteur des sorties régulées virtuelles $\mathbf{z} = [\mathbf{z}_1\ \mathbf{z}_2\ \mathbf{z}_3]^\top$ tel que montré sur la figure B.4, on obtient le formalisme standard (B.8) avec :

$$\mathbf{P}(s) = \begin{bmatrix} \mathbf{P}_{11}(s) & \mathbf{P}_{12}(s) \\ \mathbf{P}_{21}(s) & \mathbf{P}_{22}(s) \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{W}_1 & -\mathbf{W}_1\mathbf{G} \\ \mathbf{0} & \mathbf{W}_2 \\ \mathbf{0} & \mathbf{W}_3\mathbf{G} \\ \hdashline \mathbf{I} & -\mathbf{G} \end{array}\right] \tag{B.10}$$

On en déduit donc que :

$$\mathbf{T_{zw}}(s) = \mathcal{F}_l(\mathbf{P}, \mathbf{K}) = \mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(\mathbf{I} - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21} = \begin{bmatrix} \mathbf{W}_1\mathbf{S} \\ \mathbf{W}_2\mathbf{KS} \\ \mathbf{W}_3\mathbf{T} \end{bmatrix} \tag{B.11}$$

Le problème $\mathcal{H}_\infty$ sous-optimal correspondant est alors défini comme suit.

Problème B.2.3. *Le problème $\mathcal{H}_\infty$ sous-optimal de sensibilité mixte consiste à trouver un contrôleur stabilisant $\mathbf{K}(s)$ tel que*

$$\|\mathbf{T_{zw}}(s)\|_\infty = \|\mathcal{F}_l(\mathbf{P}(s), \mathbf{K}(s))\|_\infty = \left\| \begin{bmatrix} \mathbf{W}_1\mathbf{S} \\ \mathbf{W}_2\mathbf{KS} \\ \mathbf{W}_3\mathbf{T} \end{bmatrix} \right\|_\infty < 1. \tag{B.12}$$

Notons que la condition (B.12) implique que $\|\mathbf{W}_1\mathbf{S}\| < 1, \|\mathbf{W}_2\mathbf{KS}\| < 1$ et $\|\mathbf{W}_3\mathbf{T}\| < 1$.

B.2.2.3 Solution du problème $\mathcal{H}_\infty$ standard

Plusieurs méthodes existent pour réaliser la synthèse $\mathcal{H}_\infty$ classique. Celle qui semble, à l'heure actuelle, la plus adaptée numériquement a été proposée par [239]. Sous trois hypothèses principales et deux hypothèses simplificatrices, cette méthode, basée sur une approche par modèle d'état, permet de synthétiser un correcteur robuste qui est du même ordre que le système augmenté $\mathbf{P}(s)$. En appliquant la technique LMI, [240] a fourni une méthode moins restrictive au niveau des hypothèses. Du fait de la caractérisation LMI, l'algorithme proposé nécessite des temps de calcul plus importants pour des systèmes d'ordre élevé. Sous MATLAB, ces deux méthodes sont implantées dans la fonction `hinfsyn` de la *Robust Control Toolbox*.

B.2.2.4 Limitation de la synthèse $\mathcal{H}_\infty$ classique

La principale limitation des méthodes $\mathcal{H}_\infty$ classiques réside dans la structure et l'ordre du contrôleur. En effet, la synthèse $\mathcal{H}_\infty$ classique donne un contrôleur d'ordre plein sans structure particulière. L'implantation pratique d'un tel contrôleur nécessite souvent une réduction *a posteriori* de l'ordre du contrôleur tout en limitant la perte de performance associée.

La seconde limitation réside dans le fait que toutes les contraintes sont exprimées dans une seule matrice $\mathbf{T_{zw}} = \mathcal{F}_l(\mathbf{P}, \mathbf{K})$ liant l'ensemble des entrées exogènes à l'ensemble des sorties régulées. Il apparaît donc des contraintes additionnelles cachées dans les fonctions de transfert correspondant à des termes croisés inutiles. En minimisant la norme $\mathcal{H}_\infty$ de la matrice de transferts globale $\mathbf{T_{zw}}$, on minimise également celles des fonctions de transfert qui ne présentent aucun intérêt en termes de performance ou de robustesse à optimiser. Cela peut augmenter la norme $\mathcal{H}_\infty$ minimale atteignable, soit γ_{opt}.

B.2.3 Synthèse $\mathcal{H}_\infty$ structurée

B.2.3.1 Problème $\mathcal{H}_\infty$ structuré

La synthèse $\mathcal{H}_\infty$ structurée permet de choisir *a priori* la structure du correcteur tout en satisfaisant les mêmes contraintes $\mathcal{H}_\infty$. On considère la famille de contrôleurs $\mathbf{K}(s,\kappa)$ paramétrée par le vecteur $\kappa \in \mathcal{K} \subset \mathbb{R}^{n_\kappa}$, où n_κ est le nombre des paramètres ajustables du contrôleur. Une réalisation minimale [1] du correcteur $\mathbf{K}(s,\kappa)$ est alors donnée par :

$$\mathbf{K}(s,\kappa) = \mathbf{C}_K \left(s\mathbf{I} - \mathbf{A}_K\right)^{-1} \mathbf{B}_K + \mathbf{D}_K =: \left[\begin{array}{c|c} \mathbf{A}_K & \mathbf{B}_K \\ \hline \mathbf{C}_K & \mathbf{D}_K \end{array}\right] \tag{B.13}$$

Le contrôleur $\mathbf{K}(s,\kappa)$ est dit structuré si les matrices d'état $\mathbf{A}_K, \mathbf{B}_K, \mathbf{C}_K$ et $\mathbf{D}_K$ dépendent de façon lisse du vecteur de paramètres κ. Un correcteur PID ou une fonction de transfert stable d'ordre 2 sont des exemples typiques de contrôleurs structurés SISO.

Problème B.2.4. *Le problème $\mathcal{H}_\infty$ structuré consiste à trouver un contrôleur stabilisant $\mathbf{K}(s,\kappa^*)$ tel que $\|\mathcal{F}_l(\mathbf{P}(s),\mathbf{K}(s,\kappa^*))\|_\infty$ soit minimale.*

La norme minimale, notée γ_{opt}, est donnée par :

$$\gamma_{opt} = \|\mathcal{F}_l(\mathbf{P}(s),\mathbf{K}(s,\kappa^*))\|_\infty = \min_{\kappa\in\mathcal{K}_{stab}} \|\mathcal{F}_l(\mathbf{P}(s),\mathbf{K}(s,\kappa))\|_\infty$$

où $\mathcal{K}_{stab} \subset \mathcal{K}$ est l'ensemble des vecteurs $\kappa \in \mathcal{K}$ stabilisant internement le système. Comme précédemment, on ne cherche pas à atteindre γ_{opt}, mais à réduire autant que possible la norme $\mathcal{H}_\infty$ de la fonction de transfert $\mathbf{T}_{\mathbf{zw}}(s) = \mathcal{F}_l(\mathbf{P}(s),\mathbf{K}(s,\kappa))$.

