

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



École Nationale Polytechnique
Département d'automatique
Laboratoire de Commande des Processus



Thèse de Doctorat en Génie Électrique
Option : Automatique

Intitulé

**Contribution à la Commande Adaptative
Robuste des Véhicules Autonomes Volants
à Décollage et Atterrissage Verticaux**

Présentée par

Rabie Riadh BENREZKI

Magistère en Contrôle de Processus et Robotique de l'USTHB

Soutenue le 05 Décembre 2019 devant le Jury composé de :

Président	BOUDJEMA Farès	(Professeur, ENP-Alger)
Directeur de thèse	TADJINE Mohamed	(Professeur, ENP-Alger)
Co-directeur de thèse	TAYEBI Abdelhamid	(Professeur, Lakehead University, Canada)
Examineurs	BALI Nouredine	(Professeur, USTHB-Alger)
	BOUCHOUCHA Mouloud	(MC/A, EMP)
	SAARI Hammoud	(Professeur, ENSM Bousmail)
	BERKOUK El Madjid	(Professeur, ENP-Alger)

ENP 2019

Laboratoire de Commande des Processus (LCP), Ecole Nationale Polytechnique (ENP)
10, avenue des Frères Oudek, Hassen Badi, BP.182, 16200, El-Harrach, Alger, Algérie

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



École Nationale Polytechnique
Département d'automatique
Laboratoire de Commande des Processus



Thèse de Doctorat en Génie Électrique
Option : Automatique

Intitulé

**Contribution à la Commande Adaptative
Robuste des Véhicules Autonomes Volants
à Décollage et Atterrissage Verticaux**

Présentée par

Rabie Riadh BENREZKI

Magistère en Contrôle de Processus et Robotique de l'USTHB

Soutenue le 05 Décembre 2019 devant le Jury composé de :

Président	BOUDJEMA Farès	(Professeur, ENP-Alger)
Directeur de thèse	TADJINE Mohamed	(Professeur, ENP-Alger)
Co-directeur de thèse	TAYEBI Abdelhamid	(Professeur, Lakehead University, Canada)
Examineurs	BALI Nouredine	(Professeur, USTHB-Alger)
	BOUCHOUCHA Mouloud	(MC/A, EMP)
	SAARI Hammoud	(Professeur, ENSM Bousmail)
	BERKOUK El Madjid	(Professeur, ENP-Alger)

ENP 2019

Laboratoire de Commande des Processus (LCP), Ecole Nationale Polytechnique (ENP)
10, avenue des Frères Oudek, Hassen Badi, BP.182, 16200, El-Harrach, Alger, Algérie

Remerciements

Ce sujet de recherche a été proposé et dirigé par Monsieur **Tadjine Mohamed**, en tant que directeur de thèse, Professeur à l'ENP et dérigé par Monsieur **Tayebi Abdelhamid** Professeur à l'université Lakehead, Thunder bay, Canada. Je trouve ici l'expression de ma sincère gratitude pour la confiance qu'ils m'ont accordée en me proposant ce sujet. Cette thèse n'aurait pas été achevée sans l'aide et les précieux conseils de mes deux encadrants. Ils m'ont toujours encouragé et soutenu durant toutes ces années de préparation de cette thèse. Leurs nombreux conseils, orientations, engagements, et leurs disponibilités m'ont permis d'acquérir des connaissances complémentaires. Je tiens à remercier Pr **Tayebi Abdelhamid** pour son chaleureux accueil au sein de son laboratoire "Automatic Control Laboratory", durant toute la période du stage PNE.

Je remercie chaleureusement les membres de Jury qui m'ont fait l'honneur d'évaluer ce modeste travail de thèse.

Je tiens à remercier Mr Boudjema, professeur à l'ENP, pour avoir accepté de présider ma soutenance de doctorat.

Je tiens à remercier Mr Saari professeur à l'ENSM et Mr Bali professeur à l'USTHB, pour avoir expertisé ma thèse de doctorat et pour avoir accepté d'être examinateurs dans le jury de ma soutenance.

Je tiens à remercier Mr Berkouk professeur à l'ENP et Mr Bouchoucha Maître de conférence A à l'EMP, pour avoir expertisé ma thèse de doctorat et pour avoir accepté d'être examinateurs dans le jury de ma soutenance.

Je tiens à remercier mon ami et chef de l'équipe CSE Mr Omar Kermia Maître de recherche A au CDTA pour son aide et soutien technique et morale énorme. Je tiens à remercier mes chères amis Smain Tchenderli, Fouad Yacef, Zakaria Guettatfi, Ibrahim Mezzah, Ramdan Kechad pour leurs sympathies, encouragements et aides. Je tiens à remercier mon étudiant d'ingénieur-master de l'école nationale polytechnique de Constantine Derouiche Oualid et mon ancien étudiant de master à l'université d'EVRY VAL D'ESSONNE pour leur contribution dans la phase des tests d'implémentation. J'ai une pensée pour mes collègues du CDTA qui m'ont encouragé durant mes années d'étude de doctorat.

Rabie Riadh BENREZKI

Dédicaces

A mes chères parents et ma sœur pour leurs soutiens et encouragements énormes.

A mes filles chéries Allae et Merieme dont le sourire éblouit chaque jour de ma vie.

For Her, my idea of beautiful; beloved Fatima LALMI.

ملخص

في هذه الأطروحة ، اقترحنا قانون تحكم قوي مخصص للتحكم الذاتي في الطائرات بدون طيار ذات صعود و نزول عمودي VTOLs في حالة عدم معرفة تامة للقصور الذاتي للنظام وفي وجود تحيز جيروسكوبي ، وفقدان كفاءة المشغلات (LOE) وبحضور الاضطرابات الخارجية. في هذا السياق ، ركزنا دراستنا على تطوير قانون تحكم تكيفي مستقر قوي والبسيط. هذا الأمر يجعل من الممكن تتبع المسار. وضحنا فعالية نظام التحكم المقترح في نتائج المحاكاة المقدمة ، والمطبقة على نموذج المروحة الواقعية (Ducted fan). نجاعة التحكم المقترح تظهر من خلال نتائج محاكات و نتائج تنفيذ تطبيق عملي على نظام مفتوح المصدر " Crazyflie 2.0".

الكلمات المفتاحية: طائرة بدون طيار ذات صعود و نزول عمودي, نفس فعالية, تحكم قوي متأقلم, " Crazyflie 2.0"

Abstract

In this thesis, we have proposed a trajectory tracking robust control law for autonomous Vertical Takeoff and Landing (VTOL) Unmanned Aerial Vehicles (UAVs) in the case where the inertia of the system is totally unknown and in the presence of a gyroscopic bias, a loss of efficiency of the actuators (LOE) and in the presence of external disturbances. In this context, we focused our study on the development of a stable, robust and simple to implement adaptive control law. The effectiveness of the proposed control system is illustrated through simulation results and practical implementation of two approaches on the open source platforme "Crazyflie 2.0".

Key Words : UAV-VTOLs, LOE, Adaptive robust control, " Crazyflie 2.0

Résumé

Dans cette thèse, nous avons proposé une loi de contrôle robuste de suivi de trajectoire pour les véhicules aériens sans pilote (UAV) autonomes à décollage et atterrissage verticaux (VTOL) dans le cas où l'inertie du système est totalement inconnue et en présence d'un biais gyroscopique, perte d'efficacité des actionneurs (LOE) et en présence de perturbations externes. Dans ce contexte, nous avons concentré notre étude sur le développement d'une loi de commande adaptative stable, robuste et simple à mettre en œuvre. L'efficacité du système de contrôle proposé est illustrée par des résultats de simulation et une mise en œuvre pratique de deux approches sur la plate-forme open source «Crazyflie 2.0».

Mots clés : UAV-VTOLs, LOE, Commande robuste adaptative, " Crazyflie 2.0

Table des matières

Table des matières

Liste des figures

Liste des tableaux

Nomenclatures

Introduction générale 14

1 Modélisation dynamique d'un corps rigide, notion de représentation d'attitude et outils mathématiques 18

1.1 Introduction 19

1.2 Représentation d'attitude 19

1.2.1 Repères inertiel et mobile 19

1.2.2 Angles d'Euler 20

1.2.3 Matrice des Cosinus Directeurs (Rotation) 21

1.2.4 Quaternion unitaire 22

1.3 Modèle dynamique d'un VTOL 25

1.4 Reconstruction d'attitude 26

1.5 Concept de la stabilité 27

1.5.1 Système autonome 27

1.5.2 Système non-autonome 29

1.5.3 Inégalité de Young 31

1.6	Outils mathématiques	32
1.6.1	La Matrice Antisymétrique	32
1.6.2	Projection	32
1.7	Conclusion	33
2	La plateforme Crazyflie 2.0	34
2.1	Introduction	35
2.2	Crazyflie	35
2.2.1	Caractéristiques générales	35
2.2.2	Microcontrôleurs	38
2.2.3	Centrale inertielle (IMU)	38
2.2.4	Autres capteurs	39
2.2.5	Batterie et alimentation	39
2.2.6	Moteurs	39
2.2.7	Hélices	39
2.3	Crazyradio	39
2.4	Firmware partie logicielle	40
2.4.1	Le système d'exploitation temps réel FreeRTOS	40
2.4.2	Stabilisation du Crazyflie	41
2.5	Crazy RealTime Protocol (CRTP)	44
2.6	PC client	45
2.7	Le dispositif Loco Positioning System	46
2.7.1	Le système LPS	46
2.7.2	Mode de positionnement	47
2.7.3	Performance	48
2.7.4	Two Way Ranging (TWR)	49
2.8	Conclusion	51
3	Commande adaptative d'un VTOL-UAVs avec inertie inconnue, Gyro-bias et LOE d'actionneur	52
3.1	Introduction	53

3.2	Formulation du problème	54
3.2.1	Modèle du system	54
3.2.2	Énoncé du problème	55
3.3	Synthèse du contrôleur	55
3.3.1	Synthèse de la commande virtuel de translation	55
3.3.2	Extraction de l'attitude et de la poussée désirées	56
3.3.3	Synthèse du contrôleur rotationnel	59
3.4	Analyse de stabilité	61
3.5	Simulations	64
3.6	Conclusion	70
4	Résultats expérimentaux	72
4.1	Introduction	73
4.2	Contrôle de position en utilisant une approche simplifiée	73
4.2.1	Conception de l'entrée de commande virtuel pour la dynamique translationnelle	74
4.2.2	Extraction de l'attitude et de la poussée désirées	76
4.2.3	Commande de l'attitude	76
4.2.4	Génération de trajectoire polynomial	77
4.2.5	Résultats de simulation	78
4.2.6	Résultats d'implémentation	80
4.3	Contrôle de position en utilisant l'approche du Backstepping où les perturbations sont nulles et les paramètres du système sont connus	87
4.3.1	Synthèse du contrôleur	87
4.3.2	Résultats de simulation	90
4.3.3	Résultats d'implémentations	91
4.4	Conclusion	96
	Conclusion générale et perspectives	100
A	Annexe	101

A.1	Calcul de $\dot{\tilde{\Omega}}$ et $J\dot{\tilde{\Omega}}$	102
A.1.1	Calcul de $\dot{\tilde{\Omega}}$	102
A.1.2	Calcul de $J\dot{\tilde{\Omega}}$	105

Bibliographie		110
----------------------	--	------------

Liste des figures

1	Modélisation dynamique du Quadrotor	18
1.1	Rotation de \mathcal{B} par rapport à I	22
1.2	Repère inertiel, repère fixe et forces exogènes	26
1.3	Equilibre stable	29
2	La plateforme Crazyflie 2.0	34
2.1	Crazyflie 2.0	35
2.2	Architecture du CF	36
2.3	Architecture du système CF	38
2.4	Système de stabilisation	41
2.5	trame de "commander"	44
2.6	Forme de CRTP header	45
2.7	Couche du protocole CRTP	45
2.8	Client crazyflie	46
2.9	LPS deck	47
2.10	LPS node	47
2.11	LPS Tab	49
2.12	LPS Tab	49
2.13	Protocole Two Way Ranging	50
3	Commande adaptative d'un VTOL-UAVs avec inertie inconnue, Gyro-bias et LOE d'actionneur	52

3.1	3D plot of the actual and the desired trajectory	66
3.2	Norme de l'erreur de position	67
3.3	Norme de l'erreur de vitesse	67
3.4	Norme de l'erreur d'attitude	68
3.5	Norme de $\ \tilde{\Omega}\ $	68
3.6	Norme de l'erreur de l'estimation du biais gyroscopique	69
3.7	Norme du torque	69
3.8	La poussé du système	70
4	Résultats expérimentaux	72
4.1	Schéma fonctionnel de l'approche simplifié	74
4.2	Normes des erreurs	79
4.3	Trajectoires en 3D	79
4.4	Position	81
4.5	Norme de l'erreur de la position	81
4.6	Norme de l'erreur de l'attitude	82
4.7	Norme de l'erreur de la vitesse	82
4.8	Norme de l'erreur de la vitesse angulaire	83
4.9	Position	84
4.10	Norme de l'erreur de la position	84
4.11	Norme de l'erreur de l'attitude	85
4.12	L'attitude	85
4.13	Norme de l'erreur de la vitesse	86
4.14	Norme de l'erreur de la vitesse angulaire	86
4.15	Normes des erreurs	90
4.16	Trajectoires en 3D	91
4.17	Position	92
4.18	Norme de l'erreur de la position	92
4.19	Norme de l'erreur de l'attitude	93
4.20	Norme de l'erreur de la vitesse	93

Liste des figures

4.21	Norme de l'erreur de la vitesse angulaire	94
4.22	Position	95
4.23	Norme de l'erreur de la position	95
4.24	Norme de l'erreur de l'attitude	96
4.25	L'attitude	96

Liste des tableaux

1	Modélisation dynamique du Quadrotor	18
2	La plateforme Crazyflie 2.0	34
3	Commande adaptative d'un VTOL-UAVs avec inertie inconnue, Gyro-bias et LOE d'actionneur	52
4	Résultats expérimentaux	72

Nomenclatures

Liste des Abréviations

<i>UAVs</i>	Unmanned aerial vehicles
<i>VTOL</i>	Vertical Takeoff and Landing
<i>LOE</i>	Loss of Effectiveness
<i>LiPo</i>	Lithium polymer
<i>VRPN</i>	Virtual-Reality Peripheral Network
<i>UVF</i>	Unité de calcul en Virgule Flottante
<i>PCM</i>	Protection Circuit Module
<i>EKF</i>	Extended Kalman Filter
<i>CF</i>	Crazyflie
<i>CRTP</i>	Crazy RealTime Protocol
<i>LPS</i>	Loco Positioning System
<i>GPS</i>	Global Positioning System
<i>TWR</i>	Two Way Ranging
<i>TDoA</i>	Time Difference of Arrival
<i>UWB</i>	Ultra Wide Band

Liste des Symboles

p	Position du centre de masse de Quadrotor
v	Vitesse linéaire du corps rigide
Ω	Vitesse angulaire
R	Matrice de rotation de \mathcal{I} à \mathcal{B}
Q	Quaternion d'une rotation de \mathcal{I} à \mathcal{B}
\mathcal{I}	Repère inertiel
\mathcal{B}	Repère attaché au corps
Δ	Matrice représente LOE des l'actionneurs
μ_d	Matrice Matrice d'inertie
J	Entrée de contrôle virtuel
τ	Couples de commande
$\bar{\Omega}$	Loi de commande virtuelle
V	Fonction de Lyapunov candidate

Introduction générale

De nombreux systèmes mécaniques peuvent être modélisés comme un corps rigide ou une interconnexion de plusieurs corps rigides. Par exemple, les véhicules marins et aérospatiaux tels que les spatonefs, les véhicules aériens sans pilote (UAV), les satellites et les véhicules sous-marins autonomes peuvent être tous considérés comme des systèmes de corps rigides [1].

La conception d'algorithmes de contrôle efficaces d'attitude et d'altitude est d'une grande importance pour les applications nécessitant un positionnement précis des systèmes corps rigides tels que les satellites et les engins spatiaux.

Par conséquent, de nombreux chercheurs des communautés de l'aérospatiale et du contrôle ont mis au point les problèmes de contrôle, ce qui a mené à de nombreuses activités depuis les années 1960. Ces problèmes fondamentaux présentent de nombreux défis théoriques et pratiques.

Plusieurs problèmes liés à ce sujet ont été étudiés dans la littérature. Parmi ces problèmes, on peut citer le contrôle d'attitude du corps rigide qui a été étudié sous différentes perspectives et en utilisant différentes représentations d'attitude (voir, par exemple, [3–7]). L'autre problème qui a également été étudié dans la littérature, mais à un degré moindre que le problème de contrôle d'attitude, est le problème de suivi de position qui consiste à considérer la dynamique de translation et de rotation du système. En raison de la nature sous-actionnée du système, une méthode pratique courante consiste à procéder à une conception de contrôle hiérarchique où l'attitude du système est utilisée pour diriger la poussée afin de contrôler la position et la vitesse linéaires du système. L'attitude désirée est ensuite utilisée comme référence dans la dynamique de rotation pour générer les entrées de couple appropriées.

Le contrôle de la position des VTOL UAV a fait l'objet d'un intérêt considérable de la part de nombreux chercheurs. Cependant, malgré ces efforts considérables, il est encore possible d'améliorer le traitement efficace des perturbations externes, des paramètres inconnus des systèmes et des singularités de contrôle. Les contrôleurs présentés dans [8], [9] et [10], bien qu'ils réalisent des résultats de convergence globaux, mais ils ne prennent pas en compte les incertitudes des modèles, chose qui est cruciale dans les applications du monde réel. En pratique, l'inertie du système n'est pas connue avec précision en raison des variations de la masse et de la forme pendant le vol dues, par exemple, à la consommation du carburant et à la manipulation des objets *etc.* Les perturbations externes, l'imperfection des capteurs ainsi que les défaillances potentielles de l'actionneur entraînant une Perte d'Efficacité, connu sous l'acronyme anglais LOE (Lose Of Effectivness), des entrées

de commande sont également des facteurs importants à prendre en compte dans les applications pratiques. Dans [11], un contrôleur de suivi d'attitude adaptatif à temps fini, avec quelques propriétés de robustesse contre les perturbations externes et les incertitudes d'inertie, a été proposé. Dans [12], une loi de commande tolérante aux défauts utilisant une allocation de contrôle en ligne a été proposée pour gérer simultanément les perturbations externes, l'inertie inconnue et la perte d'efficacité des actionneurs, pour le problème du contrôle d'attitude. Un schéma de contrôle non linéaire adaptatif tolérant aux défauts a été proposé dans [13] pour le contrôle de l'attitude et de l'altitude d'un quadrotor afin de prendre en compte le LOE des actionneurs. Dans [14], un contrôleur adaptatif a été développé pour une classe des systèmes non linéaires qui présentent des défaillances d'actionneurs. Ce dernier a été appliqué au modèle longitudinal d'un avion.

La contribution principale de notre travail est la proposition d'un contrôleur de suivi adaptatif pour les UAV VTOL dans le cas d'une inertie du système inconnue, un biais gyroscopique et une perte d'efficacité partielle des actionneurs et en considérant aussi que le système subit des perturbations externes inconnues. Comparé à l'approche proposée par [12] appliquée au contrôle d'attitude avec des mesures de vitesse angulaire parfaites, notre travail aborde un problème plus complexe qui consiste en une commande en position d'un UAV, qui inclut notamment le problème de la commande d'attitude. De plus, dans le présent travail, nous considérons un biais gyroscopique inconnu chose qui n'a pas été considéré dans [12]. Comme c'est le cas dans [8] et [9], la stratégie de conception qu'on propose, repose sur la méthode du backstepping en raison de la structure de la dynamique de translation et de rotation du système.

Une implémentation expérimentale de deux approches nonlinéaires a été mise œuvre avec succès sur la plateforme open source crazyflie 2.0 afin de valider la faisabilité et tester les performances de ces deux approches.

PUBLICATIONS

Le travail réalisé dans cette thèse a fait l'objet des publications suivantes:

Publication internationale

- **Benrezki RR**, Tayebi A, Tadjine M. "Adaptive trajectory tracking control for VTOL-UAVs with unknown inertia, gyro-bias, and actuator LOE" *International Journal of Robust and Nonlinear Control*, 2018, vol. 28, no 17, p.

5247-5261.

<https://doi.org/10.1002/rnc.4308>. **IF : 3.856**

Conférences internationales

- **Benrezki RR**, Tadjine M, Yacef F and Kermia O. "Passive fault tolerant control of quadrotor UAV using a nonlinear PID" Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on *ROBIO'15*, zuhai, China 2015, pp. 1285-1290.

ORGANISATION DE LA THESE

Cette thèse est divisée en 4 chapitres. Le chapitre 1 présente une introduction aux UAV à savoir les différentes représentations d'attitude, la modélisation du corps rigide ainsi qu'une méthode de reconstruction d'attitude passive, qui sera utilisée lors de la synthèse de la loi de commande. À la fin de ce chapitre, nous présentons quelques outils et fondements mathématiques nécessaires pour la suite de notre étude.

Dans le chapitre 2 nous présentons la plateforme open source crazyflie 2.0 afin de montrer ses avantages très intéressants du point de vue architecture mécanique rigide et légère, son architecture électronique compacte et son architecture logicielle qui repose sur un noyau de système d'exploitation temps réel qui est le FreeRTOS. Le système de localisation en 3D LPS (Loco Sositioning System), conçu par la même société, est également à son tour présenté dans ce chapitre. Cette plateforme a été utilisée pour faire l'implémentation des lois de commande démontrées dans le chapitre 4.

Dans le chapitre 3, nous synthétisons une loi de commande adaptative de suivi de position pour les UAV VTOL dans le cas où l'inertie du système est inconnue, l'existence d'un biais gyroscopique et d'une perte d'efficacité partielle des actionneurs et en considérant aussi que le système subit des perturbations externes inconnues. Les performances du schéma de commande proposé dans ce chapitre sont validées par simulation.

Dans le chapitre 4, nous montrons la synthèse ainsi que les résultats d'implémentation de deux algorithmes de commande de suivi de trajectoire sur la plateforme crazyflie 2.0. Il s'agit d'un premier algorithme qui est une approche pratique nommée approche simplifiée. La deuxième approche repose sur la stratégie du backstepping.

Chapitre 1

Modélisation dynamique d'un
corps rigide, notion de
représentation d'attitude et outils
mathématiques

1.1 Introduction

Dans ce chapitre, nous allons passer en revue quelques fondements mathématiques utilisés dans le développement et l'analyse des lois de commande. Une composante majeure de ce contexte implique la définition des paramétrisations d'attitude utilisées pour décrire l'orientation du système (corps rigide). Une description de ces paramétrisations d'attitude et de certaines de ces propriétés est fournie dans la section 1.2. Cependant, pour une description plus complète de ces paramétrisations, le lecteur est invité à consulter [15],[2] et [16]. Les équations qui régissent sur la dynamique de l'UAV (modèle du système) sont définies dans la section 1.3. Dans la section 1.4, nous décrivons les algorithmes de reconstruction d'attitude, qui cherchent à récupérer une solution de forme fermée pour l'attitude d'un corps rigide, sans utiliser d'observateur ou de filtre. Dans les sections 1.5 et 1.6 nous allons présenter quelques définitions sur la stabilité des systèmes physiques et quelques outils mathématiques nécessaires, en donnant certains détails mathématiques préliminaires qui seront utiles lors de la discussion des algorithmes de contrôle et d'estimation dans le chapitre 3.

1.2 Représentation d'attitude

1.2.1 Repères inertiel et mobile

Pour représenter la position et l'orientation (position angulaire) d'un aéronef (corps rigide), nous utilisons des repères de coordonnées formés par un ensemble de trois vecteurs orthonormés qui obéissent à la règle de la main droite. Nous introduisons deux référentiels [17]

1. Le repère inertiel :

Le repère inertiel, noté \mathcal{I} , est attaché à un point de la surface de la terre. Nous associons à \mathcal{I} l'ensemble des axes $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$, où \hat{e}_1 pointe vers le nord, \hat{e}_2 pointe vers l'est et \hat{e}_3 pointe vers le centre de la terre supposé plat.

2. Le repère mobile :

Le repère mobile orthonormé, noté \mathcal{B} , est attaché au centre de gravité du véhicule. Nous associons à \mathcal{B} l'ensemble des axes $\{e_1, e_2, e_3\}$, où cette base

orthonormée est prise de telle sorte que e_1 est dirigé vers le devant le véhicule, e_2 pointe vers la droite et e_3 est dirigé vers le bas.

1.2.2 Angles d'Euler

La représentation en angles d'Euler est la méthode la plus intuitive pour la paramétrisation d'attitude. Introduite par Leonard Euler, cette méthode permet de paramétrer l'attitude du repère mobile liée au centre de gravité du corps par rapport au repère inertiel par trois angles de rotation successifs (Euler) autour des axes du repère inertiel. L'orientation d'un corps rigide dans un espace euclidien tridimensionnel est donnée par (ϕ, θ, ψ) où, les trois paramètres sont appelés roulis, tangage et lacet respectivement. Une rotation autour des axes \hat{x} , \hat{y} et \hat{z} est montrée ci-dessous :

$$R_{\hat{x}}(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix}$$
$$R_{\hat{y}}(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$
$$R_{\hat{z}}(\psi) = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Comme le vecteur de base orthogonal est modifié après chaque rotation, il est possible de représenter une orientation quelconque en effectuant une rotation autour de trois axes, noté ensemble symétrique, ou une rotation autour de deux axes seulement, noté ensemble asymétrique, à condition que les rotations successives ne se font pas autour du même axe. De cette manière, il est possible de définir douze séquences d'angles d'Euler, associées à 12 matrices d'attitude. Par exemple, effectuant 3 rotations dans l'ordre $(x - y - z)$, la matrice directeur résultante peut être écrite comme suit:

$$\begin{aligned}
 R &= R_{\hat{x}}(\phi)R_{\hat{y}}(\theta)R_{\hat{z}}(\psi) \\
 &= \begin{pmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{pmatrix}
 \end{aligned}$$

avec $c(\cdot)$ et $s(\cdot)$ désignant les opérateurs cosinus et sinus. Les ensembles d'angle d'Euler représentent les paramètres minimaux de paramétrisation d'attitude. Ils sont faciles à visualiser en raison de leurs définitions physiques claires; par contre ils présentent tous l'inconvénient d'être singuliers (Gimbal Lock). Pour les ensembles d'angle d'Euler asymétriques, $\theta = \pm\frac{\pi}{2}$ donne une singularité géométrique. Pour les ensembles d'angle d'Euler symétriques, la singularité géométrique se produit lorsque $\theta = \pm k\pi$ où k est un entier quelconque ($k = 1, 2, \dots$).

1.2.3 Matrice des Cosinus Directeurs (Rotation)

La matrice de rotation, également connue sous le nom de matrice de cosinus directeur (DCM), est la représentation la plus répandue de l'attitude d'un corps rigide, noté R chaque rotation du repère mobile \mathcal{B} par rapport au repère inertiel \mathcal{I} comme illustré dans Fig 1.1. Une matrice $R \in \mathbb{R}^{3 \times 3}$ orthogonale et tridimensionnelle est considérée comme une matrice de rotation si elle est élément dans l'ensemble :

$$SO(3) := \{R \in \mathbb{R}^{3 \times 3} | \det(R) = 1 | R^T R = R R^T = I_{3 \times 3}\} \quad (1.1)$$

Le groupe orthogonal spécial (*Special Orthogonal group*) $SO(3) \subset \mathbb{R}^{3 \times 3}$ est formé des matrices de rotation, sous l'opération de multiplication de matrice. L'abréviation SO fait référence aux propriétés des matrices de rotation. Cet ensemble est souvent utilisé car il offre une représentation globale et unique de l'orientation d'un repère de référence. Nous appelons cet ensemble, espace de rotation. La matrice de rotation $R \in SO(3)$ peut être utilisée pour mapper les coordonnées vectorielles d'une image à une autre. L'orientation du repère mobile \mathcal{B} par rapport au repère inertiel \mathcal{I} peut être décrite par trois vecteurs [18]

$${}^{\mathcal{I}}\hat{X}_{\mathcal{B}} = \begin{pmatrix} \hat{x}_{\mathcal{B}} \cdot \hat{x}_{\mathcal{I}} \\ \hat{y}_{\mathcal{B}} \cdot \hat{x}_{\mathcal{I}} \\ \hat{z}_{\mathcal{B}} \cdot \hat{x}_{\mathcal{I}} \end{pmatrix} \quad {}^{\mathcal{I}}\hat{Y}_{\mathcal{B}} = \begin{pmatrix} \hat{x}_{\mathcal{B}} \cdot \hat{y}_{\mathcal{I}} \\ \hat{y}_{\mathcal{B}} \cdot \hat{y}_{\mathcal{I}} \\ \hat{z}_{\mathcal{B}} \cdot \hat{y}_{\mathcal{I}} \end{pmatrix} \quad {}^{\mathcal{I}}\hat{Z}_{\mathcal{B}} = \begin{pmatrix} \hat{x}_{\mathcal{B}} \cdot \hat{z}_{\mathcal{I}} \\ \hat{y}_{\mathcal{B}} \cdot \hat{z}_{\mathcal{I}} \\ \hat{z}_{\mathcal{B}} \cdot \hat{z}_{\mathcal{I}} \end{pmatrix} \quad (1.2)$$

où, $\hat{x}_{\mathcal{I}}$, $\hat{y}_{\mathcal{I}}$ et $\hat{z}_{\mathcal{I}}$ sont les vecteurs de coordonnées attachés au repère \mathcal{I} . $\hat{x}_{\mathcal{B}}$, $\hat{y}_{\mathcal{B}}$ et $\hat{z}_{\mathcal{B}}$ sont les vecteurs de coordonnées attachés au repère \mathcal{B} , et ${}^{\mathcal{I}}\hat{X}_{\mathcal{B}}$, ${}^{\mathcal{I}}\hat{Y}_{\mathcal{B}}$ et ${}^{\mathcal{I}}\hat{Z}_{\mathcal{B}}$ sont des vecteurs unitaires orthogonaux l'un à l'autre. $u \cdot v = \|u\| \|v\| \cos \vartheta$ représente le produit scalaire et ϑ est l'angle entre les vecteurs u et v . La matrice de rotation R peut être définie comme suit [19]

$${}^{\mathcal{I}}R = \begin{pmatrix} {}^{\mathcal{I}}\hat{X}_{\mathcal{B}} & {}^{\mathcal{I}}\hat{Y}_{\mathcal{B}} & {}^{\mathcal{I}}\hat{Z}_{\mathcal{B}} \end{pmatrix} = \begin{pmatrix} \hat{x}_{\mathcal{B}} \cdot \hat{x}_{\mathcal{I}} & \hat{y}_{\mathcal{B}} \cdot \hat{x}_{\mathcal{I}} & \hat{z}_{\mathcal{B}} \cdot \hat{x}_{\mathcal{I}} \\ \hat{x}_{\mathcal{B}} \cdot \hat{y}_{\mathcal{I}} & \hat{y}_{\mathcal{B}} \cdot \hat{y}_{\mathcal{I}} & \hat{z}_{\mathcal{B}} \cdot \hat{y}_{\mathcal{I}} \\ \hat{x}_{\mathcal{B}} \cdot \hat{z}_{\mathcal{I}} & \hat{y}_{\mathcal{B}} \cdot \hat{z}_{\mathcal{I}} & \hat{z}_{\mathcal{B}} \cdot \hat{z}_{\mathcal{I}} \end{pmatrix} \quad (1.3)$$

La correspondance entre un vecteur $b \in \mathbb{R}^3$ exprimé en \mathcal{B} et un vecteur $r \in \mathbb{R}^3$ exprimé en \mathcal{I} peut être écrite comme

$$r = Rb \quad (1.4)$$

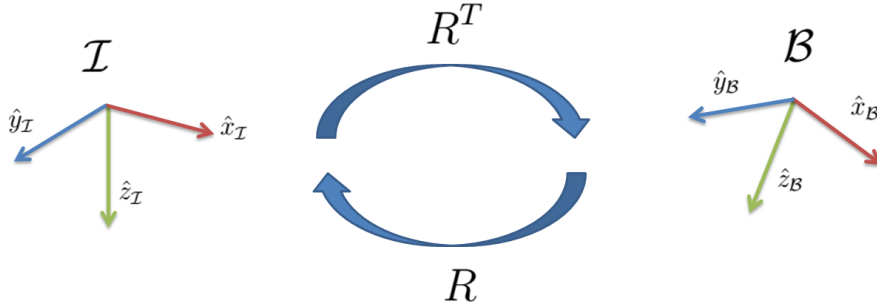


Figure 1.1: Rotation de \mathcal{B} par rapport à \mathcal{I}

1.2.4 Quaternion unitaire

La représentation quaternion unitaire d'une attitude est une paramétrisation à quatre éléments définie sur l'espace défini par

$$\mathcal{Q} := \{Q = (q_0, q) \in \mathbb{R} \times \mathbb{R}^3 \mid q_0^2 + q^T q = 1\} \quad (1.5)$$

1.2 Représentation d'attitude

Pour désigner le quaternion unitaire, on utilise la notation $Q = (q_0, q) \in \mathcal{Q}$, où $q_0 \in \mathbb{R}$ et $q \in \mathbb{R}^3$.

