POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Commande adaptative robuste par modèle de référence d'un quadricoptère

JÉRÉMIE PILON
Département de génie électrique

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

Août 2019

© Jérémie Pilon, 2019.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Commande adaptative robuste par modèle de référence d'un quadricoptère

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

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

DÉDICACE

À mes parents...

REMERCIEMENTS

Le travail effectué dans le cadre de cette maîtrise a été soutenu financièrement par le Conseil de Recherches en Sciences Naturelles et en Génie du Canada (CRSNG) et des Fonds de recherche du Québec en Nature et Technologie (FRQNT). Je remercie ces organismes de m'avoir octroyé ces bourses d'études qui m'ont permis de mener ce projet à terme.

Je tiens à remercier particulièrement mon directeur de recherche, David Saussié, pour son aide tout au long de la maîtrise. J'ai trouvé en lui un mentor, un chercheur de haut calibre et un ami. Son soutien inconditionnel pour toutes les activités que j'ai effectuées durant la maîtrise, que ce soit pour l'enseignement ou pour mes activités parascolaires, m'a permis de me développer au maximum de ma capacité, bien plus que seul un projet de recherche aurait pu le faire. Sa rigueur, sa passion pour la recherche et l'enseignement et sa générosité sont des qualités que j'espère pouvoir refléter dans tout travail que j'effectuerai à la suite de cette maîtrise.

Je remercie ensuite les membres du jury, M. Lahcen Saydy et M. Richard Gourdeau d'avoir lu mon mémoire de maîtrise et d'avoir accepté de participer à la soutenance. J'aimerais également remercier M. Lahcen Saydy pour ses précieux conseils, pour ses discussions enrichissantes et pour ses cours de maîtrise qui ont permis de faire grandement progresser ce travail de recherche. Je voudrais aussi remercier M. Richard Gourdeau pour la confiance qu'il m'a accordée lorsqu'il m'a choisi comme chargé de laboratoire, ainsi que pour toutes les connaissances qu'il m'a inculquées durant ce trimestre.

J'aimerais remercier tous mes collègues de laboratoire : André, Catherine, Olivier, Florian, Mathieu A., Mathieu G.-D., Mathieu S. et Justin. Les moments où nous avons travaillé ensemble au laboratoire sont parmi les plus beaux de mes études. Je vous remercie d'avoir participé de près ou de loin à mon projet de maîtrise et de m'avoir donné du soutien dans les hauts et les bas qui constituent un tel projet. J'aimerais également remercier particulièrement Duc-Tien Nguyen. Son aide a été un élément clef dans le succès de cette maîtrise. Sans son appui et ses connaissances techniques, il m'aurait fallu beaucoup plus de temps pour faire l'implémentation de mes lois de commande sur l'AscTec Pelican. Je tiens à remercier tous mes autres amis qui, sans être au laboratoire, m'ont appuyé dans de ce projet avec leurs encouragements : Nicolas, Vincent L., Mathieu D. et Philippe. Je veux aussi remercier M. Richard Labib et Mme Caroline Mailhot qui m'ont permis de progresser en devenant plus empathique et plus à l'écoute. Votre confiance donnée durant toutes ces années m'a été

essentielle pour mener à terme ce projet.

Pour son aide et son appui sans bornes, pour ses encouragements sans fin, pour son écoute sans limites, je tiens à remercier Maria. Si les deux dernières années ont été magnifiques, tu as permis d'unir tout cela dans le bonheur. Merci !

Enfin, mes derniers remerciements vont à ma famille. Sans votre appui et vos encouragements, je n'aurais pas pu accomplir tout ce que j'ai fait aujourd'hui. Je remercie mes trois sœurs : Laurence, Marjolaine et Marie-France d'avoir toujours cru en moi. Enfin, j'ai une pensée particulière pour mes parents. Sans votre travail acharné, sans votre soutien, sans votre contribution, sans vos mots d'encouragement et sans tout l'amour que vous m'avez donné toutes ces années, ce mémoire n'aurait jamais été fait. Je vous le dédie !

RÉSUMÉ

Depuis une dizaine d'années maintenant, les drones de type multirotor prennent une place grandissante dans le domaine civil. Ce sont cependant des systèmes naturellement instables qui nécessitent des algorithmes de commande pour fonctionner adéquatement. Bien que la modélisation et la commande des multirotors aient été largement étudiées, il reste encore à améliorer leur fiabilité pour une utilisation plus étendue. Une commande fiable nécessite une bonne connaissance des caractéristiques du drone et de sa dynamique. Toutefois, les sources d'incertitudes sont nombreuses : effets aérodynamiques, dégradation des moteurs, perturbations externes, etc. Ces incertitudes peuvent remettre en cause l'intégrité du drone et conduire à un écrasement. Dans ce contexte, la commande adaptative est une candidate idéale pour améliorer les aspects de fiabilité et de robustesse. Un contrôleur adaptatif permet d'estimer en temps réel certains paramètres incertains du système, et d'ajuster en conséquence les gains de la loi de commande afin de préserver la stabilité et la performance du système durant le vol.

Parmi tous les types de commandes adaptatives, celles avec un modèle de référence sont parmi les plus utilisées en pratique. En effet, leur structure variable permet de les concevoir pour rejeter différents types d'incertitudes et de perturbations. À cette fin, il est important de bien modéliser et de comprendre d'où ces incertitudes proviennent. L'objectif de ce projet de recherche est de développer une commande adaptative par modèle de référence permettant de rejeter des incertitudes paramétriques et non paramétriques assorties et mal-assorties.

Dans la première partie de ce travail, on développe la modélisation complète de la dynamique du quadricoptère. Puis, on la linéarise autour d'un point d'équilibre, celui du drone en vol stationnaire. À cette modélisation, les incertitudes et les perturbations sont ajoutées. Cela permet d'obtenir des simulations réalistes de l'appareil tout en facilitant la synthèse du contrôleur adaptatif qui devra les compenser.

Dans la seconde partie de ce travail, la commande adaptative est synthétisée et déployée sur l'AscTec Pelican. Les tests effectués en laboratoire permettent de démontrer qu'elle est bien en mesure de rejeter les perturbations paramétriques et non paramétriques assorties. Toutefois, elle ne parvient pas à rejeter les effets des perturbations non paramétriques mal-assorties. Pour y parvenir, il faut ajouter modifier la loi d'adaptation lors de la synthèse afin d'augmenter sa robustesse.

Finalement, les perturbations non paramétriques mal-assorties sont considérées grâce à l'uti-

lisation de modifications robustes qui viennent modifier la loi d'adaptation. Parmi celles-ci, l'opérateur de projection donne les meilleurs résultats, car sa synthèse repose sur des analyses de la stabilité effectuées avec des applications gardiennes et la μ–analyse. Les tests expérimentaux mettent en évidence les avantages de cette technique par rapport aux autres.

ABSTRACT

For the past decade, drones of the multirotor type have been taking an increasingly big place in the civil domain. However, they are naturally unstable systems and they require control algorithms to function properly. Even though the model of multirotors has been thoroughly studied, an increase in their reliability is still necessary to expand their uses in everyday applications. A reliable control algorithm entails an appropriate knowledge of the drone's characteristics and dynamic model. However, there are multiple different sources of uncertainties: aerodynamic effects, degradation of the motors, external disturbances, etc. These uncertainties can affect the integrity of the quadrotor and, eventually, lead to a crash of the device. In this context, an adaptive controller is an excellent candidate to improve reliability and robustness of the multirotor. An adaptive controller can estimate in real time certain unknown parameters of the system. Then, it can adjust gains in the control law in order to preserve the stability and performances of the system during flight.

Among all the different types of adaptive controllers, those using a reference model are the most implemented in practice. Indeed, their variable structure allows them to be designed to compensate precise sources of uncertainties. Hence, it is important to correctly model where the uncertainties come from. The objective of this research project is to develop a model reference adaptive controller that is able to compensate parametric uncertainties and matched and unmatched non-parametric uncertainties.

The first part of this work is to develop a complete dynamical model for the quadrotor. Then, the model is linearized about an equilibrium point: the drone in hover flight. The uncertainties and disturbances will then be added to the linearized model. With this model, we will be able to obtain more realistic simulations while facilitating the synthesis of the adaptive controller which will compensate these uncertainties.

In the second part of this work, the adaptive controller will be synthesized and deployed on the AscTec Pelican. The tests conducted in the laboratory show that the controller is capable of compensating parametric uncertainties as well as matched non-parametric uncertainties. However, the control law is not able to reject unmatched non-parametric uncertainties. Robust modifications must be added to the control algorithm in order to reach this objective.

Finally, the unmatched non-parametric uncertainties are compensated by developing robust modifications that change the adaptive law. Among these modifications, the projection operator gives the best results, because its synthesis is based on stability analyses done using

guardian maps and μ–analysis. Experimental results show the advantages of this technique with respect to other robust modifications.

TABLE DES MATIÈRES

DÉDICACE iii

REMERCIEMENTS iv

RÉSUMÉ vi

ABSTRACT viii

TABLE DES MATIÈRES x

LISTE DES TABLEAUX xiii

LISTE DES FIGURES xiv

LISTE DES SIGLES ET ABRÉVIATIONS xvi

LISTE DES ANNEXES xx

CHAPITRE 1 INTRODUCTION 1
- 1.1 Mise en situation 1
- 1.2 Objectifs de recherche 4
- 1.3 Plan du mémoire 5

CHAPITRE 2 REVUE DE LITTÉRATURE 6
- 2.1 Revue de littérature 6
  - 2.1.1 Modélisation d'un multirotor 6
  - 2.1.2 Synthèse des lois de commande 7
  - 2.1.3 Amélioration de la robustesse 10
  - 2.1.4 Commande robuste 13
- 2.2 Objectifs de recherche 15
  - 2.2.1 Sous-objectif #1 : modélisation de la dynamique d'un multirotor, des incertitudes et des perturbations 16
  - 2.2.2 Sous-objectif #2 : synthèse d'une MRAC augmentée 17
  - 2.2.3 Sous-objectif #3 : synthèse des modifications robustes traditionnelles pour la MRAC augmentée 17
  - 2.2.4 Sous-objectif #4 : synthèse de la MRAC avec opérateur de projection 18

CHAPITRE 3 MODÉLISATION DU QUADRICOPTÈRE ET MONTAGE EXPÉRIMENTAL . . . 20
3.1 Modèle dynamique du quadricoptère . . . 21
3.1.1 Repères de référence et définition des angles . . . 21
3.1.2 Équations cinématiques . . . 22
3.1.3 Équations dynamiques . . . 23
3.1.4 Modèle linéaire du drone . . . 30
3.2 Modélisation des incertitudes . . . 32
3.2.1 Incertitudes pour un système linéaire . . . 33
3.3 Montage expérimental . . . 34
3.3.1 Processus de synthèse des lois de commande . . . 37
3.3.2 Changement de repère . . . 37
3.4 Conclusion du chapitre . . . 39

CHAPITRE 4 SYNTHÈSE DES LOIS DE COMMANDE CLASSIQUES ET ADAPTATIVES . . . 40
4.1 Synthèse d'un contrôleur nominal par la commande LQ . . . 40
4.1.1 Considérations pratiques . . . 43
4.2 Synthèse de la MRAC . . . 43
4.2.1 Formulation de la commande adaptative . . . 43
4.2.2 Analyse de la stabilité . . . 47
4.3 Implémentation des contrôleurs et technique d'ajustement . . . 50
4.4 Simulation des contrôleurs . . . 53
4.4.1 Vol avec une masse additionnelle . . . 53
4.4.2 Vol en présence d'une perte d'efficacité des moteurs . . . 54
4.4.3 Rejet des perturbations assorties : étude de cas . . . 56
4.5 Résultats expérimentaux . . . 58
4.5.1 Critère de performance pour la MRAC . . . 58
4.5.2 Vol avec une masse additionnelle . . . 59
4.5.3 Vol en présence d'une perte d'efficacité des moteurs . . . 61
4.6 Limites de la MRAC et perturbations mal-assorties . . . 63
4.7 Conclusion du chapitre . . . 64

CHAPITRE 5 MODIFICATIONS ROBUSTES TRADITIONNELLES . . . 65
5.1 Problème de dérive des gains . . . 66
5.1.1 Préliminaires mathématiques . . . 66
5.1.2 Analyse de la stabilité . . . 68

5.1.3 Présence du phénomène sur le quadricoptère 70
5.2 Modification de la zone morte 71
5.3 Modification–σ 76
5.4 Modification–ϵ 82
5.5 Conclusion du chapitre 85

CHAPITRE 6 OPÉRATEUR DE PROJECTION 86
6.1 Introduction 86
6.2 Présentation de l'opérateur de projection 87
6.2.1 Concepts et définitions de base 87
6.2.2 Opérateur de projection 87
6.2.3 Exemple d'utilisation 89
6.2.4 Analyse de la stabilité de la MRAC construite avec l'opérateur de projection 91
6.2.5 Considérations pratique : implémentation de l'opérateur de projection 93
6.3 Obtention des bornes 94
6.3.1 Les applications gardiennes 95
6.3.2 Proposition de l'algorithme d'optimisation 97
6.3.3 Fonctionnement général de l'algorithme 100
6.3.4 Résultats de l'algorithme 104
6.4 Résultats expérimentaux 106
6.5 Conclusion du chapitre 110

CHAPITRE 7 CONCLUSION 111
7.1 Discussion et contributions 111
7.2 Recherches futures 112

RÉFÉRENCES 114

ANNEXES 119

LISTE DES TABLEAUX

Tableau 3.1 Paramètres dr l'Asctec Pelican 30
Tableau 4.1 Oscillations dans le plan x-y durant les tests LOE 56
Tableau 4.2 Résultats des ITSE pour le LQR et la MRAC 60
Tableau 4.3 Différences entre les ITSE avec le LQR et la MRAC 60
Tableau 4.4 Différences entre la perte d'altitude maximale 62
Tableau 4.5 Déplacements maximaux dans le plan x-y avec les pannes de moteur . 62
Tableau 5.1 Bornes maximales sur le bruit des états affectant les contrôleurs . . . 73
Tableau 5.2 Bornes maximales pour le balancement du quadricoptère 75
Tableau A.1 Valeurs expérimentales pour la poussée T et le couple Q du moteur de l'AscTec Pelican 121

LISTE DES FIGURES

Figure 1.1 Deux types de multirotors 1
Figure 3.1 Configurations de l'AscTec Pelican 20
Figure 3.2 Architecture du montage expérimental pour l'AscTec Pelican 36
Figure 3.3 Modèle du drone et représentation des repères 38
Figure 4.1 Architecture du contrôleur de base 41
Figure 4.2 Schéma bloc complet du contrôleur nominal en position 44
Figure 4.3 Schéma bloc du contrôleur nominal en position (boucle externe) . . . 44
Figure 4.4 Schéma bloc du contrôleur nominal en roulis et en tangage (boucle interne) 44
Figure 4.5 Schéma bloc du contrôleur nominal en altitude 45
Figure 4.6 Schéma bloc du contrôleur nominal en lacet 45
Figure 4.7 Schéma bloc général de la MRAC 50
Figure 4.8 Schéma bloc détaillé pour la loi adaptative 50
Figure 4.9 Méthodologie pour la synthèse et l'ajustement des lois de commande 51
Figure 4.10 Simulation d'un vol avec une masse additionnelle de 200 g au centre de masse 54
Figure 4.11 Simulations pour les LOE 55
Figure 4.12 Gains adaptatifs pour le rejet de la force de traînée 57
Figure 4.13 Résultats expérimentaux avec l'ajout d'une masse additionnelle . . . 59
Figure 4.14 Résultats expérimentaux pour les LOE 61
Figure 5.1 Illustration du théorème 5.1.2 67
Figure 5.2 Phénomène de dérive des gains durant un vol 70
Figure 5.3 Fonctions pour la modification de la zone morte 71
Figure 5.4 Simulations pour la modification robuste de la zone morte 73
Figure 5.5 Tests de la zone morte avec des bornes légèrement supérieures au bruit 74
Figure 5.6 Tests de la zone morte avec des bornes au-delà du balancement naturel 76
Figure 5.7 Simulation pour σ croissant pour le gain associé à l'état w 77
Figure 5.8 Stabilisation des gains 78
Figure 5.9 Phénomène d'oubli provoqué par la modification–σ 79
Figure 5.10 Limitation du processus d'adaptation 80
Figure 5.11 Dégradation des performances 81
Figure 5.12 Simulations de la modification–ϵ pour le gain adaptatif associé à l'état w 82
Figure 5.13 Simulations pour la modification robuste de la zone morte 83

Figure 5.14 Résultats expérimentaux avec la modification–ϵ pour le gain adaptatif associé à l'état r 84
Figure 6.1 L'opérateur de projection 88
Figure 6.2 Simulation pour l'exemple illustratif de l'opérateur de projection . . . 91
Figure 6.3 Régions de stabilité d'intérêt 96
Figure 6.4 Application gardienne d'intérêt 97
Figure 6.5 Objectif de l'algorithme d'optimisation pour l'obtention des bornes . 99
Figure 6.6 Compensation des pôles à la suite d'une perturbation $\mathcal{P}$ 103
Figure 6.7 Compensation des pôles à la suite d'une perturbation $\mathcal{P}$ 105
Figure 6.8 Gains adaptatifs pour les résultats optimaux de l'algorithme d'optimisation 107
Figure 6.9 Gains adaptatifs pour les résultats restrictifs de l'algorithme d'optimisation 108
Figure 6.10 Performances de la MRAC avec opérateur de projection 109
Figure A.1 Projection des aires selon les axes du repère objet 120
Figure C.1 Résultats expérimentaux pour la modification–σ lorsque $\sigma = 0.001$ . . 127
Figure C.2 Résultats expérimentaux pour la modification–σ lorsque $\sigma = 0.01$ . . 128
Figure C.3 Résultats expérimentaux pour la modification–σ lorsque $\sigma = 0.1$ . . . 129
Figure C.4 Résultats expérimentaux pour la modification–ϵ lorsque $\sigma = 0.001$ . . 130
Figure C.5 Résultats expérimentaux pour la modification–ϵ lorsque $\sigma = 0.01$ . . . 131
Figure C.6 Résultats expérimentaux pour la modification–ϵ lorsque $\sigma = 0.1$ . . . 132
Figure D.1 Formes d'une LFT 136
Figure D.2 Schéma bloc pour l'analyse de la stabilité robuste 136
Figure D.3 Exemple d'échantillonnage pour $n = 2$ et $p = 10$ 141

LISTE DES SIGLES ET ABRÉVIATIONS

Acronymes

ASMC	*Adaptive Sliding Mode Control*, commande adaptative par mode de glissement
CMRAC	*Combined Model Reference Adaptive Control*, commande adaptative par modèle de référence combinée
ENU	*East-North-Up*, est-nord-haut
ECS	*Electronic Speed Controller*, contrôleur électronique de la vitesse
FCU	*Flight Control Unit*, unité de commande de vol
FLU	*Front-Left-Up*, devant-gauche-haut
HLP	*High Level Processor*, processeur de haut niveau
IMU	*Inertial Measurement Unit*, unité de mesure inertielle
ITSE	*Integral of the Time-weighted Squared Error*, intégrale de l'erreur quadratique pondérée par le temps
LFT	*Linear Fractional Transformation*, transformation linéaire fractionnaire
LHS	*Latin Hypercube Sampling*, échantillonnage par hypercube latin
LLP	*Low Level Processor*, processeur de bas niveau
LOE	*Loss Of Effectiveness*, perte d'efficacité
MADS	*Mesh Adaptive Direct Search*, recherche directe par grille adaptative
MRAC	*Model Reference Adaptive Control*, commande adaptative par modèle de référence
MSF	*Multi-Sensor Fusion Framework*, module de fusion multi-capteurs
PID	Proportionnel Intégral Dérivé
SMC	*Sliding Mode Control*, commande par surface de glissement
UUB	Ultimement Uniformément Bornée
VRPN	*Virtual Reality Peripheral Network*, réseau périphérique de réalité virtuelle
VSS	Valeur singulière structurée

Mécanique du vol

- Repères :

$\mathcal{F}_b = \{O_b,\ \mathbf{x}_b,\ \mathbf{y}_b,\ \mathbf{z}_b\}$	Repère objet
$\mathcal{F}_i = \{O_i,\ \mathbf{x}_i,\ \mathbf{y}_i,\ \mathbf{z}_i\}$	Repère inertiel

- Caractéristiques physiques du quadricoptère :

m	Masse du quadricoptère
$\mathbf{I}$	Matrice des moments d'inertie du quadricoptère
l	Longueur de bras du quadricoptère
k_T	Coefficient de poussée des moteurs
k_D	Coefficient de traînée des moteurs
$\boldsymbol{\Upsilon}$	Matrice d'allocation des moteurs

- Paramétrage du quadricoptère :

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
u	Vitesse du centre de masse selon $\mathbf{x}_b$
v	Vitesse du centre de masse selon $\mathbf{y}_b$
w	Vitesse du centre de masse selon $\mathbf{z}_b$

- Forces et moments appliqués au quadricoptère

$\mathbf{F}_t$	Force de poussée
$\mathbf{F}_g$	Force gravitationnelle
$\mathbf{F}_d$	Force induite par le corps de l'appareil
$\mathbf{M}_t$	Moment induit par la force de poussée appliquée sur un bras de levier
$\mathbf{M}_q$	Moment induit par la traînée des hélices

Systèmes de commande de vol

- Incertitudes :

$\mathbf{A}_u,\ \mathbf{\Lambda}$	Incertitudes paramétriques
$\mathbf{f}$	Incertitudes non paramétriques assorties
$\boldsymbol{\xi}$	Incertitudes non paramétriques mal-assorties

- Commande classique :

$\mathbf{Q}$	Matrice de pondération pour les états
$\mathbf{R}$	Matrice de pondération pour les entrées

- Commande adaptative :

$\hat{\boldsymbol{\theta}}$	Matrice des gains adaptatifs
$\boldsymbol{\omega}$	Vecteur des régressions
$\mathbf{\Gamma}_{\boldsymbol{\theta}}$	Matrice des coefficients d'adaptation des gains

Commande robuste

- μ–analyse

$\mathbf{\Delta}$	Bloc d'incertitudes des modèles LFT

- Applications gardiennes

$\mathcal{S}$	Ensemble de stabilité généralisée
$\partial\mathcal{S}$	Frontière de $\mathcal{S}$
$\overline{\mathcal{S}}$	Adhérence de $\mathcal{S}$

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
$\mathbb{S}^n$	Ensemble des matrices réelles symétriques d'ordre n
$\otimes$	Produit de Kronecker
$\oplus$	Somme de Kronecker
$\sigma(\mathbf{A})$	Valeurs singulières de la matrice $\mathbf{A}$
$\lambda(\mathbf{A})$	Valeurs propres de la matrice $\mathbf{A}$
x	Une variable scalaire
$\mathbf{x}$	Un vecteur

$\mathbf{X}$	Une matrice
$\mathbf{v}^1$	Projection du vecteur $\mathbf{v}$ dans le repère $\mathcal{F}_1$
${}^1\dot{\mathbf{v}}$	Dérivée du vecteur $\mathbf{v}$ par rapport au repère $\mathcal{F}_1$
${}^1\dot{\mathbf{v}}^2$	Projection dans le repère $\mathcal{F}_2$ de la dérivée du vecteur $\mathbf{v}$ par rapport au repère $\mathcal{F}_1$
$\mathbf{R}_{2/1}$	Rotation permettant le passage du repère $\mathcal{F}_1$ vers le repère $\mathcal{F}_2$
$\mathbf{R}_{\mathbf{x}}(\theta)$	Matrice de rotation autour de l'axe $\mathbf{x}$ de θ radians

LISTE DES ANNEXES

Annexe A Dynamique du quadricoptère 119

Annexe B Synthèse des lois de commande classiques et adaptatives 123

Annexe C Modifications robustes traditionnelles 125

Annexe D Opérateur de projection 133

CHAPITRE 1 INTRODUCTION

1.1 Mise en situation

Depuis une dizaine d'années, les drones de type multirotors prennent une place grandissante dans diverses applications tant du domaine civil que du domaine commercial. On retrouve parmi eux les quadricoptères équipés de quatre hélices (Fig. 1.1a) et les hexacoptères équipés de six hélices (Fig. 1.1b).

(a) Un quadricoptère

(b) Un hexacoptère

Figure 1.1 Deux types de multirotors

C'est la facilité de les construire et de les piloter qui a mené à leur démocratisation dans toutes sortes d'activités : la livraison de colis à domicile, l'inspection de bâtiments et la surveillance de terrains dangereux sont des exemples parmi tant d'autres. Si par le passé, les drones étaient largement pilotés manuellement, aujourd'hui, c'est l'automatisation de ces appareils qui devient un enjeu majeur. Cependant, la commande autonome se bute à un problème qui peut rendre son utilisation difficile : les incertitudes et les perturbations paramétriques et non paramétriques. Les incertitudes paramétriques sont reliées aux erreurs provenant des paramètres du drone. Citons par exemple, de mauvaises estimations de la masse, des moments d'inertie et des coefficients de traînée et de poussée des moteurs. Quant aux incertitudes non paramétriques, celles-ci sont modélisées par des fonctions mathématiques complexes et non plus seulement des constantes. Cela inclut, entre autres, les dynamiques négligées et le bruit des capteurs.

C'est au début du XXe siècle que la théorie de la commande prend son essor dans les sphères

scientifiques. Durant les années 1920 et 1930, on assiste au développement du correcteur proportionnel intégral dérivé (PID). Ce correcteur est intuitif, tant dans son utilisation que dans sa synthèse, et son succès est rapide. C'est pour cette raison qu'il est aujourd'hui le correcteur le plus répandu sur le marché. Durant ces années et jusqu'à la fin des années 1940, de grands scientifiques ont contribué à l'avancement de la théorie de la commande. Notons particulièrement les travaux de H. Nyquist sur la stabilité des systèmes dynamiques et H. W. Bode sur la réponse fréquentielle des systèmes dynamiques [1]. Ces méthodes d'analyse et de commande reposant principalement sur la rétroaction définissent la période classique de la commande.

Vers la fin des années 1950, les scientifiques ont commencé à s'intéresser à des algorithmes de commande plus sophistiqués pour considérer, par exemple, les erreurs de modélisation. La première loi de commande adaptative est proposée en 1958 par H. Whitaker [2]. Cet algorithme cherche à estimer les gains des régulateurs classiques en fonction de l'erreur entre le modèle réel et un modèle de référence. Son article donne naissance à la commande adaptative par modèle de référence, plus connu en anglais sous le nom de *Model Reference Adaptive Control* (MRAC). Dans ses travaux, Whitaker avait négligé de démontrer formellement la stabilité de son contrôleur et la convergence de son algorithme de commande. En 1966, P. Parks prouve que cette méthode peut, dans certains cas, mener à des instabilités [3]. Il propose alors un nouveau design du contrôleur MRAC en utilisant les notions de stabilité de Lyapunov. Toutefois, une implantation hâtive de la méthode de Whitaker avec des notions élémentaires et fautives des théories de la stabilité de Lyapunov mène, en 1967, à l'écrasement d'un avion de chasse américain X-15, et à la mort de son pilote [4]. À la suite de cet incident, la commande adaptative est ignorée par le secteur industriel qui la juge inexploitable.

Au moment même où Whitaker et Parks développent les rudiments de la commande adaptative, R. Kálmán et J. Bertram travaillent à propager les méthodes d'analyse de stabilité de Lyapunov, jusqu'alors peu connues en Occident. Les travaux de recherche qui émergent de ces discussions permettent alors de mieux comprendre la stabilité de Lyapunov qui est devenue, aujourd'hui, l'une des principales méthodes d'analyse de la stabilité des systèmes dynamiques. Les recherches effectuées durant cette décennie relancent l'intérêt pour le domaine de la commande adaptative, en se basant à présent sur des méthodes d'analyse et de synthèse rigoureuses.

Durant les années 1970, de nouvelles méthodes adaptatives sont développées telles que le placement de pôle adaptatif inventé par K. J. Åström et P. Egardt [5,6]. Les travaux effectués durant ces années se concentrent surtout sur la commande adaptative par modèle de référence. Malgré leurs nouveaux succès, Egardt démontre que ces contrôleurs peuvent être instables

en présence de faibles perturbations non paramétriques. C'est durant les années 1980 que les scientifiques s'intéressent davantage à ces perturbations, ce qui mènera à l'élaboration de la commande robuste. Les recherches dans ce domaine ont permis de développer des améliorations pour les contrôleurs adaptatifs par l'inclusion de modifications robustes telles que la modification-σ, la modification-ϵ et la modification de la zone morte. Ces modifications ont permis d'améliorer significativement la robustesse des contrôleurs adaptatifs en présence de perturbations externes non paramétriques.

Les trois décennies de 1960 à 1980 ont aussi été très fertiles pour le développement de la commande non linéaire. Après la publication par Kálmán de ses travaux sur l'observabilité et la commandabilité des systèmes linéaires [7], il faudra attendre 10 ans pour que ses idées soient adaptées aux systèmes non linéaires. C'est avec ces nouveaux outils que les scientifiques s'attaqueront à la synthèse de toutes sortes de méthodes de commande non linéaire. Parmi celles-ci, le contrôleur par surface de glissement deviendra une méthode très répandue permettant également la commande robuste de systèmes dynamiques non linéaires.

Malgré toutes ces percées théoriques, l'implantation de lois de commande adaptatives et non linéaires tant sur des avions que sur des multirotors représente un défi important, surtout avant les années 1980. En effet, ces méthodes de commande nécessitent parfois de puissants processeurs pour arriver à effectuer tous les calculs en temps réel. Il était ainsi presque impossible d'embarquer de volumineux appareils de calcul à bord des avions et des drones. Or, avec la miniaturisation des ordinateurs vers la fin du XXe siècle, il devient possible d'embarquer ces systèmes. Aujourd'hui, les ordinateurs ont été remplacés par des microcontrôleurs qui peuvent être embarqués sur des drones, et même sur des microdrones, voire des nanodrones.

Malgré la miniaturisation des ordinateurs de bord, il subsiste encore plusieurs défis techniques concernant l'implémentation des lois de commande sur des multirotors, notamment la diversification des tâches et des conditions de vol. Dans le premier cas, on souhaite que le drone puisse, par exemple, utiliser des caméras, des lidars, un GPS, voire un bras robotique. Dans le second cas, on veut que l'appareil puisse voler en présence de vent ou malgré des bris mécanique. En mode manuel, on se fie généralement à l'expertise du pilote pour manipuler adéquatement le drone pour différentes utilisations et dans différentes conditions de vol. Grâce à ses habiletés, il rend le multirotor polyvalent. Cependant, pour un drone autonome sans pilote, ce sont les lois de commande qui doivent assurer cette polyvalence.

Les méthodes issues de la commande classique (lieu des racines, diagramme de Bode) sont rarement suffisantes pour synthétiser des lois de commande assurant la stabilité et la performance des multirotors pour des conditions diverses d'opération. La commande moderne,

qu'elle soit adaptative ou robuste, demeure la candidate idéale pour atteindre les performances souhaitées. C'est pourquoi, depuis plus de 10 ans, les chercheurs développent de nouveaux algorithmes de commande adaptative robuste et les implantent dans des multirotors.

Le développement de meilleurs outils de commande aura des retombées significatives. Tout d'abord, l'augmentation de la fiabilité et de la robustesse permettront une utilisation plus étendue de ces engins autonomes. Ensuite, la diminution du nombre d'écrasements augmentera la sécurité du public et se traduira par des économies d'argent substantielles pour les compagnies œuvrant dans ce domaine. Enfin, cela contribuera à augmenter la confiance du public envers les appareils autonomes.

1.2 Objectifs de recherche

L'objectif général de ce projet est de compenser les incertitudes paramétriques et non paramétriques d'un quadricoptère par l'utilisation d'algorithmes de commande adaptative. Le projet est réalisé en trois étapes.

Durant la première étape, une modélisation de la dynamique du drone est effectuée. Cette modélisation permettra de mettre en évidence les incertitudes reliées aux paramètres du modèle tels que la masse, l'inertie et les coefficients de poussée et de traînée des moteurs. De plus, on considérera les incertitudes reliées aux dynamiques négligées de l'appareil et aux perturbations externes. En effet, pour s'assurer d'une bonne synthèse des contrôleurs adaptatifs, il faut que le modèle tienne compte de tous les types d'incertitudes et perturbations pouvant affecter le multirotor.

Durant la seconde étape, une synthèse des lois de commande adaptative par modèle de référence est effectuée. Nous nous assurerons de la validité de l'algorithme en garantissant la convergence des lois de commande. De plus, la validation de ces lois par simulation nous permettra de procéder à l'implantation sur un appareil physique. Nous procéderons à l'ajustement itératif des lois de commande. Enfin, nous comparerons les performances de la MRAC avec une méthode classique linéaire quadratique de commande de quadricoptère pour en dégager les avantages et les inconvénients.

La dernière étape de ce projet consistera à améliorer la robustesse de la loi de commande. En effet, nous ajouterons des modifications robustes pour que le drone puisse rejeter certaines perturbations non paramétriques. Entre autres, nous développerons et implémenterons une MRAC construite avec un opérateur de projection. Pour que cette méthode soit performante,

il faut lier deux domaines différents de la théorie de la commande : la commande adaptative et la commande robuste. Ce lien entre les deux domaines nous permettra d'obtenir un algorithme original, plus performant et efficace que la plupart des autres méthodes de commande.

1.3 Plan du mémoire

Dans le chapitre 2, nous faisons une revue de la littérature concernant les divers sujets du mémoire. Nous présenterons succinctement l'histoire de la théorie de la commande pour terminer sur les recherches les plus récentes concernant la commande adaptative robuste des multirotors.

C'est dans le chapitre 3 que nous aborderons la modélisation mathématique des lois dynamiques qui régissent le comportement du quadricoptère. Puis, sous certaines hypothèses simplificatrices, nous réduirons le modèle non linéaire du drone. Nous terminerons par l'obtention d'un modèle linéaire de l'appareil. De plus, une description sera donnée quant au montage expérimental en laboratoire. Le modèle linéarisé nous permettra, dans le chapitre 4, de développer et de synthétiser des lois de commande classiques et adaptatives. Nous procéderons ensuite à l'ajustement de ces lois en vue de leur implantation sur le quadricoptère. Puis, nous validerons leur fonctionnement avec des tests en laboratoire.

Dans le chapitre 5, nous décrirons les modifications robustes classiques apportées aux lois de commande afin de compenser des perturbations non paramétriques. Une analyse comparative entre ces différentes méthodes sera alors effectuée dans le but de déterminer les forces et les faiblesses de chacune. Au chapitre 6, nous présenterons l'opérateur de projection, une modification robuste supplémentaire. Un algorithme d'optimisation qui permet de la synthétiser sera alors présenté.

Ces résultats permettront de donner des recommandations sur l'architecture idéale des lois de commande pour le vol de quadricoptère sous l'influence des perturbations assorties et mal-assorties ainsi que d'incertitudes paramétriques et non paramétriques. Enfin, nous ferons un retour sur les objectifs de recherche dans le chapitre 7 et nous terminerons par une brève ouverture sur les recherches futures qui pourront faire suite à ce travail de maîtrise.

CHAPITRE 2 REVUE DE LITTÉRATURE

On commence ce travail de recherche par une revue de littérature dont l'objectif est de faire un état de l'art des différents sujets abordés dans ce mémoire.

L'objectif de ce projet est de concevoir une commande adaptative par modèle de référence pour un multirotor afin de compenser des incertitudes et des perturbations paramétriques et non paramétriques. À cette fin, nous commencerons par modéliser la dynamique du multirotor, ainsi que les incertitudes et leur influence sur le système.

À la suite de cette phase de modélisation, nous procéderons à la synthèse des lois de commande adaptative. Pour y parvenir, nous devons d'abord construire le modèle de référence nécessaire au fonctionnement de la MRAC ; nous étudierons à cet effet plusieurs méthodes de commande classique et moderne. Nous aborderons ensuite les différentes lois d'adaptation et nous tenterons d'en déterminer les avantages et les inconvénients.

Enfin, nous nous intéresserons à l'augmentation de la robustesse du contrôleur. Nous effectuerons d'abord un survol des méthodes d'analyse de la robustesse, et nous répertorierons ensuite les modifications robustes couramment utilisées en commande adaptative. Cette revue nous permettra ainsi de déterminer les outils mathématiques d'analyse et de conception nécessaires pour atteindre notre but.

2.1 Revue de littérature

2.1.1 Modélisation d'un multirotor

La modélisation d'un multirotor est un sujet largement traité depuis la dernière décennie. Un multirotor est considéré comme un corps rigide à 6 degrés de liberté, équipé de moteurs et d'hélices, et généralement sous-actionné ; les hélices génèrent les forces et les moments qui permettent de modifier son orientation et sa position. Les équations non linéaires du mouvement ont par exemple été abordées dans [8, 9].

Outre la modélisation traditionnelle, plusieurs effets aérodynamiques secondaires peuvent être considérés afin d'obtenir un modèle dynamique plus réaliste. Certains de ces effets peuvent être négligeables pour des appareils de grandes tailles, tandis que d'autres affectent tous les types de multirotors. Par exemple, le battement des pales des rotors, considéré par [10] et [11] est un effet aérodynamique qui affecte tous les appareils lorsqu'il y a un mouvement latéral.

Cependant, la traînée induite affecte surtout les appareils de petite taille qui ont des rotors pratiquement rigides [12, 13].

La plupart des modèles présentés dans les articles précédents ont en commun qu'ils ne considèrent pas les incertitudes dans la modélisation. Les incertitudes dites paramétriques sont reliées à des aspects physiques du multirotor, e.g., la masse, les moments d'inertie et la poussée des moteurs [14, 15]. Les incertitudes non paramétriques proviennent des dynamiques négligées : les forces aérodynamiques secondaires, la présence de vent et les délais de transmission dans le système [16, 17]. La connaissance et la modélisation des incertitudes sont intrinsèquement liées au développement de la commande adaptative et de la commande robuste.

Pour la synthèse des lois de commande classique et adaptative, nous allons travailler à partir d'un modèle linéaire. À cet effet, on linéarise le modèle non linéaire autour d'un point d'équilibre pour obtenir un système linéaire du multirotor. Plusieurs points d'équilibre peuvent être choisis, comme par exemple, le vol stationnaire ou le vol à vitesse et altitude constantes. Le point d'équilibre le plus utilisé en pratique demeure cependant le vol stationnaire. Le principal avantage de ce modèle linéarisé est qu'il présente un découplage naturel entre les différentes dynamiques du multirotor. Il est alors possible de synthétiser quatre lois de commande indépendantes les unes des autres : une pour l'altitude, une pour le mouvement latéral, une pour le mouvement longitudinal et une pour le lacet.

Pour le développement d'algorithmes de commande adaptative, une modélisation adéquate des incertitudes est nécessaire. Bien qu'il soit possible de considérer ces incertitudes dans le modèle non linéaire du multirotor, il est préférable de reformuler les équations dynamiques comme un système linéaire incertain invariant dans le temps, tel qu'exprimé par [18]. Lorsque le système est donné sous cette forme, on peut alors considérer l'ensemble des incertitudes reliées au multirotor tel que mentionné ci-dessus.

2.1.2 Synthèse des lois de commande

Synthèse des commandes classiques

La théorie de la commande adaptative a été initiée dans les années 1950 et a mené en 1958 à la première implémentation d'un contrôleur adaptatif sur un avion de chasse X-15 [2]. Cet avion est équipé de lois de commande adaptative qui ont pour but d'ajuster en temps réel les gains de commande. Or, la convergence de ces contrôleurs n'est pas bien définie. En effet, la loi de commande adaptative, basée sur la règle du MIT n'a pas une garantie générale de

convergence. Cela mène l'avion à entrer dans un cycle limite qui, éventuellement, conduit à l'écrasement de l'appareil [4]. Ainsi, la théorie de la commande adaptative est très dépendante d'une analyse de la stabilité et de la convergence.

À cette époque, les analyses de stabilité ne sont pas basées sur des concepts mathématiquement rigoureux. Elles le sont encore moins pour des systèmes dynamiques non linéaires. À la suite de la propagation des notions de la stabilité de Lyapunov en Occident, de nouvelles lois adaptatives sont développées. Le lecteur peut se référer à [19] pour les notions liées à la théorie de la stabilité des systèmes non linéaires. Les contrôleurs adaptatifs se divisent en deux groupes principaux : les lois adaptatives directes et indirectes. Avec les méthodes directes, on calcule l'effort de commande nécessaire pour amener l'erreur entre un système réel et un modèle de référence à zéro. Quant aux méthodes indirectes, elles ont pour but d'estimer les états et les paramètres incertains du système réel pour adapter les gains d'un contrôleur stabilisant [20]. Le lecteur peut se référer à la riche documentation concernant la théorie de la commande adaptative [5, 21–23].

La MRAC suit naturellement les travaux commencés par Whitaker en 1958, en intégrant des éléments clefs de la théorie de la stabilité de Lyapunov [24]. En effet, une preuve de convergence assure le comportement asymptotique du contrôleur en régime permanent. On peut construire la MRAC avec deux structures : pure ou augmentée. Dans le cas pur, la MRAC est la seule loi de commande présente sur le système tandis que dans le cas augmenté, elle assiste un contrôleur de base dans l'effort de commande. De plus, la MRAC peut être définie sous trois formes : directe, indirecte ou combinée. Le cas combiné, appelé en anglais *Combined Model Reference Adaptive Control* (CMRAC), développé simultanément par [25] et [26], synthétise une MRAC utilisant les informations provenant des lois directes et indirectes ce qui permet d'obtenir un meilleur correcteur. Cette affirmation, bien que validée expérimentalement, n'a pas encore été formellement prouvée, et est appelée la conjecture de la CMRAC [27]. L'implémentation de la MRAC pure est moins utilisée en pratique sur un multirotor que la méthode augmentée. Notons toutefois les contributions de [28] pour l'implémentation de la MRAC pure pour compenser les effets de grandes incertitudes paramétriques.

Avec la MRAC augmentée, l'automaticien choisit de combiner les bienfaits de la commande adaptative avec la simplicité de synthèse des contrôleurs classiques, e.g., le correcteur PID, ou la commande linéaire quadratique (LQ) et son régulateur linéaire quadratique, i.e., *Linear Quadratic Regulator* (LQR). En effet, on synthétise d'abord une loi de commande sur le système linéaire grâce à une de ces méthodes éprouvées, et on choisit comme modèle de référence idéal le système en boucle fermée ainsi obtenu. L'objectif de la loi adaptative revient alors à apporter les corrections nécessaires afin de suivre le modèle de référence dans le

cas d'incertitudes ou d'erreurs de modèle. En revanche, avec la MRAC pure, l'algorithme adaptatif est la seule loi de commande sur le multirotor. Elle est donc moins intuitive à utiliser que sa version augmentée, car le modèle de référence doit être construit *a priori*. Ainsi, augmenter le contrôleur classique avec une commande adaptative permet d'améliorer sa robustesse et ses performances.

Dans [29], les auteurs implémentent une MRAC augmentée sur un multirotor. Ils vérifient ensuite son comportement en vol à la suite d'une panne partielle d'un moteur. Ils comparent les réponses temporelles pour cet événement pour un correcteur adaptatif qui utilise la règle du MIT, pour une MRAC augmentée et pour un contrôleur de base LQR. Ils démontrent alors que la commande MRAC augmentée répond plus rapidement à une faute des moteurs que les deux autres compensateurs. Cette robustesse est observée lorsque les fautes surviennent avant le décollage ou durant le vol.

Les auteurs de [30] implémentent la MRAC augmentée, la CMRAC augmentée et un LQR sur un multirotor dans le but de les comparer, en vol, à la suite d'une panne partielle de moteur. Ils montrent ainsi que la méthode combinée possède un plus petit dépassement en mode suivi avant et après une panne de moteur par rapport à la MRAC augmentée et au contrôleur de base. De plus, la CMRAC s'adapte plus rapidement à la suite de la panne de moteur, ce qui lui permet d'avoir la plus petite erreur de suivi des trois types de commande. Cela dit, la MRAC et la CMRAC présentent de nettes améliorations par rapport à la commande LQ.

Enfin, on considère les contributions de [31] qui implémentent la MRAC augmentée sur un multirotor pour déterminer l'efficacité de la méthode en suivi de trajectoire. Ils comparent cette méthode à deux autres types de commandes non linéaires robustes : la commande par mode de glissement, i.e., *Sliding Mode Control* (SMC), et la commande par mode de glissement adaptative, i.e., *Adaptive Sliding Mode Control* (ASMC). Durant leurs travaux, ils démontrent que cette méthode est facile à implémenter tant dans un simulateur qu'expérimentalement par rapport aux autres méthodes. De plus, les résultats obtenus sont excellents ; meilleurs que ceux obtenus avec la SMC, mais légèrement inférieurs aux résultats obtenus avec la ASMC.

Les résultats et les conclusions des articles ci-dessus confirment la facilité à déployer la MRAC sur un multirotor. Les résultats obtenus montrent que cette commande est performante pour le rejet des perturbations paramétriques et non paramétriques. Néanmoins, la MRAC est conçue pour rejeter plusieurs types de perturbations. Or, la revue de littérature n'a pas permis de démontrer qu'elle améliorait effectivement les performances lorsque le multirotor vole en présence de toutes ces incertitudes.

2.1.3 Amélioration de la robustesse

Deux types de perturbations non paramétriques peuvent affecter un système dynamique : les perturbations non paramétriques assorties (*matched*) et les perturbations non paramétriques mal-assorties (*unmatched*). La première catégorie désigne les perturbations qui sont présentes dans le chemin direct de la commande, e.g., la force du vent et les dynamiques négligés. Le second type désigne les perturbations qui n'affectent pas directement les entrées du système, e.g., la présence de bruit de mesure. Il est plus difficile de rejeter les perturbations mal-assorties que les perturbations assorties [23]. En effet, il est possible de reformuler le problème de synthèse de la MRAC pour qu'elle puisse rejeter directement les perturbations assorties ; on peut alors prouver la convergence. Ce n'est toutefois pas le cas avec les perturbations mal-assorties. Avec ces perturbations, il n'est pas possible de garantir la stabilité du système avec la MRAC seule. Même si les perturbations demeurent bornées, les gains adaptatifs peuvent diverger. C'est un phénomène mieux connu sous le nom de dérive des gains [23]. La divergence des gains adaptatifs peut causer des problèmes de stabilité et mener à l'écrasement de l'appareil. Il faut procéder à l'ajout de modifications robustes pour empêcher ce phénomène de survenir ; ces modifications ont pour but de rejeter des perturbations non paramétriques mal-assorties.

Modification de la zone morte

La première modification robuste développée pour contrer le phénomène de dérive des gains est la modification de la zone morte, inventée par [32]. Cette modification consiste à arrêter le processus d'adaptation si l'erreur entre le modèle idéal et le système physique est inférieure à une borne prédéterminée. L'ajout de cette modification change la convergence de la commande adaptative. En effet, elle passe d'une stabilité asymptotique globale vers une stabilité ultimement uniformément bornée (UUB). À l'origine, la fonction mathématique représentant cette modification robuste est comparable à une fonction échelon de Heaviside. Cette modification n'est pas Lipschitz, i.e., la dérivée de la fonction n'est pas bornée. Cela peut causer le phénomène de réticence ou broutement (*chattering*), i.e., des oscillations à haute fréquence. Les travaux de [33] permettent d'obtenir une modification de la zone morte qui prévient la dérive des gains tout en étant lipschitzienne. Cette seconde méthode est plus complexe à implémenter que la précédente, mais elle permet d'éliminer les oscillations à haute fréquence qui peuvent causer l'instabilité.

Modification–σ

Les deux formes de la modification de la zone morte souffrent du même désavantage : il est nécessaire de connaître *a priori* la borne de l'erreur en deçà de laquelle il faut arrêter le processus d'adaptation. Or, il peut être difficile, voire impossible, de connaître cette borne, ce qui peut compromettre la bonne implémentation de cette modification robuste. Dans les travaux de [34], la loi adaptative est modifiée par l'inclusion d'un facteur d'oubli. Ce changement est appelé la modification–σ et elle comporte deux avantages :

— elle empêche le phénomène de dérive des gains ;
— elle retire la nécessité de connaître une borne de l'erreur *a priori*.

Cette modification, comme la précédente, change la stabilité de la MRAC ; on passe d'une stabilité asymptotique globale vers une stabilité UUB.

Dans [18], les auteurs analysent la synthèse de la MRAC avec différentes modifications robustes : la modification de la zone morte et la modification–σ. Ils montrent, en simulation, que les deux méthodes préviennent la dérive de gains. De plus, il n'y a aucune oscillation à haute fréquence avec la modification–σ, ce qui n'est pas le cas de la modification de la zone morte non lipschitzienne. Néanmoins, ils mettent en évidence un des problèmes de la modification–σ, le phénomène d'éclatement (*bursting*) des gains. Lorsque l'erreur entre le modèle de référence et le système diminue, il peut y avoir des épisodes d'explosion des gains, ce qui peut rendre le système complètement instable. Dans [35], les auteurs prouvent que ces phénomènes peuvent et vont survenir, même si les perturbations sont petites. Cependant, il est possible d'éliminer ce phénomène si le système est sous excitation persistante, i.e., si le système est commandé par un signal suffisamment riche et d'amplitude suffisante. Or, pour des applications de suivi de trajectoire, il n'est pas toujours possible de fournir une excitation persistante pour les entrées du système. Les auteurs de [35] proposent une solution alternative, mais ils ne démontrent pas qu'elle est parfaitement robuste, car de petites perturbations ne mènent pas nécessairement vers une erreur en sortie qui devient asymptotiquement petite.

Enfin, la modification–σ présente un dernier désavantage. Quand l'erreur en suivi tend vers zéro, les gains du système tendent vers leur valeur initiale. Le système oublie ce qu'il a appris durant l'adaptation. Ainsi, au prochain changement de la commande de référence où à la prochaine perturbation, le système doit s'adapter à nouveau.

Modification-ϵ

La modification–ϵ proposée par [36] améliore la modification robuste précédente en retirant la nécessité d'une excitation persistante pour garantir la robustesse du système. En effet, le facteur d'oubli de la modification–σ est remplacé par un facteur d'oubli qui est proportionnel à l'erreur entre le modèle de référence et le système physique ; le facteur d'oubli n'est plus fixe et peut varier dans le temps. Cela a pour effet d'éliminer le phénomène d'oubli de l'adaptation. Comme avec les autres modifications robustes, la stabilité obtenue avec cette méthode est UUB et elle n'est pas asymptotique. Toutefois, si l'entrée de référence est en excitation persistante, alors le système est exponentiellement stable. De plus, les simulations effectuées par [23] montrent que les gains adaptatifs tendent vers leurs valeurs idéales en excitation persistante.

Dans les travaux de [37], les auteurs souhaitent compenser le phénomène de dérive des gains qui est provoqué par des tremblements dans le système. En ajoutant la modification–ϵ à leur algorithme de commande, ils sont capables d'éliminer ce phénomène. Cependant, si les paramètres de design de la modification robuste sont mal choisis, des épisodes d'éclatement des gains sont encore possibles. Le système peut également répondre très lentement, car la modification–ϵ ralentit le système. Les travaux de [38] confirment cela. En effet, plus la modification robuste est forte, i.e., plus le facteur d'oubli est grand, plus le système est ralenti. En simulation, ils montrent que la présence de vent peut déstabiliser le système et mener à un épisode d'éclatement si la modification n'est pas bien réglée.

Retours sur les modifications robustes

Les modifications robustes ont pour but de garantir la stabilité du système sous l'influence de perturbations bornées non paramétriques mal-assorties. Par design, la MRAC ne garantit pas la stabilité du système dans de telles circonstances. Sans la présence d'une modification robuste, il est possible de démontrer que le système peut devenir instable et qu'il est vulnérable au phénomène de dérive des gains. La modification de la zone morte, la modification–σ et la modification–ϵ sont robustes par rapport aux perturbations non paramétriques mal-assorties et elles éliminent le phénomène de dérive des gains. Cependant, ces modifications possèdent plusieurs inconvénients, le principal étant qu'elles ralentissent le système. Elles vont ainsi à l'encontre de l'objectif de la commande adaptative, car elles réduisent son adaptabilité. D'autres méthodes doivent être développées pour permettre au système de conserver sa rapidité tout en étant robuste aux perturbations non paramétriques mal-assorties.

Opérateur de projection

Les méthodes précédentes modifiaient la loi adaptative de la MRAC en ajoutant un facteur d'oubli ou en limitant l'erreur de suivi. Ces méthodes présentent des solutions intéressantes, mais qui contiennent plusieurs désavantages, dont celui de nuire à l'adaptabilité du système. Plus récemment, l'utilisation de l'opérateur de projection pour la synthèse des lois adaptatives permet d'éviter les nombreux défauts des modifications robustes traditionnelles. En effet, l'utilisation de cet opérateur peut empêcher les pertes de performances du système asservi tout en garantissant une stabilité UUB pour tous les signaux du système soumis à des perturbations non paramétriques mal-assorties.

L'opérateur de projection a largement été étudié par le passé [21–23]. Il peut être utilisé dans les lois adaptatives pour confiner les gains adaptatifs dans un ensemble convexe où la stabilité de Lyapunov est garantie. Initialement, les fonctions de base avec lesquelles on construisait l'opérateur de projection n'étaient pas lipschitziennes, ce qui rendait son implémentation impossible expérimentalement. Les travaux de [39] ont permis d'obtenir un opérateur de projection lipschitzien en ajoutant une frontière limite autour de l'ensemble convexe. Notons également la contribution de [40] qui donne un opérateur de projection qui est différentiable autant que souhaité. Cette propriété est particulièrement importante pour les méthodes plus avancées de commande non linéaire.

Malgré ses succès théoriques, il existe très peu d'articles qui démontrent l'utilisation pratique de l'opérateur de projection pour la synthèse de lois adaptatives pour la MRAC. Il en existe encore moins concernant l'implémentation expérimentale de l'opérateur de projection sur un multirotor. Une synthèse est proposée dans [23], puis elle est appliquée en simulation à la gouverne de direction d'un avion. Les simulations démontrent que les lois de commande sont robustes aux perturbations externes et qu'elles peuvent limiter les gains adaptatifs dans un ensemble convexe. Dans cet ensemble, les gains peuvent évoluer librement sans affecter les performances du système. Les auteurs notent également que la MRAC construite avec l'opérateur de projection permet d'empêcher la saturation des intégrateurs (phénomène *anti-windup*) présente dans les lois de commande. Ils concluent en recommandant l'utilisation combinée de l'opérateur de projection et de la modification de la zone morte.

2.1.4 Commande robuste

Dans les sections précédentes, la commande adaptative et ses méthodes de synthèse ont été présentées. Nous avons montré que les algorithmes de commande issus de ces méthodes ne nécessitent pas la connaissance *a priori* des perturbations et des incertitudes présentes dans

le système. En effet, c'est à partir des mesures des variations entre le système réel et un modèle de référence que la commande adaptative permet de compenser ces incertitudes.

Les méthodes de commande robuste ont pour but d'atteindre le même résultat, mais avec une architecture de commande fixe. Pour atteindre cet objectif, il faut connaître *a priori* la forme des incertitudes et des perturbations, ainsi que leurs bornes. Les méthodes de synthèse de la commande robuste permettent d'obtenir une architecture de commande qui garantira la stabilité du système dans ces conditions incertaines. Pour déterminer si le système est robustement stable, plusieurs techniques d'analyse sont disponibles dont la μ–analyse et les applications gardiennes. Celles-ci ont pour objectif d'analyser la stabilité robuste et la performance robuste d'un système ayant de multiples incertitudes [41, Chapitre 11]. On parle alors de :

— **stabilité robuste**, si le système demeure stable pour toutes les incertitudes ;
— **performance robuste**, si les performances du système sont conservées pour toutes les incertitudes.

μ–analyse

La valeur singulière structurée (VSS) μ permet de mesurer la performance robuste d'un système quand son modèle contient plusieurs incertitudes. Pour utiliser la μ–analyse, il faut que le système soit exprimée sous la forme d'une *Linear Fractional Transformation* (LFT). Cette forme a la particularité de rassembler dans une matrice $\mathbf{\Delta}$ toutes les incertitudes du système. La μ–analyse permet alors d'étudier la stabilité du système en présence de ces incertitudes. Le lecteur peut se référer à [41, Chapitre 11] pour les principes fondamentaux de la μ–analyse.

Applications gardiennes

Développées par [42], les applications gardiennes sont un outil d'analyse de la stabilité généralisée de familles paramétrées de matrices réelles et de polynômes. La stabilité généralisée correspond au confinement des valeurs propres d'une matrice ou des racines d'un polynôme dans un domaine Ω du plan complexe. Le domaine Ω est choisi selon des considérations de performance : stabilité, amortissement désiré, temps de réponse, etc.