B.2.3.2 Mise en forme du problème $\mathcal{H}_\infty$ structuré

On rappelle que les spécifications classiques peuvent être transposées en critères $\mathcal{H}_\infty$ normalisés de la forme :

$$\|W_i(s)T_i(s)\|_\infty < 1,\ i = 1,\ldots,m \tag{B.14}$$

où W_i est la fonction de pondération visant à spécifier la performance d'intérêt liant une entrée et une sortie spécifique et T_i représente la fonction de transfert considérée. De façon générale, on regroupe tous les critères dans une seule contrainte généralisée :

$$\|\mathbf{H}(s)\|_\infty < 1, \tag{B.15}$$

1. Une réalisation $(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D})$ d'une matrice de transferts rationnels réels propres $\mathbf{G}(s)$ est minimale si $\mathbf{A}$ a la plus petite dimension possible [170].

où

$$\begin{aligned}\mathbf{H}(s) &= \text{diag}\Big(W_1(s)T_1(s), \ldots, W_M(s)T_M(s)\Big) \\ &= \text{diag}\Big(H_1(s), \ldots, H_M(s)\Big).\end{aligned} \tag{B.16}$$

Chaque fonction de transfert $H_i(s) = W_i(s)T_i(s)$ peut être exprimée sous la forme LFT :

$$H_i(s) = \mathcal{F}_l\Big(P_i(s), \mathbf{K}(s,\kappa)\Big) \tag{B.17}$$

Du fait que l'interconnexion des LFT est aussi une LFT [201], on retrouve la forme standard de $\mathbf{H}(s)$ comme suit :

$$\mathbf{H}(s) = \mathcal{F}_l\Big(\mathbf{P}(s), \text{diag}\Big(\mathbf{K}(s,\kappa), \ldots, \mathbf{K}(s,\kappa)\Big)\Big) \tag{B.18}$$

où $\mathbf{P}(s)$ se déduit de $\text{diag}\,(P_1, \ldots, P_M)$ en permutant les entrées et les sorties (cf. [234]). Il reste enfin à réorganiser de manière similaire les paramètres κ dans $\text{diag}\,(\mathbf{K}(s,\kappa), \ldots, \mathbf{K}(s,\kappa))$ via une permutation des lignes et des colonnes. On obtient donc une structure de paramètres ajustables diagonale par bloc, c'est-à-dire :

$$\mathbf{H}(s) = \mathcal{F}_l\Big(\hat{\mathbf{P}}(s), \text{diag}\Big(\kappa_1\mathbf{I}_{n_1}, \ldots, \kappa_{n_k}\mathbf{I}_{n_{n_k}}\Big)\Big). \tag{B.19}$$

Notons que la répétition de $\mathbf{K}(s,\kappa)$ dans la forme standard de $\mathbf{H}(s)$ ne cause pas de problème, sauf une surcharge de calcul négligeable lors de la synthèse [241, 242].

Ainsi, la synthèse $\mathcal{H}_\infty$ structurée présente l'avantage majeur de pouvoir traiter les contraintes de façon indépendante. En fait, chaque critère $\mathcal{H}_\infty$ correspondant à un jeu d'entrée/sortie spécifique est formulé via un système augmenté P_i. De la sorte, lorsqu'on minimise la norme $\mathcal{H}_\infty$ de la matrice de transferts diagonale par bloc $\mathbf{H}(s)$, on minimise seulement la norme de canaux de performance imposés dans le cahier des charges et non pas celle de la matrice globale $\mathbf{T_{zw}}$. Bien évidemment, la forme standard (B.18) n'est pas applicable pour la synthèse $\mathcal{H}_\infty$ classique à cause de sa configuration structurée.

Un autre atout important de la méthode $\mathcal{H}_\infty$ structurée réside dans sa capacité de synthèse multimodèle [13]. [2] Pour cela, il suffit de répéter le contrôleur $\mathbf{K}(s,\kappa)$ plusieurs fois dans la forme standard (B.18). Considérons par exemple le cas où chaque $H_i(s) = W_i(s)T_i(s)$ est reliée à de multiples systèmes $P_i^{(1)}, \ldots, P_i^{(N)}$. On retrouve la forme standard où le système augmenté $\mathbf{P}(s)$ se déduit de $\text{diag}\Big(P_1^{(1)}, \ldots, P_1^{(N)}, \ldots, P_M^{(1)}, \ldots, P_M^{(N)}\Big)$. Le reste demeure similaire au développement dans le cas de multi-contraintes abordé ci-dessus.

2. La synthèse est dite multimodèle si elle permet de synthétiser un unique contrôleur en considérant plusieurs modèles simultanément.

B.2.3.3 Solution du problème $\mathcal{H}_\infty$ structuré

Vu la complexité dans la résolution analytique du problème $\mathcal{H}_\infty$ structurée [243], certains auteurs ont suggéré de le reformuler sous la forme d'un problème d'optimisation, soit par exemple la formulation proposée dans [242] :

$$\min_{\mathbf{K}(s,\kappa)\ \text{stabilisant}} \|\mathcal{F}_l(\mathbf{P}(s),\mathbf{K}(s,\kappa))\|_\infty \Leftrightarrow \min_{\kappa\in\mathcal{K}_{stab}} \max_{\omega\in[0,\infty)} \bar{\sigma}\Big[\mathcal{F}_l\left(\hat{\mathbf{P}}(j\omega), \mathrm{diag}\left(\kappa_1\mathbf{I}_{n_1},\ldots,\kappa_{n_k}\mathbf{I}_{n_{n_k}}\right)\right)\Big] \tag{B.20}$$

Ce problème d'optimisation de type min-max est non convexe et non lisse. Parmi les résolutions numériques les plus notables, mentionnons les algorithmes d'optimisation non lisse qui sont implantés sous MATLAB dans les fonctions `hinfstruct` et `looptune` [241,242]. Ces fonctions permettent d'ajuster de manière itérative les paramètres du contrôleur à architecture fixe pour stabiliser internement le système bouclé et minimiser des critères $\mathcal{H}_\infty$.