Le quaternion unitaire est souvent considéré comme une représentation *axis-angle* du fait que l'orientation relative de deux référentiels peut toujours être exprimée par une seule rotation d'un angle ϕ autour d'un axe (de longueur unitaire) \hat{e} , où dans ce cas le quaternion unitaire peut être écrit comme [17]

$$Q = (q_0, q) = (\cos(\phi/2), \sin(\phi/2)\hat{e}) \quad (1.6)$$

En raison de l'utilisation de quatre éléments, le quaternion unitaire est un paramétrage excessif de l'espace de rotation $SO(3)$. C'est-à-dire que la transformation à partir de $\mathbb{S}^3 \rightarrow SO(3)$ est une carte à deux voix. Cette caractéristique entraîne certaines difficultés dues à la multiplicité des solutions d'équilibre (qui se rapportent au même point dans l'espace de rotation) lors de l'utilisation de cette représentation. Cela est dû au fait qu'un repère de coordonnées dont l'orientation est décrite par le quaternion unitaire Q est physiquement équivalent au repère de coordonnées dont l'orientation est définie par le quaternion unitaire $-Q$ [20].

Par conséquent, si l'orientation de deux repères mobiles différents (chacun pris par rapport au repère inertiel) est décrite par le quaternion unitaire Q et $-Q$, l'orientation des deux repères mobiles est physiquement équivalente et ils ont la même matrice de rotation en utilisant le paramétrage $SO(3)$.

En fait, la matrice de rotation à neuf éléments peut être exprimée en fonction du quaternion unitaire à travers le mappage de *Rodrigues* $\mathcal{R}(Q) : \mathcal{Q} \rightarrow SO(3)$ est donnée par

$$\mathcal{R}(Q) = I_{3 \times 3} + 2S(q)^2 - 2q_0S(q), \quad (1.7)$$

à partir de laquelle on peut facilement confirmer $R(Q) = R(-Q)$

La composition de deux rotations décrites par deux quaternions Q_1 et Q_2 résulte par le *produit* des deux quaternions, noté $Q_3 = (q_{03}, q_3) \in \mathcal{Q}$ est donné par

$$Q_3 = Q_1 \odot Q_2 = (q_{01}q_{02}, q_{01}q_2 + q_{02}q_1 + S(q_1)q_2) \quad (1.8)$$

Un élément d'identité $Q_I = (1, 0, 0, 0)$ est le résultat du produit d'un quaternion unitaire $Q = (q_0, q)$ avec son inverse $Q^{-1} = (q_0, -q)$.

En utilisant la propriété de transformation vectorielle décrite en (1.4). Pour un vecteur $v \in \mathbb{R}^3$ donné, où $v_{\mathcal{I}} = Rv_{\mathcal{B}}$, on peut écrire

$$\bar{v}_{\mathcal{I}} = Q \odot \bar{v}_{\mathcal{B}} \odot Q^{-1} \quad (1.9)$$

où Q est le quaternion unitaire associé à la matrice de rotation R et $\bar{v}_{\mathcal{I}} = (0 \quad v_{\mathcal{I}})$ et $\bar{v}_{\mathcal{B}} = (0 \quad v_{\mathcal{B}})$

De nombreux chercheurs s'appuient sur la représentation des quaternions en raison des avantages significatifs qu'elle offre par rapport à la représentation $SO(3)$. Un avantage peut être attribué aux propriétés mathématiques uniques du quaternion, qui donnent souvent des démonstrations simplifiées par rapport aux travaux qui reposent uniquement sur le $SO(3)$. Un autre avantage du quaternion unitaire provient du fait qu'il est représenté par un vecteur (de quatre éléments) au lieu d'une matrice orthogonale 3×3 , ce qui peut être utile. Par exemple, lors de la mise en œuvre d'un algorithme de contrôle basé sur le quaternion unitaire (ou lors des simulations), la préservation des propriétés de la représentation d'attitude devient importante. Pour s'assurer qu'un vecteur conserve les propriétés d'un quaternion, on peut simplement normaliser le vecteur en faisant la division par sa norme. Par ailleurs, préserver les propriétés $SO(3)$ d'une matrice 3×3 implique des algorithmes plus compliqués, tels que l'algorithme d'orthonormalisation de Gram-Schmidt (voir, par exemple, [21]).

Quaternion unitaire et Angles d'Euler

Le quaternion unitaire peut être généré en utilisant les angles d'Euler comme suit [18] :

$$Q = \begin{pmatrix} q_0 \\ q \end{pmatrix} = \begin{pmatrix} c(\frac{\phi}{2})c(\frac{\theta}{2})c(\frac{\psi}{2}) + s(\frac{\phi}{2})s(\frac{\theta}{2})s(\frac{\psi}{2}) \\ s(\frac{\phi}{2})c(\frac{\theta}{2})c(\frac{\psi}{2}) - c(\frac{\phi}{2})s(\frac{\theta}{2})s(\frac{\psi}{2}) \\ c(\frac{\phi}{2})s(\frac{\theta}{2})c(\frac{\psi}{2}) + s(\frac{\phi}{2})c(\frac{\theta}{2})s(\frac{\psi}{2}) \\ c(\frac{\phi}{2})c(\frac{\theta}{2})s(\frac{\psi}{2}) - s(\frac{\phi}{2})s(\frac{\theta}{2})c(\frac{\psi}{2}) \end{pmatrix} \quad (1.10)$$

avec $c(\cdot)$ et $s(\cdot)$ désignant les opérateurs cosinus et sinus respectivement. Le quaternion peut également être converti en angles d'Euler comme suit :

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} \text{atan2}(2\eta q_1 + q_2 q_3), 1 - 2(q_1^2 + q_2^2) \\ \arcsin(2(\eta q_2 - q_3 q_1)) \\ \text{atan2}(2(\eta q_3 + q_1 q_2), 1 - 2(q_2^2 + q_3^2)) \end{pmatrix} \quad (1.11)$$

ou, $\text{atan2}(\cdot) : \mathbb{R} \times \mathbb{R} \rightarrow (-\pi, \pi]$ désigne la tangente inverse à quatre quadrants. Il est facile de vérifier que $Q = Q_I$ pour $\phi = 0, \theta = 0$ et $\psi = 0$

1.3 Modèle dynamique d'un VTOL

Le modèle dynamique d'un UAV VTOL (Fig 1.2 présente l'exemple d'un ducted fan) est composé de deux parties. Une première partie qui représente la dynamique de translation et une second partie représente la dynamique de rotation. Cette décomposition est utile lors de la synthèse du contrôleur car la connaissance de la nature cascade du système permettra de diviser la problématique de conception du contrôleur en des sou-problèmes simples à résoudre, comme nous allons le voir dans le chapitre 3. Les commande synthétisées concerne tous les VTOL en générale, pour plus de détaille sur la distribution d'énergie dans le cas du Ducted fan le lecteur est orienté vers [22]. Ci-dessous on présente le modèle de la dynamique de translation et de rotation du système.

Dynamique de la translation

$$\dot{p} = v \quad (1.12)$$

$$\dot{v} = g\hat{e}_3 - \frac{T}{m}R^T\hat{e}_3 \quad (1.13)$$

Dynamique de la rotation

$$\dot{Q} = \frac{1}{2} \begin{bmatrix} -q^T \\ q_0 I_{3 \times 3} + S(q) \end{bmatrix} \Omega \quad (1.14)$$

$$J\dot{\Omega} = -S(\Omega)J\Omega + \Delta\tau + d \quad (1.15)$$

où p et v représentent respectivement la position et la vitesse linéaire du repère mobile \mathcal{B} par rapport au repère inertiel \mathcal{I} , exprimée en \mathcal{I} . $T \in \mathbb{R}$ est la poussée du système, qui est supposée être dirigée le long de axe e_3 . g désigne l'accélération due à la gravité et m représente la masse du système, $J \in \mathbb{R}^{3 \times 3}$ dénote la matrice d'inertie du système et $\Omega \in \mathbb{R}^3$ est la vitesse angulaire exprimés en \mathcal{B} . L'entrée de

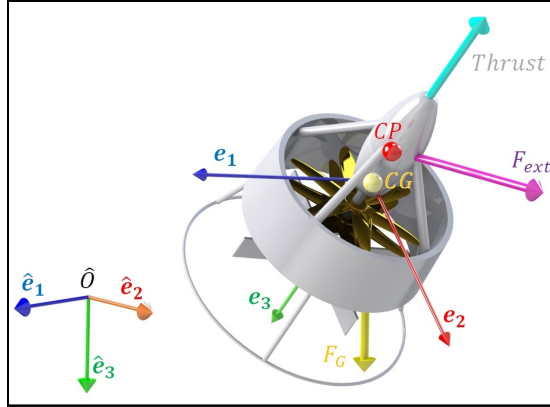


Figure 1.2: Repère inertiel, repère fixe et forces exogènes

commande du système notée $\tau \in \mathbb{R}^3$ est le couple exogène appliqué au VTOL et $d \in \mathbb{R}^3$ est le couple de perturbation externe.

1.4 Reconstruction d'attitude

En général, le problème d'extraction d'attitude pour une seule paire de vecteurs peut être formulé comme suit: étant donné deux vecteurs u et v , où $\|u\| = \|v\|$, nous souhaitons trouver une orientation $R_d \in SO(3)$ qui satisfait

$$R_d u = v \quad (1.16)$$

Notez que (1.16) peut également être exprimé en utilisant le quaternion unitaire comme suit:

$$\begin{bmatrix} 0 \\ v \end{bmatrix} = Q_d^{-1} \odot \begin{bmatrix} 0 \\ u \end{bmatrix} \odot Q_d \quad (1.17)$$

où nous cherchons une expression pour le quaternion Q_d . Dans [23], une solution intuitive à ce problème a été décrite. Une solution similaire a été proposée dans le lemme suivant:

Lemme 1.1. [9] Étant donné deux vecteurs u et v , où $\|u\| = \|v\| \neq 0$, et où $u \neq -v$; alors, une solution pour l'unité quaternion $Q_d = (q_{d0}, q_d)$ qui satisfait (1.17) existe et est donnée par

$$q_{d0} = \frac{1}{\|u\|} \sqrt{\frac{\|u\|^2 + u^T v}{2}} \quad (1.18)$$

$$q_d = \frac{1}{\|u\|} \sqrt{\frac{1}{2(\|u\|^2 + u^T v)}} S(v)u \quad (1.19)$$

Remarque 1.1. Comme discuté dans [23], cette extraction utilise le vecteur qui est orthogonal à la fois à u et à v comme axe de rotation. En raison de ce choix, cette méthode extrait la solution au problème qui minimise l'angle de rotation. Compte tenu de la solution du quaternion unitaire par (1.18) et (1.19), une matrice de rotation qui vérifie (1.16) peut également être trouvée en utilisant la méthode présentée dans [24].

1.5 Concept de la stabilité

L'étude des systèmes dynamiques pose différents types de problèmes de stabilité. Nous présentons dans cette section les principes de base de la stabilité des points d'équilibre. Nous nous intéressons principalement à la méthode directe de Lyapunov, une méthode qui a trouvé le total succès dans la communauté des automaticiens. Cette dernière est basée sur le concept d'énergie, elle détermine la stabilité des systèmes en construisant une fonction scalaire des états du système dont on examinera la variation temporelle [25]. L'étude de la stabilité diffère en fonction de la nature autonome ou non-autonome des systèmes. Le concept de stabilité de Lyapunov est valable pour ces deux types de systèmes. Nous allons présenter la méthode directe de Lyapunov pour les systèmes autonomes ainsi que le théorème de LaSalle qui permet de compléter la démonstration de la stabilité asymptotique. La stabilité au sens de Lyapunov est étendue aux systèmes non autonomes où plusieurs variantes existent comme les concepts de stabilité uniforme, asymptotique uniforme et exponentielle. Nous présenterons aussi le Lemme de Barbalat, une fois utilisé correctement pour les systèmes dynamiques, en particulier les systèmes non autonomes, il peut permettre de résoudre de manière satisfaisante de nombreux problèmes de stabilité asymptotiques [26].

1.5.1 Système autonome

Un système autonome peut être écrit sous la forme suivante

$$\dot{x} = f(x), \quad x(t_0) = x_0 \quad (1.20)$$

où $f : D \rightarrow \mathbb{R}^n$ est localement lipschitienne dans le domaine $D \subset \mathbb{R}^n$. Supposant que $x_e \in D$ et un point d'équilibre du système (1.20) (*i.e.* $f(x_e) = 0$). Sans perte de généralités, l'étude de stabilité vas être présenté en prenant le cas où le point d'équilibre est l'origine, *i.e.* $x_e = 0$. Un simple changement de variable est présenté dans [25] montre comment ramené le poin d'équilibre à l'origine dans le cas où il est non nul. L'étude de la stabilité au sens de Lyapunov se fait de la manière suivante

Définition 1.1. [25] L'état d'équilibre $x = 0$ est dit stable si, pour tout $R > 0$, il existe $r > 0$, tel que si $\|x(0)\| < r$, alors $\|x(t)\| < R$ pour tout $t \geq 0$. Sinon, le point d'équilibre est dit instable.

Définition 1.2. [25] Si le point d'équilibre est stable et que r peut être choisi de sorte que

$$\|x(0)\| < r \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0 \quad (1.21)$$

alors le point d'équilibre x_e est dit asymptotiquement stable

Théorème 1.1. [25] Soit $x_e = 0$ un point d'équilibre du système (1.20) et $D \subset \mathbb{R}^n$ un domaine qui contient ce point d'équilibre. Soit $V : D \rightarrow \mathbb{R}$ une fonction continuellement différentiable telle que

- $V(0) = 0$ et $V(x) > 0$ dans $D - \{0\}$
- $\dot{V}(x) \leq 0$ dans D

Alors $x_e = 0$ est stable.

De plus, si $\dot{V} < 0$ dans $D - \{0\}$

Alors x_e est asymptotiquement stable.

1.5.1.1 Principe d'invariance de La Salle-Krasovskii

Il permet de démontrer la stabilité asymptotique quand la dérivé de la fonction de Lyapunov est $\dot{V} \leq 0$ (semi définit négative).

Théorème 1.2. Soit $\Omega_c = \{x \in D | V(x) \leq c\}$ positive et invariant par rapport à (1.20) où $V : D \rightarrow \mathbb{R}$ est continuellement différentiable tel que ça dérivé est semi définit négative $\dot{V} \leq 0$. Soit $S = \{x \in \Omega_c | \dot{V} = 0\}$ et soit M le plus grand ensemble invariant dans S alors pour tout $x(0) \in \Omega_c$, $x(t) \rightarrow M$

1.5.2 Système non-autonome

Un système non-autonome peut être écrit sous la forme suivante

$$\dot{x} = f(t, x) \quad (1.22)$$

dont $x_e = 0$ est son point d'équilibre.

Définition 1.3. [25] l'origine est un point d'équilibre stable au sens de Lyapunov pour le système si $\forall \epsilon > 0, \forall t_0 \geq 0$, s'il existe un scalaire positif $\delta(\epsilon, t_0)$ tel que $\|x_0\| \leq \delta(\epsilon, t_0) \Rightarrow \|x(t, t_0, x_0)\| \leq \epsilon, \forall t \geq t_0$. Voir Fig 1.3.

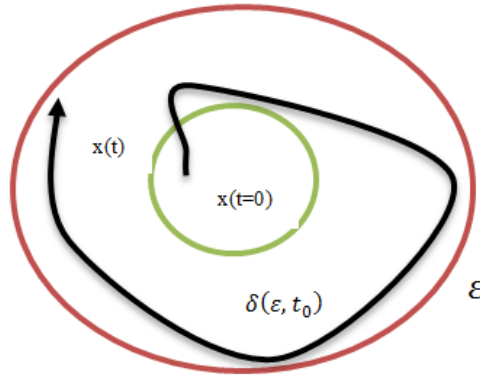


Figure 1.3: Equilibre stable

Définition 1.4. [27] Une fonction continue $\alpha : [0, \infty) \rightarrow [0, \infty)$ est de class \mathcal{K} ($\alpha \in \mathcal{K}$) si elle strictement croissante et que $\alpha(0) = 0$.

De plus si $\alpha(r) \rightarrow \infty$ et $r \rightarrow \infty$ alors α est de class \mathcal{K}_∞ ($\alpha \in \mathcal{K}_\infty$).

Définition 1.5. [27] Une fonction $\beta : [0, \infty) \times [0, \infty) \rightarrow [0, \infty)$ est de class \mathcal{KL} si :

- $\beta(\cdot, S)$ est de class \mathcal{K} pour chaque s fixé
- $\beta(r, \cdot)$ est décroissante et $\beta(r, s) \rightarrow 0$ quand $S \rightarrow \infty$ pour chaque r fixé.

Définition 1.6. [27] $x = 0$ est uniformément stable si il existe une fonction class \mathcal{K} $\alpha(\cdot)$ et une constante c tel que

$$\|x(t)\| \leq \alpha\|x(t_0)\|, \quad \forall t \geq t_0 \quad \text{et} \quad \|x(t_0)\| \leq c$$

si $c = \infty$ alors $x = 0$ est globalement uniformément stable

Définition 1.7. $x = 0$ est uniformément asymptotiquement stable si il existe une fonction class \mathcal{K} $\beta(\cdot, \cdot)$ tel que

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0) \quad \forall t \geq t_0 \quad \text{et} \quad \|x(t_0)\| \leq c$$

Si $c = \infty$ (pour toute condition initiale) alors $x = 0$ est globalement asymptotiquement uniformément stable

Définition 1.8. $x = 0$ est uniformément exponentiellement stable si $\beta(r, S) = kre^{-\lambda S}$

pour $k > 0, \lambda > 0$ $\|x(t)\| \leq k\|x(0)\|e^{-\lambda(t-t_0)}$

Théorème 1.3. [27] Considérons le système (1.22), Soit $D \subset \mathbb{R}^n$ incluant x_e . Supposons que $f(t, x)$ est continue dans t et qu'elle est localement Lipschitienne dans x pour tout $t \geq 0$ et $x \in D$

Soit $V(t, x)$ une fonction continuellement différentiable, tel que

- $W_1(x) \leq V(t, x) \leq W_2(x)$
- \dot{V} est semi-définie négative i.e. $\dot{V} = \frac{\partial V}{\partial t} + \frac{\partial V \dot{x}}{\partial x} \leq 0$

pour tout $t \geq 0$ et $x \in D$, où $W_1(x)$ et $W_2(x)$ sont deux fonctions définies positives dans D . Alors $x_e = 0$ est uniformément stable.

Théorème 1.4. Si x_e est un point d'équilibre du système(1.22) uniformément stable et qu'il existe une fonction W_3 définie positive. Alors si

$$\dot{V}(t, x) \leq -W_3(x) \tag{1.23}$$

x_e est uniformément asymptotiquement stable.

Si $D \in \mathbb{R}^n$ et $W_1(x)$ est radialement non bornée. Alors x_e est globalement uniformément asymptotiquement stable.

Théorème 1.5. Considérons le système (1.22), Soit $D \subset \mathbb{R}^n$ incluant x_e . Supposons que $f(t, x)$ est continue dans t et qu'elle est localement Lipschitienne dans x pour tout $t \geq 0$ et $x \in D$

Soit $V(t, x)$ une fonction continuellement différentiable, tel que

1.5 Concept de la stabilité

- $k_1 \|x\|^a \leq V(t, x) \leq k_2 \|x\|^a$
- $\dot{V}(t, x) \leq -k_3 \|x\|^a$

pour tout $t \geq 0$ et $x \in D$ où a, k_1, k_2, k_3 sont strictement positive. Alors le point d'équilibre x_e est uniformément exponentiellement stable.

Si $D \in \mathbb{R}^n$ alors le point d'équilibre globalement uniformément asymptotiquement stable.

1.5.2.1 Lemme de Barbalat

Certaines méthodes mathématiques permettent de démontrer la convergence du système commandé lorsque la méthode d'analyse de la stabilité donne un résultat incomplet.

Dans le cas où la dérivée de la fonction de Lyapunov est semi-définie négative pour un système non-autonome, le lemme de Barbalat permet de prouver la convergence vers zéro de l'erreur résultante d'une boucle de commande ou même d'une erreur d'estimation comme dans le cas de la commande adaptative directe ou indirecte [28].

Lemme 1.2. [26] Si la fonction différentiable $f(t)$ a une limite finie quand $t \rightarrow \infty$ et si \dot{f} est uniformément continue, alors $\dot{f}(t) \rightarrow 0$ quand $t \rightarrow \infty$

L'application de ce lemme pour l'analyse de la stabilité de Lyapunov ce fait grâce au lemme suivant.

Lemme 1.3. [26] Si la fonction scalaire $V(x, t)$ satisfait les conditions suivantes

- $V(x, t)$ a une borne inférieure
- $\dot{V}(x, t)$ est semi définie négative
- $\dot{V}(x, t)$ est uniformément continue dans le temps

Alors $\dot{V}(x, t) \rightarrow 0$ quand $t \rightarrow \infty$

1.5.3 Inégalité de Young

Cette inégalité permet de réécrire le produit de deux variables a et b réelles positives ou nulles, pour tous réels p et q strictement positifs tels que $\frac{1}{p} + \frac{1}{q} = 1$, on a :

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q} \tag{1.24}$$

1.6 Outils mathématiques

1.6.1 La Matrice Antisymétrique

Tout au long de ce travail, nous utilisons de manière extensive la matrice asymétrique. En fait, la maîtrise des propriétés de cette matrice est très important et comme M.D. Shuster avait dit, «on pourrait dire sans justifier, que l'étude théorique de l'attitude est l'étude de la matrice antisymétrique $[[u]]$ » [15]. Soit $q = [q_1 \ q_2 \ q_3]^T \in \mathbb{R}^3$ et $p = [p_1 \ p_2 \ p_3]^T \in \mathbb{R}^3$ désignent deux vecteurs arbitraires. La matrice asymétrique $S(q) : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \times \mathbb{R}^3$ est donnée par

$$S(q) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \quad (1.25)$$

Ci-dessous quelques propriétés utiles de cette matrice sont données

$$S(q)^T = -S(q), \quad (1.26)$$

$$S(q)p = -S(p)q, \quad (1.27)$$

$$S(q)q = 0, \quad (1.28)$$

$$S(q)S(p) = -(q^T p)I + qp^T. \quad (1.29)$$

Pour une liste plus complète, le lecteur est renvoyé à [15], page 446 :

1.6.2 Projection

Tout au long de ce manuscrit, nous utiliserons le mécanisme de projection dans l'adaptation pour nous assurer que les estimations des paramètres restent dans un domaine précis. Cela va permettre de garantir un certain niveau de robustesse en évitant que les estimations des paramètres ne dérivent pas dans le temps en présence de mesures bruitées. Cela permettra également, dans certains cas, d'empêcher les estimations de paramètres de s'annuler et d'éviter les problèmes de singularité dans la loi de commande. Soit $\hat{\vartheta} \in \mathbb{R}^m$ l'estimation de $\vartheta \in \mathbb{R}^m$ et $\tilde{\vartheta} = \vartheta - \hat{\vartheta}$. La fonction de projection est définie comme suit [29]:

$$\text{Proj}(\hat{\vartheta}, y, \mathcal{P}) = \begin{cases} y & \text{if } \mathcal{P}(\hat{\vartheta}) \leq 0 \text{ or } \nabla_{\hat{\vartheta}} \mathcal{P}^T y \leq 0 \\ \left(I - c(\hat{\vartheta}) \frac{\nabla_{\hat{\vartheta}} \mathcal{P} \nabla_{\hat{\vartheta}} \mathcal{P}^T}{\nabla_{\hat{\vartheta}} \mathcal{P}^T \nabla_{\hat{\vartheta}} \mathcal{P}} \right) y & \text{otherwise} \end{cases} \quad (1.30)$$

où $\nabla_{\hat{\vartheta}} \mathcal{P}(\hat{\vartheta}) = \frac{\partial \mathcal{P}(\hat{\vartheta})}{\partial \hat{\vartheta}}$ et $c(\hat{\vartheta}) = \min\{1, \frac{\mathcal{P}(\hat{\vartheta})}{\epsilon}\}$, pour certains $\epsilon > 0$, avec $\mathcal{P}(\hat{\vartheta})$ étant une fonction convexe qui sera définie plus tard. Cette projection garantit que pour tout $\hat{\vartheta}(0) \in \Pi := \{\hat{\vartheta} \in \mathbb{R}^3 \mid \mathcal{P}(\hat{\vartheta}) \leq 0\}$, on a $\hat{\vartheta}(t) \in \Pi_\epsilon := \{\hat{\vartheta} \in \mathbb{R}^3 \mid \mathcal{P}(\hat{\vartheta}) \leq \epsilon\}$, pour tous $t \geq 0$. De plus, la fonction de projection bénéficie des propriétés utiles suivantes: $\tilde{\vartheta}^T \text{Proj}(\hat{\vartheta}, y, \mathcal{P}) \geq \tilde{\vartheta}^T y$ et $\|\text{Proj}(\hat{\vartheta}, y, \mathcal{P})\| \leq \|y\|$.

1.7 Conclusion

Dans ce chapitre, les différentes représentation d'attitude ont été présentées en énumérant les avantages et les inconvénients de chaque méthode à fin de justifier le choix des quaternions. Le modèle dynamique de la translation et de la rotation du VTOL UAV a été présenté suivie par la présentation d'un lemme qui montre la reconstruction d'attitude et de poussée à partir d'une accélération désirée. À la fin du chapitre nous avons présenté quelques fondements mathématiques nécessaires à savoir les principes fondamentaux du concept de la stabilité, le lemme de Barbalat et le mécanisme de projection. Ces outils seront nécessaires dans le design et l'analyse de la stabilité dans les chapitres 3 et 4.

Chapitre 2

La plateforme Crazyflie 2.0

2.1 Introduction

Récemment, diverses plateformes open source traitant des UAV de type quadrotors ont été développées. Une étude comparative de ces plateformes quadrotor est donnée dans [30]. Dans ce chapitre nous allons présenter l'architecture matérielle et logicielle de la plateforme nano quad "Crazyflie 2.0" (voir Fig 2.1) conçue par la société suédoise Bitcraze. Cette plateforme sera utilisée par la suite pour implémenter et valider des algorithmes de commandes.



Figure 2.1: Crazyflie 2.0

2.2 Crazyflie

2.2.1 Caractéristiques générales

Le Crazyflie 2.0 est un projet open-source, le code source ainsi que les informations sur le matériel sont disponibles gratuitement en téléchargement. De nouvelles fonctionnalités sont ajoutées grâce aux mises à jour régulières, celles-ci peuvent être effectuées via la liaison radio ou via Bluetooth. Avec le Bluetooth Low Energy, ce

2.2 Crazyflie

module pourra être contrôlé avec un smartphone iOS ou Android ou à partir d'un ordinateur utilisant le module Crazyradio PA. Ce dernier offre la possibilité de gérer les cartes d'extensions installées, de contrôler les paramètres de vols, ainsi que de diriger l'appareil par le biais d'un joystick ou d'une manette disposant de 4 axes analogiques.

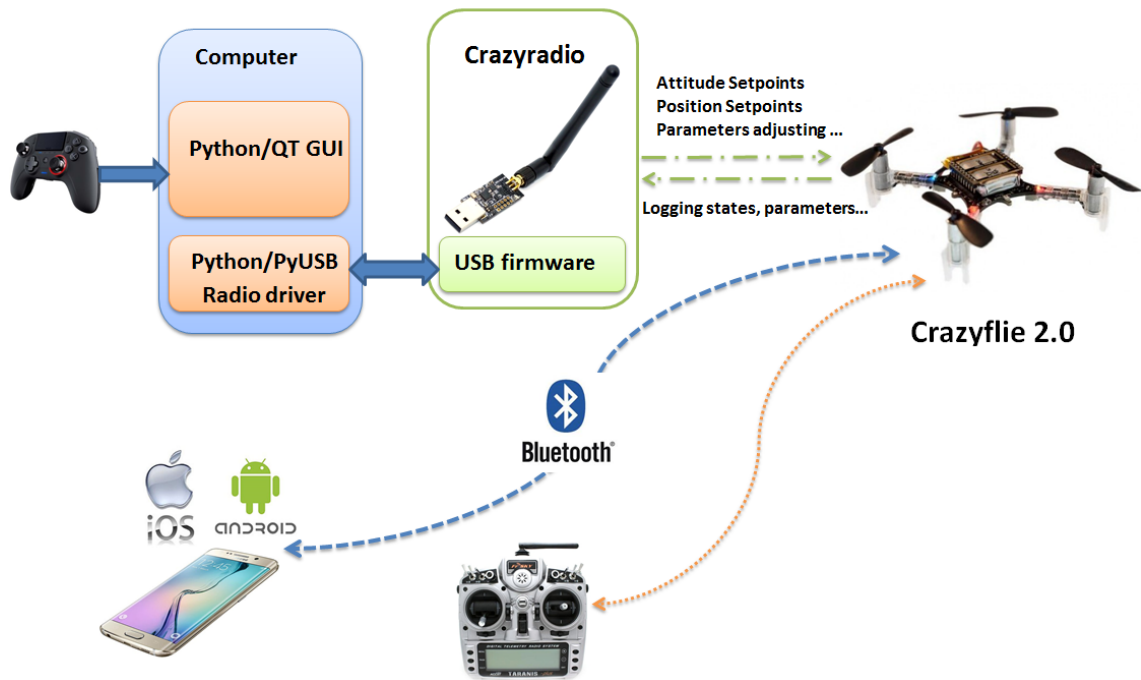


Figure 2.2: Architecture du CF

Au niveau Application (Haut niveau), la plateforme se divise en deux systèmes distincts comme le montre la figure 2.2. On trouve le Firmware (section 2.4) qui réside dans le drone lui-même et qui est chargé des fonctions critiques comme la mesure des capteurs, l'exploitation des données et la stabilisation de vol en plus de d'autres tâches moins prioritaires. Le deuxième système qui permet la gestion du CF est le PC client (section 2.6) qui est exécuté sur une station de base (PC ou autres) séparée du drone. Ce client entretient les paramètres du contrôle du vol, le flot de données de location VRPN (Virtual-Reality Peripheral Network) ainsi que l'enregistrement des données du log. Ces deux systèmes communiquent entre eux à travers un port USB radio qui est branché au PC. Le tableau ci-dessous récapitule les détails techniques du nano quad.