Les applications gardiennes sont des fonctions scalaires des paramètres incertains de la famille de matrices ou de polynômes considérée, qui ont la propriété de s'annuler dès que l'on est à la limite du domaine de stabilité Ω. Il est ainsi possible de construire des applications

gardiennes correspondant à une grande variété de domaines Ω selon le confinement de valeurs propres ou de pôles que l'on souhaite vérifier. Grâce aux applications gardiennes, on peut alors déterminer des ensembles de valeurs des paramètres qui assurent que la famille de matrices ou de polynômes demeure stable par rapport à Ω.

Polyvalentes, les applications gardiennes analysent la stabilité généralisée d'un système en fonction de certains paramètres d'intérêt. Cependant, la méthode de résolution est généralement graphique. Or, s'il y a plus de trois paramètres d'intérêt, l'analyse devient difficile, voire impossible. Il faut donc passer à des méthodes numériques pour obtenir les informations souhaitées sur la stabilité généralisée en fonction de tous ses paramètres.

Lien entre les deux méthodes d'analyse

Les travaux de [43] font la transition entre les applications gardiennes et la μ–analyse lorsque le système étudié est une matrice à intervalle, i.e., certains des éléments de la matrice peuvent varier sur un intervalle. Donc, la matrice possède un minimum et un maximum selon les éléments qui la composent et qui peuvent varier.

Si le système est nominalement stable, alors il est possible d'exprimer cette matrice à intervalle comme une LFT. Les travaux de [44] démontrent que l'utilisation de produits de Kronecker et de sommes de Kronecker sur des LFT donne aussi des LFT. Cette conclusion est importante, car la somme et le produit de Kronecker sont fortement utilisés dans la construction des applications gardiennes. Ces conclusions permettent de démontrer qu'il est possible d'utiliser les concepts définis par [42] directement sur les LFT. Enfin, les auteurs de [43] prouvent que la matrice à intervalle est stable par rapport à son application gardienne, si le résultat de la μ–analyse est inférieur à une constante donnée.

2.2 Objectifs de recherche

Forts de cette revue de littérature, nous proposons l'objectif global dans le cadre de ce projet de recherche.

Développer un contrôleur adaptatif par modèle de référence robuste aux perturbations et incertitudes paramétriques et non paramétriques, assorties et mal-assorties, pour des applications de maintien de position d'un quadricoptère.

Cet objectif de recherche est séparé en quatre sous-objectifs qui sont

1. Modélisation de la dynamique d'un multirotor, des incertitudes et des perturbations ;
2. Synthèse d'une MRAC augmentée ;
3. Synthèse des modifications robustes traditionnelles pour la MRAC augmentée ;
4. Synthèse de la MRAC avec opérateur de projection

Nous reviendrons en détail sur chacun de ces sous-objectifs dans les sous-sections suivantes.

2.2.1 Sous-objectif #1 : modélisation de la dynamique d'un multirotor, des incertitudes et des perturbations

La modélisation de la dynamique du multirotor est primordiale pour toute synthèse de lois de commande. Si la modélisation est adéquate, alors les simulations seront plus réalistes et la synthèse des lois de commande n'en sera que plus fiable. Les sources d'incertitudes doivent être bien identifiées afin de synthétiser une loi de commande adaptative qui permettra de les compenser.

Méthodologie et contribution. La revue de littérature n'a pas permis d'obtenir un modèle complet de la dynamique du multirotor qui prend en compte toutes les perturbations et les incertitudes. Nous proposons de partir du modèle non linéaire établi par les travaux de [12], et de le simplifier afin de ne conserver que les termes principaux qui influencent la dynamique de l'appareil en vol stationnaire. Ce modèle sera ensuite linéarisé, et nous y ajouterons les perturbations et les incertitudes en suivant les propositions de [18, 23]. L'ajout progressif des incertitudes et des perturbations n'est généralement pas abordé de façon satisfaisante selon les sources consultées. En les introduisant progressivement au modèle linéarisé, la synthèse de la MRAC deviendra plus intuitive. Une grande attention sera portée à bien expliquer la provenance des sources des incertitudes et des perturbations.

Le modèle complet permettra ainsi d'obtenir des simulations réalistes et d'effectuer divers tests pour mesurer la performance des lois de commande. De plus, les articles qui présentent des résultats expérimentaux utilisent rarement un critère de performance rigoureux pour la validation de leurs contrôleurs. À cette fin, nous utiliserons une méthode d'analyse de la performance basée sur l'intégrale de l'erreur quadratique pondérée en temps, i.e., *Integral Time-Weighted Squared Error* (ITSE), pour déterminer la convergence temporelle du système réel par rapport au modèle de référence [1].

2.2.2 Sous-objectif #2 : synthèse d'une MRAC augmentée

Trois phases sont nécessaires pour la synthèse d'une MRAC augmentée. D'abord, il faut effectuer la synthèse d'un contrôleur nominal. Ce contrôleur est généralement obtenu sous la forme d'un PID, d'un placement de pôle ou d'un LQR. La seconde phase consiste à synthétiser la loi d'adaptation de la MRAC qui permet d'assurer la stabilité du système face aux incertitudes et perturbations choisies. Enfin, les deux lois de commande sont implémentées sur le multirotor et ajustées expérimentalement pour obtenir les performances temporelles désirées.

Méthodologie et contribution. Pour ce sous-objectif, le contrôleur nominal choisi sera obtenu par la commande LQ. La synthèse des lois adaptatives sera basée en partie sur les travaux de [30] qui permettent de compenser les incertitudes paramétriques et non paramétriques assorties. Le réglage des lois de commande est généralement ardu et requiert plusieurs itérations. Notre contribution anticipée est la proposition d'une méthode d'ajustement des paramètres de design qui permet d'obtenir rapidement et efficacement un système expérimental opérationnel. Cette méthode reposera sur un processus itératif entre MATLAB/SIMULINK et ROS (*Robot Operating System*) dans l'environnement virtuel de simulation, Gazebo.

Généralement, les tests effectués en simulation pour l'ajustement des paramètres de design consistent à exciter une seule partie de la dynamique du multirotor à la fois. Or, nous contestons cette méthode en proposant des tests qui excitent l'ensemble des dynamiques de l'appareil simultanément. Ces simulations permettront de comprendre plus intuitivement les effets des paramètres de design sur les performances du système tout en fournissant un maximum d'information à chacune des itérations.

L'utilisation de cette méthode de simulation pour le réglage des lois de commande nous a permis d'obtenir en quelques semaines un système opérationnel, alors qu'il faut généralement plus de temps avec la méthode de réglage traditionnelle. De plus, des détails sont donnés quant à l'intuition nécessaire pour ajuster les lois de commande pour que chacune des itérations soit numériquement plus performante que la précédente.

2.2.3 Sous-objectif #3 : synthèse des modifications robustes traditionnelles pour la MRAC augmentée

La synthèse d'une loi de commande robuste aux perturbations non paramétriques mal-assorties est une tâche relativement difficile qui est abordée seulement depuis une vingtaine d'années. Dans le contexte d'une MRAC, les modifications robustes viennent modifier la loi

d'adaptation pour rejeter ces perturbations. Or, ces modifications robustes traditionnelles ont généralement comme désavantage de ralentir le processus d'adaptation du système en arrêtant l'adaptation lorsque l'erreur entre le modèle idéal et le système réel est suffisamment petite. Les trois modifications robustes traditionnelles présentées précédemment ont été très peu implémentées sur un multirotor. En simulation, elles donnent de bons résultats, mais le manque de validations expérimentales n'a pas encore permis de démontrer que cela est aussi vrai en pratique.

Méthodologie et contribution. Dans ce projet de recherche, nous déployons ces trois modifications robustes sur le multirotor en nous fiant sur les expressions fournies dans [23]. Notre objectif est de vérifier expérimentalement si ces modifications permettent de rejeter des perturbations non paramétriques mal-assorties. À cette fin, nous effectuerons une étude comparative entre les différentes méthodes dans le but de déterminer les forces et les faiblesses de chacune. Ultimement, nous souhaitons être en mesure de fournir des recommandations quant à la meilleure modification à implémenter sur un multirotor.

2.2.4 Sous-objectif #4 : synthèse de la MRAC avec opérateur de projection

Le ralentissement de l'adaptabilité du système causé par les modifications robustes précédentes va à l'encontre de l'objectif de la commande adaptative. C'est un inconvénient majeur lorsque l'on souhaite que le système soit rapide et réactif aux changements. Ainsi, des recherches ont été effectuées pour développer une autre modification robuste qui permet de rejeter les perturbations non paramétriques mal-assorties sans limiter la vitesse d'adaptation du système. En construisant la MRAC avec l'opérateur de projection, il est possible d'atteindre cet objectif [23, 39].

Méthodologie et contribution. Pour utiliser adéquatement l'opérateur de projection, il faut définir, *a priori*, un grand nombre de paramètres. Ces constantes serviront de bornes pour limiter les gains adaptatifs. Or, notre revue de littérature n'a pas permis de trouver une méthode qui permet d'obtenir ces bornes de façon systématique. Notre première contribution à ce sujet est de proposer un algorithme qui peut les obtenir systématiquement. L'avantage de cet algorithme est qu'il unifie deux domaines de la théorie de la commande : la commande adaptative et la commande robuste. Grâce aux applications gardiennes [42] et à la μ–analyse [41], il nous est possible de construire un algorithme d'optimisation qui repose sur les travaux de [43]. Cet algorithme, en retour, nous donne les bornes sur les gains adaptatifs qui sont nécessaires afin de synthétiser l'opérateur de projection.

Notre seconde contribution consiste en la validation expérimentale du fonctionnement de

la MRAC construite à l'aide de l'opérateur de projection. Durant la revue de littérature, nous n'avons trouvé qu'un seul article discutant de l'implantation de cet opérateur dans une loi d'adaptation [30]. Cependant, aucun détail n'était donné quant aux considérations pratiques à prendre pour l'implémenter convenablement. De plus, l'article n'aborde pas le sujet des bornes des gains adaptatifs. En combinant notre algorithme pour l'obtention des bornes des gains adaptatifs avec une description suffisamment détaillée de l'implantation de cet opérateur, nous fournissons tout ce qui est nécessaire pour que de futurs chercheurs puissent implémenter facilement une MRAC construite avec l'opérateur de projection sur un multirotor.

CHAPITRE 3 MODÉLISATION DU QUADRICOPTÈRE ET MONTAGE EXPÉRIMENTAL

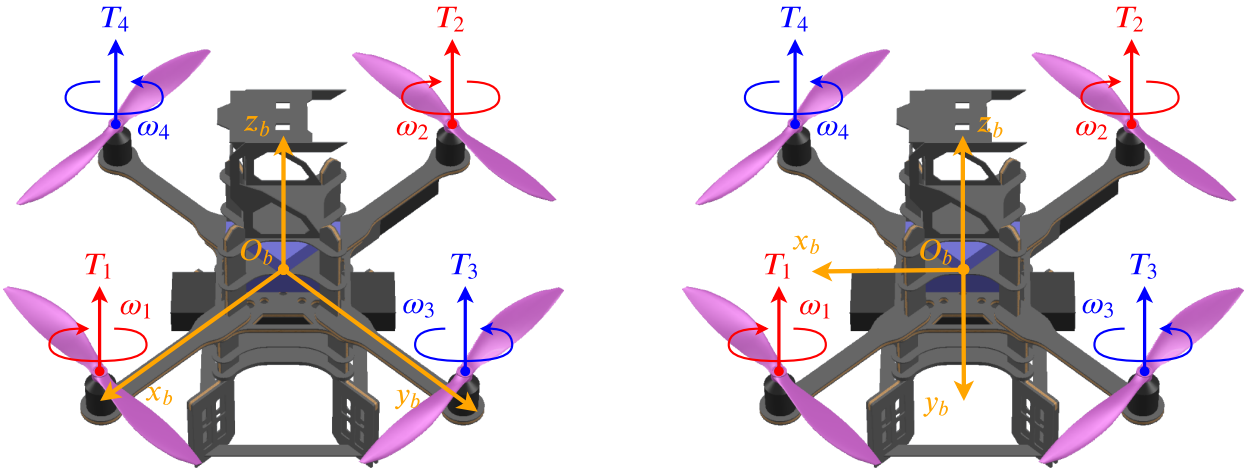
Dans ce chapitre, nous abordons la modélisation non linéaire complète d'un quadricoptère (Fig. 3.1). Nous nous référerons à [8, 9, 12] pour la modélisation de la dynamique du drone, à [45] pour la modélisation des moteurs et à [18, 23, 30] pour la modélisation des incertitudes. Les forces et les moments qui affectent le drone sont définis à l'aide de [12, 46]. Notons que trois hypothèses simplificatrices sont utilisées pour obtenir le modèle dynamique :

HS1. La Terre est plate avec un vecteur gravité constant ;

HS2. Le quadricoptère est un solide indéformable, donc considéré comme un corps rigide ;

HS3. Le quadricoptère possède une double symétrie géométrique et propulsive dans les plans $\mathbf{x}_b$–$\mathbf{z}_b$ et $\mathbf{y}_b$–$\mathbf{z}_b$.

Ces hypothèses permettent d'obtenir un modèle simplifié de la dynamique du drone. De **HS1**, on néglige les effets de courbure de la Terre et l'on considère que le vecteur de gravité est constant et pointe vers le bas. De **HS2**, on considère le quadricoptère comme un corps rigide et l'on peut négliger ses déformations, et donc les modes flexibles associés. Enfin, de **HS3**, on peut simplifier la matrice des moments d'inertie pour qu'elle soit diagonale.



(a) Configuration « + » (b) Configuration « × »

Figure 3.1 Configurations de l'AscTec Pelican

Dans la section 3.1, nous développons les équations non linéaires de la dynamique du quadricoptère. Nous linéarisons ensuite ces équations dans le cas du vol stationnaire. La section

3.2 est consacrée à la modélisation des incertitudes pouvant affecter le quadricoptère : les incertitudes paramétriques et les perturbations non paramétriques assorties et mal-assorties. Enfin, nous donnons le montage expérimental utilisé pour le quadricoptère Pelican d'AscTec [1] dans la section 3.3 ainsi que quelques considérations pratiques pour la bonne mise en œuvre des lois de commande.

3.1 Modèle dynamique du quadricoptère

3.1.1 Repères de référence et définition des angles

Deux repères sont nécessaires afin de développer les équations de la dynamique du quadricoptère.

- **Repère inertiel** $\mathcal{F}_i = \{O_i,\ \mathbf{x}_i,\ \mathbf{y}_i,\ \mathbf{z}_i\}$
  Le repère inertiel noté $\mathcal{F}_i$ est un repère fixe dont l'origine O_i est un point arbitraire, e.g., la position initiale du quadricoptère. L'orientation du repère suit la convention *East-North-Up* (ENU). Plus spécifiquement, l'axe $\mathbf{x}_i$ point vers l'Est, l'axe $\mathbf{y}_i$ vers le Nord et l'axe $\mathbf{z}_i$ vers le haut.
- **Repère objet** $\mathcal{F}_b = \{O_b,\ \mathbf{x}_b,\ \mathbf{y}_b,\ \mathbf{z}_b\}$
  Le repère objet noté $\mathcal{F}_b$ est lié au quadricoptère et son origine O_b est généralement placée au centre de masse. Ce repère suit la convention *Front-Left-Up* (FLU). On utilise l'indice b pour *Body*. L'axe $\mathbf{x}_b$ est orienté vers l'« avant » du drone, l'axe $\mathbf{z}_b$ vers le haut et l'axe $\mathbf{y}_b$ complète le trièdre direct sur la gauche. Deux configurations sont généralement utilisées pour l'assignation des axes du repère objet :
  - la configuration « + » où les axes $\mathbf{x}_b$ et $\mathbf{y}_b$ sont positionnés le long des bras du quadricoptère (Fig. 3.1a) ;
  - la configuration « × » où les axes $\mathbf{x}_b$ et $\mathbf{y}_b$ sont positionnés entre les bras du quadricoptère (Fig. 3.1b).

  Nous verrons dans la section 3.3.2 les avantages et les inconvénients de ces deux configurations. Les équations dynamiques peuvent néanmoins être développées indépendamment de la configuration choisie.

Le passage du repère inertiel $\mathcal{F}_i$ vers le repère objet $\mathcal{F}_b$ nécessite trois rotations :

1. Rotation autour de l'axe $\mathbf{z}_i$ d'un angle ψ, appelé angle de lacet. Le nouveau repère obtenu est noté $\mathcal{F}_1 = \{O_1,\ \mathbf{x}_1,\ \mathbf{y}_1,\ \mathbf{z}_1\}$;

1. Ascending Technologies, `http://www.asctec.de/en/` .

2. Rotation autour de l'axe $\mathbf{y}_1$ d'un angle θ, appelé angle de tangage. Le nouveau repère obtenu est noté $\mathcal{F}_2 = \{O_2,\ \mathbf{x}_2,\ \mathbf{y}_2,\ \mathbf{z}_2\}$;
3. Rotation autour de l'axe x_2 d'un angle ϕ, appelé angle de roulis. Le repère obtenu est le repère objet $\mathcal{F}_b$.

Les trois angles de rotation sont appelés les angles d'Euler, rassemblé dans le vecteur $\mathbf{\Phi} = [\phi\ \ \theta\ \ \psi]^\top$ et les matrices de rotation correspondantes sont données par :

$$\mathbf{R}_{\mathbf{z}_i}(\psi) = \begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_{\mathbf{y}_1}(\theta) = \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix}, \quad \mathbf{R}_{\mathbf{x}_2}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix}, \tag{3.1}$$

avec, pour un angle γ quelconque, $c_\gamma = \cos\gamma$ et $s_\gamma = \sin\gamma$. La matrice de passage du repère inertiel vers le repère objet est appelée la matrice de rotation d'Euler et elle est donnée par

$$\begin{aligned} \mathbf{R}_{b/i} &= \mathbf{R}_{\mathbf{x}_2}(\phi)\mathbf{R}_{\mathbf{y}_1}(\theta)\mathbf{R}_{\mathbf{z}_i}(\psi), \\ &= \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix}. \end{aligned} \tag{3.2}$$

Cette matrice de rotation nous permet de projeter un vecteur quelconque du repère inertiel $\mathcal{F}_i$ dans le repère objet $\mathcal{F}_b$:

$$\mathbf{x}^b = \mathbf{R}_{b/i}\mathbf{x}^i. \tag{3.3}$$

De plus, comme une matrice de rotation est orthogonale, son inverse est égale à sa transposée. On a ainsi $\mathbf{R}_{i/b} = \mathbf{R}_{b/i}^{-1} = \mathbf{R}_{b/i}^\top$.

3.1.2 Équations cinématiques

Deux équations cinématiques sont nécessaires pour la modélisation du comportement du quadricoptère : l'équation de navigation et l'équation d'Euler. La première permet de relier la vitesse du drone dans le repère objet avec sa vitesse dans le repère inertiel tandis que la seconde permet de relier la vitesse angulaire de l'appareil dans le repère objet avec le taux de variation des angles d'Euler.

Équation de navigation

On note $\mathbf{v}_{cm/i}$ le vecteur vitesse du centre de masse par rapport au repère inertiel $\mathcal{F}_i$ et $\mathbf{v}^b_{cm/i} = [u\ \ v\ \ w]^\top$ sa projection dans le repère objet $\mathcal{F}_b$. On définit aussi $\mathbf{p}_{cm/i}$ le vecteur

position du centre de masse par rapport à l'origine O_i du repère $\mathcal{F}_i$, ainsi que sa projection dans le repère inertiel $\mathbf{p}^i_{cm/i} = [x \quad y \quad z]^\top$. En dérivant le vecteur $\mathbf{p}_{cm/i}$ dans le repère $\mathcal{F}_i$, l'équation de navigation est alors donnée par :

$$\begin{aligned} {}^i\dot{\mathbf{p}}^i_{cm/i} &= \mathbf{v}^i_{cm/i} \\ &= \mathbf{R}_{i/b}\mathbf{v}^b_{cm/i}. \end{aligned} \tag{3.4}$$

Équation d'Euler

On note $\boldsymbol{\omega}_{b/i}$ le vecteur vitesse de rotation de $\mathcal{F}_b$ par rapport à $\mathcal{F}_i$ et $\boldsymbol{\omega}^b_{b/i} = [p \quad q \quad r]^\top$ sa projection dans le repère objet $\mathcal{F}_b$. On veut projeter les trois vitesses de rotation élémentaires dans le repère objet. Ces vitesses de rotation élémentaires sont :

- $\boldsymbol{\omega}^i_{1/i} = [0 \;\; 0 \;\; \dot{\psi}]^\top$ dans le repère inertiel ;
- $\boldsymbol{\omega}^1_{2/1} = [0 \;\; \dot{\theta} \;\; 0]^\top$ dans le repère $\mathcal{F}_1$;
- $\boldsymbol{\omega}^2_{b/2} = [\dot{\phi} \;\; 0 \;\; 0]^\top$ dans le repère $\mathcal{F}_2$.

Ainsi, la projection dans le repère objet est

$$\begin{aligned} \boldsymbol{\omega}^b_{b/i} &= \boldsymbol{\omega}^2_{b/2} + \mathbf{R}_{\mathbf{x}_2}(\phi)\,\boldsymbol{\omega}^1_{2/1} + \mathbf{R}_{\mathbf{x}_2}(\phi)\,\mathbf{R}_{\mathbf{y}_1}(\theta)\,\boldsymbol{\omega}^i_{1/i}, \\ &= \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \\ &= \mathbf{H}(\boldsymbol{\Phi})\,\dot{\boldsymbol{\Phi}}. \end{aligned} \tag{3.5}$$

3.1.3 Équations dynamiques

Les équations de la dynamique du quadricoptère peuvent être divisées en deux types : les équations de forces et les équations de moments. Les équations de forces régissent principalement le mouvement de translation du quadricoptère tandis que les équations de moments régissent principalement le mouvement de rotation de l'appareil. On dit « principalement », car les équations sont tout de même couplées.

Équations de forces

L'application dans le repère inertiel de la deuxième loi de Newton nous donne la relation :

$$m\mathbf{a} = m\,{}^i\dot{\mathbf{v}}_{cm/i} = \mathbf{F}, \tag{3.6}$$

où m désigne la masse du quadricoptère, $\mathbf{a} = {}^{i}\dot{\mathbf{v}}_{cm/i}$ l'accélération linéaire du centre de masse du quadricoptère dans le repère inertiel et $\mathbf{F}$ la somme des forces extérieures.

L'équation de Coriolis nous permet de relier les dérivées de $\mathbf{v}_{cm/i}$ dans le repère inertiel ${}^{i}\dot{\mathbf{v}}_{cm/i}$ et dans le repère objet ${}^{b}\dot{\mathbf{v}}_{cm/i}$ grâce au vecteur instantanée de rotation $\boldsymbol{\omega}_{b/i}$ sous la forme :

$$ {}^{i}\dot{\mathbf{v}}_{cm/i} = {}^{b}\dot{\mathbf{v}}_{cm/i} + \boldsymbol{\omega}_{b/i} \times \mathbf{v}_{cm/i}, \tag{3.7} $$

où $\times$ désigne le produit vectoriel. La somme des forces $\mathbf{F}$ rassemble généralement :

- $\mathbf{F}_t$ les forces propulsives des rotors ;
- $\mathbf{F}_g$ les forces gravitationnelles ;
- $\mathbf{F}_a$ les forces aérodynamiques.

Toutefois, étant donné les objectifs de la commande adaptative, il n'est pas nécessaire de construire un modèle parfait de la dynamique du quadricoptère, mais un modèle suffisant pour des besoins de commande. Ainsi, nous nous contenterons d'inclure les forces de propulsion des rotors et les forces gravitationnelles. Ces forces sont les principales qui affectent le quadricoptère, surtout si celui-ci est en vol stationnaire. Nous reviendrons sur ce point dans la section 3.1.4. La projection dans le repère $\mathcal{F}_b$ de l'équation de forces donne alors :

$$ m\left({}^{b}\dot{\mathbf{v}}^{b}_{cm/i} + \boldsymbol{\omega}^{b}_{b/i} \times \mathbf{v}^{b}_{cm/i}\right) = \mathbf{F}^{b}_{g} + \mathbf{F}^{b}_{t}. \tag{3.8} $$

De **HS1**, on sait que le vecteur de force gravitationnelle pointe vers le bas et est constant. Dans le repère inertiel, il est donné par $\mathbf{g}^i = [0 \;\; 0 \;\; -g_0]^\top$ où $g_0 = 9.80655$ m/s^2. Ainsi, dans le repère inertiel, la force gravitationnelle est $\mathbf{F}^i_g = m\mathbf{g}^i$. Il suffit d'utiliser la matrice de rotation d'Euler pour projeter ce vecteur dans le repère objet, soit :

$$ \mathbf{F}^b_g = \mathbf{R}_{b/i}\mathbf{F}^i_g = mg_0\mathbf{R}_{b/i}\mathbf{g}^i = \begin{bmatrix} mg_0 \sin\theta \\ -mg_0 \sin\phi\cos\theta \\ -mg_0 \cos\phi\cos\theta \end{bmatrix}. \tag{3.9} $$

Dans [46, Sec. 2.5], on modélise la force verticale générée par une hélice comme

$$ T = C_T \rho A_r r^2 \omega^2 = k_T \omega^2, \tag{3.10} $$

où C_T est le coefficient de poussée de l'hélice, ρ la densité de l'air, A_r la superficie circulaire que couvre l'hélice en rotation, r le rayon de l'hélice et ω la vitesse angulaire de l'hélice. On peut rassembler l'ensemble des constantes pour former un coefficient de poussée $k_T = C_T \rho A_r r^2$.

Dans le repère objet, les forces de poussée des moteurs sont alors modélisées par :

$$\mathbf{F}_t^b = \sum_{i=1}^{4} \mathbf{T}_i^b \ , \quad \mathbf{T}_i^b = \begin{bmatrix} 0 \\ 0 \\ k_T \omega_i^2 \end{bmatrix}, \tag{3.11}$$

où $i \in \{1,\ 2,\ 3,\ 4\}$ désigne le moteur i. La force de poussée totale F_z est alors donnée par :

$$F_z = k_T \left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right). \tag{3.12}$$

Notons toutefois qu'il peut être plus intuitif de considérer l'équation de forces dans le repère inertiel. Auquel cas, on a :

$$m^i \dot{\mathbf{v}}_{cm/i}^i = \mathbf{F}_g^i + \mathbf{F}_t^i. \tag{3.13}$$

En utilisant les notations précédentes, on obtient :

$$m^i \ddot{\mathbf{p}}_{cm/i}^i = m\mathbf{g}^i + \mathbf{R}_{b/i}^\top \mathbf{F}_t^b \tag{3.14}$$

ce qui permet de s'affranchir de l'équation de navigation. On retrouvera ces équations complètes développées à l'annexe A.1.

Équations de moments

Le théorème du moment cinétique, déduit de la deuxième loi de Newton, nous donne la relation :

$${}^i\dot{\mathbf{h}}_{cm} = \mathbf{M}, \tag{3.15}$$

où $\mathbf{h}_{cm}$ désigne le moment cinétique au centre de masse et $\mathbf{M}$ la somme des moments appliqués au quadricoptère. L'équation de Coriolis nous permet alors d'écrire :

$${}^i\dot{\mathbf{h}}_{cm} = {}^b\dot{\mathbf{h}}_{cm} + \boldsymbol{\omega}_{b/i} \times \mathbf{h}_{cm}. \tag{3.16}$$

Le moment total $\mathbf{M}$ appliqué au drone peut être composé de plusieurs éléments :

- $\mathbf{M}_t$ le moment induit par la force de poussée $\mathbf{T}_i$ des rotors et du bras de levier ;
- $\mathbf{M}_q$ le moment induit par la traînée des hélices ;
- $\mathbf{M}_g$ le moment gyroscopique.

Pour notre modélisation, nous conserverons les deux premiers moments, i.e., $\mathbf{M}_t$ et $\mathbf{M}_q$.

L'expression du moment cinétique dans le repère objet est donnée par :

$$\mathbf{h}_{cm}^b = \mathbf{I}^b \boldsymbol{\omega}_{b/i}^b \ , \quad \mathbf{I}^b = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix}, \tag{3.17}$$

où $\mathbf{I}^b \in \mathbb{S}^3$ désigne la matrice d'inertie du quadricoptère, symétrique et calculée dans le repère objet. L'hypothèse de double symétrie du drone (**HS3**) nous permet de simplifier $\mathbf{I}^b$ en une matrice diagonale donnée par $\mathbf{I}^b = \operatorname{diag}(I_{xx}, I_{yy}, I_{zz})$. La projection dans le repère $\mathcal{F}_b$ de l'équation de moments donne alors :

$$\mathbf{I}^b \, {}^b\dot{\boldsymbol{\omega}}_{b/i}^b + \boldsymbol{\omega}_{b/i}^b \times \mathbf{I}^b \boldsymbol{\omega}_{b/i}^b = \mathbf{M}_t^b + \mathbf{M}_q^b, \tag{3.18}$$

où ${}^b\dot{\boldsymbol{\omega}}_{b/i}^b = [\dot{p} \ \ \dot{q} \ \ \dot{r}]^\top$. Le moment $\mathbf{M}$ créé par une force $\mathbf{F}$ exercée sur un bras de levier $\mathbf{r}$ peut être déterminé par $\mathbf{M} = \mathbf{r} \times \mathbf{F}$. En notant $\mathbf{r}_i^b$ la position de l'hélice i dans le repère objet $\mathcal{F}_b$, l'expression générale de $\mathbf{M}_t^b$ est donnée par :

$$\mathbf{M}_t^b = \sum_{i=1}^{4} \mathbf{r}_i^b \times \mathbf{T}_i^b, \tag{3.19}$$

avec pour une configuration « + » :

$$\begin{bmatrix} \mathbf{r}_1^b & \mathbf{r}_2^b & \mathbf{r}_3^b & \mathbf{r}_4^b \end{bmatrix} = \begin{bmatrix} l & -l & 0 & 0 \\ 0 & 0 & l & -l \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.20}$$

et pour une configuration « × » :

$$\begin{bmatrix} \mathbf{r}_1^b & \mathbf{r}_2^b & \mathbf{r}_3^b & \mathbf{r}_4^b \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2}l & -\frac{\sqrt{2}}{2}l & -\frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l \\ \frac{\sqrt{2}}{2}l & -\frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & -\frac{\sqrt{2}}{2}l \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.21}$$

où l est la distance entre le centre de masse du quadricoptère et l'hélice. On suppose ici que les hélices sont au même niveau que le centre de masse selon l'axe $\mathbf{z}_b$ (Fig. 3.1). Selon [46, Sec. 2.5.], on peut modéliser le moment induit par la traînée des hélices par

$$Q = C_Q \rho A_r r^2 \omega^2 = k_D \omega^2, \tag{3.22}$$

où C_Q est le coefficient de traînée des hélices. On peut alors rassembler les constantes pour

former un nouveau coefficient de traînée $k_D = C_Q \rho A_r r^2$. Le moment induit est alors donné par

$$\mathbf{M}_q^b = \sum_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ s_r k_D \omega_i^2 \end{bmatrix}, \tag{3.23}$$

où s_r désigne le sens de rotation des hélices. Si l'hélice tourne dans le sens horaire, alors $s_r = +1$, sinon $s_r = -1$. On peut ainsi regrouper les équations (3.19) et (3.23) et déterminer l'expression des moments qui influencent le quadricoptère. Pour une configuration « + », on a ainsi :

$$\mathbf{M}^b = \mathbf{M}_t^b + \mathbf{M}_q^b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} l k_T \left(\omega_3^2 - \omega_4^2\right) \\ l k_T \left(\omega_2^2 - \omega_1^2\right) \\ k_D \left(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2\right) \end{bmatrix}, \tag{3.24}$$

et pour une configuration « × »

$$\mathbf{M}^b = \mathbf{M}_t^b + \mathbf{M}_q^b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} l k_T \left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) \\ \frac{\sqrt{2}}{2} l k_T \left(-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2\right) \\ k_D \left(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2\right) \end{bmatrix}. \tag{3.25}$$

Force de traînée par le corps du quadricoptère

Parmi les forces agissant sur le quadricoptère, nous avons négligé la force de traînée induite par le corps du quadricoptère. Cette dynamique négligée peut être représentée par une perturbation non paramétrique assortie lors des simulations de la MRAC. Le modèle de traînée induite proposé ici repose sur l'hypothèse qu'il n'y a pas de vent qui affecte le quadricoptère. Grâce à cette simplification, on obtient dans le repère objet la modélisation suivante, tirée de [46] :

$$\mathbf{F}_d^b = \begin{bmatrix} F_{d,x} \\ F_{d,y} \\ F_{d,z} \end{bmatrix} = -\frac{1}{2} \rho C_d \begin{bmatrix} A_x u \left|u\right| \\ A_y v \left|v\right| \\ A_z w \left|w\right| \end{bmatrix}, \tag{3.26}$$

où ρ est la masse volumique de l'air, C_d la constante de traînée du quadricoptère et A_x, A_y, A_z les aires projetées respectivement dans les plans $\mathbf{y}_b$–$\mathbf{z}_b$, $\mathbf{x}_b$–$\mathbf{z}_b$ et $\mathbf{x}_b$– $\mathbf{y}_b$ telles qu'illustrées à la figure A.1 de l'annexe A.2. La modélisation de la force de traînée ne sera pas utilisée directement dans la synthèse des contrôleurs. On l'utilisera plutôt comme une perturbation non paramétrique assortie, ce qui demeure réaliste par rapport au système étudié. Cela permet, entre autres, de démontrer si la MRAC peut réellement compenser de telles incertitudes.

Modèle non linéaire complet

Les équations obtenues dans les sections précédentes (Eqs. 3.4, 3.5, 3.8, 3.18) nous permettent d'obtenir le modèle non linéaire complet du quadricoptère. On représente toutes ces équations sous la forme d'un modèle d'état :

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right), \\ \mathbf{y} &= \mathbf{h}\left(\mathbf{x}, \mathbf{u}\right), \end{aligned} \tag{3.27}$$

où $\mathbf{x}$ désigne le vecteur d'état, $\mathbf{v}$ le vecteur d'entrée, $\mathbf{y}$ le vecteur de sortie, et $\mathbf{f}$, $\mathbf{h}$ des fonctions vectorielles non linéaires. Les vecteurs $\mathbf{x}$, $\mathbf{v}$ et $\mathbf{y}$ sont respectivement :

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x & y & z & u & v & w & \phi & \theta & \psi & p & q & r \end{bmatrix}^\top, \\ \mathbf{v} &= \begin{bmatrix} F_z & M_x & M_y & M_z \end{bmatrix}^\top, \\ \mathbf{y} &= \begin{bmatrix} x & y & z & \psi \end{bmatrix}^\top. \end{aligned} \tag{3.28}$$

Le développement complet du modèle d'état est donné à l'annexe A.1.

Matrice d'allocation et commande virtuelle

Bien qu'il soit possible de synthétiser le contrôleur pour qu'il commande directement la vitesse de rotation des moteurs, il est plus intuitif de travailler directement avec la force F_z et les moments M_x, M_y et M_z. En effet, les équations se découplent naturellement et chacune des entrées n'influence qu'une partie de la dynamique du quadricoptère. Nous montrons cela formellement à l'annexe A.1. Cependant, si les poussées T_i des moteurs devaient être les entrées du système (via la commande de leur vitesse de rotation ω_i), nous aurions alors une influence des entrées sur l'ensemble de la dynamique du quadricoptère. Il serait plus difficile de concevoir les lois de commande et de les implanter efficacement sur le quadricoptère.

Ceci dit, il faut pouvoir convertir la force F_z et les moments M_x, M_y et M_z en poussées T_i des moteurs, et subséquemment en vitesses de rotation ω_i. Si l'on considère que le vecteur d'entrée $\mathbf{v}$ est la commande virtuelle, on désigne par $\mathbf{u} = [T_1 \;\; T_2 \;\; T_3 \;\; T_4]^\top$ la commande réelle avec $T_i = k_T \omega_i^2$. On peut ainsi définir une matrice d'allocation $\boldsymbol{\Upsilon}$ pour relier la commande

virtuelle à la commande réelle, i.e., $\mathbf{v} = \boldsymbol{\Upsilon}\mathbf{u}$. On a ainsi pour une configuration « + » :

$$\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}^{+} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & l & -l \\ -l & l & 0 & 0 \\ c & c & -c & -c \end{bmatrix}, \tag{3.29}$$

et pour une configuration « × » :

$$\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}^{\times} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{\sqrt{2}}{2}l & -\frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & -\frac{\sqrt{2}}{2}l \\ -\frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & -\frac{\sqrt{2}}{2}l \\ c & c & -c & -c \end{bmatrix}, \tag{3.30}$$

avec $c = k_D/k_T$ le rapport entre le coefficient de traînée induite et le coefficient de poussée. On peut vérifier que ces matrices d'allocation redonnent bien les équations (3.11), (3.24) et (3.25).

Identification des paramètres du drone

Pour terminer cette sous-section, nous aborderons les valeurs numériques des paramètres du drone. La plupart de celles-ci ont été obtenues expérimentalement et le tableau 3.1 rassemble les différentes valeurs des paramètres. Toutefois, quelques explications sont nécessaires pour décrire la façon d'obtenir certains d'entre eux. Pour la masse m, nous avons pesé le quadricoptère (avec sa batterie) et la longueur du bras l a été obtenue en la mesurant directement sur le quadricoptère. Pour déterminer les moments d'inertie selon $\mathbf{x}_b$, $\mathbf{y}_b$ et $\mathbf{z}_b$, nous avons utilisé le modèle CAD (*Computer-Aided Design*) du drone et le logiciel approprié SolidWorks.

Les coefficients de poussée et de traînée ont été obtenus en utilisant un banc d'essai dédié *RCBenchmark*[2], un module de support pour moteurs et hélices. Ce support permet, entre autres, de mesurer la poussée, le couple et la vitesse de rotation de l'hélice. Durant les tests, on recueille ces valeurs et l'on identifie les coefficients k_T et k_D grâce aux approximations quadratiques pour la force $T = k_T\omega^2$ et pour le moment $Q = k_D\omega^2$. On suppose ensuite que les quatre moteurs possèdent les mêmes caractéristiques. Les données et les courbes recueillies sont présentées à l'annexe A.3.

2. Site internet : https ://www.rcbenchmark.com/pages/series-1580-thrust-stand-dynamometer

Tableau 3.1 Paramètres dr l'Asctec Pelican

Paramètres	**Description**	**Valeur**
m	Masse	1.717 kg
l	Longueur du bras	0.215 m
(I_{xx}, I_{yy}, I_{zz})	Moments d'inertie	(0.0203,0.0209,0.0339) $\mathrm{kg} \cdot \mathrm{m}^2$
k_T	Coefficient de poussée	$1.435 \cdot 10^{-5} \mathrm{N} \cdot \mathrm{s}^2/\mathrm{rad}^2$
k_D	Coefficient de traînée	$2.664 \cdot 10^{-7} \mathrm{N} \cdot \mathrm{s}^2/\mathrm{rad}^2$

3.1.4 Modèle linéaire du drone

Pour développer les algorithmes de commande, il est souvent nécessaire de procéder à une linéarisation jacobienne du système étudié autour d'un point d'équilibre. Cette linéarisation permet de simplifier le système en conservant seulement les termes ayant le plus d'impact sur sa dynamique. Le processus de linéarisation étant une approximation de Taylor au premier ordre, on peut s'attendre à ce que l'approximation et le système non linéaire complet se comportent de façon similaire si le quadricoptère ne s'éloigne pas trop du point d'équilibre. Pour un multirotor, le point d'opération le plus souvent utilisé est le vol stationnaire. Ce point est particulièrement intéressant, car il permet de découpler le système en quatre sous-systèmes : la dynamique d'altitude en z, la dynamique de lacet en ψ, la dynamique longitudinale en x et θ et la dynamique latérale en y et ϕ.

À l'équilibre en vol stationnaire, nous avons $\dot{\mathbf{x}} = \mathbf{0}$, i.e., $\mathbf{f}(\mathbf{x}_e, \mathbf{v}_e) = \mathbf{0}$ avec $\mathbf{x}_e$ et $\mathbf{v}_e$ les valeurs à l'équilibre de $\mathbf{x}$ et $\mathbf{v}$. En vol stationnaire, le quadricoptère se maintient à un point arbitraire dans l'espace ; les vitesses de translation et de rotation sont nulles conduisant ainsi à $u_e = v_e = w_e = p_e = q_e = r_e = 0$. De plus le quadricoptère est à l'horizontal, soit $\phi_e = \theta_e = 0$. Les valeurs à l'équilibre de x_e, y_e, z_e et ψ_e sont arbitraires, car elles n'influencent pas la dynamique du système. Par simplicité, on impose donc $x_e = y_e = z_e = \psi_e = 0$. Enfin, la force totale de poussée vaut $F_z = mg$ pour compenser le poids du quadricoptère et les moments sont tous nuls, $M_x = M_y = M_z = 0$.

Le modèle d'état linéarisé est alors donné par

$$\begin{cases} \Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{v}, \\ \Delta\mathbf{y} = \mathbf{C}\Delta\mathbf{x} + \mathbf{D}\Delta\mathbf{v}, \end{cases} \tag{3.31}$$

où les vecteurs d'état $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_e$, d'entrée $\Delta\mathbf{v} = \mathbf{v} - \mathbf{v}_e$ et de sortie $\Delta\mathbf{y} = \mathbf{y} - \mathbf{y}_e$, représentent

les déviations des grandeurs par rapport à leurs valeurs d'équilibre, et sont définis comme :

$$\begin{aligned}
\Delta\mathbf{x} &= \begin{bmatrix}\Delta x & \Delta y & \Delta z & \Delta u & \Delta v & \Delta w & \Delta\phi & \Delta\theta & \Delta\psi & \Delta p & \Delta q & \Delta r\end{bmatrix}^\top, \\
\Delta\mathbf{v} &= \begin{bmatrix}\Delta F_z & \Delta M_x & \Delta M_y & \Delta M_z\end{bmatrix}^\top, \\
\Delta\mathbf{y} &= \begin{bmatrix}\Delta x & \Delta y & \Delta z & \Delta\psi\end{bmatrix}^\top.
\end{aligned} \tag{3.32}$$

Les matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ et $\mathbf{D}$ sont obtenues par linéarisation jacobienne, soit :

$$\begin{aligned}
\mathbf{A} &= \left.\frac{\partial\mathbf{f}(\mathbf{x},\mathbf{v})}{\partial\mathbf{x}}\right|_{\mathbf{x}_e,\mathbf{v}_e} \in \mathbb{R}^{12\times 12}, & \mathbf{B} &= \left.\frac{\partial\mathbf{f}(\mathbf{x},\mathbf{v})}{\partial\mathbf{x}}\right|_{\mathbf{x}_e,\mathbf{v}_e} \in \mathbb{R}^{12\times 4}, \\
\mathbf{C} &= \left.\frac{\partial\mathbf{h}(\mathbf{x},\mathbf{v})}{\partial\mathbf{v}}\right|_{\mathbf{x}_e,\mathbf{v}_e} \in \mathbb{R}^{4\times 12}, & \mathbf{D} &= \left.\frac{\partial\mathbf{h}(\mathbf{x},\mathbf{v})}{\partial\mathbf{x}}\right|_{\mathbf{x}_e,\mathbf{v}_e} \in \mathbb{R}^{4\times 4}.
\end{aligned} \tag{3.33}$$

Les matrices entièrement développées sont données dans l'annexe A.4. Il est cependant possible d'en extraire les douze équations d'état qui se présentent finalement sous des formes très simples :

$$\begin{aligned}
\Delta\dot{x} &= \Delta u, & \Delta\dot{u} &= g_0\Delta\theta, & \Delta\dot{\phi} &= \Delta p, & \Delta\dot{p} &= \frac{\Delta M_x}{I_{xx}}, \\
\Delta\dot{y} &= \Delta v, & \Delta\dot{v} &= -g_0\Delta\phi, & \Delta\dot{\theta} &= \Delta q, & \Delta\dot{q} &= \frac{\Delta M_y}{I_{yy}}, \\
\Delta\dot{z} &= \Delta w, & \Delta\dot{w} &= \frac{\Delta F_z}{m}, & \Delta\dot{\psi} &= \Delta r, & \Delta\dot{r} &= \frac{\Delta M_z}{I_{zz}}.
\end{aligned} \tag{3.34}$$

Comme annoncé précédemment, on peut alors construire quatre sous-systèmes découplés, correspondant chacun à une dynamique du quadricoptère :

— **Dynamique en altitude** :

$$\Delta\dot{\mathbf{x}}_z = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\Delta\mathbf{x}_z + \begin{bmatrix}0\\ m^{-1}\end{bmatrix}\Delta\mathbf{v}_z, \tag{3.35}$$

avec $\Delta\mathbf{x}_z = [\Delta z \;\; \Delta w]^\top$ et $\Delta\mathbf{v}_z = \Delta_z$.

— **Dynamique en lacet** :

$$\Delta\dot{\mathbf{x}}_\psi = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\Delta\mathbf{x}_\psi + \begin{bmatrix}0\\ I_{zz}^{-1}\end{bmatrix}\Delta\mathbf{v}_\psi, \tag{3.36}$$

avec $\Delta\mathbf{x}_\psi = [\Delta\psi \;\; \Delta r]^\top$ et $\Delta\mathbf{v}_\psi = \Delta M_z$.

— **Dynamique longitudinale** :

$$\Delta\dot{\mathbf{x}}_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g_0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Delta\mathbf{x}_x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ I_{yy}^{-1} \end{bmatrix} \Delta\mathbf{v}_x, \tag{3.37}$$

avec $\Delta\mathbf{x}_x = [\Delta x \;\; \Delta u \;\; \Delta\theta \;\; \Delta q]^\top$ et $\Delta\mathbf{v}_x = \Delta M_y$.

— **Dynamique latérale** :

$$\Delta\dot{\mathbf{x}}_y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g_0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Delta\mathbf{x}_y + \begin{bmatrix} 0 \\ 0 \\ 0 \\ I_{xx}^{-1} \end{bmatrix} \Delta\mathbf{v}_y, \tag{3.38}$$

avec $\Delta\mathbf{x}_y = [\Delta y \;\; \Delta v \;\; \Delta\phi \;\; \Delta p]^\top$ et $\Delta\mathbf{v}_y = \Delta M_x$.

Cette propriété du modèle linéarisé permet donc de diviser la synthèse des lois de commande en quatre parties distinctes. Le design en est ainsi facilité et permet d'obtenir de meilleurs résultats que si un seul contrôleur était synthétisé à partir du modèle complet de la dynamique linéarisée du quadricoptère. De plus, pour des vitesses faibles et de petits angles, ce modèle linéaire demeure valide.

3.2 Modélisation des incertitudes

Dans le but de concevoir une loi de commande pouvant compenser différents types d'incertitudes, nous abordons ici les représentations mathématiques possibles dans le cas d'un quadricoptère. Quatre types d'incertitudes et de perturbations peuvent affecter la dynamique du quadricoptère [18] :

Incertitudes paramétriques : l'ensemble des incertitudes liées à des variables constantes du modèle. Par exemple, parmi les équations (3.35) à (3.38), on notera que les paramètres m, I_{xx}, I_{yy} et I_{zz} pourraient être mal connus. Puisqu'ils sont constants par définition, alors toute incertitude liée à ces paramètres serait une incertitude paramétrique.

Incertitudes non paramétriques : l'ensemble des incertitudes ne provenant pas des paramètres. Ces incertitudes peuvent généralement être exprimées par une fonction de transfert. On notera, par exemple, le bruit de mesure, les délais temporels dans le système, les dynamiques négligées, etc.

Incertitudes assorties : une incertitude assortie est une sous-catégorie d'une incerti-

tude non paramétrique. Pour être considérée comme assortie, l'incertitude doit être présente dans le même canal que l'entrée. Pour un système linéaire, on exprime mathématiquement cette incertitude comme $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\left(\mathbf{u} + \mathbf{f}(\mathbf{x})\right)$, où $\mathbf{f}(\mathbf{x})$ représente ici l'incertitude assortie. Toute dynamique négligée qui affecterait la force T_z ou les moments M_x, M_y et M_z est une incertitude assortie. Par exemple, la force de traînée induite par le corps du quadricoptère et qui a été négligée dans la modélisation du quadricoptère est une incertitude assortie.

Incertitudes mal-assorties : une incertitude mal-assortie est l'autre sous-catégorie d'une incertitude non paramétrique. Cette fois, l'incertitude ne doit pas être présente dans le même canal que l'entrée. Pour un système linéaire, on représente ce genre d'incertitude comme $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \boldsymbol{\xi}(t)$, où $\boldsymbol{\xi}(t)$ est l'incertitude non paramétrique mal-assortie. Par exemple, des perturbations externes comme la présence de vent ou du bruit de mesure sont des perturbations mal-assorties.

Il est important de définir séparément les incertitudes non paramétriques assorties et mal-assorties, car il n'est pas possible de toutes les compenser de la même façon. En effet, les incertitudes assorties sont généralement plus faciles à compenser et plusieurs lois de commande y parviennent sans trop de difficulté. Cependant, les incertitudes mal-assorties sont plus difficiles à compenser et elles nécessitent souvent la synthèse de modifications robustes pour arriver à bien rejeter leur effet sur le système dynamique.

3.2.1 Incertitudes pour un système linéaire

On considère un système linéaire incertain tel qu'énoncé par [18], qui comprend toutes les formes d'incertitudes précédentes. Ce système est donné par

$$\dot{\mathbf{x}} = \mathbf{A}_u\mathbf{x} + \mathbf{B}\boldsymbol{\Lambda}\left(\mathbf{u} + \mathbf{f}(\mathbf{x})\right) + \boldsymbol{\xi}(t), \tag{3.39}$$

où $\mathbf{A}_u \in \mathbb{R}^{n\times n}$ est <u>inconnue</u>, $\mathbf{B} \in \mathbb{R}^{n\times m}$ est connue, $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ est une matrice choisie définie positive <u>inconnue</u>, $\mathbf{x} \in \mathbb{R}^n$ est le vecteur d'état et $\mathbf{u} \in \mathbb{R}^m$ est le vecteur d'entrée. La matrice inconnue $\mathbf{A}_u$ représente les incertitudes paramétriques présentes dans le modèle dynamique et la matrice $\boldsymbol{\Lambda}$ représente les incertitudes paramétriques pouvant affecter le vecteur d'entrées. En effet, cette matrice peut représenter des erreurs de modélisation (par exemple, de mauvaises valeurs pour m, I_{xx}, I_{yy} et I_{zz} dans le cas du quadricoptère) et elle peut représenter des pertes d'efficacité dans un ou plusieurs moteurs. On suppose de plus que la paire $(\mathbf{A}_u, \mathbf{B}\boldsymbol{\Lambda})$ est commandable.

Les incertitudes non paramétriques assorties $\mathbf{f}(\mathbf{x})$ représentent généralement des incertitudes

qui dépendent des états du système, car ils sont présents dans le même canal que l'entrée. L'expression de ces incertitudes peut être donnée sous la forme :

$$\mathbf{f}(\mathbf{x}) = \mathbf{\Theta}^\top \mathbf{\Phi}(\mathbf{x}), \tag{3.40}$$

où $\mathbf{\Theta}^\top \in \mathbb{R}^{m \times N}$ est une matrice <u>inconnue</u> de coefficients constants et $\mathbf{\Phi}(\mathbf{x}) \in \mathbb{R}^N$ est un vecteur d'ordre N où chacune des composantes est une somme de fonctions $\phi_i(\mathbf{x})$ qui sont localement Lipschitz continues. Cette définition permet d'englober toutes sortes de dynamiques négligées telles que l'effet de traînée induite par le corps du quadricoptère ainsi que l'effet de l'atmosphère sur la force de poussée. Il est important de noter que les incertitudes non paramétriques assorties sont présentes dans le même canal que l'entrée $\mathbf{u}$. En effet, ces incertitudes doivent pouvoir influencer les entrées et non pas l'entièreté du système. C'est ce qui différencie ces incertitudes par rapport aux incertitudes mal-assorties $\boldsymbol{\xi}(t)$. Par exemple, le bruit de mesure et celui de commande ou des perturbations externes, e.g., la présence de vent, sont des incertitudes non paramétriques mal-assorties.

3.3 Montage expérimental

Configuration du Asctec Pelican

La figure 3.2 donne le montage général pour le fonctionnement de l'AscTec Pelican. D'abord, le Pelican est équipé d'un ordinateur de bord, le *AscTec Mastermind* et d'une unité de commande de vol, i.e, *Flight Control Unit* (FCU), le *AscTec Autopilot*. L'ordinateur de bord utilise le système d'exploitation Ubuntu 14.04 avec la version ROS Jade. Il est équipé d'un double cœur Atom de 1.66 GHz avec 4 Go de mémoire vive. La présence de l'ordinateur de bord permet d'effectuer des calculs liés aux lois de commande à même le quadricoptère, i.e., en ligne. Pour obtenir des performances acceptables, la vitesse de fonctionnement est fixée à 200 Hz pour la boucle interne et à 100 Hz pour la boucle externe.

Deux processeurs ARM7 sont présents dans le FCU, un processeur de bas niveau, ou *Low-Level Processor* (LLP) et un processeur de haut niveau, ou *High-Level Processor* (HLP). De plus, le FCU contient une unité de mesure inertielle, ou *Inertial Measurement Unit* (IMU). L'ordinateur de bord peut communiquer avec les différentes composantes du FCU en utilisant les protocoles standards SPI, I2C, UART, etc. Les processeurs permettent de séparer les différents processus qui effectuent la commande du quadricoptère. Par exemple, le contrôle des moteurs et la fusion des différentes données des capteurs s'effectuent dans le LLP. Pour le développeur, il est seulement possible de concevoir des lois de commande utilisant l'interface

ROS [3] avec le HLP, qui communique ensuite avec le LLP via le protocole SPI. Autrement dit, toutes les fonctionnalités implémentées dans le LLP sont inaccessibles. Plusieurs méthodes de commande sont alors possibles ; celle choisie est la commande de bas niveau, en commandant directement la vitesse de rotation des moteurs. Dans l'interface du HLP, une vitesse de rotation est représentée par un entier de 1 à 200. La valeur choisie est ensuite envoyée au LLP qui la traduit en une vitesse de rotation en rad/s grâce à son contrôleur de moteur.

En utilisant un entier pour représenter la vitesse de rotation des moteurs, une erreur de discrétisation est introduite dans le système. En effet, il serait préférable d'avoir une échelle continue. Or, le moteur tourne de 0 à 8000 tours par minute. On obtient donc une discrétisation de 40 tours par minute. Les conséquences de cette discrétisation seront discutées en détail à la section 5.2.

Configuration du laboratoire

Le laboratoire de robotique mobile et des systèmes autonomes, ou encore en anglais, *Mobile Robotics and Autonomous Systems Laboratory* (MRASL) de Polytechnique Montréal fournit un espace de vol qui est équipé d'un système de positionnement VICON. Ce système est composé de 12 caméras infrarouges qui permettent de localiser précisément des éléments réflecteurs installés sur le quadricoptère. Ces informations sont combinées dans un module de fusion multi-capteurs, *MultiSensor Fusion Framework* (MSF) [4], dans le but d'obtenir des données plus précises et moins bruitées sur les états de l'appareil. Le quadricoptère est connecté au serveur du VICON (présent sur un ordinateur externe) via un réseau périphérique de réalité virtuelle, *Virtual Reality Peripheral Network* (VRPN) [5]. Ce serveur fonctionne à une vitesse de 200 Hz ce qui est conforme à la vitesse de fonctionnement des différents processus de l'ordinateur de bord.

Un second ordinateur externe est utilisé pour les tests en laboratoire. Cette station de base permet d'initialiser tous les paramètres nécessaires au vol du drone. De plus, c'est à partir de cet ordinateur qu'on surveille, à distance, les différents processus exécutés sur le drone. Ce poste de travail est connecté au quadricoptère via une connexion Wi-Fi. Enfin, toutes les commandes de haut niveau (décollage, atterrissage, déplacement, perte d'efficacité des moteurs, etc.) sont envoyées à partir de cette station de base.