Plus récemment, l'algorithme avancé `systune` intégré dans la *Robust Control Toolbox* de MATLAB [13] permet de prendre en compte d'autres types de critères tels que le confinement de pôles, la norme $\mathcal{H}_2$, etc. Le problème d'optimisation prend la forme :

$$\begin{aligned} \min_{\kappa} \quad & \max_{i,k} \|T^{(k)}_{w_i\to z_i}\| \\ \text{t.q.} \quad & \max_{j,k} \|T^{(k)}_{w_j\to z_j}\| \le 1 \end{aligned} \tag{B.21}$$

où $\|\cdot\|$ dénote soit la norme $\mathcal{H}_\infty$ soit la norme $\mathcal{H}_2$, $\|T^{(k)}_{w_i\to z_i}\|$ représente la contrainte flexible correspondant au i^{e} jeu d'entrée/sortie du k^{e} système dans une synthèse multimodèle et $\|T^{(k)}_{w_j\to z_j}\|$ représente la contrainte dure à respecter. [3]

B.2.3.4 Avantages et limitations de la synthèse $\mathcal{H}_\infty$ structurée

La synthèse $\mathcal{H}_\infty$ structurée présente de nombreux avantages par rapport à l'approche classique, parmi lesquels :

- l'architecture de contrôle fixe,
- l'unique considération des fonctions de transfert d'intérêt (jeux d'entrées/sorties),
- la capacité à gérer la synthèse multimodèle,
- la capacité à inclure les critères de type confinement de pôles, norme $\mathcal{H}_2$, etc.

Les méthodes de synthèse $\mathcal{H}_\infty$ structurée, malgré leurs qualités, n'offrent aucune garantie quant à la convergence d'une solution numérique. Cette non garantie découle du choix *a priori* de l'architecture du contrôleur.

3. Pour une simplicité de notation, d'autres types de contraintes ne sont pas formalisés dans (B.21).

B.3 Autoséquencement de gains par LAE

B.3.1 Contrôleur latéral

Considérons le modèle d'état réduit de la dynamique latérale d'un hexacoptère :

$$\begin{bmatrix} \Delta\dot{y} \\ \Delta\ddot{y} \\ \Delta\dot{\phi} \\ \Delta\dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta\dot{y} \\ \Delta\phi \\ \Delta p \end{bmatrix} + \begin{bmatrix} \\ \mathbf{0} \\ \\ \frac{d}{2I_{xx}}\mathbf{\Lambda}_y \end{bmatrix} \begin{bmatrix} \Delta T_1 \\ \vdots \\ \Delta T_6 \end{bmatrix} \tag{B.22}$$

où $\mathbf{\Lambda}_y = [1-\gamma_1 \;\; 2(1-\gamma_2) \;\; 1-\gamma_3 \;\; 1-\gamma_4 \;\; 2(1-\gamma_5) \;\; 1-\gamma_6]$. Comme pour le cas du contrôleur longitudinal, on choisit le séquencement de gains affine en fonction de LAE :

$$\mathbf{K} = \mathbf{K}_0 + \sum_{i=1}^{6} \mathbf{K}_i\gamma_i \;, \quad \mathbf{K}_i = \mathbf{K}_{i_0} + \sum_{i=1}^{6} \mathbf{K}_{i_i}\gamma_i. \tag{B.23}$$

Dans ce cas, $\mathbf{K}$ est une matrice de dimension 6×4 et $\mathbf{K}_i$ est un vecteur colonne de dimension 6. En considérant la configuration particulière de l'hexacoptère, on peut simplifier ces gains de la façon suivante :

$$\mathbf{K}_{0,l} = \begin{cases} [a_{0y} \;\; b_{0y} \;\; c_{0y} \;\; d_{0y}] & \text{si } j = 0, l \in \{1,3,4,6\} \\ [2a_{0y} \;\; 2b_{0y} \;\; 2c_{0y} \;\; 2d_{0y}] & \text{si } j = 0, l \in \{2,5\} \\ [a_{1y} \;\; b_{1y} \;\; c_{1y} \;\; d_{1y}] & \text{si } j = l \in \{1,3,4,6\} \\ [2a_{1y} \;\; 2b_{1y} \;\; 2c_{1y} \;\; 2d_{1y}] & \text{si } j = l \in \{2,5\} \\ [a_{2y} \;\; b_{2y} \;\; c_{2y} \;\; d_{2y}] & \text{si } j \neq l, \{j,l\} \in \{1,3,4,6\} \\ [2a_{2y} \;\; 2b_{2y} \;\; 2c_{2y} \;\; 2d_{2y}] & \text{si } j \neq l, \{j,l\} \in \{2,5\} \end{cases} \tag{B.24}$$

$$\mathbf{K}_{i_{j,l}} = \begin{cases} i_{0y} & \text{si } j = 0, l \in \{1,3,4,6\} \\ 2i_{0y} & \text{si } j = 0, l \in \{2,5\} \\ i_{1y} & \text{si } j = l \in \{1,3,4,6\} \\ 2i_{1y} & \text{si } j \neq l, \{j,l\} \in \{2,5\} \\ i_{2y} & \text{si } j \neq l, \{j,l\} \in \{1,3,4,6\} \\ 2i_{2y} & \text{si } j \neq l, \{j,l\} \in \{2,5\} \end{cases} \tag{B.25}$$

Quant à l'implémentation du contrôleur autoséquencé sur le système non linéaire, les gains de la partie nominale sont donnés par :

$$[K_y \ K_{\dot{y}} \ K_\phi \ K_p \ K_{i,y}] = 6d[a_{0y} \ b_{0y} \ c_{0y} \ d_{0y} \ i_{0y}]. \tag{B.26}$$

B.3.2 Contrôleurs de l'altitude et du lacet

Les modèles d'état réduits utilisés à des fins de synthèse sont les suivants :

$$\begin{bmatrix} \Delta\dot{z} \\ \Delta\ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\dot{z} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{1\times 6} \\ \sum_{i=1}^{6} \frac{1-\gamma_i}{m} T_i \end{bmatrix} \tag{B.27}$$

$$\begin{bmatrix} \Delta\dot{\psi} \\ \Delta\ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\psi \\ \Delta\dot{\psi} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{1\times 6} \\ \sum_{i=1}^{6} \frac{(-1)^i(1-\gamma_i)}{I_{zz}} T_i \end{bmatrix} \tag{B.28}$$

Les dimensions des gains paramétriques $\mathbf{K}$ et $\mathbf{K}_i$ sont respectivement 6×2 et 6×1. Similairement, l'implémentation sur le système non linéaire est effectuée avec :

$$[K_z \ K_{\dot{z}} \ K_{i,z}] = 6[a_{0z} \ b_{0z} \ i_{0z}] \tag{B.29}$$

$$[K_\psi \ K_{\dot{\psi}} \ K_{i,\psi}] = 6c[a_{0\psi} \ b_{0\psi} \ i_{0\psi}]. \tag{B.30}$$

B.3.3 Synthèse du contrôleur avec `systune`

Soit `CL0` l'ensemble de modèles d'état généralisés de la dynamique longitudinal (4.10) en boucle fermée avec le séquencement de gains (4.14), défini par la fonction MATLAB `connect`. Soit `ref_x`, `pos_x`, `u` et `AP` respectivement l'entrée de référence, la sortie (position en x), la commande (T_1, T_3, T_4 et T_6) et le point d'intérêt en entrée (`AnalysisPoint`).