2.2 Crazyflie

- Poids: 27 gr	
- Dimensions: 92 x 92 x 29 (de moteur à moteur et avec le support)	
Compatible	<ul style="list-style-type: none">- iOS, Android (avec le Bluetooth LE)- Windows, MacOSX, Linux (nécessite le Crazyradio ou Crazyradio PA)
Radio	<ul style="list-style-type: none">- amplificateur radio de 20 dBm testé à plus de 1 km (portée à vue) avec le Crazyradio PA- bluetooth Low Energy compatible iOS et Android- radio compatible avec Crazyflie et Crazyradio
Microcontrôleurs	<ul style="list-style-type: none">- STM32F405, microcontrôleur principal (Cortex-M4, 168 MHz, 192 KB SRAM, 1 MB de mémoire flash)- nRF51822, gestion alimentation et radio (Cortex-M0, 32 MHz, 16 KB SRAM, 128 KB de mémoire flash)- EEPROM 2 KB
Connecteur USB	<ul style="list-style-type: none">- chargeur LiPo intégré avec trois modes (100 mA, 500 mA, 980 mA)- utilisation en USB OTG possible
Capteurs	<ul style="list-style-type: none">- gyroscope 3 axes (MPU-9250)- accéléromètre 3 axes (MPU-9250)- magnétomètre 3 axes (MPU-9250)- capteur de pression haute précision (LPS25H)
Caractéristiques de vol	<ul style="list-style-type: none">- 7 minutes de temps de vol maxi avec la batterie originale- 40 minutes de recharge avec la batterie originale- charge maxi recommandée: 15 grammes
Connecteurs supplémentaires	<ul style="list-style-type: none">- Vcc (3 Vcc, 100 mA max)- GND- VCOM (VBAT ou VUSB, 1 A max)- VUSB (entrée et sortie)- I2C (400 kHz)- SPI- 2 x UART- 4 x GPIO

2.2.2 Microcontrôleurs

Le crazyflie 2.0 est équipé de deux microcontrôleurs. Le premier est un STM32F405 qui embarque un processeur ARM Cortex-M4 et est utilisé pour exécuter l'application principale. Le processeur a une architecture 32 bits et peut fonctionner à 168 MHz. L'ARM Cortex-M4 intègre une unité de calcul en virgule flottante (UVF) qui supporte tous les types de données et les instructions de traitement de données [31]. Le deuxième microcontrôleur du Crazyflie s'occupe de la gestion de l'alimentation et de la radio. Ce microcontrôleur est un ARM Cortex M0 (nRF51822) qui fonctionne à 32MHz et possède un module de communication Bluetooth. La figure 2.3 illustre le rôle de chaque microcontrôleur [32].

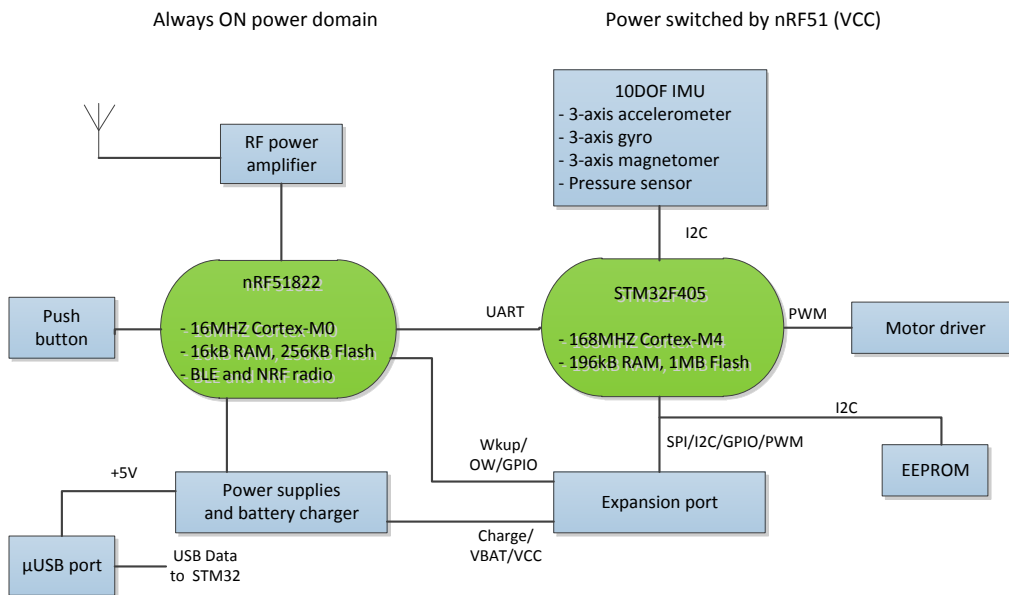


Figure 2.3: Architecture du système CF

2.2.3 Centrale inertielle (IMU)

Le crazyflie 2.0 est équipé de l'IMU MPU-9250. Cette dernière renferme un gyroscope 3 axes, un accéléromètre 3 axes et un magnétomètre 3 axes. La configuration expérimentale se base entièrement sur l'IMU pour avoir des mesures d'orientation. Ces mesures sont données dans le repère du capteur qui est orienté dans la configuration en "X" où son axe des x étant orienté entre les moteurs 4 et 1.

2.2.4 Autres capteurs

Le crazyflie est équipé, également, d'un magnétomètre et d'un baromètre. Cependant, le protocole radio utilisé limite la taille de la charge utile des paquets envoyés du Crazyflie au Crazyradio à 29 bytes. Ceci correspond à un maximum de sept nombre en virgule flottante sur le microcontrôleur ARM.

2.2.5 Batterie et alimentation

La batterie utilisée par le Crazyflie 2.0 est une Lithium-Polymer (LiPo) à cellule unique (3.7V, 240mAh). Ce type de batterie est le plus populaire dans l'industrie des R/C et offre le meilleur compromis poids énergie stockée. La batterie est fournie avec un système de protection (PCM) attaché à elle pour éviter la surcharge et le court-circuit. Le taux de décharge de la batterie est de 15C, ce qui permet une durée de vol de 7 minutes environ.

2.2.6 Moteurs

Contrairement à la plupart des drones à hélice qui possèdent des moteurs brushless (sans balais), le crazyflie 2.0 possède quatre moteurs à balais à courant continu. Les moteurs sont de type coreless (avec le rotor sans noyau de fer) ce qui en théorie évite la perte d'énergie et assure une accélération plus importante. Ils peuvent produire 12000rpm par volt avec une tension nominale de 4.2V.

2.2.7 Hélices

Le Crazyflie possède des hélices conventionnelles 45mm en plastique. Elles ont tendance à se tordre à cause des collisions et ont besoin d'être changées régulièrement pour des conditions de vol optimales.

2.3 Crazyradio

Pour communiquer avec le crazyflie il est approprié d'utiliser le Crazyradio. Ce dernier est une clé USB qui intègre le Nordic Semiconductor nRF24LU1+. La fréquence de communication est la bande 2.4GHz ISM. Le nRF24LU1+ est programmable par USB et le débit peut atteindre 2Mbs.

2.4 Firmware partie logicielle

La partie logicielle du Crazyflie est basée sur FreeRTOS, un système d'exploitation temps-réel open source (section 2.4.1). Le code source est modifiable sur les deux microcontrôleurs (STM32F405 et nRF51822) par contre seulement la partie logicielle sur le microcontrôleur principal a été modifiée pour les besoins de l'expérimentation. Avant de relater les modifications et autres ajouts que nous avons opérés, nous allons présenter le système FreeRTOS.

2.4.1 Le système d'exploitation temps réel FreeRTOS

FreeRTOS est un projet mature (première version publiée en 2003), stable et très largement employé, notamment dans l'industrie. Il a été intégré sur plus de 30 architectures de microcontrôleurs différentes, c'est-à-dire quasiment toutes les architectures présentes sur le marché ; ce qui en fait le système le plus portable. Le code source du noyau en lui-même se compose uniquement de quatre fichiers source en langage C. Le reste du code source du projet se compose des portages (couches d'abstraction) pour les nombreuses plates-formes matérielles et compilateurs supportés, et d'exemples. Ce système reste ainsi très léger quant aux ressources matérielles exigées (généralement moins de 10 Ko de code programme) ce qui le rend parfaitement adapté aux systèmes embarqués les plus limités. Le noyau FreeRTOS fournit des fonctionnalités avancées telles que : - le multi-tâche, préemptif par défaut, coopératif en option, via un ordonnanceur gérant les priorités - des structures de données de base (listes, files) ; - des mécanismes de communication entre tâches ("Inter Process Communication") : sémaphores, mutex ; - des timers logiciels ; - et une gestion optionnelle de l'allocation de mémoire ("heap") : trois mécanismes différents disponibles. Ainsi, le code source du crazyflie est composé de plusieurs fichiers source .c et de headers .h structurant les différentes couches d'abstraction. La première partie (drivers et répertoires Hal) se charge de la partie hardware, alors que toutes les opérations sont réalisées à un niveau d'abstraction plus haut. Dans la section qui suit nous allons voir les principales fonctions (tâches dans FreeRTOS) qui contribuent à la stabilisation du Crazyflie [31].

2.4.2 Stabilisation du Crazyflie

La taille miniature du crazyflie présente plusieurs challenges. Par exemple l'inertie réduite du Crazyflie requière des controlleurs qui doivent réagir avec une latence très réduite. la tâche de stabilisation est supposée s'exécuter au moins 500 fois par seconde, raison pour laquelle son niveau de priorité est le plus élevé dans le firmware. Pour être le mieux efficace possible, tous les calculs nécessaires à la stabilisation du drone sont réalisés au niveau du firmware.

La stabilisation est réalisée en mesurant les états du quadrirotor et en utilisant des contrôleurs pour commander son attitude et sa vitesse angulaire. La figure Fig 2.4 ci-dessous décrit le fonctionnement de la tâche de stabilisation.

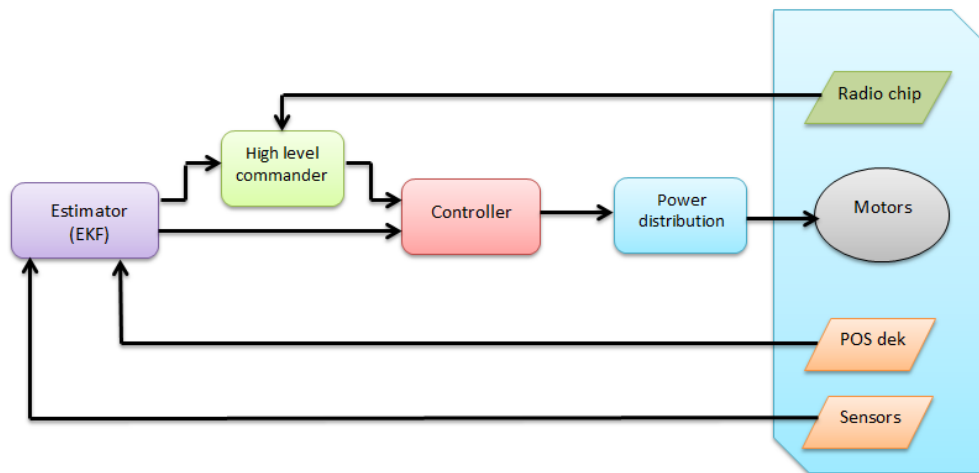


Figure 2.4: Système de stabilisation

La stabilisation est réalisée en trois phases, chacune d'elles dépend du type de la tâche voulu. Dans le cas de cette thèse, il est question de suivie de trajectoire. La première phase est la mesure et l'estimation des états du quadrotor à savoir la vitesse angulaire, l'accélération, ainsi que la position absolue. Ensuite, nous avons la phase de commande, qui détermine la position désirée. Enfin, la phase de régulation qui déplace le drone à la position désirée.

2.4.2.1 Estimation

Il existe, dans la littérature, plusieurs méthodes pour l'estimation de l'attitude et de la position. Néanmoins, certaines méthodes sont plus adaptées au cas du Crazyflie que d'autres. En effet, dans un système où nous avons besoin d'avoir des données de

mesure de manière continue, les mesures provenant des capteurs peuvent être incertaines pour des raisons multiples telles que les erreurs de mesure ou l'inconsistance. Pour cette raison un filtrage et une fusion de données est nécessaire. Le firmware open source du crazyflie comprend un filtre complémentaire pour faire uniquement l'estimation de l'orientation et un filtre de Kalman étendu (EKF Extended Kalman Filter) pour faire l'estimation de l'orientation et de la position.

Dans cette thèse nous nous intéressons à l'implémentation de lois de commandes de suivi de trajectoire, pour cette raison, seul le EKF sera présenté ici, car il fournit l'information sur la position et l'orientation.

Le filtre de Kalman est un algorithme qui permet d'estimer les états d'un système à partir d'une séquence de mesures obtenue grâce à un ou plusieurs capteurs. Le modèle dynamique du système résultant du filtre de Kalman se note comme suit :

$$\begin{aligned}x_k &= f(x_{k-1}, u_k) + w_k \\y_k &= h(x_k) + v_k\end{aligned}$$

où x_k , u_k , et k représentent, l'état du système, la commande et la valeur discrète du temps respectivement. Ici, w_k et v_k représentent les bruits du système et d'observateur qui sont supposés d'être des bruits gaussiens multivariés à moyens nuls, Q_k et R_k sont leurs covariances respectivement. Les fonctions f et h sont utilisées d'une manière indirecte pour calculer la prédiction de l'état et de la mesure, en calculant les matrices de dérivées partielles (le jacobien) correspondantes.

Pour le filtre de Kalman d'origine, les hypothèses sont telles que le système doit être linéaire et les bruits Gaussiens. Mais pour les systèmes non linéaires tels que les CF, le filtre de Kalman d'origine ne peut pas s'appliquer directement. Pour pallier ce problème, l'utilisation de l'EKF est recommandée, ce dernier est une variante du filtrage de Kalman qui peut traiter ce type de système. L'EKF effectue un développement de Taylor du premier ordre pour obtenir une approximation linéaire du système. Plus de détails sont fournis dans [33]

L'algorithme du filtre se déroule selon les étapes récurrentes suivantes :

- La prédiction : à un moment donné, l'état estimé actuel est connu, ainsi que le modèle dynamique. À partir de là, l'état suivant est prédit et devient le nouvel état estimé, dont les équations sont données comme suit :

$$\hat{x}_k = f(\hat{x}_{k-1}, u_k) \quad \text{prédiction de l'état estimé } x$$

$$P_k = F_k P_{k-1} F_k^T + Q_k \quad \text{prédiction de la covariance estimé}$$

- La correction (ou mise à jour) : l'état estimé doit être corrigé pour qu'il corresponde à l'état réel. Les mesures sont ensuite utilisées pour mettre à jour l'état estimé. Comme l'estimation accumule des erreurs provenant de différentes sources, l'erreur cumulée est définie par une matrice de covariance modifiée à chaque étape de l'algorithme. Les équations pour faire la mis à jour sont données comme suit :

$$\hat{x}_k = \hat{x}_{k-1} + K_k \tilde{y}_k \quad \text{mis à jour de l'estimation d'état}$$

$$P_k = (I - K_k H_k) P_k \quad \text{mis à jour de l'estimation de la covariance}$$

où

$$\tilde{y}_k = z_k - h(\hat{x}_k) \quad \text{l'innovation ou mesure résiduelle}$$

$$K_k = \frac{P_k H_k^T}{H_k P_k H_k^T + R_k} \quad \text{le gain de Kalman presque optimal}$$

où les matrices de transition d'état et d'observation sont définies comme étant les jacobiens suivants :

$$F_k = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}, u_k}$$

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}}$$

2.4.2.2 Commande

Le rôle de la commande consiste à fournir l'attitude et/ou la position à atteindre, et cela dépend du fait que nous pilotions manuellement les CF à l'aide du joystick ou que nous le pilotions via un pilotage automatique. Dans le premier cas, les consignes de commande du joystick sont envoyés par radio aux CF à partir du PC client et le paquet de données reçu est

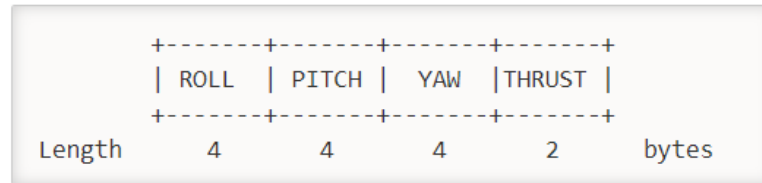


Figure 2.5: trame de "commander"

Dans le cas du pilotage automatique, les points de consigne de commandement (x , y , z et ψ) sont envoyés aux CF par radio en utilisant des scripts python. Les points de consigne sont ensuite utilisés dans un contrôleur de haut niveau (planificateur) implémenté dans le firmware pour générer une trajectoire lisse en utilisant des polynômes d'ordre 5 ou 7.

Afin de faciliter la tâche de commande automatique nous avons créé une tâche dans le firmware qui permet de lancer des séquences de pilotage automatique, sans avoir besoin de recevoir des commandes venant d'un PC client.

2.4.2.3 Régulation

Une fois que la position courante et désirée sont connues, le drone doit se déplacer de manière appropriée. Le Crazyflie comprend deux algorithmes de régulation différents. Le premier est un PID qui calcule une la valeur de l'erreur de la différence entre le setpoint désiré et une variable mesurée à chaque boucle du contrôle. Le deuxième algorithme de régulation est celui de Mellinger. Il opère en deux phases : - Un algorithme de contrôle qui peut suivre des trajectoires agressives nécessitant des accélérations larges. - Un générateur de trajectoire qui permet la génération de trajectoire à travers une série de positions et des angles lacet, tout en garantissant un passage sûr à travers des couloirs bien précis et en satisfaisant des contraintes sur les vitesses et les accélérations.

Par ailleurs, nous avons implémenté d'autres algorithmes de régulation à savoir le backstepping et le backstepping adaptatif dont les détails du design du contrôleur sont donnés dans le chapitre 3.

2.5 Crazy RealTime Protocol (CRTP)

Le Crazyflie communique avec le Crazyradio en utilisant le Crazy RealTime Protocol (CRTP). Une trame CRTP contient un header (entête) d'un byte et 29 bytes de

charge utile. Le header contient trois champs : i) le port qui spécifie quelle application la trame est destinée, ii) le lien, iii) le canal qui spécifie la fonctionnalité.

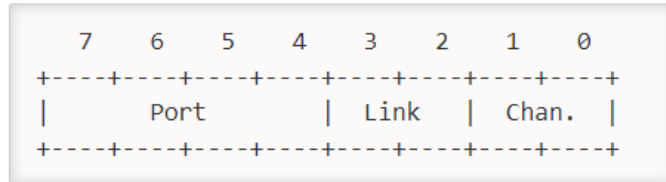


Figure 2.6: Forme de CRTP header

Le CRTP est mis en œuvre en 3 couches de manière ISO / OSI:

- CRTP link : est responsable du transfert des paquets entre le Crazyie et l'hôte. Il gèrera généralement la taille des paquets et la détection des erreurs.
- CRTP Packet : la gestion envoie le paquet au sous-système approprié dans Crazyflie et dans le logiciel de contrôle hôte.
- Ports (Application) : représente les sous-systèmes qui envoient et reçoivent des messages.

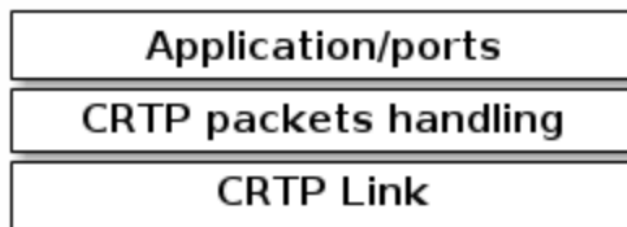


Figure 2.7: Couche du protocole CRTP

2.6 PC client

Bitcrase distribue gratuitement une librairie open-source développée en Python qui fournit des appels simple au Crazyradio qui peuvent être envoyé au Crazyflie. La librairie PC client, Fig 2.8 permet aussi de définir des callbacks pour les données provenant du quadrotor [34].

2.7 Le dispositif Loco Positioning System

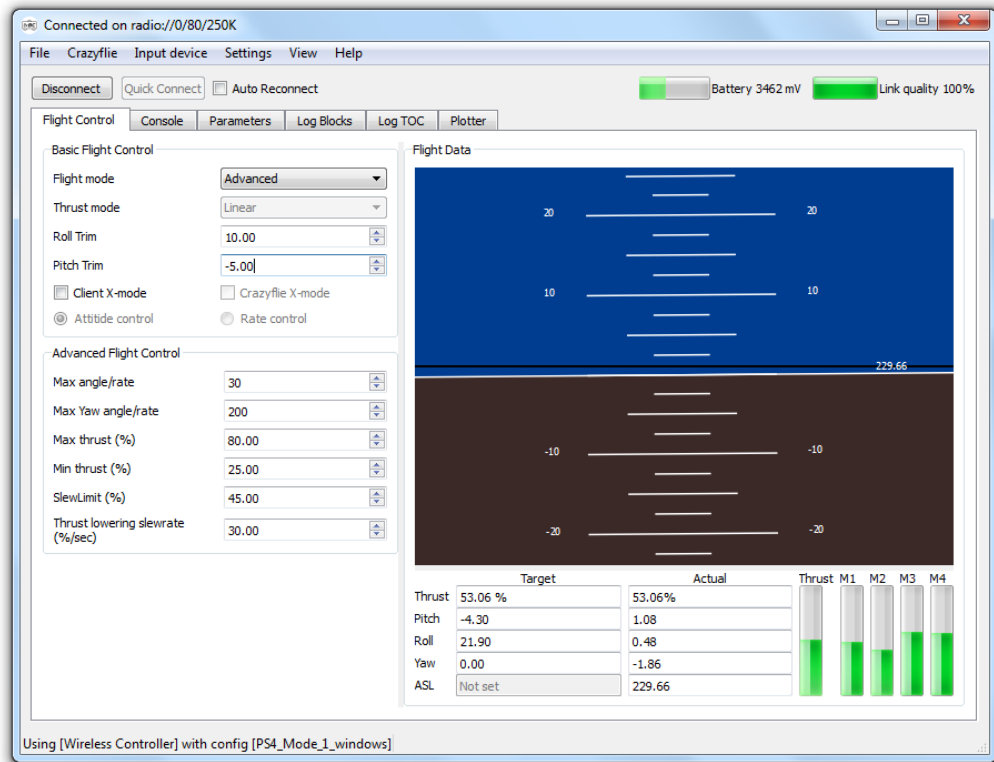


Figure 2.8: Client crazyflie

2.7 Le dispositif Loco Positioning System

Le système de positionnement que nous utilisons, le Loco Positioning System (LPS) (normalement accompagné du Crazyflie 2.0), est un système de localisation en intérieur développé par Bitcraze. La solution consiste à utiliser un système externe pouvant fournir des informations sur la position actuelle. Le système GPS peut être utilisé à l'extérieur, mais à l'intérieur, les options sont limitées et souvent complexes ou coûteuses. C'est pourquoi le système Loco Positioning a été développé.

2.7.1 Le système LPS

Le système LPS est un système de positionnement local utilisé pour trouver la position 3D absolue des objets dans l'espace. Il est similaire à un système GPS miniature. La base du système est un ensemble d'ancres (ou « balises ») Fig 2.10 qui sont positionnées dans la pièce (à comparer aux satellites dans le GPS), elles servent de référence. L'autre partie du système est constituée d'un ou de plusieurs Tags Fig 2.9 (à comparer au récepteur GPS) qui sont fixés aux objets à positionner.

2.7 Le dispositif Loco Positioning System

En envoyant de courts messages radio haute fréquence entre les ancres et les Tags, le système mesure la distance entre chaque ancre et les Tags et calcule la position des balises à partir de ces informations. Toutes les informations nécessaires pour calculer la position sont disponibles dans le Tag qui permet l'estimation de la position sur le drone même, contrairement à beaucoup d'autres systèmes de positionnement où la position est calculée dans un ordinateur externe et envoyée au Crazyflie. En ajoutant la connaissance de sa position à un Crazyflie 2.0, il est capable de voler de manière autonome sans contrôle manuel. Cela ouvre un éventail de cas d'utilisation et d'applications important [35].



Figure 2.9: LPS deck

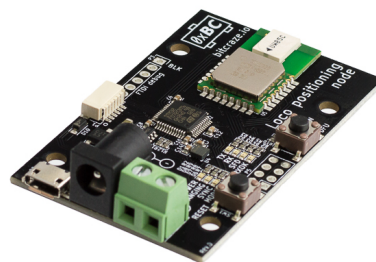


Figure 2.10: LPS node

2.7.2 Mode de positionnement

Le Loco Positioning System intègre deux modes de positionnement différents : Two Way Ranging (TWR) et Time Difference of Arrival (TDoA). En mode TWR, le Tag ping les ancres séquentiellement, ce qui lui permet de mesurer la distance entre le Tag

et les ancrés. En utilisant cette information, un minimum théorique de 4 ancrés est nécessaire pour calculer la position 3D d'une étiquette, mais un nombre plus réaliste est 6 pour ajouter de la redondance et de la précision. Un avantage de ce mode est qu'il reste précis même lorsque le Tag quitte l'espace délimité par les ancrés. Dans ce mode, il est également très facile d'ajouter plus d'ancrés pour étendre la portée du système, par exemple pour couvrir plusieurs pièces. Le principal inconvénient est que la balise communique activement avec les ancrés, ce qui signifie que l'ajout de balises utilisera plus de bande passante radio et nécessitera que chaque Tag partage le système de balises. TWR a été utilisé avec succès pour un Crazyflie. Faire voler plusieurs Crazyflies, ce mode est possible en utilisant le mode expérimental TDMA qui a été testé pour jusqu'à 4 Crazyflies, mais avec des performances réduites. En mode TDoA, le système d'ancrés envoie en continu des paquets de synchronisation. Un Tag écoutant ces paquets peut calculer la distance relative à deux ancrés en mesurant la différence de temps d'arrivée des paquets. À partir des informations du TDoA, il est possible de calculer la position 3D dans l'espace. Le grand avantage de ce mode est que le Tag est seulement passivement à l'écoute, donc les nouveaux Tags n'ajoutent aucune charge au système, ce qui permet de positionner n'importe quel nombre de tags ou Crazyflies. Cela en fait un mode parfait pour l'essaimage. Un inconvénient est que TDoA est plus sensible au placement de l'ancrage, idéalement le Tag doit toujours être à l'intérieur, ou très proche, de l'espace délimité par le système d'ancrage. Cela signifie que TDoA fonctionne mieux avec 8 ancrés placés dans les coins de l'espace de vol. De plus, puisque le système d'ancrés est synchronisé, il n'est pas possible (du moins pas aisément) d'utiliser plus de 8 ancrés pour étendre la portée du système [31].

2.7.3 Performance

Le système Loco Positioning est basé sur la puce Decawave DWM1000 et a une précision théorique de 10 cm. Les performances de positionnement 3D dépendent de la configuration du système (positionnement des ancrés, mode de positionnement) et de l'environnement. Le système Loco Positioning a été principalement conçu pour une utilisation en intérieur. L'interface PC client permet le réglage et le monitoring du système LPS comme illustré dans la Fig 2.11 :

Pour de meilleurs résultats les ancrés doivent être placés d'une manière à ce qu'il ne soient pas dans le plan. La Fig 2.12 illustre une architecture d'installation des ancrés proposé par Bitcraze :

2.7 Le dispositif Loco Positioning System

2.7.4 Two Way Ranging (TWR)

Cette technique est issue de la technologie UWB (Ultra Wide Band) qui est l'une des technologies de communication sans fil les plus compétitives pour la localisation intérieure, notamment pour ses caractéristiques très intéressantes en ce qui

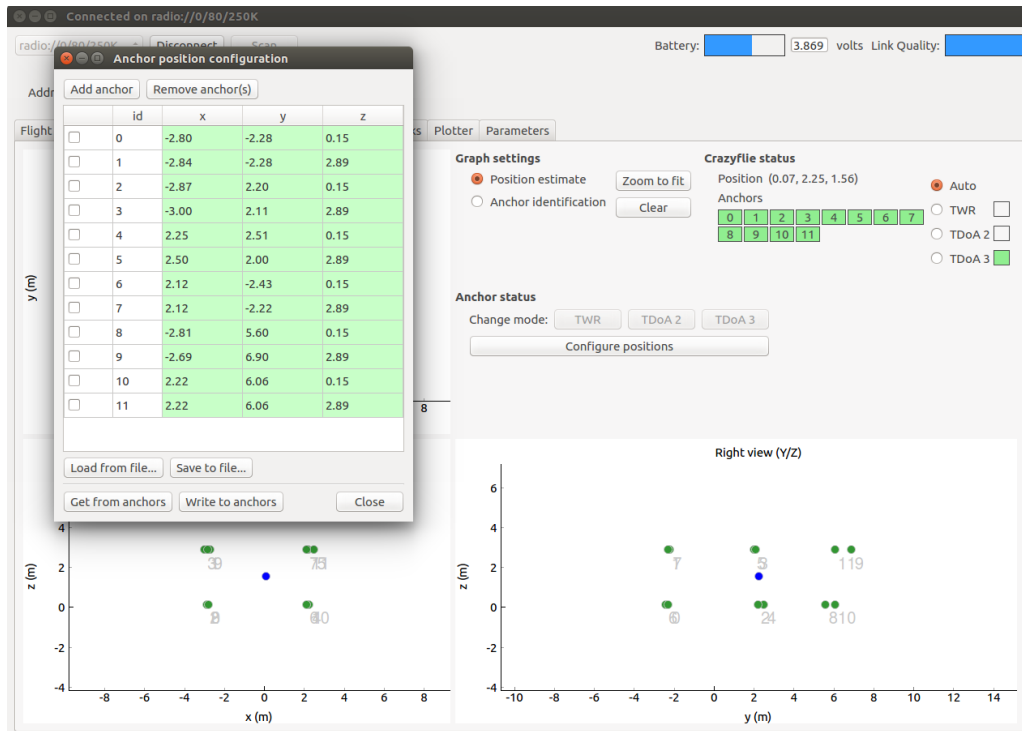


Figure 2.11: LPS Tab

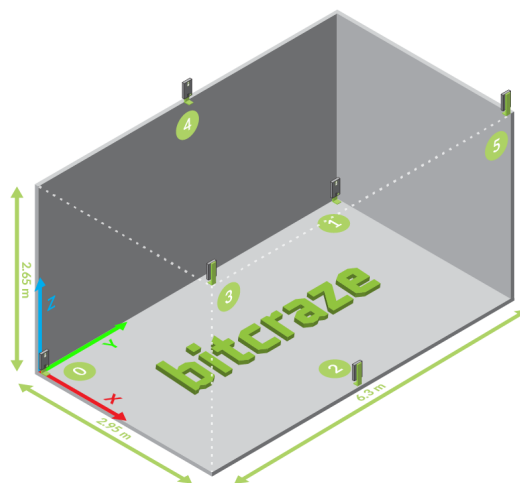


Figure 2.12: LPS Tab

concerne la complexité de l'utilisation, le coût de la réalisation, la couverture et la précision. Cette technologie permet la transmission de données numériques sur une largeur de bande de 500 MHz ou plus avec une faible puissance. À la différence des systèmes RFID classiques, qui fonctionnent sur des bandes étroites, la technologie UWB émet simultanément des signaux sur plusieurs bandes de fréquences qui s'étalent de 3.1 à 10.6 GHz. Une façon d'obtenir un tel contenu spectral large est de générer des impulsions de très courte durée, inférieure à la nanoseconde. Cette forte résolution temporelle permet d'effectuer des mesures de temps de propagation d'un signal relativement précises entre un émetteur et un récepteur. Elle peut également être utilisée à proximité d'autres signaux RF sans provoquer d'interférences en raison des différences dans les types de signaux et du spectre utilisés [32]. De plus, le système UWB est moins sensible aux phénomènes de réflexion, réfraction ou diffraction d'onde dans un environnement multi-trajet, contrairement aux systèmes à bande étroite, ce qui réduit les erreurs lors des calculs des temps de propagation. Les caractéristiques techniques du système UWB semblent bien adaptées pour des applications de localisation. Le positionnement d'un noeud mobile avec cette technologie repose principalement sur les informations provenant des signaux radio circulant entre l'émetteur (celui qui veut se localiser) et un certain nombre de points de référence.