3. Référence Wiki : `http://wiki.ros.org/asctec_mav_framework`
4. Référence Wiki : `https://github.com/ethz-asl/ethzasl_msf/wiki`
5. Dépôt Github : `https://github.com/ethz-asl/ros_vrpn_client`

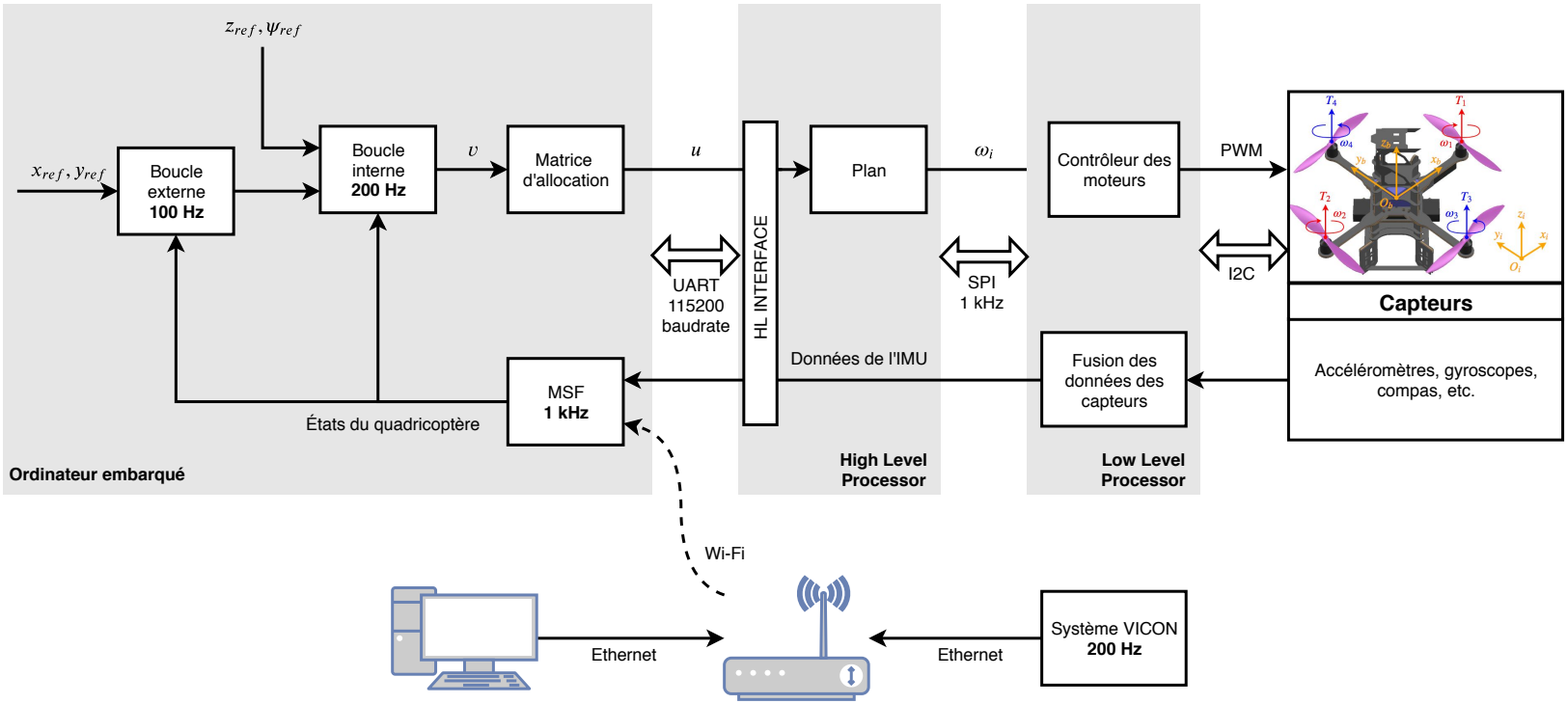


Figure 3.2 Architecture du montage expérimental pour l'AscTec Pelican

3.3.1 Processus de synthèse des lois de commande

L'environnement pour le développement expérimental des algorithmes de commande est similaire à celui utilisé dans [47]. Le développement d'un contrôleur passe par trois phases. D'abord, le contrôleur est conçu et simulé dans MATLAB/SIMULINK. Le modèle utilisé pour la simulation est simple et la plupart des incertitudes sont négligées. De plus, plusieurs effets sont ignorés : force de traînée, présence de vent, effets gyroscopiques, etc. Une fois que le contrôleur donne de bons résultats, le code source en C++ est généré à partir du modèle SIMULINK grâce à l'outil SIMULINK CODER. Le quadricoptère est ensuite simulé dans ROTORS [6] développé par [48] pour ROS Gazebo. L'environnement ROTORS permet de tenir en compte plusieurs éléments présents sur le quadricoptère comme les gyroscopes, les accéléromètres, et les moteurs, et plusieurs effets aérodynamiques. De plus, il vient avec un modèle complet pour l'AscTec Pelican. Ainsi, ROTORS est un environnement de simulation plus complet que MATLAB/SIMULINK. Si les simulations sont acceptables, le contrôleur est prêt à être déployé sur le quadricoptère. Sinon, il faut une itération supplémentaire MATLAB–ROS Gazebo. L'avantage d'utiliser ce processus itératif est que l'environnement simplifié MATLAB/SIMULINK permet de développer rapidement une loi de commande permettant de stabiliser le quadricoptère. Ensuite, on utilise ROTORS pour détecter des problèmes plus subtils avec la loi de commande en considérant toutes les composantes du drone. Avec ce processus, on peut accélérer la synthèse et le réglage d'un contrôleur performant pour le quadricoptère.

3.3.2 Changement de repère

Lors de l'achat d'un quadricoptère, le matériel disponible sur celui-ci, notamment l'IMU, est généralement préconfiguré selon la configuration « $\times$ » ou « $+$ ». Or, si la commande en forces et moments du quadricoptère peut s'effectuer indépendamment de la configuration de l'appareil, celle-ci a un impact sur sa stabilité et sa manœuvrabilité en vol. En effet, d'après [49], la configuration « $\times$ » est plus stable, car les quatre moteurs sont utilisés pour générer les moments M_x et M_y . Si le quadricoptère est utilisé dans des situations incertaines où la stabilité de l'appareil est importante, cette configuration est donc à privilégier. À l'opposé, la configuration « $+$ » est plus réactive et elle est généralement utilisée pour permettre des manoeuvres plus acrobatiques.

Le quadricoptère Pelican est configuré en « $+$ ». Cependant, le but de la commande adaptative étant d'augmenter la stabilité de l'engin en présence d'incertitudes, nous allons donc

6. Dépôt Github : `https://github.com/ethz-asl/rotors_simulator`

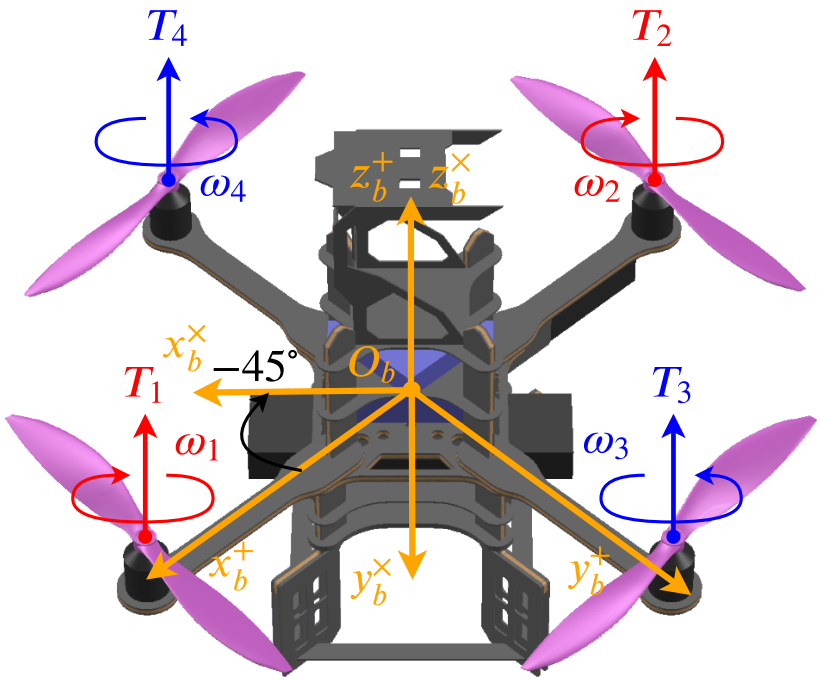


Figure 3.3 Modèle du drone et représentation des repères

privilégier l'implémentation des lois de commande pour une configuration « × ». Comme nous ne pouvons modifier physiquement le positionnement des capteurs, il faut les convertir vers la configuration « × ». Pour les données représentant un vecteur physique, i.e., des données qui peuvent former une base orthonormée, une rotation de $-45°$ autour de l'axe $\mathbf{z}_b$ permet de les convertir adéquatement. La figure 3.3 illustre la rotation effectuée pour placer les dans la configuration « × ». Concrètement, on obtiendra les vecteurs suivants :

$$\begin{aligned}
\mathbf{p}_{cm/i}^{i,\times} &= \mathbf{R}_{\mathbf{z}_b}(-45°)\,\mathbf{p}_{cm/i}^{i,+}, \\
\mathbf{v}_{cm/i}^{b,\times} &= \mathbf{R}_{\mathbf{z}_b}(-45°)\,\mathbf{v}_{cm/i}^{b,+}, \\
\boldsymbol{\omega}_{b/i}^{b,\times} &= \mathbf{R}_{\mathbf{z}_b}(-45°)\,\boldsymbol{\omega}_{b/i}^{b,+}.
\end{aligned} \tag{3.41}$$

Cependant, les angles d'Euler ne représentent pas un vecteur physique. Pour obtenir les angles adéquats dans l'orientation « × », il faut déterminer la matrice de transition complète entre le repère inertiel et le repère objet orienté en « × ». Les équipements installés sur le quadricoptère permettent de récupérer la matrice $\mathbf{R}_{i/b}^{+}$, où le + mis en exposant est associé à la configuration « + » de l'appareil. Il est alors possible de modifier la matrice obtenue pour

la projeter dans le repère adéquat, pour la configuration désirée du drone :

$$\mathbf{R}_{i/b}^{\times} = \mathbf{R}_{i/b}\mathbf{R}_{\mathbf{z}_b}(45^\circ)\,. \tag{3.42}$$

Avec cette matrice, on peut utiliser les formules habituelles pour déterminer les angles d'Euler, soient

$$\begin{aligned}
\phi &= \operatorname{atan2}\left(\mathbf{R}_{i/b}^{b,\times}(3,2),\ \mathbf{R}_{b/i}^{b,\times}(3,3)\right),\\
\theta &= \operatorname{asin}\left(-\mathbf{R}_{b/i}^{b,\times}(3,1)\right),\\
\psi &= \operatorname{atan2}\left(\mathbf{R}_{i/b}^{b,\times}(2,1),\ \mathbf{R}_{b/i}^{b,\times}(1,1)\right).
\end{aligned} \tag{3.43}$$

Enfin, une dernière modification est nécessaire pour s'assurer que le changement de configuration est bien implémenté. Puisque les lois de commandes sont synthétisées en fonction de la commande virtuelle $\mathbf{v}$, elles sont indépendantes de l'orientation du drone. Cela est vrai, tant que les états reçus par le contrôleur sont conséquents avec la matrice d'allocation $\boldsymbol{\Upsilon}$, car cette matrice envoie les commandes aux moteurs selon la configuration du drone. À cet effet, on utilisera la matrice d'allocation $\boldsymbol{\Upsilon}^{\times}$ définie dans l'équation 3.30.

3.4 Conclusion du chapitre

Dans ce chapitre, trois aspects fondamentaux concernant la modélisation du quadricoptère ont été abordés. D'abord, les équations non linéaires complètes ont été développées. Ces équations ont ensuite été linéarisées autour du point d'équilibre du quadricoptère en vol stationnaire. Ensuite, nous avons considéré les différentes incertitudes et perturbations qui peuvent affecter le quadricoptère en vol. Cela nous a permis d'obtenir un modèle linéaire incertain complet du quadricoptère. C'est à partir de ce modèle qu'il sera possible de synthétiser les lois de commande.

Enfin, nous avons décrit le montage expérimental utilisé au MRASL, ainsi que le processus utilisé pour synthétiser et ajuster les lois de commande. Nous avons également abordé des éléments techniques qui sont particuliers à l'AscTec Pelican. Ces différents éléments ont mené à des modifications qui sont nécessaires pour assurer une commande adéquate du quadricoptère dans l'enceinte de vol du laboratoire.

CHAPITRE 4 SYNTHÈSE DES LOIS DE COMMANDE CLASSIQUES ET ADAPTATIVES

L'objectif de ce chapitre est de synthétiser une loi de commande adaptative avec modèle de référence qui a pour but d'assister une commande classique pour compenser des incertitudes paramétriques et non paramétriques assorties. La commande de base choisie est la commande LQ, car selon les travaux de [45], ce type de commande est particulièrement performant et robuste ; à cet effet, le LQR est synthétisé à la section 4.1 sur le modèle nominal. Dans la section 4.2, on se rapporte ensuite aux travaux expérimentaux de [30] et aux concepts théoriques formulés par [21, 23] pour développer une MRAC. On complète cette section avec une preuve de stabilité.

Ces contrôleurs sont ensuite implémentés sur le simulateur du AscTec Pelican. Un processus itératif d'ajustement des paramètres est utilisé pour obtenir les contrôleurs les plus performants possible. La technique de réglage est présentée à la section 4.3. À la suite du réglage des gains, le quadricoptère est simulé pour valider le bon fonctionnement de la MRAC. Ce conclusions sont présentées à la section 4.4.

Ensuite, les algorithmes de commande sont déployés sur le système physique. La section 4.5 donne un compte-rendu des différentes expériences effectuées en laboratoire. Enfin, les limites de cette méthode par rapport aux incertitudes non paramétriques mal-assorties sont abordées à la section 4.6.

4.1 Synthèse d'un contrôleur nominal par la commande LQ

On synthétise d'abord une loi de commande classique sur le modèle d'état nominal (3.31). Nous avons vu que la dynamique linéarisée du quadricoptère peut être découplée en quatre sous-systèmes. On peut aussi pousser notre analyse en distinguant deux échelles de temps sur notre système. En effet, les équations (3.34) montrent que certains états sont obtenus après deux intégrations, tandis que d'autres sont obtenus après quatre intégrations, e.g.,

$$\begin{aligned} &\Delta M_x \xrightarrow{\int} \Delta p \xrightarrow{\int} \Delta\phi \xrightarrow{\int} \Delta\dot{y} \xrightarrow{\int} \Delta y, \\ &\Delta F_z \xrightarrow{\int} \Delta\dot{z} \xrightarrow{\int} \Delta z. \end{aligned} \tag{4.1}$$

L'attitude donnée par $\Delta\phi$, $\Delta\theta$ et $\Delta\psi$ et l'altitude donnée par Δz sont régies par une dynamique de double intégrateur tandis que la position donnée par Δx et Δy est régie par

une dynamique de quadruple intégrateur. L'architecture de commande nominale donnée à la figure 4.1 reflète ces deux échelles de temps de fonctionnement en séparant le contrôleur en deux boucles. La boucle interne commande les états rapides associés à l'attitude et à l'altitude, et fonctionne à un taux de rafraîchissement de 200 Hz, comme indiqué lors de la présentation du matériel expérimental (Fig. 3.2) La boucle externe commande les états lents associé à la position fonctionne à un taux de rafraîchissement de 100 Hz.

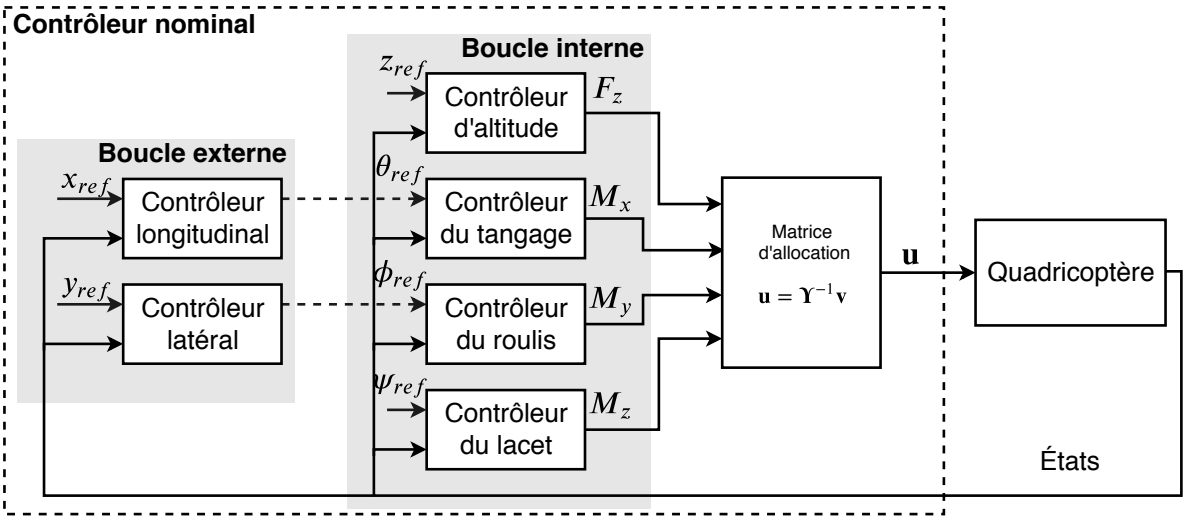


Figure 4.1 Architecture du contrôleur de base

La loi de commande nominale prend la forme d'une commande par retour d'état donnée par $\Delta\mathbf{v} = -\mathbf{K}\Delta\mathbf{x}$. Avant de déterminer la méthode de résolution pour l'obtention de la matrice de gains $\mathbf{K}$, le système est d'abord augmenté avec des intégrateurs portant sur le vecteur des erreurs de suivi $\Delta\mathbf{e}$ défini par :

$$\Delta\mathbf{e} = \Delta\mathbf{r} - \Delta\mathbf{y} = \Delta\mathbf{r} - \mathbf{C}\Delta\mathbf{x}, \tag{4.2}$$

où $\Delta\mathbf{r} = [\Delta x_{ref}\ \Delta y_{ref}\ \Delta z_{ref}\ \Delta\psi_{ref}]^\top$ constitue le vecteur des entrées de références et $\Delta\mathbf{y} = [\Delta x\ \Delta y\ \Delta z\ \Delta\psi]^\top$ les sorties à asservir. En notant $\Delta\mathbf{x}_i = [\Delta x_{i,x}\ \Delta x_{i,y}\ \Delta x_{i,z}\ \Delta x_{i,\psi}]^\top$ le vecteur d'état des intégrateurs sur les erreurs, on a la dynamique suivante :

$$\Delta\dot{\mathbf{x}}_i = \Delta\mathbf{e}. \tag{4.3}$$

Le modèle d'état augmenté est alors donné par :

$$\Delta\dot{\mathbf{x}}_a = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0}_{12\times 4} \\ -\mathbf{C} & \mathbf{0}_4 \end{bmatrix}}_{\mathbf{A}_a} \Delta\mathbf{x}_a + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0}_4 \end{bmatrix}}_{\mathbf{B}_a} \Delta\mathbf{v} + \underbrace{\begin{bmatrix} \mathbf{0}_{12\times 4} \\ \mathbf{I}_4 \end{bmatrix}}_{\mathbf{B}_c} \Delta\mathbf{r},$$
$$\Delta\mathbf{y} = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{0}_4 \end{bmatrix}}_{\mathbf{C}_a} \Delta\mathbf{x}_a,$$

où $\Delta\mathbf{x}_a = [\Delta\mathbf{x}^\top \quad \Delta\mathbf{x}_i^\top]^\top$ est le nouveau vecteur d'état augmenté. La commande issue du contrôleur nominal est alors donnée par :

$$\Delta\mathbf{v}_{bl} = -\mathbf{K}_a\Delta\mathbf{x}_a = -\begin{bmatrix} \mathbf{K}_x & -\mathbf{K}_{x_i} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{x}_i \end{bmatrix}, \tag{4.4}$$

où l'indice *bl* signifie *baseline*, ce qui permettra de distinguer ce signal de commande de celui généré par la commande adaptative. Enfin, on exprime le système en boucle fermée sous la forme :

$$\Delta\dot{\mathbf{x}}_a = (\mathbf{A}_a - \mathbf{B}_a\mathbf{K}_a)\,\Delta\mathbf{x}_a + \mathbf{B}_c\Delta\mathbf{r}. \tag{4.5}$$

La matrice de gains $\mathbf{K}_a$ est synthétisée grâce à la commande optimale LQ. Ce problème consiste à minimiser la fonction de coût $\mathcal{J}$ donnée par :

$$\mathcal{J} = \frac{1}{2}\int_0^{t_{\max}} \left\{ \begin{bmatrix} \Delta\mathbf{x}^\top & \Delta\mathbf{x}_i^\top \end{bmatrix} \mathbf{Q} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{x}_i \end{bmatrix} + \Delta\mathbf{v}^\top\mathbf{R}\Delta\mathbf{v} \right\} \mathrm{d}t. \tag{4.6}$$

Les matrices de pondération $\mathbf{Q}$ et $\mathbf{R}$ sont respectivement semi-définie positive et définie positive, et généralement choisies diagonales par le concepteur, i.e., $\mathbf{Q} = \mathrm{diag}(q_1, \ldots, q_n)$ et $\mathbf{R} = \mathrm{diag}(r_1, \ldots, r_m)$. Elles déterminent les performances que l'on souhaite obtenir. La matrice $\mathbf{Q}$ détermine l'influence des états sur $\mathcal{J}$ et la matrice $\mathbf{R}$ détermine l'influence des entrées. Plus la valeur de q_i est élevée, plus on souhaite que l'état correspondant tende vers 0 rapidement. Plus la valeur de r_i est élevée, moins on souhaite solliciter l'entrée correspondante. Le problème de la commande LQ a été résolu en 1960 par [50]. Pour un horizon infini ($t_{\max} \to \infty$), la solution est donnée par

$$\mathbf{K}_a = \mathbf{R}^{-1}\mathbf{B}_a^\top\mathbf{P}_a, \tag{4.7}$$

où $\mathbf{P}_a$ est la solution symétrique semi-définie positive de l'équation de Riccati algébrique à

temps continu :

$$\mathbf{P}_a\mathbf{A}_a + \mathbf{A}_a^\top\mathbf{P}_a - \mathbf{P}_a\mathbf{B}_a\mathbf{R}^{-1}\mathbf{B}_a^\top\mathbf{P}_a + \mathbf{Q} = \mathbf{0}. \tag{4.8}$$

4.1.1 Considérations pratiques

À la section 3.1.4, nous avons linéarisé les équations dynamiques en supposant que l'angle de lacet ψ était nul. Or, en pratique, l'angle de lacet couvre généralement 360° selon la direction prise par le drone. Si l'angle de lacet n'est pas nul, il faut effectuer une correction entre les contrôleurs de la boucle externe en x et y qui agissent dans le référentiel inertiel et les contrôleurs de la boucle interne qui agissent dans le référentiel objet. Si l'on définit u_x et u_y comme les commandes sortant des contrôleurs de la boucle externe en x et en y, alors, on procède à la correction suivante :

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix}^b = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}^i. \tag{4.9}$$

Enfin, on rappelle que la linéarisation s'effectue autour d'un point d'équilibre. Ainsi, à la commande fournie par le contrôleur nominal, il faut ajouter la commande à l'équilibre :

$$\mathbf{u}_{bl} = \boldsymbol{\Upsilon}^\times \left(\Delta\mathbf{v}_{bl} + \mathbf{v}_e\right), \tag{4.10}$$

où $\mathbf{v}_e = [mg \;\; 0 \;\; 0 \;\; 0]^\top$.

Le découplage naturel du système en quatre sous-systèmes (Eqs. (3.35), (3.36), (3.37) et (3.38)) donne une matrice de gains $\mathbf{K_a}$ ayant la structure particulière suivante :

$$\mathbf{K}_a = \begin{bmatrix} 0 & 0 & K_z & 0 & 0 & K_w & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -K_{i,z} & 0 \\ 0 & K_y & 0 & 0 & K_v & 0 & K_\phi & 0 & 0 & K_p & 0 & 0 & 0 & -K_{i,y} & 0 & 0 \\ K_x & 0 & 0 & K_u & 0 & 0 & 0 & K_\theta & 0 & 0 & K_q & 0 & -K_{i,x} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_\psi & 0 & 0 & K_r & 0 & 0 & 0 & -K_{i,\psi} \end{bmatrix} \tag{4.11}$$

Les figures 4.2 à 4.6 représentent alors les schémas blocs des différents contrôleurs obtenus grâce à la commande nominale LQ.

4.2 Synthèse de la MRAC

4.2.1 Formulation de la commande adaptative

Le LQR synthétisé précédemment est conçu pour un système qui ne présente aucune incertitude ou perturbation. Si ce n'est pas le cas, le contrôleur nominal ne fonctionnera pas

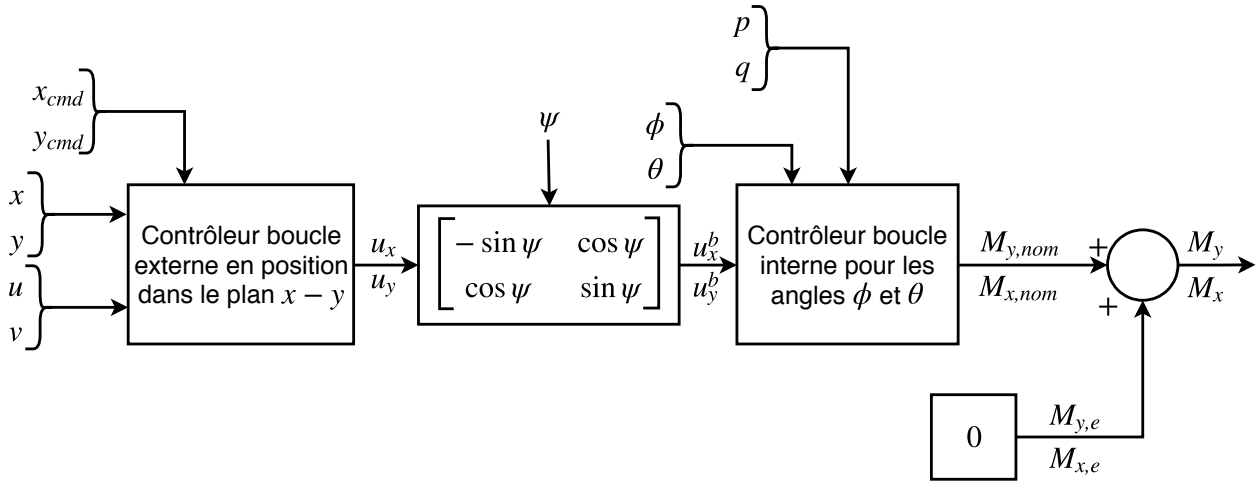


Figure 4.2 Schéma bloc complet du contrôleur nominal en position

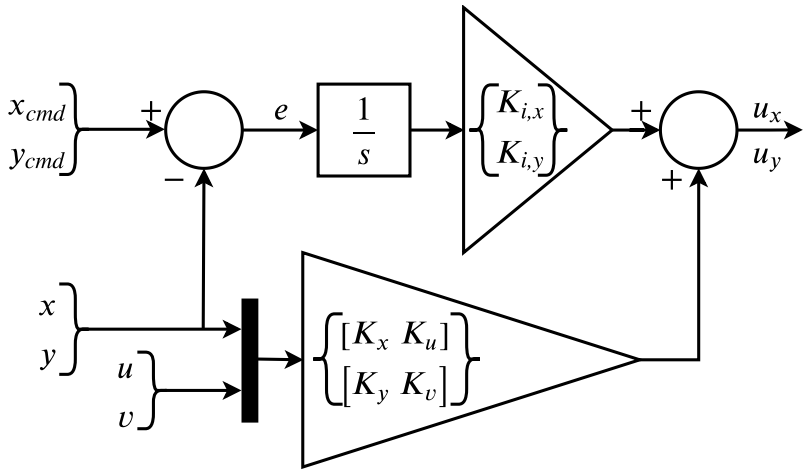


Figure 4.3 Schéma bloc du contrôleur nominal en position (boucle externe)

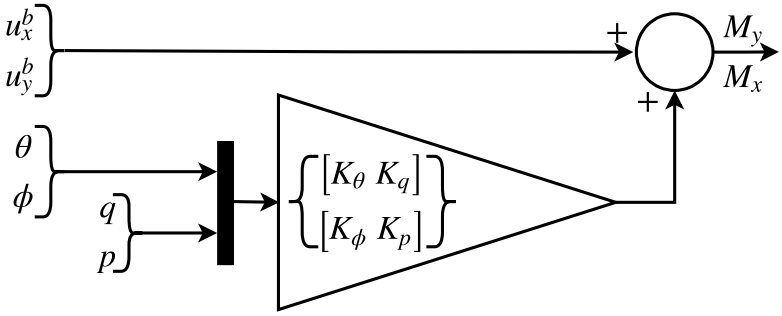


Figure 4.4 Schéma bloc du contrôleur nominal en roulis et en tangage (boucle interne)

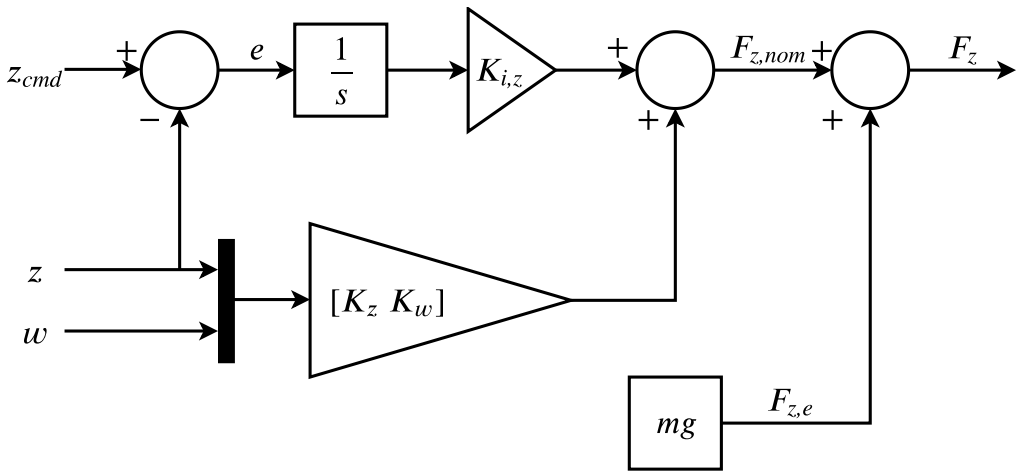


Figure 4.5 Schéma bloc du contrôleur nominal en altitude

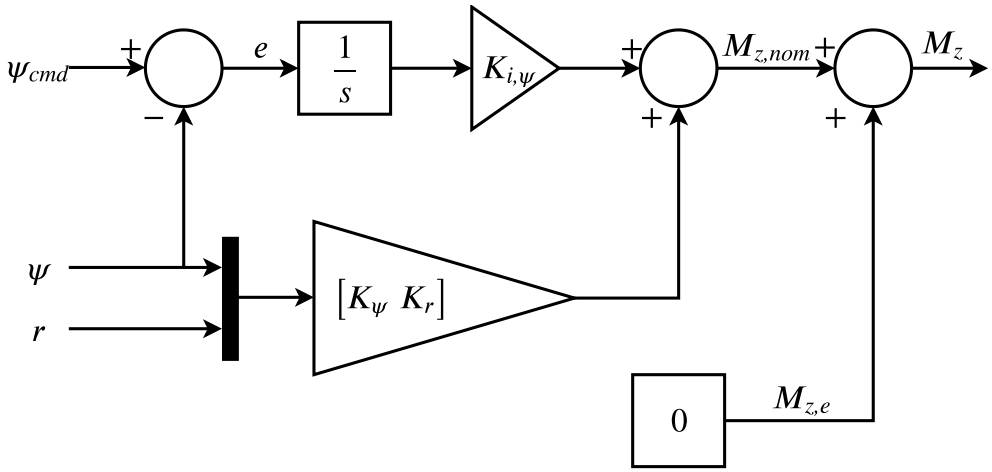


Figure 4.6 Schéma bloc du contrôleur nominal en lacet

de façon optimale et les performances vont se dégrader. Éventuellement, cela peut mener à l'instabilité, voire même à l'écrasement de l'appareil.

L'objectif de la commande adaptative est d'assurer la stabilité du système et de garantir des performances temporelles optimales de l'appareil en vol même s'il est en présence d'incertitudes et de perturbations. La MRAC garantit aussi que le système physique peut suivre un modèle de référence idéal donné par :

$$\Delta\dot{\mathbf{x}}_{ref} = \mathbf{A}_{ref}\Delta\mathbf{x}_{ref} + \mathbf{B}_{ref}\Delta\mathbf{r}. \tag{4.12}$$

Deux formes sont possibles pour la synthèse de la MRAC. Dans la forme pure, la MRAC

est le seul algorithme de commande présent sur le quadricoptère, alors que pour la forme augmentée, la MRAC vient assister un contrôleur nominal dans l'effort de commande. À la suite de la revue de littérature, nous avons conclu que la MRAC augmentée est la meilleure en pratique. Cela s'explique par le fait que cette approche est plus intuitive ; elle permet en effet d'obtenir un modèle de référence directement à partir du contrôleur nominal. En effet, si l'on considère le système linéaire sans perturbation avec le contrôleur nominal en boucle fermée, on aura :

$$\Delta\dot{\mathbf{x}}_{a,ref} = \underbrace{(\mathbf{A}_a - \mathbf{B}_a\mathbf{K}_a)}_{\mathbf{A}_{ref}}\Delta\mathbf{x}_{a,ref} + \underbrace{\mathbf{B}_c}_{\mathbf{B}_{ref}}\Delta\mathbf{r}. \tag{4.13}$$

Utiliser ce système en boucle fermée comme modèle de référence plutôt qu'un modèle quelconque apporte deux principaux avantages lors de la conception de la MRAC :

1. La conception de la MRAC est simplifiée, car le concepteur n'a pas à utiliser un modèle tiers lors de la synthèse de la MRAC.
2. Le réglage de la MRAC est plus efficace. En effet, puisque la MRAC est dépendante de la qualité du contrôleur nominal. S'il est bien réglé, alors la MRAC sera plus performante.

Avec ce modèle de référence, nous pouvons maintenant passer à la synthèse de la commande adaptative. Cette synthèse se fait en trois étapes. D'abord nous posons des conditions de correspondance entre le modèle de référence et le système incertain. Puis, nous déterminons la forme que doit avoir la commande adaptative pour que le système incertain soit semblable au modèle de référence. Enfin, nous fixons la dynamique des gains adaptatifs pour assurer la convergence du système incertain vers le modèle de référence.

Hypothèse 4.2.1. *Soit une matrice de référence* $\mathbf{A}_{ref} \in \mathbb{R}^{n\times n}$ *et les matrices inconnues* $\mathbf{A}_u \in \mathbb{R}^{n\times n}$ *et* $\mathbf{\Lambda} \in \mathbb{R}^{m\times m}$*. On suppose qu'il existe des matrices de gains* $\mathbf{K}_u \in \mathbb{R}^{m\times n}$ *et* $\mathbf{K}_r \in \mathbb{R}^{m\times m}$*, qui peuvent être inconnues, telles que*

$$\begin{aligned} \mathbf{A}_{ref} &= \mathbf{A}_u + \mathbf{B}\mathbf{\Lambda}\mathbf{K}_u, \\ \mathbf{B}_{ref} &= \mathbf{B}\mathbf{\Lambda}\mathbf{K}_r. \end{aligned} \tag{4.14}$$

On considère le système incertain donné à l'équation (3.39) appliqué au cas du système augmenté de l'équation (4.5). De plus, on considère qu'il n'y a pas d'incertitudes non paramétriques mal-assorties, donc $\boldsymbol{\xi} = \mathbf{0}$. En utilisant la première condition de correspondance,

on obtient

$$\begin{aligned}\Delta\dot{\mathbf{x}}_a &= \mathbf{A}_u\Delta\mathbf{x}_a + \mathbf{B}\boldsymbol{\Lambda}\left(\Delta\mathbf{v} - \mathbf{f}\right),\\ &= \left(\mathbf{A}_{ref} - \mathbf{B}\boldsymbol{\Lambda}\mathbf{K}_u\right)\Delta\mathbf{x}_a + \mathbf{B}\boldsymbol{\Lambda}\left(\Delta\mathbf{v} - \mathbf{f}\right),\\ &= \mathbf{A}_{ref}\Delta\mathbf{x}_a + \mathbf{B}\boldsymbol{\Lambda}\left(\Delta\mathbf{v} - \mathbf{K}_u\Delta\mathbf{x}_a - \mathbf{f}\right).\end{aligned} \tag{4.15}$$

Pour obtenir le modèle de référence souhaité, $\mathbf{B}_{ref}\Delta r$ est additionné et soustrait du résultat précédent. Puis, en utilisant la seconde condition de correspondance, on a

$$\begin{aligned}\Delta\dot{\mathbf{x}}_a &= \mathbf{A}_{ref}\Delta\mathbf{x}_a + \mathbf{B}\boldsymbol{\Lambda}\left(\Delta\mathbf{v} - \mathbf{K}_u\Delta\mathbf{x}_a - \mathbf{f}\right) + \mathbf{B}_{ref}\Delta\mathbf{r} - \underbrace{\mathbf{B}_{ref}\Delta\mathbf{r}}_{\mathbf{B}\boldsymbol{\Lambda}\mathbf{K}_r\Delta\mathbf{r}},\\ &= \mathbf{A}_{ref}\Delta\mathbf{x}_a + \mathbf{B}_{ref}\Delta\mathbf{r} + \mathbf{B}\boldsymbol{\Lambda}\left(\Delta\mathbf{v} - \mathbf{K}_u\Delta\mathbf{x}_a - \mathbf{K}_r\Delta\mathbf{r} - \mathbf{f}\right).\end{aligned} \tag{4.16}$$

Grâce à ce résultat, nous pouvons formuler la structure que doit avoir la commande adaptative pour obtenir exactement le modèle de référence :

$$\Delta\mathbf{v}_{ad} = \hat{\mathbf{K}}_u\Delta\mathbf{x}_a + \hat{\mathbf{K}}_r\Delta\mathbf{r} + \hat{\mathbf{f}} = \hat{\boldsymbol{\theta}}^\top\boldsymbol{\omega}, \tag{4.17}$$

où l'indice *ad*, signifie *adaptatif*, $\hat{\boldsymbol{\theta}}^\top = [\hat{\mathbf{K}}_u \;\; \hat{\mathbf{K}}_r \;\; \hat{\mathbf{f}}] \in \mathbb{R}^{m\times(n+m+1)}$ est la matrice des gains adaptatifs et $\boldsymbol{\omega} = [\Delta\mathbf{x}_a^\top \;\; \Delta\mathbf{r}^\top \;\; 1]^\top$ est le vecteur des régressions. Si l'on remplace l'expression de la commande adaptative dans le système en boucle fermée, on obtient :

$$\Delta\dot{\mathbf{x}}_a = \mathbf{A}_{ref}\Delta\mathbf{x}_a + \mathbf{B}_{ref}\Delta\mathbf{r} + \mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\omega}, \tag{4.18}$$

où $\Delta\hat{\boldsymbol{\theta}}^\top = [\hat{\mathbf{K}}_u - \mathbf{K}_u \;\; \hat{\mathbf{K}}_r - \mathbf{K}_r \;\; \hat{\mathbf{f}} - \mathbf{f}]$ est l'erreur d'estimation des paramètres adaptatifs. Si l'estimation est parfaite, i.e., $\Delta\hat{\boldsymbol{\theta}}^\top = \mathbf{0}$, alors on retrouvera le système en boucle fermée idéal de l'équation (4.12). Nous devons déterminer à présent la dynamique des gains adaptatifs pour nous assurer de la stabilité de la loi de commande adaptative. Cette dynamique devra aussi garantir que l'hypothèse 4.2.1 est vraie.

4.2.2 Analyse de la stabilité

Pour déterminer la dynamique des gains adaptatifs, il faut s'assurer de la stabilité du contrôleur. À cet effet, on s'intéresse à l'erreur entre le modèle de référence et le système physique

qui est définie comme $\Delta\boldsymbol{\varepsilon} = \Delta\mathbf{x}_a - \Delta\mathbf{x}_{a,ref}$. La dynamique de cette erreur est donnée par :

$$\begin{aligned}
\Delta\dot{\boldsymbol{\varepsilon}} &= \Delta\dot{\mathbf{x}}_a - \Delta\dot{\mathbf{x}}_{a,ref}, \\
&= \left(\mathbf{A}_{ref}\Delta\mathbf{x}_a + \mathbf{B}_{ref}\Delta\mathbf{r} + \mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\omega}\right) - \left(\mathbf{A}_{ref}\Delta\mathbf{x}_{a,ref} + \mathbf{B}_{ref}\Delta\mathbf{r}\right), \\
&= \mathbf{A}_{ref}\left(\Delta\mathbf{x}_a - \Delta\mathbf{x}_{a,ref}\right) + \mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\omega}, \\
&= \mathbf{A}_{ref}\Delta\boldsymbol{\varepsilon} + \mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\omega}.
\end{aligned} \tag{4.19}$$

La convergence de cette dynamique vers l'origine garantit la stabilité du contrôleur. Nous utilisons les notions de stabilité de Lyapunov pour l'analyser. Le lecteur est invité à se référer à [19, Chapitre 4] pour de plus amples détails concernant ces notions. Toutefois, nous rappelons le lemme et le corollaire suivants, qui sont essentiels pour la suite de cette démonstration.

Lemme 4.2.1 (Barbalat)**.** *Soit $f : \mathbb{R} \to \mathbb{R}$, une fonction uniforme continûment différentiable sur $[0, \infty)$. On suppose que $\lim_{t\to\infty}\int_0^t f(\tau)d\tau$ existe et est finie. Alors $\lim_{t\to\infty} f(t) = 0$.*

Corollaire 4.2.1. *Si une fonction scalaire $f : \mathbb{R} \to \mathbb{R}$ est deux fois continûment différentiable sur $[0, \infty)$ et a une limite finie, i.e., $\lim_{t\to\infty} f(t) < \infty$ et que sa seconde dérivée est bornée, alors $\lim_{t\to\infty} \dot{f}(t) = 0$.*

On propose la fonction candidate de Lyapunov suivante :

$$V = \Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\Delta\boldsymbol{\varepsilon} + \mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\Gamma}_\theta^{-1}\Delta\hat{\boldsymbol{\theta}}\boldsymbol{\Lambda}\right), \tag{4.20}$$

où $\boldsymbol{\Gamma}_\theta \in \mathbb{R}^{(m+n+1)\times(m+n+1)}$ est une matrice définie positive dont nous donnerons la forme ultérieurement et $\mathbf{P}_l \in \mathbb{R}^{n\times n}$ est la solution de l'équation de Lyapunov :

$$\mathbf{P}_l\mathbf{A}_{ref} + \mathbf{A}_{ref}^\top\mathbf{P}_l = -\mathbf{Q}_l. \tag{4.21}$$

où les matrices $\mathbf{P}_l$ et $\mathbf{Q}_l$ sont définies positives. On rappelle que $\boldsymbol{\Lambda}$ représente des incertitudes paramétriques au niveau des entrées. Elle est définie positive et diagonale, i.e., $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$, et chaque élément $\lambda_i \in [0, 1]$. Ces conditions sur $\boldsymbol{\Gamma}_\theta$, $\boldsymbol{\Lambda}$, $\mathbf{P}_l$ et $\mathbf{Q}_l$ font en sorte que V est positive, une condition nécessaire pour une fonction candidate de Lyapunov. La dérivée temporelle de la fonction candidate $\dot{V}$ est donnée par :

$$\dot{V} = -\Delta\boldsymbol{\varepsilon}^\top\mathbf{Q}_l\Delta\boldsymbol{\varepsilon} + 2\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\left\{\boldsymbol{\Gamma}_\theta{}^{-1}\Delta\dot{\hat{\boldsymbol{\theta}}} + \boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\right\}\boldsymbol{\Lambda}\right). \tag{4.22}$$

L'annexe B.1 donne la démarche complète pour obtenir cette dérivée. La forme actuelle de $\dot{V}$

ne nous permet pas de conclure sur son signe. Or, pour que la dynamique soit stable, il faut que $\dot{V} \leq 0$. Si le second terme est nul, alors on peut borner la fonction de la façon suivante :

$$\dot{V} = -\Delta\boldsymbol{\varepsilon}^\top \mathbf{Q}_l \Delta\boldsymbol{\varepsilon} \leq -\lambda_{min}\left(\mathbf{Q}_l\right) \|\boldsymbol{\varepsilon}\|^2 \leq 0. \tag{4.23}$$

Sous cette forme, la dérivée de la fonction candidate est semi-définie négative et le système est stable. En imposant la loi d'adaptation suivante :

$$\Delta\dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}_\theta \boldsymbol{\omega} \Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{B}, \tag{4.24}$$

on sait que $\dot{V} \leq 0$, ce qui implique que l'erreur en boucle fermée est uniformément stable. On définit maintenant $\boldsymbol{\Gamma}_\theta$ comme la matrice des coefficients d'adaptation des gains. Cette matrice est utilisée pour régler la force de la commande adaptative pour qu'elle génère un effort de commande du même ordre de grandeur que la commande nominale. Par simplicité, elle est choisie diagonale $\boldsymbol{\Gamma}_\theta = \mathrm{diag}(\gamma_1, \ldots \gamma_{(m+n+1)})$. Ceci permet d'ajuster la force de la commande en fonction de chacun des éléments de $\boldsymbol{\omega}$ individuellement.

Ainsi, $\Delta\boldsymbol{\varepsilon}$ et $\Delta\hat{\boldsymbol{\theta}}$ sont uniformément bornés dans le temps. Puisque $\Delta\mathbf{r}$ est nécessairement borné et que $\mathbf{A}_{ref}$ est stable, alors $\Delta\mathbf{x}_{a,ref}$ et $\Delta\mathbf{x}_a$ sont aussi uniformément bornés. On a donc que $\Delta\dot{\mathbf{x}}_a$ et $\Delta\dot{\boldsymbol{\varepsilon}}$ sont également bornés, car tous les éléments des équations (4.18) et (4.19) sont bornés. Ceci implique que la fonction candidate V est bornée. Donc, sa limite lorsque $t \to \infty$ existe et elle est finie. De plus, la seconde dérivée de V est :

$$\ddot{V} = -2\Delta\boldsymbol{\varepsilon}^\top \mathbf{Q}_l \Delta\dot{\boldsymbol{\varepsilon}}. \tag{4.25}$$

Celle-ci est également bornée, car tous les éléments qui la composent sont bornés. Donc, en utilisant le corollaire 4.2.1, on a $\lim_{t\to\infty} \dot{V}(t) = 0$. On conclut que $\Delta\boldsymbol{\varepsilon}$ tend vers l'origine globalement, uniformément et asymptotiquement, ce qui complète la preuve de stabilité. Cela nous permet également de confirmer que l'hypothèse 4.2.1 est respectée, car le système incertain est garanti de converger vers le modèle de référence.

La structure générale de la MRAC est illustrée à la figure 4.7 sous la forme d'un schéma bloc et la figure 4.8 donne le schéma bloc utilisé pour implémenter la loi d'adaptation sur l'AscTec Pelican.

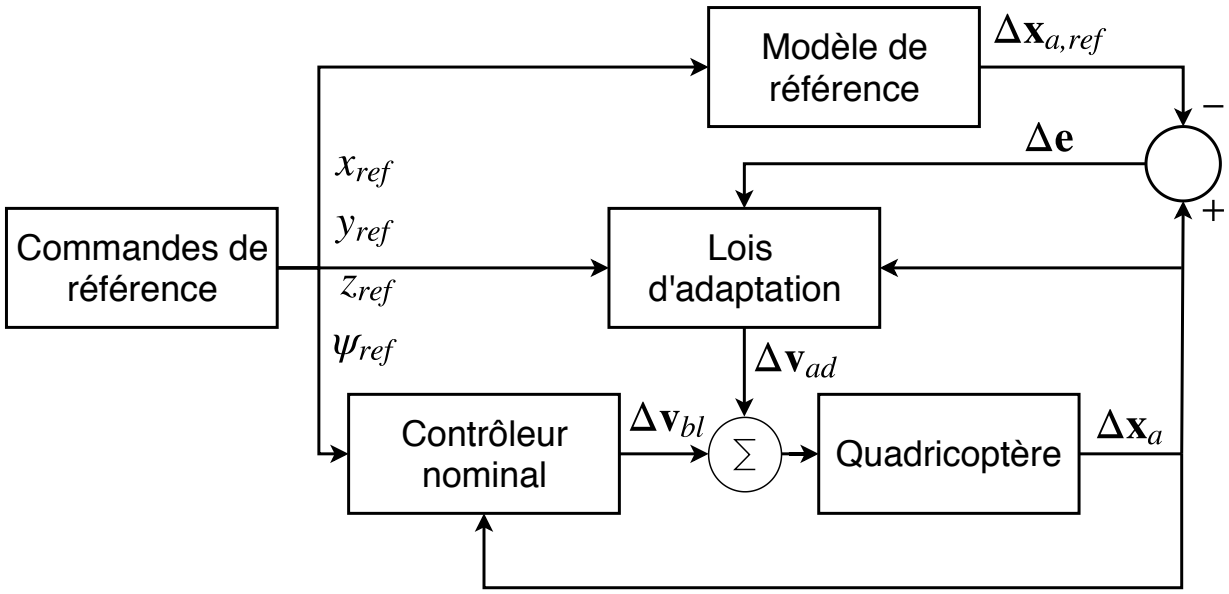


Figure 4.7 Schéma bloc général de la MRAC

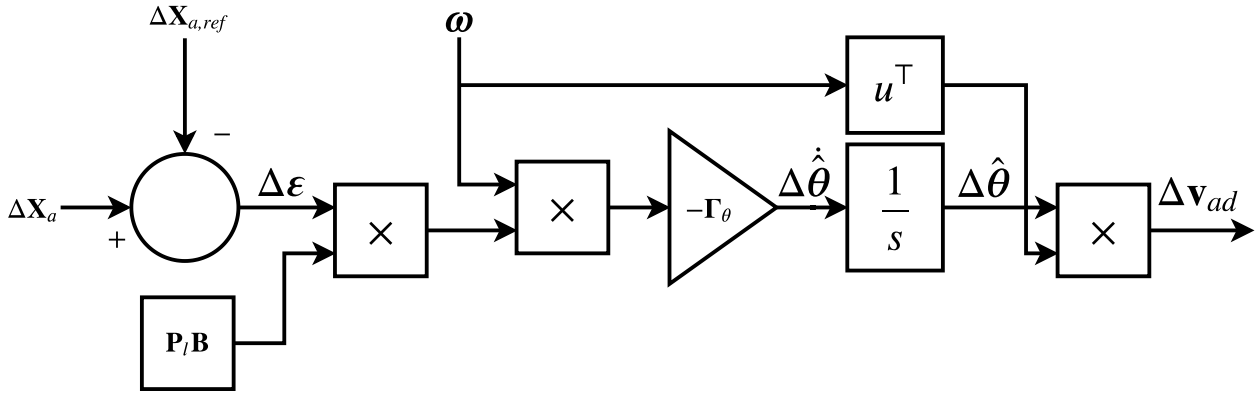


Figure 4.8 Schéma bloc détaillé pour la loi adaptative

4.3 Implémentation des contrôleurs et technique d'ajustement

L'implémentation des algorithmes de commande sur des systèmes physiques est un processus complexe qui nécessite souvent plusieurs itérations avant d'obtenir des résultats adéquats. Nous illustrons ce processus avec la figure 4.9 où les flèches en bleu représentent les itérations qui sont effectuées. Si les résultats d'une étape ne sont pas concluants, un retour en arrière est effectué pour les améliorer. Pour obtenir une MRAC augmentée d'un contrôleur de base, on effectue cette méthodologie deux fois, car il faut procéder à l'ajustement de la matrice de

gains $\mathbf{K}_a$ du LQR et de la loi d'adaptation $\Delta\dot{\hat{\boldsymbol{\theta}}}$. Pour la commande LQ, le concepteur fait le réglage par les matrices de pondération $\mathbf{Q}$ et $\mathbf{R}$, tandis que pour la commande adaptative, l'ajustement est fait grâce à la matrice $\mathbf{Q}_l$ de l'équation de Lyapunov et à la matrice des coefficients d'adaptation des gains $\mathbf{\Gamma}_\theta$. La revue de littérature n'a pas permis de trouver une méthode de réglage fiable et systématique pour ces deux contrôleurs.

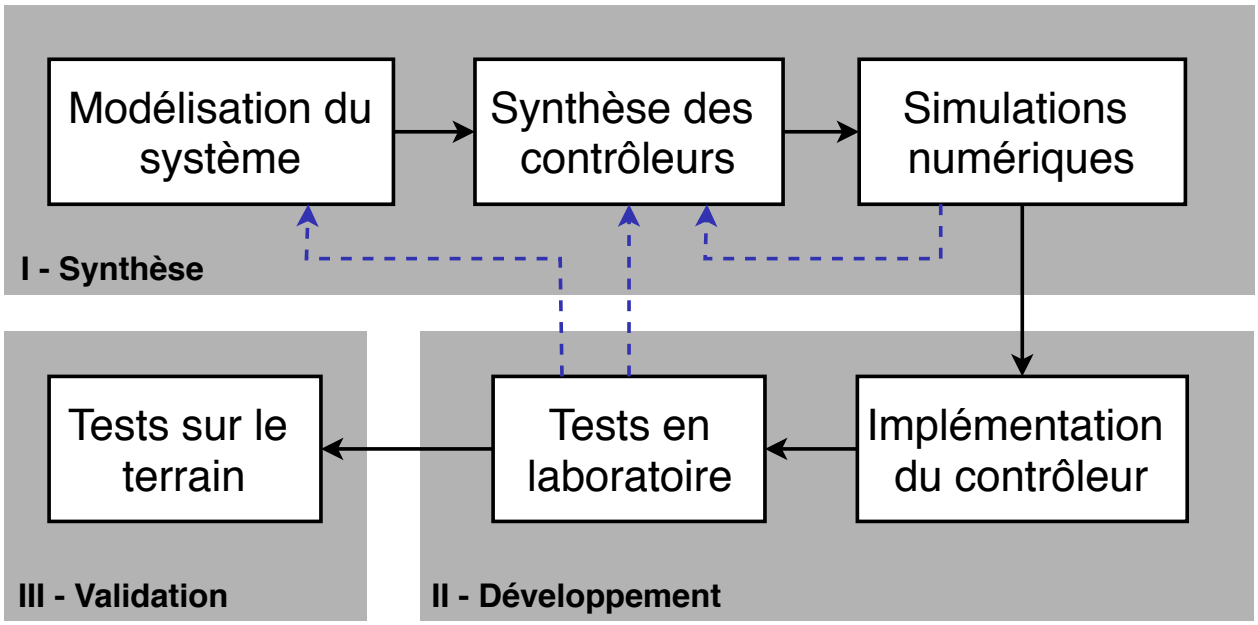


Figure 4.9 Méthodologie pour la synthèse et l'ajustement des lois de commande

Généralement, les contrôleurs sont ajustés en testant les performances du système dans des conditions idéales. Comme discuté à la section 3.1.4, on sépare la dynamique du quadricoptère en quatre sous-systèmes indépendants. Une des méthodes d'ajustement consiste à régler, en simulation, les gains associés à un sous-système à la fois. Lorsque les performances sont adéquates, on passe à un autre sous-système. Or, ne stimuler qu'un sous-système à la fois donne des résultats mitigés qui sont généralement insuffisants en pratique lorsque le quadricoptère est soumis à des conditions non idéales.

Une des contributions de ce projet de maîtrise est de proposer une méthode d'ajustement des lois de commande permettant d'obtenir de meilleurs résultats, tant en simulation que lors de vols expérimentaux. Avec notre méthode, les résultats sont obtenus plus rapidement et en moins d'itérations que le méthode plus traditionnelle. Sans pouvoir garantir que cette méthode est systématique, nous avons pu constater son efficacité pour l'ajustement des lois de commande de l'AscTec Pelican.

La technique développée repose sur l'imposition de conditions de vol non idéales pour le quadricoptère. Par exemple, nous avons considéré les cas d'une perte d'efficacité sur un moteur et de l'ajout d'une masse additionnelle près du centre de masse de l'appareil. Plus il y a de sous-systèmes impliqués lors des simulations et des tests, plus il est facile de distinguer des améliorations au niveau de la performance lors de l'ajustement des lois de commande. Notre hypothèse repose sur le raisonnement suivant :

- En simulation, l'environnement est toujours idéalisé par rapport aux tests physiques. Ainsi, il est possible de séparer les quatre sous-systèmes en simulation, mais pas lors de situations réelles.
- Le système physique contient des perturbations, des délais de transmission, des imperfections en ce qui concerne les moteurs, la batterie, etc. Ce n'est pas le cas du système simulé.
- Cette contradiction entre les modèles fera en sorte que les simulations n'apportent pas les informations souhaitées à propos du système physique.
- Si les simulations ne sont pas suffisamment fiables, alors le processus d'ajustement des contrôleurs est inefficace, car il faut vérifier expérimentalement si le contrôleur simulé est bon à chacune des itérations.