La procédure employée pour effectuer la synthèse du contrôleur autoséquencé via `systune` de manière à rencontrer les contraintes assignées dans le tableau 4.1 est la suivante :

1. Performance en poursuite.

```
TrackReq = TuningGoal.Tracking('ref_x','pos_x',3.25,0.0001);
TrackReq.focus = [0 0.1];
OverReq = TuningGoal.Overshoot('ref_x','pos_x',2);
```

2. Pôles en boucle fermée.

```
PoleReq = TuningGoal.Poles();
PoleReq.MinDamping = 0.6;
```

```
PoleReq.MinDecay = 1;
PoleReq.MaxFrequency = 10;
```

3. Marges de stabilité.

```
MarginReq = TuningGoal.Margins(AP.Location,12,45);
MarginReq.Focus = [0 0.1];
```

4. Effort de commande.

```
gmax_a = frd([0.1 1 10],[0 1 10]);
SatReq = TuningGoal.Gain('ref_x','u',gmax_a);
```

5. Atténuation de perturbations.

```
gmax_d = frd([0.1 0.1 10],[0 0.1 1]);
DisReq = TuningGoal.Gain(AP.Location,'pos_x',gmax_d);
```

6. Rejet de bruits de mesures.

```
gmax_n = frd([1 1 0.001 0.0001],[0 10 100 1000]);
NoiReq = TuningGoal.Gain('noise_x','pos_x',gmax_n);
```

7. Mise en œuvre de la fonction `systune`.

```
AllReqs = [TrackReq,PoleReq,MarginReq,DisReq,NoiReq,SatReq];
[CL,fSoft,hSoft,Info] = systune(CL0,AllReqs,[OverReq]);
```

ANNEXE C OPÉRATEUR DE PROJECTION

Nous procéderons ici à quelques rappels concernant l'opérateur de projection reportés dans [230]. Le lecteur peut trouver la démonstration des lemmes suivants dans cet article.

C.1 Convexité

Nous rappelons d'abord les notions de fonctions et d'ensembles convexes qui sont importantes pour définir l'opérateur de projection.

Définition C.1.1 ([244]). *Un ensemble $\Omega \subset \mathbb{R}^n$ est convexe si*

$$\forall \mathbf{x}, \mathbf{y} \in \Omega \Rightarrow \lambda \mathbf{x} + (1-\lambda)\mathbf{y} \in \Omega, \quad \forall \lambda \in [0,1] \tag{C.1}$$

Si une ligne relie deux points d'un ensemble convexe, tous les points de cette ligne figurent aussi dans cet ensemble.

Définition C.1.2 ([244]). *Une fonction $f : \mathbb{R}^n \to \mathbb{R}$ est convexe si*

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \Rightarrow f\left(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}\right) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}), \quad \forall \lambda \in [0,1] \tag{C.2}$$

Si une ligne relie deux points d'une fonction convexe, tous les points de la fonction entre ces deux extrêmes se situent sous cette ligne.

Lemme C.1.1 ([244]). *Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction convexe. Pour toute constante $\delta > 0$, le sous-ensemble $\Omega_\delta \triangleq \{\boldsymbol{\theta} \in \mathbb{R}^n \,|\, f(\boldsymbol{\theta}) \leq \delta\}$ est convexe.*

Lemme C.1.2 ([230]). *Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction convexe de classe $\mathcal{C}^1$ (continûment différentiable). Considérons le sous-ensemble :*

$$\Omega_\delta = \{\boldsymbol{\theta} \in \mathbb{R}^n \,|\, f(\boldsymbol{\theta}) \leq \delta\} \subset \mathbb{R}^n \tag{C.3}$$

avec le constant $\delta > 0$. Soit $\boldsymbol{\theta}^*$ un point intérieur de Ω_δ, i.e., $f(\boldsymbol{\theta}^*) < \delta$, et $\boldsymbol{\theta}$ un point sur la frontière de Ω_δ, i.e., $f(\boldsymbol{\theta}) = \delta$. Alors, l'inégalité suivante est vérifiée :

$$(\boldsymbol{\theta}^* - \boldsymbol{\theta})^\top \nabla f(\boldsymbol{\theta}) \leq 0 \tag{C.4}$$

où $\nabla f(\boldsymbol{\theta}) = \left[\frac{\partial f(\boldsymbol{\theta})}{\partial \theta_1} \quad \cdots \quad \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_n}\right]^\top \in \mathbb{R}^n$ est le gradient de la fonction f évalué au point $\boldsymbol{\theta}$.

C.2 Opérateur de projection

Définition C.2.1 ([230]). *L'opérateur de projection de deux vecteurs $\boldsymbol{\theta}, \mathbf{y} \in \mathbb{R}^n$ est défini par :*

$$\text{Proj}\,(\boldsymbol{\theta}, \mathbf{y}) \triangleq \begin{cases} \mathbf{y} - \dfrac{\nabla f(\boldsymbol{\theta})(\nabla f(\boldsymbol{\theta}))^\top}{\|\nabla f(\boldsymbol{\theta})\|^2}\mathbf{y} f(\boldsymbol{\theta}) & \text{si } f(\boldsymbol{\theta}) > 0 \text{ et } \mathbf{y}^\top \nabla f(\boldsymbol{\theta}) > 0 \\ \mathbf{y} & \text{sinon} \end{cases} \tag{C.5}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ est une fonction convexe et $\nabla f(\boldsymbol{\theta}) = \left[\frac{\partial f(\boldsymbol{\theta})}{\partial \theta_1} \quad \cdots \quad \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_n}\right]^\top$.

Le concept de l'opérateur de projection est illustré à la figure C.1.