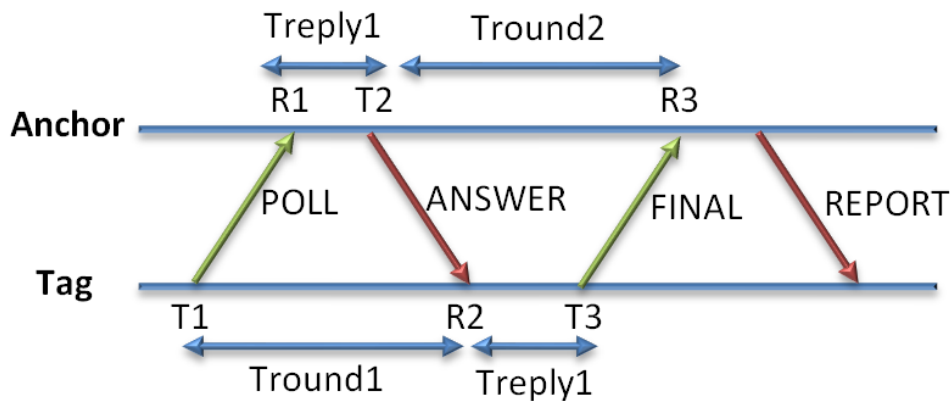


Figure 2.13: Protocole Two Way Ranging

Selon le protocole TWR, les échanges de messages entre le Tag et l'ancrer, en cinq étapes : 1. Le Tag envoie un message de type Poll à l'ancrer et note le temps d'envoi T1. Il se met ensuite à l'écoute d'un message Anchor Response de la part de l'Anchor. Si aucune réponse n'arrive au Tag avant l'écoulement du temps alloué, l'échange courant sera interrompu et il envoie un nouveau message Poll. 2. Lorsque

l'ancer reçoit un message de type Poll qui lui est adressé, il note le temps de réception R1 et envoie un message de type ancer Response au Tag et note le temps d'envoi T2. 3. Lorsque le Tag reçoit la réponse de la part de l'ancer, il note à son tour le temps d'arrivée du message R2, envoie un message de type Final Response et note le temps d'envoi T3. 4. Quand l'ancer reçoit le message final de la part du Tag, il note le temps de réception R3. 5. À ce stade, nous avons tout le nécessaire pour calculer le temps d'aller simple ToF (Time of Flight) après avoir sauvegardé tous les timestamps de l'échange. L'ancer envoie donc son dernier message du type Report en ajoutant dans le champ de données du paquet les timestamps R1, T2 et R3 afin de permettre au Tag de calculer la distance. 6. Une fois que le Tag a reçu le dernier message de l'ancer qui contient les timestamps, il calcule le temps ToF selon la formule suivante

$$t_f = \frac{T_{round_1} \times T_{round_2} - T_{reply_1} \times T_{reply_2}}{T_{round_1} + T_{round_2} + T_{reply_1} + T_{reply_2}} \quad (2.1)$$

2.8 Conclusion

Dans ce chapitre nous avons présenté la plateforme open source crazyflie 2.0 qui offre des avantages très intéressants du point de vue architecture mécanique, architecture électronique et logicielle. L'architecture électronique est basée sur deux microcontrôleurs le premier est un ARM Cortex-M4 doté d'un processeur STM32F405 qui support le firmware et un deuxième microcontrôleur ARM CortexM0 qui permet la communication RF. Le firmware est basé sur l'open source OS (FreeRTOS) ce qui permet de faciliter la gestion des taches en temps réel. Ainsi l'implémentation des algorithmes de commande et d'estimation devient plus simple. De plus, le crazyflie est doté d'un système de positionnement embarqué. Dans le chapitre 4 nous présenterons les résultats d'implémentation des algorithmes de commandes nonlinéaires sur la plateforme crazyfile.

Chapitre 3

Commande adaptative d'un
VTOL-UAVs avec inertie
inconnue, Gyro-bias et LOE
d'actionneur

3.1 Introduction

Dans ce chapitre nous présentons l'élaboration et l'analyse d'une stratégie de commande robuste applicable aux systèmes (UAV). Le développement des stratégies de commande pour ce type de système est un problème classique. Malgré le nombre important des travaux et solutions existantes, il reste toujours un sujet de recherche actif. Cela est dû aux nombreux domaines d'application tels que les VTOL-UAV, les satellites, la robotique, etc. Plusieurs problématiques et scénarios ont été traités dans la littérature ([8–10, 22, 36–42]). Nous proposons dans cette section une commande adaptative pour le suivi de trajectoire. Nous considérons que le biais gyroscopique est inconnu, ce qui constitue une contrainte qui n'a pas été considérée dans [8], [9] et [10]. L'inertie du système est supposée aussi totalement inconnue lors de la synthèse de la commande, lorsqu'il y a la présence de LOE et de forces de perturbation et de couples constants appliquées au système. Notre stratégie de conception repose sur l'approche du Backstepping en raison de la structure et la nature de la dynamique de translation et de rotation du système. Tout d'abord, une entrée de commande intermédiaire a priori bornée pour la dynamique de translation du véhicule est conçue pour atteindre l'objectif de suivi de position. Quand on considère la commande de la dynamique de translation (position et vitesse), typiquement, seule la poussée du système est supposée comme entrée de commande disponible. La direction du vecteur de la poussée est alignée avec l'axe vertical du corps rigide (body fixed fram axis). Par conséquent la commande de la translation nécessite une sorte de commande d'attitude pour pouvoir satisfaire des objectifs de vol tels que le vol stationnaire ou le suivi de trajectoire. Pour pouvoir faire la commande en position, une étape d'extraction de la magnitude et de la direction de la poussée depuis une accélération linéaire désirée est nécessaire. Cela permet de déterminer l'entrée de poussée requise et l'orientation désirée en utilisant l'extraction d'attitude présentée dans [9]. Cette extraction dépend d'un seul vecteur qui décrit l'accélération désirée. Ensuite, le couple de contrôle est conçu pour que la dynamique de rotation, en but de ramener l'orientation à sa valeur désirée. L'algorithme d'extraction utilisé nécessite des hypothèses légères sur l'accélération désirée pour éviter la singularité. Afin de s'assurer que ces hypothèses sont satisfaites ainsi que la différentiabilité de l'entrée de contrôle de la translation intermédiaire, on spécifie une commande virtuelle à priori bornée, pour le bon suivi de l'accélération désirée. Cela est assuré par l'usage des fonctions bornées. Le mécanisme de projection est impliqué dans l'adaptation pour nous assurer que les estimations des paramètres restent dans un

domaine précis. Cela nous a permis de garantir un certain niveau de robustesse en évitant que les estimations des paramètres ne dérivent dans le temps en présence de mesures bruitées. Cela nous a permis également, d'empêcher les estimations des paramètres de s'annuler et d'éviter les problèmes de singularité dans la loi de commande développée.

Pour évaluer la robustesse du contrôleur proposé lorsque la force de perturbation n'est pas constante, des résultats de simulation sont fournis, qui considèrent un modèle des forces aérodynamiques qui s'exercent sur le système en présence d'un vent externe uniforme, ayant une vitesse constante. Les résultats présentés dans ce chapitre sont basés sur nos travaux dans [43]

3.2 Formulation du problème

3.2.1 Modèle du system

Le modèle mathématique d'un VTOL-UAV, considéré pour la conception de commandes dans cette thèse est décrit par les équations dynamiques de translation et de rotation suivantes :

$$\dot{p} = v \quad (3.1)$$

$$\dot{v} = g\hat{e}_3 - \frac{T}{m}R^T\hat{e}_3 \quad (3.2)$$

$$\dot{Q} = \frac{1}{2} \begin{bmatrix} -q^T \\ q_0 I_{3 \times 3} + S(q) \end{bmatrix} \Omega \quad (3.3)$$

$$J\dot{\Omega} = -S(\Omega)J\Omega + \Delta\tau + d \quad (3.4)$$

Où p et v représentent, respectivement, la position et la vitesse du véhicule par rapport au repère inertielle $\mathcal{I} := \{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$. La masse du véhicule est désignée par m et l'accélération gravitationnelle par g . Le scalaire T représente l'amplitude de la poussée appliquée au véhicule dans la direction e_3 du repère attaché au corps $\mathcal{B} := \{e_1, e_2, e_3\}$. La matrice R représente l'orientation du repère \mathcal{B} par rapport au repère \mathcal{I} , et Q est le quaternion unitaire associé à la matrice de rotation R . Le vecteur $\tau \in \mathbb{R}^3$ indique les couples de commande fournis par les actionneurs. La matrice $\Delta := \text{diag}(\Delta_1, \Delta_2, \Delta_3)$, avec $0 < \Delta_i \leq 1$, $i \in \{1, 2, 3\}$, représente LOE de l'actionneur. Le vecteur $\Omega \in \mathbb{R}^3$ est la vitesse angulaire du véhicule exprimée dans le

repère \mathcal{B} attaché au corps rigide . La matrice $J \in \mathbb{R}^{3 \times 3}$ dénote la matrice d'inertie constante symétrique définie positive de l'UAV par rapport au repère attaché à l'UAV et $d \in \mathbb{R}^3$ est le couple de perturbation externe.

3.2.2 Énoncé du problème

L'objectif principal est de concevoir les entrées de contrôle T et τ , de sorte que la position du véhicule $p(t)$ suit une trajectoire désirée $p_d(t)$, en supposant que la matrice d'inertie J , la matrice LOE Δ , le biais gyro b et la perturbation d sont constants et inconnus. En fait, nous supposons que la vitesse angulaire mesurée Ω_s est donnée par $\Omega_s = \Omega + b$. Nous faisons également la supposition légère suivante qui est nécessaire pour prouver nos résultats techniques:

Hypothèse 3.1. La seconde, troisième et quatrième dérivées de la trajectoire désirée $p_d(t)$ sont uniformément continues. De plus, nous supposons que $\|\ddot{p}_d\| \leq \delta_p$ et $\hat{e}_3^T \ddot{p}_d < \delta_{pz} < g$ où δ_p et δ_{pz} sont des constantes positives.

3.3 Synthèse du contrôleur

3.3.1 Synthèse de la commande virtuel de translation

Définissons les signaux d'erreur de position et de vitesse comme suit :

$$\tilde{p} = p - p_d, \quad (3.5)$$

$$\tilde{v} = v - \dot{p}_d. \quad (3.6)$$

Nous utilisant la vitesse du système v comme une commande virtuel pour cela nous devons utiliser la poussée u_t et l'attitude du système R comme entrées de commandes virtuelles pour la dynamique de translation (3.1) et (3.2). Par conséquent, nous définissons

$$\mu = g\hat{e}_3 - \frac{T}{m}R^T\hat{e}_3 \quad (3.7)$$

qui représente l'accélération du système due à la gravité et la poussée du système exprimée dans le repère inertiel, où $R = R(q_0, q)$ tel que défini par (1.7) est une rotation associée au quaternion unitaire Q , (1.7). La dynamique de translation

(3.2) devient

$$\dot{v} = \tilde{\mu} + \mu_d, \quad (3.8)$$

où

$$\tilde{\mu} = \mu - \mu_d \quad (3.9)$$

avec μ_d étant l'entrée de contrôle virtuel pour le système double intégrateur suivant:

$$\begin{aligned} \dot{\tilde{p}} &= \tilde{v} \\ \dot{\tilde{v}} &= \mu_d - \ddot{p}_d + \tilde{\mu} \end{aligned} \quad (3.10)$$

Notez que

$$\mu_d := g\hat{e}_3 - u_t R_d^T \hat{e}_3, \quad (3.11)$$

où $R_d(Q_d)$ est la rotation associée au quaternion unitaire désiré Q_d , et $u_t := \frac{T}{m}$.

Nous proposons la loi de commande intermédiaire *a priori* bornée suivante qui stabilise (3.10), quand $\tilde{\mu}$ est mis à zéro :

$$\mu_d = \ddot{p}_d - k_p \Gamma_v^{-1} \tanh(\tilde{p}) - k_v \tanh(\tilde{v}), \quad (3.12)$$

où $k_p, k_v > 0$, $\Gamma_v = \Gamma_v^T > 0$ et $\tanh(u)$, pour $u = [u_1 \ u_2 \ u_3]^T \in \mathbb{R}^3$, est donné par

$$\tanh(u) = [\tanh(u_1), \tanh(u_2), \tanh(u_3)]^T. \quad (3.13)$$

Notez que, en vertu de l'Hypothèse 3.1 et la propriété $0 \leq \tanh(\cdot) < 1$, il est clair que μ_d satisfait

$$\|\mu_d\| < \bar{\mu}_d = k_p \|\Gamma_v^{-1}\| + \delta_p + k_v \quad (3.14)$$

par conséquent, sa troisième composante est délimitée comme suit:

$$|\hat{e}_3^T \mu_d| < \bar{\mu}_{d3} = k_p \|\hat{e}_3^T \Gamma_v^{-1}\| + \delta_{pz} + k_v \quad (3.15)$$

3.3.2 Extraction de l'attitude et de la poussée désirées

Pour utiliser la vitesse du système v comme un contrôle virtuel, nous devons utiliser l'attitude du système R et la poussée u_t comme entrées virtuelles de la dynamique de translation (3.1) et (3.2).

L'utilisation de μ en tant que contrôle virtuel nécessite l'extraction de la poussée

u_t et l'attitude désirée (ou l'orientation désirée) $Q_d = (q_0^d, q_d)$ depuis μ_d . Autrement dit, nous avons besoin d'une transformation $\chi_{u_t} : \mathbb{R}^3 \rightarrow \mathbb{R}$ et $\chi_Q : \mathbb{R}^3 \rightarrow \mathbb{S}^3$, de sorte que les signaux $u_t = \chi_{u_t}(\mu_d)$ et $(q_0^d, q_d) = \chi_Q(\mu_d) = (\chi_{q_0}(\mu_d), \chi_q(\mu_d))$ satisfont (3.11). Le choix de χ_{u_t} et χ_Q est détaillé ci-dessous.

En utilisant l'accélération virtuelle μ_d donnée dans (3.12), nous extrayons l'attitude et la poussée désirées en utilisant l'algorithme fourni dans [8] et [9]. Nous appliquons les résultats de l'algorithme d'extraction du quaternion, détaillé dans la Section 1.4 à notre cas particulier. Rappelons que (3.7), (4.10) et (3.11), en supposons que μ_d soit l'entrée de commande virtuelle qui atteint l'objectif de suivi de position pour la dynamique de translation. Nous devons extraire la poussée u_t et l'orientation désirée R_d ou Q_d pour un μ_d donné satisfaisant (3.11). Une solution pour $u_t = \chi_{u_t}$ et $Q_d = (\chi_{q_0}, \chi_q)$ qui satisfait (3.11) est donnée par le lemme suivant :

Lemme 3.4. [9] Étant donné μ_d qui est supposée $\mu_d \notin L$

$$L = \{\mu_d \in \mathbb{R}^3; \mu_d = \text{col}[0, 0, \mu_{d3}]; \mu_{d3} \in [g, \infty)\} \quad (3.16)$$

une solution pour les transformations $u_t = \chi_{u_t}$ et χ_Q qui fait l'extraction de la poussée du système $u_t = \chi_{u_t}$ et l'attitude désirée $Q_d = (\chi_{q_0}, \chi_q)$ et qui satisfait (3.11) est donnée par

$$\chi_{u_t} = \|\mu_d - g\hat{e}_3\| \quad (3.17)$$

$$\chi_{q_0^d} = \sqrt{\frac{1}{2} + \frac{g - \mu_{d3}}{2u_t}} \quad (3.18)$$

$$\chi_{q_d} = \frac{1}{2u_t q_0^d} (\mu_{d2} \quad -\mu_{d1} \quad 0)^T \quad (3.19)$$

Preuve. Notez que (3.11) peut aussi être écrite comme $R_d^T \hat{e}_3 = u_t^{-1}(g\hat{e}_3 - \mu_d)$. En choisissant l'entrée de la poussée du système (3.17), il est clair que cela ressemble au problème décrit par Lemme 1.1 avec $v = \hat{z}$ et $u = (g\hat{e}_3 - \mu_d)/\|g\hat{e}_3 - \mu_d\|$ et $\|u\| = \|v\| = 1$. L'application de ces valeurs de u et v à (1.18) et (1.19) donne les expressions données par (3.18) et (3.19).

Remarque 3.2. La singularité (3.16) correspond au cas où l'accélération verticale du véhicule est supérieure ou égale à l'accélération due à la gravité. Notez que la singularité ne s'applique qu'à l'attitude désirée Ω_d , et l'attitude réelle du système peut prendre n'importe quelle valeur sans rencontrer de singularité. Comme la

singularité correspond à un mode de fonctionnement indésirable de l'aéronef, éviter cette singularité ne limite pas de manière significative le mode de fonctionnement normal du système. Il est possible d'éviter cette singularité en utilisant une loi bornée pour l'accélération virtuelle désirée μ_d . C'est le but principal de l'utilisation de la fonction tanh décrite par (3.13) et du mécanisme de projection décrit dans la section 1.6.2 .

$$u_t = \|\mu_d - g\hat{e}_3\| \quad (3.20)$$

$$q_0^d = \sqrt{\frac{1}{2} + \frac{g - \mu_{d3}}{2u_t}} \quad (3.21)$$

$$q_d = \frac{1}{2u_t q_0^d} (\mu_{d2} \quad -\mu_{d1} \quad 0)^T \quad (3.22)$$

à condition que

$$\mu_d \neq (0 \quad 0 \quad \kappa), \quad \text{for } \kappa \geq g \quad (3.23)$$

La condition sur μ_d ci-dessus (3.23) signifie que le véhicule ne peut pas s'engager dans une décente vertical avec une accélération supérieure ou égale à l'accélération gravitationnelle. De plus, la bornitude *a priori* de μ_d permet d'éviter les singularités qui peuvent apparaître dans (3.21) et (3.22). Puisque la borne supérieure de μ_{d3} , satisfait $\bar{\mu}_{d3} < g$, l'algorithme d'extraction est libre de toute singularité , et $0 < g - \bar{\mu}_{d3} < u_t < g + \bar{\mu}_{d3}$.

La vitesse angulaire désirée peut être obtenue en utilisant la methode donnée par [9], on peut exprimer Ω_d en termes de l'accélération désirée et de sa dérivée comme suit

$$\Omega_d = 2 \left(\begin{array}{c} -q_d^T \\ q_0^d I_{3 \times 3} + S(q_d) \end{array} \right)^T \dot{Q}_d \quad (3.24)$$

Au cours du processus de conception de la commande, il est nécessaire d'obtenir une expression pour la vitesse angulaire Ω_d , qui peut être trouvée en utilisant (21) et en différenciant l'expression pour l'attitude souhaitée Ω_d . La valeur de Ω_d est obtenue par une simple différenciation de l'attitude extraite donnée par (3.18) et (3.19). Cette procédure donne finalement l'expression suivante:

D'après [9], on peut exprimer Ω_d en termes de l'accélération désirée et de sa dérivée comme suit

$$\Omega_d = M(\mu_d)\dot{\mu}_d \quad (3.25)$$

Où

$$M(\mu_d) = \frac{1}{\|\eta\|^2\xi} \cdot \begin{bmatrix} -\mu_{d1}\mu_{d2} & -\mu_{d2}^2 + \|\eta\|\xi & \mu_{d2}\xi \\ \mu_{d1}^2 - \|\eta\|\xi & \mu_{d1}\mu_{d2} & -\mu_{d1}\xi \\ \mu_{d2}\|\eta\| & -\mu_{d1}\|\eta\| & 0 \end{bmatrix} \quad (3.26)$$

et

$$\dot{\mu}_d = p_d^{(3)} + \Psi_5 + \Psi_2 \tanh(\tilde{v}) + \Psi_3 \tilde{q} + \Psi_4 \tilde{v}, \quad (3.27)$$

avec $\eta := [\eta_1 \ \eta_2 \ \eta_3]^T := \mu_d - g\hat{e}_3$ and $\xi := \|\eta\| + g - \mu_{d3}$.

En utilisant le fait que $\tilde{Q} = Q_d^{-1} \odot Q = (\tilde{q}_0, \tilde{q})$ soit le quaternion associé à l'erreur de rotation RR_d^T , l'expression de $\tilde{\mu}$ (utilisée dans (3.27)) est obtenue comme suit

$$\begin{aligned} \tilde{\mu} &:= \mu - \mu_d = u_t(R_d^T - R^T)\hat{e}_3 = u_t R^T(RR_d^T - I)\hat{e}_3 \\ &= 2u_t R^T(-S(\tilde{q}) + \tilde{q}_0 I)S(\hat{e}_3)\tilde{q} = 2u_t R^T S(\tilde{q})\tilde{q} := \Psi_1^T \tilde{q} \end{aligned} \quad (3.28)$$

Où $\bar{q} = S(\hat{e}_3)\tilde{q} + \tilde{q}_0\hat{e}_3$ et

$$\begin{aligned} \Psi_1 &= -2u_t S(\bar{q})R \\ \Psi_2 &= k_v^2 D_t(\tilde{v}) \\ \Psi_3 &= -k_v D_t(\tilde{v})\Psi_1^T \\ \Psi_4 &= -k_p \Gamma_v^{-1} D_t(\tilde{p}) \\ \Psi_5 &= k_v k_p D_t(\tilde{v})\Gamma_v^{-1} \tanh(\tilde{p}) \end{aligned} \quad (3.29)$$

Où $D_t(u)$ est défini, quelque soit $u = [u_1 \ u_2 \ u_3]^T \in \mathbb{R}^3$, comme suit

$$D_t(u) = \text{diag}(\text{sech}^2(u_1), \text{sech}^2(u_2), \text{sech}^2(u_3)). \quad (3.30)$$

3.3.3 Synthèse du contrôleur rotationnel

La dynamique des erreurs est donnée par

$$\dot{\tilde{p}} = \tilde{v} \quad (3.31)$$

$$\dot{\tilde{v}} = \mu_d + \tilde{\mu} - \ddot{p}_d \quad (3.32)$$

$$\dot{\tilde{Q}} = \frac{1}{2} \begin{bmatrix} \tilde{q}^T(\Omega_d - \tilde{\Omega} + \tilde{b} - \bar{\Omega}) \\ \tilde{q}_0(\bar{\Omega} + \tilde{\Omega} - \tilde{b} - \Omega_d) + S(\tilde{q})(\bar{\Omega} + \tilde{\Omega} - \tilde{b} + \Omega_d) \end{bmatrix} \quad (3.33)$$

$$J\dot{\tilde{\Omega}} = -S(\Omega)J\Omega - J(\dot{\tilde{\Omega}} + \dot{\hat{b}}) + \Delta\tau + d \quad (3.34)$$

Où μ_d est donnée par (3.12), $\tilde{\Omega} = \Omega_s - \hat{b} - \bar{\Omega}$ avec $\bar{\Omega}$ étant la loi de commande virtuelle à concevoir plus tard. Le vecteur $\tilde{b} = b - \hat{b}$ est l'erreur d'estimation du biais du gyro, où \hat{b} est l'estimation du biais gyro b .

Pour atteindre notre objectif, nous proposons le couple de contrôle suivant pour la dynamique de rotation

$$\tau = \hat{\Delta}^{-1} \left(-\gamma_q \tilde{q} - F(\Omega_s, \varpi + \hat{b}, f_{\hat{\Omega}}) \hat{\Theta} - \hat{d} - K_{\Omega} \tilde{\Omega} \right) \quad (3.35)$$

avec

$$\bar{\Omega} = \frac{-1}{\gamma_q} \Psi_1 \Gamma_v \tilde{v} + M(\mu_d)(p_d^{(3)} + \Psi_4 \tilde{v} + \Psi_5) - K_q \tilde{q} \quad (3.36)$$

Où $\gamma_q > 0$, $K_{\Omega} = K_{\Omega}^T > 0$, $K_q = K_q^T > 0$. La matrice $f_{\hat{\Omega}}$, ainsi que les vecteurs ϖ et $\hat{\Omega}$ sont définis dans (A.7), Annexe A.1. Le mapping $F : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^{3 \times 36}$ tel que $(x, y, z) \mapsto F(x, y, z)$ est défini comme

$$F(x, y, z) = [-(S(x)M_1(x) + M_1(y)) \quad M_2(x) \quad -I_{3 \times 3} \quad -M_3(z)], \quad (3.37)$$

Où les mappings $M_1 : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 6}$, $M_2 : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 9}$ et $M_3 : \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^{3 \times 18}$ sont définis dans (A.12) et (A.14), Annexe A.1.

Les estimations $\hat{\Theta} \in \mathbb{R}^{36}$, $\hat{b} \in \mathbb{R}^3$, $\hat{d} \in \mathbb{R}^3$ et $\hat{\Delta} \in \mathbb{R}^3$ sont mises à jour par les lois d'adaptation suivantes :

$$\dot{\hat{\Theta}} = \Gamma_{\theta} \text{Proj}(\hat{\Theta}, -F(\Omega_s, \varpi + \hat{b}, f_{\hat{\Omega}})^T \tilde{\Omega}, \mathcal{P}_{\theta}) \quad (3.38)$$

$$\dot{\hat{b}} = \Gamma_b \text{Proj}(\hat{b}, -\gamma_q \tilde{q}, \mathcal{P}_b) \quad (3.39)$$

$$\dot{\hat{d}} = \Gamma_d \text{Proj}(\hat{d}, \tilde{\Omega}, \mathcal{P}_d) \quad (3.40)$$

$$\dot{\hat{\Delta}}_i = \gamma_i \text{Proj}(\hat{\Delta}_i, \tilde{\Omega}^T \Lambda_i \tau, \mathcal{P}_{\Delta_i}), \quad i = 1, 2, 3 \quad (3.41)$$

Où $\gamma_i > 0$ for $i = 1, 2, 3$, $\Lambda_1 = \text{diag}(1, 0, 0)$, $\Lambda_2 = \text{diag}(0, 1, 0)$, $\Lambda_3 = \text{diag}(0, 0, 1)$, $\Gamma_b = \Gamma_b^T > 0$, $\Gamma_d = \Gamma_d^T > 0$, $\Gamma_{\theta} = \Gamma_{\theta}^T > 0$. Les fonctions de projection convexes \mathcal{P}_{θ} , \mathcal{P}_b , et \mathcal{P}_{Δ} sont données par

$$\mathcal{P}_{\theta}(\hat{\Theta}) = \|\hat{\Theta}\|^2 - \Theta_m^2 \quad (3.42)$$

$$\mathcal{P}_b(\hat{b}) = \|\hat{b}\|^2 - b_m^2 \quad (3.43)$$

$$\mathcal{P}_d(\hat{d}) = \|\hat{d}\|^2 - d_m^2 \quad (3.44)$$

$$\mathcal{P}_{\Delta_i}(\hat{\Delta}_i) = (\hat{\Delta}_i - \sigma)^2 - \Delta_m \quad (3.45)$$

où Θ_m , b_m et d_m sont les bornes supérieures des normes des paramètres inconnus Θ (définis dans le (A.11), Annexe A.1), b et d , respectivement. Les paramètres Δ_m , σ et ϵ sont choisis tels que $\sigma + \sqrt{\Delta_m + \epsilon} \leq 1$ et $\sigma - \sqrt{\Delta_m + \epsilon} > 0$, qui à son tour garantit que $0 < \hat{\Delta}_i \leq 1$. Notez que la projection a été utilisée en trop $\hat{\Delta}$ pour empêcher $\hat{\Delta}$ de passer à zéro et pour que $\hat{\Delta}^{-1}$ utilisée dans la loi de commande soit non-singulier. Les autres projections ont été utilisées pour éviter que les estimations ne dérivent pas (sous l'effet de l'intégration) en présence de bruit de mesure.