En stimulant l'ensemble de l'appareil, il est possible de négliger les imperfections mineures du quadricoptère. La correspondance entre les simulations et les vols expérimentaux est alors accrue. Pour l'AscTec Pelican, nous avons considéré une perte d'efficacité de 30% dans le moteur 1 lors des simulations. De plus l'ajustement des deux contrôleurs a été effectué afin d'obtenir les critères de performance suivant :

- dépassement maximal d'au plus 5% pour tous les mouvements de l'appareil ;
- temps de réponse de 2 à 4 secondes pour un changement en altitude ;
- temps de réponse de 5 à 8 secondes pour les déplacements dans le plan x–y ;
- temps de réponse de 5 à 12 secondes pour une rotation en lacet.

Pour la commande LQ, les matrices de pondération initiales suivantes ont été choisies :

$$\begin{aligned} \mathbf{Q} &= \mathrm{diag}(200, 200, 25, 50, 50, 1, 1, 1, 75, 50, 50, 1, 150, 150, 75, 100), \\ \mathbf{R} &= \mathrm{diag}\,(1, 2500, 2500, 5000)\,. \end{aligned} \tag{4.26}$$

Après plusieurs itérations de notre méthode, nous avons modifié ces matrices pour obtenir :

$$\begin{aligned} \mathbf{Q} &= \mathrm{diag}(50, 50, 100, 1, 1, 1, 300, 300, 100, 1, 1, 1, 130, 130, 100, 100), \\ \mathbf{R} &= \mathrm{diag}(3, 1200, 1200, 5000). \end{aligned} \tag{4.27}$$

Dans des conditions idéales, les performances du quadricoptère avec ces deux ensembles de matrices de pondération sont équivalentes. Toutefois, le deuxième cas est le plus performant lors de situations problématiques.

Pour la commande adaptative, il y a un total de 40 paramètres à ajuster, comparativement à 20 pour la commande LQ. Les matrices $\mathbf{\Gamma}_\theta$ et $\mathbf{Q}_l$ sont dépendantes : des modifications dans $\mathbf{\Gamma}_\theta$ produiront les mêmes effets sur la commande adaptative que des modifications dans $\mathbf{Q}_l$. Ainsi, pour simplifier l'ajustement de la MRAC, nous avons fixé $\mathbf{Q}_l = \mathbf{I}$. Nous séparons ensuite la matrice $\mathbf{\Gamma}_\theta$ en quatre parties, une pour chaque sous-système, soit $\mathbf{\Gamma}_{\theta,x}$, $\mathbf{\Gamma}_{\theta,y}$, $\mathbf{\Gamma}_{\theta,z}$ et $\mathbf{\Gamma}_{\theta,\psi}$. Pour rendre l'ajustement de ces matrices plus intuitif, la procédure est divisée en deux étapes. D'abord on crée une matrice de ratio $\mathbf{\Gamma}_r$ où les éléments diagonaux $\gamma_{r,i}$ varient dans l'intervalle $[0, 1]$. On détermine quels sont les états qui ont besoin d'une forte adaptation en vol. Ces états ont une valeur de $\gamma_{r,i}$ proche de 1. Les états qui n'ont pas besoin d'une adaptation aussi forte ont une valeur de $\gamma_{r,i}$ plus proche de 0. Nous avons une matrice de ratio par sous-système, i.e., $\mathbf{\Gamma}_{\theta,r,x}$, $\mathbf{\Gamma}_{\theta,r,y}$, $\mathbf{\Gamma}_{\theta,r,z}$ et $\mathbf{\Gamma}_{\theta,r,\psi}$. Voici des recommandations obtenues à la suite des tests effectués en laboratoire pour ajuster ces matrices de ratios :

- les ratios des états rapides doivent être supérieurs à ceux des états lents ;
- les ratios associés aux commandes de référence et aux perturbation non paramétriques assorties doivent être plus faible, car ces signaux varient lentement.

La seconde étape consiste à ajuster la force globale de la commande adaptative. Cela est fait à l'aide d'une constante qui multiplie la matrice de ratios. Ces constantes sont : γ_x, γ_y, γ_z, γ_ψ, une pour chaque sous-système. Ces valeurs sont ajustées pour que l'effort de commande de la commande adaptative soit du même ordre de grandeur que celle générée par le contrôleur nominale. La succession d'itérations entre les simulations et les tests en laboratoire a donné l'ajustement suivant :

$$\begin{aligned}
\gamma_x &= 0.25, \quad \mathbf{\Gamma}_{\theta,r,x} = \mathrm{diag}(0.3, 0.3, 1, 0, 0.1, 0.1, 0.1), \\
\gamma_y, &= 0.25 \quad \mathbf{\Gamma}_{\theta,r,y} = \mathrm{diag}(0.3, 0.3, 1, 0, 0.1, 0.1, 0.1), \\
\gamma_z, &= 50 \quad \mathbf{\Gamma}_{\theta,r,z} = \mathrm{diag}(1, 1, 1, 1, 1), \\
\gamma_\psi, &= 0.25 \quad \mathbf{\Gamma}_{\theta,r,\psi} = \mathrm{diag}(1, 1, 1, 1, 1).
\end{aligned} \tag{4.28}$$

4.4 Simulation des contrôleurs

4.4.1 Vol avec une masse additionnelle

La présence d'une masse supplémentaire sur le quadricoptère ajoute une incertitude paramétrique sur la masse m. Simuler ce cas permet de vérifier si la commande adaptative rejette

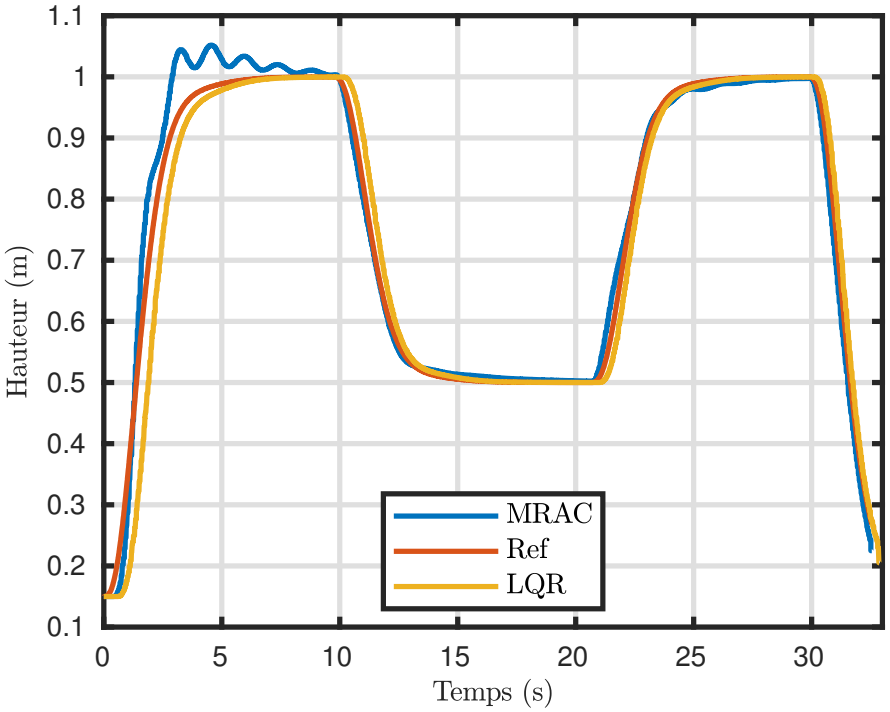


Figure 4.10 Simulation d'un vol avec une masse additionnelle de 200 g au centre de masse

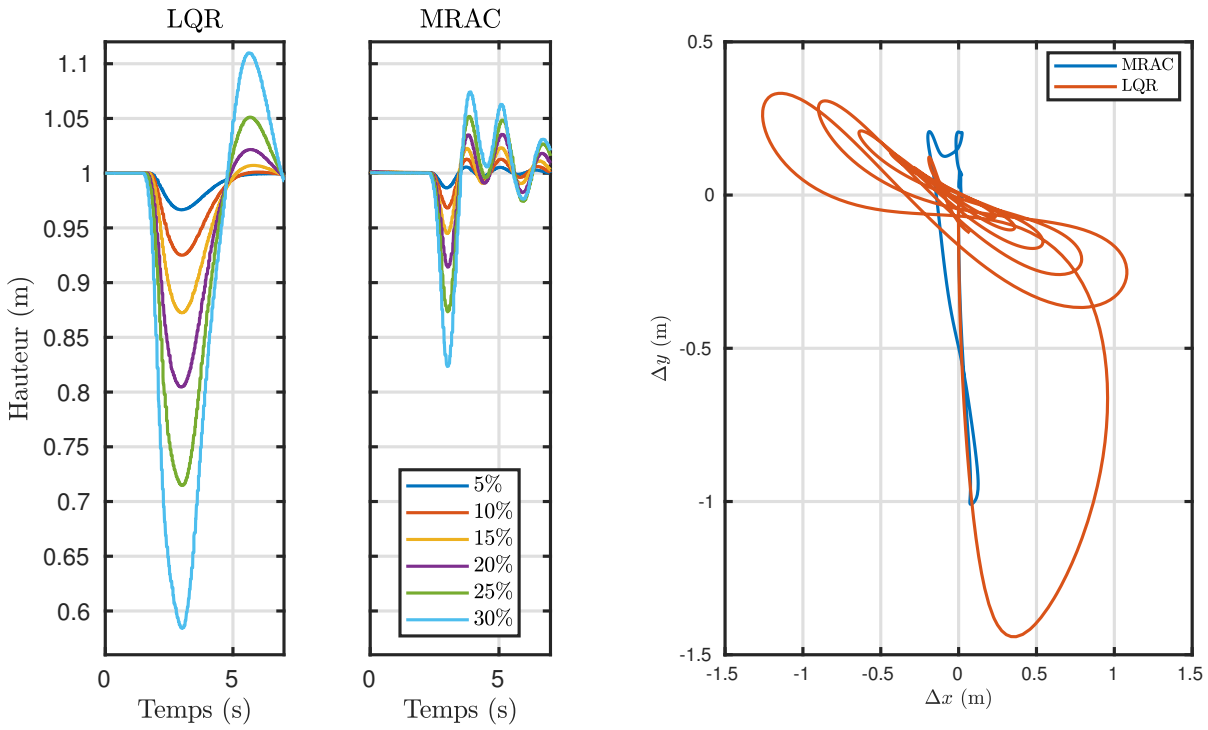
cette incertitude ; les résultats sont donnés à la figure 4.10. On remarque d'abord que le quadricoptère commandé par le LQR décolle en retard par rapport au modèle de référence. Ce retard est conservé pour chaque commande successive à $t = 10$ s et à $t = 21$ s. Avec la MRAC, le quadricoptère décolle plus rapidement et converge parfaitement sur le modèle de référence pour les commandes subséquentes. Cependant, on observe un dépassement avec des oscillations lors de la première commande entre $t = 1$ s et $t = 10$ s. Les oscillations et le dépassement disparaissent pour les commandes suivantes. Cela démontre bien l'adaptation effectuée par la MRAC pour converger vers le modèle de référence. Bien que cette simulation soit concluante, elle donne de l'information limitée sur la MRAC, car elle n'implique pas les autres dynamiques du quadricoptère. Cependant, ces résultats permettent de démontrer que la MRAC permet de compenser un cas de base pour les incertitudes paramétriques.

4.4.2 Vol en présence d'une perte d'efficacité des moteurs

Pour les scénarios de perte d'efficacité, *Loss Of Effectiveness* (LOE), on considère 6 cas : une panne dans le premier moteur allant de 5% à 30% par incréments de 5%. Ces pannes sont des incertitudes paramétriques qui affectent la matrice $\mathbf{\Lambda}$ et qui causent deux effets :

– une perte d'altitude du quadricoptère ;
– un déplacement dans le plan x–y.

Les résultats sont donnés à la figure 4.11a pour la perte d'altitude et à la figure 4.11b pour le déplacement en position.



(a) Perte d'altitude du quadricoptère

(b) Vue du déplacement dans le plan x-y pour un LOE de 30%

Figure 4.11 Simulations pour les LOE

Pour la perte en altitude, la MRAC fournit deux améliorations par rapport au LQR. D'abord, les pertes d'altitudes sont deux fois moins importantes. Par exemple, pour une LOE de 30%, le quadricoptère chute de 16 cm avec la MRAC alors que la chute atteint 43 cm avec le LQR. Le second avantage est que la MRAC retourne plus rapidement à son point d'équilibre que la commande nominale. En effet, elle met environ 1 s à récupérer de sa chute en altitude tandis que le LQR met environ 3 s. Ces simulations laissent présager que pour des valeurs plus grandes de LOE, la MRAC pourrait empêcher le quadricoptère de s'écraser alors que la commande nominale n'y parviendrait pas.

Pour les déplacements dans le plan x–y, nous avons recensé l'oscillation maximale pour chacune des LOE. Ces valeurs sont fournies dans le tableau 4.1. On remarque d'abord que la

MRAC est bien meilleure pour limiter les déplacements en x, car elle arrive à mieux découpler les dynamiques. En effet, pour un LOE de 30%, le déplacement maximal en x est de 20 cm pour la MRAC et de 125 cm pour le LQR. La LOE affecte principalement le quadricoptère en y. Avec une panne de 30%, le quadricoptère se déplace de 100 cm avec la MRAC, ce qui est nettement meilleur que le déplacement de 144 cm obtenu avec le contrôleur nominal. Avec une forte adaptation selon x et y, on voit que la commande adaptative est plus performante que la commande nominale.

Tableau 4.1 Oscillations dans le plan x-y durant les tests LOE

LOE Test	Oscillations maximales			
	LQR		MRAC	
	x	y	x	y
5%	0.0110	0.1532	0.0052	0.1281
10%	0.0534	0.3309	0.0265	0.2663
15%	0.1476	0.5402	0.0582	0.4287
20%	0.3300	0.7885	0.0979	0.6085
25%	0.6611	1.0820	0.1383	0.8003
30%	1.2560	1.4410	0.1964	1.009

Enfin, la figure 4.11b montre une vue dans le plan x–y des oscillations du quadricoptère pour une LOE de 30%. On voit que le quadricoptère oscille significativement autour de l'origine avec le LQR, alors que ce n'est pas le cas avec la MRAC. De plus, la commande adaptative découple bien les dynamiques longitudinale et latérale, car il n'y a presque pas d'oscillations selon l'axe x. La MRAC permet de limiter les déplacements dus à la panne de moteur tout en stabilisant plus rapidement le quadricoptère à l'origine.

4.4.3 Rejet des perturbations assorties : étude de cas

La MRAC est synthétisée dans le but de rejeter, entre autres, des incertitudes non paramétriques assorties. Ces incertitudes sont généralement liées à des dynamiques négligées lors de la modélisation du quadricoptère. Par exemple, nous avons discuté de la force de traînée induite par le corps de l'appareil. Or, lors des tests expérimentaux, les dynamiques négligées sont présentes. Ainsi, il est physiquement difficile de démontrer que la MRAC permet réellement de s'adapter aux dynamiques négligées expérimentalement. Néanmoins, il est possible de le faire en simulation.

Notons que la MRAC ne peut s'adapter à des incertitudes non paramétriques assorties que

si elles sont localement lipschitziennes. En se référant à (3.26), on voit que si la vitesse de l'appareil varie instantanément, alors la force de traînée induite ne sera pas lipschitzienne. Or, puisque les moteurs ont un temps de réponse qui est non nul, cela implique que la vitesse du quadricoptère ne peut pas varier instantanément. Ainsi, la traînée induite par le corps de l'appareil est localement lipschitzienne et la MRAC doit pouvoir la rejeter.

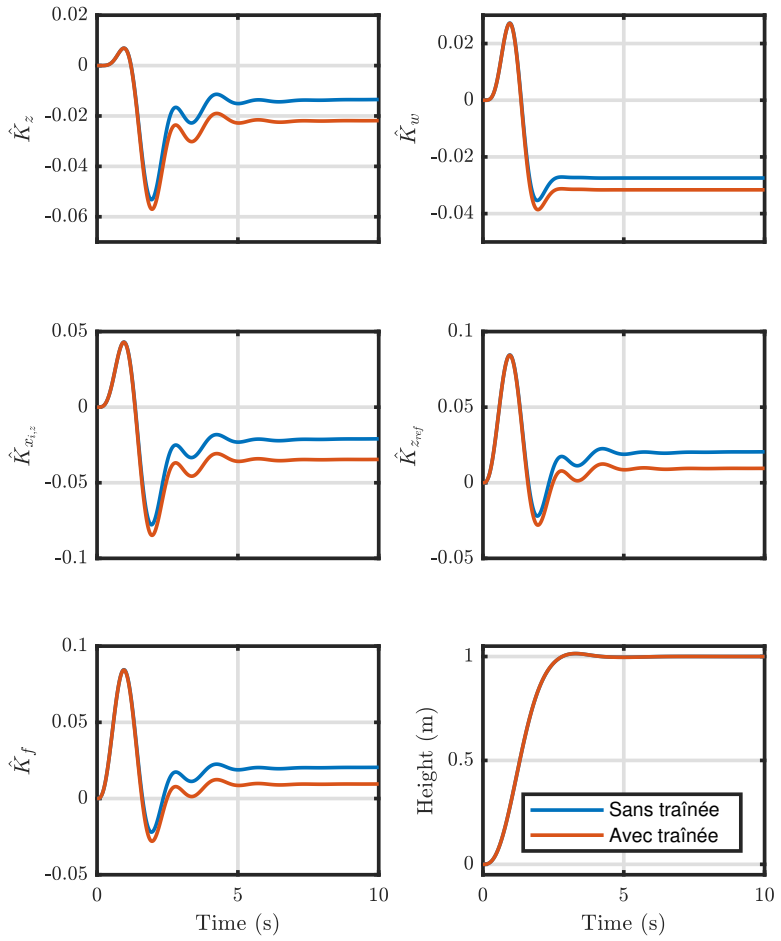


Figure 4.12 Gains adaptatifs pour le rejet de la force de traînée

Dans les simulations précédentes, nous avons donné les états du quadricoptère pour illustrer le bon fonctionnement de l'algorithme de commande adaptative. Cependant, l'effet de la traînée induite n'est pas suffisamment fort pour perturber visiblement les états du quadricoptère. Ainsi, nous montrons la variation des gains adaptatifs lorsque la traînée est ajoutée au modèle simulé. Les résultats sont présentés à la figure 4.12. On voit que la réponse temporelle n'est

pas affectée par la traînée induite. Pourtant l'évolution des gains adaptatifs est différente. La MRAC s'est donc adaptée à la perturbation non paramétrique mal-assortie.

Si la perturbation n'était pas localement lipschitzienne, alors les gains ne se stabiliseraient pas en régime permanent. En effet, cela causerait un phénomène appelé dérive des gains. Nous reviendrons en détail sur cela à la section 5.1. Néanmoins, cette simulation a permis de confirmer que la loi adaptative permet bien de rejeter les perturbations non paramétriques assorties si elles sont localement lipschitziennes.

4.5 Résultats expérimentaux

4.5.1 Critère de performance pour la MRAC

L'objectif de la MRAC est de réduire l'erreur entre le système physique et le modèle de référence. Avec l'analyse de stabilité nous avons conclu que l'erreur entre le modèle de référence et le système physique tend asymptotiquement vers zéro. De plus, cette convergence est garantie en présence de perturbations paramétriques et non paramétriques assorties. On veut maintenant valider les simulations expérimentalement sur l'AscTec Pelican. Pour comparer adéquatement la MRAC avec le contrôleur nominal. on utilise un critère de performance qui permet de détecter des améliorations, s'il y en a. Le critère retenu est l'ITSE tel que formulé dans [1] :

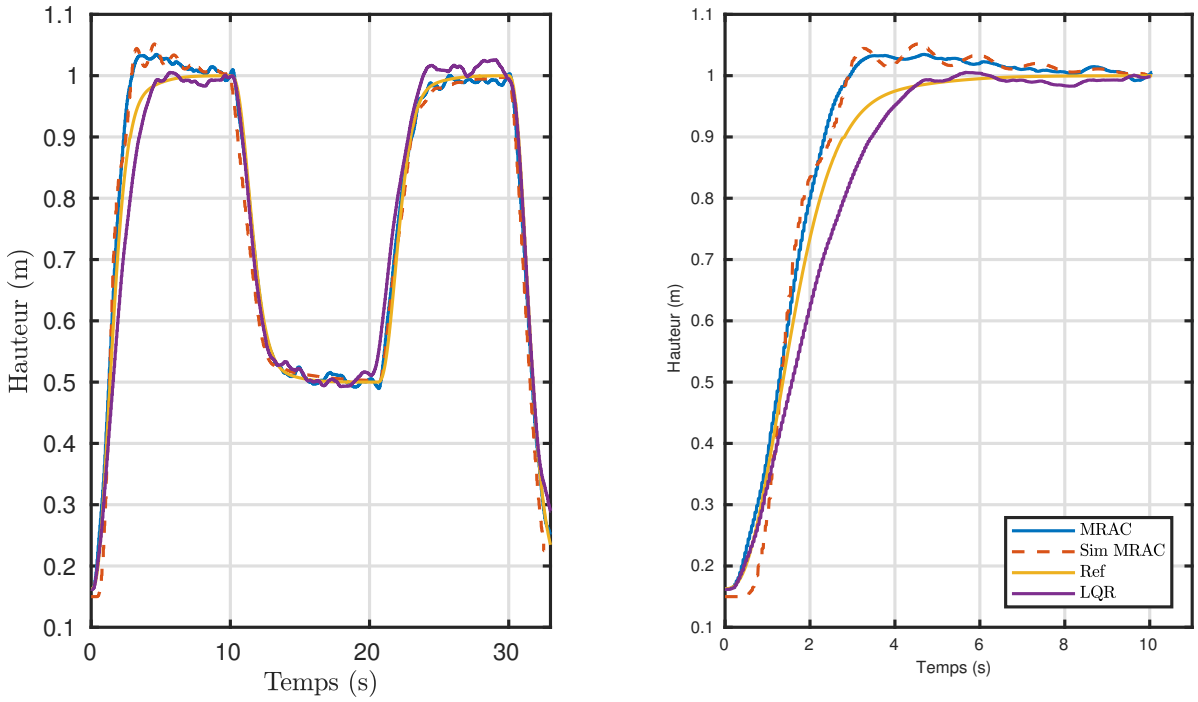
$$\text{ITSE} = \int_0^{T_f} t\epsilon^2(t)\mathrm{d}t, \tag{4.29}$$

où $\epsilon(t) = y(t) - y_{ref}(t)$ est l'erreur entre la sortie d'intérêt du modèle physique et celle du modèle de référence, et T_f le temps de vol. Les ITSE qui nous intéressent sont celles qui concernent les sorties x, y, z et ψ. Pour s'assurer que les données sont reproductibles, on effectue les vols plusieurs fois. Ainsi, on considère souvent la moyenne des ITSE pour l'ensemble des vols effectués.

On utilise ce critère pour deux raisons. D'abord on veut pénaliser l'erreur tout au long du vol. Ensuite, puisque la MRAC doit converger asymptotiquement vers le modèle de référence, on pénalise fortement les erreurs survenant dans un temps prolongé. L'ITSE est le critère de performance idéal pour mesurer rigoureusement ces deux conditions. Ainsi, les comparaisons entre le LQR et la MRAC seront valides.

4.5.2 Vol avec une masse additionnelle

Comme expliqué dans la section 4.4.1, une masse de 200 g est ajoutée au drone. Cependant, il n'est pas possible de l'ajouter directement au centre de masse. En effet, le seul endroit disponible est situé 9 cm au-dessus du centre de masse, 1 cm suivant $+\mathbf{y}_b$ et 0.5 cm suivant $-\mathbf{x}_b$. Dans ce contexte, la masse additionnelle ajoute des perturbations paramétriques tant sur la masse m du quadricoptère que sur la matrice des moments d'inertie $\mathbf{I}^b$. Ce test expérimental permet donc d'affecter l'ensemble des paramètres du quadricoptère simultanément.



(a) Échelons multiples

(b) Un seul échelon

Figure 4.13 Résultats expérimentaux avec l'ajout d'une masse additionnelle

L'ajout d'une masse supplémentaire ralentit le décollage du quadricoptère. La figure 4.13b illustre les 10 premières secondes de vol de la figure 4.13a. On voit que le LQR suit difficilement le modèle de référence à cause de la masse additionnelle, tandis que la MRAC est initialement trop forte, ce qui produit un dépassement. La courbe de simulation a été tracée pour montrer la ressemblance entre les simulations et les données expérimentales. De $t = 23$ s à $t = 30$ s, le système équipé du LQR oscille significativement autour du point d'équilibre. Cela n'est pas le cas avec la MRAC, qui malgré un dépassement initial, semble mieux suivre le modèle de

référence. Une analyse approfondie des ITSE va confirmer cette affirmation.

Pour s'assurer que les données sont reproductibles, l'expérience a été menée 20 fois avec les deux contrôleurs. Pour chacun des vols, on calcule alors les ITSE associées (Tab. 4.2). Pour bien mettre en évidence les différences entre les deux algorithmes de commande, le tableau 4.3 donne les différences en pourcentage entre les données du LQR et celles de la MRAC pour chacune des données du premier tableau. On remarque donc que la MRAC est plus performante pour toutes les données statistiques, i.e., pour la moyenne, pour l'écart type, pour le minimum et pour le maximum.

Tableau 4.2 Résultats des ITSE pour le LQR et la MRAC

	LQR				MRAC			
	ITSE_x	ITSE_y	ITSE_z	ITSE_ψ	ITSE_x	ITSE_y	ITSE_z	ITSE_ψ
Moy	0.470	0.674	0.076	0.717	0.311	0.430	0.063	0.206
Std	0.220	0.252	0.011	0.255	0.090	0.141	0.004	0.125
Min	0.249	0.311	0.061	0.440	0.174	0.213	0.056	0.090
Max	1.053	1.115	0.104	1.351	0.455	0.685	0.076	0.615

Tableau 4.3 Différences entre les ITSE avec le LQR et la MRAC

	Erreur intégrale au carré dépendante du temps			
	ITSE_x	ITSE_y	ITSE_z	ITSE_ψ
Moyenne	-33.8%	-36.2%	-17.1%	-71.3%
Écart type	-59.1%	-44.0%	-54.5%	-50.9%
Minimum	-30.1%	-31.5%	-8.2%	-79.5%
Maximum	-56.8%	-38.6%	-26.9%	-54.4%

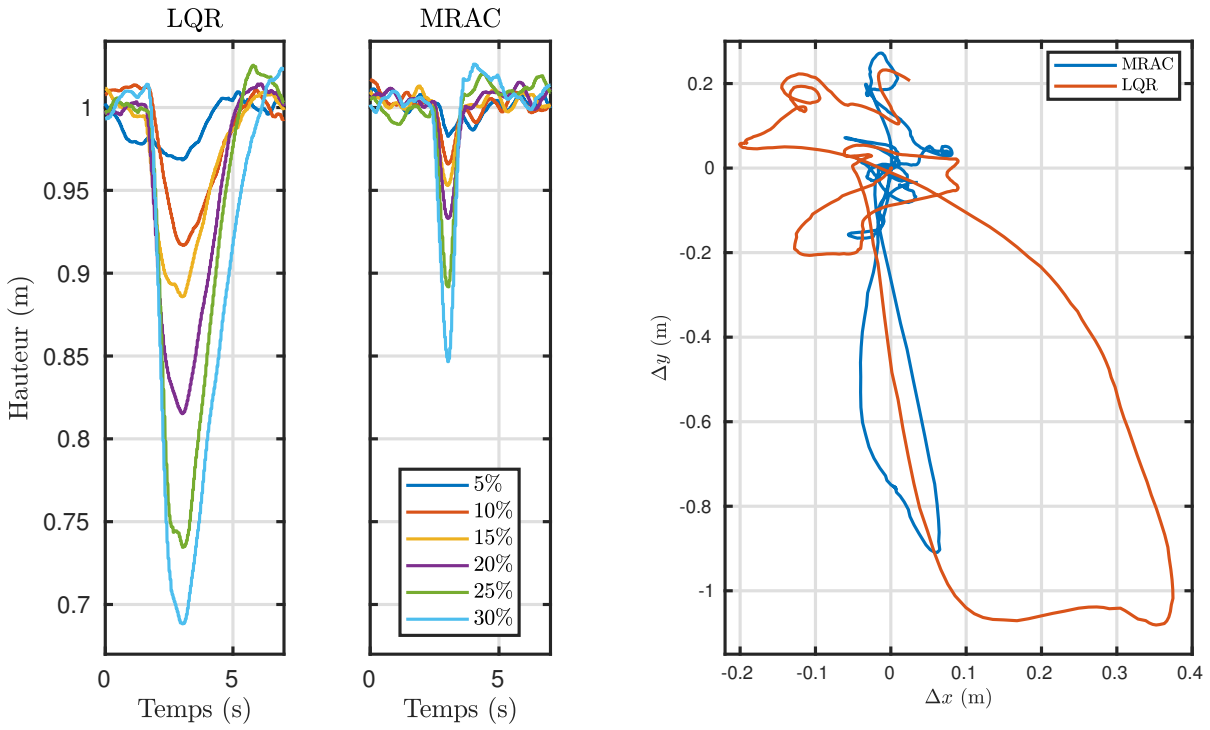
En se référant au tableau 4.3, on constate que l'écart type est nettement inférieur avec la MRAC. Cela veut dire que la commande adaptative réduit la variabilité dans les résultats obtenus. Les données de vol sont plus stables et elles varient moins autour du point d'équilibre. Un écart type réduit signifie que les résultats obtenus sont plus conséquents d'un vol à l'autre. Les lignes « Min » et « Max » donnent l'équivalent du meilleur et du pire scénario de vol. On voit que le pire scénario de la MRAC correspond à la moyenne des résultats obtenus avec le contrôleur nominal.

On conclut donc que la MRAC est meilleure pour rejeter les incertitudes paramétriques par comparaison avec le LQR lorsque ces incertitudes sont produites par l'ajout d'une masse sur le quadricoptère.

4.5.3 Vol en présence d'une perte d'efficacité des moteurs

On reprend les simulations effectuées pour les pannes de moteur sur l'AscTec Pelican avec les mêmes incréments. La figure 4.14a donne l'ensemble des résultats obtenus pour la chute en altitude provoquée par la panne de moteur. Les résultats sont similaires à ceux obtenus en simulation, mais quelques différences doivent être abordées :

- La commande LQ est plus performante qu'initialement anticipée. Jusqu'à une LOE de 20%, les résultats sont similaires aux simulations. Cependant, avec une LOE de 25%, le LQR récupère 3 cm en altitude de plus que les simulations et il est plus efficace de près de 10 cm lorsque la panne atteint 30%. Selon nous, cela est attribuable à la contribution de l'effet de sol.
- La MRAC retourne aussi rapidement à son point d'équilibre que durant les simulations. Toutefois, le LQR prend de 1 à 2 secondes supplémentaires pour y retourner.
- Les oscillations observées après la chute en altitude à la figure 4.11a ne sont pas présentes durant les tests expérimentaux.



(a) Perte d'altitude du quadricoptère

(b) Vue du déplacement dans le plan x–y pour un LOE de 30%

Figure 4.14 Résultats expérimentaux pour les LOE

Le tableau 4.4 donne les résultats des chutes en altitude pour les deux lois de commande.

On voit que dans tous les cas, la chute d'altitude avec la MRAC est au moins la moitié de celle obtenue avec le LQR. Tant que la panne de moteur ne compromet pas la capacité du quadricoptère à voler, alors la commande adaptative permettra d'empêcher un écrasement. De plus, elle le fera pour des pannes supérieures à celles que la commande nominale peut compenser.

Tableau 4.4 Différences entre la perte d'altitude maximale

LOE	Perte d'altitude		
	LQR (m)	MRAC (m)	Différence (%)
5%	0.0316	0.0165	-47.8
10%	0.0829	0.0335	-59.6
15%	0.1142	0.0465	-59.3
20%	0.1843	0.0664	-64.0
25%	0.2651	0.1082	-59.2
30%	0.3117	0.1534	-50.7

On analyse maintenant le déplacement causé par la panne de moteur dans le plan x–y. Le tableau 4.5 donne les résultats pour les deux algorithmes de commande. On remarque d'abord qu'en comparaison avec les simulations, toutes les données expérimentales sont inférieures ; selon nous, cela provient des effets de sol. En effet, plus les pannes sont importantes, et plus le quadricoptère chute en altitude. À environ 80 cm du sol, ces effets deviennent non négligeables. Ainsi, le quadricoptère équipé seulement du LQR se comporte mieux qu'en simulation. Quant à la MRAC, elle réussit à découpler parfaitement la dynamique ; les pannes de moteur affectent peu le déplacement selon l'axe x et les déplacements obtenus pour l'axe y sont similaires aux simulations. En résumé, les performances obtenues avec la MRAC sont au moins 30% meilleures selon l'axe x et au moins 10% meilleures selon l'axe y que celles du LQR.

Tableau 4.5 Déplacements maximaux dans le plan x-y avec les pannes de moteur

LOE	Déplacement maximal					
	LQR		MRAC		Différence (%)	
	x	y	x	y	x	y
5%	0.0411	0.1879	0.0117	0.1221	-71.5%	-35.0%
10%	0.1437	0.2501	0.0911	0.2269	-36.6%	-9.2%
15%	0.2380	0.4948	0.0888	0.3866	-62.8%	-21.9%
20%	0.2600	0.6723	0.0628	0.4917	-75.8%	-26.9%
25%	0.2529	0.8538	0.0848	0.6760	-66.5%	-20.8%
30%	0.3745	1.0780	0.0641	0.9087	-82.9%	-15.7%

La figure 4.14b illustre le déplacement dans le plan x–y pour une panne de moteur de 30%. On remarque que le système avec le LQR oscille beaucoup moins autour de l'origine que durant les simulations. Ceci est attribuable à une erreur de modélisation de l'appareil dans ROS–Gazebo. Cette figure illustre aussi comment le quadricoptère bénéficie des effets de sols. Au point $(x = 0.38,\ y = -1.1)$, le quadricoptère est au plus près du sol. Les effets de sol lui permettent d'être plus réactif, tant pour le déplacement en x qu'en y. Cela explique pourquoi il est capable de mieux récupérer de sa chute que ce qui a pu être prédit par la simulation. Enfin, on note que les simulations sur ROS–Gazebo ne tiennent pas compte des effets de sol.

4.6 Limites de la MRAC et perturbations mal-assorties

La MRAC synthétisée dans ce chapitre a pour but de compenser des incertitudes paramétriques et non paramétriques assorties. Pour obtenir un système respectant ces critères, les incertitudes non paramétriques mal-assorties $\boldsymbol{\xi}(t)$ sont supposées nulles. Or, cette supposition est fausse, car ces incertitudes sont présentes dans tout système physique sous la forme de bruit de mesure, de bruit de commande, de vent, etc. Il n'est donc pas possible de les négliger.

Dans ce contexte, la MRAC n'est plus garantie d'être robuste. On peut assister alors à un phénomène appelé dérive des gains. L'intuition est la suivante : avec la présence de bruit, il n'est plus possible pour le système d'atteindre une erreur parfaitement nulle entre le modèle réel et le modèle de référence. Ainsi, le processus d'adaptation ne s'arrête jamais et les gains adaptatifs continuent d'évoluer. Les gains deviennent de plus en plus fort pour forcer le quadricoptère à converger vers le modèle de référence. Or, comme il n'est pas atteignable, les gains vont finir par diverger. Cela peut causer l'instabilité et mener à l'écrasement de l'appareil. Notons que si le système est en excitation persistante, ce raisonnement n'est pas valide. Or, pour un système qui opère en vol stationnaire ou en suivi de trajectoire, l'excitation persistante est impossible. De le raisonnement ci-haut est valide.

Pour limiter ce phénomène, il faut modifier la loi adaptative en ajoutant des modifications robustes. Il existe deux types de modification robuste pour éliminer ce phénomène. Le premier type consiste à modifier le critère d'erreur en définissant une zone où le processus d'adaptation doit s'arrêter. Le second consiste à borner les gains adaptatifs pour les empêcher de diverger. Nous explorerons le premier type dans le chapitre 5, et le second dans le chapitre 6.

4.7 Conclusion du chapitre

Nous avons d'abord procédé à la synthèse d'un LQR dans le but d'obtenir un contrôleur nominal qui fonctionne adéquatement dans des conditions idéales. Ensuite, une MRAC a été ajoutée à l'algorithme de commande dans le but d'améliorer les performances du quadricoptère si celui-ci est soumis à des incertitudes paramétriques et des perturbations non paramétriques assorties. L'analyse de stabilité garantit que la loi d'adaptation converge, assurant ainsi la stabilité du contrôleur.

Dans le but de faciliter l'implémentation de ces lois de commande sur un quadricoptère, nous avons développé une technique d'ajustement des gains permettant d'obtenir rapidement des contrôleurs fonctionnels grâce à un processus itératif efficace et systématique. Ainsi, nous avons obtenu des contrôleurs performants qui garantissent les bonnes performances de l'appareil en vol.

Les trois scénarios de simulation ont permis de montrer que la MRAC est effectivement capable de rejeter des incertitudes paramétriques et des incertitudes non paramétriques assorties. Ces constats théoriques ont ensuite été validés en pratique lors de vols expérimentaux de l'AscTec Pelican. De plus, nous avons été en mesure d'observer que les performances du quadricoptère étaient meilleures avec la MRAC par comparaison avec la commande nominale LQ. Ce chapitre a permis de quantifier les améliorations auxquelles on peut s'attendre avec une MRAC tout en expliquant comment l'ajuster rapidement et efficacement.

CHAPITRE 5 MODIFICATIONS ROBUSTES TRADITIONNELLES

Les modifications robustes ont pour objectif d'augmenter la robustesse de la loi adaptative afin de limiter l'effet de dérive des gains causé par les incertitudes non paramétriques mal-assorties. La section 5.1 servira d'explication mathématique sur le phénomène de dérive des gains en démontrant que celui-ci affectera assurément le système en présence de ces perturbations.

Dans ce chapitre, nous aborderons trois modifications robustes : la modification de la zone morte, la modification–σ, ainsi que la modification–ϵ. Ensemble, elles constituent les modifications robustes traditionnelles, car ce sont les méthodes les plus utilisées en pratique et les premières à avoir été développées. Ces modifications robustes ont aussi la caractéristique d'être indirectes, car elles n'agissent pas sur les gains adaptatifs, mais sur l'erreur entre le modèle de référence et le système physique. La modification de la zone morte sera abordée à la section 5.2. La modification–σ et la modification–ϵ seront abordées respectivement aux sections 5.3 et 5.4.

Ce chapitre prendra la forme d'une étude comparative. En effet, nous souhaitons déterminer les avantages et les inconvénients de chacune de ces modifications robustes. Nous voulons être en mesure de déterminer quelle modification robuste donne les meilleurs résultats une fois qu'elle est implémentée sur l'AscTec Pelican. La section 5.5 servira de recommandation à cet effet.

5.1 Problème de dérive des gains

Nous rappelons que le système linéaire contenant des incertitudes paramétriques et non paramétriques assorties et mal-assorties est donné par :

$$\dot{\mathbf{x}} = \mathbf{A}_u\mathbf{x} + \mathbf{B}\boldsymbol{\Lambda}\left(\mathbf{u} + \mathbf{f}(\mathbf{x})\right) + \boldsymbol{\xi}(t). \tag{5.1}$$

On s'intéresse à l'analyse de la stabilité de ce système si les incertitudes non paramétriques mal-assorties ne sont pas nulles, i.e., $\boldsymbol{\xi}(t) \neq \mathbf{0},\ \forall t \geq 0$.

5.1.1 Préliminaires mathématiques

L'analyse de la stabilité du système en présence de ces incertitudes nécessite des connaissances plus approfondies de la théorie de stabilité de Lyapunov. Le lecteur peut se référer à [19, Chapitre 4] pour tous les détails concernant cette théorie. On reproduira ici le lemme 4.3 et le théorème 4.18 qui sont essentiels pour la démonstration effectuée dans la prochaine section.

Lemme 5.1.1. *Soit $V : \mathbb{R}^n \to \mathbb{R}$, une fonction définie positive sur un domaine $D \subset \mathbb{R}^n$ qui contient l'origine. Soit $B_r \subset D$, une boule de rayon $r > 0$. Alors il existe des fonctions de classe $\mathcal{K}$, α_1 et α_2, définies sur $[0,\ r]$ telles que*

$$\alpha_1\left(\|\mathbf{x}\|\right) \leq V(\mathbf{x}) \leq \alpha_2\left(\|\mathbf{x}\|\right), \quad \forall \mathbf{x} \in B_r. \tag{5.2}$$

De plus, si $D = \mathbb{R}^n$, alors les fonctions α_1 et α_2 sont définies sur $[0, \infty)$ et les inégalités précédentes sont valides pour tout $\mathbf{x} \in \mathbb{R}^n$. Si $V(\mathbf{x})$ est radialement non bornée, alors α_1 et α_2 peuvent être de classe $\mathcal{K}_\infty$.

Le lecteur pourra se référer à [19] pour la définition des classes $\mathcal{K}$ et $\mathcal{K}_\infty$. On considère ensuite le théorème suivant.

Théorème 5.1.2. *Soit $D \subset \mathbb{R}^n$, un domaine qui contient l'origine et $V : [0, \infty) \times D \to \mathbb{R}$, une fonction continûment différentiable telle que*

$$\begin{gathered} \alpha_1\left(\|\mathbf{x}\|\right) \leq V(t, \mathbf{x}) \leq \alpha_2\left(\|\mathbf{x}\|\right), \\ \frac{\partial V}{\partial t} + \frac{\partial V}{\partial \mathbf{x}} f(t, \mathbf{x}) \leq -W_3\left(\mathbf{x}\right), \quad \forall \|\mathbf{x}\| \geq \mu > 0, \end{gathered} \tag{5.3}$$

$\forall t \geq 0$ et $\forall \mathbf{x} \in D$, où α_1 et α_2 sont des fonctions de classe $\mathcal{K}$ et $W_3(\mathbf{x})$ est une fonction

continue et définie positive. Si $r > 0$ est tel que $B_r \subset D$ et

$$\mu < \alpha_2^{-1}(\alpha_1(r)), \tag{5.4}$$

alors, il existe une fonction β de classe $\mathcal{KL}$ tel que pour toutes les conditions initiales $\mathbf{x}(t_0)$ satisfaisant $\|\mathbf{x}(t_0)\| \leq \alpha_2^{-1}(\alpha_1(r))$, il $\exists T \geq 0$ qui dépend de $\mathbf{x}(t_0)$ et μ tel que

$$\begin{aligned} &\|\mathbf{x}(t)\| \leq \beta(\|\mathbf{x}(t_0)\|, t - t_0), && \forall t_0 \leq t \leq t_0 + T, \\ &\|\mathbf{x}(t)\| \leq \alpha_2^{-1}(\alpha_1(\mu)), && \forall t \geq t_0 + T. \end{aligned} \tag{5.5}$$

De plus, si $D = \mathbb{R}^n$ et α_1 est de classe $\mathcal{K}_\infty$, alors les inégalités précédentes sont vraies peu importe la condition initiale $\mathbf{x}(t_0)$ et sans restriction aucune sur la grandeur de μ.

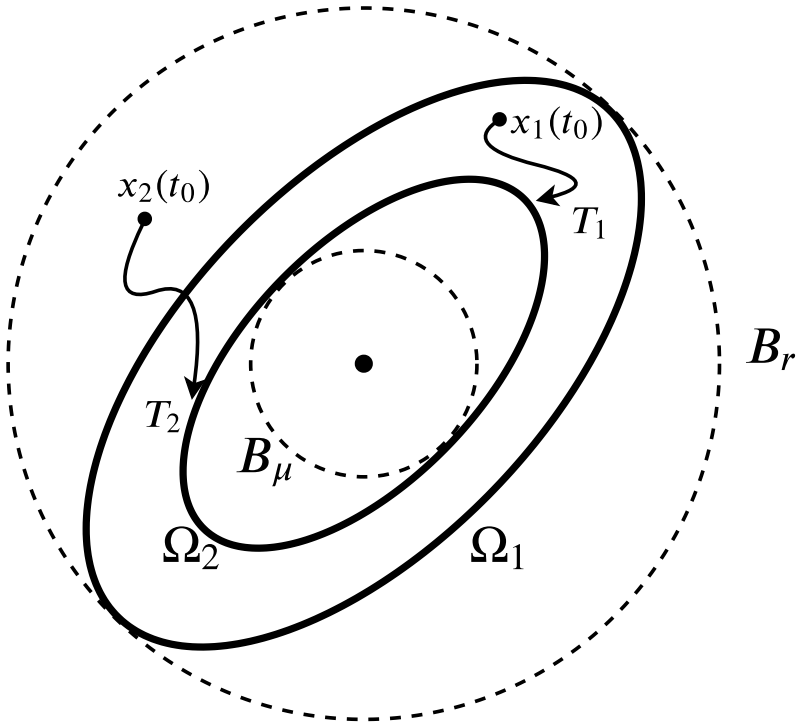


Figure 5.1 Illustration du théorème 5.1.2

Le lecteur peut se référer à [19] pour la preuve de ce théorème et du lemme précédent ainsi que pour la définition d'une classe $\mathcal{KL}$. Nous allons maintenant procéder à l'illustration des conclusions de ce théorème. On définit une boule suffisamment grande $B_r = \{\mathbf{x} \in \mathbb{R} \mid \|x\| \leq r\} \subset D$. La deuxième inégalité de (5.3) définit une fonction candidate de Lyapunov comme étant négative si $\|\mathbf{x}\| \geq \mu$. On définit alors une boule $B_\mu = \{\mathbf{x} \in \mathbb{R} \mid \|x\| \leq \mu\} \subset D$ où la

dérivée de $V(t, \mathbf{x})$ n'est pas nécessairement définie négative. Avec la première inégalité, on a que $V(t, \mathbf{x})$ est bornée par α_1 et α_2. Grâce aux bornes sur $\|\mathbf{x}\|$ imposées par B_r et B_μ, on peut définir les deux ensembles suivants :

- si $\|\mathbf{x}\| \leq r$, alors $\alpha_1(\|\mathbf{x}\|) \leq \alpha_1(r)$. On définit alors $\Omega_1 = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq \alpha_1(r)\}$ et $\Omega_1 \subset B_r$;
- si $\|\mathbf{x}\| \geq \mu$, alors $\alpha_2(\|\mathbf{x}\|) \geq \alpha_2(\mu)$. On définit alors $\Omega_2 = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq \alpha_2(\mu)\}$ et $B_\mu \subset \Omega_2$.

Puisque $\alpha_1(r) > \alpha_2(\mu)$, alors il faut que $\Omega_2 \subset \Omega_1$. Cette condition est vérifiée si et seulement si

$$\alpha_2(\mu) < \alpha_1(r) \Rightarrow \mu < \alpha_2^{-1}(\alpha_1(r)) \Rightarrow \|\mathbf{x}\| < \alpha_2^{-1}(\alpha_1(r)). \tag{5.6}$$

La figure 5.1 permet d'illustrer les différents ensembles et sous-ensembles présents dans le théorème 5.1.2. Les conclusions suivantes sont possibles grâce à ce théorème. D'abord, si les inégalités en (5.3) sont respectées et que la supposition (5.4) est vraie, alors, pour toute condition initiale se situant dans Ω_1, il y a un temps T maximum avant lequel le système entrera dans Ω_2. Par exemple, dans la figure 5.1, le système partant de $\mathbf{x}_1(t_0)$ entrera dans Ω_2 dans un temps fini T_1. Par contre, ce n'est pas nécessairement vrai pour le système partant de la condition initiale $\mathbf{x}_2(t_0)$, car elle n'est pas initialement dans Ω_1. Ceci dit, si $V(t, \mathbf{x})$ est radialement non bornée, alors α_1 et α_2 peuvent être exprimées comme des fonctions de classe $\mathcal{K}_\infty$ (selon le lemme 5.1.1). Également puisque $D = \mathbb{R}^n$, alors toutes les conditions initiales finiront par atteindre Ω_2, car la boule B_r peut être choisie arbitrairement grande et, par conséquent, Ω_1 aussi. Donc, toutes les conditions initiales $\mathbf{x}(t_0)$ se situeront dans Ω_1 ce qui implique qu'elles vont toutes atteindre Ω_2 dans un temps fini.

Les conclusions de ce théorème seront utilisées dans la section suivante, lors de l'analyse de la stabilité. En effet, on pourra montrer que les gains adaptatifs ne sont pas bornés lorsqu'il y a des perturbations non paramétriques mal-assorties.

5.1.2 Analyse de la stabilité

On utilise la même fonction candidate de Lyapunov que celle proposée à l'équation (4.20) :

$$V = \Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\boldsymbol{\varepsilon} + \operatorname{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top \boldsymbol{\Gamma}_\theta^{-1} \Delta\hat{\boldsymbol{\theta}} \boldsymbol{\Lambda}\right). \tag{5.7}$$

Pour que la dynamique du système soit stable, la dérivée de la fonction candidate doit être semi-définie négative. La dérivée qui contient les incertitudes non paramétriques assorties

est :

$$\dot{V} = -\Delta\boldsymbol{\varepsilon}^\top \mathbf{Q}_l \Delta\boldsymbol{\varepsilon} + 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \boldsymbol{\xi} + 2\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top \left\{\boldsymbol{\Gamma}^{-1}\Delta\dot{\hat{\boldsymbol{\theta}}} + \boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{B}\right\} \boldsymbol{\Lambda}\right). \tag{5.8}$$

La démarche complète permettant d'obtenir cette équation est donnée à l'annexe C.1. Nous reprenons la loi d'adaptation proposée à l'équation (4.24) qui permet de simplifier $\dot{V}$ à :

$$\dot{V} = -\Delta\boldsymbol{\varepsilon}^\top \mathbf{Q}_l \Delta\boldsymbol{\varepsilon} + 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \boldsymbol{\xi}. \tag{5.9}$$

Pour analyser le signe de $\dot{V}$, on obtient une borne supérieure de l'équation précédente. Pour chacun des termes de la dérivée on a :

$$-\Delta\boldsymbol{\varepsilon}^\top \mathbf{Q}_l \Delta\boldsymbol{\varepsilon} \leq -\lambda_{min}\left(\mathbf{Q}_l\right) \|\Delta\boldsymbol{\varepsilon}\|^2 \quad \text{et} \quad \Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \boldsymbol{\xi} \leq \|\Delta\boldsymbol{\varepsilon}\| \, \lambda_{max}\left(\mathbf{P}_l\right) \boldsymbol{\xi}_{max}. \tag{5.10}$$

Pour que cette dérivée soit semi-définie négative, il faut que :

$$-\lambda_{min}(\mathbf{Q}_l) \|\Delta\boldsymbol{\varepsilon}\|^2 + \|\Delta\boldsymbol{\varepsilon}\| \, \lambda_{max}\left(\mathbf{P}_l\right) \boldsymbol{\xi}_{max} \leq 0 \Rightarrow \|\Delta\boldsymbol{\varepsilon}\| \geq \frac{\lambda_{max}\left(\mathbf{P}_l\right)}{\lambda_{min}\left(\mathbf{Q}_l\right)} \boldsymbol{\xi}_{max} := \varepsilon_0. \tag{5.11}$$

L'équation précédente permet de confirmer que les inégalités (5.3) du théorème 5.1.2 sont respectées. En effet, puisque $V(\Delta\boldsymbol{\varepsilon}, \Delta\hat{\boldsymbol{\theta}})$ est définie positive et est radialement non bornée, alors, par le lemme 5.1.1, il existe des fonctions de classe $\mathcal{K}_\infty$ qui peuvent la borner. De plus, la dérivée de la fonction candidate de Lyapunov est définie négative si $\|\Delta\boldsymbol{\varepsilon}\| \geq \varepsilon_0$. Il existe donc des ensembles Ω_1 et Ω_2 tels que donnés à la figure 5.1 qui représentent le problème de stabilité de la commande adaptative. De plus, ce cas se situe dans celui, plus large, où la boule B_r peut être choisie arbitrairement grande. Ainsi, pour toute condition initiale, l'erreur $\Delta\boldsymbol{\varepsilon}$ entrera dans un temps fini dans Ω_2 et n'en ressortira plus. On définit maintenant l'ensemble $E_0 = \{\Delta\boldsymbol{\varepsilon} \in \mathbb{R}^n : \|\Delta\boldsymbol{\varepsilon}\| \leq \varepsilon_0\} \subset \Omega_2$ et on cherche à analyser la dynamique du système dans Ω_2. Dans E_0, il est impossible de conclure sur le signe de $\dot{V}$. Or, sans restreindre les valeurs de $\Delta\hat{\boldsymbol{\theta}}$, on peut montrer que $V(\Delta\boldsymbol{\varepsilon}, \Delta\hat{\boldsymbol{\theta}})$ n'est pas bornée, même si l'erreur du système est bornée et ne sortira jamais de l'ensemble Ω_2. On ne peut donc pas garantir la stabilité de la loi adaptative par le lemme 4.2.1. Cette affirmation est démontrée formellement à l'annexe C.2.

Ainsi, la loi d'adaptation proposée au chapitre précédent ne permet pas d'assurer la stabilité du système, peu importe la valeur de $\boldsymbol{\xi}_{max}$. Puisque V ne tend pas vers l'origine lorsque $t \to \infty$, le processus d'adaptation ne s'arrête jamais. C'est ce principe qui est à l'origine du phénomène de dérive des gains.

5.1.3 Présence du phénomène sur le quadricoptère

Avant de s'attaquer à limiter le phénomène de dérive des gains, on s'assure abord qu'il est bien présent lors des vols de l'AscTec Pelican. La figure 5.2 donne l'évolution temporelle de tous les gains adaptatifs lors d'un vol stationnaire. La seule incertitude non paramétrique mal-assortie qui affecte l'appareil est le bruit de mesure.

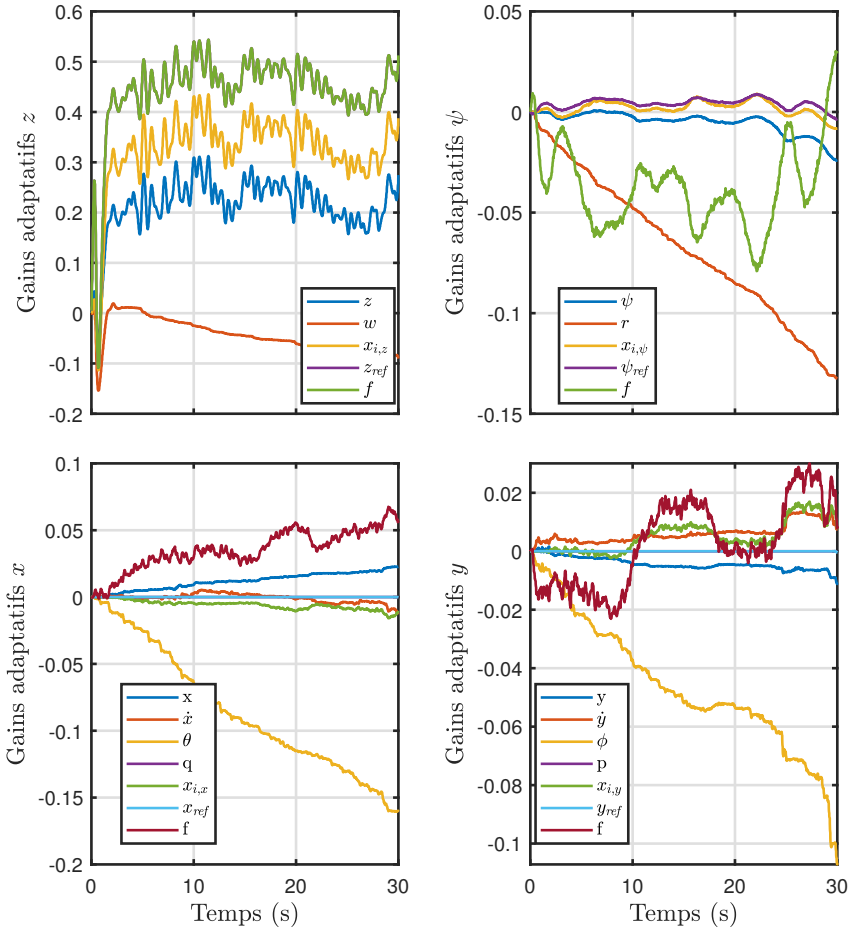


Figure 5.2 Phénomène de dérive des gains durant un vol

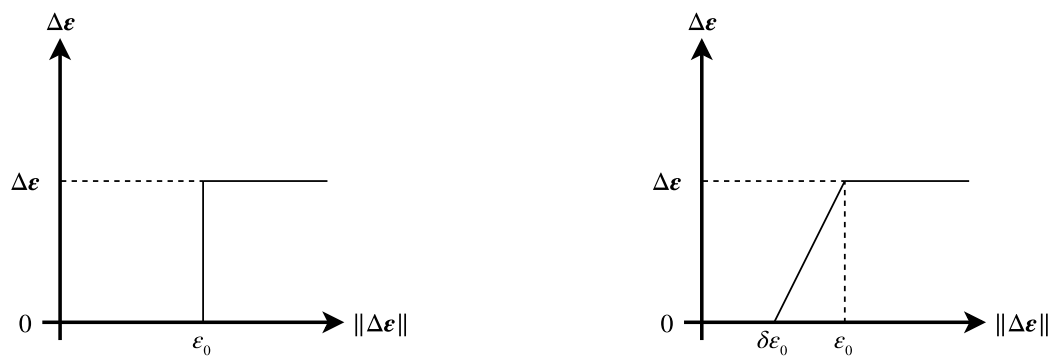
On voit que certains gains sont affectés par le phénomène de dérive. Par exemple, les gains des états r, ϕ et θ sont ceux qui présentent la plus forte divergence. Néanmoins, d'autres gains montrent aussi une divergence plus faible, par exemple, les gains des états w, ψ, $\dot{x}$, $x_{i,x}$ et $\dot{y}$. Ainsi, le bruit de mesure cause des effets indésirables sur le système s'il n'est pas rejeté. Cela pourrait mener, potentiellement, à l'instabilité et même à l'écrasement du quadricoptère. Il

faut donc remédier à ce phénomène pour assurer la robustesse complète de la loi adaptative.