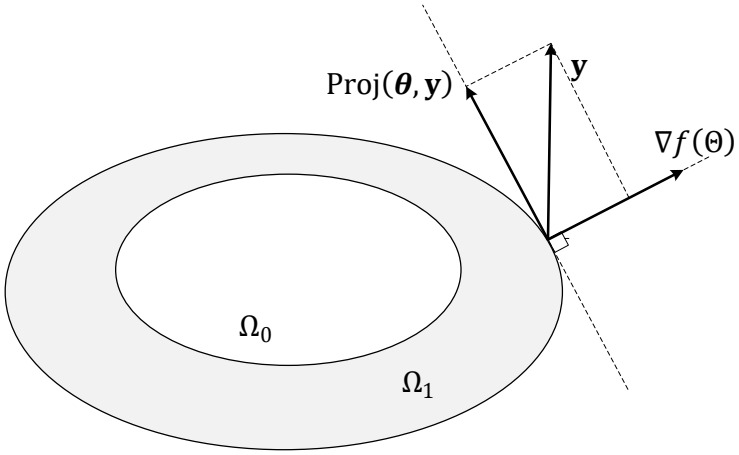


Figure C.1 Illustration de l'opérateur de projection

On rencontre souvent la loi d'adaptation avec modification robuste par l'opérateur de projection $\dot{\boldsymbol{\theta}} = \text{Proj}\,(\boldsymbol{\theta}, \mathbf{y})$ qui remplace la loi d'origine $\dot{\boldsymbol{\theta}} = \mathbf{y}$. Dans ce cas, l'interprétation géométrique de l'opérateur de projection est la suivante :

1. Si $\boldsymbol{\theta}$ se situe dans $\Omega_0 \triangleq \{\boldsymbol{\theta} \in \mathbb{R}^n \,|\, f(\boldsymbol{\theta}) \leq 0\}$, alors l'opérateur ne modifie pas $\mathbf{y}$.
2. Si $\boldsymbol{\theta}$ se situe entre Ω_0 et $\Omega_1 \triangleq \{\boldsymbol{\theta} \in \mathbb{R}^n \,|\, f(\boldsymbol{\theta}) \leq 1\}$ et $\mathbf{y}$ pointe vers l'intérieur de Ω_1, i.e., $\mathbf{y}^\top \nabla f(\boldsymbol{\theta}) < 0$, alors l'opérateur ne modifie pas $\mathbf{y}$.
3. Si $\boldsymbol{\theta}$ se situe entre Ω_0 et Ω_1 et $\mathbf{y}$ pointe vers l'extérieur de Ω_1, alors l'opérateur le modifie pour le garder à l'intérieur de Ω_1. En particulier, si $\boldsymbol{\theta}$ est sur la frontière de Ω_1, la $\text{Proj}\,(\boldsymbol{\theta}, \mathbf{y})$ est tangent à cette frontière. Par conséquent, $\boldsymbol{\theta}$ ne sortira jamais de cet ensemble Ω_1. Rappelons que $\frac{\nabla f(\boldsymbol{\theta})(\nabla f(\boldsymbol{\theta}))^\top}{\|\nabla f(\boldsymbol{\theta})\|^2}\mathbf{y}$ est la projection de $\mathbf{y}$ sur $\nabla f(\boldsymbol{\theta})$.

Ainsi, si on choisit une condition initiale $\boldsymbol{\theta}(t = 0) = \boldsymbol{\theta}_0 \in \Omega_1$, on peut donc assurer que $\boldsymbol{\theta}(t) \in \Omega_1,\ \forall t \geq 0$. Cela est reformulé dans le lemme suivant :

Lemme C.2.1 ([230]). *Soit $f : \mathbb{R}^n \to \mathbb{R}$, une fonction continûment différentiable et convexe. Considérons le système dynamique :*

$$\dot{\boldsymbol{\theta}} = \text{Proj}\left(\boldsymbol{\theta}, \mathbf{y}(t)\right) \tag{C.6}$$

où $\boldsymbol{\theta} \in \mathbb{R}^n$ est le vecteur d'état et $\mathbf{y}(t) \in \mathbb{R}^n$ est une fonction vectorielle continue par morceaux. Si $\boldsymbol{\theta}(t=0) = \boldsymbol{\theta}_0 \in \Omega_1 \triangleq \{\boldsymbol{\theta} \in \mathbb{R}^n \,|f(\boldsymbol{\theta}) \leq 1\}$, alors $\boldsymbol{\theta}(t) \in \Omega_1,\ \forall t \geq 0$. Autrement dit, Ω_1 est un ensemble invariant par rapport à (C.6).

Le lemme suivant nous donne une propriété importante de l'opérateur de projection.

Lemme C.2.2 ([230]). *Soit $\boldsymbol{\theta}^* \in \Omega_0$. L'inégalité suivante est valide :*

$$\left(\boldsymbol{\theta} - \boldsymbol{\theta}^*\right)^\top \left(\text{Proj}\left(\boldsymbol{\theta}, \mathbf{y}\right) - \mathbf{y}\right) \leq 0. \tag{C.7}$$

Dans le cadre de la commande adaptative, on utilise souvent l'extension matricielle de l'opérateur de projection.

Définition C.2.2 ([230]). *L'opérateur de projection de deux matrices de même dimension :*

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1 & \dots & \boldsymbol{\theta}_m \end{bmatrix} \in \mathbb{R}^{n\times m},\ \ \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \dots & \mathbf{y}_m \end{bmatrix} \in \mathbb{R}^{n\times m} \tag{C.8}$$

est défini par :

$$\text{Proj}\left(\boldsymbol{\Theta}, \mathbf{Y}\right) \triangleq \begin{bmatrix} \text{Proj}\left(\boldsymbol{\theta}_1, \mathbf{y}_1\right) & \dots & \text{Proj}\left(\boldsymbol{\theta}_m, \mathbf{y}_m\right) \end{bmatrix}. \tag{C.9}$$

Rappelons que :

$$\text{Proj}\left(\boldsymbol{\theta}_i, \mathbf{y}_i\right) = \begin{cases} \mathbf{y}_i - \dfrac{\nabla f_i(\boldsymbol{\theta}_i)(\nabla f_i(\boldsymbol{\theta}_i))^\top}{\|\nabla f_i(\boldsymbol{\theta}_i)\|^2}\mathbf{y}_i f_i(\boldsymbol{\theta}_i) & \text{si } f_i(\boldsymbol{\theta}_i) > 0 \text{ et } \mathbf{y}_i^\top \nabla f(\boldsymbol{\theta}_i) > 0 \\ \mathbf{y}_i & \text{sinon} \end{cases} \tag{C.10}$$

où chaque $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \dots, m$ est une fonction convexe.

On peut aussi étendre l'inégalité (C.7) de la façon suivante.