3.4 Analyse de stabilité

Le schéma de la commande de suivi de trajectoire proposé peut être résumé par le théorème suivant :

Théorème 3.6. *Considérons le VTOL UAV décrit par (3.1) - (3.4), sous le couple de contrôle (3.35), la poussée de contrôle $T = mu_t$, avec u_t donnée par (3.20) et μ_d donnée par (3.12), et les lois d'adaptation (3.38) - (3.41). Supposons que la trajectoire désirée est telle que l'Hypothèse 3.1 est valide. Supposons que les gains*

3.4 Analyse de stabilité

de contrôle sont choisis tels que $\bar{\mu}_{d3} < g$ et

$$\begin{aligned}\lambda_{\min}(\Gamma_v) &> \frac{\gamma_q k_v^2 \epsilon}{\delta_{\mu_d}^2} \\ \lambda_{\min}(K_q) &> \left(\frac{2\sqrt{2}k_v \bar{u}_t}{\delta_{\mu_d}} + \frac{k_v^2}{2\epsilon} \right)\end{aligned}\quad (3.46)$$

où $\lambda_{\min}(\cdot)$ dénote la valeur propre minimale de son argument, $\epsilon > 0$, et \bar{u}_t et δ_{μ_d} sont les bornes du total de la poussée, $\delta_{\mu_d} < u_t < \bar{u}_t$ définit comme $\bar{u}_t = \bar{\mu}_d + g$ et $\delta_{\mu_d} = g - \bar{\mu}_{d3}$. Ensuite, pour toute condition initiale, l'entrée de poussée du système u_t est bornée et non nulle, de sorte que $0 < u_t < \bar{u}_t$, tous les signaux en boucle fermée sont bornés, et

$$\lim_{t \rightarrow +\infty} \left(\tilde{v}(t), \tilde{p}(t), \tilde{q}(t), \tilde{b}, (\Omega(t) - \Omega_d(t)) \right) = 0 \quad (3.47)$$

Preuve. Considérons la fonction de Lyapunov candidate suivante

$$\begin{aligned}V = & k_p \mathbb{1}^T \ln(\cosh(\tilde{p})) + \frac{1}{2} \tilde{v}^T \Gamma_v \tilde{v} + 2\gamma_q (1 - \tilde{q}_0) + \frac{1}{2} \tilde{\Omega}^T J \tilde{\Omega} + \frac{1}{2} \tilde{\Theta}^T \Gamma_{\theta}^{-1} \tilde{\Theta} \\ & + \frac{1}{2} \tilde{b}^T \Gamma_b^{-1} \tilde{b} + \frac{1}{2} \tilde{d}^T \Gamma_d^{-1} \tilde{d} + \frac{1}{2} \sum_{i=1}^3 \gamma_i^{-1} \tilde{\Delta}_i^2\end{aligned}\quad (3.48)$$

où $\mathbb{1} = [1 \ 1 \ 1]^T$, $\tilde{\Theta} = \Theta - \hat{\Theta}$, $\tilde{b} = b - \hat{b}$, $\tilde{d} = d - \hat{d}$ et $\tilde{\Delta}_i = \Delta_i - \hat{\Delta}_i$, $i = 1, 2, 3$. Le vecteur Θ est défini dans (A.11), Annexe A.1. La dérivée temporelle de (3.48), le long des trajectoires, du système en boucle fermée est donné par :

$$\begin{aligned}\dot{V} = & k_p \tilde{v}^T \tanh(\tilde{p}) + \tilde{v}^T \Gamma_v (\mu_d + \tilde{\mu} - \dot{p}_d) - \gamma_q \tilde{q}^T (\Omega_d - \tilde{\Omega} - \bar{\Omega} + \tilde{b}) - \tilde{\Theta}^T \Gamma_{\theta}^{-1} \dot{\tilde{\Theta}} \\ & + \tilde{\Omega}^T \left(F(\Omega_s, \varpi + \dot{\hat{b}}, f_{\dot{\hat{\Omega}}}) \Theta + \Delta\tau + d \right) - \tilde{b}^T \Gamma_b^{-1} \dot{\tilde{b}} - \tilde{d}^T \Gamma_d^{-1} \dot{\tilde{d}} - \sum_{i=1}^3 \gamma_i^{-1} \tilde{\Delta}_i \dot{\tilde{\Delta}}_i\end{aligned}\quad (3.49)$$

où nous avons utilisé l'expression de $J\dot{\tilde{\Omega}}$ donnée ci-dessous (les détails sont donnés dans (A.10), Annexe A.1)

$$J\dot{\tilde{\Omega}} = J \left(\dot{\Omega} - (\dot{\bar{\Omega}} + \dot{\hat{b}}) \right) = F(\Omega_s, \varpi + \dot{\hat{b}}, f_{\dot{\hat{\Omega}}}) \Theta + \Delta\tau + d \quad (3.50)$$

En utilisant les expressions de l'accélération désirée μ_d et l'orientation désirée Ω_d

donnée par (3.12) et (3.25), respectivement, \dot{V} devient

$$\begin{aligned} \dot{V} = & \tilde{v}^T \Gamma_v (-k_v \tanh(\tilde{v}) + \Psi_1^T \tilde{q}) - \gamma_q \tilde{q}^T \left(M(\mu_d)(p_d^{(3)} + \Psi_5 + \Psi_2 \tanh(\tilde{v}) + \Psi_3 \tilde{q} + \Psi_4 \tilde{v}) \right. \\ & \left. - \tilde{\Omega} - \bar{\Omega} + \tilde{b} \right) + \tilde{\Omega}^T \left(F(\Omega_s, \varpi + \dot{\tilde{b}}, f_{\tilde{\Omega}}) \Theta + \hat{\Delta} \tau + \tilde{\Delta} \tau + d \right) - \tilde{\Theta}^T \Gamma_\theta^{-1} \dot{\tilde{\Theta}} \\ & - \tilde{b}^T \Gamma_b^{-1} \dot{\tilde{b}} - \tilde{d}^T \Gamma_d^{-1} \dot{\tilde{d}} - \sum_{i=1}^3 \gamma_i^{-1} \tilde{\Delta}_i \dot{\tilde{\Delta}}_i \end{aligned} \quad (3.51)$$

où $\hat{\Delta} = \text{diag}(\hat{\Delta}_1, \hat{\Delta}_2, \hat{\Delta}_3)$ et $\tilde{\Delta} = \text{diag}(\tilde{\Delta}_1, \tilde{\Delta}_2, \tilde{\Delta}_3)$. Prenant $\bar{\Omega}$ et le couple de contrôle τ comme indiqué dans (A.1) et (3.35), on obtient

$$\begin{aligned} \dot{V} \leq & -\tilde{v}^T \Delta_v \tanh(\tilde{v}) - \gamma_q \tilde{q}^T \Delta_q \tilde{q} - \tilde{\Omega}^T K_\Omega \tilde{\Omega} - \tilde{\Theta}^T \left(\Gamma_\theta^{-1} \dot{\tilde{\Theta}} + F(\Omega_s, \varpi + \dot{\tilde{b}}, f_{\tilde{\Omega}})^T \tilde{\Omega} \right) \\ & - \tilde{b}^T \left(\Gamma_b^{-1} \dot{\tilde{b}} + \gamma_q \tilde{q} \right) - \tilde{d}^T \left(\Gamma_d^{-1} \dot{\tilde{d}} - \tilde{\Omega} \right) - \sum_{i=1}^3 \tilde{\Delta}_i \left(\gamma_i^{-1} \dot{\tilde{\Delta}}_i - \tilde{\Omega}^T \Lambda_i \tau \right) \end{aligned} \quad (3.52)$$

où

$$\begin{aligned} \Delta_v & := K_v \left(\Gamma_v - \frac{\gamma_q k_v^2 \epsilon}{\delta_{\mu_d}^2} I_{3 \times 3} \right), \\ \Delta_q & := K_q - \left(\frac{2\sqrt{2} k_v \bar{u}_t}{\delta_{\mu_d}} + \frac{k_v^2}{2\epsilon} \right) I_{3 \times 3}, \end{aligned} \quad (3.53)$$

qui sont clairement, définis positifs dans les conditions données dans (3.46). Notez que l'inégalité de Young suivante a été utilisée pour obtenir (3.52)

$$|\gamma_q \tilde{q}^T M(\mu_d) \Psi_2 \tanh(\tilde{v})| \leq \frac{\gamma_q k_v^2}{2\epsilon} \tilde{q}^T \tilde{q} + \frac{\gamma_q k_v^2 \epsilon}{\delta_{\mu_d}^2} \tilde{v}^T \tanh(\tilde{v}) \quad (3.54)$$

avec $\epsilon > 0$. Nous avons également utilisé le fait que $\|S(\bar{q})\| \leq 1$ et $u_t < \bar{u}_t = g + \bar{\mu}_d$ pour obtenir

$$\|M(\mu_d) \Psi_3\| \leq \frac{2\sqrt{2} k_v \bar{u}_t}{\delta_{\mu_d}}, \quad (3.55)$$

qui a également été utilisé pour obtenir (3.52). Notez que la norme Frobenius de la matrice $M(\mu_d)$, *i.e.*, $\|M(\mu_d)\|_F = \sqrt{2/(\|\eta\|^2 + \|\eta\|\|\eta_3\|) + 1/\|\eta\|^2}$, a été également utilisée pour obtenir les bornes dans (3.54) et (3.55). Puisque $\inf\{\|\eta\|\} = \inf\{\|\eta_3\|\} = \delta_{\mu_d}$, on a $\|M(\mu_d)\|_F \leq \sqrt{2}/\delta_{\mu_d}$.

En utilisant les lois d'adaptation (3.38) - (3.41) dans (3.52), et en utilisant les propriétés de la projection, on obtient :

$$\dot{V} \leq -\tilde{v}^T \Delta_v \tanh(\tilde{v}) - \gamma_q \tilde{q}^T \Delta_q \tilde{q} - \tilde{\Omega}^T K_\Omega \tilde{\Omega} \quad (3.56)$$

Il est clair que \dot{V} est semi-défini négative et donc que V non-croissante le long des trajectoires du système en boucle fermée. Par conséquent, toutes les variables en boucle fermée sont bornées et V converge vers une limite non négative. On peut facilement vérifier que \ddot{V} est bornée en raison de l'Hypothèse 1 et de la bornitude des variables en boucle fermée.

Par conséquent, \dot{V} est uniformément continue et donc, en vertu de Barbalat's lemme, on conclut que $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$, qui à son tour, implique que \tilde{v} , $\tilde{\Omega}$ et \tilde{q} ont tendent vers zéro car t tend vers l'infini. Par conséquent, à partir de (3.28), il est clair que $\tilde{\mu}$ tend également vers zéro. De plus, on peut facilement vérifier que $\ddot{\tilde{v}}$ est borné, et par conséquent $\lim_{t \rightarrow \infty} \dot{\tilde{v}}(t) = 0$, qui implique que $\lim_{t \rightarrow \infty} (-k_p \tanh(\tilde{p}(t)) + \tilde{\mu}(t)) = 0$, et par conséquent $\lim_{t \rightarrow \infty} \tilde{p}(t) = 0$. De plus, on peut vérifier que $\ddot{\tilde{q}}$ est borné, ce qui implique que $\lim_{t \rightarrow \infty} \dot{\tilde{q}}(t) = 0$, qui, en vue de (3.33), implique que $\lim_{t \rightarrow \infty} (\tilde{\Omega}(t) - \tilde{b}(t) - \Omega_d(t)) = 0$. Avec les résultats obtenus ci-dessus, il est clair, à partir de (A.1), que $\tilde{\Omega}$ tend vers $M(\mu_d)p_d^{(3)}$ qui n'est rien d'autre que la valeur de Ω_d quand t tend vers l'infinité. Cela implique que $\lim_{t \rightarrow \infty} \tilde{b}(t) = 0$. De la définition de $\tilde{\Omega} = \Omega + \tilde{b} - \bar{\Omega}$, il est clair, avec les résultats ci-dessus, que $\lim_{t \rightarrow \infty} (\Omega(t) - \Omega_d(t)) = 0$, ce qui complète la preuve.

3.5 Simulations

Pour tester l'efficacité du schéma de contrôle proposé, nous allons considérer le modèle réaliste du Ducted-fan VTOL-UAV suivant [9], avec un bruit de mesure :

$$\dot{p} = v \quad (3.57)$$

$$\dot{v} = \frac{1}{m} F_G - \frac{T}{m} R^T \hat{e}_3 - \frac{1}{ml} R^T S(\hat{e}_3) \tau + \frac{1}{m} F_{ext} \quad (3.58)$$

$$\dot{Q} = \frac{1}{2} \begin{bmatrix} -q^T \\ q_0 I_{3 \times 3} + S(q) \end{bmatrix} \Omega \quad (3.59)$$

$$J \dot{\Omega} = -S(\Omega) J \Omega + \Delta \tau + \varepsilon_l S(\hat{e}_3) R F_{ext} + d \quad (3.60)$$

où F_{ext} représente la perturbation aérodynamique externe, exprimée dans le repère inertiel, appliquée au centre de pression (CP) situé sur l'axe e_3 à une distance de

3.5 Simulations

ε du centre de gravité du système (CG) comme indiqué dans la Fig. 1.2. La force $F_G = mg\hat{e}_3$ est la force gravitationnelle. Le terme $\frac{1}{ml}R^T S(\hat{e}_3)\tau$ représente le couplage entre la dynamique de rotation et la dynamique de translation. Notez que les termes $\frac{1}{ml}R^T S(\hat{e}_3)\tau$, $\frac{1}{m}F_{ext}$ et $\varepsilon_l S(\hat{e}_3)R F_{ext}$ ont été volontairement omis dans le modèle simplifié (3.1) - (3.4) afin de simplifier la conception du contrôle.

L'effet de force aérodynamique, F_{ext} , appliqué au CP est l'addition de deux forces de perturbation : 1) la force de frottement F_{drag} et 2) F_{ram} la force de ram-traînée tangentielle, qui forme avec la force de poussée axiale la force résultante qui est égale et opposée requise pour changer l'impulsion du flux d'air. Le modèle de la perturbation appliquée est donné par :

$$\begin{aligned} F_{ext} &= F_{drag} + F_{ram} \\ F_{drag} &= \|v_\omega - v\| R^T C_d R (v_\omega - v) \\ F_{ram} &= \sqrt{\frac{T\rho A}{2}} R^T I_{xy} R (v_\omega - v) \end{aligned}$$

où $v_\omega = [-1 \ -1 \ 0]^T$ est une vitesse du vent constante et inconnue [9, 10]. Le paramétré $\rho = 1.2kg/m^3$ représente la densité de l'air et $A = 0.114m^2$ représente aire de la section transversale du duct, et $C_d \in \mathbb{R}$ est une matrice qui se compose de la constante aérodynamique dépendante du système exprimée dans le repère fixe du système, qui est choisi pour être $C_d = diag[0.1, 0.1, 0.05]kg/m$. $I_{xy} = diag(1, 1, 0)$. Le CP est situé sur e_3 axe au-dessus du CG à distance $\varepsilon_l = 0.1m$ représente le bras de levier des effets aérodynamiques. Les paramètres du corps rigide UAV sont les suivants: la matrice d'inertie $J = diag(0.6, 0.6, 0.3)kg \cdot m^2$, la masse $m = 5kg$, et le bras de levier de contrôle de couple $l = 0.5$. L'accélération gravitationnelle est prise comme $g = 9.81m/s^2$, et la trajectoire de référence souhaitée a été spécifiée $p_d(t) = [10 \sin(0.1t + \frac{2\pi}{3}), 15 \sin(0.1t + \frac{2\pi}{5}), 5t]^T$. La perturbation externe est choisie comme $d = [0.7 \ -0.3 \ 0.5]^T N \cdot m$. La perte d'efficacité LOE a été appliquée au système à $t = 15s$ avec $\Delta = diag(0.6, 0.5, 0.45)$. Le biais du gyro est pris comme $b = [0.07 \ 0.06 \ 0.05]^T rad/s$. Les conditions initiales ont été prises comme : $p(0) = [2 \ 1 \ 0]m$, $v(0) = [5 \ 0 \ 0.5]^T m/s$, $(q_0(0), q(0)) = (1, [0 \ 0 \ 0]^T)$, $\Omega(0) = [0 \ 0 \ 0]^T$, $\hat{\Delta}(0) = [1 \ 1 \ 1]^T$, $\hat{b}(0) = [0 \ 0 \ 0]^T$ et tous les éléments de $\hat{\Theta}(0)$ égaux à 0. Les gains du contrôleur et des adaptations ont été choisis comme suit : $k_p = 0.5$, $k_v = 0.7$, $\Gamma_v = diag(0.4, 0.4, 4)$, $\Gamma_b = diag(0.035, 0.035, 0.035)$, $\Gamma_d = diag(0.5, 0.5, 0.5)$, $\Gamma_\theta = 10^{-3}I_{36 \times 36}$, $\gamma_q = 30$, $K_q = 8I_{3 \times 3}$ et $K_\Omega = 2I_{3 \times 3}$. Les paramètres de projection

3.5 Simulations

ont été choisis comme suit : $\theta_m = 3$, $b_m = 0.6$, $d_m = 2$, $\Delta_m = 0.2$, $\sigma = 0.5$, et $\epsilon = 0.01$. Les mesures (position, vitesses linéaires et angulaires) sont supposées être corrompues avec un bruit moyen nul de variance de 10^{-3} .

Les simulations présentées dans les Figs. 3.1 –3.8 montrent l'efficacité du système de contrôle proposé pour suivre la trajectoire désirée malgré une inertie inconnue, le biais gyroscopique, les LOE et les perturbations externes agissant sur l'UAV. Les simulations effectuées sur le modèle (3.1) - (3.4), sur lesquelles nous avons basé notre plan de contrôle, sans bruit de mesures, sont données en bleu, et les simulations effectuées sur le modèle réaliste (3.57) - (3.60) avec bruit de mesure sont indiquées en rouge.

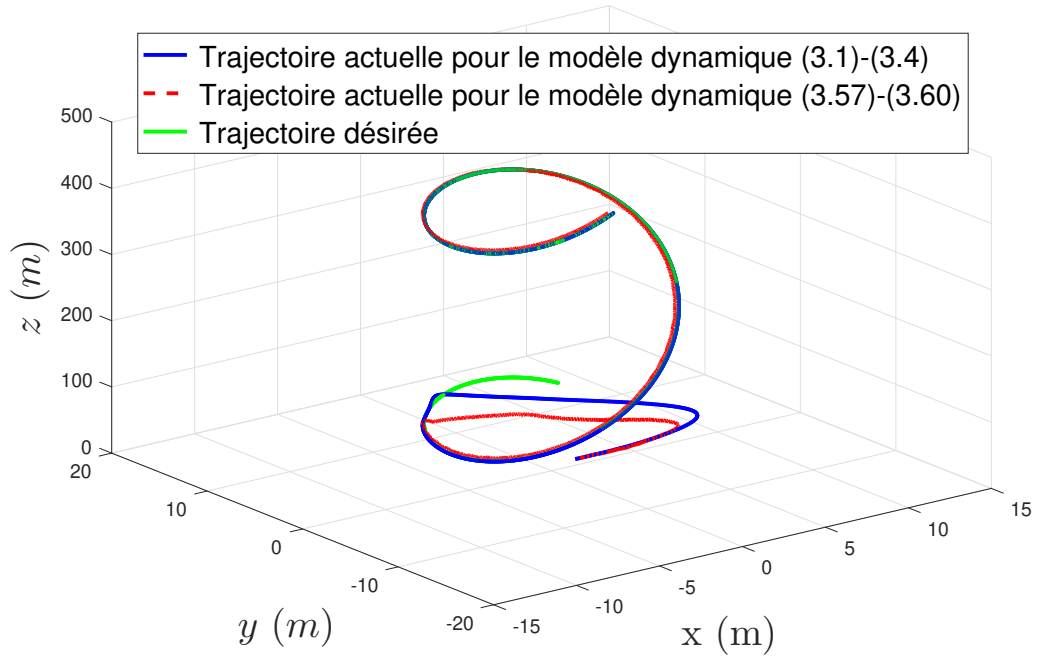


Figure 3.1: 3D plot of the actual and the desired trajectory

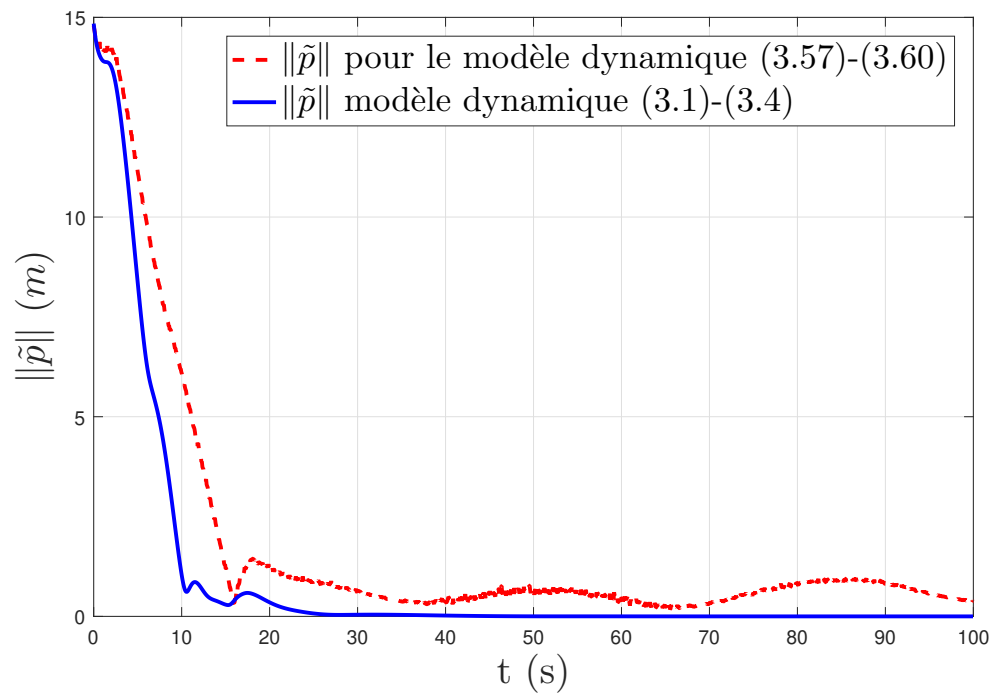


Figure 3.2: Norme de l'erreur de position

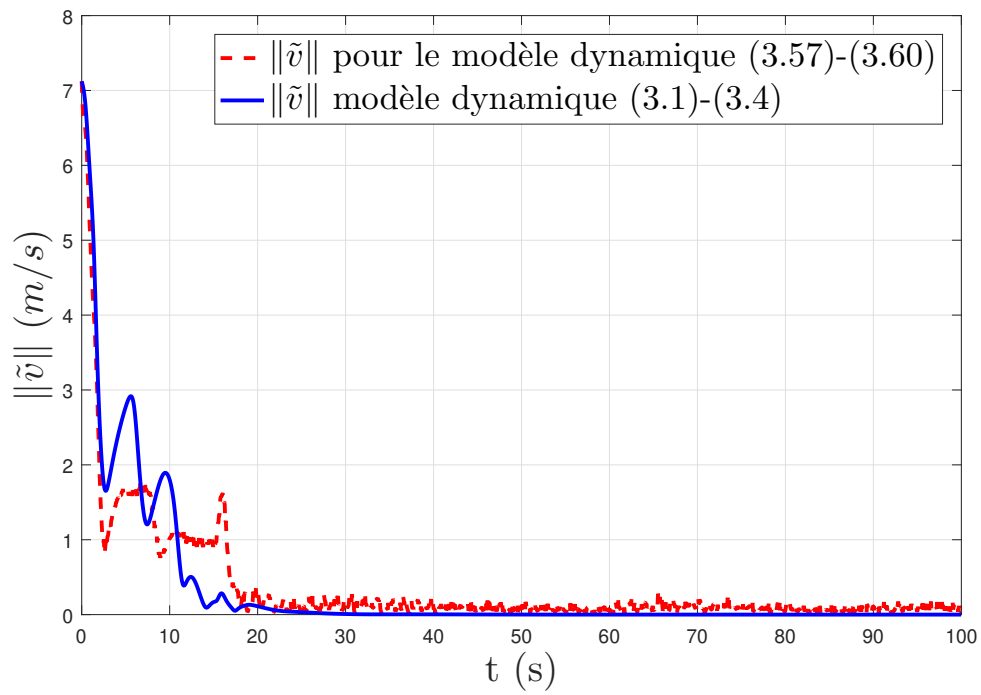


Figure 3.3: Norme de l'erreur de vitesse

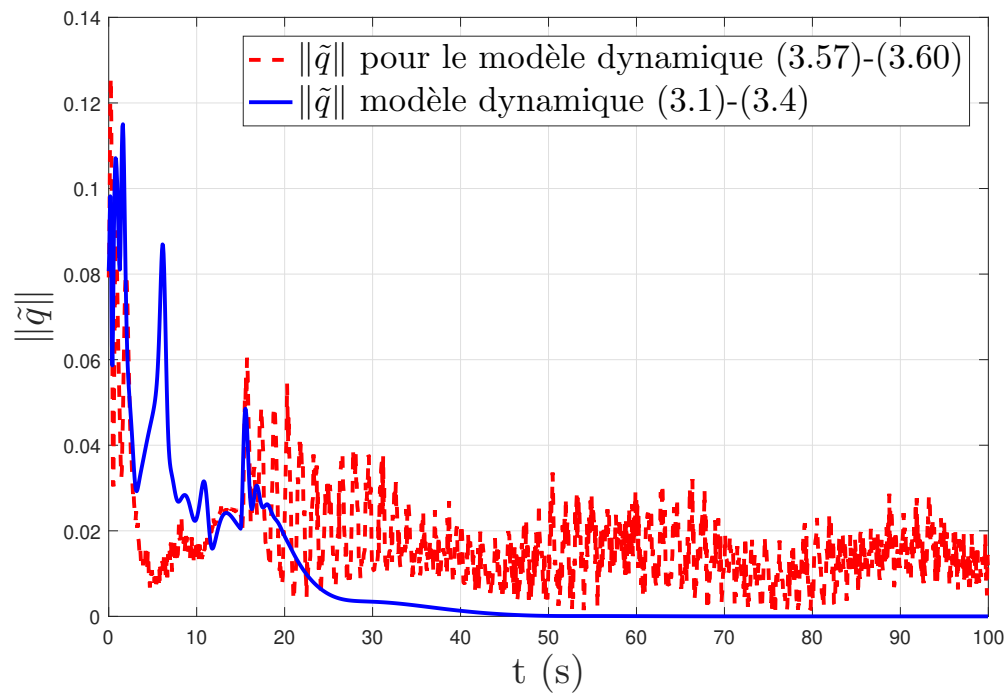


Figure 3.4: Norme de l'erreur d'attitude

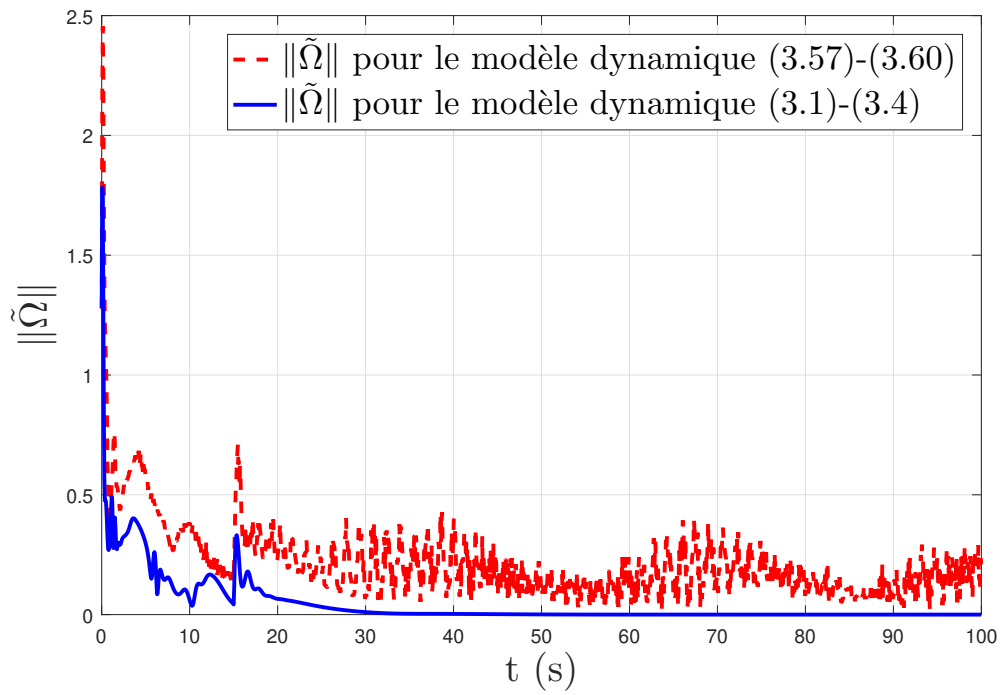


Figure 3.5: Norme de $\|\tilde{\Omega}\|$

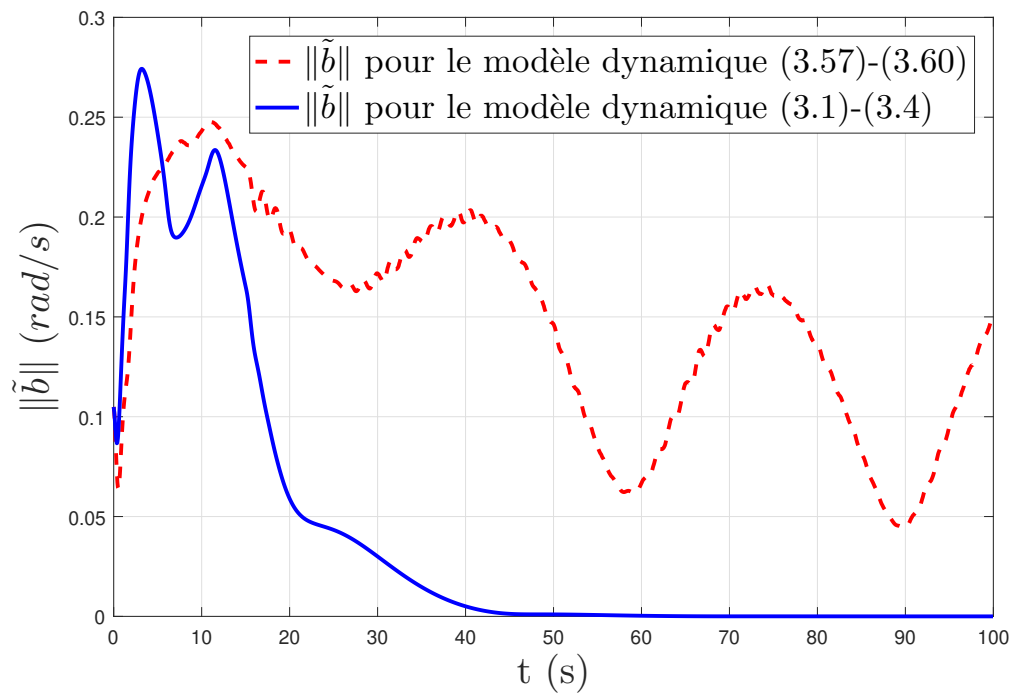


Figure 3.6: Norme de l'erreur de l'estimation du biais gyroscopique

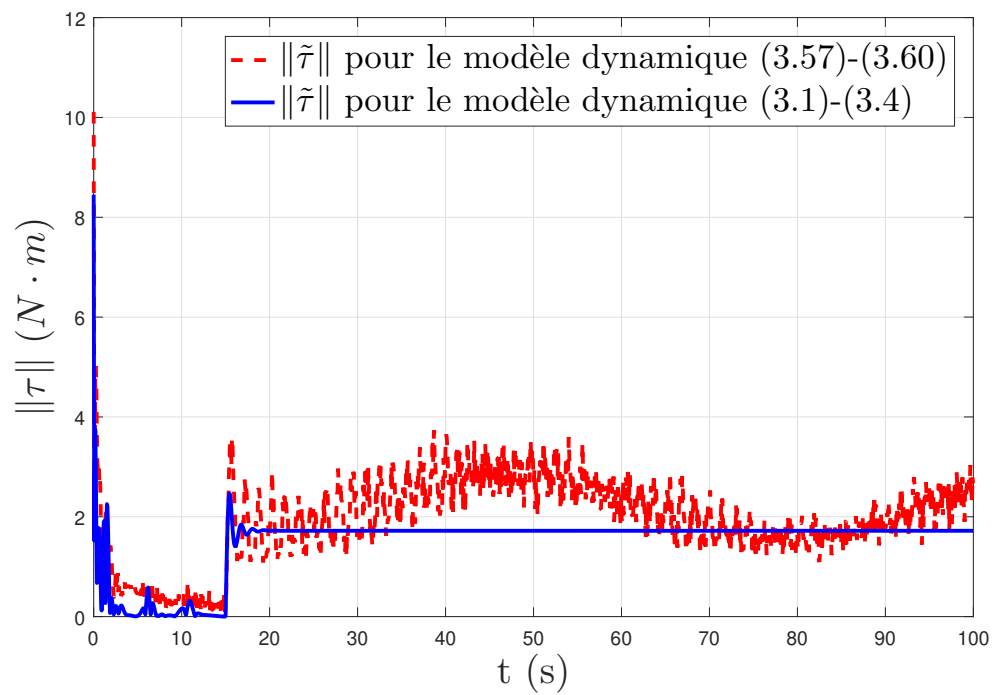


Figure 3.7: Norme du torque

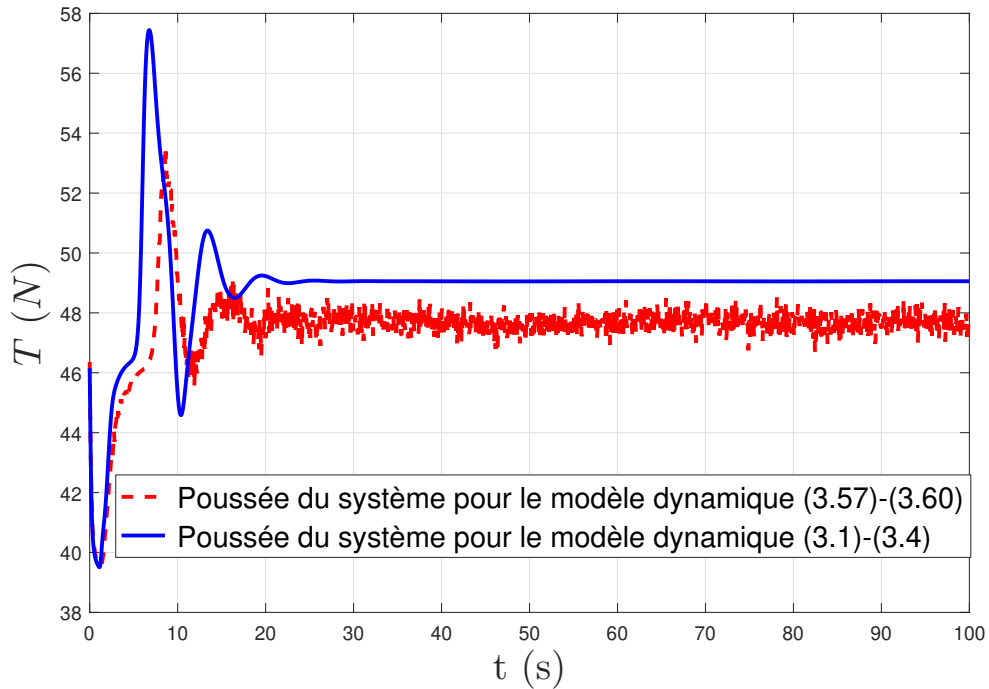


Figure 3.8: La poussé du système

3.6 Conclusion

Un schéma de contrôle adaptatif pour le suivi de trajectoire d'un VTOL-UAV, avec une matrice d'inertie inconnue, un biais gyroscopique, une perte d'efficacité des actionneurs et sujet à des couples de perturbations externes, a été présentée. Le contrôleur a été conçu de manière hiérarchique à partir de la dynamique translationnelle pour laquelle une loi de contrôle virtuelle *a priori* bornée a été conçue en premier lieu, suivie de la conception du couple de contrôle pour la dynamique de rotation en utilisant l'orientation désirée et l'accélération virtuelle conçue lors de la première étape. La limite *a priori* de l'accélération virtuelle permet d'éviter les problèmes de singularité pour l'extraction de l'attitude et de la poussée désirée de l'accélération désirée. Des mécanismes de projection ont été introduits pour doter les schémas de contrôle adaptatif proposés de propriétés de robustesse contre le bruit de mesure, et pour empêcher le passage à zéro de certaines estimations de paramètres afin d'éviter des problèmes de singularité dans la loi de commande. Une analyse de stabilité du système global en boucle fermée a été rigoureusement réalisée en utilisant les techniques de Lyapunov. Nous montrons la limite globale et la convergence vers zéro des erreurs de suivi, tout en garantissant que le contrôle de la poussée soit

3.6 Conclusion

tout le temps positif .