5.2 Modification de la zone morte

La première modification robuste développée pour empêcher le phénomène de dérive des gains est la modification de la zone morte, inventée par [32]. Cette modification consiste à arrêter le processus d'adaptation dès que l'erreur est inférieure à un certain seuil. Elle a la forme suivante :

$$\Delta\boldsymbol{\varepsilon} = \begin{cases} \Delta\boldsymbol{\varepsilon}, & \|\Delta\boldsymbol{\varepsilon}\| > \varepsilon_0, \\ 0, & \|\Delta\boldsymbol{\varepsilon}\| \leq \varepsilon_0. \end{cases} \tag{5.12}$$



(a) Zone morte non-Lipschitz

(b) Zone morte Lipschitz

Figure 5.3 Fonctions pour la modification de la zone morte

La modification de [32] est illustrée à la figure 5.3a. Cependant, cette modification n'est pas lipschitzienne, car sa dérivée n'est pas bornée. Ainsi, l'implémentation de l'équation précédente sur le quadricoptère va causer un phénomène de broutement (*chattering*), autour de $\|\Delta\boldsymbol{\varepsilon}\| = \varepsilon_0$. Pour pallier ce défaut, on considère la modification de la zone morte lipschitzienne proposée par [33] et qui est illustrée à la figure 5.3b. Elle est donnée par :

$$\Delta\boldsymbol{\varepsilon} = \max\left(0, \min\left(1, \frac{\|\Delta\boldsymbol{\varepsilon}\| - \delta\varepsilon_0}{(1-\delta)\varepsilon_0}\right)\right)\Delta\boldsymbol{\varepsilon}. \tag{5.13}$$

Pour implémenter cette modification robuste, deux options sont possibles. Dans un premier cas, on prend la norme de l'erreur pour le contrôleur entier. Par exemple, pour le contrôleur d'altitude, on a $\|\Delta\boldsymbol{\varepsilon}\| = (z - z_{ref})^2 + (w - w_{ref})^2 + (e_z - e_{z,ref})^2 \leq \varepsilon_0$. Dans un second cas, on borne l'erreur pour chacun des états individuellement et l'on fixe cette erreur à zéro si elle est sous la borne choisie. Par exemple, pour le contrôleur d'altitude les trois modifications

sont :

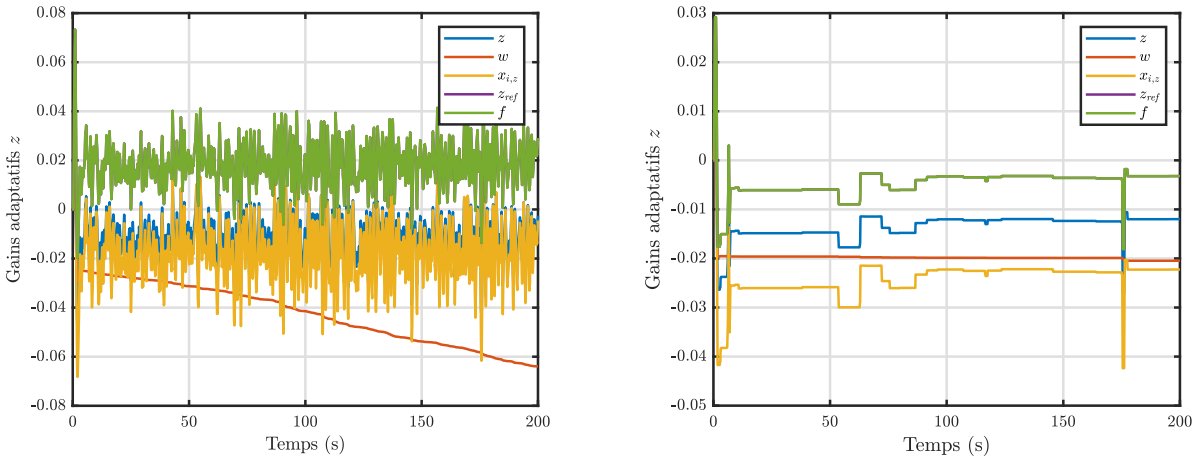
$$\begin{aligned}
\Delta\varepsilon_z &= \max\left(0,\ \min\left(1,\ \frac{\Delta\boldsymbol{\varepsilon}_z^2 - \delta\varepsilon_{0,z}}{(1-\delta)\varepsilon_{0,z}}\right)\right)\Delta\boldsymbol{\varepsilon}_z, \\
\Delta\varepsilon_w &= \max\left(0,\ \min\left(1,\ \frac{\Delta\boldsymbol{\varepsilon}_w^2 - \delta\varepsilon_{0,w}}{(1-\delta)\varepsilon_{0,w}}\right)\right)\Delta\boldsymbol{\varepsilon}_w, \\
\Delta\varepsilon_{e_z} &= \max\left(0,\ \min\left(1,\ \frac{\Delta\boldsymbol{\varepsilon}_{e_z}^2 - \delta\varepsilon_{0,e_z}}{(1-\delta)\varepsilon_{0,e_z}}\right)\right)\Delta\boldsymbol{\varepsilon}_{e_z},
\end{aligned} \tag{5.14}$$

où $\Delta\boldsymbol{\varepsilon}_z = z - z_{ref}$, $\Delta\boldsymbol{\varepsilon}_w = w - w_{ref}$ et $\Delta\boldsymbol{\varepsilon}_{x_{i,z}} = x_{i,z} - x_{i,z,ref}$. Cette méthode est plus intuitive, car elle limite chaque état individuellement, ce qui n'est pas le cas de la première méthode. De plus, si les états n'ont pas les mêmes ordres de grandeur, les limiter individuellement est plus efficace, car on évite une disparité entre les états qui causerait certains états à pouvoir varier plus que d'autres.

Toutefois, le défi majeur pour l'implémentation de la modification de la zone morte est qu'il faut déterminer *a priori* les bornes ε_0. Or, déterminer ces bornes n'est pas toujours une tâche aisée. Si elles sont choisies trop petites, alors le phénomène de dérive des gains est toujours présent. Cela se produit car les perturbations qui influencent le quadricoptère sont trop fortes et les bornes ne sont jamais atteintes. L'adaptation ne s'arrête jamais et l'appareil est encore susceptible à la dérive des gains.

Ceci dit, si la zone morte est bien construite, elle peut empêcher le phénomène de dérive des gains. Par exemple, dans le cas où les incertitudes sont dues à du bruit de mesure, il faut fixer les bornes ε_0 juste au-dessus du niveau du bruit. Le système sera alors capable d'atteindre ces bornes et le processus d'adaptation s'arrêtera empêchant ainsi la dérive des gains adaptatifs. La figure 5.4a donne les résultats d'une simulation pour les gains adaptatifs du contrôleur d'altitude en présence de bruit lorsque la modification de la zone morte n'est pas présente. La figure 5.4b donne la même simulation en présence de la modification de la zone morte lipschitzienne.

Ces deux simulations permettent de voir la grande variabilité des gains adaptatifs due au bruit. On remarque que le gain de l'état w est divergent. Si la simulation était prolongée, le gain continuerait cette dérive jusqu'à ce que le quadricoptère soit très réactif aux variations de w. Un changement brusque de w pourrait alors le déstabiliser en faisant saturer les moteurs et il pourrait s'écraser. Cependant, en utilisant la modification de la zone morte, on voit que les gains arrêtent d'évoluer et ne sont plus susceptibles au bruit.



(a) Simulation sans la modification de la zone morte (b) Simulation avec la modification de la zone morte

Figure 5.4 Simulations pour la modification robuste de la zone morte

On implémente maintenant la modification de la zone morte sur l'AscTec Pelican. D'abord, on mesure l'amplitude du bruit pour tous les états du quadricoptère. Ces mesures sont comptabilisées et les bornes ε_0 sont fixées juste au-dessus du niveau du bruit. Ces bornes sont données dans le tableau 5.1. La modification de la zone morte lipschitzienne est implémentée avec ces bornes sur le quadricoptère. La figure 5.5 donne les résultats d'un vol stationnaire.

Tableau 5.1 Bornes maximales sur le bruit des états affectant les contrôleurs

Contrôleur x		Contrôleur y		Contrôleur z		Contrôleur ψ	
x	± 0.00225 m	y	± 0.00225 m	z	± 0.0010 m	ψ	± 0.0007 rad
u	± 0.03000 m/s	v	± 0.03000 m/s	w	± 0.0125 m/s	r	± 0.0400 rad/s
θ	± 0.00070 rad	ϕ	± 0.00070 rad	$x_{i,z}$	± 0.0010 m	$x_{i,\psi}$	± 0.0007 rad
q	± 0.04000 rad/s	p	± 0.04000 rad/s				
$x_{i,x}$	± 0.00225 m	$x_{i,y}$	± 0.00225 m				

On voit que l'ajout de la modification de la zone morte n'a pas permis d'éviter le phénomène de dérive des gains. Les gains des états p, q, r et w sont toujours fortement influencés par le phénomène. Pour comprendre pourquoi la modification robuste échoue à stabiliser les gains adaptatifs, il faut s'intéresser aux composantes de l'AscTec Pelican.

Influence des moteurs sur le système physique : considérations pratiques

Pour qu'un multirotor puisse voler adéquatement, il est nécessaire que les moteurs soient parfaitement synchronisés et bien commandés. La composante qui effectue la commande

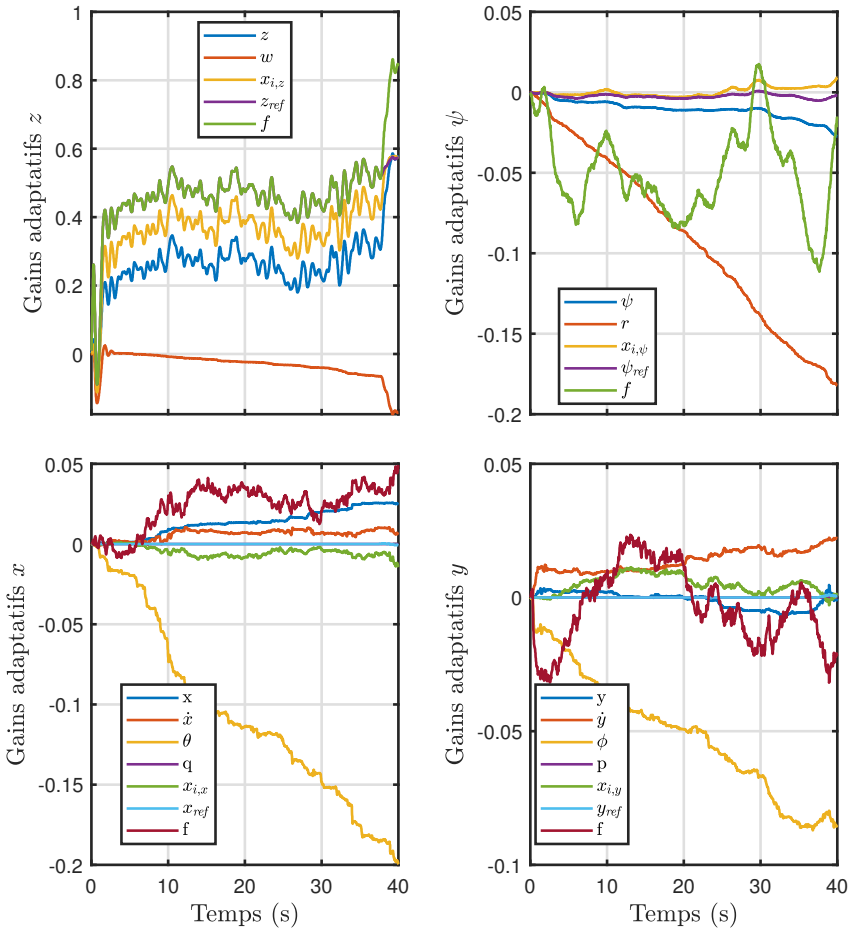


Figure 5.5 Tests de la zone morte avec des bornes légèrement supérieures au bruit

des moteurs est le contrôleur électronique de vitesse, *Electronic Speed Controller* (ECS). Généralement, ce contrôleur est fait par le commerçant et il n'est pas possible de le modifier. Pour l'AscTec Pelican cette composante est comme une boîte noire, i.e., nous ne pouvons pas modifier son code source.

Les moteurs peuvent tourner à une vitesse allant de 0 à 8000 rpm. Or, c'est le ECS qui commande cette vitesse de rotation à partir d'un nombre entier de 0 à 200. Ce faisant, les vitesses de rotation sont discrétisées par incréments de 40 rpm. L'intervalle discrétisé et non continu du ECS réduit la qualité de la commande, car il n'est pas toujours possible d'envoyer la vitesse de rotation exacte aux moteurs. Par exemple, si la valeur à envoyer est de 65 rpm, le ECS acceptera 1 ou 2 comme valeur d'entrée. Cela correspond à des vitesses de 40 rpm ou 80 rpm. Sans les valeurs exactes envoyées aux moteurs, le quadricoptère oscille autour de

son point d'équilibre. Ce balancement est non négligeable et il perturbe fortement les états de l'appareil.

Ainsi, la modification de la zone morte implémentée précédemment ne peut pas empêcher la dérive des gains, car ce balancement provoque des oscillations qui sont supérieures au bruit de mesure ; les bornes du tableau 5.1 sont trop restrictives. Cependant, si elles sont ajustées pour tenir compte du balancement du drone, la zone morte devient trop permissive. Le tableau 5.2 donne les valeurs de ces nouvelles bornes. On remarque qu'elles sont de plusieurs ordres de grandeur supérieures à celles utilisées précédemment. Avec ces bornes, la zone morte est si grande qu'elle nuit à l'adaptabilité du système. La figure 5.6 donne les résultats expérimentaux à la suite de la mise en œuvre de la modification de la zone morte construite avec ces nouvelles valeurs.

Tableau 5.2 Bornes maximales pour le balancement du quadricoptère

Contrôleur x		Contrôleur y		Contrôleur z		Contrôleur ψ	
x	$\pm$ 0.05 m	y	$\pm$ 0.05 m	z	$\pm$ 0.02 m	ψ	$\pm$ 0.05 rad
u	$\pm$ 0.2 m/s	v	$\pm$ 0.2 m/s	w	$\pm$ 0.2 m/s	r	$\pm$ 0.2 rad/s
θ	$\pm$ 0.05 rad	ϕ	$\pm$ 0.05 rad	$x_{i,z}$	$\pm$ 0.02 m	$ex_{i,\psi}$	$\pm$ 0.05 rad
q	$\pm$ 0.2 rad/s	p	$\pm$ 0.2 rad/s				
$x_{i,x}$	$\pm$ 0.05 m	$x_{i,y}$	$\pm$ 0.05 m				

On voit d'abord que la modification ne permet pas de borner tous les gains, e.g., pour les états w, r et θ. Cette grande zone morte ne donne pas suffisamment de temps au système pour s'adapter et il n'arrive pas à converger vers le modèle de référence. On remarque également la présence d'oscillations à haute fréquence pour le contrôleur en z, malgré le fait que la modification soit lipschitzienne. En se référant à la figure 5.3b, on voit que si les transitions entre $\delta\varepsilon_0$ et ε_0 se font trop rapidement, cela peut provoquer des oscillations. Avec une zone morte aussi grande, le quadricoptère passe trop rapidement et fréquemment par cette frontière ce qui mène à l'effet de broutement.

On voit donc que la modification de la zone morte n'est pas idéale pour rendre le système robuste aux incertitudes non paramétriques mal-assorties à cause des contraintes matérielles de l'AscTec Pelican. En effet, si la zone morte est trop restrictive, elle n'empêche pas le phénomène de dérive des gains. Par contre, si elle est construite en fonction des contraintes matérielles du quadricoptère, elle nuit à l'adaptabilité du système. Elle va alors à l'encontre de la MRAC en définissant une trop grande zone où il n'y a pas d'adaptation.

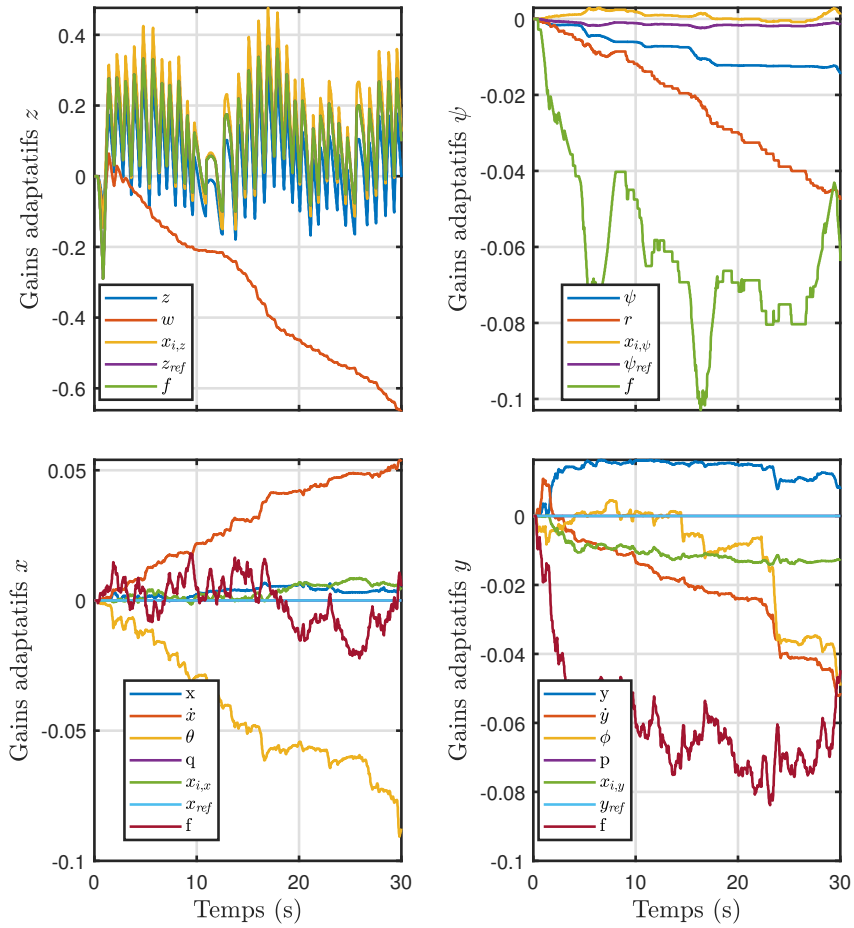


Figure 5.6 Tests de la zone morte avec des bornes au-delà du balancement naturel

5.3 Modification–σ

La modification robuste de la zone morte possède deux inconvénients majeurs. D'abord, dans le cas où le système est très bruité, elle a de la difficulté à empêcher le phénomène de dérive des gains. L'autre inconvénient est qu'il est nécessaire de connaître *a priori* les bornes à imposer aux différents états. Ces inconvénients ont poussé [34] à proposer une autre modification robuste, la modification–σ. Cette modification modifie la loi d'adaptation originale en ajoutant un facteur d'oubli $-\sigma\Delta\hat{\boldsymbol{\theta}}$ à la dynamique des gains adaptatifs :

$$\Delta\dot{\hat{\boldsymbol{\theta}}} = -\left(\boldsymbol{\Gamma}_\theta \boldsymbol{\omega} \Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{B} - \sigma\Delta\hat{\boldsymbol{\theta}}\right) \tag{5.15}$$

avec σ est une constante à déterminer. La modification–σ empêche le phénomène de dérive des gains tout en retirant la nécessité de connaître les bornes *a priori*. Il suffit alors de fixer la

constante σ, qui définit la force du facteur d'oubli. Plus cette constante est élevée et plus les gains se stabilisent rapidement. Pour une preuve de stabilité de cette modification robuste, le lecteur peut se référer à [23, Chapitre 11.2].

Cette modification robuste est implémentée de la même manière que la commande nominale en la divisant selon les quatre sous-systèmes. Il y aura donc quatre modifications–σ et quatre constantes d'oubli σ_x, σ_y, σ_z et σ_ψ. Chacun des facteurs d'oubli agit sur plusieurs états simultanément. Par exemple, σ_x est le facteur d'oubli pour les gains associés aux états x, u, θ, q, $x_{i,x}$, x_{ref} et f_x.

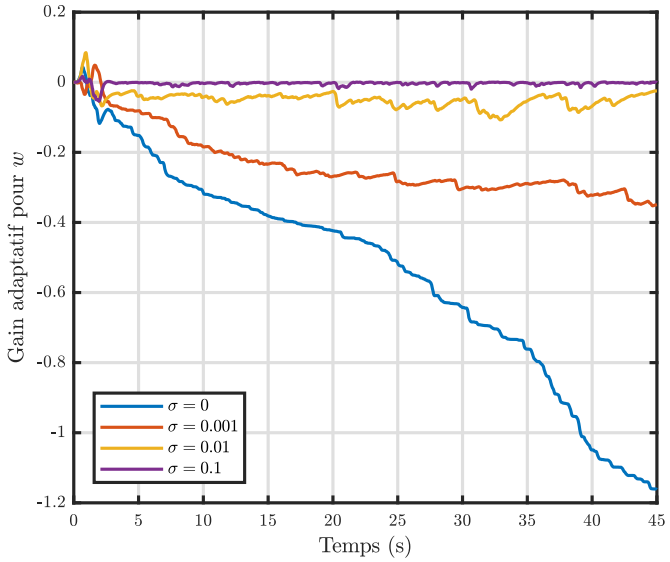


Figure 5.7 Simulation pour σ croissant pour le gain associé à l'état w

Précédemment, nous avons montré en simulation que le gain de l'état w était affecté par le phénomène de dérive des gains (Fig. 5.4a). Pour les simulations de la modifications–σ, nous ne considérons que ce gain adaptatif. Les simulations de la figure 5.7 sont effectuées pour quatre valeurs du facteur d'oubli, soit $\sigma = \{0,\ 0.001,\ 0.01,\ 0.1\}$. Si la modification–$\sigma$ n'est pas active, i.e., si le facteur d'oubli est nul, alors le gain diverge. Toutefois, dès que le facteur d'oubli est non nul, le gain se stabilise. On remarque aussi que plus le facteur d'oubli est élevé, plus le gain reste près de l'origine. Ainsi, lorsque $\sigma = 0.1$, le gain adaptatif ne varie presque plus. Si la valeur du facteur d'oubli est choisie adéquatement, il est possible d'empêcher la dérive des gains, sans pour autant limiter l'intervalle de variation du gain adaptatif.

Les tests expérimentaux précédents ont montré des divergences significatives pour les gains des états ϕ et θ. Après avoir implémenté la modification–σ sur l'AscTec Pelican, nous consi-

dérons ces gains en particulier pour la suite de l'analyse. La figure 5.8 donne les résultats d'un vol stationnaire du quadricoptère où une panne de moteur de 20% survient à $t = 2$ s. Les résultats montrent que pour un facteur d'oubli $\sigma < 0.1$, la dérive des gains affecte encore les deux états. Pourtant, si la modification–σ est suffisamment forte avec $\sigma = 0.1$, les gains sont stabilisés. Ainsi, cette modification robuste est capable d'empêcher le phénomène de dérive des gains. Cependant, elle possède plusieurs inconvénients qui la rendent inutilisable.

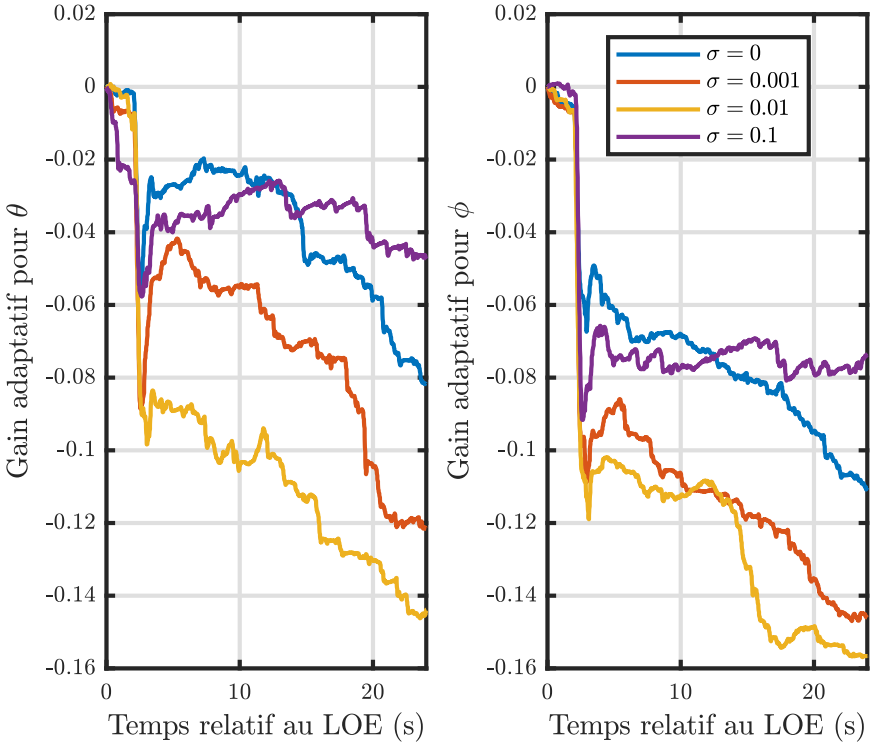


Figure 5.8 Stabilisation des gains

Inconvénients de la modification–σ

Le premier inconvénient de la modification–σ est celui de l'oubli de l'apprentissage. Si l'erreur $\Delta\boldsymbol{\varepsilon}$ entre le modèle de référence et le modèle réel est suffisamment petite, alors la loi adaptative de l'équation 5.15 peut être approximée par :

$$\Delta\dot{\hat{\boldsymbol{\theta}}} \approx -\sigma\Delta\hat{\boldsymbol{\theta}}. \tag{5.16}$$

Les gains se comporteront donc comme une fonction exponentielle décroissante vers l'origine. En quelque sorte, le système désapprend ses adaptations passées. Pour illustrer ce comporte-

ment, nous avons considéré le gain de l'état w à la suite d'une panne de moteur de 20%. La figure 5.9 montre les résultats obtenus pour différentes valeurs du facteur d'oubli. Dès que la modification–σ est active, le système commence à oublier son adaptation. Plus le facteur d'oubli est fort, plus l'effet est prononcé. Par exemple, quand $\sigma = 0.01$ il faut 8 secondes au système pour oublier son apprentissage, tandis qu'il faut 1 seconde lorsque $\sigma = 0.1$. Ainsi, si la modification est trop forte, elle va nuire à l'objectif de la MRAC, car le système désapprendra ce à quoi il vient de s'adapter.

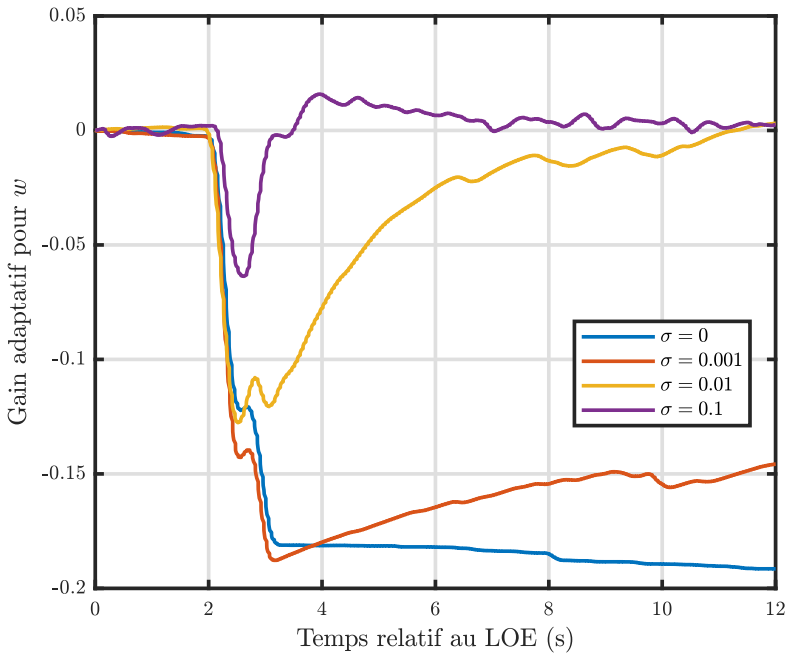


Figure 5.9 Phénomène d'oubli provoqué par la modification–σ

Le deuxième inconvénient de la modification–σ est qu'elle dégrade les performances lorsque le facteur d'oubli σ est trop fort. La figure 5.10 donne les cinq gains adaptatifs pour le contrôleur d'altitude du quadricoptère en fonction du facteur d'oubli. Le brusque changement des gains est attribuable à un événement de panne de moteur de 20%. Les résultats démontrent bien une limitation du processus d'adaptation en fonction de la valeur de σ. Pour $\sigma = 0.001$ l'évolution des gains adaptatifs n'est pas influencée par la modification robuste, car elle est similaire au cas où la modification n'est pas active. Toutefois, pour $\sigma = 0.01$, on constate qu'à partir de $t = 13$ s, les gains sont ne sont jamais supérieurs à 0.8. Pour les facteurs d'oubli moins forts, les gains pouvaient varier jusqu'à 1.0. Lorsque $\sigma = 0.1$, les gains sont limités à l'intervalle $[0.2,\ 0.4]$, une nette dégradation par rapport aux cas précédents.

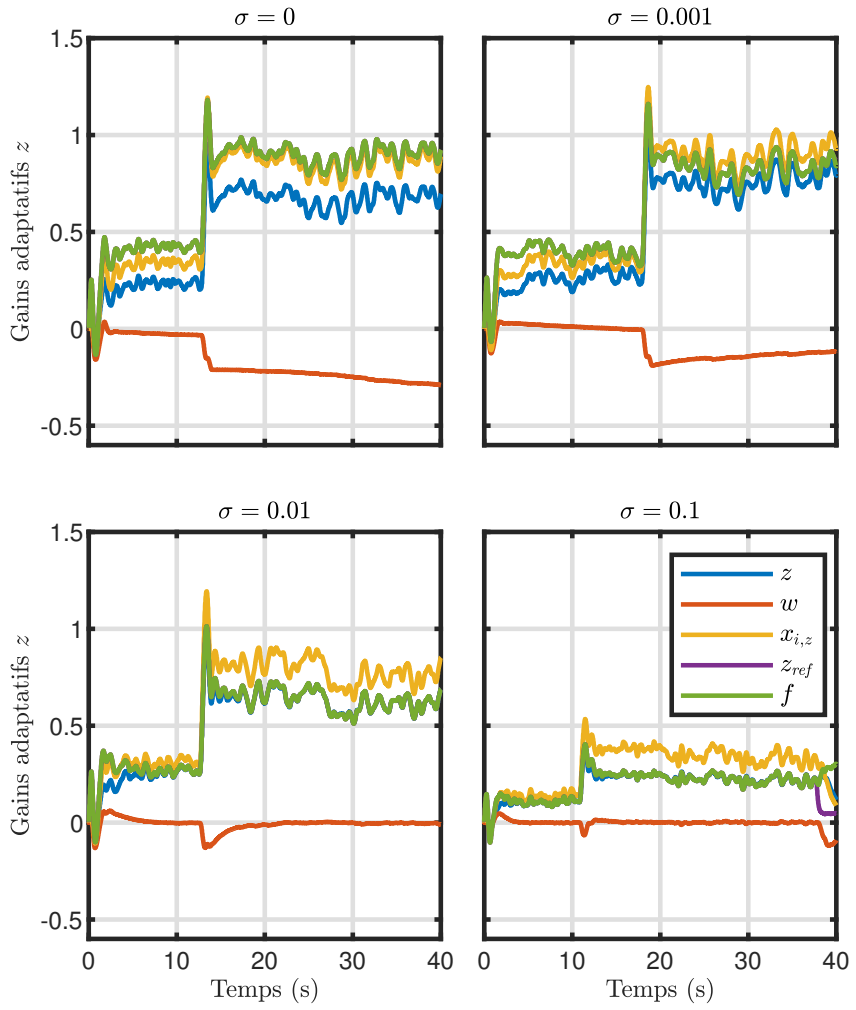


Figure 5.10 Limitation du processus d'adaptation

Une restriction des gains à un intervalle de variation plus petit cause nécessairement une dégradation des performances du quadricoptère. On peut constater cela en analysant la chute en altitude due à la panne de moteur. La figure 5.11 montre bien que le facteur d'oubli nuit à la MRAC dans son adaptation à la perturbation. En effet pour $\sigma = 0.01$, on voit que le quadricoptère prend 3 secondes supplémentaires à atteindre son point d'équilibre. Lorsque $\sigma = 0.1$, les performances du quadricoptère sont significativement dégradées. Le quadricoptère chute de 2 cm supplémentaires et retourne à son point d'équilibre aussi lentement que s'il était commandé par le LQR (Fig. 4.14a). Si la modification–σ est trop forte, les performances du quadricoptère se dégraderont et la MRAC devient inefficace.

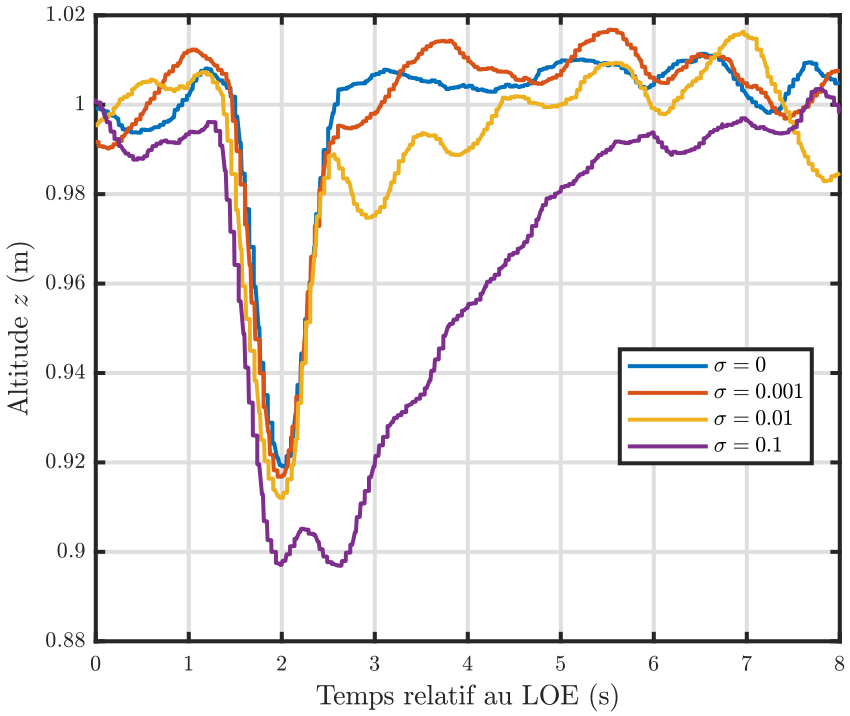


Figure 5.11 Dégradation des performances

Le troisième inconvénient de cette modification robuste est qu'il n'est pas possible de connaître précisément la valeur à laquelle les gains adaptatifs se stabiliseront. Même si la preuve de stabilité [23, Chapitre 11.2] démontre que tous les signaux du système sont UUB, il est impossible de connaître la borne des gains. Si cette borne s'avère être trop élevée, le système peut se déstabiliser. Or, pour déterminer cette borne, il faut connaître l'amplitude des perturbations non paramétriques mal-assorties qui affecteront le système. Cependant, nous ne pouvons pas connaître *a priori* cette information.

Le dernier inconvénient de la modification–σ concerne son implémentation. En effet, le modification robuste affecte le contrôleur globalement et non chacun des états individuellement. Pourtant, ce ne sont pas tous les états qui nécessitent la même valeur du facteur d'oubli (voir l'annexe C.3 pour s'en convaincre). C'est donc l'état qui doit avoir le facteur d'oubli le plus fort qui dictera la force de la modification–σ pour le contrôleur entier. Ainsi, certains états seront trop limités et cela impactera les performances du quadricoptère. La modification–σ souffre d'un déséquilibre de force lors de son implémentation dans une loi de commande. Toutes ces raisons font en sorte que même si la modification–σ donne de bons résultats lors

des simulations, nous ne recommandons pas son implémentation sur un quadricoptère.

5.4 Modification–ϵ

Pour pallier les problèmes de la modification–σ, les auteurs de [36] ont proposé la modification–ϵ. Ils ont modifié sa formulation mathématique pour que la variable d'oubli ne soit plus une constante fixe, mais qu'elle dépende également de l'erreur $\Delta\boldsymbol{\varepsilon}$ entre le modèle de référence et le système réel. La modification–ϵ est exprimée comme :

$$\Delta\dot{\hat{\boldsymbol{\theta}}} = -\left(\boldsymbol{\Gamma}_\theta \boldsymbol{\omega} \Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{B} - \sigma \left\|\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{B}\right\| \Delta\hat{\boldsymbol{\theta}}\right). \tag{5.17}$$

Le lecteur peut se référer à [23, Chapitre 11.3] pour une preuve de stabilité pour cette modification robuste, qui démontre que tous les signaux sont UUB. De plus, grâce au facteur $\left\|\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{B}\right\|$, le phénomène d'oubli de l'apprentissage ne peut plus se produire. En effet, lorsque $\Delta\boldsymbol{\varepsilon} \to \mathbf{0}$, on obtient l'approximation $\Delta\dot{\hat{\boldsymbol{\theta}}} \approx \mathbf{0}$. Donc, si l'erreur devient petite, les gains cesseront d'évoluer et ils se stabiliseront sans retourner à l'origine. Comme la modification–σ, la modification–ϵ est implémentée sur chacun des quatre sous-systèmes et il y a quatre constantes à déterminer.

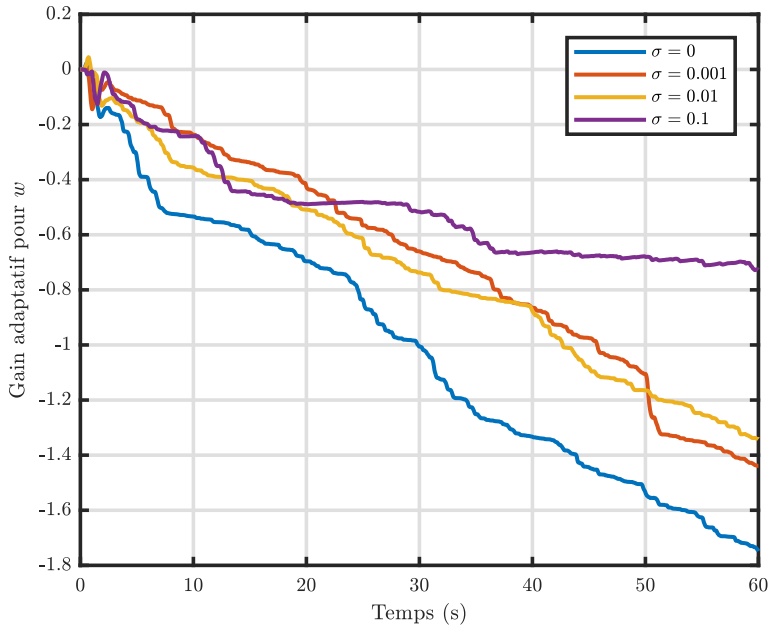
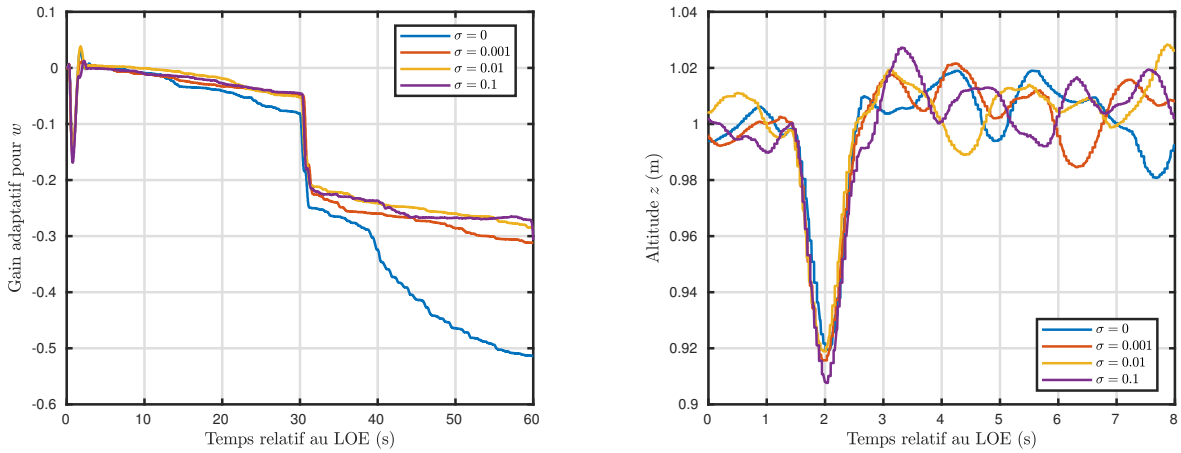


Figure 5.12 Simulations de la modification–ϵ pour le gain adaptatif associé à l'état w

On effectue d'abord des simulations de cette modification pour le gain de l'état w. La figure

5.12 montre les résultats obtenus pour $\sigma = \{0,\ 0.001,\ 0.01,\ 0.1\}$. On constate que seul $\sigma = 0.1$ stabilise le gain adéquatement. Si l'on compare avec les simulations pour la modification–σ, on voit que le phénomène d'oubli de l'adaptation n'est présent pour aucune des simulations. De plus, les gains peuvent varier sur un intervalle suffisamment grand avant d'être bornés.



(a) Résultats expérimentaux de modification–ϵ pour le gain w

(b) Performance du quadricoptère en altitude avec la modification–ϵ

Figure 5.13 Simulations pour la modification robuste de la zone morte

Pour les tests expérimentaux, on effectue des vols avec une panne de moteur de 20% survenant à $t = 30$ s. La longue durée de vol permet une dérive des gains significative. La LOE est utilisée pour vérifier si les performances du système sont affectées par la présence de la modification–ϵ. La figure 5.13a illustre les résultats de vol pour le gain associé à l'état w. On remarque d'abord que la modification robuste est meilleure qu'anticipée, car même les plus petites valeurs de σ empêche la dérive du gain. La descente rapide à $t = 30$ s permet également de constater que la modification–ϵ ne réduit pas l'adaptabilité du système. Mis à part le décalage avec la courbe de $\sigma = 0$, la LOE provoque la même variation du gain adaptatif pour les quatre vols. Ainsi, les performances du système ne devraient pas être affectées.

La figure 5.13b montre l'altitude du quadricoptère durant les quatre vols. On remarque qu'à l'exception de $\sigma = 0.1$, tous les vols ont la même chute en altitude causée par la panne de moteur. On note toutefois une faible dégradation des performances pour le cas où $\sigma = 0.1$. En effet, le quadricoptère descend d'un centimètre supplémentaire ce qui correspond à 12.5% d'écart avec la valeur des vols. Cela dit, l'impact de la modification–ϵ sur les performances du quadricoptère en vol est considéré négligeable. La modification–ϵ est une nette amélioration par rapport à la modification–σ, malgré ses quelques inconvénients.

Inconvénients de la modification–ϵ

Tout comme la modification–σ, le premier inconvénient de la modification–ϵ est qu'elle est imprévisible. Il n'y a aucune garantie qu'une certaine valeur de σ arrivera à stabiliser tous les gains simultanément. Par exemple, la figure 5.14 montre que le gain associé à l'état r n'est stabilisé par aucune des valeurs de $\sigma = \{0.001,\ 0.01,\ 0.1\}$. Il faudrait choisir une valeur plus élevée. Or, nous avons montré que la modification–ϵ commence à affecter les performances du quadricoptère lorsque $\sigma > 0.1$. Pour le contrôleur de la rotation en lacet, une valeur permettant de borner le gain de l'état r impacterait négativement ses performances. L'annexe C.3.1 contient d'autres exemple où $\sigma = 0.1$ n'est pas suffisant pour borner tous les gains.

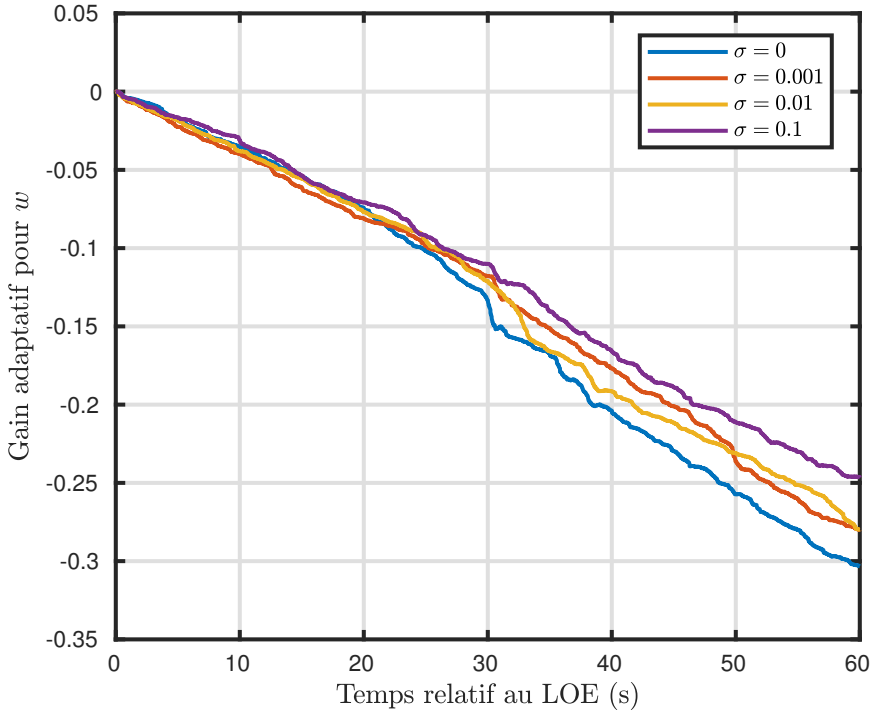


Figure 5.14 Résultats expérimentaux avec la modification–ϵ pour le gain adaptatif associé à l'état r

Un autre désavantage provient du fait que la modification–ϵ affecte l'ensemble d'un contrôleur et non les états individuels. Les conséquences de cela sont les mêmes que pour la modification–σ. Avec une seule valeur du facteur d'oubli par contrôleur, il n'est pas possible qu'elle soit optimale pour tous les états. Bien que cette modification permette d'être robuste aux incertitudes non paramétriques mal-assorties, elle nuit à l'objectif de la commande adaptative en réduisant la vitesse d'adaptation. De plus, son implémentation non idéale réduit son efficacité.

5.5 Conclusion du chapitre

Dans la première partie de ce chapitre, nous avons étudié les effets des perturbations non paramétriques mal-assorties sur le système. Nous avons montré que dans ces conditions, l'adaptation ne s'arrête jamais. Cela cause le phénomène de dérive des gains adaptatifs. Si ce problème n'est pas adressé, les gains peuvent diverger et causer l'instabilité, menant éventuellement à l'écrasement du quadricoptère.

Afin d'empêcher ce phénomène, nous avons étudié trois modifications robustes dites traditionnelles. Or, ces méthodes procèdent de façon indirecte, car elle n'agissent pas sur les gains adaptatifs. En effet, elles arrêtent le processus d'adaptation lorsque l'erreur entre le modèle de référence et le système réel est suffisamment petite. Ces modifications fonctionnent adéquatement en simulation, mais elles présentent plusieurs inconvénients tant dans leur implémentation que lorsqu'elles sont déployées sur l'AscTec Pelican.

Le premier inconvénient est que ces méthodes agissent sur le contrôleur global plutôt que sur chacun des gains individuellement. Cela rend leur ajustement difficile et cause des problèmes de performance en vol. Le second inconvénient est que ces méthodes agissent à l'encontre de la MRAC, car elles réduisent l'adaptabilité du système. Pour ces raisons, nous ne recommandons pas l'utilisation de ces modifications robustes pour compenser les effets des perturbations non paramétriques mal-assorties. Nous devons donc déterminer une modification robuste alternative pour compenser les incertitudes non paramétriques mal-assorties.

CHAPITRE 6 OPÉRATEUR DE PROJECTION

6.1 Introduction

L'opérateur de projection est une modification robuste développée pour compenser les inconvénients des modifications robustes traditionnelles. L'opérateur de projection est conçu pour agir de directement sur les gains en leur imposant des bornes précises. Cela retire l'imprévisibilité des modifications robustes précédentes. De plus, tant que les gains adaptatifs n'ont pas atteint les bornes de l'opérateur de projection, ils peuvent varier librement. En procédant de cette façon, les performances du quadricoptère ne sont pas affectés par la modification robuste. Nous présentons l'opérateur de projection et les notions mathématiques nécessaires pour le définir à la section 6.2.

Cependant, l'opérateur de projection fait intervenir un grand nombre de paramètres ajustables qui sont les bornes des différents gains adaptatifs du système. Obtenir ces bornes est une tâche difficile. En effet, elles ne peuvent pas être choisies arbitrairement, car cela aura des impacts négatifs sur les performances du quadricoptère. Il n'existe aucune méthode systématique pour les obtenir. À la section 6.3, nous présentons un algorithme qui obtient les bornes de façon systématique. Cet algorithme combine plusieurs notions de la commande robuste afin d'obtenir les bornes des gains adaptatifs.

Enfin, nous présenterons les résultats de l'algorithme à la section 6.4. Nous effectuerons également une preuve de concept en déployant l'opérateur de projection sur l'AscTec Pelican et en effectuant des tests en laboratoire. Nous montrerons alors que cette modification robuste permet de bien rejeter les incertitudes non paramétriques mal-assorties et qu'elle est plus performante que les méthodes traditionnelles.

6.2 Présentation de l'opérateur de projection

6.2.1 Concepts et définitions de base

Les fonctions et les ensembles convexes sont les éléments de base qui sont utilisés pour définir l'opérateur de projection. Nous procéderons ici à quelques rappels importants concernant ces notions.

Définition 6.2.1. *Un sous-ensemble $\Omega \subset \mathbb{R}^n$ est convexe si*

$$\forall \mathbf{x}, \mathbf{y} \in \Omega,\ \forall \lambda \in [0,1] \Rightarrow \lambda \mathbf{x} + (1-\lambda)\mathbf{y} \in \Omega. \tag{6.1}$$

Ainsi, si une ligne est tracée entre deux points appartenant à l'ensemble convexe, alors tous les points de la ligne figurent aussi dans l'ensemble convexe.

Définition 6.2.2. *Une fonction $f : \mathbb{R}^n \to \mathbb{R}$ est convexe si*

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n,\ \forall \lambda \in [0,1] \Rightarrow f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}). \tag{6.2}$$

Ainsi, si deux points d'une fonction convexe sont reliés par une droite, alors tous les points de la fonction convexes entre ces deux extrêmes doivent se situer sous cette droite.

Il est également possible de combiner les notions de fonctions et d'ensembles convexes. Le lemme suivant est issu de ce concept et il servira dans la construction de l'opérateur de projection.

Lemme 6.2.1. *Soit $f : \mathbb{R}^n \to \mathbb{R}$, une fonction convexe. Alors, pour toute constante $\delta > 0$, le sous-ensemble $\Omega_\delta = \{\boldsymbol{\theta} \in \mathbb{R}^n \,|\, f(\boldsymbol{\theta}) \leq \delta\}$ est convexe.*

Le lecteur peut se référer à l'annexe D.1 pour la démonstration de ce lemme.

6.2.2 Opérateur de projection

Soit $f : \mathbb{R}^n \to \mathbb{R}$, une fonction convexe. On l'utilise pour définir, avec le lemme 6.2.1, deux sous-ensembles convexes Ω_0 et Ω_1 :

$$\begin{aligned} \Omega_0 &= \{\boldsymbol{\theta} \in \mathbb{R}^n \,|\, f(\boldsymbol{\theta}) \leq 0\}, \\ \Omega_1 &= \{\boldsymbol{\theta} \in \mathbb{R}^n \,|\, f(\boldsymbol{\theta}) \leq 1\}. \end{aligned} \tag{6.3}$$

Alors, $\Omega_0 \subset \Omega_1$.

Définition 6.2.3. *L'opérateur de projection de deux vecteurs* $\boldsymbol{\theta}, \mathbf{y} \in \mathbb{R}^n$ *est exprimé comme :*

$$\text{Proj}\,(\boldsymbol{\theta}, \mathbf{y}) := \begin{cases} \mathbf{y} - \dfrac{\nabla f(\boldsymbol{\theta})(\nabla f(\boldsymbol{\theta}))^\top}{\|\nabla f(\boldsymbol{\theta})\|^2}\mathbf{y} f(\boldsymbol{\theta}) & \text{si } f(\boldsymbol{\theta}) > 0 \textit{ et } \mathbf{y}^\top \nabla f(\boldsymbol{\theta}) > 0, \\ \mathbf{y} & \text{sinon} \end{cases} \tag{6.4}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ *est une fonction convexe et* $\nabla f(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_1} & \cdots & \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_n} \end{bmatrix}^\top$.

La figure 6.1 représente le fonctionnement de l'opérateur de projection en fonction des ensemble Ω_0 et Ω_1.

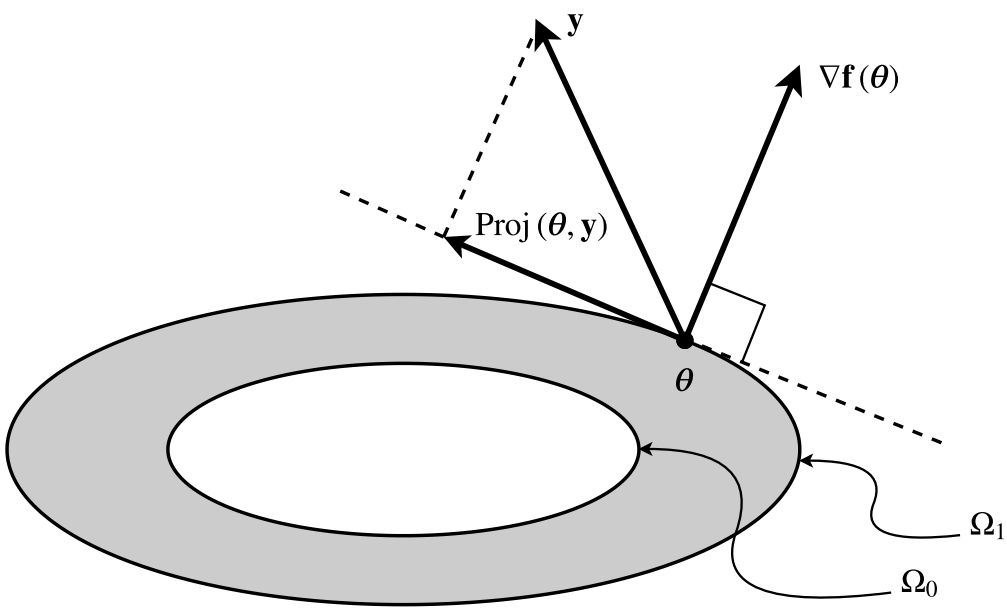


Figure 6.1 L'opérateur de projection

Intuitivement, l'opérateur de projection modifie le vecteur $\mathbf{y}$ de la façon suivante :

1. Si $\boldsymbol{\theta}$ se situe dans Ω_0, alors $\mathbf{y}$ n'est pas modifié par l'opérateur.
2. Si $\boldsymbol{\theta}$ se situe entre Ω_0 et Ω_1, i.e., si $0 < f(\boldsymbol{\theta}) \leq 1$, alors il y a deux possibilités :
   (a) Si $\mathbf{y}$ pointe vers l'intérieur de Ω_1, alors $\mathbf{y}$ n'est pas modifié par l'opérateur.
   (b) Si $\mathbf{y}$ pointe vers l'extérieur de Ω_1, alors l'opérateur de projection le corrige pour le garder à l'intérieur de Ω_1. Ultimement, si $\boldsymbol{\theta}$ est sur la frontière où $f(\boldsymbol{\theta}) = 1$, le vecteur de projection corrige $\mathbf{y}$ pour qu'il soit tangent à cette frontière. Ainsi, $\boldsymbol{\theta}$ ne quittera jamais l'ensemble Ω_1.