Lemme C.2.3 ([230]). *Soit $\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1 & \dots & \boldsymbol{\theta}_m \end{bmatrix} \in \mathbb{R}^{n\times m}$ telle que $\boldsymbol{\theta}_i \in \Omega_0 \triangleq \left\{\boldsymbol{\theta}_i \in \mathbb{R}^n \,|f_i(\boldsymbol{\theta}_i) \leq 0\right\}$, $i = 1, \dots, m$. L'inégalité suivante est valide :*

$$\text{trace}\left\{\left(\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\right)^\top \left(\text{Proj}\left(\hat{\boldsymbol{\Theta}}, \mathbf{Y}\right) - \mathbf{Y}\right)\right\} \leq 0 \tag{C.11}$$

avec $\hat{\boldsymbol{\Theta}} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_1, \dots, \hat{\boldsymbol{\theta}}_m \end{bmatrix} \in \mathbb{R}^{n\times m}$ et $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1, \dots, \mathbf{y}_m \end{bmatrix} \in \mathbb{R}^{n\times m}$.

ANNEXE D STABILITÉ ROBUSTE ET APPLICATION GARDIENNE

D.1 Produit et somme de Kronecker

Le produit de Kronecker de deux matrices quelconques $\mathbf{A} \in \mathbb{R}^{k\times l}$ et $\mathbf{B} \in \mathbb{R}^{m\times n}$, noté $\mathbf{A} \otimes \mathbf{B}$, est la matrice de dimension $km \times ln$, donnée par :

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1l}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{k1}\mathbf{B} & \cdots & a_{kl}\mathbf{B} \end{bmatrix}. \tag{D.1}$$

La somme de Kronecker de deux matrices carrées $\mathbf{A} \in \mathbb{R}^{m\times m}$ et $\mathbf{B} \in \mathbb{R}^{n\times n}$, noté $\mathbf{A} \oplus \mathbf{B}$, est la matrice de dimension $mn \times mn$, définie comme :

$$\mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_m \otimes \mathbf{B}. \tag{D.2}$$

Le produit de Kronecker est bilinéaire et associatif :

— Distribution : $\mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) = \mathbf{A} \otimes \mathbf{B} + \mathbf{A} \otimes \mathbf{C}$ et $(\mathbf{B} + \mathbf{C}) \otimes \mathbf{A} = \mathbf{B} \otimes \mathbf{A} + \mathbf{C} \otimes \mathbf{A}$;

— Association : $(k\mathbf{A}) \otimes \mathbf{B} = \mathbf{A} \otimes (k\mathbf{B}) = k\,(\mathbf{A} \otimes \mathbf{B})$.

Le produit de Kronecker n'est pas commutatif : en général $\mathbf{A} \otimes \mathbf{B} \neq \mathbf{B} \otimes \mathbf{A}$.
Le produit de Kronecker dispose d'autres propriétés parmi lesquelles :

— Produit mixte : $(\mathbf{AB}) \otimes (\mathbf{CD}) = (\mathbf{A} \otimes \mathbf{C})\,(\mathbf{B} \otimes \mathbf{D})$

— Inversion : $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$

D.2 Produit bialterné

Soient $\mathbf{A}$ et $\mathbf{B}$ deux matrices carrées de dimension n et V^n le $\frac{n(n-1)}{2}$-tuplet constitué des paires d'entiers (p, q), $p, q \in \{1, 2, \ldots, n\}, p < q$, listé lexicalement. Alors

$$V^n = \{(1,2), (1,3), \ldots, (1,n), (2,3), \ldots, (n-1,n)\}. \tag{D.3}$$

Soit V_i^n le i^{e} élément de V^n et

$$f\,[(p,q);(r,s)] = \frac{1}{2}\left(\det\begin{bmatrix} a_{pr} & a_{ps} \\ b_{qr} & b_{qs} \end{bmatrix} + \det\begin{bmatrix} b_{pr} & b_{ps} \\ a_{qr} & a_{qs} \end{bmatrix}\right). \tag{D.4}$$

Le produit bialterné [233] de $\mathbf{A}$ et $\mathbf{B}$, noté $\mathbf{A} \odot \mathbf{B}$, est la matrice carrée de dimension $\frac{n(n-1)}{2}$ dont les coefficients sont donnés par :

$$(\mathbf{A} \odot \mathbf{B})_{ij} = f\left(V_i^n, V_j^n\right). \tag{D.5}$$

Quelques propriétés élémentaires du produit bialterné sont :

(i) $\mathbf{A} \odot (\mathbf{B} + \mathbf{C}) = \mathbf{A} \odot \mathbf{B} + \mathbf{A} \odot \mathbf{C}$ et $(\mathbf{B} + \mathbf{C}) \odot \mathbf{A} = \mathbf{B} \odot \mathbf{A} + \mathbf{C} \odot \mathbf{A}$;

(ii) $(k\mathbf{A}) \odot \mathbf{B} = \mathbf{A} \odot (k\mathbf{B}) = k\,(\mathbf{A} \odot \mathbf{B})$;

(iii) $\mathbf{A} \odot \mathbf{B} = \mathbf{B} \odot \mathbf{A}$.

D.3 Transformation fractionnaire linéaire

Plusieurs problèmes d'analyse et de synthèse robuste (e.g., problème $\mathcal{H}_\infty$, μ-analyse, μ-synthèse, etc.) sont reformulés sous la forme d'une LFT où toutes les incertitudes du système sont rassemblées dans une matrice Δ.

Définition D.3.1 ([170])**.** *Soit* $\mathbf{M}$ *une matrice complexe partitionnée sous la forme :*

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \in \mathbb{C}^{(p_1+p_2)\times(q_1+q_2)} \tag{D.6}$$

et $\Delta \in \mathbb{C}^{q_2 \times p_2}$ *une matrice arbitraire. La LFT supérieure* $\mathcal{F}_u(\mathbf{M}, \Delta)$ *et la LFT inférieure* $\mathcal{F}_l(\mathbf{M}, \Delta)$ *sont définies par*

$$\mathcal{F}_u(\mathbf{M}, \Delta) := \mathbf{M}_{22} + \mathbf{M}_{21}\Delta(\mathbf{I} - \mathbf{M}_{11}\Delta)^{-1}\mathbf{M}_{12},$$
$$\mathcal{F}_l(\mathbf{M}, \Delta) := \mathbf{M}_{11} + \mathbf{M}_{12}\Delta(\mathbf{I} - \mathbf{M}_{22}\Delta)^{-1}\mathbf{M}_{21},$$

où les matrices $\mathbf{I} - \mathbf{M}_{11}\Delta$ *et* $\mathbf{I} - \mathbf{M}_{22}\Delta$ *sont supposées inversibles.*

Notons que la LFT inférieure peut être obtenue facilement à partir de la LFT supérieure qui est la seule disponible dans la LFR Toolbox [234] ou la LMI Toolbox de MATLAB. En effet, $\mathcal{F}_l(\mathbf{M}, \Delta) = \mathcal{F}_u(\mathbf{N}, \Delta)$ avec $\mathbf{M}_{11} = \mathbf{N}_{22}$, $\mathbf{M}_{12} = \mathbf{N}_{21}$, $\mathbf{M}_{21} = \mathbf{N}_{12}$ et $\mathbf{M}_{22} = \mathbf{N}_{11}$.