Les résultats de simulation présentés, sur un drone réaliste à ventilateur canalisé incluant le couplage et les effets aérodynamiques de friction et de traînée, illustrent l'efficacité du système de contrôle proposé malgré une matrice d'inertie inconnue, les biais du gyro, la perte d'efficacité et les perturbations persistantes.

Chapitre 4

Résultats expérimentaux

4.1 Introduction

Dans ce chapitre nous allons présenter les résultats d'implémentation de deux approches de commande sur la plateforme open source Crazyflie 2.0. Au début le but était d'implémenter la loi de commande que nous avons présentée dans le chapitre 3, mais en raison de problèmes techniques on s'est limité à l'implémentation de deux approches qui présentent moins de complexité de calcul. La première approche que nous avons implémentée consiste à un contrôleur simplifié qui présente moins de complexité en faisant la stabilisation de chaque dynamique à part. La deuxième approche est semblable à celle présentée dans le chapitre précédent mais sans considérer, pour des raisons pratiques, les contraintes prise en considération à savoir l'inertie inconnue, le biais gyroscopique et la perte d'efficacité des actionneurs. La commande en position de l'UAV VTOL n'est pas faite d'une manière directe mais plutôt d'une manière indirecte par le biais de la poussée totale du système et de l'orientation. Cela peut être fait de différentes manières. Pour les deux approches la stratégie de commande sera divisée en trois étapes :

Etape 1: Contrôleur de position;

Etape 2: Extraction d'attitude et de la poussée;

Etape 3: Contrôleur d'attitude.

4.2 Contrôle de position en utilisant une approche simplifiée

$$\dot{p} = v \quad (4.1)$$

$$\dot{v} = g\hat{e}_3 - \frac{T}{m}R^T\hat{e}_3 \quad (4.2)$$

$$\dot{Q} = \frac{1}{2} \begin{bmatrix} -q^T \\ q_0 I_{3 \times 3} + S(q) \end{bmatrix} \Omega \quad (4.3)$$

$$J\dot{\Omega} = -S(\Omega)J\Omega + \tau \quad (4.4)$$

Le schéma synoptique simplifié du système de commande du quadrotors est illustré dans la figure 4.1, la commande virtuel (μ_d) est générée par le bloc de contrôle de position. Elle est ensuite utilisée pour spécifier l'orientation et la poussée désirées

en faisant une extraction. L'orientation et la poussée désirées seront utilisées par la suite pour contrôler la position du drone VTOL.

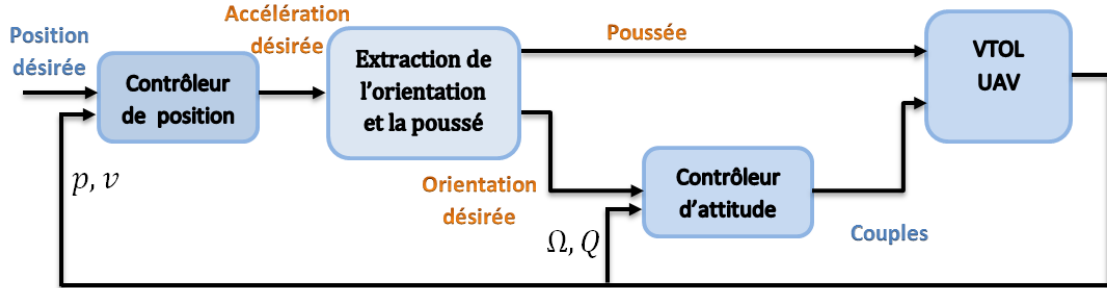


Figure 4.1: Schéma fonctionnel de l'approche simplifié

4.2.1 Conception de l'entrée de commande virtuel pour la dynamique translationnelle

En choisissant $p_d(t)$ comme une trajectoire référence désirée, on peut définir l'erreur de positions et de vitesse comme suit [17] [44]

$$\tilde{p} = p - p_d, \quad (4.5)$$

$$\tilde{v} = v - \dot{p}_d. \quad (4.6)$$

donc l'accélération du système est défini comme suit

$$\mu = g\hat{e}_3 - \frac{T}{m}R^T\hat{e}_3 \quad (4.7)$$

La dynamique d'erreur de translation (4.2) devient

$$\dot{\tilde{p}} = \tilde{v} \quad (4.8)$$

$$\dot{\tilde{v}} = \tilde{\mu} + \mu_d, \quad (4.9)$$

Définissant l'erreur d'accélération comme

$$\tilde{\mu} = \mu - \mu_d \quad (4.10)$$

la dynamique de translation peut être réécrite comme suit

$$\begin{aligned}\dot{\tilde{p}} &= \tilde{v} \\ \dot{\tilde{v}} &= \mu_d - \ddot{p}_d + \tilde{\mu}\end{aligned}\tag{4.11}$$

Considérant la fonction candidate de Lyapunov suivante

$$V = k_p[1 \ 1 \ 1] \ln(\cosh(\tilde{p})) + \tilde{v}^T \tilde{v}\tag{4.12}$$

Concevoir l'accélération désirée μ_d comme une entrée de commande virtuelle en supposant que $\tilde{\mu}$ comme suit

$$\mu_d = -k_p \tanh(\tilde{p}) - k_v \tanh(\tilde{v}) + \ddot{p}_d\tag{4.13}$$

Par conséquent, à la lumière de (4.11) et (4.13) on donne l'expression de la dérivé de V comme suit

$$\dot{V} = -k_v \tilde{v}^T \tanh(\tilde{v})\tag{4.14}$$

\dot{V} est semi-défini négative, ce qui garantit la bornitude de \tilde{p} et \tilde{v} . Invoquant le principe d'invariance de LaSalle

$$\dot{V} = 0 \Rightarrow \tilde{v} = 0 \Rightarrow \dot{\tilde{v}} = 0\tag{4.15}$$

ce qui engendre

$$\dot{\tilde{v}} = 0 = -\tanh(\tilde{p})\tag{4.16}$$

ce qui implique que $\tilde{p} \rightarrow 0$ quand $t \rightarrow \infty$

On peut aussi montrer que μ_d est bornée par

$$\|\mu_d\| \leq \|k_p \tanh(\tilde{p})\| + \|k_v \tanh(\tilde{v})\| + \|\ddot{p}_d\| \leq k_p + k_v + \sigma_r\tag{4.17}$$

où σ_r est la limite supérieure de la dérivée seconde de la trajectoire prédéfinie.

4.2.2 Extraction de l'attitude et de la poussée désirées

Reprenant les résultats présentés dans la section 3.3.2, la poussée et l'attitude désirée sont données par

$$u_t = \|\mu_d - g\hat{e}_3\| \quad (4.18)$$

$$q_0^d = \sqrt{\frac{1}{2} + \frac{g - \mu_{d3}}{2u_t}} \quad (4.19)$$

$$q_d = \frac{1}{2u_t q_0^d} (\mu_{d2} \quad -\mu_{d1} \quad 0)^T \quad (4.20)$$

à condition que

$$\mu_d \neq (0 \quad 0 \quad \kappa), \quad \text{for } \kappa \geq g \quad (4.21)$$

depuis (4.13) on peut voir que

$$|\mu_d| \leq k_p + k_v + \sigma_r \quad (4.22)$$

ce qui implique

$$k_p + k_v + \sigma_r < g \quad (4.23)$$

4.2.3 Commande de l'attitude

Utilisant le résultat du travail présenté dans [6], l'attitude désirée extraite dans section précédente est ensuite utilisée comme une consigne pour le contrôleur d'attitude. L'erreur d'attitude au sens du quaternion peut être donnée comme suit

$$\tilde{Q} = Q_d^{-1} \odot Q \quad (4.24)$$

Ce qui mène à la dynamique d'erreur suivante

$$\dot{\tilde{Q}} = \frac{1}{2} \begin{bmatrix} -q^T \\ \tilde{q}_0 I_{3 \times 3} + S(\tilde{q}) \end{bmatrix} \Omega \quad (4.25)$$

$$J\dot{\Omega} = -S(\Omega)J\Omega + \tau \quad (4.26)$$

Où la vitesse angulaire désirée Ω_d est 0, donc $\tilde{\Omega} = \Omega$. Comme démontré dans [6]

la stabilité asymptotique globale de l'attitude au point d'équilibre ($R = I, \Omega = 0$) du est assurée en utilisant l'entrée de commande suivante

$$\tau = -\Gamma\Omega - \alpha\tilde{q} \quad (4.27)$$

où $\Gamma = \Gamma^T > 0$ and $\alpha > 0$. Choississant la fonction de Lyapunov candidate suivante

$$V = \alpha\tilde{q}^T\tilde{q} + \alpha(\tilde{q}_0 - 1)^2 + \frac{1}{2}\Omega^T J^{-1}\Omega \quad (4.28)$$

Sa dérivée par rapport au temps au long des trajectoires de (4.25) est donnée par

$$\dot{V} = \alpha\tilde{q}_0\tilde{q}^T\Omega + \alpha\tilde{q}^T S(q)\Omega - \alpha\tilde{q}_0\tilde{q}^T\Omega + \alpha\tilde{q}^T\Omega - \Omega^T S(\Omega)J\Omega + \Omega^T\tau \quad (4.29)$$

appliquant la loi de commande (4.27) la dérivée V devienne comme suit

$$\dot{V} = -\Omega^T\Gamma\Omega \leq 0 \quad (4.30)$$

ce qui implique la bornitude de tous les signaux et la convergence asymptotique de Ω vers 0 lorsque t tend vers ∞ . En utilisant le principe d'invariance de LaSalle, on peut montrer que \tilde{q} converge asymptotiquement vers zéro lorsque t tend vers ∞ .

4.2.4 Génération de trajectoire polynomial

Afin de déplacer le drone VTOL d'une position initiale p_0 à un instant donné t_0 à une position finale p_f à un instant t_f nous utilisant une planification de trajectoire polynomiale pour générer la trajectoire désirée. La trajectoire est générée en donnant un profile bien définit, où la position, la vitesse et l'accélération du drone seront spécifiées à chaque instant de temps entre t_0 et t_f . Pour générer les trajectoires désirées nous utilisant de fonctions polynomiales par morceaux de degré 5 en fonction du temps tels que

$$r(t) = \sum_{i=0}^{i=5} a_i t^i \quad (4.31)$$

où

$$r(t) = [p_d(t)^T \psi(t)]^T \quad (4.32)$$

ou les six conditions aux limites sont données comme suit [44]:

$$\begin{aligned}
 r(t_i) &:= r_i, & r(t_f) &= r_f \\
 \dot{r}(t_i) &:= \dot{r}_i, & \dot{r}(t_f) &= \dot{r}_f \\
 \ddot{r}(t_i) &:= \ddot{r}_i, & \ddot{r}(t_f) &= \ddot{r}_f
 \end{aligned} \tag{4.33}$$

Dans ce cas, (si $T = t_f - t_i$) les coefficients du polynôme sont

$$\begin{aligned}
 a_0 &= r_i \\
 a_1 &= \dot{r}_i \\
 a_2 &= \frac{1}{2}\ddot{r}_i \\
 a_3 &= \frac{1}{2T^3}[20(r_f - r_i) - (8\dot{r}_f + 12\dot{r}_i)T - (3\ddot{r}_f - \ddot{r}_i)T^2] \\
 a_4 &= \frac{1}{2T^4}[30(r_f - r_i) + (14\dot{r}_f + 16\dot{r}_i)T + (3\ddot{r}_f - 2\ddot{r}_i)T^2] \\
 a_5 &= \frac{1}{2T^5}[12(r_f - r_i) - 6(\dot{r}_f + \dot{r}_i)T - (\ddot{r}_f - \ddot{r}_i)T^2]
 \end{aligned} \tag{4.34}$$

À partir de là, tous les paramètres de trajectoire de vol en translation sont définis pour toutes les instances de temps entre t_i et t_f . De plus, en choisissant une valeur de T suffisamment grande, on peut assurer $\|\ddot{r}(t)\| = \sigma_r$ est suffisamment petit pour ne pas nuire aux performances du contrôleur (4.13).

4.2.5 Résultats de simulation

Dans cette section, nous illustrons à travers les résultats de simulation les performances de la commande. Les simulations sont effectuées pour l'objectif de suivi d'une trajectoire ellipsoïde. Les paramètres du corps rigide UAV sont les suivants: la matrice d'inertie $J = \text{diag}(0.6, 0.6, 0.3)kg \cdot m^2$, la masse $m = 1.28kg$, et le bras de levier de contrôle de couple $l = 0.5$. L'accélération gravitationnelle est prise comme $g = 9.81m/s^2$, et la trajectoire de référence souhaitée a été spécifiée $p_d(t) = [10 \sin(0.1t + \frac{2\pi}{3}), 15 \sin(0.1t + \frac{2\pi}{5}), 5t]^T$. Les conditions initiales ont été prises comme : $p(0) = [2 \ 1 \ 0]m$, $v(0) = [5 \ 0 \ 0.5]^T m/s$, $(q_0(0), q(0)) = (1, [0 \ 0 \ 0]^T)$, $\Omega(0) = [0 \ 0 \ 0]^T$. Les paramètres sont les suivants: $k_p = 1.75, k_v = 2, \gamma_q =$

4.2 Contrôle de position en utilisant une approche simplifiée

$$6 \times 10^{-3}, K_{\Omega} = \text{diag}(0.5, 0.54, 0.215), K_q = 4$$

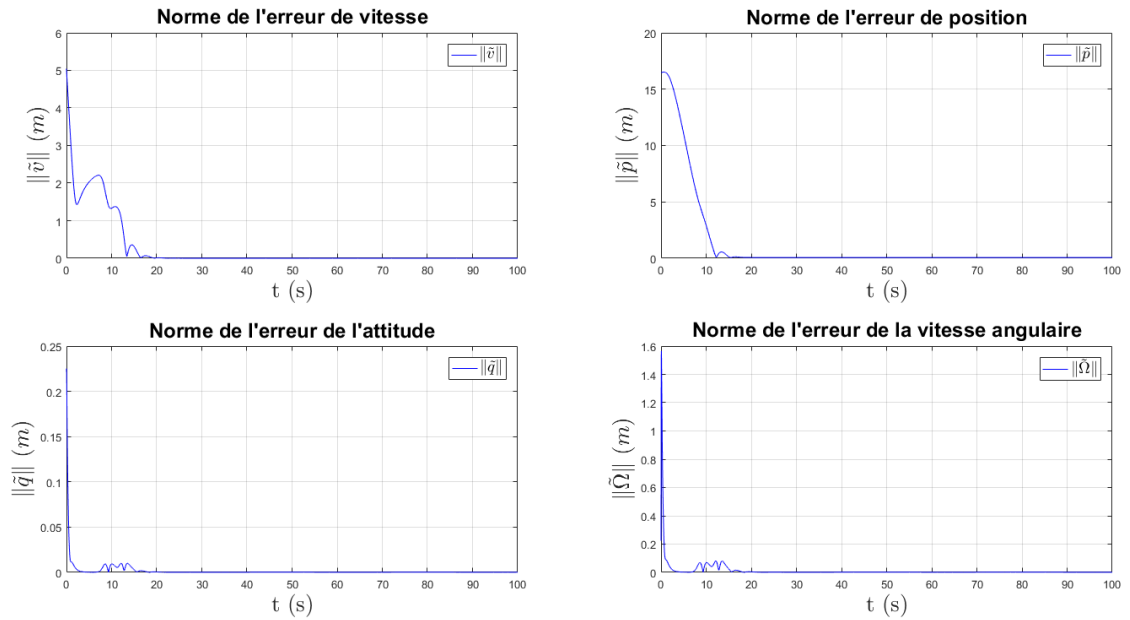


Figure 4.2: Normes des erreurs

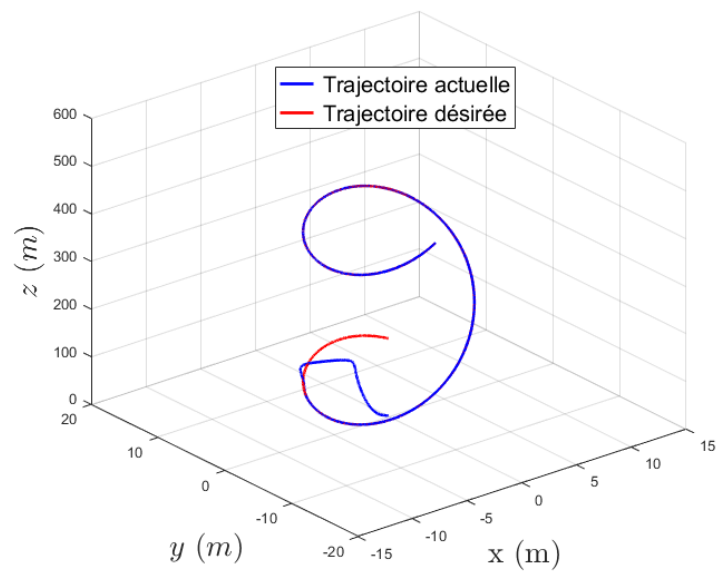


Figure 4.3: Trajectoires en 3D

4.2.6 Résultats d'implémentation

Les lois de commande présentées dans les section 4.2.1 et 4.2.3 ont été implémentées sur la plateforme open-source Crazyflie 2.0. Les résultats de mise en œuvre sont présenté pour des commandes de décollage automatique, de maintien de position ainsi que le déplacement vers des points bien précis dans l'espace.

Le décollage automatique utilise la trajectoire décrite dans la section 4.2.4, où les composantes x et y de la trajectoire de référence sont maintenues constantes tandis que la composante z finale est fixée à $1m$. Dans le suivi de trajectoire, le drone doit voler jusqu'à un certain nombre de points. Dans cette dernière expérience de suivi de trajectoire, il est montré que le contrôleur pratique est capable de suivre une référence souhaitée avec une erreur de position moyenne de $30cm$ et une erreur de vitesse moyenne de $5cm/s$. L'erreur de position est relativement importante en raison du faible gain choisi, ce dernier garantie le maintien de faible oscillations, car de grandes oscillations pourraient endommager le drone.

4.2.6.1 Expérience 1: Décollage automatique maintien de position et atterrissage automatique

Dans cette première expérience, nous mettons en œuvre l'approche simplifier sur notre système de pilotage automatique. L'expérience consiste à faire un décollage automatique, un maintien de position et enfin faire un atterrissage automatique. L'opération de décollage automatique sert à amener le crazyflie à une hauteur de $1 m$ en $2 s$ tout en préservant les coordonnées x, y . Le maintien de position veille à garder le quadrotor aux même coordonnée tout en le stabilisant son attitude à $R = I_{3 \times 3}$. Les deux opérations de décollage et d'atterrissage se font en suivant des trajectoires polynomiales comme montré dans la section 4.2.4. Les gains et les paramètres impliqués dans le contrôleur sont choisi comme suit $k_p = 2.2$, $k_v = 2.3$, $\Gamma = diag(5000, 5000, 9000)$, $\alpha = 80000$. Les résultats expérimentaux sont reportés dans les Fig 4.4 - 4.8.

On peut voir que le système de contrôle veille à ce que les erreurs de vitesse et d'attitude soit faible et qu'il assure un suivie acceptable de position. L'erreur de position est comprise entre $0.04m$ et $0.36m$. Elle est due en grande partie aux incertitudes induites par le système de positionnement LPS.

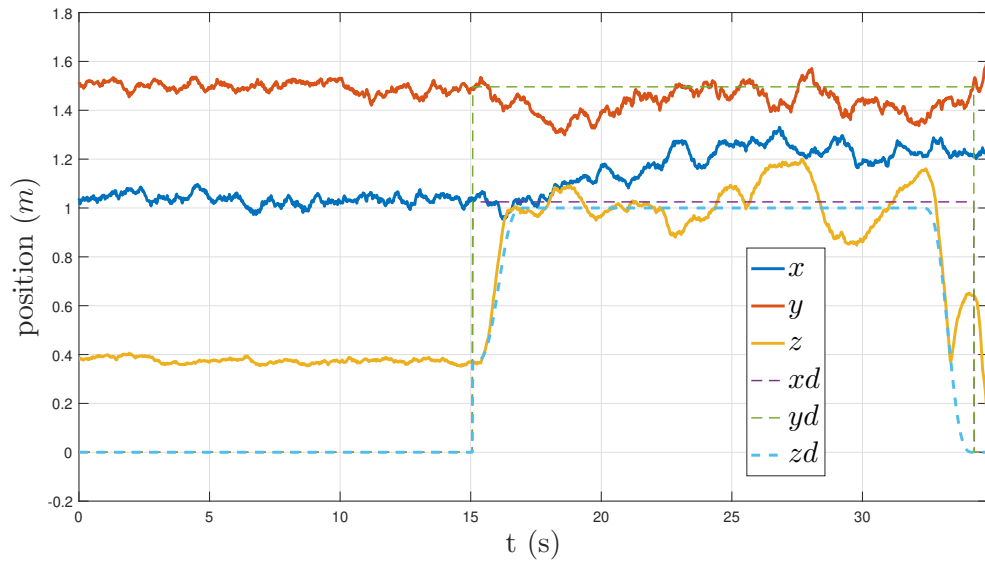


Figure 4.4: Position

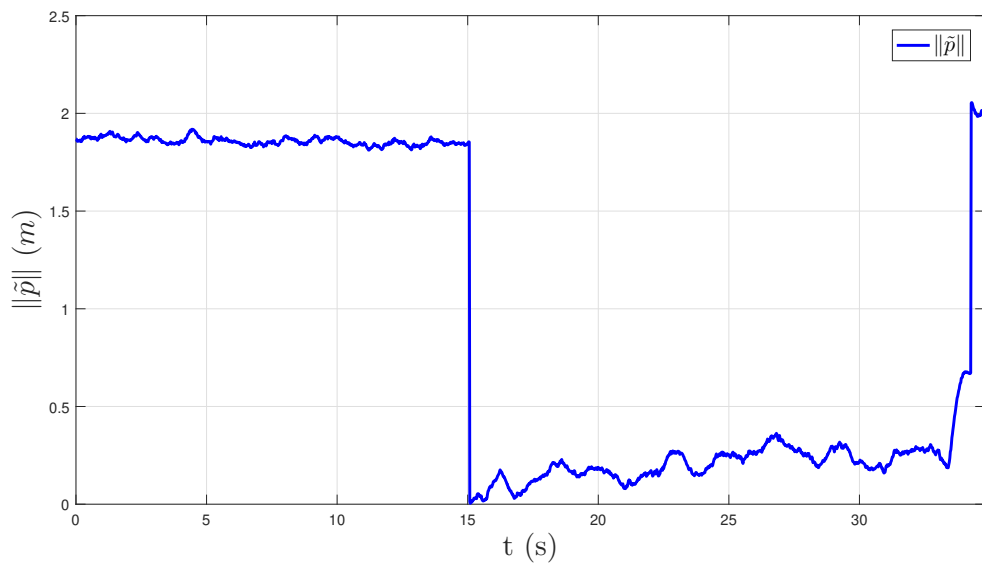


Figure 4.5: Norme de l'erreur de la position

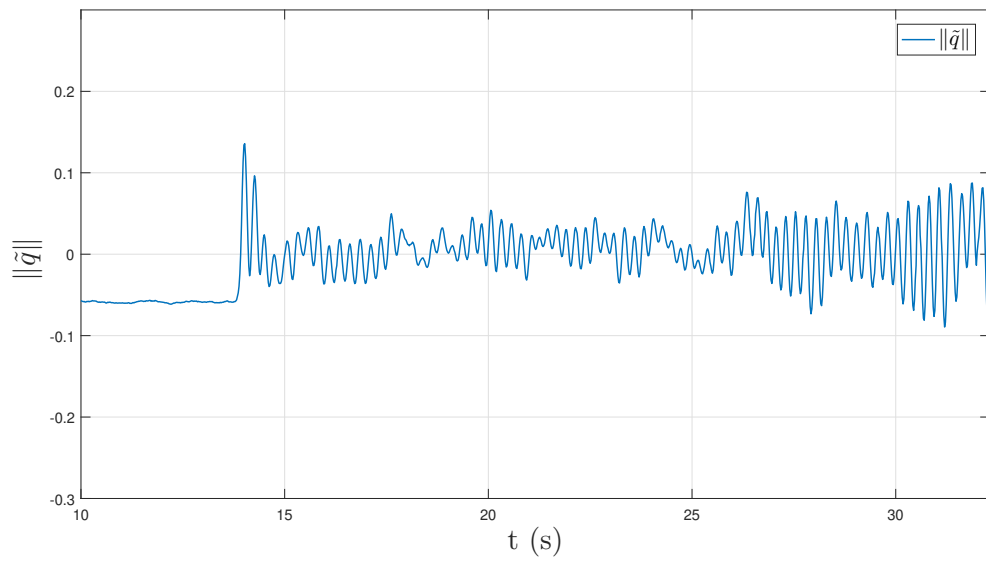


Figure 4.6: Norme de l'erreur de l'attitude

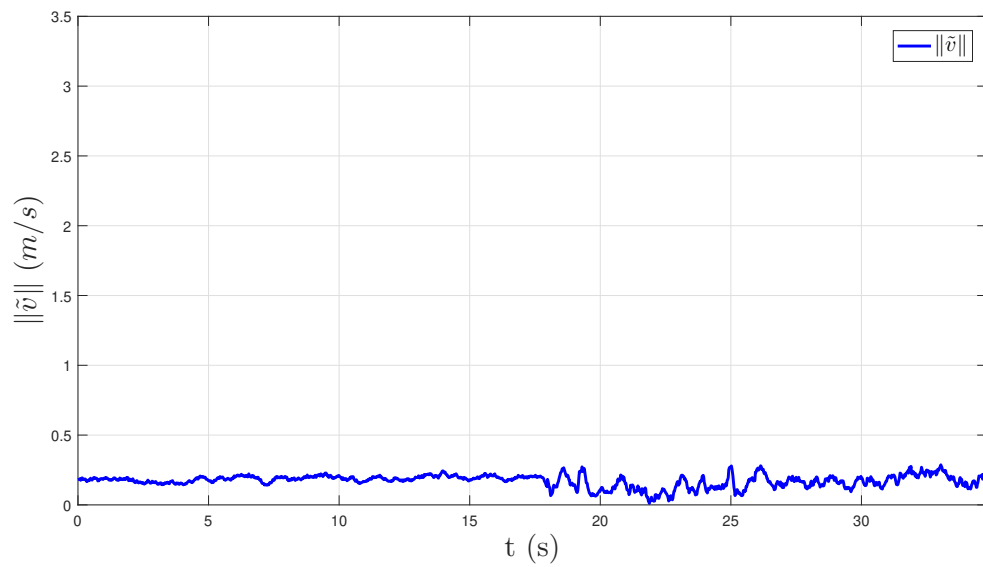


Figure 4.7: Norme de l'erreur de la vitesse

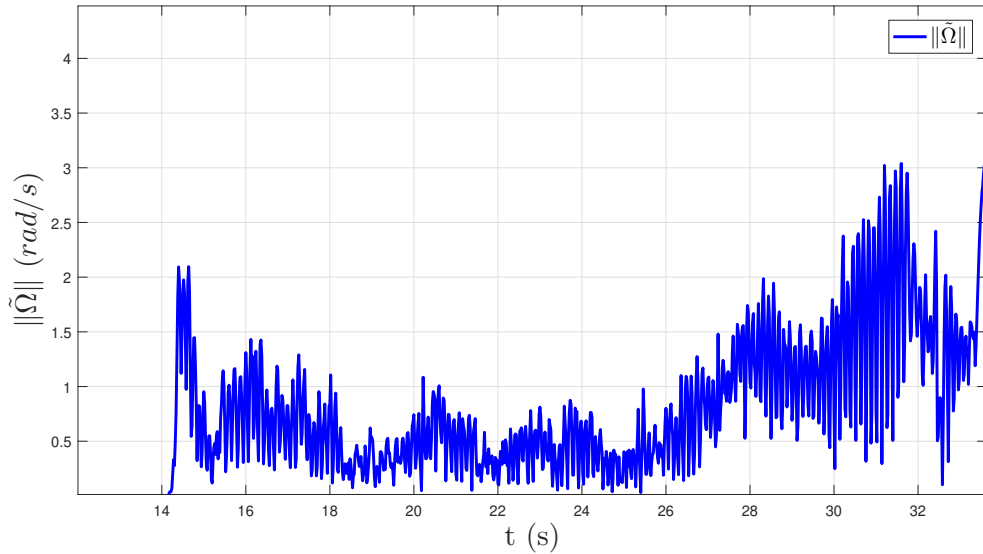


Figure 4.8: Norme de l'erreur de la vitesse angulaire

4.2.6.2 Expérience 2: Aller vers un point dans l'espace

Dans cette expérience, le crazyflie a décollé automatiquement du sol et son attitude est stabilisée à $R = I_{3 \times 3}$ (c.-à-d. $\phi = 0 \text{ deg}$, $\theta = 0 \text{ deg}$ et $\psi = 0 \text{ deg}$) et au coordonnées x , y de démarrage et à une altitude $z = 1m$. Dans la phase du vol stationnaire, la poussée totale des moteurs doit être égale à la force de gravité. Ensuite, le crazyflie est envoyé à un point désiré ($x = 0.5m$ $y = 0.5m$ $z = 1.5m$) en faisant le tracking d'une trajectoire polynomiale (section 4.2.4). Enfin, le quadrotor atterrit automatiquement. Les gains impliqués dans le contrôleur sont les même utilisés dans l'expérience 1. Les résultats expérimentaux sont reportés dans les Fig 4.9 - 4.14. L'algorithme de contrôle implémenté assure le suivi de la trajectoire désirée avec une erreur moyenne de 20 cm qui est due à l'erreur du système de positionnement LPS (la précision de l'ancre est de 10 cm et elle due aussi à la faible valeur du terme proportionnel k_p pour assurer une oscillation minimale. De plus, comme le montre la figure 4.12, les angles d'Euler restent toujours entre $(-5^\circ, +5^\circ)$, ce qui prouve la stabilité de l'attitude.