Deux lemmes supplémentaires sont nécessaires pour démontrer, dans une section subséquente, la stabilité de la MRAC synthétisée avec l'opérateur de projection.

Lemme 6.2.2. *Soit* $0 < \delta < 1$ *et le sous-ensemble* $\Omega_\delta = \{\boldsymbol{\theta} \in \mathbb{R}^n \,|\, f(\boldsymbol{\theta}) \leq \delta\}$. *Soit* $\boldsymbol{\theta}^* \in \Omega_\delta$ *tel que* $f(\boldsymbol{\theta}^*) < \delta$, *i.e.,* $\boldsymbol{\theta}^*$ *est un point intérieur de* Ω_δ *ne se situant pas sur la frontière. Alors, pour tout* $\boldsymbol{\theta} \in \Omega_1$ *nous avons l'inégalité suivante :*

$$(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top (\mathrm{Proj}\,(\boldsymbol{\theta}, \mathbf{y}) - \mathbf{y}) \leq 0. \tag{6.5}$$

Lemme 6.2.3. *Soit* $f : \mathbb{R}^n \to \mathbb{R}$, *une fonction convexe et continûment différentiable. En utilisant la définition de l'opérateur de projection* (6.4), *on considère la dynamique d'un vecteur* $\boldsymbol{\theta} \in \mathbb{R}^n$ *telle que donnée par*

$$\dot{\boldsymbol{\theta}} = \mathrm{Proj}\,(\boldsymbol{\theta}, \mathbf{y}(t))\,, \tag{6.6}$$

où $\boldsymbol{\theta}$ *représente les états d'un système dynamique quelconque et* $\mathbf{y} \in \mathbb{R}^n$ *est une fonction vectorielle variant dans le temps et continue par morceaux. Alors* Ω_1 *est un invariant.*

Les démonstrations pour ces lemmes sont données à l'annexe D.2. Dans les définitions et les lemmes précédents, nous utilisons une fonction $f(\boldsymbol{\theta})$. Cette fonction est essentielle pour la construction de l'opérateur de projection, car elle dicte la façon dont le vecteur $\boldsymbol{\theta}$ est borné. Nous rappelons qu'elle doit être convexe et continûment différentiable. De façon générale [23, 51–53], cette fonction prend deux formes :

$$f(\boldsymbol{\theta}) = \frac{(1+\epsilon)\,\boldsymbol{\theta}^\top\boldsymbol{\theta} - \|\boldsymbol{\theta}_{max}\|^2}{\epsilon\,\|\boldsymbol{\theta}_{max}\|^2} \quad \text{ou} \quad f(\boldsymbol{\theta}) = \frac{1+\epsilon}{\epsilon}\left(\frac{\theta_1^2}{\theta_{1,max}^2} + \cdots + \frac{\theta_n^2}{\theta_{n,max}^2}\right) - \frac{1}{\epsilon} \tag{6.7}$$

La première forme borne tout le vecteur $\boldsymbol{\theta}$ simultanément grâce à la constante $\left\|\boldsymbol{\theta}_{max}^2\right\|$. La seconde forme borne chacun des éléments de $\boldsymbol{\theta}$ individuellement grâce aux constantes $\{\theta_{1,max}^2, \ldots, \theta_{n,max}^2\}$.

6.2.3 Exemple d'utilisation

On considère la modélisation d'ordre 1 de la dynamique d'un moteur :

$$\tau\dot{\omega} + \omega = K_m V_{in}, \tag{6.8}$$

où τ est la constante de temps du moteur, ω sa vitesse angulaire, K_m le gain statique du moteur et V_{in} la tension d'entrée. On suppose que les valeurs nominales des constantes sont

$\tau = 1$ s et $K_m = 1$ rad/s/V. On conçoit une commande de base par retour d'état $V_{in,bl} = -k\omega + \omega_{ref}$ afin que le système nominal en boucle fermée corresponde à $T(s) = \Omega(s)/\Omega_{ref}(s) = 1/(s+2)$. Pour respecter cette contrainte, le gain du retour d'état doit être $k = 1$.

Pour cet exemple, on considère la présence d'incertitudes paramétriques. On synthétise donc une commande adaptative afin de compenser ces incertitudes. La commande a la forme $V_{in,ad} = \hat{k}\omega$. On définit l'erreur $\epsilon = \omega - \omega_m$ où ω_m est la vitesse de rotation du modèle de référence donné par $T(s)$. Pour assurer la stabilité du système, la loi d'adaptation est donnée par

$$\dot{\hat{k}} = -\gamma\omega\varepsilon, \tag{6.9}$$

où γ représente la force de la commande adaptative qui sera fixée à 20 dans cet exemple.

Afin de limiter l'effort de commande de la partie adaptative, nous souhaitons définir un point de saturation pour le gain adaptatif. Nous choisissons la limite $|\hat{k}| \leq 1$. On utilise donc l'opérateur de projection pour redéfinir la dynamique de la loi d'adaptation. Pour cela, on utilise la fonction convexe suivante :

$$f(\hat{k}) = \frac{1+\epsilon}{\epsilon}\left(\frac{\hat{k}^2}{\hat{k}_{max}^2}\right) - \frac{1}{\epsilon}, \tag{6.10}$$

où $\hat{k}_{max} = 1$ et $\epsilon = 0.5$. Cela nous permet de définir les deux ensembles de base pour l'opérateur de projection.

$$\begin{aligned} \Omega_0 &= \{\hat{k} \in \mathbb{R} | f(\hat{k}) \leq 0\} = \{\hat{k} \in \mathbb{R} |\ |\hat{k}| \leq 1\}, \\ \Omega_1 &= \{\hat{k} \in \mathbb{R} | f(\hat{k}) \leq 1\} = \{\hat{k} \in \mathbb{R} |\ |\hat{k}| \leq 0.816\}. \end{aligned} \tag{6.11}$$

Ainsi, l'opérateur de projection commencera à limiter le gain adaptatif dès que celui-ci dépassera la valeur de $\hat{k} = 0.816$ et le gain ne dépassera jamais la valeur de $\hat{k} = 1$. Comme perturbation paramétrique, la valeur de τ est modifiée de 1 s à 4 s. La figure 6.2 donne l'évolution du gain adaptatif $\hat{k}$ pour une loi d'adaptation synthétisée avec et sans l'opérateur de projection. On remarque que sans l'opérateur de projection, le gain adaptatif peut évoluer librement jusqu'à atteindre la valeur idéale de $\hat{k} = 3$. Avec l'opérateur, le gain est bien limité à 1. De plus, l'opérateur de projection commencer à modifier l'évolution du gain dès qu'il dépasse la valeur critique de 0.816.

Les gains adaptatifs seront bornés de la même façon pour la MRAC augmentée. Or, avant de pouvoir appliquer l'opérateur de projection, il faut être en mesure d'obtenir les bornes optimales pour ces gains adaptatifs. Les prochaines sous-sections serviront à définir une

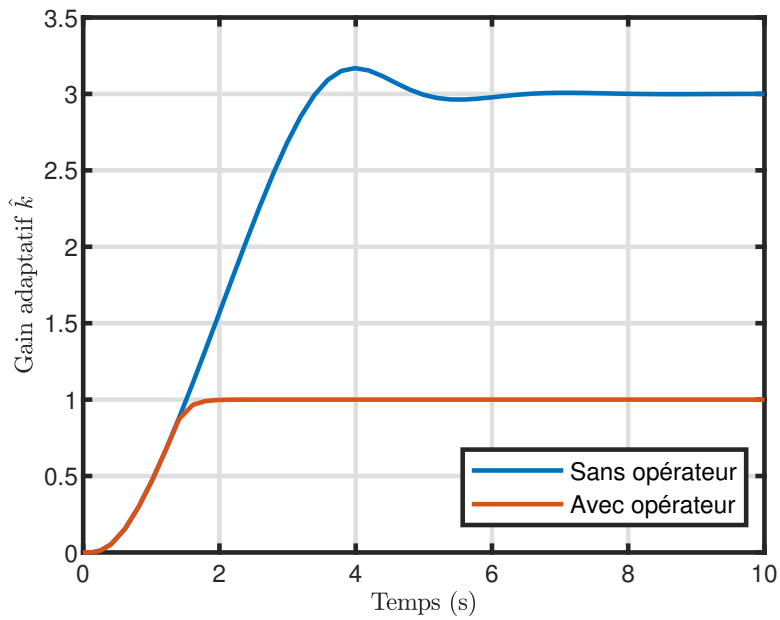


Figure 6.2 Simulation pour l'exemple illustratif de l'opérateur de projection

méthode pour l'obtention des gains de façon systématique et rigoureuse.

6.2.4 Analyse de la stabilité de la MRAC construite avec l'opérateur de projection

Avant de procéder à l'analyse de stabilité de la MRAC avec l'opération de projection, on présente d'abord une variante de l'opérateur, la projection–$\boldsymbol{\Gamma}$.

Définition 6.2.4. *L'opérateur de projection–$\boldsymbol{\Gamma}$ de deux vecteurs $\boldsymbol{\theta}, \mathbf{y} \in \mathbb{R}^n$ effectue l'adaptation des paramètres selon une matrice $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times n}$ symétrique et définie positive :*

$$\mathrm{Proj}_{\boldsymbol{\Gamma}}(\boldsymbol{\theta}, \mathbf{y}) := \begin{cases} \boldsymbol{\Gamma}\mathbf{y} - \boldsymbol{\Gamma}\dfrac{\nabla f(\boldsymbol{\theta})(\nabla f(\boldsymbol{\theta}))^\top}{(\nabla f(\boldsymbol{\theta}))^\top \boldsymbol{\Gamma} \nabla f(\boldsymbol{\theta})}\boldsymbol{\Gamma}\mathbf{y} f(\boldsymbol{\theta}) & \text{si } f(\boldsymbol{\theta}) > 0 \text{ et } \mathbf{y}^\top \boldsymbol{\Gamma} \nabla f(\boldsymbol{\theta}) > 0, \\ \boldsymbol{\Gamma}\mathbf{y} & \text{sinon} \end{cases} \tag{6.12}$$

Dans ce cas-là, le pendant du lemme 6.2.2 devient :

Lemme 6.2.4. *Soit $0 < \delta < 1$ et le sous-ensemble $\Omega_\delta = \{\boldsymbol{\theta} \in \mathbb{R}^n \,|\, f(\boldsymbol{\theta}) \leq \delta\}$. Soit $\boldsymbol{\theta}^* \in \Omega_\delta$ tel que $f(\boldsymbol{\theta}^*) < \delta$, i.e., $\boldsymbol{\theta}^*$ est un point intérieur de Ω_δ ne se situant pas sur la frontière. Alors, pour tout $\boldsymbol{\theta} \in \Omega_1$ nous avons l'inégalité suivante :*

$$(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \left(\boldsymbol{\Gamma}^{-1}\mathrm{Proj}_{\boldsymbol{\Gamma}}(\boldsymbol{\theta}, \mathbf{y}) - \mathbf{y}\right) \leq 0. \tag{6.13}$$

Pour borner les gains adaptatifs, on commence par formuler la loi d'adaptation de l'équation (4.24) avec l'opérateur de projection–$\mathbf{\Gamma}$ où $\mathbf{\Gamma} = \mathbf{\Gamma}_{\boldsymbol{\theta}}$ défini dans le chapitre précédent. On obtient :

$$\Delta\dot{\hat{\boldsymbol{\theta}}} = \mathrm{Proj}_{\mathbf{\Gamma}_\theta}\left(\Delta\hat{\boldsymbol{\theta}}, -\boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\right). \tag{6.14}$$

On rappelle que la fonction candidate de Lyapunov proposée à l'équation (4.20) est :

$$V = \Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\Delta\boldsymbol{\varepsilon} + \mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\mathbf{\Gamma}_\theta^{-1}\Delta\hat{\boldsymbol{\theta}}\mathbf{\Lambda}\right), \tag{6.15}$$

et sa dérivée est :

$$\dot{V} = -\Delta\boldsymbol{\varepsilon}^\top\mathbf{Q}_l\Delta\boldsymbol{\varepsilon} + 2\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\boldsymbol{\xi} + 2\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\left\{\mathbf{\Gamma}_\theta^{-1}\Delta\dot{\hat{\boldsymbol{\theta}}} + \boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\right\}\mathbf{\Lambda}\right). \tag{6.16}$$

En remplaçant par la nouvelle loi d'adaptation donnée à l'équation (6.14), on obtient :

$$\begin{aligned}\dot{V} = &-\Delta\boldsymbol{\varepsilon}^\top\mathbf{Q}_l\Delta\boldsymbol{\varepsilon} + 2\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\boldsymbol{\xi}\\ &+ 2\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\left\{\mathbf{\Gamma}_\theta^{-1}\mathrm{Proj}_{\mathbf{\Gamma}_\theta}\left(\Delta\hat{\boldsymbol{\theta}}, -\boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\right) + \boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\right\}\mathbf{\Lambda}\right).\end{aligned} \tag{6.17}$$

Avec l'équation (5.11) on sait que les deux premiers termes sont semi-définis négatifs si $\Delta\boldsymbol{\varepsilon} \geq \varepsilon_0$. Il reste donc à analyser le dernier terme de cette dérivée. On fait la correspondance avec le lemme 6.2.4 pour y arriver. En effet, sachant que $\Delta\hat{\boldsymbol{\theta}}^\top = \hat{\boldsymbol{\theta}}^\top - \boldsymbol{\theta}^\top$ et que $\boldsymbol{\theta}$ est un point intérieur de Ω_1, on a que $\Delta\hat{\boldsymbol{\theta}}^\top$ correspond à $(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top$. De plus, $\mathbf{\Gamma}_\theta$ correspond à $\mathbf{\Gamma}$ et $-\boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}$ correspond à $\mathbf{y}$. On peut donc utiliser les conclusions du lemme 6.2.4 pour montrer que :

$$\Delta\hat{\boldsymbol{\theta}}^\top\left\{\mathbf{\Gamma}_\theta^{-1}\mathrm{Proj}_{\mathbf{\Gamma}_\theta}\left(\Delta\hat{\boldsymbol{\theta}}, -\boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\right) + \boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\right\} \leq 0. \tag{6.18}$$

De plus, comme la matrice $\mathbf{\Lambda}$ est définie positive et que la trace d'une matrice conserve le signe de son argument, alors

$$2\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\left\{\mathbf{\Gamma}_\theta^{-1}\mathrm{Proj}_{\mathbf{\Gamma}_\theta}\left(\Delta\hat{\boldsymbol{\theta}}, -\boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\right) + \boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\right\}\mathbf{\Lambda}\right) \leq 0 \tag{6.19}$$

Ainsi, à l'extérieur de l'ensemble

$$\Omega = \left\{\left(\Delta\boldsymbol{\varepsilon},\ \Delta\hat{\boldsymbol{\theta}}\right) \in \mathbb{R}^n \times \mathbb{R}^{(n+m+1)} : \|\Delta\boldsymbol{\varepsilon}\| \leq \varepsilon_0 \cap \left\|\Delta\hat{\boldsymbol{\theta}}\right\| \leq \Delta\hat{\boldsymbol{\theta}}_{max}\right\}, \tag{6.20}$$

la dérivée de la fonction candidate de Lyapunov est semi-définie négative, et le système est

stable. Par comparaison avec l'équation (5.11), nous avons ajouté une contrainte $\left\|\Delta\hat{\boldsymbol{\theta}}\right\| \leq \Delta\hat{\boldsymbol{\theta}}_{max}$. En effet, si $\|\Delta\boldsymbol{\varepsilon}\| \leq \varepsilon_0$, deux cas sont possibles :

- $\dot{V} \leq 0$, alors le système est stable, car $\dot{V} \leq 0$ globalement ;
- $\dot{V} > 0$, alors les gains adaptatifs satureront et éventuellement, la dynamique du système quittera Ω. Une fois à l'extérieur de Ω, $\dot{V} \leq 0$ et le système se stabilisera à la frontière de Ω.

Donc, en ajoutant la contrainte sur la norme des gains adaptatifs, nous sommes certains qu'en dehors de Ω, $\dot{V} \leq 0$. Nous devons toutefois confirmer que la condition $\left\|\Delta\hat{\boldsymbol{\theta}}\right\| \leq \Delta\hat{\boldsymbol{\theta}}_{max}$ est toujours respectée. Les gains adaptatifs commence toujours leur évolution à partir de l'origine. Ainsi, par le lemme 6.2.3, ils ne sortiront jamais de l'ensemble convexe défini par l'opérateur de projection. La condition sur la norme des gains adaptatifs est donc toujours vérifiée et l'analyse de la stabilité est valide.

Cette analyse de la stabilité fournit également des indications sur les paramètres ajustables qui sont disponibles à l'automaticien. Le premier est la frontière ε_0, qui est une tolérance sur l'erreur minimale à atteindre par le système. Cette tolérance peut être contrôlée par l'entremise de $\mathbf{Q}_l$ et de $\mathbf{P}_l$ qui se retrouvent dans l'équation de Lyapunov (4.21).

Le second paramètre de design est la borne $\Delta\hat{\boldsymbol{\theta}}_{max}$ sur les gains adaptatifs. Plus cette borne est permissive, plus le système peut tolérer des perturbations importantes. Cependant, si elle est trop permissive le système sera encore sensible au phénomène de dérive des gains. Il faut donc choisir consciencieusement les bornes de l'opérateur de projection pour ne pas limiter l'adaptabilité du système, ni ne laisser le quadricoptère être affecté par la dérive des gains.

6.2.5 Considérations pratique : implémentation de l'opérateur de projection

En pratique, l'opérateur de projection est implémentable facilement, tant en C/C++ qu'en MATLAB/SIMULINK. En effet, le seul calcul requis est celui de la dérivée de la fonction convexe $f(\boldsymbol{\theta})$. Or, peu importe la forme utilisée pour f, la dérivée est calculée directement à partir des gains adaptatifs :

$$\nabla f(\boldsymbol{\theta}) = 2\frac{1+\epsilon}{\epsilon\left\|\boldsymbol{\theta}_{max}\right\|^2}\boldsymbol{\theta} \quad \text{ou} \quad \nabla f(\boldsymbol{\theta}) = 2\frac{1+\epsilon}{\epsilon}\begin{bmatrix} \frac{\theta_1}{\theta_{1,max}^2} \\ \vdots \\ \frac{\theta_n}{\theta_{n,max}^2} \end{bmatrix}. \tag{6.21}$$

Pour l'AscTec Pelican, nous recommandons l'utilisation de la seconde forme pour la fonction $f(\boldsymbol{\theta})$. En effet, une des lacunes des modifications robustes traditionnelles est qu'elles ne

bornent pas chacun des gains individuellement. Ainsi, en choisissant de synthétiser l'opérateur de projection avec cette forme pour f, nous nous affranchissons de ce défaut, car chaque gain aura sa borne. Une des conséquences de ce choix au niveau de l'implémentation de l'opérateur sur le quadricoptère est qu'il y a un grand nombre de paramètres à ajuster :

- un gain pour chacun des 16 états ;
- un gain pour chacune des 4 commandes de référence ;
- un gain pour chacune des 4 incertitudes non paramétriques assorties.

Cela fait un total de 24 gains pour l'ensemble de l'architecture de commande. Or, comme chacun des gains possède une borne maximale et une borne minimale, il y a 48 paramètres à déterminer.

La seconde difficulté à implémenter l'opérateur de projection provient du manque de documentation concernant ce sujet. Depuis la présentation de cet opérateur lisse sur un système non-linéaire par [39], très peu de recherches ont été effectuées sur le sujet. Les auteurs de [30] mentionnent l'avoir utilisé pour leur commande adaptative, mais ils n'expliquent pas comment ils ont obtenu la fonction $f(\boldsymbol{\theta})$ et ses bornes. Enfin, une règle empirique est proposée dans [54], mais elle manque de rigueur, ce qui la rend inutilisable en pratique.

6.3 Obtention des bornes

Dans cette section, nous présentons une méthode originale pour obtenir les bornes des gains adaptatifs qui sont nécessaires pour synthétiser la fonction convexe $f(\boldsymbol{\theta})$. Notre procédure est rigoureuse et systématique et elle permet de comprendre les impacts des bornes sur le système dynamique. Notre algorithme d'obtention des bornes est effectué en plusieurs étapes :

1. Définition d'un ensemble du plan complexe où les pôles du système doivent demeurer pour respecter plusieurs contraintes de performances.
2. Définition d'une application gardienne qui assure la stabilité généralisée de l'ensemble proposé du plan complexe.
3. Construction avec les bornes minimales et maximales des gains adaptatifs d'un système paramétré qui sera utilisé avec l'application gardienne.
4. Utilisation de la μ–analyse pour vérifier si les pôles du système paramétré se situent dans la région du plan complexe protégée par l'application gardienne.
5. Optimisation des valeurs de bornes pour maximiser l'intervalle de variation de chacun des gains adaptatifs.

Nous donnons le formalisme concernant les applications gardiennes à la sous-section 6.3.1. Ensuite, nous nous référons aux travaux de [43] pour l'analyse des application gardiennes. Nous proposons l'algorithme d'obtention des bornes à la sous-section 6.3.2. Enfin, la sous-section 6.3.4 présente les résultats obtenus avec notre méthode pour les différents contrôleurs du système.

6.3.1 Les applications gardiennes

L'ensemble de cette section est basée sur les notions présentées dans [42, 55]. On commence par définir $\mathcal{S}(\Omega)$, l'ensemble de stabilité généralisée par rapport à une région Ω du plan complexe :

$$\mathcal{S}(\Omega) = \left\{ \mathbf{A} \in \mathbb{R}^{n\times n} : \lambda(\mathbf{A}) \in \Omega \right\}, \tag{6.22}$$

où $\lambda(\mathbf{A})$ sont les valeurs propres de la matrice $\mathbf{A}$. L'ensemble Ω, défini dans le plan complexe, représente les endroits où les pôles (ou les valeurs propres) du système peuvent évoluer. Cet ensemble est défini à partir de spécifications physiques : dépassement, temps de réponse, temps de montée, etc. On dira qu'un système est Ω–stable si, peu importe les incertitudes de son modèle, ses pôles sont toujours contenus dans l'ensemble Ω.

Définition 6.3.1. *Soit $\mathcal{X}$ l'ensemble des matrices carrées réelles de dimension n et $\mathcal{S}$, un sous-ensemble ouvert de $\mathcal{X}$. Soit $\nu : \mathcal{X} \mapsto \mathbb{C}$. On dit que ν garde $\mathcal{S}$, si, pour tout $\mathbf{x} \in \overline{\mathcal{S}}$, on a*

$$\nu(\mathbf{x}) = 0 \Longleftrightarrow x \in \partial\mathcal{S}, \tag{6.23}$$

où $\overline{\mathcal{S}}$ est l'adhérence de $\mathcal{S}$ et $\partial\mathcal{S}$ est la frontière de $\mathcal{S}$. On rappelle que l'adhérence est le plus petit ensemble fermé contenant $\mathcal{S}$. Il existe plusieurs applications gardiennes permettant de protéger divers endroits du plan complexe. Certaines d'entre elles sont présentées dans le lemme suivant :

Lemme 6.3.1. *Soit une matrice $\mathbf{A} \in \mathbb{R}^{n\times n}$, alors les applications gardiennes suivantes permettent de garder un espace correspondant du plan complexe. Ces applications gardiennes sont définies avec le produit et la somme de Kronecker.*[1] *L'annexe D.3 donne de plus amples détails concernant ces opérations mathématiques.*

- $\nu(\mathbf{A}) := \det(\mathbf{A} \oplus \mathbf{A})$ *garde la stabilité Hurwitz.*
- $\nu(\mathbf{A}) := \det((\mathbf{A} + \sigma\mathbf{I}_n) \oplus (\mathbf{A} + \sigma\mathbf{I}_n))$ *garde le demi-plan gauche ouvert décalé de $-\sigma$.*

1. Il aurait été possible de définir ces applications gardiennes en utilisant le produit bialterné ou le Schläflian. Or, ce gain en efficacité est contrebalancé par une augmentation de la complexité lors de l'implémentation. Ainsi, pour maintenir un algorithme d'optimisation qui est simple, nous avons préféré n'utiliser que la somme et le produit de Kronecker.

– $\nu(\mathbf{A}) := \det\left(e^{-j\theta}\mathbf{A} \oplus e^{j\theta}\mathbf{A}\right)$ *garde le cône d'angle* $\pi - 2\theta$ *dans le demi-plan gauche.*

La figure 6.3 illustre les ensembles protégés par ces applications gardiennes.

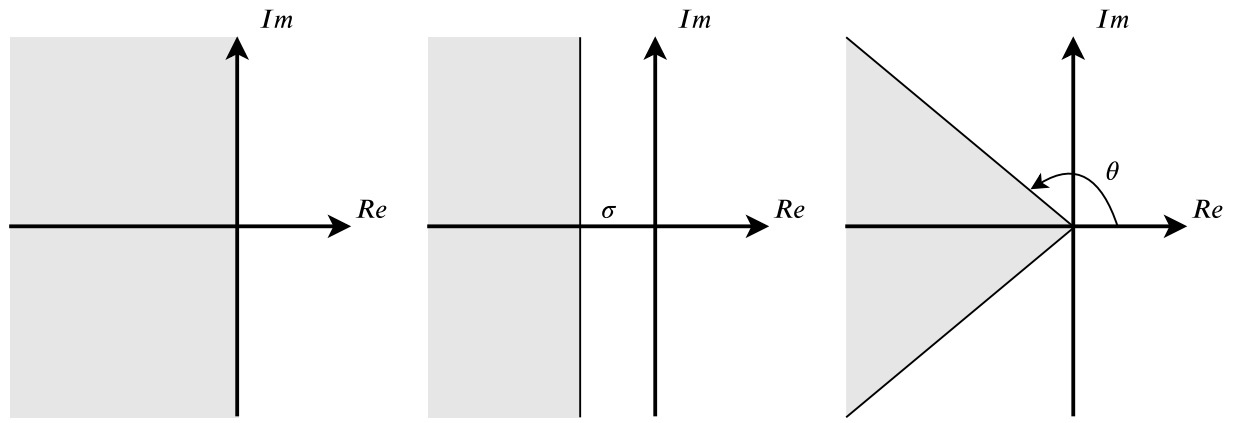


Figure 6.3 Régions de stabilité d'intérêt

Il est possible de combiner simultanément des applications gardiennes pour protéger un ensemble composé de plusieurs frontières distinctes.

Lemme 6.3.2. *Soit* $\mathcal{S}$*,* $\mathcal{S}_1$ *et* $\mathcal{S}_2$ *des sous-ensembles de* $\mathbb{R}^{n\times n}$*. Soit* ν_1 *qui garde* $\mathcal{S}_1$ *et* ν_2 *qui garde* $\mathcal{S}_2$*. Alors,* $\mathcal{S}_1 \cap \mathcal{S}_2$ *est gardé par* $\nu : \mathbf{A} \mapsto \nu_1(\mathbf{A})\nu_2(\mathbf{A})$*. Plus particulièrement, si* $\mathcal{S}_1 = \mathcal{S}(\Omega_1)$ *et* $\mathcal{S}_2 = \mathcal{S}(\Omega_2)$*, alors* $\mathcal{S}(\Omega_1 \cap \Omega_2)$ *est gardé par* ν*.*

Nous proposons maintenant l'application gardienne qui est utilisée pour l'obtention des bornes des gains adaptatifs et qui protège l'ensemble donné à la figure 6.4. Nous choisissons les frontières suivantes :

1. La frontière définie avec σ_1 assure que le système ne devienne pas trop rapide. Ainsi, nous empêchons les moteurs de saturer dû à une commande trop agressive.
2. La frontière définie par σ_2 est choisie pour que le système ne devienne pas trop lent. Nous prévenons ainsi que le système ne devienne léthargique et qu'il soit incapable de rejeter des perturbations rapides.
3. Les frontières définies par θ empêchent que l'amortissement des pôles complexes conjugués ne devienne trop petit. Ainsi, nous voulons assurer que le système ne présente pas de fortes oscillations en suivi de trajectoire.

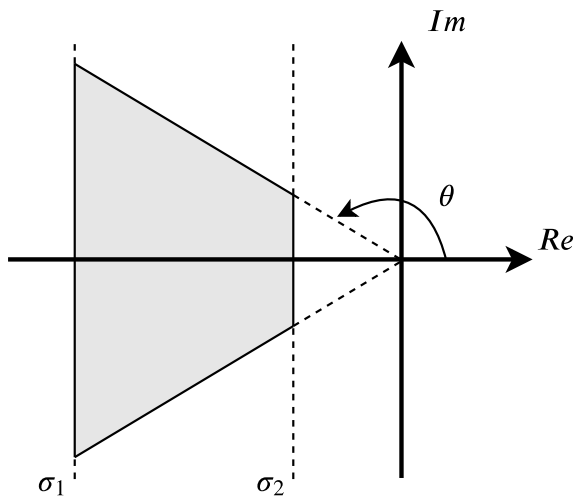


Figure 6.4 Application gardienne d'intérêt

Par le lemme 6.3.2 l'application gardienne qui garde cet ensemble est donnée par :

$$\begin{aligned}\nu(\mathbf{A}) &= \nu_{\sigma_1}(\mathbf{A})\nu_{\sigma_2}(\mathbf{A})\nu_{\theta}(\mathbf{A}) \\ &= \det\left((\mathbf{A}+\sigma_1\mathbf{I}_n)\otimes(\mathbf{A}+\sigma_1\mathbf{I}_n)\right)\cdot\det\left((\mathbf{A}+\sigma_2\mathbf{I}_n)\otimes(\mathbf{A}+\sigma_2\mathbf{I}_n)\right) \\ &\qquad\qquad\cdot\det\left(e^{-j\theta}\mathbf{A}\oplus e^{j\theta}\mathbf{A}\right)\end{aligned} \tag{6.24}$$

6.3.2 Proposition de l'algorithme d'optimisation

On cherche à obtenir les bornes des gains adaptatifs tout en limitant les pôles du système à un ensemble défini du plan complexe. Cependant, ce ne sont pas tous les gains adaptatifs qui influencent les pôles. Par exemple, pour le contrôleur d'altitude, il y a 5 gain adaptatifs : un gain par état $\hat{k}_z$, $\hat{k}_w$ et $\hat{k}_{x_{i,z}}$, un gain sur la référence $\hat{k}_{z_{ref}}$ et un gain pour la perturbation non paramétrique assortie $\hat{k}_{f_z}$. Or, les gains $\hat{k}_{z_{ref}}$ et $\hat{k}_{f_z}$ n'influencent pas les pôles du système. Nous pouvons donc ignorer temporairement ces gains pour notre algorithme et ne considérer que les gains adaptatifs agissant sur les états du système. De plus, on suppose qu'il n'y a aucune perturbation qui affecte le système, donc $\mathbf{\Lambda}=\mathbf{I}$, $\mathbf{f}(\mathbf{x})=\mathbf{0}$ et $\boldsymbol{\xi}(t)=\mathbf{0}$. En reprenant l'équation (4.18) avec ces simplifications, on a :

$$\Delta\dot{\mathbf{x}}_a = \mathbf{A}_{ref}\Delta\mathbf{x}_a + \mathbf{B}_{ref}\Delta\mathbf{r} + \mathbf{B}\Delta\mathbf{v}_{ad}. \tag{6.25}$$

Puisque nous ignorons les incertitudes non paramétriques, la commande adaptative est donnée par $\Delta\mathbf{v}_{ad} = \hat{\mathbf{K}}_u \Delta\mathbf{x}_a$. La dynamique du système en boucle fermée est donc :

$$\Delta\dot{\mathbf{x}}_a = \left(\mathbf{A}_{ref} + \mathbf{B}\hat{\mathbf{K}}_u\right)\Delta\mathbf{x}_a + \mathbf{B}_{ref}\Delta\mathbf{r} = \mathbf{A}_p(\hat{\mathbf{K}}_u)\Delta\mathbf{x}_a + \mathbf{B}_{ref}\Delta\mathbf{r}, \tag{6.26}$$

où $\mathbf{A}_p(\hat{\mathbf{K}}_u)$ est la matrice paramétrée en fonction de $\hat{\mathbf{K}}_u$. Cette matrice est utilisée avec l'application gardienne définie en (6.24). Pour des systèmes paramétrés, l'analyse effectuée avec des applications gardiennes se fait principalement par des méthodes graphiques. Cette analyse est possible pour des matrices ayant deux paramètres, mais elle se complique grandement dès qu'il y en a plus que trois. On rappelle que les contrôleurs d'altitude et de rotation en lacet ont chacun 3 états et donc 6 paramètres : trois bornes maximales et trois bornes minimales. Les contrôleurs des positions longitudinale et latérale ont 5 états et donc 10 paramètres à déterminer. L'analyse graphique de l'application gardienne est alors impossible.

Les travaux de [43] permettent de s'affranchir de l'analyse graphique des applications gardiennes si le système respecte deux conditions. D'abord, il doit être nominalement Ω–stable. Puisque le système nominal est le modèle de référence, il l'est par design. Ensuite, on doit pouvoir le représenter sous la forme d'une matrice à intervalle. Celle-ci est définie mathématiquement comme

$$\begin{aligned} \mathbf{A}_I &= \left[\underline{\mathbf{A}} \quad \overline{\mathbf{A}}\right], \\ &= \left\{\mathbf{A} = [a_{ij}] : \underline{a}_{ij} \leq a_{ij} \leq \overline{a}_{ij}, \quad i,\ j = 1, \ldots, n\right\}. \end{aligned} \tag{6.27}$$

Si ces conditions sont respectées, il est possible de ramener l'analyse des applications gardiennes à un calcul de la VSS. On peut alors déterminer si le système est stable par rapport à l'espace défini dans le plan complexe. Les détails concernant la transition entre l'application gardienne et la μ–analyse sont présentés dans l'annexe D.4. Néanmoins, nous devons démontrer qu'il est possible de représenter notre problème sous la forme d'une matrice à intervalle.

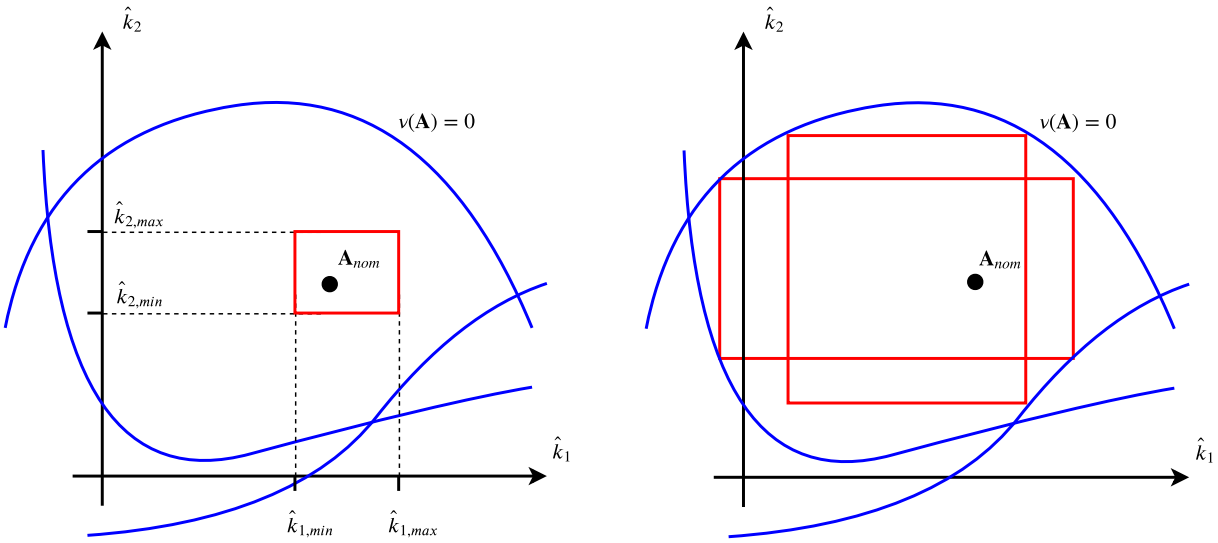
À cette fin, on définit $\hat{\mathbf{K}}_{min}$ et $\hat{\mathbf{K}}_{max}$ les bornes minimales et maximale sur les gains adaptatifs associés aux états $\Delta\mathbf{x}_a$. On peut représenter $\mathbf{A}_p(\hat{\mathbf{K}}_u)$ comme une matrice à intervalle, avec $\mathbf{A}_p(\hat{\mathbf{K}}_u) = [\underline{\mathbf{A}}_p \ \ \overline{\mathbf{A}}_p]$ où

$$\underline{\mathbf{A}}_p = \mathbf{A}_{ref} + \mathbf{B}_{ref}\hat{\mathbf{K}}_{u,min}, \quad \overline{\mathbf{A}}_p = \mathbf{A}_{ref} + \mathbf{B}_{ref}\hat{\mathbf{K}}_{u,max}. \tag{6.28}$$

Nous pouvons donc utiliser les travaux de [43] pour l'analyse des applications gardiennes. En effet, nous pouvons déterminer si, pour un certain choix de bornes $\hat{\mathbf{K}}_{min}$ et $\hat{\mathbf{K}}_{max}$, les pôles

du système ne sortent pas de l'ensemble Ω du plan complexe. Il faut alors d'optimiser les bornes maximales et minimales où le système demeure Ω–stable.

Exemple illustratif. On considère une matrice $\mathbf{A} \in \mathbb{R}^{2\times 2}$ et deux gains adaptatifs $\hat{k}_1$ et $\hat{k}_2$. Une application gardienne $\nu(\mathbf{A})$ a été définie pour garder les valeurs propres de $\mathbf{A}$ dans un certain ensemble Ω du plan complexe. On suppose que, le système nominal $\mathbf{A}_{nom}$ est stable par rapport à cet ensemble. On cherche à déterminer les bornes minimales et maximales de ces gains qui respectent la contrainte imposée par l'ensemble Ω. La figure 6.5a donne un aperçu de la description de cet exemple. Les courbes en bleu représentent le moment où $\nu(\mathbf{A}) = 0$. Cela veut donc dire que les pôles sont sur la frontière de l'ensemble Ω et qu'ils ne respectent pas les critères imposés.



(a) Bornes conservatrices (b) Bornes optimisées

Figure 6.5 Objectif de l'algorithme d'optimisation pour l'obtention des bornes

Une des particularité des applications gardiennes est qu'elles définissent des zones maximales de stabilité généralisée autour d'un point nominal comme étant celles délimités par les courbes $\nu(\mathbf{A}) = 0$. Ces zones sont des aires en 2D, des volumes en 3D, etc. On dit qu'elles conservent *l'inertie* du système : si un point dans la zone est Ω–stable, alors toute la zone est Ω–stable. Dans la figure 6.5a, on voit que le système nominal rend toute la zone centrale à l'intérieure des trois courbes bleues Ω–stable. Le rectangle rouge donne également un exemple des bornes sur $\hat{k}_1$ et $\hat{k}_2$ qui maintiennent le système Ω–stable. Or, ce rectangle est trop conservateur. En effet, la figure 6.5b montre des zones optimales qui assurent la stabilité Ω du système tout en maximisant l'aire du rectangle rouge.

Cet exemple illustratif explique l'objectif de l'algorithme d'obtention des bornes des gains. Grâce à cet exemple, nous proposons la fonction suivante à optimiser :

$$V\left(\hat{\mathbf{K}}_{min},\ \hat{\mathbf{K}}_{max}\right) = \prod_{i=1}^{n}\left(\hat{\mathbf{K}}_{i,max} - \hat{\mathbf{K}}_{i,min}\right), \tag{6.29}$$

où $\hat{\mathbf{K}}_{i,max} - \hat{\mathbf{K}}_{i,min}$ représente l'intervalle de variation pour chacun des gains adaptatifs. Cette fonction représente donc le volume d'un hyperrectangle à n dimensions. En effet, on tente de maximiser l'intervalle de variation pour chacun des gains adaptatifs individuellement. De plus, nous définissons les contraintes suivantes qui doivent être respectées par les solutions de l'algorithme :

Condition 1 (**C1**) : $\hat{\mathbf{K}}_{i,min} < 0$ et $\hat{\mathbf{K}}_{i,max} > 0$. Il n'est pas possible de prendre les bornes du même signe, car cela signifierait que le système nominal ne fait pas partie de l'intervalle entre les deux bornes. Or, cette condition est nécessaire pour résoudre le problème.

Condition 2 (**C2**) : Les valeurs propres de la matrice à intervalle $\mathbf{A}_p$ doivent être contenues dans l'ensemble Ω qui est protégé par l'application gardienne $\nu(\mathbf{A}_p)$. On peut vérifier cette condition en passant par la μ–analyse.

Donc, le problème d'optimisation à résoudre est formulé comme :

$$\begin{aligned} \max_{\hat{\mathbf{K}}_{min},\ \hat{\mathbf{K}}_{max}} \quad & V\left(\hat{\mathbf{K}}_{min},\ \hat{\mathbf{K}}_{max}\right) \\ \text{s.c.} \quad & \mathbf{C1},\ \mathbf{C2}. \end{aligned} \tag{6.30}$$

6.3.3 Fonctionnement général de l'algorithme

Pour obtenir des solutions qui sont des maximums locaux pour le problème formulé en (6.36), la démarche suivante est proposée :

1. Générer un ensemble de points de départ pour les bornes des gains dans une zone prédéfinie, car il pourrait y avoir plusieurs solutions maximales locales. Les points de départ sont générés de façon aléatoire avec l'algorithme de l'échantillonnage par hypercube latin, *Latin Hypercube Sampling* (LHS). L'annexe D.5 donne de plus amples détails sur ce type d'échantillonnage et comment l'implémenter. Les points générés par cet algorithme respectent toujours la condition **C1**.
2. Éliminer les points de départ qui ne respectent pas la condition **C2**.
3. Utiliser l'algorithme d'optimisation pour obtenir un maximum local pour chacun des points initiaux valides. L'algorithme d'optimisation utilisé est la recherche directe par

grille adaptive, *Mesh Adaptive Direct Search* (MADS) développé par [56]. Le lecteur est invité à se référer à [57] pour les notions concernant cet algorithme.

4. Filtrer les résultats de l'algorithme pour ne conserver que ceux qui sont d'intérêt.

La démarche proposée est intéressante et pertinente pour deux raisons. D'abord, elle permet d'obtenir les bornes de la fonction $f(\boldsymbol{\theta})$ de façon rigoureuse et systématique. La plupart des sources consultées n'abordent pas l'obtention de ces bornes qui sont pourtant essentielles pour la mise en œuvre de l'opérateur de projection. Ainsi, nous venons combler une lacune présente dans la littérature. Ensuite, cette démarche permet d'unifier deux domaines qui sont souvent séparés en commande : la commande adaptative et la commande robuste. En effet, notre démarche utilise des méthodes d'analyse en commande robuste pour ensuite être appliquée à une architecture de commande adaptative. Pour de plus amples détails sur l'algorithme d'optimisation MADS, le lecteur est invité à se référer à l'annexe D.6. La sous-section suivante servira à illustrer le bon fonctionnement de l'algorithme lorsqu'il est utilisé sur un exemple simple.

Application de l'algorithme à un exemple simple

On considère la fonction de transfert suivante pour un système :

$$T(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^3 + s^2 + s + r_1^2 + r_2^2}. \tag{6.31}$$

Cette fonction de transfert dépend de deux paramètres r_1 et r_2 ayant des valeurs nominales $r_1^0 = r_2^0 = 0.5$. On souhaite déterminer des bornes maximales et minimales pour ces paramètres tout en maintenant la stabilité Hurwitz du système. La condition nécessaire et suffisante pour garantir cette stabilité sont : $0 < r_1^2 + r_2^2 < 1$. Par conséquent, l'application gardienne dans ce cas est :

$$\nu = (r_1^2 + r_2^2)(1 - (r_1^2 + r_2^2)). \tag{6.32}$$

L'ensemble de stabilité paramétré maximal autour de (r_1^0, r_2^0) est alors le cercle unitaire ouvert moins l'origine, d'où l'intérêt de cet exemple.

Pour utiliser notre algorithme d'obtention des bornes, nous devons formuler le problème

adéquatement. Nous représentons d'abord la fonction de transfert $T(s)$ par un modèle d'état :

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -(r_1^2 + r_2^2) & -1 & -1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x} \end{aligned} \tag{6.33}$$

L'application gardienne qui protège la stabilité Hurwitz est $\nu(\mathbf{A}) = \det(\mathbf{A})$ où $\mathbf{A} = \mathbf{A}(r_1, r_2)$. On représente maintenant notre système grâce à une matrice à intervalle $\mathbf{A}_I = [\underline{\mathbf{A}} \;\; \overline{\mathbf{A}}]$, où

$$\underline{\mathbf{A}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -((r_1^0 + r_{1,min})^2 + (r_2^0 + r_{2,min})^2) & -1 & -1 \end{bmatrix}, \; \overline{\mathbf{A}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -((r_1^0 + r_{1,max})^2 + (r_2^0 + r_{2,max})^2) & -1 & -1 \end{bmatrix}. \tag{6.34}$$

La forme précédente permet de bien illustrer comment les bornes affectent le système : elles déterminent une variation par rapport à la valeur nominale. Ainsi, les bornes construiront un rectangle autour de la valeur nominale (r_1^0, r_2^0) . L'objectif de l'algorithme d'optimisation est d'obtenir l'aire la plus grande possible pour cette variation autour du point nominal. Pour appliquer le processus d'optimisation, nous considérons les conditions suivantes :

C1 : $r_{1,min} < 0$ et $r_{2,min} < 0$;
C2 : $r_{1,max} > 0$ et $r_{2,max} > 0$;
C3 : Les pôles de $\mathbf{A}_I$ à contenu dans la région du plan complexe protégée par $\nu(\mathbf{A})$.

La fonction à optimiser est :

$$V(r_{1,min}, r_{2,min}, r_{1,max}, r_{2,min}) = (r_{1,max} - r_{1,min}) \cdot (r_{2,max} - r_{2,min}) \tag{6.35}$$

Le problème d'optimisation est alors formulé comme :

$$\begin{aligned} \max_{r_{1,min},\, r_{2,min},\, r_{1,max},\, r_{2,max}} \quad & (r_{1,max} - r_{1,min}) \cdot (r_{2,max} - r_{2,min}) \\ \text{s.c.} \quad & \mathbf{C1},\ \mathbf{C2},\ \mathbf{C3} \end{aligned} \tag{6.36}$$

Nous fournissons le point de départ suivant à l'algorithme d'optimisation : $r_{1,min} = r_{2,min} = -0.03$ et $r_{1,max} = r_{2,max} = 0.03$. La condition d'arrêt pour l'algorithme est que pour deux itérations successives, la différence entre les solutions soit inférieure à 0.0002. L'algorithme

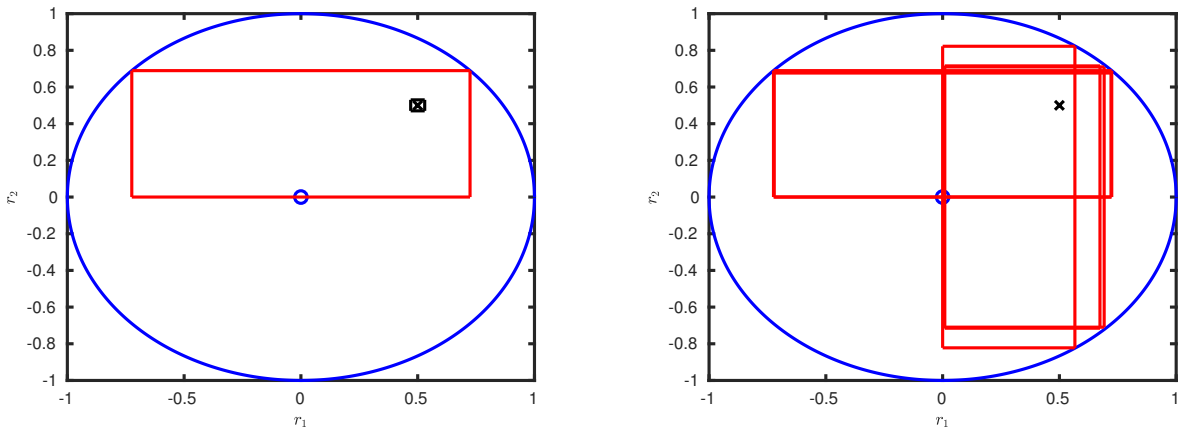
MADS nécessite environ 10 secondes pour obtenir la solution suivante :

$$\left.\begin{array}{ll} r_{1,min} = -1.2241 & r_{1,max} = 0.2241 \\ r_{2,min} = -0.4999 & r_{2,max} = 0.1893 \end{array}\right\} \rightarrow V = 2.4464 \tag{6.37}$$

Les valeurs obtenues se rapprochent des conditions nécessaires et suffisantes exprimées ci-haut. L'algorithme tend donc vers les limites imposées par l'application gardienne. En effet, dans ce rectangle, on aura :

$$r_1^2 + r_2^2 = \left[1.000 \cdot 10^{-8},\ 0.999\right] \tag{6.38}$$

Cela correspond presque exactement aux conditions : $0 < r_1^2 + r_2^2 < 1$. La figure 6.6a illustre les résultats obtenus par l'algorithme d'optimisation pour ce cas optimal. Les courbes en bleu correspondent au moment où $\nu(\mathbf{A}) = 0$. Le rectangle noir correspond aux bornes de départ de l'algorithme et le rectangle en rouge correspond à la solution optimale locale obtenue à la suite de l'optimisation. Le point nominal (r_1^0, r_2^0) est représenté par un $\times$ noir. On voit que les sommets du rectangle et les arrêtes ont atteint les frontières imposées par les applications gardiennes. L'algorithme a donc permis d'obtenir une solution respectant les conditions **C1**, **C2** et **C3**.



(a) Résultat de l'algorithme d'optimisation appliqué à un cas simple

(b) Différentes solutions de l'algorithme d'optimisation appliqué à un cas simple

Figure 6.6 Compensation des pôles à la suite d'une perturbation $\mathcal{P}$

La figure 6.6 donne les autres solutions optimales obtenues qui correspondent à des aires variant de $[2.3899, 2.4166]$. On remarque d'abord que l'algorithme obtient une aire semblable d'une condition initiale à une autre. Ce qui diffère, c'est l'axe qui sera privilégié par l'al-

gorithme pour former le côté le plus grand du rectangle. Puisqu'il y a plusieurs solutions possible, il revient à l'utilisateur de déterminer la solution optimale à conserver.

Cet exemple simple permet de démontrer les points importants à considérer lors de l'utilisation de l'algorithme d'obtention des bornes. D'abord l'algorithme est dépendant des conditions initiales. Ayant démontré le fonctionnement de notre algorithme d'obtention des bornes sur un cas simple, nous pouvons à présent présenter les résultats lorsque cet algorithme est utilisé pour obtenir les bornes de la fonction $f(\boldsymbol{\theta})$ pour l'opérateur de projection.

6.3.4 Résultats de l'algorithme

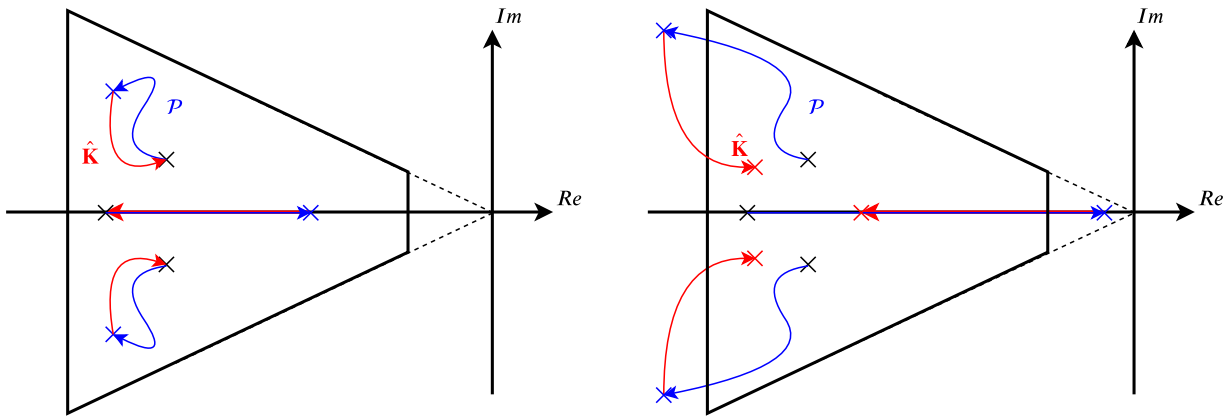
Cette section démontre le fonctionnement de notre algorithme d'obtention des bornes lorsqu'il est utilisé pour le contrôleur d'altitude. L'annexe D.7 donne un condensé du contenu de cette section pour les trois autres contrôleurs. On commence donc par déterminer l'ensemble Ω qui doit être protégé par l'application gardienne. Cet ensemble doit contenir les pôles du système nominal qui sont :

$$p_{1,2} = -1.1421 \pm 1.4770j, \quad p_3 = -1.4469. \tag{6.39}$$

On veut maintenant choisir des frontières appropriées pour l'ensemble Ω. Cet ensemble définit une région du plan complexe où les pôles seront parfaitement compensés par la MRAC, étant donné une perturbation $\mathcal{P}$. En effet, si la perturbation modifie les pôles du système, mais qu'ils restent dans Ω, alors il existe des gains $\hat{\mathbf{K}}$ qui compenseront parfaitement cette perturbation. Cela est illustré à la figure 6.7a. Toutefois, si la perturbation $\mathcal{P}$ fait sortir les pôles de l'ensemble Ω, alors il n'existe pas des gains $\hat{\mathbf{K}}$ qui ramèneront les pôles à leur valeur nominale. Cela est illustré à la figure 6.7b. Il est donc important de définir un ensemble Ω qui est suffisamment grand pour permettre à la MRAC de compenser la plupart des incertitudes pouvant affecter le quadricoptère.

Pour la partie réelle des pôles, nous choisissons $\sigma_1 = -3$ et $\sigma_2 = -0.2$. Cela permet de ramener à sa valeur nominale le temps de réponse du système si la perturbation le modifiait de 1.3 s à 20 s. Pour l'amortissement, on choisit $\theta = 0.21$ rad, pour compenser un dépassement d'au plus 50%. Ces valeurs sont utilisées pour construire l'application gardienne qui va protéger Ω.

La première étape de l'algorithme d'obtention des bornes est de générer des points de départ qui respectent les contraintes **C1** et **C2**. À la suite du filtrage des points obtenus avec le LHS, nous avons 11 points de départ. Pour chacun des points, on utilise l'algorithme de MADS pour obtenir un maximum local. Après ce processus d'optimisation, l'intervalle pour



(a) Pôles compensés (b) Pôles non compensés

Figure 6.7 Compensation des pôles à la suite d'une perturbation $\mathcal{P}$

les solutions maximales locales des hypervolumes V est [4.2536, 16.2213], avec une moyenne située à 11.0001. À la suite de simulations et de tests expérimentaux, la solution retenue est :

$$\begin{aligned} \hat{k}_{z,max} &= 0.8192, & \hat{k}_{z,min} &= -0.7954, & \hat{k}_{z,int} &= 1.6146, \\ \hat{k}_{w,max} &= 2.6545, & \hat{k}_{w,min} &= -0.0023, & \hat{k}_{w,int} &= 2.6568, \\ \hat{k}_{x_{i,z},max} &= 1.9371, & \hat{k}_{x_{i,z},min} &= -0.0315, & \hat{k}_{x_{i,z},int} &= 1.9686, \end{aligned} \tag{6.40}$$

où $\hat{k}_{\cdot,int}$ donne l'intervalle entre les deux bornes. L'hypervolume des gains de cette solution est $V = 8.4445$ Dans la section suivante, nous expliquerons comment déterminer quelles solutions sont utilisables en pratique.