Une propriété importante de la représentation par LFT est que l'interconnexion de LFT est aussi une LFT [201]. Dans ce contexte, le lemme suivant nous donne une réalisation LFT explicite du produit de Kronecker de deux LFT.

Lemme D.3.1 ([245])**.** *Soient* $\mathbf{M} \in \mathbb{C}^{(p_1+p_2)\times(q_1+q_2)}$ *et* $\mathbf{N} \in \mathbb{C}^{(r_1+r_2)\times(s_1+s_2)}$ *deux matrices*

complexes partitionnées, $\Delta_M \in \mathbb{C}^{q_2 \times p_2}$ *et* $\Delta_N \in \mathbb{C}^{s_2 \times r_2}$*. Alors*

$$\mathcal{F}_l(\mathbf{M}, \Delta_M) \otimes \mathcal{F}_l(\mathbf{N}, \Delta_N) = \mathcal{F}_l(\mathbf{G}_s, \Delta_s) \tag{D.7}$$

avec

$$\mathbf{G}_s = \left[\begin{array}{c:cc} \mathbf{M}_{11} \otimes \mathbf{N}_{11} & \mathbf{M}_{12} \otimes \mathbf{I}_{r_1} & \mathbf{M}_{11} \otimes \mathbf{N}_{12} \\ \hdashline \mathbf{M}_{21} \otimes \mathbf{N}_{11} & \mathbf{M}_{22} \otimes \mathbf{I}_{r_1} & \mathbf{M}_{21} \otimes \mathbf{N}_{12} \\ \mathbf{I}_{q_1} \otimes \mathbf{N}_{21} & \mathbf{0} & \mathbf{I}_{q_1} \otimes \mathbf{N}_{22} \end{array}\right],$$
$$\Delta_s = \begin{bmatrix} \Delta_{\mathbf{M}} \otimes \mathbf{I}_{r_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{q_1} \otimes \Delta_{\mathbf{N}} \end{bmatrix}.$$

Similairement, la somme de deux LFT est aussi une autre LFT.

Lemme D.3.2 ([202])**.** *Soient* $\mathbf{M}$ *et* $\mathbf{N}$ *deux matrices complexes de dimension* $(p+q) \times (p+q)$ *partitionnées avec* $\mathbf{M}_{11},\ \mathbf{N}_{11} \in \mathbb{C}^{p \times p}$ *;* $\mathbf{M}_{22},\ \mathbf{N}_{22} \in \mathbb{C}^{q \times q}$ *et* $\Delta_{\mathbf{M}},\ \Delta_{\mathbf{N}} \in \mathbb{C}^{q \times q}$*. Alors*

$$\mathcal{F}_l(\mathbf{M}, \Delta_{\mathbf{M}}) \oplus \mathcal{F}_l(\mathbf{N}, \Delta_{\mathbf{N}}) = \mathcal{F}_l(\mathbf{G}, \Delta) \tag{D.8}$$

avec

$$\mathbf{G} = \left[\begin{array}{c:cc} \mathbf{M}_{11} \oplus \mathbf{N}_{11} & \mathbf{M}_{12} \otimes \mathbf{I}_p & \mathbf{I}_p \otimes \mathbf{N}_{12} \\ \hdashline \mathbf{M}_{21} \otimes \mathbf{I}_p & \mathbf{M}_{22} \otimes \mathbf{I}_p & \mathbf{0} \\ \mathbf{I}_{p_1} \otimes \mathbf{N}_{21} & \mathbf{0} & \mathbf{I}_p \otimes \mathbf{N}_{22} \end{array}\right],$$
$$\Delta = \begin{bmatrix} \Delta_{\mathbf{M}} \otimes \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_p \otimes \Delta_{\mathbf{N}} \end{bmatrix}.$$

Bien que l'ordre des LFT résultantes (D.7) et (D.8) ne soient généralement pas minimal, ces réalisations sont explicites et faciles à obtenir.

Le lemme suivant, issu de l'article [246], nous permet d'étudier la stabilité robuste de systèmes possédant une structure d'incertitude rationnelle.

Lemme D.3.3 ([246])**.** *Soit* $\mathbf{M} : \mathbb{R}^n \to \mathbb{R}^{p \times q}$ *une fonction matricielle rationnelle sans singularités à l'origine. Il existe des nombres entiers non négatifs* $q_1, \ldots, q_n$ *et des matrices* $\mathbf{A} \in \mathbb{R}^{p \times q}, \mathbf{B} \in \mathbb{R}^{p \times N}, \mathbf{C} \in \mathbb{R}^{N \times q}$ *et* $\mathbf{D} \in \mathbb{R}^{N \times N}$*, avec* $N = q_1 + \ldots + q_n$*, tels que* $\mathbf{M}$ *a la représentation LFT suivante pour tout* $\mathbf{x}$ *où* $\mathbf{M}$ *est défine :*

$$\mathbf{M}(\mathbf{x}) = \mathcal{F}_l\left(\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}, \mathbf{\Delta}(\mathbf{x})\right) = \mathbf{A} + \mathbf{B}\mathbf{\Delta}(\mathbf{x}) \left[\mathbf{I} - \mathbf{D}\mathbf{\Delta}(\mathbf{x})\right]^{-1} \mathbf{C} \tag{D.9}$$

avec $\mathbf{\Delta}(\mathbf{x}) = \text{diag}\,(x_1\mathbf{I}_{q_1}, \ldots, x_n\mathbf{I}_{q_n})$.

Afin de trouver les matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ et $\mathbf{D}$, de nombreux algorithmes symboliques et numériques ont été développés dans la littérature. On peut se reporter à la référence [234] pour une revue détaillée de ce sujet.

D.4 Applications gardiennes usuelles

Nous présentons ici quelques applications gardiennes correspondantes à des régions de stabilité classiques telles qu'illustrées à la figure D.1.

— Stabilité d'Hurwitz : pour le demi-plan gauche complexe ouvert $\Omega = \mathbb{C}^o_-$, l'ensemble $\mathcal{S}(\Omega)$ est gardé par une des applications gardiennes suivantes :

$$\nu_H(\mathbf{A}) = \det(\mathbf{A} \oplus \mathbf{A}) \tag{D.10}$$

$$\nu_H(\mathbf{A}) = \det(\mathbf{A}) \det\left(\mathbf{A} \odot \mathbf{I}\right) \tag{D.11}$$

où $\oplus$ dénote la somme de Kronecker et $\odot$ dénote le produit bialterné. Notons que les dimensions des matrices carrées $\mathbf{A} \oplus \mathbf{A}$ et $\mathbf{A} \odot \mathbf{I}$ sont respectivement n^2 et $n(n-1)/2$, tous les deux présentent une complexité spatiale de $\mathcal{O}(n^2)$.