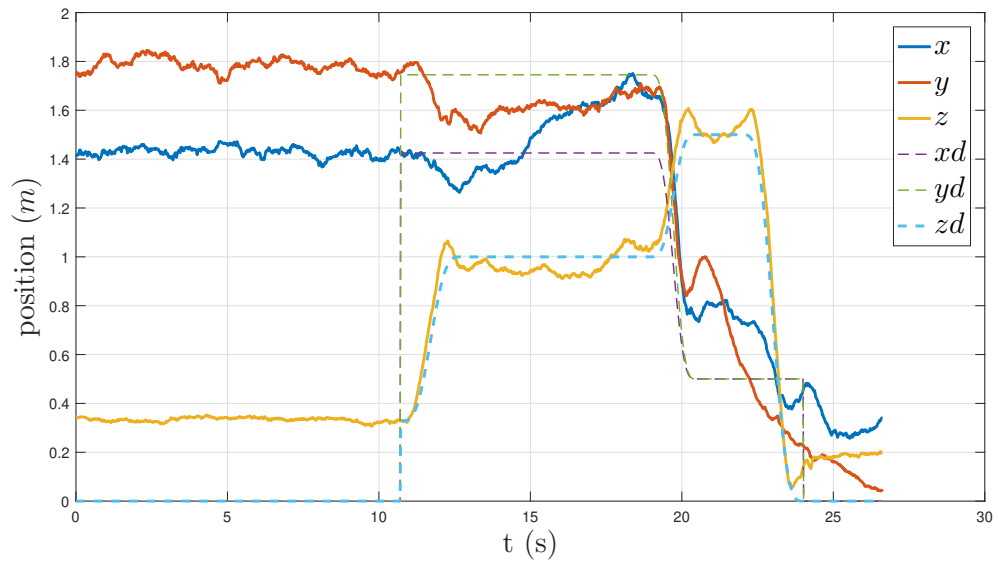


Figure 4.9: Position

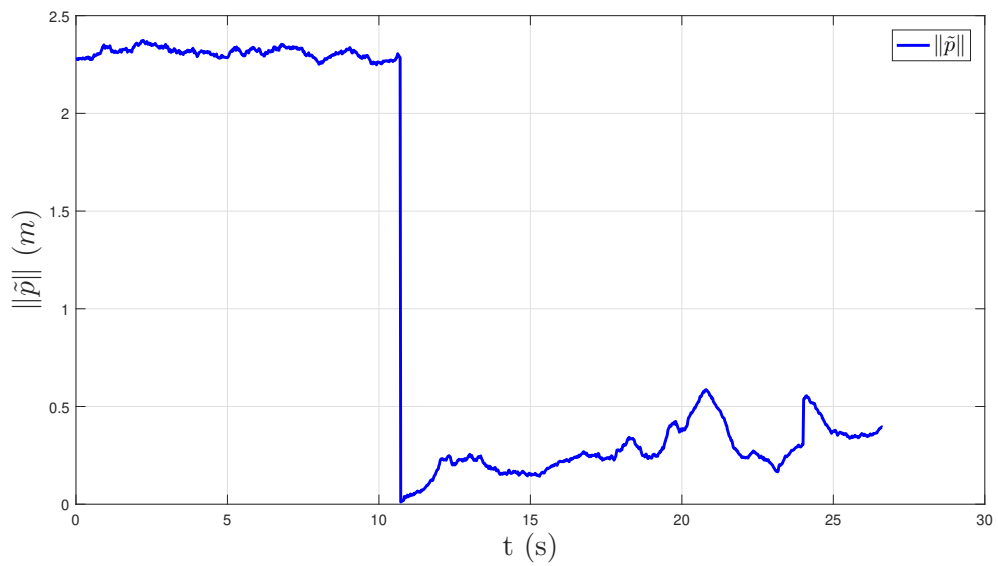


Figure 4.10: Norme de l'erreur de la position

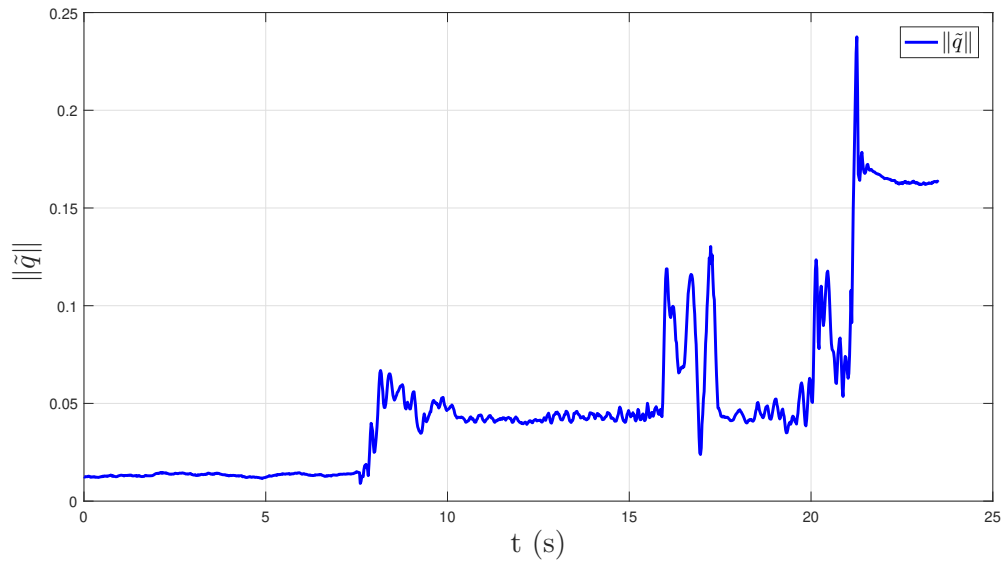


Figure 4.11: Norme de l'erreur de l'attitude

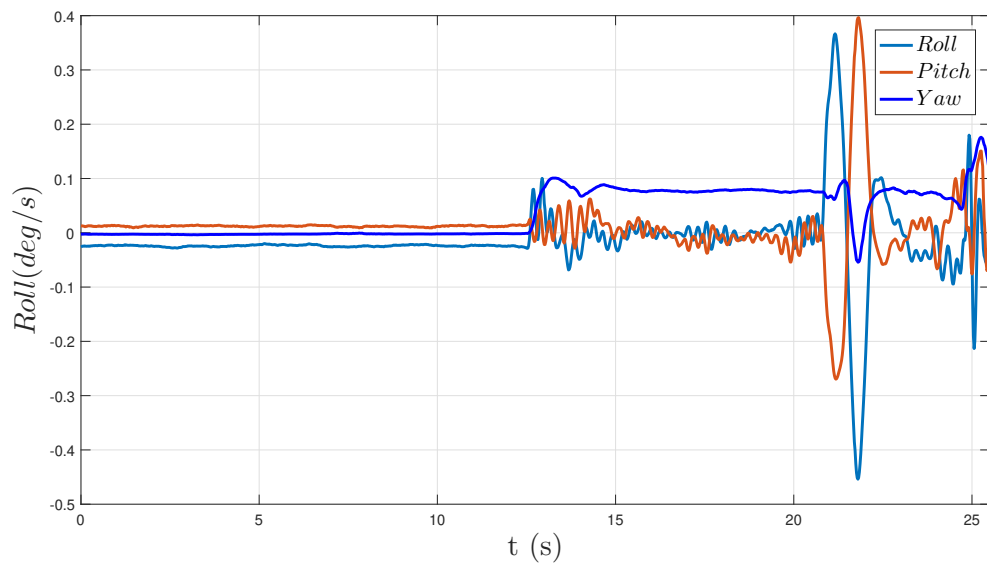


Figure 4.12: L'attitude

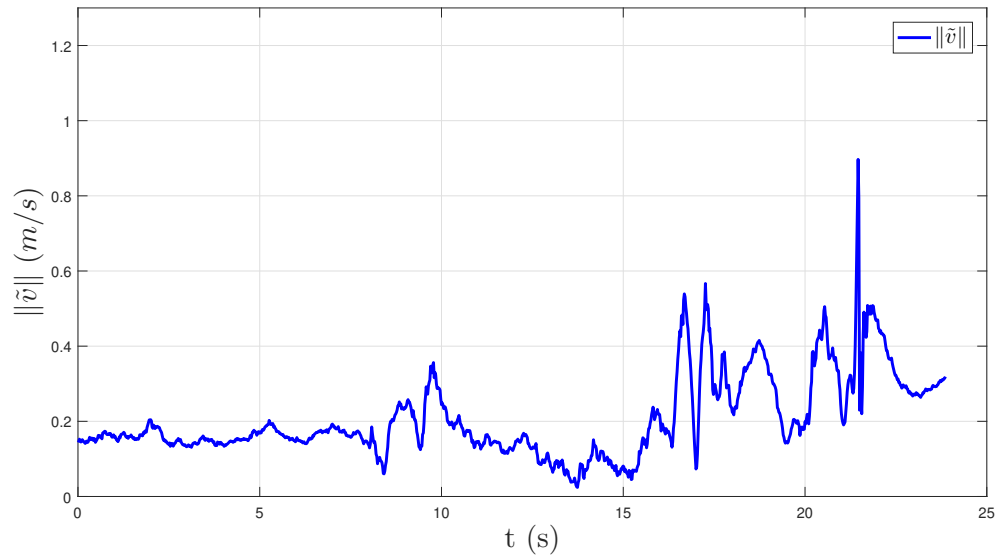


Figure 4.13: Norme de l'erreur de la vitesse

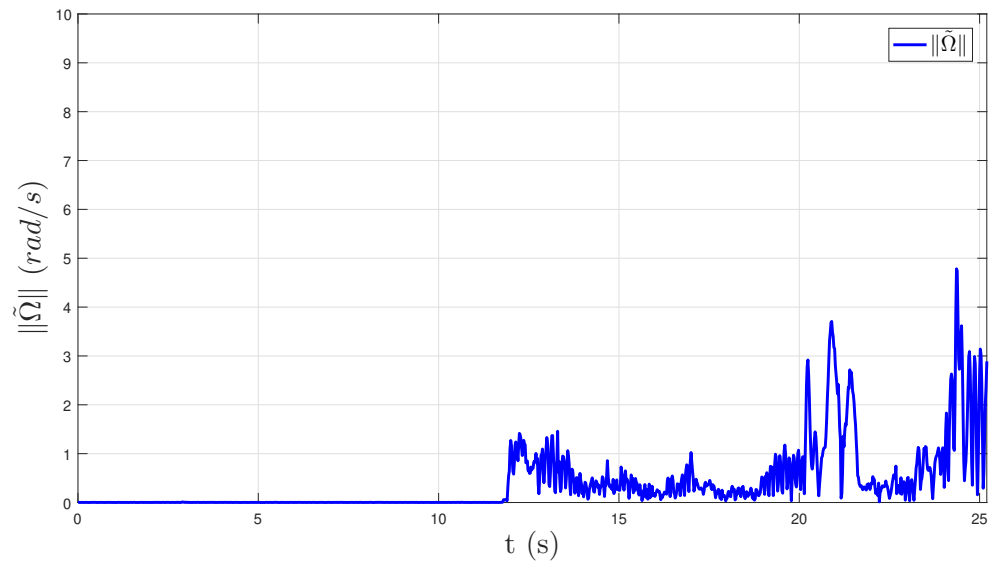


Figure 4.14: Norme de l'erreur de la vitesse angulaire

4.3 Contrôle de position en utilisant l'approche du Backstepping où les perturbations sont nulles et les paramètres du système sont connus

Le contrôleur synthétisé dans le chapitre 3 ne peut pas être appliqué tel qu'il est sur le quadrotor, pour lequel le phénomène de la perte partielle de l'efficacité est particulier car il existe un couplage entre le contrôle de la poussée et le contrôle de l'orientation.

De plus, pour des raisons de complexité de calcul lors de l'implémentation, il a été nécessaire de réfléchir à une approche réduite où le bias de la vitesse angulaire et les perturbations sont nulles et l'inertie est connue.

Dans l'approche simplifier, l'erreur de l'accélération est supposée nulle ($\tilde{\mu} = 0$). La chose qui ne garantit pas la stabilisation de la position de l'UAV VTOL quelles que soient les conditions initiales. Le terme $\tilde{\mu}$ sera pris en compte dans la conception des couples de commande. L'approche comporte deux parties distinctes. Tout d'abord on détermine l'accélération désirée à partir de laquelle la poussée et l'attitude désirée seront extraites. Ensuite, le couple de commande est synthétisé pour la dynamique de rotation, en utilisant l'orientation issue du block d'extraction.

4.3.1 Synthèse du contrôleur

En supposant que la matrice d'inertie est connu et que les perturbations son nulles, la dynamiques des erreurs (3.31) - (3.34) devient:

$$\begin{aligned}
 \dot{\tilde{p}} &= \tilde{v} \\
 \dot{\tilde{v}} &= \mu_d + \tilde{\mu} - \ddot{r}(t) \\
 \dot{\tilde{Q}} &= \frac{1}{2} \begin{bmatrix} \tilde{q}^T(\Omega_d - \tilde{\Omega} - \bar{\Omega}) \\ \tilde{q}_0(\bar{\Omega} + \tilde{\Omega} - \Omega_d) + S(\tilde{q})(\bar{\Omega} + \tilde{\Omega} + \Omega_d) \end{bmatrix} \\
 J\dot{\tilde{\Omega}} &= -S(\Omega)J\Omega - J\dot{\tilde{\Omega}} + \tau
 \end{aligned} \tag{4.35}$$

Où

4.3 Contrôle de position en utilisant l'approche du Backstepping où les perturbations sont nulles et les paramètres du système sont connus

$$\begin{aligned}
\tilde{p} &= p - p_d \\
\tilde{v} &= v - \dot{p}_d \\
\tilde{\mu} &= \mu - \mu_d \\
\tilde{\Omega} &= \Omega - \bar{\Omega}\tilde{R} &= RR_d^T
\end{aligned} \tag{4.36}$$

La procédure de synthèse de la commande virtuel est similaire à celle présentée dans la section 3.3.1.

On choisie la fonction de Lyapunov suivante

$$V = k_p \mathbb{1}^T \ln(\cosh(\tilde{p})) + \frac{1}{2} \tilde{v}^T \Gamma_v \tilde{v} + 2\gamma_q (1 - \tilde{q}_0) + \frac{1}{2} \tilde{\Omega}^T J \tilde{\Omega} \tag{4.37}$$

Sa dérivée temporelle le long des trajectoires du système en boucle fermée est donnée par :

$$\dot{V} = k_p \tilde{v}^T \tanh(\tilde{p}) + \tilde{v}^T \Gamma_v (\mu_d + \tilde{\mu} - \dot{p}_d) - \gamma_q \tilde{q}^T (\Omega_d - \tilde{\Omega} - \bar{\Omega}) + \tilde{\Omega}^T J \dot{\tilde{\Omega}} \tag{4.38}$$

On rappel l'expression de cette accélération désirée :

$$\mu_d = \ddot{p}_d - k_p \Gamma_v^{-1} \tanh(\tilde{p}) - k_v \tanh(\tilde{v}) \tag{4.39}$$

L'opération de l'extraction de l'attitude et de la poussée désirée donne aussi le même résultats obtenu dans la section 3.3.2.

De la même manière, en utilisant les expressions de l'accélération désirée μ_d et l'orientation désirée Ω_d données par (3.12) et (3.25), respectivement, \dot{V} devient :

$$\dot{V} = -k_v \tilde{v}^T \Gamma_v \tanh(\tilde{v}) - \gamma_q \tilde{q}^T (M(\mu_d) \dot{\mu}_d + \frac{u_t}{\gamma_q} S(\bar{q}) R \tilde{v} - \bar{\Omega} - \tilde{\Omega}) + \tilde{\Omega}^T J \dot{\tilde{\Omega}} \tag{4.40}$$

où l'expression de $\dot{\mu}_d$ est déjà donnée dans (3.27). L'expression de \dot{V} devient :

4.3 Contrôle de position en utilisant l'approche du Backstepping où les perturbations sont nulles et les paramètres du système sont connus

$$\begin{aligned} \dot{V} = & -k_v \tilde{v}^T \Gamma_v \tanh(\tilde{v}) - \gamma_q \tilde{q}^T \left[M(\mu_d) \left(p_d^{(3)} - \gamma_q D_t(\tilde{p}) \tilde{v} - k_v D_t(\tilde{v}) (\tilde{\mu} - \gamma_q \tanh(\tilde{p})) \right. \right. \\ & \left. \left. - k_v \tanh(\tilde{v}) \right) + \frac{u_t}{\gamma_q} S(\tilde{q}) R \tilde{v} - \bar{\Omega} - \tilde{\Omega} \right] + \tilde{\Omega}^T (-S(\Omega) J \Omega - J \dot{\tilde{\Omega}} + \tau) \end{aligned} \quad (4.41)$$

Définissons la variable auxiliaire comme suit :

$$\bar{\Omega} = M(\mu_d) \left[p^{(3)} - k_p \Gamma_v^{-1} D_t(\tilde{p}) \tilde{v} + k_v k_p D_t(\tilde{v}) \Gamma^{-1} \tanh(\tilde{p}) \right] + \frac{2}{\gamma_q} u_t S(\tilde{q}) R \Gamma_v \tilde{v} - K_q \tilde{q} \quad (4.42)$$

Pour atteindre notre objectif, nous proposons le couple de commande suivant pour la dynamique de rotation

$$\tau = S(\Omega) J \Omega + J \dot{\tilde{\Omega}} - \gamma_q \tilde{q} - K_\Omega \tilde{\Omega} \quad (4.43)$$

où $\dot{\tilde{\Omega}} := \varpi$ voir (A.7). En suivant le même cheminement de développement dans la section 3.4 equations (3.52)-(3.56), l'expression de \dot{V} devient :

$$\dot{V} \leq -\tilde{v}^T \Delta_v \tanh(\tilde{v}) - \gamma_q \tilde{q}^T \Delta_q \tilde{q} - \tilde{\Omega}^T K_\Omega \tilde{\Omega} \quad (4.44)$$

où Δ_v et Δ_q sont donnés par (3.53). Arrivons a ce résultat, on peut conclure que les états $(\tilde{p}, \tilde{v}, \tilde{Q}, \tilde{\Omega})$ restent bornés pour tout $t > 0$. En outre, les états $(\tilde{v}, \tilde{q}, \tilde{\Omega})$ tend vers $(0, 0, 0)$ quand $t \rightarrow \infty$.

En utilisant les expressions de μ_d , $\bar{\Omega}$ et τ dans (4.35), nous obtenons un système autonome et par conséquent, en faisant appelle au théorème d'invariance de LaSalle, au vu du système (4.35) avec la commande virtuel (4.39), on peut en conclure que

$$\dot{\tilde{v}} := -k_p \tanh(\tilde{p}) - k_v \tanh(\tilde{v}) - 2u_t R^T S(\tilde{q})^T \tilde{q} = 0 \quad (4.45)$$

ce qui donne en utilisant le résultat précédent

$$-k_p \tanh(\tilde{p}) = 0 \quad (4.46)$$

ce qui implique que $\tilde{p} \rightarrow 0$ quand $t \rightarrow \infty$

Cela satisfait l'objectif de contrôle de position et montre la stabilité asymptotique

4.3 Contrôle de position en utilisant l'approche du Backstepping où les perturbations sont nulles et les paramètres du système sont connus

globale du point d'équilibre $(\tilde{p}, \tilde{v}, \tilde{q}, \tilde{\Omega}) = (0, 0, 0, 0)$. La principale différence entre ce contrôleur et le contrôleur simplifié est que l'hypothèse $\tilde{\mu} \approx 0$ n'est plus nécessaire et, de ce fait, ce contrôleur sera capable de maintenir la stabilité même sous de fortes accélérations [44].

4.3.2 Résultats de simulation

Dans cette section, nous illustrons à travers les résultats de simulation les performances de la commande. Les simulations sont effectuées pour l'objectif de suivi d'une trajectoire ellipsoïde. Les paramètres du corps rigide UAV sont les suivants: la matrice d'inertie $J = \text{diag}(0.6, 0.6, 0.3) \text{kg} \cdot \text{m}^2$, la masse $m = 5 \text{kg}$, et le bras de levier de contrôle de couple $l = 0.5$. L'accélération gravitationnelle est prise comme $g = 9.81 \text{m/s}^2$, et la trajectoire de référence souhaitée a été spécifiée $p_d(t) = [10 \sin(0.1t + \frac{2\pi}{3}), 15 \sin(0.1t + \frac{2\pi}{5}), 5t]^T$. Les conditions initiales ont été prises comme : $p(0) = [2 \ 1 \ 0] \text{m}$, $v(0) = [5 \ 0 \ 0.5]^T \text{m/s}$, $(q_0(0), q(0)) = (1, [0 \ 0 \ 0]^T)$, $\Omega(0) = [0 \ 0 \ 0]^T$. Les paramètres sont les suivants: $k_p = 10, k_v = 8.5, \gamma_q = 6 \times 10^{-3}, K_\Omega = \text{diag}(0.00129, 0.0016, 0.0012), K_q = 55I_{\times 3}$

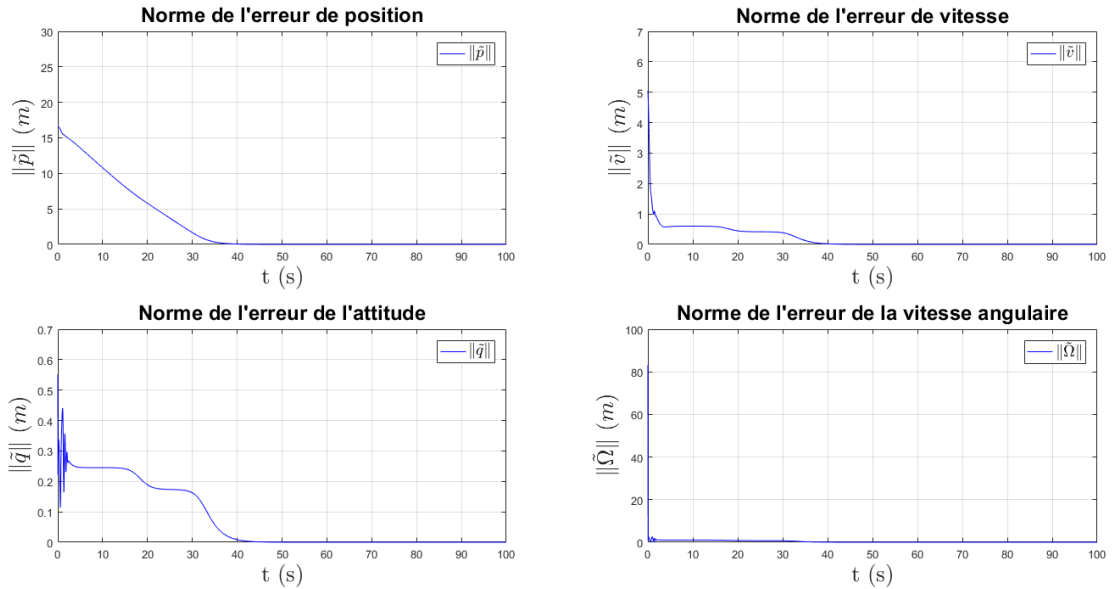


Figure 4.15: Normes des erreurs

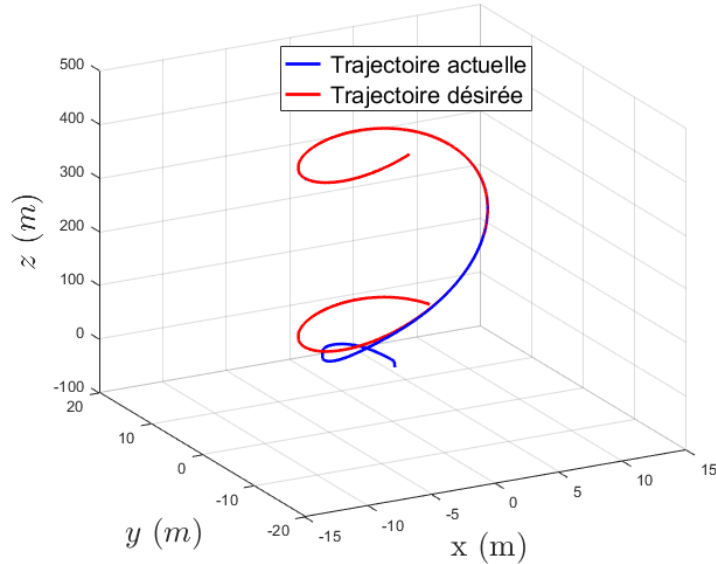


Figure 4.16: Trajectoires en 3D

4.3.3 Résultats d'implémentations

Afin d'explorer les performances en temps réel des commandes (4.39) et (4.43), deux expériences sont réalisées sur notre plateforme open source Crazyflie 2.0 décrite dans le chapitre 2.

4.3.3.1 Expérience 1: Vol stationnaire avec décollage et atterrissage automatique

Le contrôleur est mise en œuvre sur notre système de pilotage automatique. Le quadrotor effectue un suivi de trajectoire en commençant par un décollage automatique, en suivant une planification basée sur une trajectoire polynomiale, le vole stationnaire en maintenant sa position en 3D, enfin l'atterrissage automatique en suivant une planification polynomiale. L'attitude est stabilisée à $Q = Q_d$ pendant le temps de vol. La poussée totale des moteurs doit être égale à la force de gravité, elle est calculé en utilisant l'extraction présenté dans la section 4.2.2. L'expérience est réalisée avec la loi de commande (4.39) et (4.43). Les gains et les paramètres impliqués dans le contrôleur sont choisi comme suit $k_p = 2.2$, $k_v = 1.3$, $\gamma_q = 1$, $K_\Omega = \text{diag}(10000, 10000, 10500)$, $\gamma_q = 100000$. Les résultats expérimentaux sont reportés dans les Fig 4.17 - 4.21.