Choix entre les solutions possibles

L'algorithme d'optimisation fournit plusieurs solutions qui sont des maximums locaux. Or, toutes ces solutions ne sont pas équivalentes. On doit être en mesure de choisir une bonne solution, même si elle n'est pas celle ayant le hypervolume maximal. Voici quelques principes qui ont guidé notre choix :

- Les tests expérimentaux effectués sur le quadricoptère aux chapitres 4 et 5, donne une idée des requis pour les bornes des gains. Par exemple, la figure 5.10 montre que les gains adaptatifs du contrôleur d'altitude sont à peu près égaux, i.e., les valeurs de $\hat{k}_z$, $\hat{k}_w$ et $\hat{k}_{x_{i,z}}$ sont similaires. On veut donc choisir une solution qui respecte cet intervalle pour ces gains.
- Les intervalles pour chacun des gains doivent être convenables. Ainsi, on ne veut pas

qu'un des gains ait un intervalle disproportionné par rapport aux autres. Par exemple, la solution ayant le plus grand hypervolume donne les bornes suivantes pour l'état z : $\hat{k}_{z,int} = 1.0397$. Cet intervalle est trop petit par rapport aux autres : $\hat{k}_{w,int} = 4.3974$ et $\hat{k}_{x_{i,z},int} = 3.5478$.

– On veut que les gains adaptatifs aient leur intervalle dans la bonne « direction ». Pour les trois gains choisis, on sait que leurs valeurs sont généralement positives et qu'elles varient entre $[0,\ 1]$. On veut donc s'assurer que la solution maximale locale choisie permette d'obtenir des bornes qui laissent les gains évoluer librement dans cet intervalle.

Ainsi, bien qu'il existe d'autres solutions donnant un meilleur volume pour les gains adaptatifs, il faut choisir la solution locale la plus adaptée au problème étudié. Il faut également effectuer des tests en laboratoire pour déterminer la façon dont les gains évoluent en fonction des perturbations et des incertitudes qui affectent le quadricoptère. Selon nous, une panne de moteur de 20% à 30% est suffisante pour avoir une idée générale des intervalles nécessaires pour assurer que les gains adaptatifs puissent évoluer convenablement. Cela permettra à la MRAC de bien remplir son objectif sans limiter les performances du quadricoptère.

Enfin, il reste deux autres bornes à fixer pour les états $\hat{k}_{z_{ref}}$ et $\hat{f}_z$. Nous rappelons que ces bornes ont été ignorées précédemment, car elles n'influencent pas la valeur des pôles du système. Nous fixons leurs valeurs maximales et minimales pour ne pas nuire au système. Nous choisissons la borne minimale plus petite que $\min\{\hat{k}_{z,min}, \hat{k}_{w,min}, \hat{k}_{x_{i,z},min}\}$ et la borne maximale plus grande que $\max\{\hat{k}_{z,max}, \hat{k}_{w,max}, \hat{k}_{x_{i,z},max}\}$. Dans notre cas, le choix $\hat{k}_{z,ref}, \hat{f}_z = [-1, 3]$ est convenable.

6.4 Résultats expérimentaux

On déploie la MRAC construite avec l'opérateur de projection sur l'AscTec Pelican. Pour y arriver, on utilise les bornes présentées à la section précédente ainsi qu'à l'annexe D.7. Cette section sert de preuve de concept pour l'opérateur de projection. Ainsi, on ne caractérisera pas entièrement les performances du quadricoptère comme au chapitre 4. Nous démontrerons plutôt que l'opérateur de projection est fonctionnel et qu'il ne nuit pas aux performances du quadricoptère.

La figure 6.8 donne l'évolution des gains adaptatifs pour un vol stationnaire avec une panne de moteur de 20% qui survient à $t = 30$ s. Les résultats sont identiques à ceux obtenus à la section 5.1.3 pour démontrer que le phénomène de dérive des gains est présent sur le quadricoptère. On pourrait donc croire que l'opérateur de projection ne réussit pas à empêcher ce

phénomène. Cela n'est toutefois pas le cas. En effet, les bornes optimales laissent une grande marge de manœuvre à la commande adaptative. Cela provient du fait que l'ensemble Ω choisi est permissif, car il doit assurer qu'une quantité suffisantes des perturbations et d'incertitudes puisse être rejetée. Ainsi, pour que les gains saturent, il faudrait perturber grandement le système ou effectuer un vol très long.

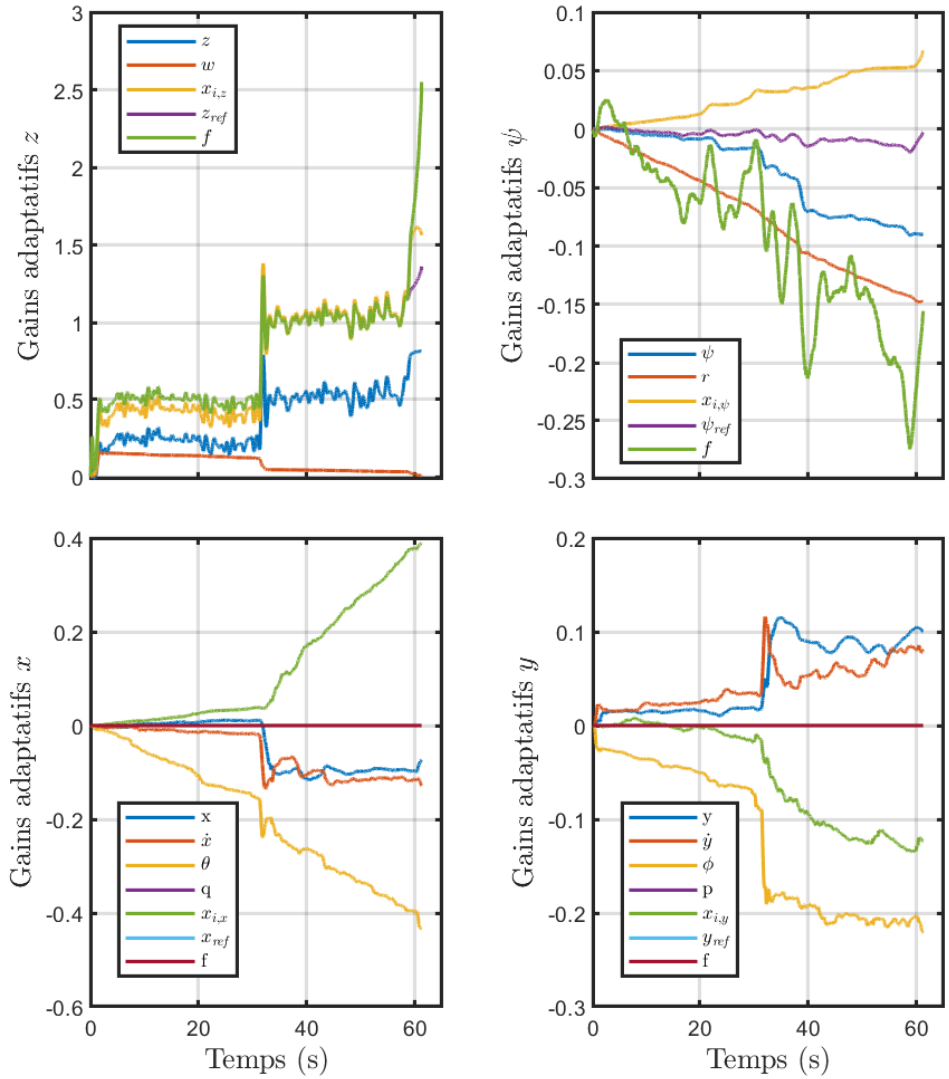


Figure 6.8 Gains adaptatifs pour les résultats optimaux de l'algorithme d'optimisation

Néanmoins, pour illustrer que l'opérateur de projection peut borner les gains adéquatement, l'expérience est refaite avec des bornes plus restrictives. En effet, nous choisissons une so-

lution de l'algorithme d'obtention de bornes qui est moins optimale que la précédente. La figure 6.9 donne l'évolution des gains adaptatifs pour un vol avec ces bornes restrictives. On voit, entre autres, que les gains des états r, $x_{i,x}$ et $x_{i,y}$ saturent à leurs bornes qui sont : $\hat{k}_r = -0.1740$, $\hat{k}_{x_{i,x},max} = 0.3917$ et $\hat{k}_{x_{i,y},min} = -0.0323$. Cela démontre que l'opérateur est bien capable de borner les gains lorsqu'ils arrivent près de leur point de saturation.

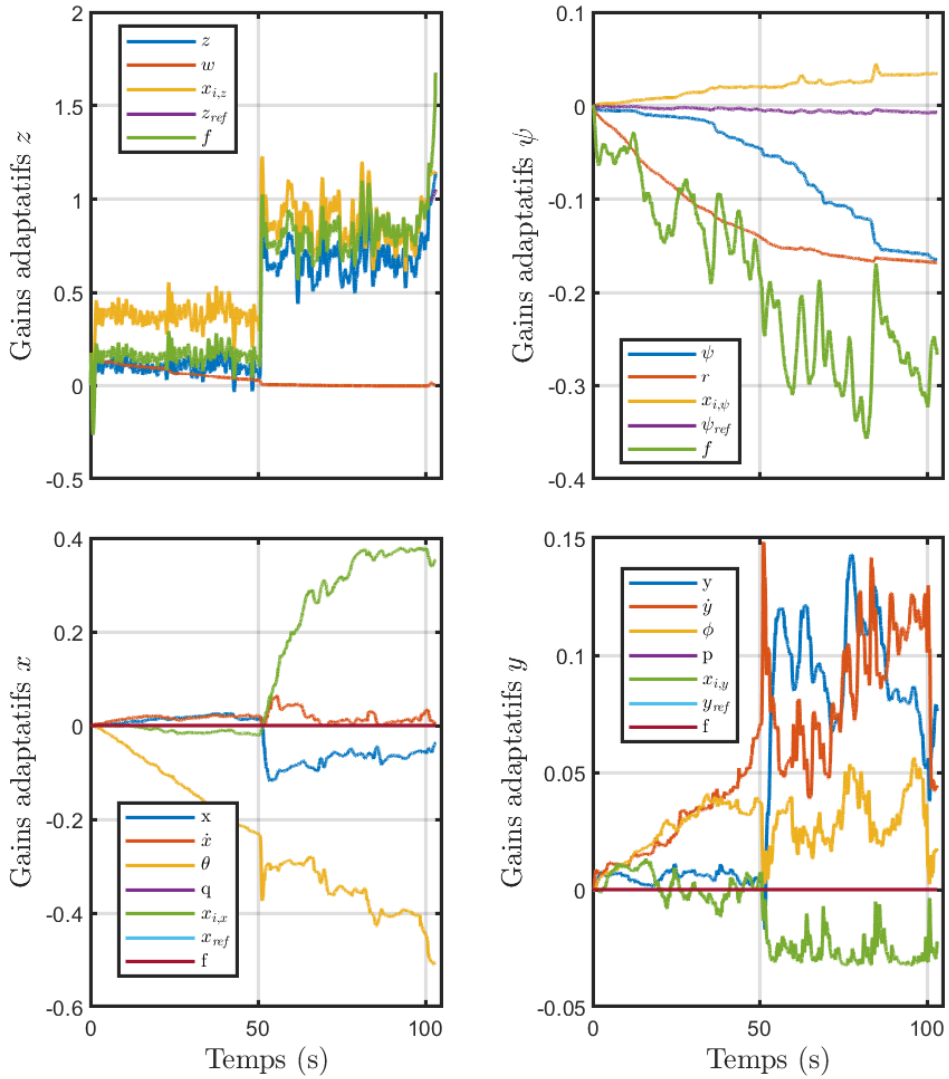
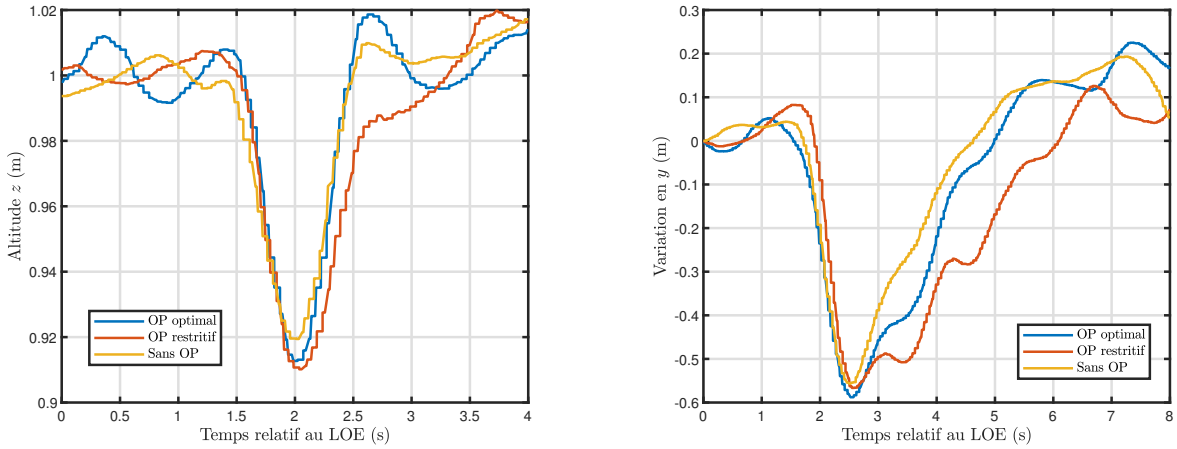


Figure 6.9 Gains adaptatifs pour les résultats restrictifs de l'algorithme d'optimisation

Les figures 6.10a et 6.10b donnent les performances du quadricoptère en altitude et en position latérale. On remarque qu'avec l'opérateur de projection, le quadricoptère chute de 1 cm de

plus que si la MRAC est synthétisée sans cette modification robuste. Comme pour les résultats de la modification–ϵ, cette chute additionnelle correspond à une diminution de 12.5% des performances. Or, nous considérons cela comme étant négligeable. La figure 6.10a montre également qu'avec des bornes plus restrictives, le quadricoptère met 1 s supplémentaire pour retourner à son point d'équilibre. On constate donc une légère dégradation des performances si les bornes des gains adaptatifs ne sont pas optimales.

La figure 6.10b montre que la panne de moteur provoque environ le même déplacement latéral pour les trois lois de commande. Par contre, le retour à l'origine est affecté par l'opérateur de projection. Même pour les bornes optimales, le quadricoptère met 0.5 s de plus pour retourner au point d'équilibre ; avec les bornes restrictives, il met 1.5 s de plus. On constate aussi que le mouvement du quadricoptère est plus saccadé dans le cas non optimal.



(a) Performances en altitude z

(b) Performances en position y

Figure 6.10 Performances de la MRAC avec opérateur de projection

Les deux figures précédentes démontrent que le choix des bornes des gains est primordial pour le bon fonctionnement de la modification robuste. En effet, s'ils sont trop restrictifs, alors les performances du quadricoptère sont dégradées. Si les bornes sont trop permissives, alors le phénomène de dérive des gains peut encore avoir un effet sur l'appareil. Il faut donc bien étudier l'influence des incertitudes paramétriques et non paramétriques dans l'optique de bien définir l'ensemble Ω où les pôles du système doivent rester. Nous reviendrons sur ce point dans les recherches futures à la section 7.2.

Ceci dit, l'opérateur de projection possède plusieurs avantages. D'abord, avec cette modification robuste, les bornes des gains adaptatifs sont précisément définies. On rappelle qu'un des défauts majeurs des modifications robustes traditionnelles est qu'elles agissent sur l'ensemble

d'un contrôleur. Pour des états n'ayant pas les mêmes ordres de grandeur, cela dégrade les performances du quadricoptère. Or, le comportement expérimental de l'opérateur de projection est prévisible et chaque borne est ajustée individuellement. Ainsi, si les bornes sont adéquatement fixées, les performances du quadricoptère ne sont presque pas altérées.

De plus, la méthode proposée pour l'obtention des bornes permet de comprendre leurs effets sur les performances du système, car la construction de Ω se fait en fonction de spécifications physiques pour les performances du quadricoptère. Pour les modifications robustes traditionnelles, il faut choisir une valeur arbitraire pour la constante d'oubli. On ne sait pas *a priori* si cette constante stabilisera tous les gains, ni comment elle va influencer les performances. Grâce à sa méthode de conception basée sur les notions de commande robuste, l'opérateur de projection ne souffre pas du même désavantage que les autres modifications robustes.

Enfin, un dernier avantage de l'opérateur de projection est que la saturation des gains donne une information importante sur le système. En effet, si tous les gains ont atteint leur point de saturation, cela suggère que le quadricoptère opère dans des conditions dangereuses. Il serait alors possible de concevoir un système d'atterrissage automatique lorsque trop de gains ont saturé. Ce genre de raisonnement de haut niveau n'est pas possible avec les modifications robustes traditionnelles.

6.5 Conclusion du chapitre

Dans ce chapitre, nous avons fait la preuve de concept concernant la synthèse d'une MRAC avec l'opérateur de projection. Les résultats expérimentaux ont permis de montrer le bon fonctionnement de la commande adaptative en présence d'incertitudes non paramétriques mal-assorties. De plus, nous avons pu démontrer la pertinence de l'algorithme d'optimisation qui permet d'obtenir les bornes minimales et maximales pour les gains adaptatifs. Enfin, les résultats expérimentaux ont démontré que cette modification robuste est plus performante que les méthodes plus traditionnelles.

CHAPITRE 7 CONCLUSION

7.1 Discussion et contributions

Le principal objectif de ce mémoire était d'effectuer la synthèse et l'implémentation d'une commande adaptative robuste par modèle de référence sur un quadricoptère dans le but de compenser des incertitudes paramétriques et non paramétriques assorties et mal-assorties. Pour y parvenir, nous avons commencé par effectuer une modélisation complète de la dynamique non linéaire du quadricoptère. Ensuite, nous avons simplifié ce modèle en le linéarisant en un point d'équilibre, celui du drone en vol stationnaire. À ce modèle linéaire, nous avons ajouté toutes les perturbations et incertitudes pouvant affecter le quadricoptère. L'inclusion progressive des incertitudes et des perturbations a permis de déterminer l'expression mathématique exacte pour que la commande adaptative puisse les compenser.

En première partie, nous avons choisi de cibler les incertitudes paramétriques et les perturbations non paramétriques assorties. En utilisant des notions de la stabilité de Lyapunov, nous avons démontré que la commande adaptative était stable et qu'elle convergeait de façon asymptotique vers son point d'équilibre garantissant ainsi le rejet de ces perturbations. Ensuite, nous avons vérifié le bon fonctionnement du contrôleur en présence de ces incertitudes. Les tests expérimentaux ont permis de montrer que la commande adaptative compense bien ces perturbations, mieux que les algorithmes de commande classique. En présence d'une masse supplémentaire ou lors d'une panne de moteur, la MRAC permet au quadricoptère de se rétablir plus rapidement et de minimiser l'impact négatif de la perturbation.

En seconde partie, nous avons procédé à l'ajout de modifications robustes pour compenser des incertitudes non paramétriques mal-assorties. En effet, sans ces modifications, le phénomène de dérive des gains pouvait affecter le quadricoptère. Si elle n'est pas compensée, cette dérive peut causer de l'instabilité et mener à l'écrasement de l'appareil. Ainsi, trois modifications robustes traditionnelles ont été déployées sur le quadricoptère : la modification de la zone morte, la modification–σ et la modification–ϵ. Toutefois, lorsque testées en laboratoire, nous avons démontré plusieurs lacunes importantes dues à ces modifications. D'abord, l'effet de dérive des gains n'est pas systématiquement annulé. Ensuite, ces modifications ne permettent pas un ajustement assez précis. Ainsi, il faut presque toujours que les modifications soient trop fortes pour assurer une stabilisation adéquate de quelques gains. Cela a pour effet de dégrader les performances du quadricoptère tout en nuisant à l'objectif de la commande adaptative. Nous ne pouvons recommander leur implémentation sur un quadricoptère.

En troisième partie, nous avons considéré une autre modification robuste. À la suite de l'échec des modifications robustes traditionnelles pour le quadricoptère, nous avons choisi d'utiliser la MRAC construite à l'aide de l'opérateur de projection. Cet opérateur permet de définir directement des bornes pour les gains adaptatifs. Or, l'obtention des bornes est un défi important qui est peu traité dans la littérature scientifique. Pour pallier ce problème, nous avons développé un algorithme d'optimisation permettant d'obtenir ces bornes. Cet algorithme fait le lien entre deux domaines de la théorie de la commande : la commande adaptative et la commande robuste. En effet, nous avons déterminé un ensemble du plan complexe où les pôles du système doivent rester. Cet ensemble est ensuite « protégé » en utilisant des applications gardiennes afin d'obtenir les bornes optimales des gains adaptatifs pour que les pôles ne sortent pas de cet ensemble. Nous avons effectué ce calcul grâce à la μ–analyse. Notre algorithme d'optimisation permet d'obtenir les bornes de gains de façon systématique.

Pour finir, la MRAC construite avec l'opérateur de projection a été testée en laboratoire. Les résultats obtenus ont permis de montrer que l'opérateur n'est pas affecté par les mêmes désavantages que les modifications robustes traditionnelles. Les performances du quadricoptère ne sont presque pas affectées et l'opérateur ne va pas à l'encontre de l'objectif de la commande adaptative. Nous avons montré que l'opérateur de projection est une méthode robuste et efficace et qu'elle peut être utilisée en combinaison avec la MRAC pour rejeter tous les types de perturbations et d'incertitudes qui peuvent affecter le système. Nous recommandons son utilisation plus fréquente sur des quadricoptères.

7.2 Recherches futures

Le choix de l'ensemble du plan complexe protégé par l'application gardienne est primordial pour obtenir des bornes de gains qui ne sont ni trop permissives, ni trop restrictives. Or, ce projet de maîtrise ne s'est pas attardé sur ce détail. En effet, nous avons préféré démontrer le fonctionnement de la méthode plutôt que de considérer la définition idéale de cet ensemble. En nous limitant à un ensemble choisi de façon plus ou moins arbitraire, nous ne pouvons pas confirmer que les bornes obtenues pour les gains adaptatifs sont les bornes optimales.

Pour obtenir un ensemble idéal, il faut avoir une meilleure compréhension des effets des perturbations et des incertitudes sur les pôles du système. Cela pourrait être à la base d'un second projet de recherche. En sachant comment ces incertitudes affectent le système, il serait alors possible de définir un meilleur ensemble. Combinée avec notre algorithme d'obtention des bornes des gains adaptatifs, on pourrait obtenir un opérateur de projection plus perfor-

mant. Nous recommandons donc l'étude des pôles du système influencés par les incertitudes comme prochaine étape à ce projet de recherche.

RÉFÉRENCES

[1] R. C. Dorf, *Modern Control Systems*. Paddison-Wesley Pub. Co., 1967.

[2] H. P. Whitaker, J. Yamron et A. Kezer, *Design of model-reference adaptive control systems for aircraft*. Massachusetts Institute of Technology, Instrumentation Laboratory, 1958.

[3] P. Parks, "Liapunov redesign of model reference adaptive control systems," *IEEE Transactions on Automatic Control*, vol. 11, n°. 3, p. 362–367, July 1966.

[4] Z. T. Dydek, A. M. Annaswamy et E. Lavretsky, "Adaptive control and the NASA X-15-3 flight revisited," *IEEE Control Systems*, vol. 30, n°. 3, p. 32–48, 2010.

[5] K. J. Åström et B. Wittenmark, *Adaptive Control*, 2e éd. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1994.

[6] B. Egardt, *Stability of Adaptive Controllers*, M. Thoma et A. V. Balakrishnan, édit. Berlin, Heidelberg : Springer-Verlag, 1979.

[7] R. E. Kálmán, "On the general theory of control systems," dans *Proceedings First International Conference on Automatic Control, Moscow, USSR*, 1960.

[8] T. Hamel *et al.*, "Dynamic modelling and configuration stabilization for an X4-flyer." *IFAC Proceedings Volumes*, vol. 35, n°. 1, p. 217 – 222, 2002, 15th IFAC World Congress.

[9] P. Pounds *et al.*, "Towards dynamically-favourable quad-rotor aerial robots," dans *Australasian Conference on Robotics and Automation 2004*. Australian Robotics & Automation Association, 2004.

[10] G. Hoffmann *et al.*, "Quadrotor helicopter flight dynamics and control : Theory and experiment," dans *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2004, p. 64 – 84.

[11] P. Pounds, R. Mahony et P. Corke, "Modelling and control of a large quadrotor robot," *Control Engineering Practice*, vol. 18, n°. 7, p. 691 – 699, 2010, special Issue on Aerial Robotics.

[12] R. Mahony, V. Kumar et P. Corke, "Multirotor aerial vehicles : Modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, n°. 3, p. 20–32, Sept 2012.

[13] M. Bangura, R. Mahony *et al.*, "Nonlinear dynamic modeling for high performance control of a quadrotor," dans *Australasian Conference on Robotics and Automation 2012*. Australian Robotics and Automation Association, 2012.

[14] D. Lee *et al.*, “Robust tracking control of an underactuated quadrotor aerial-robot based on a parametric uncertain model,” dans *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. IEEE, 2009, p. 3187–3192.

[15] 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, p. 6716–6729.

[16] A. Mokhtari et A. Benallegue, “Dynamic feedback controller of euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle,” dans *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*, vol. 3. IEEE, 2004, p. 2359–2366.

[17] L. Yu et J. Chu, “An LMI approach to guaranteed cost control of linear uncertain time-delay systems,” *Automatica*, vol. 35, n°. 6, p. 1155–1159, 1999.

[18] V. Stepanyan et K. Kalmanje, “Input and output performance of M-MRAC in the presence of bounded disturbances,” dans *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 7688–7709.

[19] H. K. Khalil, *Nonlinear systems ; 3rd ed.* Upper Saddle River, New Jersey : Prentice-Hall, 2002.

[20] K. S. Narendra et L. S. Valavani, “Direct and indirect model reference adaptive control,” *Automatica*, vol. 15, n°. 6, p. 653–664, 1979.

[21] P. A. Ioannou et J. Sun, *Robust adaptive control*. Upper Saddle River, New Jersey : Prentice-Hall, 1996, vol. 1.

[22] K. S. Narendra et A. M. Annaswamy, *Stable adaptive systems*. Englewood Cliffs, New Jersey : Prentice-Hall, 1989.

[23] E. Lavretsky et K. A. Wise, “Robust adaptive control,” dans *Robust and adaptive control*. Springer, 2013, p. 317–353.

[24] P. Osburn, *New Developments in the Design of Model Reference Adaptive Control Systems*, ser. IAS papers. Institute of the Aerospace Sciences, 1961. [En ligne]. Disponible : https://books.google.ca/books?id=i51rYgEACAAJ

[25] M. A. Duarte et K. S. Narendra, “A new approach to model reference adaptive control,” *International Journal of Adaptive Control and Signal Processing*, vol. 3, n°. 1, p. 53–73, 1989.

[26] J.-J. Slotine et W. Li, “Adaptive robot control : A new perspective,” dans *Decision and Control, 1987. 26th IEEE Conference on*, vol. 26. IEEE, 1987, p. 192–198.

[27] E. Lavretsky, “Combined/composite model reference adaptive control,” *IEEE Transactions on Automatic Control*, vol. 54, n°. 11, p. 2692–2697, 2009.

[28] M. Achtelik *et al.*, “Adaptive control of a quadcopter in the presence of large/complete parameter uncertainties,” dans *Infotech@ Aerospace 2011*, 2011, p. 1485–1500.

[29] A. Chamseddine *et al.*, “Model reference adaptive fault tolerant control of a quadrotor UAV,” dans *Infotech@ Aerospace 2011*, 2011, p. 1606–1619.

[30] Z. T. Dydek, A. M. Annaswamy et E. Lavretsky, “Adaptive control of quadrotor UAVs : A design trade study with flight evaluations,” *IEEE Transactions on control systems technology*, vol. 21, n°. 4, p. 1400–1406, 2013.

[31] A. L’Afflitto, R. B. Anderson et K. Mohammadi, “An introduction to nonlinear robust control for unmanned quadrotor aircraft : How to design control algorithms for quadrotors using sliding mode control and adaptive control techniques [focus on education],” *IEEE Control Systems*, vol. 38, n°. 3, p. 102–121, 2018.

[32] B. Peterson et K. Narendra, “Bounded error adaptive control,” *IEEE Transactions on Automatic Control*, vol. 27, n°. 6, p. 1161–1168, 1982.

[33] J.-J. Slotine et J. Coetsee, “Adaptive sliding controller synthesis for non-linear systems,” *International Journal of Control*, vol. 43, n°. 6, p. 1631–1651, 1986.

[34] P. A. Ioannou et P. V. Kokotović, *Adaptive systems with reduced models.* Springer-Verlag New York, 1983, vol. 68.

[35] L. Hsu et R. Costa, “Bursting phenomena in continuous-time adaptive systems with a σ-modification,” *IEEE Transactions on Automatic Control*, vol. 32, n°. 1, p. 84–86, 1987.

[36] 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.

[37] C. Nicol, C. Macnab et A. Ramirez-Serrano, “Robust adaptive control of a quadrotor helicopter,” *Mechatronics*, vol. 21, n°. 6, p. 927–938, 2011.

[38] C. Coza et C. J. Macnab, “A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization,” dans *Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American.* IEEE, 2006, p. 454–458.

[39] J.-B. Pomet et L. Praly, “Adaptive nonlinear regulation : Estimation from the lyapunov equation,” *IEEE Transactions on automatic control*, vol. 37, n°. 6, p. 729–740, 1992.

[40] Z. Cai, M. S. de Queiroz et D. M. Dawson, “A sufficiently smooth projection operator,” *IEEE Transactions on Automatic Control*, vol. 51, n°. 1, p. 135–139, 2006.

[41] K. Zhou *et al.*, *Robust and optimal control.* Upper Saddle River, New Jersey : Prentice-Hall, 1996, vol. 40.

[42] L. Saydy, A. L. Tits et E. H. Abed, "Guardian maps and the generalized stability of parametrized families of matrices and polynomials," *Mathematics of Control, Signals and Systems*, vol. 3, n°. 4, p. 345–371, 1990.

[43] 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, n°. 12, p. 2193–2201, 2012.

[44] J.-L. Lin et S.-J. Chen, "LFT approach to robust $\mathcal{D}$–stability bounds of uncertain linear singular systems," *IEE Proceedings-Control Theory and Applications*, vol. 145, n°. 2, p. 127–134, 1998.

[45] S. Bouabdallah, A. Noth et R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," dans *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2004, p. 2451–2456.

[46] G. J. Leishman, *Principles of helicopter aerodynamics with CD extra.* Cambridge university press, 2006.

[47] D.-T. Nguyen, D. Saussie 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)*, 2018, p. 385–393.

[48] 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. [En ligne]. Disponible : http://dx.doi.org/10.1007/978-3-319-26054-9_23

[49] K. M. Thu et A. Gavrilov, "Designing and modeling of quadcopter control system using L1 adaptive control," *Procedia Computer Science*, vol. 103, p. 528–535, 2017.

[50] R. E. Kálmán *et al.*, "Contributions to the theory of optimal control," *Bol. Soc. Mat. Mexicana*, vol. 5, n°. 2, p. 102–119, 1960.

[51] V. Natarajan et J. Bentsman, "Adaptive projection-based observers and L1 adaptive controllers for infinite-dimensional systems with full-state measurement," *IEEE Transactions on Automatic Control*, vol. 59, n°. 3, p. 585–598, 2014.

[52] 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.

[53] N. Hovakimyan et C. Cao, *L1 adaptive control theory : guaranteed robustness with fast adaptation.* Siam, 2010, vol. 21.

[54] Z. T. Dydek, “Adaptive control of unmanned aerial systems,” Thèse de doctorat, Massachusetts institute of Technology, 2010.

[55] B. R. Barmish et E. Jury, “New tools for robustness of linear systems,” *IEEE Transactions on Automatic Control*, vol. 39, n°. 12, p. 2525–2525, 1994.

[56] C. Audet et J. E. Dennis Jr, “Mesh adaptive direct search algorithms for constrained optimization,” *SIAM Journal on optimization*, vol. 17, n°. 1, p. 188–217, 2006.

[57] C. Audet et W. Hare, *Derivative-free and blackbox optimization.* Springer, 2017.

[58] K. B. Petersen, M. S. Pedersen *et al.*, “The matrix cookbook,” *Technical University of Denmark*, vol. 7, n°. 15, p. 510, 2008.

[59] G. J. Balas *et al.*, “μ-analysis and synthesis toolbox,” 1993.

ANNEXE A DYNAMIQUE DU QUADRICOPTÈRE

A.1 Équations scalaires de la dynamique du drone

Dans cet annexe, nous donnons les 12 équations scalaires de la dynamique du drone. Nous rappelons que les états sont données par :

$$\mathbf{x} = \begin{bmatrix} x & y & z & u & v & w & \phi & \theta & \psi & p & q & r \end{bmatrix}^\top. \tag{A.1}$$

L'équation de navigation (3.4) permet d'obtenir les trois premières équations

$$\begin{aligned} \dot{x} &= u\cos\theta\cos\psi + v\cos\theta\sin\psi - w\sin\theta, \\ \dot{y} &= u\left(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi\right) + v\left(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi\right) \\ &\quad + w\sin\phi\cos\theta, \\ \dot{z} &= u\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right) + v\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right) \\ &\quad + w\cos\phi\cos\theta. \end{aligned} \tag{A.2}$$

On inverse l'équation d'Euler (3.5) pour obtenir les trois équations suivantes :

$$\begin{aligned} \dot{\phi} &= p + q\sin\phi\tan\theta + r\cos\phi\tan\theta, \\ \dot{\theta} &= q\cos\phi - r\sin\phi, \\ \dot{\psi} &= q\left(\frac{\sin\phi}{\cos\theta}\right) + r\left(\frac{\cos\phi}{\cos\theta}\right). \end{aligned} \tag{A.3}$$

L'équation de forces dans le repère objet (3.8) donne :

$$\begin{aligned} \dot{u} &= qw - rv + g_0\sin\theta, \\ \dot{v} &= ru - pw - g_0\sin\phi\cos\theta, \\ \dot{w} &= pv - qu - g_0\cos\phi\cos\theta + \frac{F_z}{m}. \end{aligned} \tag{A.4}$$

alors que dans le repère inertiel, l'équation (3.14) donne :

$$\begin{aligned}
\ddot{x} &= (\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi)\,\frac{F_z}{m},\\
\ddot{y} &= (\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi)\,\frac{F_z}{m},\\
\ddot{z} &= (\cos\theta\cos\phi)\,\frac{F_z}{m} - g_0.
\end{aligned} \tag{A.5}$$

Enfin, l'équation de moments dans le repère objet (3.18) donne :

$$\begin{aligned}
\dot{p} &= \frac{I_{zz} - I_{yy}}{I_{xx}} qr + \frac{M_x}{I_{xx}},\\
\dot{q} &= \frac{I_{xx} - I_{zz}}{I_{yy}} pr + \frac{M_y}{I_{yy}},\\
\dot{r} &= \frac{I_{yy} - I_{xx}}{I_{zz}} pq + \frac{M_z}{I_{zz}}.
\end{aligned} \tag{A.6}$$

A.2 Projection des aires pour la force de traînée

La force de traînée exercée par le corps du quadricoptère s'oppose toujours au mouvement de celui-ci. Ainsi, l'aire projetée selon chacun des axes du repère objet $\mathcal{F}_b$ détermine la superficie qui nuit au mouvement. Autrement dit, plus le quadricoptère est penché selon un des axes, plus le mouvement selon cet axe sera difficile. La figure A.1 permet de comprendre intuitivement le principe de projection des aires.

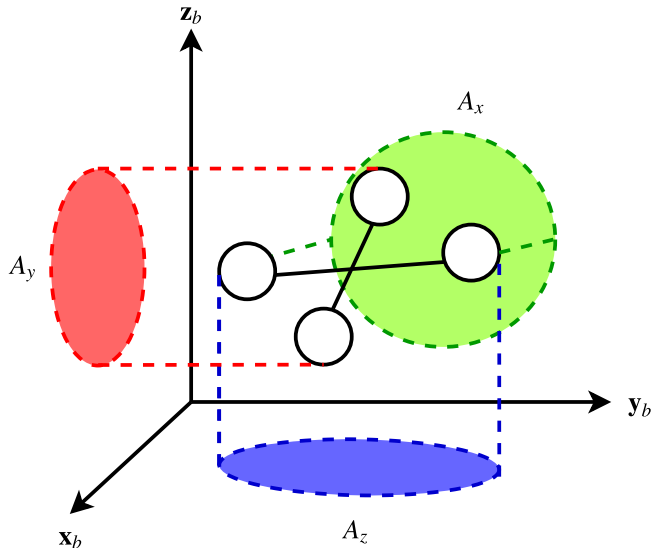


Figure A.1 Projection des aires selon les axes du repère objet

A.3 Identification des coefficients de poussée et de traînée des moteurs

Pour faire l'identification des coefficients de poussée et de traînée des moteurs du quadricoptère, on utilise le RCBenchmark. Ce banc de test permet de faire tourner le moteur à des vitesses croissantes et d'enregistrer des informations pertinentes : vitesse de rotation, poussée des moteurs et couple moteur. Comme l'obtention des données peut varier d'une expérience à l'autre, le test est effectué trois fois. Les données expérimentales sont fournies dans le tableau A.1.

Tableau A.1 Valeurs expérimentales pour la poussée T et le couple Q du moteur de l'AscTec Pelican

Test #1			Test #2			Test #3		
ω (rad/s)	T (N)	Q (Nm)	ω(rad/s)	T (N)	Q (Nm)	ω (rad/s)	T (N)	Q (Nm)
0	0	0	0	0	0	0	0	0
144.5132	0.2504	0.0043	146.1888	0.2530	0.0061	146.5030	0.2717	0.0054
288.5029	1.0584	0.0185	286.9321	1.0377	0.0200	288.3982	1.0590	0.0192
387.4631	2.0243	0.0358	388.3008	1.9783	0.0371	387.8820	1.9953	0.0357
458.9867	2.8745	0.0521	458.9867	2.8437	0.0528	458.0442	2.8775	0.0523
513.6504	3.6425	0.0653	514.1740	3.6600	0.0671	513.7551	3.7523	0.0670
572.6076	4.6755	0.0843	572.2934	4.6240	0.0849	571.2463	4.6374	0.0848
638.3716	5.9111	0.1071	635.6489	5.9164	0.1079	632.0884	5.9854	0.1069

Pour déterminer les coefficients, on fait une approximation quadratique pour la poussée T et le couple du moteur Q :

$$T = k_T\omega^2, \quad \text{et} \quad Q = k_D\omega^2. \tag{A.7}$$

Ainsi, pour chacun des tests expérimentaux, on détermine les coefficients qui sont les plus appropriés et qui suivent la structure $y = ax^2$. On utilise les fonctions de MATLAB pour obtenir les résultats suivants :

$$\begin{aligned} k_{T,1} &= 1.430 \cdot 10^{-5}, & k_{D,1} &= 2.650 \cdot 10^{-7}, \\ k_{T,2} &= 1.427 \cdot 10^{-5}, & k_{D,2} &= 2.668 \cdot 10^{-7}, \\ k_{T,3} &= 1.449 \cdot 10^{-5}, & k_{D,3} &= 2.673 \cdot 10^{-7}, \\ \overline{k}_T &= 1.435 \cdot 10^{-5}, & \overline{k}_D &= 2.664 \cdot 10^{-7}. \end{aligned} \tag{A.8}$$

Les valeurs retenues pour les coefficients sont $k_T = 1.435 \cdot 10^{-5}$ N·s^2/rad^2 et $k_D = 2.664 \cdot 10^{-7}$ Nm·s^2/rad^2.

A.4 Système linéaire : matrices d'état

Les valeurs des matrices d'états sont données ci-dessous

$$\mathbf{A} = \left[\begin{array}{c:c:ccc:c} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \multicolumn{3}{c}{\mathbf{0}_{3\times3}} & \mathbf{0}_{3\times3} \\ \hdashline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & 0 & g_0 & 0 & \mathbf{0}_{3\times3} \\ & & -g_0 & 0 & 0 & \\ & & 0 & 0 & 0 & \\ \hdashline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \multicolumn{3}{c}{\mathbf{0}_{3\times3}} & \mathbf{I}_{3\times3} \\ \hdashline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \multicolumn{3}{c}{\mathbf{0}_{3\times3}} & \mathbf{0}_{3\times3} \end{array}\right], \quad \mathbf{B} = \left[\begin{array}{cccc} \multicolumn{4}{c}{\mathbf{0}_{3\times4}} \\ \hdashline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ \hdashline \multicolumn{4}{c}{\mathbf{0}_{3\times4}} \\ \hdashline 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{array}\right], \tag{A.9}$$

$$\mathbf{C} = \left[\begin{array}{c:c:ccc:c} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \multicolumn{3}{c}{\mathbf{0}_{3\times3}} & \mathbf{0}_{3\times3} \\ \hdashline \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & 1 & \mathbf{0}_{1\times3} \end{array}\right], \qquad \mathbf{D} = [\mathbf{0}_4]\,.$$

ANNEXE B SYNTHÈSE DES LOIS DE COMMANDE CLASSIQUES ET ADAPTATIVES

B.1 Dérivée de la fonction de Lyapunov pour la MRAC

La fonction candidate de Lyapunov retenue est :

$$V\left(\Delta\boldsymbol{\varepsilon}, \Delta\hat{\boldsymbol{\theta}}\right) = \boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\boldsymbol{\varepsilon} + \mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top \boldsymbol{\Gamma}_\theta^{-1} \Delta\hat{\boldsymbol{\theta}} \boldsymbol{\Lambda}\right). \tag{B.1}$$

Les éléments qui dépendent du temps sont $\Delta\boldsymbol{\varepsilon}$ et $\Delta\hat{\boldsymbol{\theta}}$. La dérivée temporelle du premier terme donne :

$$\begin{aligned} \frac{\partial}{\partial t}\left(\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\boldsymbol{\varepsilon}\right) &= \Delta\dot{\boldsymbol{\varepsilon}}^\top \mathbf{P}_l \Delta\boldsymbol{\varepsilon} + \Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\dot{\boldsymbol{\varepsilon}}, \\ &= 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\dot{\boldsymbol{\varepsilon}}, \end{aligned} \tag{B.2}$$

en utilisant le fait que $\mathbf{P}_l = \mathbf{P}_l^\top$ et que les termes sont des scalaires. On injecte l'expression de $\Delta\dot{\boldsymbol{\varepsilon}}$ donnée par l'équation (4.19) :

$$\begin{aligned} 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\dot{\boldsymbol{\varepsilon}} &= 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \left(\mathbf{A}_{ref}\Delta\boldsymbol{\varepsilon} + \mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top \boldsymbol{\omega}\right), \\ &= 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{A}_{ref}\Delta\boldsymbol{\varepsilon} + 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top \boldsymbol{\omega}, \end{aligned} \tag{B.3}$$

On peut simplifier le premier terme grâce à l'équation de Lyapunov (4.21). On a en effet :

$$\begin{aligned} -\Delta\boldsymbol{\varepsilon}^\top \mathbf{Q}_l \Delta\boldsymbol{\varepsilon} &= \Delta\boldsymbol{\varepsilon}^\top \left(\mathbf{P}_l \mathbf{A}_{ref} + \mathbf{A}_{ref}^\top \mathbf{P}_l\right) \Delta\boldsymbol{\varepsilon}, \\ &= \Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{A}_{ref}\Delta\boldsymbol{\varepsilon} + \Delta\boldsymbol{\varepsilon}^\top \mathbf{A}_{ref}^\top \mathbf{P}_l \Delta\boldsymbol{\varepsilon}, \\ &= 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{A}_{ref}\Delta\boldsymbol{\varepsilon}, \end{aligned} \tag{B.4}$$

en utilisant à nouveau que $\mathbf{P}_l$ est symétrique et que les quantités manipulées sont scalaires. La dérivée du premier terme de $\dot{V}$ est donc :

$$\frac{\partial}{\partial t}\left(\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\boldsymbol{\varepsilon}\right) = -\Delta\boldsymbol{\varepsilon}^\top \mathbf{Q}_l \Delta\boldsymbol{\varepsilon} + 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top \boldsymbol{\omega}. \tag{B.5}$$

On s'intéresse maintenant à la dérivée du second terme de V. On peut donner la dérivée de

la trace d'une multiplication matricielle en se référant à [58]. Plus particulièrement, on a

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{X}}\left(\mathrm{tr}\left(\mathbf{A}\mathbf{X}\mathbf{B}\mathbf{X}^\top\mathbf{C}\right)\right) = \mathbf{A}^\top\mathbf{C}^\top\mathbf{X}\mathbf{B}^\top + \mathbf{C}\mathbf{A}\mathbf{X}\mathbf{B}, \tag{B.6}$$

où toutes les matrices ont des dimensions quelconques et valides. Pour notre système, on a : $\mathbf{A} = \mathbf{I}$, $\mathbf{B} = \boldsymbol{\Gamma}_\theta^{-1}$, $\mathbf{C} = \boldsymbol{\Lambda}$ et $\mathbf{X} = \Delta\hat{\boldsymbol{\theta}}^\top$. Donc, la dérivée est donnée par

$$\begin{aligned}\frac{\partial}{\partial t}\left(\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\Gamma}_\theta^{-1}\Delta\hat{\boldsymbol{\theta}}\boldsymbol{\Lambda}\right)\right) &= \frac{\partial}{\partial\Delta\hat{\boldsymbol{\theta}}^\top}\left(\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\Gamma}_\theta^{-1}\Delta\hat{\boldsymbol{\theta}}\boldsymbol{\Lambda}\right)\right)\frac{\partial}{\partial t}\left(\Delta\hat{\boldsymbol{\theta}}^\top\right), \\ &= \left\{\boldsymbol{\Lambda}^\top\Delta\hat{\boldsymbol{\theta}}^\top\left(\boldsymbol{\Gamma}_\theta^{-1}\right)^\top + \boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\Gamma}_\theta^{-1}\right\}\Delta\dot{\hat{\boldsymbol{\theta}}}.\end{aligned} \tag{B.7}$$

Avec les conditions de symétrie sur $\boldsymbol{\Lambda}$ et $\boldsymbol{\Gamma}_\theta$ posées au chapitre 4, on a :

$$\frac{\partial}{\partial t}\left(\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\Gamma}_\theta^{-1}\Delta\hat{\boldsymbol{\theta}}\boldsymbol{\Lambda}\right)\right) = 2\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\Gamma}_\theta^{-1}\Delta\dot{\hat{\boldsymbol{\theta}}}. \tag{B.8}$$

La dérivée temporelle de V est donc

$$\dot{V} = -\Delta\boldsymbol{\varepsilon}^\top\mathbf{Q}_l\Delta\boldsymbol{\varepsilon} + 2\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\omega} + 2\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\Gamma}_\theta^{-1}\Delta\dot{\hat{\boldsymbol{\theta}}}. \tag{B.9}$$

Pour obtenir une forme simplifiée de cette équation, on considère la propriété suivante tirée de [58]. Pour deux vecteurs co-dimensionnels $\mathbf{A}$ et $\mathbf{B}$, l'identité suivante est vraie : $\mathbf{A}^\top\mathbf{B} = \mathrm{tr}\left(\mathbf{B}\mathbf{A}^\top\right)$. De plus, la trace d'une multiplication matricielle est cyclique : $\mathrm{tr}\left(\mathbf{ABC}\right) = \mathrm{tr}\left(\mathbf{BCA}\right) = \mathrm{tr}\left(\mathbf{CAB}\right)$. On peut donc simplifier l'expression pour obtenir

$$\begin{aligned}\dot{V} &= -\Delta\boldsymbol{\varepsilon}^\top\mathbf{Q}_l\Delta\boldsymbol{\varepsilon} + 2\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\omega} + 2\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\Gamma}_\theta^{-1}\Delta\dot{\hat{\boldsymbol{\theta}}}, \\ &= -\Delta\boldsymbol{\varepsilon}^\top\mathbf{Q}_l\Delta\boldsymbol{\varepsilon} + 2\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B}\boldsymbol{\Lambda}\right) + \underbrace{2\mathrm{tr}\left(\boldsymbol{\Gamma}_\theta^{-1}\Delta\dot{\hat{\boldsymbol{\theta}}}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top\right)}_{\text{cyclique : } 2\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\boldsymbol{\Gamma}_\theta^{-1}\Delta\dot{\hat{\boldsymbol{\theta}}}\boldsymbol{\Lambda}\right)}, \\ &= -\Delta\boldsymbol{\varepsilon}^\top\mathbf{Q}_l\Delta\boldsymbol{\varepsilon} + 2\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top\left\{\boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top\mathbf{P}_l\mathbf{B} + \boldsymbol{\Gamma}_\theta^{-1}\Delta\dot{\hat{\boldsymbol{\theta}}}\right\}\boldsymbol{\Lambda}\right).\end{aligned} \tag{B.10}$$

Cette expression est la dérivée simplifiée de la fonction candidate de Lyapunov.

ANNEXE C MODIFICATIONS ROBUSTES TRADITIONNELLES

C.1 Dérivée de la fonction de Lyapunov avec incertitudes non paramétriques mal-assorties

On cherche maintenant à déterminer la dérivée de la fonction candidate de Lyapunov si le système contient des incertitudes non paramétriques mal-assorties en plus des autres incertitudes traitées à l'annexe B.1. On rappelle que la la fonction candidate est donnée par

$$V = \Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\boldsymbol{\varepsilon} + \mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top \boldsymbol{\Gamma}_\theta^{-1} \Delta\hat{\boldsymbol{\theta}}\boldsymbol{\Lambda}\right). \tag{C.1}$$

On dérive les termes qui dépendent du temps. Par comparaison avec le cas précédent, seul le premier terme changera. En effet, la dérivée devient

$$\begin{aligned} \frac{\partial}{\partial t}\left(\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\boldsymbol{\varepsilon}\right) &= 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\dot{\boldsymbol{\varepsilon}}, \\ &= 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \left(\mathbf{A}_{ref}\Delta\boldsymbol{\varepsilon} + \mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top \boldsymbol{\omega} + \boldsymbol{\xi}\right). \end{aligned} \tag{C.2}$$

On applique les mêmes simplifications que celles effectuées à l'annexe B.1. On obtient donc

$$\frac{\partial}{\partial t}\left(\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\boldsymbol{\varepsilon}\right) = -\Delta\boldsymbol{\varepsilon}^\top \mathbf{Q}_l \Delta\boldsymbol{\varepsilon} + 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{B}\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^\top \boldsymbol{\omega} + 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \boldsymbol{\xi}. \tag{C.3}$$

La dérivée de Lyapunov pour le système contenant toutes les formes d'incertitudes et perturbations est :

$$\dot{V} = -\Delta\boldsymbol{\varepsilon}^\top \mathbf{Q}_l \Delta\boldsymbol{\varepsilon} + 2\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \boldsymbol{\xi} + 2\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top \left\{\boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \mathbf{B} + \boldsymbol{\Gamma}_\theta^{-1}\Delta\dot{\hat{\boldsymbol{\theta}}}\right\}\boldsymbol{\Lambda}\right). \tag{C.4}$$

C.2 Borne sur la fonction candidate de Lyapunov

On souhaite borner la fonction candidate de Lyapunov. On rappelle que

$$V = \Delta\boldsymbol{\varepsilon}^\top \mathbf{P}_l \Delta\boldsymbol{\varepsilon} + \mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top \boldsymbol{\Gamma}_\theta^{-1} \Delta\hat{\boldsymbol{\theta}}\boldsymbol{\Lambda}\right). \tag{C.5}$$

Puisque $\boldsymbol{\Lambda} > 0$ et $\boldsymbol{\Gamma}_\theta^{-1} > 0$, alors le terme $\mathrm{tr}\left(\Delta\hat{\boldsymbol{\theta}}^\top \boldsymbol{\Gamma}_\theta^{-1} \Delta\hat{\boldsymbol{\theta}}\boldsymbol{\Lambda}\right) > 0$. En utilisant les propriétés des traces des matrices présentées à l'annexe B.1, on peut réécrire la trace de la façon

suivante :

$$\operatorname{tr}\left(\Delta\hat{\boldsymbol{\theta}}^{\top}\boldsymbol{\Gamma}_{\theta}^{-1}\Delta\hat{\boldsymbol{\theta}}\boldsymbol{\Lambda}\right) = \operatorname{tr}\left(\boldsymbol{\Lambda}\Delta\hat{\boldsymbol{\theta}}^{\top}\boldsymbol{\Gamma}_{\theta}^{-1}\Delta\hat{\boldsymbol{\theta}}\right). \tag{C.6}$$

Selon cette équation, la borne inférieure de la candidate de Lyapunov est donnée par

$$V \geq \lambda_{min}\left(\mathbf{P}_l\right)\left\|\Delta\boldsymbol{\varepsilon}\right\|^2 + \lambda_{min}\left(\boldsymbol{\Lambda}\right)\lambda_{min}\left(\boldsymbol{\Gamma}_{\theta}^{-1}\right)\left\|\Delta\hat{\boldsymbol{\theta}}\right\|^2. \tag{C.7}$$

Et la borne supérieure est donnée par

$$V \leq \lambda_{max}\left(\mathbf{P}_l\right)\left\|\Delta\boldsymbol{\varepsilon}\right\|^2 + \lambda_{max}\left(\boldsymbol{\Lambda}\right)\lambda_{max}\left(\boldsymbol{\Gamma}_{\theta}^{-1}\right)\left\|\Delta\hat{\boldsymbol{\theta}}\right\|^2. \tag{C.8}$$

De plus, la matrice $\boldsymbol{\Lambda}$ représente les pannes de moteurs pouvant survenir en vol. La matrice est diagonale et ses éléments sont strictement compris entre 0 et 1. Donc, on sait que $\lambda_{min}\left(\boldsymbol{\Lambda}\right) \geq 0$ et $\lambda_{max}\left(\boldsymbol{\Lambda}\right) \leq 1$. Ainsi, on peut simplifier les bornes pour obtenir le résultat suivant :

$$\lambda_{min}\left(\mathbf{P}_l\right)\left\|\Delta\boldsymbol{\varepsilon}\right\|^2 \leq V \leq \lambda_{max}\left(\mathbf{P}_l\right)\left\|\Delta\boldsymbol{\varepsilon}\right\|^2 + \lambda_{max}\left(\boldsymbol{\Gamma}_{\theta}^{-1}\right)\left\|\Delta\hat{\boldsymbol{\theta}}\right\|^2. \tag{C.9}$$

Puisque la fonction de Lyapunov est semi-définie négative, on sait que la convergence de l'erreur $\Delta\boldsymbol{\varepsilon}$ est garantie en dehors de l'ensemble $E_0 = \{\Delta\boldsymbol{\varepsilon} \in \mathbb{R}^n \mid \|\Delta\boldsymbol{\varepsilon}\| \leq \varepsilon_0\}$. Toutefois, si $\Delta\boldsymbol{\varepsilon} \in E_0$, alors on ne peut pas garantir que $\dot{V} \leq 0$. Puisque la loi adaptative est $\Delta\dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}_{\theta}\boldsymbol{\omega}\Delta\boldsymbol{\varepsilon}^{\top}\mathbf{P}_l\mathbf{B}$, alors le système continuera de s'adapter, même si l'erreur reste bornée. Ainsi, $\Delta\hat{\boldsymbol{\theta}}$ pourra continuer à croître vers l'infini. Puisque la borne supérieure de la fonction candidate de Lyapunov dépend de $\left\|\Delta\hat{\boldsymbol{\theta}}\right\|$ et que cette norme n'est pas bornée, alors la fonction candidate n'est pas bornée supérieurement.

C.3 Résultats complets pour la modification–σ

Dans la section 5.3, nous avons présenté des résultats de vol en attirant l'attention du lecteur vers les données les plus probantes. Or, pour chacun des vols, il y a 24 gains adaptatifs qui évoluent dans le temps. Cette section présente l'ensemble des résultats pour les vols avec $\sigma = \{0.001,\ 0.01,\ 0.1\}$. Durant les vols, une panne de moteur de 20% survient entre $t = 20$ et $t = 30$ s.