— Marge de stabilité : soit Ω le demi-plan gauche ouvert décalé de α. Une application gardienne correspondante à Ω est donnée par :

$$\nu_\alpha(\mathbf{A}) = \det(\mathbf{A} - \alpha\mathbf{I}) \det\left(\mathbf{A} \odot \mathbf{I} - \alpha\mathbf{I} \odot \mathbf{I}\right). \tag{D.12}$$

— Amortissement : soit Ω le cône d'angle 2θ dans le demi-plan gauche, l'ensemble $\mathcal{S}(\Omega)$ est gardé par une des applications gardiennes suivantes :

$$\nu_\theta(\mathbf{A}) = \det\left(e^{j\theta}\mathbf{A} \ominus e^{-j\theta}\mathbf{A}\right) \quad \text{or} \tag{D.13}$$

$$\nu_\xi(\mathbf{A}) = \det(\mathbf{A}) \det\left(\mathbf{A}^2 \odot \mathbf{I} + \left(1 - 2\xi^2\right)\mathbf{A} \odot \mathbf{A}\right) \tag{D.14}$$

où $\mathbf{A} \ominus \mathbf{B} = \mathbf{A} \oplus (-\mathbf{B})$ et $\xi \triangleq \cos\theta$ désigne l'amortissement limite.

— Stabilité de Schur : si Ω est le cercle de rayon $\omega > 0$, l'ensemble $\mathcal{S}(\Omega)$ est gardé par une des applications suivantes :

$$\nu_\omega(\mathbf{A}) = \det\left(\mathbf{A} \otimes \mathbf{A} - \omega^2\mathbf{I} \otimes \mathbf{I}\right) \tag{D.15}$$

$$\nu_\omega(\mathbf{A}) = \det\left(\mathbf{A} \odot \mathbf{A} - \omega^2\mathbf{I} \odot \mathbf{I}\right) \det\left(\mathbf{A}^2 - \omega^2\mathbf{I}\right) \tag{D.16}$$

où $\otimes$ représente le produit de Kronecker.

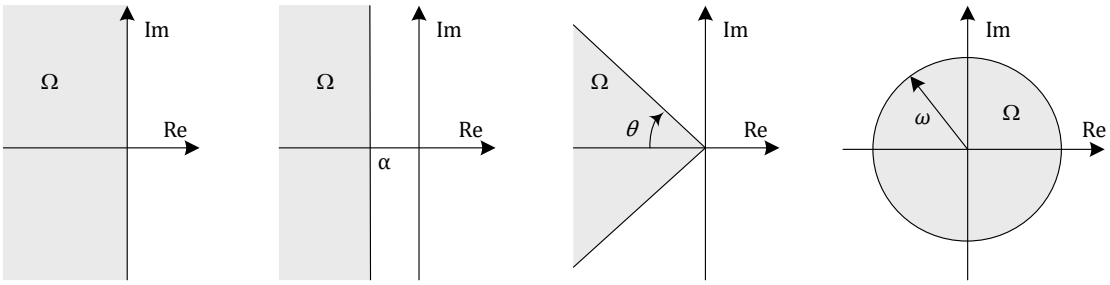


Figure D.1 Régions de stabilité d'intérêt

En pratique, il est parfois indispensable de construire une application gardienne correspondante à l'intersection de plusieurs domaines distincts. Le lemme suivant permet de combiner des applications gardiennes pour protéger un tel ensemble composé.

Lemme D.4.1 ([192])**.** *Soient les ensembles $\mathcal{S}(\Omega_1)$ et $\mathcal{S}(\Omega_2)$ qui sont respectivement gardés par les applications gardiennes ν_1 et ν_2. Alors $\mathcal{S}(\Omega_1 \cap \Omega_2)$ est gardé par $\nu = \nu_1\nu_2$.*

Le lecteur est renvoyé à [192] pour plus de détails concernant la construction d'applications gardiennes correspondantes aux différentes régions Ω.

D.5 Preuve du corollaire 6.2.2

La démonstration du corollaire 6.2.2 est similaire à celle de [192, Théorème 1]. Dans ce cas, on considère le système en boucle fermée :

$$\mathbf{A}_{BF}(\gamma) = \mathbf{A} - \mathbf{B}\mathbf{K}_0 + \gamma\mathbf{B}(\mathbf{K}_0 - \mathbf{K}_1) + \gamma^2\mathbf{B}\mathbf{K}_1. \tag{D.17}$$

Soit $\nu_H(\mathbf{A}_{BF}) = \det(\mathbf{A}_{BF} \oplus \mathbf{A}_{BF}) = \det \mathcal{F}(\mathbf{A}_{BF})$ l'application gardienne pour la stabilité d'Hurwitz (cf. Eq. (D.10)). Puisque le système bouclé est nominalement stable, soit $\nu_H(\mathbf{A}_{BF}(0)) = \det \mathcal{F}(\mathbf{A} - \mathbf{B}\mathbf{K}_0) = \det(\mathbf{F}_0) \neq 0$, alors $\mathbf{F}_0$ est inversible. $\mathcal{F}(\cdot)$ étant linéaire, on en déduit :

$$\begin{aligned}
\nu_H &= \det \mathcal{F}(\mathbf{A}_{BF}) = \det(\mathbf{F}_0 + \gamma\mathbf{F}_1 + \gamma^2\mathbf{F}_2) \\
&= \det(\mathbf{F}_0) \det\left(\mathbf{I} + \gamma\mathbf{F}_0^{-1}\mathbf{F}_1 + \gamma^2\mathbf{F}_0^{-1}\mathbf{F}_2\right) \\
&\propto \det(\mathbf{F}_0) \det \underbrace{\left(\lambda^2\mathbf{I} + \lambda\mathbf{F}_0^{-1}\mathbf{F}_1 + \mathbf{F}_0^{-1}\mathbf{F}_2\right)}_{\chi(\lambda)} = \det(\mathbf{F}_0) \det \chi(\lambda)
\end{aligned}$$

où $\lambda := 1/\gamma, \lambda \in [1, \infty)$. Comme $\det \chi(\lambda)$ est le polynôme caractéristique de la matrice $\mathbf{M}$ (cf. Eq. (6.62)), ν_H ne s'annule pas dans $[0, \bar{\gamma}]$ si $\mathbf{M}$ n'a pas de valeurs propres dans $[1/\bar{\gamma}, \infty)$. La preuve est donc complète.