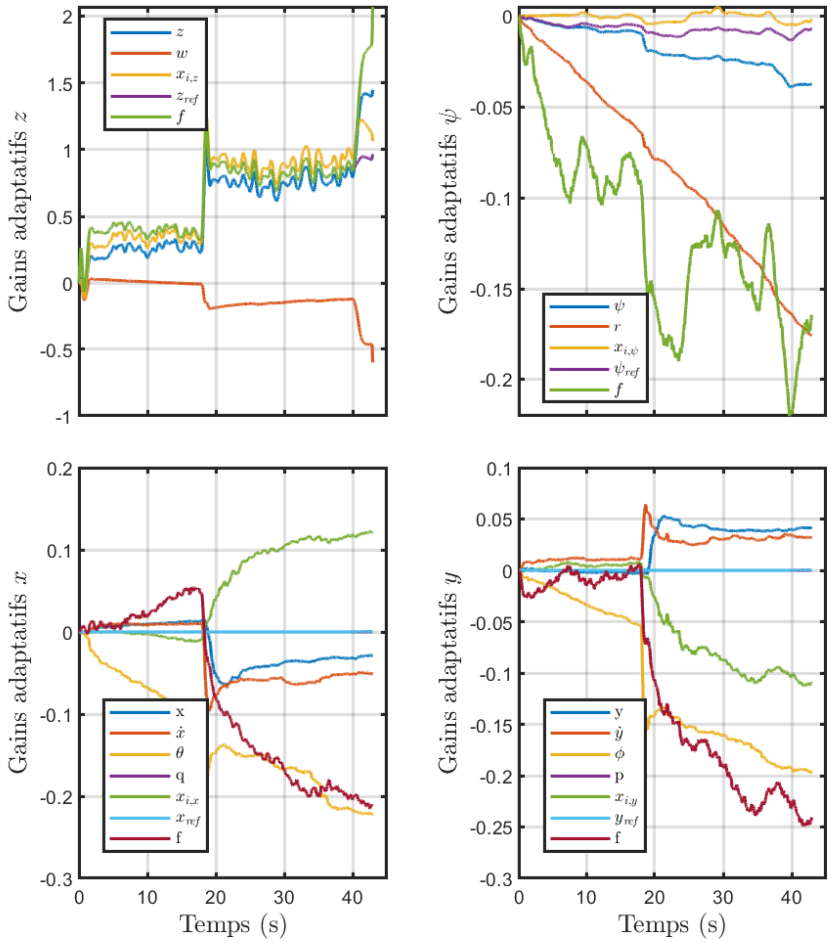


Figure C.1 Résultats expérimentaux pour la modification–σ lorsque $\sigma = 0.001$

La figure C.1 permet de constater que certains des gains adaptatifs continuent de diverger. En particulier, cela survient pour les états r, θ, f_x, ϕ et f_y. On voit que le phénomène d'oubli de l'adaptation est présent pour les états w, x et $\dot{x}$. Enfin, la modification–σ réussit à stabiliser les gains des états y et $\dot{y}$.

La figure C.2 montre encore que certains gains sont stabilisés correctement, par exemple, pour les états y et $\dot{y}$. L'effet de l'oubli de l'apprentissage est encore plus significatif pour les états w, x et $\dot{x}$. Enfin, les gains des états r, ϕ et θ ne sont toujours pas bornés par la modification robuste.

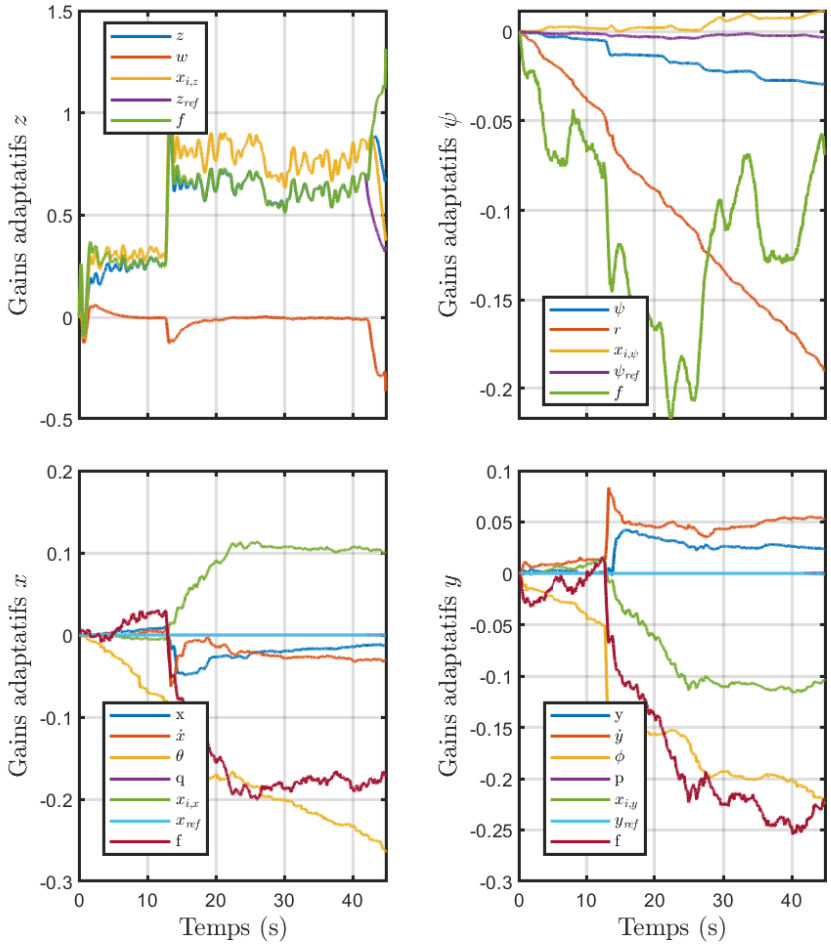


Figure C.2 Résultats expérimentaux pour la modification–σ lorsque $\sigma = 0.01$

Enfin, la figure C.3 donne les résultats lorsque le facteur d'oubli est le plus fort. Malgré tout, certains gains ne se stabilisent pas correctement, entre autres les gains pour r et f_ψ. Cependant, plusieurs gains se stabilisent correctement : ϕ, θ, $x_{i,x}$ et $x_{i,y}$. L'effet de l'oubli de l'apprentissage est aussi très significatif pour les états w, x et $\dot{x}$ et certains gains qui étaient stables pour $\sigma = 0.01$ sont affectés par ce phénomène. Par exemple, les gains associés aux

états y et $\dot{y}$ sont maintenant affectés par le phénomène d'oubli de l'apprentissage.

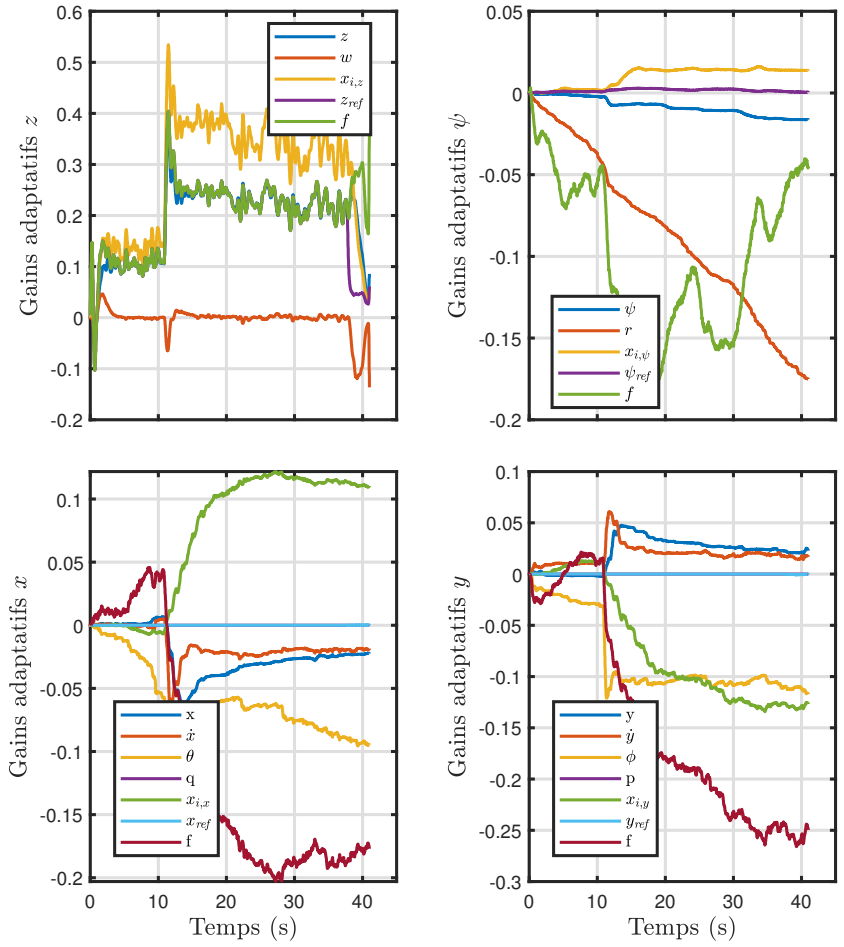


Figure C.3 Résultats expérimentaux pour la modification–σ lorsque $\sigma = 0.1$

C.3.1 Résultats complets pour la modification–ϵ

Dans la section 5.4, nous avons présenté des résultats de vol en attirant l'attention du lecteur vers les données les plus probantes. Or, pour chacun des vols, il y a 24 gains adaptatifs qui évoluent dans le temps. Cette section présente l'ensemble des résultats pour les vols avec $\sigma = \{0.001,\ 0.01,\ 0.1\}$. Durant les vols, une panne de moteur de 20% survient entre $t = 25$ et $t = 35$ s.

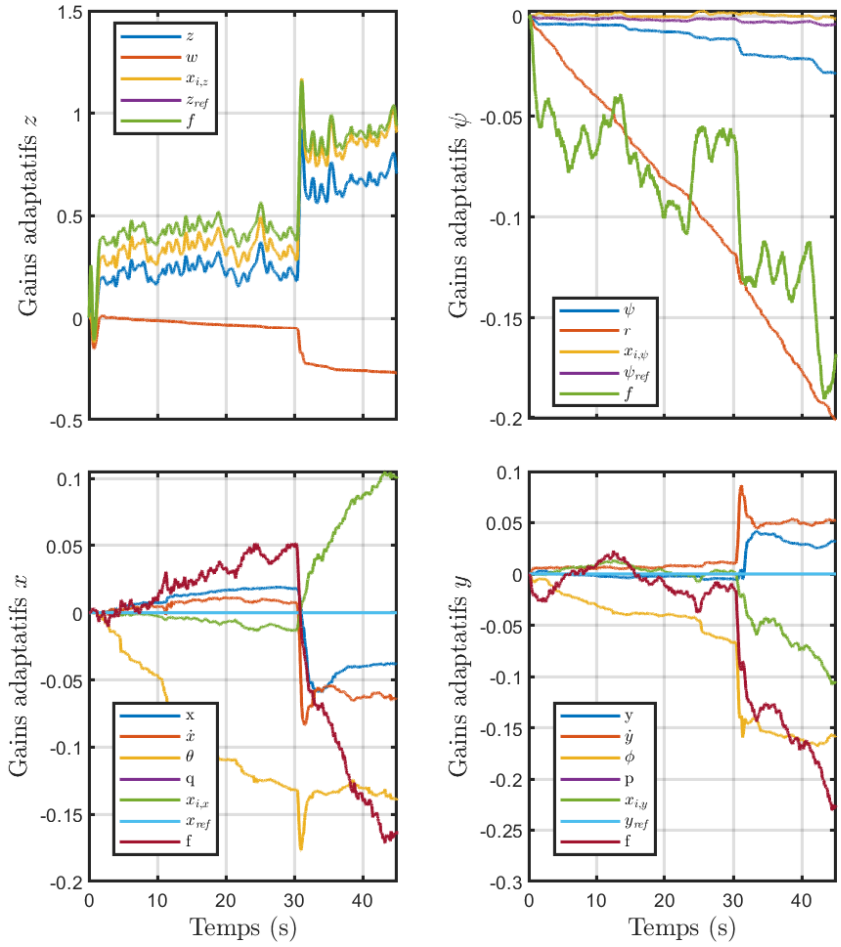


Figure C.4 Résultats expérimentaux pour la modification–ϵ lorsque $\sigma = 0.001$

La figure C.4 permet de constater que certains des gains adaptatifs sont déjà stabilisés, pour les états ϕ et θ par exemple. Toutefois, certains gains continuent de diverger : r, f_x et f_y. Comme discuté, le phénomène d'oubli de l'adaptation n'est pas visible pour la modification–ϵ.

La figure C.5 montre encore que certains gains sont stabilisés efficacement, et d'autres ne le sont pas. Or, il ne semble pas y avoir une grande différence avec la valeur de $\sigma = 0.001$.

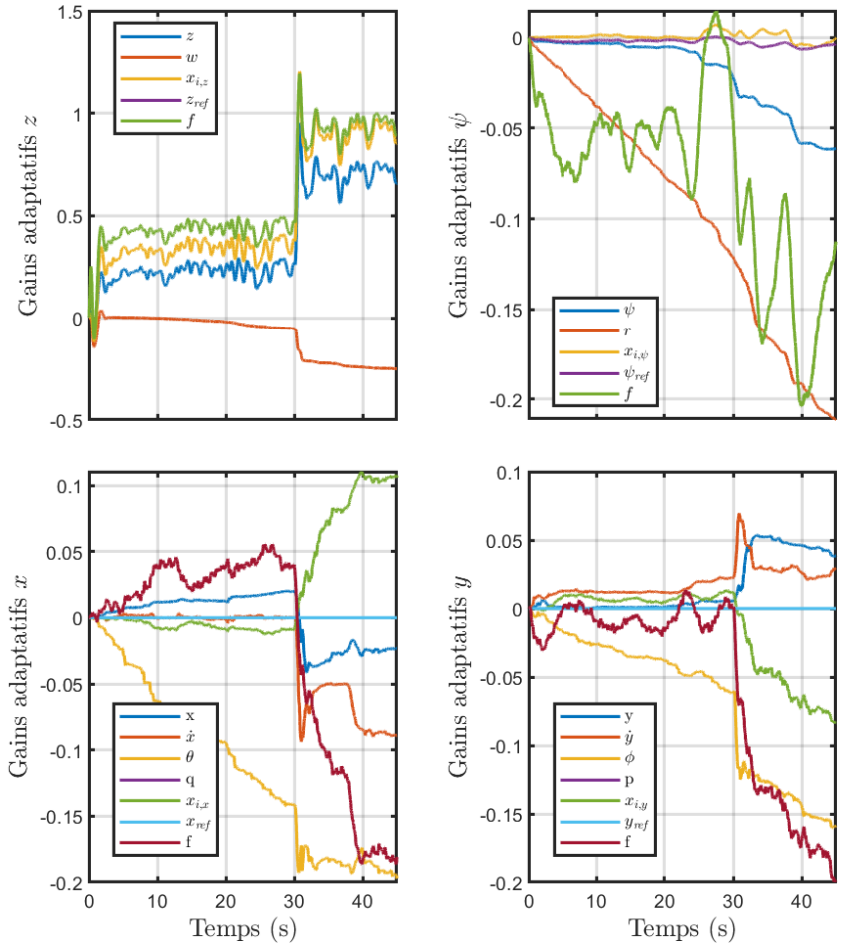


Figure C.5 Résultats expérimentaux pour la modification–ϵ lorsque $\sigma = 0.01$

Enfin, la figure C.6 donne les résultats lorsque le facteur d'oubli est à son plus fort. Malgré tout, certains gains ne se stabilisent pas correctement : r et f_ψ. Toutefois, les gains des états ϕ, θ, $x_{i,x}$ et $x_{i,y}$ se stabilisent correctement. Même pour cette forte valeur de σ, le phénomène d'oubli de l'apprentissage n'est pas observé. Cela confirme donc que la modification–ϵ permet bien de borner les gains tout en minimisant les désavantages que présentait la modification–σ.

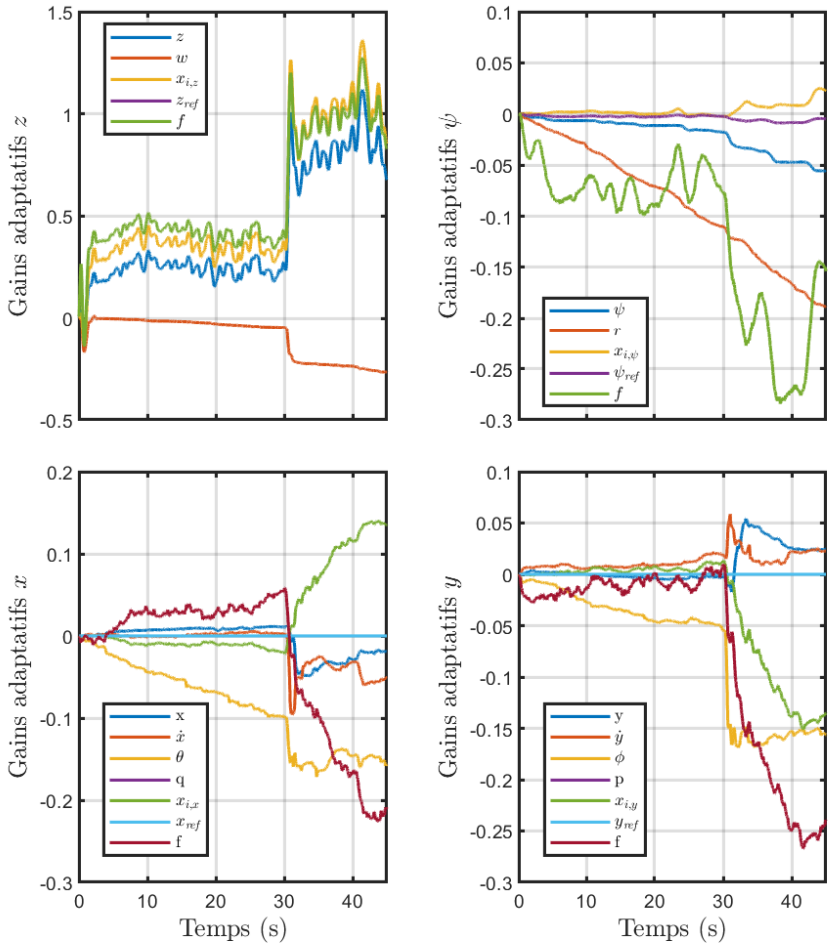


Figure C.6 Résultats expérimentaux pour la modification–ϵ lorsque $\sigma = 0.1$

ANNEXE D OPÉRATEUR DE PROJECTION

D.1 Démonstration des lemmes pour les définitions de base

Démonstration du lemme 6.2.1. Soit $\boldsymbol{\theta}_1,\ \boldsymbol{\theta}_2 \in \Omega_\delta$. Alors, $f(\boldsymbol{\theta}_1) \leq \delta$ et $f(\boldsymbol{\theta}_2) \leq \delta$. De plus, puisque $f(\boldsymbol{\theta})$ est convexe, alors pour $\lambda \in [0,1]$,

$$f\underbrace{(\lambda\boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_2)}_{\boldsymbol{\theta}} \leq \lambda f(\boldsymbol{\theta}_1) + (1-\lambda)\, f(\boldsymbol{\theta}_2) \leq \lambda\delta + (1-\lambda)\delta = \delta. \tag{D.1}$$

Ainsi, $f(\boldsymbol{\theta}) \leq \delta$ et, conséquemment, $\boldsymbol{\theta} \in \Omega_\delta$ ce qui complète la preuve. □

D.2 Démonstration des lemmes pour l'opérateur de projection

Pour la démonstration du lemme 6.2.2, on utilise d'abord le lemme suivant.

Lemme D.2.1. *Soit $f : \mathbb{R}^n \to \mathbb{R}$, une fonction convexe et différentiable. On choisit $0 < \delta < 1$ et on considère le sous-ensemble $\Omega_\delta = \{\boldsymbol{\theta} \in \mathbb{R}^n \,|f(\boldsymbol{\theta}) \leq \delta\}$. Soit $\boldsymbol{\theta}^* \in \Omega_\delta$ tel que $f(\boldsymbol{\theta}^*) < \delta$, i.e., $\boldsymbol{\theta}^*$ est un point intérieur de Ω_δ et ne se situe pas sur la frontière. De plus, soit $\boldsymbol{\theta} \in \Omega_\delta$ tel que $f(\boldsymbol{\theta}) = \delta$, i.e., $\boldsymbol{\theta}$ se situe sur la frontière de Ω_δ. Alors, l'inégalité suivante est valide :*

$$(\boldsymbol{\theta}^* - \boldsymbol{\theta})^\top \nabla f(\boldsymbol{\theta}) \leq 0, \quad \nabla f(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_1} & \cdots & \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_n} \end{bmatrix}^\top \in \mathbb{R}^n, \tag{D.2}$$

où $\nabla f(\boldsymbol{\theta})$ est le gradient de la fonction f évalué au point θ.

Démonstration. Puisque $f(\mathbf{x})$ est convexe, on a alors :

$$f(\lambda\boldsymbol{\theta}^* + (1-\lambda)\boldsymbol{\theta}) \leq \lambda f(\boldsymbol{\theta}^*) + (1-\lambda) f(\boldsymbol{\theta}). \tag{D.3}$$

On peut écrire cette équation comme :

$$f(\boldsymbol{\theta} + \lambda(\boldsymbol{\theta}^* - \boldsymbol{\theta})) \leq f(\boldsymbol{\theta}) + \lambda(f(\boldsymbol{\theta}^*) - f(\boldsymbol{\theta})). \tag{D.4}$$

Donc, si on choisit $0 < \lambda \leq 1$, on a :

$$\frac{f(\boldsymbol{\theta} + \lambda(\boldsymbol{\theta}^* - \boldsymbol{\theta})) - f(\boldsymbol{\theta})}{\lambda} \leq \underbrace{f(\boldsymbol{\theta}^*)}_{<\delta} - \underbrace{f(\boldsymbol{\theta})}_{=\delta} < \delta - \delta = 0. \tag{D.5}$$

Si l'on choisit la limite $\lambda \to \infty$, et qu'on multiplie le numérateur et le dénominateur par $(\boldsymbol{\theta}^* - \boldsymbol{\theta})$, alors on obtient bien l'équation (D.2) grâce à la définition de la dérivée, ce qui complète la démonstration. □

Nous pouvons maintenant procéder avec la démonstration du lemme 6.2.2.

Démonstration. En utilisant la définition de l'opérateur de projection (6.4) et le résultat du lemme D.2.1, on obtient :

$$(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top (\mathrm{Proj}\,(\boldsymbol{\theta}, \mathbf{y}) - \mathbf{y}) = \left\{ \begin{array}{ll} -\dfrac{\overbrace{(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \nabla f(\boldsymbol{\theta})}^{>0}}{(\nabla f(\boldsymbol{\theta}))^\top \nabla f(\boldsymbol{\theta})} \overbrace{\left[(\nabla f(\boldsymbol{\theta}))^\top \mathbf{y}\right]}^{>0} \overbrace{f(\boldsymbol{\theta})}^{>0} & \text{si } f(\boldsymbol{\theta}) > 0 \text{ et } \mathbf{y}^\top \nabla f(\boldsymbol{\theta}) > 0 \\ 0 & \text{sinon} \end{array} \right\} \leq 0, \tag{D.6}$$

ce qui complète la démonstration. □

Démonstration du lemme 6.2.3. L'existence d'une solution unique pour le système (6.6) est donnée par le fait que l'opérateur de projection est localement lipschitzien en $\boldsymbol{\theta}$ alors que l'entrée externe $\mathbf{y}(t)$ est continue par morceaux dans le temps. On cherche à montrer que :

$$f(\boldsymbol{\theta}_0) \Rightarrow f(\boldsymbol{\theta}(t)) \leq 1, \quad \forall t \geq 0. \tag{D.7}$$

À cette fin, on évalue la dérivée temporelle de $f(\boldsymbol{\theta})$:

$$\begin{aligned} \dot{f}(\theta) &= (\nabla f(\boldsymbol{\theta}))^\top \mathrm{Proj}\,(\boldsymbol{\theta}, \mathbf{y})\,, \\ &= \begin{cases} (\nabla f(\boldsymbol{\theta}))^\top \mathbf{y}(1 - f(\boldsymbol{\theta})) & \text{si } f(\boldsymbol{\theta}) > 0 \text{ et } \mathbf{y}^\top \nabla f(\boldsymbol{\theta}) > 0, \\ (\nabla f(\boldsymbol{\theta}))^\top \mathbf{y} & \text{sinon.} \end{cases} \end{aligned} \tag{D.8}$$

On s'intéresse alors à toutes les possibilités concernant la dérivée de $f(\boldsymbol{\theta})$:

$$\begin{aligned} &1.\ \dot{f}(\boldsymbol{\theta}) > 0, \quad \text{si} \quad 0 < f(\boldsymbol{\theta}) < 1 \text{ et } \mathbf{y}^\top \nabla f(\boldsymbol{\theta}) > 0; \\ &2.\ \dot{f}(\boldsymbol{\theta}) = 0, \quad \text{si} \quad f(\boldsymbol{\theta}) = 1 \text{ et } \mathbf{y}^\top \nabla f(\boldsymbol{\theta}) > 0; \\ &3.\ \dot{f}(\boldsymbol{\theta}) \leq 0, \quad \text{si} \quad f(\boldsymbol{\theta}) \leq 0 \text{ ou } \mathbf{y}^\top \nabla f(\boldsymbol{\theta}) \leq 0. \end{aligned} \tag{D.9}$$

Peu importe la condition initiale, on aura toujours $f(\boldsymbol{\theta}(t)) \leq 1,\ \forall t \geq 0$ ce qui complète la preuve. □

D.3 Somme et produit de Kronecker

Soit deux matrices quelconques $\mathbf{A} \in \mathbb{R}^{k\times l}$ et $\mathbf{B} \in \mathbb{R}^{m\times n}$. Alors, le produit de Kronecker est défini comme :

$$\begin{aligned}
\mathbf{A} \otimes \mathbf{B} &= \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1l}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{k1}\mathbf{B} & \cdots & a_{kl}\mathbf{B} \end{bmatrix}, \\
&= \begin{bmatrix}
a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1n} & \cdots & a_{1l}b_{11} & a_{1l}b_{12} & \cdots & a_{1l}b_{1n} \\
a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2n} & \cdots & a_{1l}b_{21} & a_{1l}b_{22} & \cdots & a_{1l}b_{2n} \\
\vdots & \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots \\
a_{11}b_{m1} & a_{11}b_{m2} & \cdots & a_{11}b_{mn} & \cdots & a_{1l}b_{m1} & a_{1l}b_{m2} & \cdots & a_{1l}b_{mn} \\
\vdots & \vdots & & \vdots & \ddots & \vdots & \vdots & & \vdots \\
a_{k1}b_{11} & a_{k1}b_{12} & \cdots & a_{k1}b_{1n} & \cdots & a_{kl}b_{11} & a_{kl}b_{12} & \cdots & a_{kl}b_{1n} \\
a_{k1}b_{21} & a_{k1}b_{22} & \cdots & a_{k1}b_{2n} & \cdots & a_{kl}b_{21} & a_{kl}b_{22} & \cdots & a_{kl}b_{2n} \\
\vdots & \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots \\
a_{k1}b_{m1} & a_{k1}b_{m2} & \cdots & a_{k1}b_{mn} & \cdots & a_{kl}b_{m1} & a_{kl}b_{m2} & \cdots & a_{kl}b_{mn}
\end{bmatrix}.
\end{aligned} \tag{D.10}$$

Soit deux matrices carrées quelconques $\mathbf{A} \in \mathbb{R}^{m\times m}$ et $\mathbf{B} \in \mathbb{R}^{n\times n}$. Alors, la somme de Kronecker est donnée par

$$\mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_m \otimes \mathbf{B}. \tag{D.11}$$

D.4 Transition entre la matrice à intervalles à la μ–analyse

Cette annexe reprend en partie les travaux de [43] sur la transition entre une matrice à intervalle et la μ–analyse. L'objectif de cette annexe est de donner la procédure complète et détaillée expliquant leur démarche. Nous donnerons aussi des indications quant à l'implémentation de leur procédure dans l'algorithme d'optimisation développé au chapitre 6. Pour bien comprendre la procédure, quelques définitions, théorèmes et lemmes sont nécessaires.

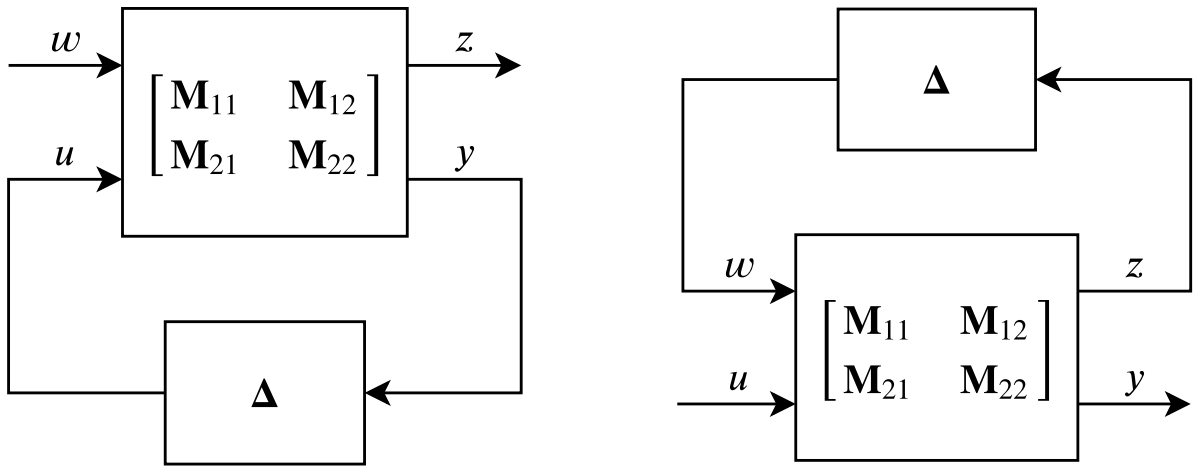
Définition D.4.1 (Tiré de [41])**.** *Soit* $\mathbf{M}$*, une matrice complexe partitionnée sous la forme :*

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}$$

Alors, la LFT supérieure $F_u(\mathbf{M}, \mathbf{\Delta})$ *et la LFT inférieure* $F_l(\mathbf{M}, \mathbf{\Delta})$ *sont données par*

$$\begin{aligned} F_u(\mathbf{M}, \mathbf{\Delta}) &= \mathbf{M}_{22} + \mathbf{M}_{21}\mathbf{\Delta}(\mathbf{I} - \mathbf{M}_{11}\mathbf{\Delta})^{-1}\mathbf{M}_{12}, \\ F_l(\mathbf{M}, \mathbf{\Delta}) &= \mathbf{M}_{11} + \mathbf{M}_{12}\mathbf{\Delta}(\mathbf{I} - \mathbf{M}_{22}\mathbf{\Delta})^{-1}\mathbf{M}_{21}, \end{aligned} \tag{D.12}$$

où $\mathbf{I} - \mathbf{M}_{11}\mathbf{\Delta}$ *et* $\mathbf{I} - \mathbf{M}_{22}\mathbf{\Delta}$ *sont inversibles.*



(a) LFT inférieure (b) LFT supérieure

Figure D.1 Formes d'une LFT

Théorème D.4.1 (Tiré de [59])**.** *Soit* $G(s)$*, une matrice de transfert qui est propre, stable et réelle. Alors, le système à rétroaction donné à la figure D.2 est robustement stable pour tout* $\mathbf{\Delta}$ *avec* $\|\mathbf{\Delta}\|_\infty \leq 1/\gamma$ *si et seulement si* $\sup_{\omega\in\mathbb{R}} \mu_{\mathbf{\Delta}}(G(j\omega)) < \gamma$*.*

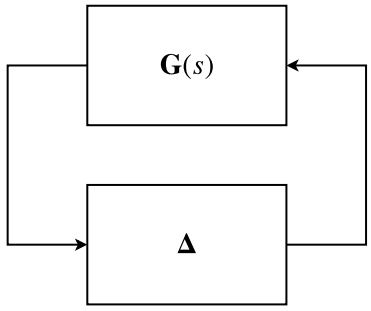


Figure D.2 Schéma bloc pour l'analyse de la stabilité robuste

Lemme D.4.2 (Tiré de [44])**.** *Soit* $\mathbf{M}$ *et* $\mathbf{N}$ *deux matrices complexes* $\mathbb{C}^{(p+q)\times(p+q)}$ *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* $\mathbf{\Delta}_{\mathbf{M}},\ \mathbf{\Delta}_{\mathbf{N}} \in \mathbb{C}^{q\times q}$ *alors*

$$F_l(\mathbf{M}, \mathbf{\Delta}_{\mathbf{M}}) \oplus F_l(\mathbf{N}, \mathbf{\Delta}_{\mathbf{N}}) = F_l(\mathbf{G}, \mathbf{\Delta}), \tag{D.13}$$

où

$$\mathbf{G} = \left[\begin{array}{c|cc} \mathbf{M}_{11} \otimes \mathbf{N}_{11} & \mathbf{M}_{12} \otimes \mathbf{I}_p & \mathbf{I}_p \otimes \mathbf{N}_{12} \\ \hline \mathbf{M}_{21} \otimes \mathbf{I}_p & \mathbf{M}_{22} \otimes \mathbf{I}_p & \mathbf{0} \\ \mathbf{I}_p \otimes \mathbf{N}_{21} & \mathbf{0} & \mathbf{I}_p \otimes \mathbf{N}_{22} \end{array}\right], \qquad \mathbf{\Delta} = \begin{bmatrix} \mathbf{\Delta}_{\mathbf{M}} \otimes \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_p \otimes \mathbf{\Delta}_{\mathbf{N}} \end{bmatrix}. \tag{D.14}$$

Lemme D.4.3 (Tiré de [44])**.** *Soit* $\mathbf{\Delta}_{\mathbf{M}} \in \mathbb{C}^{q_2 \times p_2}$ *et* $\mathbf{\Delta}_{\mathbf{N}} \in \mathbb{C}^{s_2 \times r_2}$ *et soit les matrices* $\mathbf{M}$ *et* $\mathbf{N}$ *qui sont partitionnées de la façon suivante :*

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \in \mathbb{C}^{(p_1+p_2)\times(q_1+q_2)}, \qquad \mathbf{N} = \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} \\ \mathbf{N}_{21} & \mathbf{N}_{22} \end{bmatrix} \in \mathbb{C}^{(r_1+r_2)\times(s_1+s_2)}. \tag{D.15}$$

Alors,

$$F_l(\mathbf{M}, \mathbf{\Delta}_{\mathbf{M}}) \otimes_l (\mathbf{N}, \mathbf{\Delta}_{\mathbf{N}}) = F_l(\mathbf{G}_p, \mathbf{\Delta}_{\mathbf{G}_p}), \tag{D.16}$$

où

$$\mathbf{G}_p = \left[\begin{array}{c|cc} \mathbf{M}_{11} \otimes \mathbf{N}_{11} & \mathbf{M}_{12} \otimes \mathbf{I}_{r_1} & \mathbf{M}_{11} \otimes \mathbf{N}_{12} \\ \hline \mathbf{M}_{21} \otimes \mathbf{N}_{11} & \mathbf{M}_{22} \otimes \mathbf{I}_{r_1} & \mathbf{M}_{21} \otimes \mathbf{N}_{12} \\ \mathbf{I}_{q_1} \otimes \mathbf{N}_{21} & \mathbf{0} & \mathbf{I}_{s_1} \otimes \mathbf{N}_{22} \end{array}\right], \qquad \mathbf{\Delta}_{\mathbf{G}_p} = \begin{bmatrix} \mathbf{\Delta}_{\mathbf{M}} \otimes \mathbf{I}_{r_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{q_1} \otimes \mathbf{\Delta}_{\mathbf{N}} \end{bmatrix}. \tag{D.17}$$

Les lemmes précédents nous permettront de construire, à partir du système à intervalle, une LFT inférieure en fonction de l'application gardienne donnée. D'abord, on rappelle qu'une matrice à intervalle est donnée par :

$$\begin{aligned} \mathbf{A}_I &= \left[\underline{\mathbf{A}} \quad \overline{\mathbf{A}}\right], \\ &= \left\{\mathbf{A} = [a_{ij}] : \underline{a}_{ij} \leq a_{ij} \leq \overline{a}_{ij}, \quad i,\ j = 1,\ 2,\ \ldots,\ n\right\}. \end{aligned} \tag{D.18}$$

On souhaite représenter $\mathbf{A}_I$ comme une LFT inférieure. On définit d'abord la matrice nomi-

nale $\mathbf{A}_0$ et la matrice de différence $\tilde{\mathbf{A}}_{ij}$ comme étant

$$\mathbf{A}_0 = \frac{1}{2}(\underline{\mathbf{A}} + \overline{\mathbf{A}}), \qquad \tilde{\mathbf{A}}_{ij} = \frac{1}{2}(\overline{\mathbf{A}} - \underline{\mathbf{A}}) = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \tilde{a}_{ij} & \vdots \\ 0 & \cdots & 0 \end{bmatrix} = [\tilde{a}_{ij}]\,. \tag{D.19}$$

Soit q, le nombre de valeurs non nulles dans $\tilde{\mathbf{A}}_{ij}$. Si $\mathbf{A}_I$ est d'ordre n, alors $q \leq n^2$. On peut alors renommer les matrices $\tilde{\mathbf{A}}_{ij}$ par $\mathbf{A}_1,\ \mathbf{A}_2,\ \ldots,\ \mathbf{A}_q$. Cela nous permet d'exprimer la matrice à intervalle comme

$$\mathbf{A}_I = \begin{bmatrix} \underline{\mathbf{A}} & \overline{\mathbf{A}} \end{bmatrix} = \mathbf{A}_0 + \sum_{i=1}^{q} \alpha_i \mathbf{A}_i, \quad -1 \leq \alpha_i \leq 1,\ \forall i = 1,\ \ldots,\ q. \tag{D.20}$$

On sait également que $\text{rang}(\mathbf{A}_i) = 1,\ i = 1,\ \ldots,\ q$. Alors, il existe des vecteurs colonnes $\mathbf{X}_i,\ \mathbf{Y}_i \in \mathbb{R}^n$ tels que

$$\mathbf{A}_i = \mathbf{X}_i \mathbf{Y}_i^\top, \quad i = 1,\ \ldots,\ q. \tag{D.21}$$

Donc la matrice à intervalle $\mathbf{A}_I$ peut être exprimée comme une LFT de la façon suivante :

$$\mathbf{A}_I = F_l(\mathbf{M}, \mathbf{\Delta}) = F_l\left(\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}, \mathbf{\Delta}\right), \tag{D.22}$$

où

$$\begin{aligned} &\mathbf{M}_{11}, = \mathbf{A}_0 \qquad\qquad \mathbf{M}_{12} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \ldots & \mathbf{X}_q \end{bmatrix}, \\ &\mathbf{M}_{21} = \begin{bmatrix} \mathbf{Y}_1^\top \\ \mathbf{Y}_2^\top \\ \vdots \\ \mathbf{Y}_q^\top \end{bmatrix}, \qquad \mathbf{M}_{22} = \mathbf{0}_q, \end{aligned} \tag{D.23}$$

et $\mathbf{\Delta} = \text{diag}\,\{\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_q\}$ avec $-1 \leq \alpha_i \leq 1,\ \forall i = 1,\ \ldots,\ q$.

Le système dynamique peut alors être donné temporellement comme $\delta\mathbf{x}(t) = F_l(\mathbf{M}, \mathbf{\Delta})\mathbf{x}(t)$. Pour étudier la stabilité du système par rapport à une application gardienne, il suffit d'effectuer les opérations sur la LFT. Par exemple,

1. Pour la stabilité Hurwitz, il suffit de considérer le système $F_l(\tilde{\mathbf{M}}, \tilde{\mathbf{\Delta}}) = F_l(\mathbf{M}, \mathbf{\Delta}) \oplus F_l(\mathbf{M}, \mathbf{\Delta})$.
2. Pour une frontière verticale définie par σ il faut considérer le système $F_l(\tilde{\mathbf{M}}_l, \tilde{\mathbf{\Delta}}) = (F_l(\mathbf{M}, \mathbf{\Delta}) + \sigma\mathbf{I}) \oplus (F_l(\mathbf{M}, \mathbf{\Delta}) + \sigma\mathbf{I})$.

3. Pour une frontière en angle définie par θ, il faut considérer le système $F_l(\tilde{\mathbf{M}}_s, \tilde{\mathbf{\Delta}}) = (e^{-j\theta} F_l(\mathbf{M}, \mathbf{\Delta})) \oplus (e^{j\theta} F_l(\mathbf{M}, \mathbf{\Delta}))$.

Le lecteur peut se référer aux applications gardiennes correspondantes définies à la section 6.3.1. Les différentes opérations sur $F_l(\mathbf{M}, \mathbf{\Delta})$ modifient la structure de $\mathbf{\Delta}$. Or, il est important de conserver une structure appropriée pour la μ–analyse.

Exemple. Soit un système à intervalle $\mathbf{A}_I$. On l'exprime sous la forme d'une LFT pour obtenir $F_l(\mathbf{M}, \mathbf{\Delta})$ où $\mathbf{\Delta} = \operatorname{diag}\{\alpha_1, \ \ldots, \ \alpha_q\}$. Si on s'intéresse à la stabilité Hurwitz du système, on le modifiera de la façon suivante

$$F_l(\tilde{\mathbf{M}}, \tilde{\mathbf{\Delta}}) = F_l(\mathbf{M}, \mathbf{\Delta}) \oplus F_l(\mathbf{M}, \mathbf{\Delta}) \tag{D.24}$$

avec

$$\tilde{\mathbf{M}} = \left[\begin{array}{c|cc} \mathbf{M}_{11} \otimes \mathbf{N}_{11} & \mathbf{M}_{12} \otimes \mathbf{I}_p & \mathbf{I}_p \otimes \mathbf{N}_{12} \\ \hline \mathbf{M}_{21} \otimes \mathbf{I}_p & \mathbf{M}_{22} \otimes \mathbf{I}_p & \mathbf{0} \\ \mathbf{I}_p \otimes \mathbf{N}_{21} & \mathbf{0} & \mathbf{I}_p \otimes \mathbf{N}_{22} \end{array}\right] = \left[\begin{array}{c|c} \tilde{\mathbf{M}}_{11} & \tilde{\mathbf{M}}_{12} \\ \hline \tilde{\mathbf{M}}_{21} & \tilde{\mathbf{M}}_{22} \end{array}\right], \ \tilde{\mathbf{\Delta}} = \begin{bmatrix} \mathbf{\Delta} \otimes \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \otimes \mathbf{\Delta} \end{bmatrix}. \tag{D.25}$$

Si $q = 2$ et $n = 3$, alors on aura la structure suivante pour la matrice des incertitudes : $\tilde{\mathbf{\Delta}} = \operatorname{diag}\{\alpha_1, \ \alpha_1, \ \alpha_1, \ \alpha_2, \ \alpha_2, \ \alpha_2, \ \alpha_1, \ \alpha_2, \ \alpha_1, \ \alpha_2, \ \alpha_1, \ \alpha_2\}$. Or, on souhaite avoir la structure suivante : $\mathbf{\Delta}_p = \operatorname{diag}\{\alpha_1 \mathbf{I}_{2n}, \ \alpha_2 \mathbf{I}_{2n}, \ \ldots, \ \alpha_q \mathbf{I}_{2n}\}$. Donc, on définit la matrice de permutation $\mathbf{P}$ qui donne $\mathbf{\Delta}_p = \mathbf{P}\tilde{\mathbf{\Delta}}\mathbf{P}^\top$.

Avec cette matrice de permutation, la LFT sera également modifiée de la façon suivante :

$$\begin{aligned} F_l(\tilde{\mathbf{M}}, \tilde{\mathbf{\Delta}}) &= F_l\left(\begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} \tilde{\mathbf{M}} \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^\top \end{bmatrix}, \mathbf{P}\tilde{\mathbf{\Delta}}\mathbf{P}^\top\right), \\ &= F_l\left(\begin{bmatrix} \tilde{\mathbf{M}}_{11} & \tilde{\mathbf{M}}_{12}\mathbf{P}^\top \\ \mathbf{P}\tilde{\mathbf{M}}_{21} & \mathbf{0} \end{bmatrix}, \mathbf{\Delta}_p\right), \\ &=: F_l(\mathbf{\Psi}, \mathbf{\Delta}_p). \end{aligned} \tag{D.26}$$

Ceci nous amène au théorème central de cette annexe.

Théorème D.4.4 (Tiré de [43])**.** *Soit une application gardienne qui protège un ensemble Ω. Soit $\mathbf{A}_0$ la matrice nominale d'un système à intervalle. On suppose que cette matrice est Ω–stable. Alors, le système est robustement Ω–stable si et seulement si, pour chacune des*

frontières θ et σ qui définissent l'application gardienne,

$$\mu_{\mathbf{\Delta}_p}(\mathbf{M}_s) < 1 \quad \textit{et} \quad \mu_{\mathbf{\Delta}_p}(\mathbf{M}_l) < 1, \tag{D.27}$$

où $\mathbf{M}_s := \mathbf{P}\tilde{\mathbf{M}}_{s,21}\tilde{\mathbf{M}}_{s,11}^{-1}\tilde{\mathbf{M}}_{s,12}\mathbf{P}^\top$, $\mathbf{M}_l := \mathbf{P}\tilde{\mathbf{M}}_{l,21}\tilde{\mathbf{M}}_{l,11}^{-1}\tilde{\mathbf{M}}_{l,12}\mathbf{P}^\top$ *et* $\mathbf{\Delta}_p = \mathrm{diag}\{\alpha_1\mathbf{I}_{2n},\ \alpha_2\mathbf{I}_{2n},\ \ldots,\ \alpha_q\mathbf{I}_{2n}\}$.

Ce théorème explique que pour une matrice à intervalle représentée sous la forme d'une LFT et pour une application gardienne qui utilise cette LFT pour ces différentes frontières, alors il est possible de déterminer si la matrice à intervalle est Ω–stable en passant par la μ–analyse. Avec le *toolbox* MATLAB, il est possible de calculer la valeur de $\mu_{\mathbf{\Delta}_p}(\mathbf{M})$. Si cette valeur est inférieure à 1, alors le système est Ω–stable. Sinon, il ne l'est pas.

D.5 Échantillonnage par hypercube latin

L'échantillonnage par hypercube latin permet d'obtenir des points qui sont uniformément pseudo-aléatoirement répartis sur un ensemble d'intérêt. Pour cet échantillonnage, on reprend la méthode proposée par [57] et qui est donné à l'algorithme D.1.

Algorithme D.1 : Échantillonnage par hypercube latin

Données : Soit un hyperrectangle $[l,\ u] \subset \mathbb{R}^n$ et p, le nombre de point requis pour obtenir un échantillon complet.

Initialisation

Soit Π, une matrice $n \times p$ pour laquelle chacune des n rangées est une permutation aléatoire du vecteur $[1,\ 2,\ \ldots,\ p]$.

pour $i = 1,\ 2,\ \ldots,\ n$ **faire**

  pour $j = 1,\ 2,\ \ldots,\ p$ **faire**

$$x_i^j = l_i + \frac{(\Pi_{i,j} - r_{i,j})}{p}(u_i - l_i)$$

où $r_{i,j} \in \mathbb{R}$ est un nombre aléatoire dans l'intervalle $[0,\ 1]$

  fin

fin

Résultat : Retourner $\{x^1,\ x^2,\ \ldots,\ x^p\} \subset \mathbb{R}^n$.

Exemple. La figure D.3 donne un exemple d'échantillonnage avec un hypercube latin. Pour 10 points, dans des intervalles $[0,\ 10]$, l'espace sera divisé en 10 colonnes et 10 rangées. L'échantillonnage par hypercube latin fait en sorte qu'il n'y ait jamais deux points dans la même colonne et la même rangée. Ceci a comme avantage de maximiser la distribution des

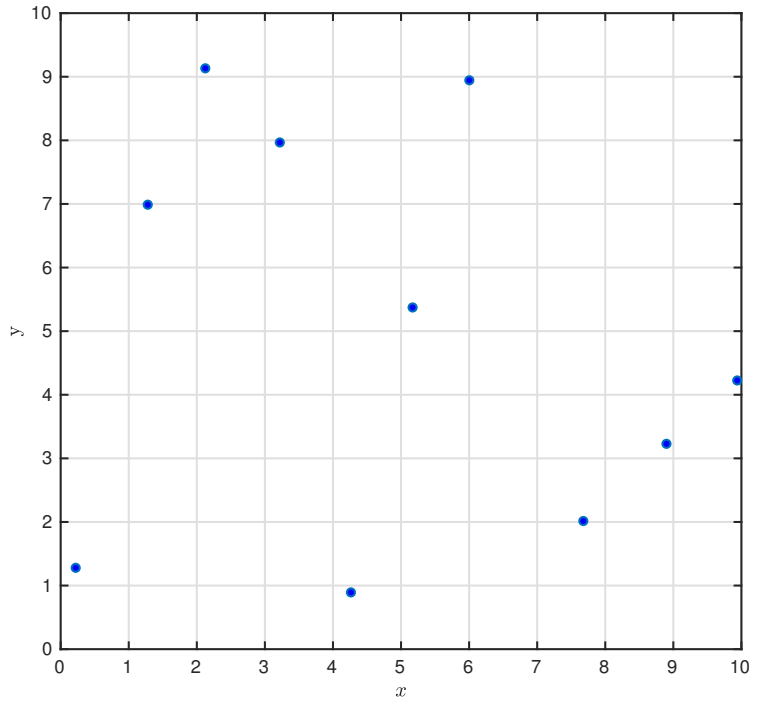


Figure D.3 Exemple d'échantillonnage pour $n = 2$ et $p = 10$

points dans l'espace. Ainsi, pour l'algorithme d'optimisation sans dérivées qui dépend de la position des points initiaux, on maximise les chances de trouver un bon point de départ. Cela fera en sorte qu'on a plus de chances de trouver un maximum local intéressant.

D.6 Explications concernant l'algorithme d'optimisation sans dérivées MADS

Le problème d'optimisation donné par (6.36) est hautement complexe dû à la condition **C2**. En effet, cette condition requiert la vérification d'une application gardienne en passant par la μ–analyse. Nous considérons alors des méthodes d'optimisation sans dérivées ou de boîte noire qui ne nécessitent pas l'évaluation du gradient.

L'algorithme MADS développé par [56] permet de considérer des problèmes d'optimisation sans dérivées tel que celui donné par (6.36). Pour utiliser cet algorithme d'optimisation, nous reformulons le problème sous la forme d'une boîte noire. Étant donné des valeurs initiales pour les gains adaptatifs $\mathbf{K}_{min}$ et $\mathbf{K}_{max}$, nous obtiendrons une solution maximale locale correspondante. L'algorithme fonctionne de la façon suivante :

1. Il commence par déterminer un certain nombre de directions potentielles de descente

en fonction d'une distance de sonde pour les gains $\mathbf{K}_{min}$ et $\mathbf{K}_{max}$.

2. Il vérifie l'hypervolume obtenu pour chacune des directions potentielles. Pour le respect des conditions **C1** et **C2**, on utilise la barrière infinie donnée par

$$V\left(\hat{\mathbf{K}}_{min},\ \hat{\mathbf{K}}_{max}\right) = \begin{cases} \prod_{i=1}^{n}\left(\hat{\mathbf{K}}_{i,max} - \hat{\mathbf{K}}_{i,min}\right) & \text{si } \mathbf{C1} \text{ et } \mathbf{C2} \text{ respectés,} \\ -\infty & \text{sinon.} \end{cases} \tag{D.28}$$

   Ainsi, on s'assure que l'algorithme d'optimisation respecte toujours les conditions imposées.

3. Après avoir calculé l'hypervolume pour chacune des directions potentielles de descente, deux cas sont possibles :
   (a) L'algorithme choisit la direction de descente donnant la plus grande augmentation de l'hypervolume. Il actualise les valeurs de $\mathbf{K}_{min}$ et $\mathbf{K}_{max}$ et une nouvelle itération est requise.
   (b) Aucune des directions de descente n'offre un meilleur hypervolume. La distance de sonde est réduite et une nouvelle itération est requise. Si la distance de sonde est diminuée sous un certain seuil, la condition d'arrêt est activée. Les valeurs actuelles de $\mathbf{K}_{min}$ et $\mathbf{K}_{max}$ sont considérées comme une solution maximale locale pour l'hypervolume.

L'algorithme MADS présente deux principaux avantages. D'abord c'est une méthode d'optimisation sans dérivées éprouvée qui possède une bonne garantie de convergence pour des systèmes hautement complexes. Son second avantage est que les directions de sonde deviennent un ensemble asymptotiquement dense. En considérant un nombre croissant de directions de descente, la convergence de l'algorithme vers un maximum local est plus efficace.

D.7 Obtention des bornes pour les autres contrôleurs

D.7.1 Contrôleur de la rotation en lacet ψ

Les pôles du système idéal sont situés à

$$p_{1,2} = -1.3916 \pm 1.5260j, \quad p_3 = -0.9791. \tag{D.29}$$

On peut maintenant définir l'application gardienne pour ce contrôleur. Pour la partie réelle, on choisit les frontières $\sigma_1 = -3$ et $\sigma_2 = -0.1$. Pour l'amortissement, on choisit $\theta = 0.21$ rad.

Avec l'échantillonnage par hypercube latin, on obtient 17 points initiaux. Les résultats de l'algorithme donnent un hypervolume situé entre [6.4735, 27.0622] avec une moyenne de 15.9055. La solution choisie pour l'implémentation pratique n'est pas la solution optimale. En effet, la solution retenue a un hypervolume de $V = 13.9464$. Les bornes correspondantes sont :

$$\begin{aligned} \hat{k}_{\psi,max} &= 0.3324, & \hat{k}_{\psi,min} &= -0.7996, & \hat{k}_{\psi,int} &= 1.1320, \\ \hat{k}_{r,max} &= 3.3391, & \hat{k}_{r,min} &= -0.2984, & \hat{k}_{r,int} &= 3.6375, \\ \hat{k}_{x_{i,\psi},max} &= 3.4083, & \hat{k}_{x_{i,\psi},min} &= -0.0088, & \hat{k}_{x_{i,\psi},int} &= 3.4171. \end{aligned} \tag{D.30}$$

D.7.2 Contrôleur de la position en x

Les pôles du système idéal sont situés à

$$p_{1,2} = -1.9221 \pm 2.5359j, \quad p_3 = -6.8920, \quad p_4 = -1.6424, \quad p_5 = -1.0025. \tag{D.31}$$

On peut maintenant définir l'application gardienne pour ce contrôleur. Pour la partie réelle, on choisit les frontières $\sigma_1 = -9$ et $\sigma_2 = -0.4$. Pour l'amortissement, on choisit $\theta = 0.21$ rad.

Avec l'échantillonnage par hypercube latin, on obtient 26 points initiaux. Les résultats de l'algorithme donnent un hypervolume situé entre [0.1071, 0.6124] avec une moyenne de 0.3446. La solution choisie a un hypervolume de $V = 0.6104$ et les bornes correspondantes sont :

$$\begin{aligned} \hat{k}_{x,max} &= 0.4933, & \hat{k}_{x,min} &= -0.6392, & \hat{k}_{x,int} &= 1.1325, \\ \hat{k}_{u,max} &= 0.3672, & \hat{k}_{u,min} &= -0.1754, & \hat{k}_{u,int} &= 0.5426, \\ \hat{k}_{\theta,max} &= 0.3992, & \hat{k}_{\theta,min} &= -0.7408, & \hat{k}_{\theta,int} &= 1.1400, \\ \hat{k}_{q,max} &= 0.4378, & \hat{k}_{q,min} &= -0.3853, & \hat{k}_{q,int} &= 0.8231, \\ \hat{k}_{x_{i,x},max} &= 1.0401, & \hat{k}_{x_{i,x},min} &= -0.0185, & \hat{k}_{x_{i,x},int} &= 1.0586. \end{aligned} \tag{D.32}$$

D.7.3 Contrôleur de la position en y

Les pôles du système idéal sont situés à

$$p_{1,2} = -3.5408 \pm 3.4115j, \quad p_{3,4} = -1.3348 \pm 1.6527j, \quad p_5 = -1.4512. \tag{D.33}$$

On peut maintenant définir l'application gardienne pour ce contrôleur. Pour la partie réelle, on choisit les frontières $\sigma_1 = -6$ et $\sigma_2 = -0.4$. La frontière pour l'amortissement est placée

à $\theta = 0.21$ rad.

Avec l'échantillonnage par hypercube latin, on obtient 20 points initiaux. Les résultats de l'algorithme donnent un hypervolume situé entre $[0.0700,\ 0.9109]$ avec une moyenne de 0.4302. La solution choisie a un hypervolume de $V = 0.5921$ et les bornes correspondantes sont :

$$\begin{aligned}
\hat{k}_{y,max} &= 0.3097, & \hat{k}_{y,min} &= -0.0281, & \hat{k}_{y,int} &= 0.3378, \\
\hat{k}_{v,max} &= 0.3685, & \hat{k}_{v,min} &= -0.1052, & \hat{k}_{v,int} &= 0.4737, \\
\hat{k}_{\phi,max} &= 0.2739, & \hat{k}_{\phi,min} &= -3.0353, & \hat{k}_{\phi,int} &= 3.3092, \\
\hat{k}_{p,max} &= 0.4245, & \hat{k}_{p,min} &= -0.1306, & \hat{k}_{p,int} &= 0.5551, \\
\hat{k}_{x_{i,y},max} &= 1.8646, & \hat{k}_{x_{i,y},min} &= -0.1497, & \hat{k}_{x_{i,y},int} &= 2.0143.
\end{aligned} \tag{D.34}$$