4.3 Contrôle de position en utilisant l'approche du Backstepping où les perturbations sont nulles et les paramètres du système sont connus

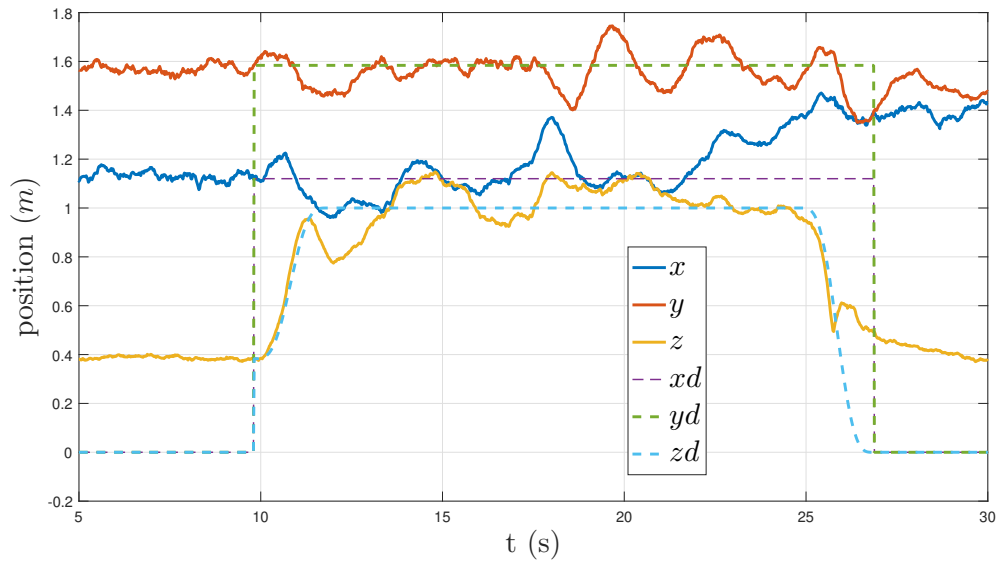


Figure 4.17: Position

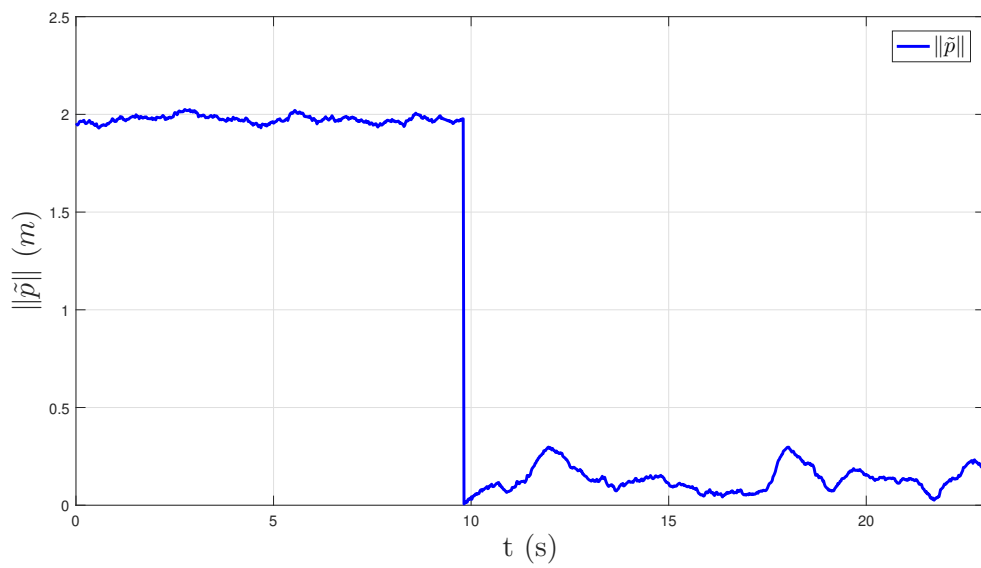


Figure 4.18: Norme de l'erreur de la position

4.3 Contrôle de position en utilisant l'approche du Backstepping où les perturbations sont nulles et les paramètres du système sont connus

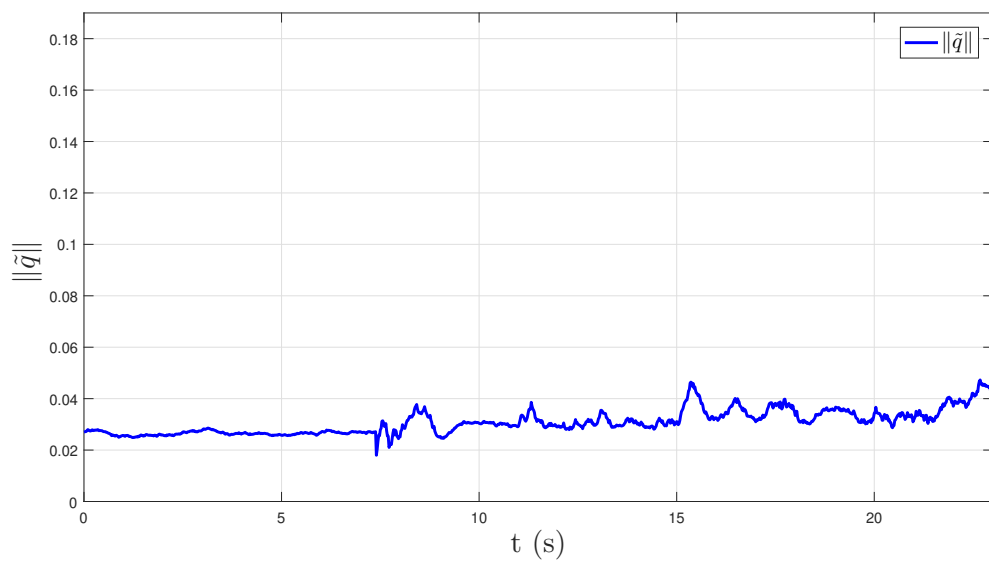


Figure 4.19: Norme de l'erreur de l'attitude

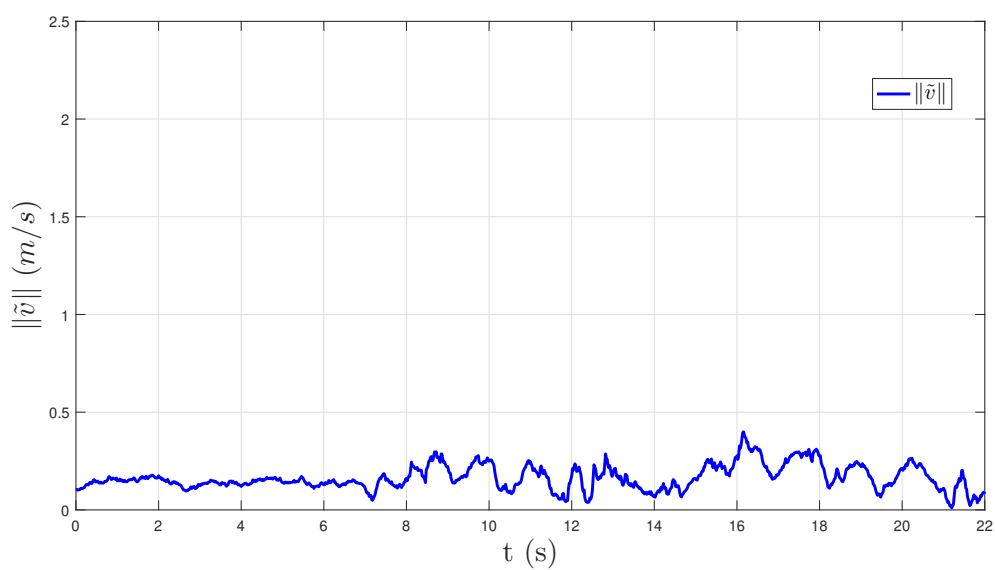


Figure 4.20: Norme de l'erreur de la vitesse

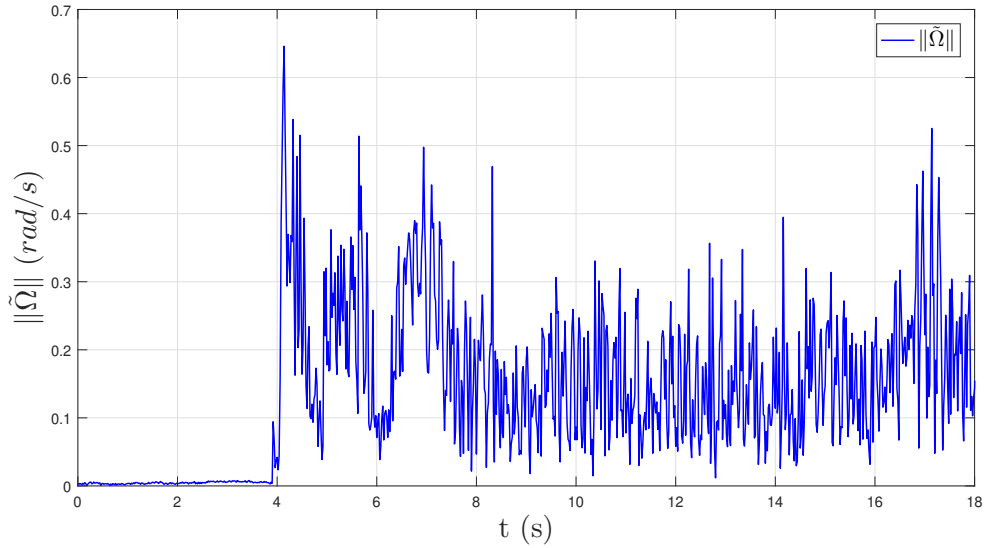


Figure 4.21: Norme de l'erreur de la vitesse angulaire

4.3.3.2 Expérience 2: Aller vers un point dans l'espace

Dans cette expérience, le quadrotor est décollé du sol et son attitude est stabilisée à $R = I_{3 \times 3}$ (c.-à-d. $\phi = 0 \text{ deg}$, $\theta = 0 \text{ deg}$ et $\psi = 0 \text{ deg}$) pendant le temps de survol. En cas de vol stationnaire, la poussée totale des moteurs doit être égale à la force de gravité. Les gains impliqués dans le contrôleur sont les mêmes utilisés dans l'expérience 1. Les résultats expérimentaux sont reportés dans les Fig 4.22 - 4.25. Les figures montrent que le contrôleur est capable de suivre une trajectoire souhaitée avec une erreur de position moyenne de 18cm , ce qui est dû à une erreur commise par le système de position de la locomotive où la précision de chaque ancre est supposée être de 10cm .

4.3 Contrôle de position en utilisant l'approche du Backstepping où les perturbations sont nulles et les paramètres du système sont connus

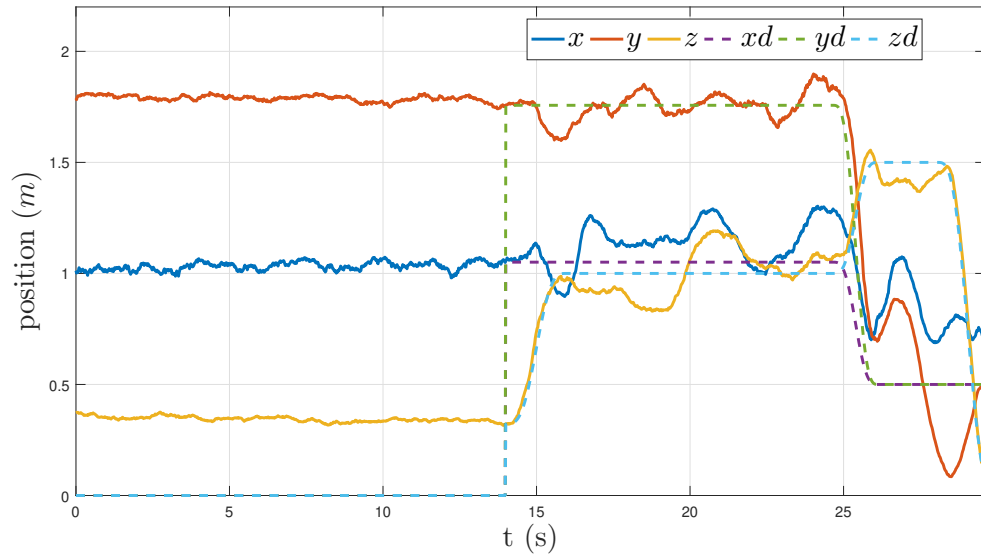


Figure 4.22: Position

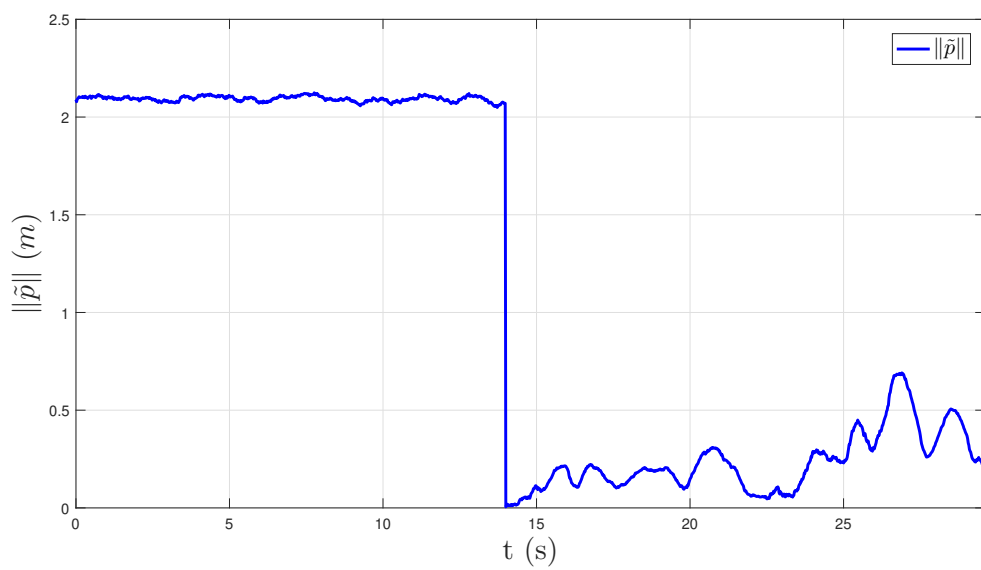


Figure 4.23: Norme de l'erreur de la position

4.4 Conclusion

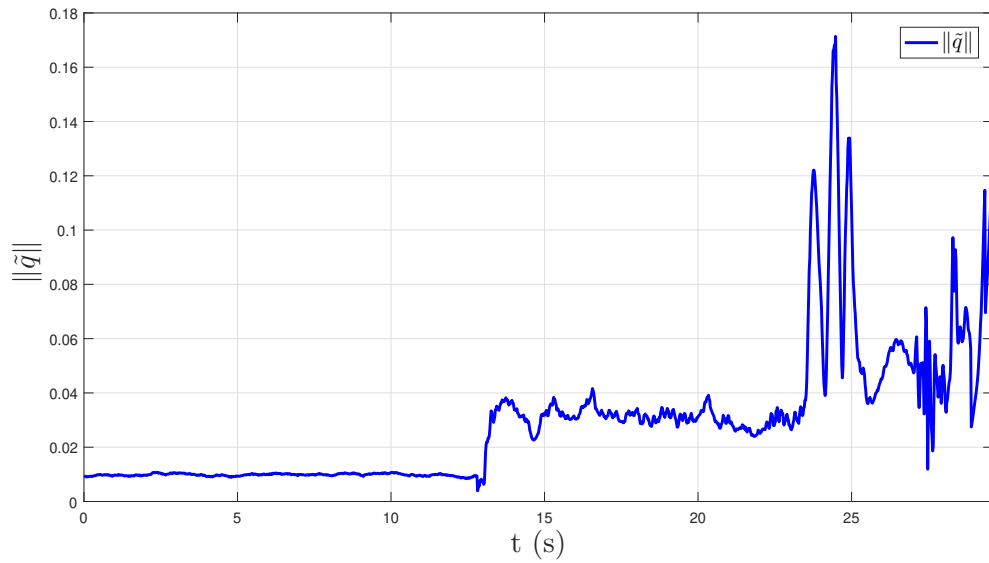


Figure 4.24: Norme de l'erreur de l'attitude

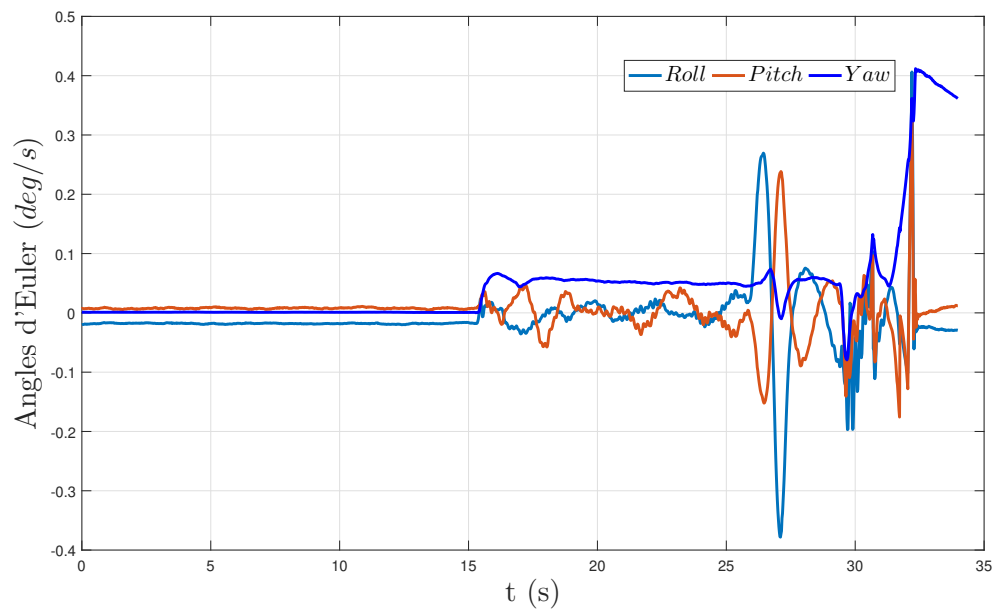


Figure 4.25: L'attitude

4.4 Conclusion

Deux approches ont été implémentées avec succès sur la plateforme open-source Crazyflie 2.0. Néanmoins l'approche simplifiée est plus simple à implémenter en

4.4 Conclusion

terme de complexité et de temps de calcul. Les résultats que nous avons obtenus ne montrent pas une grande différence entre les performances des deux approches. En raison de contraintes techniques, l'algorithme adaptatif synthétisé dans le chapitre 3 n'a pas pu être implémenté à cause d'une divergence de l'estimateur de position en raison du temps de calcul important des termes $\dot{\hat{\Omega}}$ (A.7) et $F(\Omega_s, \varpi + \hat{b}, f_{\hat{\Omega}})$ figurant dans l'expression de la commande τ et la loi d'adaptation $\dot{\hat{\Theta}}$. Le problème va être résolu dans les prochains travaux.

Conclusion générale

Le but de cette thèse est de proposer des lois de commandes robustes dédiées aux pilotage autonome des VTOL UAV. À notre connaissance, il n'y a aucun travail dans la littérature permettant d'obtenir des résultats de convergence globale pour le problème de suivi de position des VTOL UAV dans le cas où l'inertie du système est totalement inconnue et en présence d'un biais gyroscopique, une perte d'efficacité des actionneurs (LOE) et en présence aussi de perturbations externes. Dans ce contexte, nous avons focalisé notre étude sur le développement de loi de commande stables, robustes et simples à implémenter. L'efficacité du système de contrôle proposé a été confirmée lors des résultats de simulation présentés, appliqués au modèle d'un drone réaliste à ventilateur canalisé (Ducted fan) incluant le couplage et les effets aérodynamiques de friction et de traînée.

Nous avons commencé par présenter un background important pour la synthèse des lois de commande dédiées aux systèmes VTOL UAV. Nous avons commencé par donner des notions importantes de la représentation d'attitude. En suite, nous avons présenté le modèle dynamique d'un VTOL avec ses deux composantes translationnelles et rotationnelles. La dynamique rotation utilisée repose sur les quaternions car cette paramétrisation offre des avantages significatifs par rapport à la représentation $SO(3)$ en raison de ses propriétés mathématiques uniques qui donnent souvent des démonstrations simplifiées par rapport aux travaux qui se basent uniquement sur le $SO(3)$. Le choix des quaternions est justifié par un autre avantage car le quaternion unitaire est représenté par un vecteur de quatre éléments seulement au lieu d'une matrice orthogonale 3×3 , donc moins de complexité en terme de temps de calcul.

La nature cascade du modèle du système impose l'utilisation du backstepping où l'intermédiaire entre les deux dynamiques est l'accélération désirée. Cette dernière n'est pas utilisable directement dans la dynamique de rotation, pour cela nous avons illustré une méthode de reconstruction d'attitude passive, qui fait l'extraction de la poussé et de l'attitude qui seront utilisées lors de la synthèse de la loi de commande

pour la dynamique de rotation. Nous avons rappelé, à la fin du premier chapitre au travers des définitions, quelques notions sur la stabilité des systèmes physiques et des outils mathématiques, qui nous ont servies dans le présent manuscrit.

Un schéma de contrôle adaptatif pour le suivi de trajectoire d'un VTOL UAV a été présenté dans le chapitre 3, en considérant que la matrice d'inertie est totalement inconnue et qu'un biais gyroscopique et une perte d'efficacité des actionneurs sont présent et influant sur le système. De plus le drone est supposé sujet à des couples de perturbations externes. Le contrôleur a été conçu de manière hiérarchique en suivant la stratégie du backstepping.

En premier lieu la dynamique de translation est stabilisée par une commande intermédiaire, qui est l'accélération désirée. Nous nous assurons que cette entrée de commande virtuelle soit *a priori* bornée et au moins deux fois différentiables. Cela représente un ingrédient clé garantissant l'existence d'une solution à l'algorithme d'extraction de l'attitude et de la poussée, ainsi que la limitation des entrées de commande réelles. En conséquence, le schéma de contrôle proposé garantit une poussée a priori bornée.

Pour toutes les lois d'adaptation nous avons utilisé le mécanisme de projection afin de garantir un certain niveau de robustesse en évitant que les estimations des paramètres ne dérivent pas dans le temps en présence de mesures bruitées. Cela permettra également, dans certains cas, d'empêcher les estimations de paramètres de s'annuler et d'éviter les problèmes de singularité dans l'expression du couple de commande.

Nous avons présenté dans le chapitre 2 la plateforme open source crazyflie 2.0 que nous avons utilisé pour faire l'implémentation des lois de commande démontrées dans le chapitre 4 en donnant l'architecture matérielle et logicielle ainsi que les caractéristiques du système. Le positionnement est assuré par un dispositif nommé LPS (Localization system) qui se base sur la technologie UWB (Ultra Wide Band)

Les résultats d'implémentation de deux algorithmes de commande de suivi de trajectoire sur la plateforme crazyflie 2.0 sont présentés dans le chapitre 4. Les deux contrôleurs s'appuient sur l'estimateur EKF pour fournir l'état actuel du système. L'EKF traite les mesures de l'IMU et les mesures du système de positionnement LPS pour l'estimation de l'attitude et de la position. Les résultats expérimentaux obtenus montrent des performances acceptables que ce soit pour la stabilisation d'attitude ou pour le contrôle de position. Cependant, le contrôleur basé sur le backstepping présente une charge de calcul beaucoup plus lourde comparé à celle de

l'approche simplifiée. Néanmoins, rajouter un module de mesure d'altitude comme le "Flow deck" est souhaitable dans les travaux futurs afin de diminuer les erreurs de mesure de l'altitude et améliorer les résultats de suivi.

Nous avons tenté une implémentation de l'algorithme de contrôle adaptatif synthétisé dans le chapitre 3 mais des problèmes techniques dus à la quantité de calcul, ont empêché la réussite de l'opération.

Perspectives

Nous proposons, comme perspective de ce travail une approche de contrôle de position qui élimine la nécessité de mesurer l'attitude du système en utilisant les mesures vectorielles directement dans la loi de contrôle. Nous utilisons spécifiquement un magnétomètre et un accéléromètre pour fournir les deux mesures vectorielles. Où l'accéléromètre sera utilisé pour mesurer l'accélération apparente du système, plutôt que le vecteur de gravité uniquement.

Il sera aussi très intéressant de résoudre le problème technique afin d'accomplir l'implémentation de l'algorithme adaptatif. Un autre travail intéressant serait la conception de stratégies de suivi de trajectoire avec des techniques d'évitement d'obstacles, permettant une navigation plus autonome des UAVs.

Annexe A

Annexe

A.1 Calcul de $\dot{\bar{\Omega}}$ et $J\dot{\bar{\Omega}}$

A.1.1 Calcul de $\dot{\bar{\Omega}}$

Rappelons l'expression de $\bar{\Omega}$

$$\bar{\Omega} = \frac{-1}{\gamma_q} \Psi_1 \Gamma_v \tilde{v} + M(\mu_d)(p_d^{(3)} + \Psi_4 \tilde{v} + \Psi_5) - K_q \tilde{q} \quad (\text{A.1})$$

Afin d'obtenir son dérivé temporel, nous nous concentrons d'abord sur l'obtention des dérivées de certains signaux. Les expressions de $\dot{\tilde{q}}$ et $\dot{\tilde{q}}$ sont réécrites sous une forme appropriée comme suit:

$$\dot{\tilde{q}} = \Upsilon_{\tilde{q}} + f_{\dot{\tilde{q}}} b \quad (\text{A.2})$$

avec $\Upsilon_{\tilde{q}} := \frac{1}{2} \tilde{q}_0 (\Omega_s - \Omega_d) + \frac{1}{2} S(\tilde{q}) (\Omega_d + \Omega_s)$ et $f_{\dot{\tilde{q}}} := -\frac{1}{2} (S(\tilde{q}) + \tilde{q}_0 I_{3 \times 3})$.

À la lumière de (A.2), $\dot{\tilde{q}}$ is rewritten as

$$\begin{aligned} \dot{\tilde{q}} &= S(\hat{e}_3) \dot{\tilde{q}} + \dot{\tilde{q}}_0 \hat{e}_3 = S(\hat{e}_3) \dot{\tilde{q}} + \frac{1}{2} \hat{e}_3 \tilde{q}^T (\Omega_d - \Omega) \\ &= S(\hat{e}_3) (\Upsilon_{\tilde{q}} + f_{\dot{\tilde{q}}} b) + \frac{1}{2} \hat{e}_3 \tilde{q}^T (\Omega_d - \Omega_s + b) := \Upsilon_{\tilde{q}} + f_{\dot{\tilde{q}}} b \end{aligned} \quad (\text{A.3})$$

où $\Upsilon_{\tilde{q}} := S(\hat{e}_3) \Upsilon_{\tilde{q}} + \frac{1}{2} \hat{e}_3 \tilde{q}^T (\Omega_d - \Omega_s)$ et $f_{\dot{\tilde{q}}} := S(\hat{e}_3) f_{\dot{\tilde{q}}} + \frac{1}{2} \hat{e}_3 \tilde{q}^T$.

La dérivée du temps du terme $\frac{-1}{\gamma_q} \Psi_1 \Gamma_v \tilde{v}$ est donné par

$$\begin{aligned} \frac{d}{dt} \left(\frac{-1}{\gamma_q} \Psi_1 \Gamma_v \tilde{v} \right) &= \frac{2}{\gamma_q u_t} S(\tilde{q}) R \Gamma_v \tilde{v} (\mu_d - g \hat{e}_3)^T \dot{\mu}_d - \frac{2}{\gamma_q} u_t S(R \Gamma_v \tilde{v}) \dot{\tilde{q}} \\ &\quad - \frac{2}{\gamma_q} u_t S(\tilde{q}) S(\Omega) R \Gamma_v \tilde{v} + \frac{2}{\gamma_q} u_t S(\tilde{q}) R \Gamma_v \dot{\tilde{v}} := \Upsilon_{\dot{\Psi}} + f_{\dot{\Psi}} b \end{aligned} \quad (\text{A.4})$$

où

$$\begin{aligned} \Upsilon_{\dot{\Psi}} &:= \frac{2}{\gamma_q u_t} S(\tilde{q}) R \Gamma_v \tilde{v} (\mu_d - g \hat{e}_3)^T \dot{\mu}_d + \frac{2}{\gamma_q} u_t S(\tilde{q}) R \Gamma_v \dot{\tilde{v}} \\ &\quad - \frac{2}{\gamma_q} u_t S(\tilde{q}) S(\Omega_s) R \Gamma_v \tilde{v} - \frac{2}{\gamma_q} u_t S(R \Gamma_v \tilde{v}) \Upsilon_{\tilde{q}} \end{aligned} \quad (\text{A.5})$$

et

$$f_{\dot{\Psi}} := -\frac{2}{\gamma_q} u_t S(\tilde{q}) S(R \Gamma_v \tilde{v}) - \frac{2}{\gamma_q} u_t S(R \Gamma_v \tilde{v}) f_{\dot{\tilde{q}}} \quad (\text{A.6})$$

A.1 Calcul de $\dot{\bar{\Omega}}$ et $J\dot{\bar{\Omega}}$

Compte tenu des résultats ci-dessus, la dérivée temporelle de $\bar{\Omega}$ est donnée par la forme compacte suivante:

$$\dot{\bar{\Omega}} := \varpi + f_{\dot{\bar{\Omega}}}b \quad (\text{A.7})$$

avec

$$\begin{aligned} \varpi := & f_D(\mu_d, p_d^{(3)} + \Psi_5)\dot{\mu}_d - k_p f_D(\mu_d, \Gamma_v^{-1} D_t(\tilde{p})\tilde{v})\dot{\mu}_d + M(\mu_d)(p_d^{(4)} + \dot{\Psi}_5) \\ & - k_p M(\mu_d)\Gamma_v^{-1} (G(\tilde{p}, \tilde{v})\tilde{v} + D_t(\tilde{p})\dot{\tilde{v}}) + \Upsilon_{\dot{\Psi}} - K_q \Upsilon_{\dot{\bar{q}}} \end{aligned}$$

et

$$f_{\dot{\bar{\Omega}}} := f_{\dot{\Psi}} - K_q f_{\dot{\bar{q}}}$$

où

$$\dot{\Psi}_5 := k_p k_v G(\tilde{v}, \Gamma_v^{-1} \tanh(\tilde{p}))\dot{\tilde{v}} + k_p k_v D_t(\tilde{v})\Gamma_v^{-1} D_t(\tilde{p})\tilde{v}$$

Le mapping $G : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ est défini de telle sorte que, pour tout $u := [u_1 \ u_2 \ u_3]^T \in \mathbb{R}^3$ et $w := [w_1 \ w_2 \ w_3]^T \in \mathbb{R}^3$, on a

$$G(u, w) = \text{diag} \left(\frac{\partial \text{sech}^2(u_i)}{\partial u_i} w_i \right)_{i=1,2,3} = \text{diag}(\alpha_1, \alpha_2, \alpha_3)$$

où $\alpha_i := -2\text{sech}^2(u_i) \tanh(u_i) w_i$

La fonction $f_D : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ est défini comme suit:

$$f_D(\mu_d, w) := \frac{\partial}{\partial \mu_d} M(\mu_d)w = \gamma_M^{-1} M(\mu_d)w \frac{\partial}{\partial \mu_d} (\gamma_M) + \gamma_M \frac{\partial}{\partial \mu_d} (\gamma_M^{-1} M(\mu_d)w) \quad (\text{A.8})$$

où

$$\gamma_M := \|\eta\|^{-2} \xi^{-1}, \quad \xi := \|\eta\| + g - \mu_{d_3}$$

A.1 Calcul de $\dot{\tilde{\Omega}}$ et $J\dot{\tilde{\Omega}}$

La dérivée partielle impliquée dans le premier terme de (A.8) est donnée par

$$\begin{aligned}\frac{\partial}{\partial \mu_d} (\gamma_M) &= \gamma_M^2 (2\eta^T \|\eta\| - 2\eta^T \hat{e}_3^T \eta + \|\eta\| \eta^T - \|\eta\|^2 \hat{e}_3^T) \\ &= -\gamma_M^2 (3\eta^T (\|\eta\| - \hat{e}_3^T \eta) + \hat{e}_3^T (\eta \eta^T - \eta^T \eta I_{3 \times 3})) \\ &= \gamma_M^2 (g \hat{e}_3 - \mu_d)^T (3\xi I_{3 \times 3} + S(\hat{e}_3)S(g \hat{e}_3 - \mu_d))\end{aligned}$$

La dérivée partielle du second terme impliqué dans (A.8) est obtenue comme suit

$$\frac{\partial}{\partial \mu_d} (\gamma_M^{-1} M(\mu_d) w) := \Lambda_1(\mu_d, w) + \Lambda_2(\mu_d, w) + \Lambda_3(\mu_d, w)$$

où

$$\begin{aligned}\Lambda_1(\mu_d, w) &:= \frac{\partial}{\partial \mu_d} \begin{bmatrix} -\mu_{d1} \mu_{d2} w_1 - \mu_{d2}^2 w_2 + \mu_{d2} \xi w_3 \\ \mu_{d1}^2 w_1 + \mu_{d1} \mu_{d2} w_2 - \mu_{d1} \xi w_3 \\ \mu_{d2} u_t w_1 - \mu_{d1} u_t w_2 \end{bmatrix} \\ &= \begin{bmatrix} -\mu_{d2} w_1 & -\mu_{d1} w_1 - 2w_2 \mu_{d2} + \xi w_3 & 0 \\ 2w_1 \mu_{d1} + w_2 \mu_{d2} - \xi w_3 & \mu_{d1} w_2 & 0 \\ -u_t w_2 & u_t w_1 & 0 \end{bmatrix} \\ \Lambda_2(\mu_d, w) &:= \frac{\partial}{\partial \mu_d} \begin{bmatrix} u_t \xi w_2 \\ -u_t \xi w_1 \\ \mu_{d2} u_t w_1 - \mu_{d1} u_t w_2 \end{bmatrix} = \begin{bmatrix} \xi w_2 & -\xi w_1 & \mu_{d2} w_1 - \mu_{d1} w_2 \end{bmatrix}^T \alpha_1(\mu_d)^T \\ \Lambda_3(\mu_d, w) &:= \frac{\partial}{\partial \mu_d} \begin{bmatrix} u_t \xi w_2 + \mu_{d2} \xi w_3 \\ -u_t \xi w_1 - \mu_{d1} \xi w_3 \\ 0 \end{bmatrix} = \begin{bmatrix} u_t w_2 + \mu_{d2} w_3 & -u_t w_1 - \mu_{d1} w_3 & 0 \end{bmatrix}^T \alpha_2(\mu_d)^T\end{aligned}$$

avec $\alpha_1(\mu_d) := \frac{\partial}{\partial \mu_d} u_t = \eta / \|\eta\|$ and $\alpha_2 := \frac{\partial}{\partial \mu_d} \xi = \alpha_1 - \hat{e}_3$.

Enfin l'expression de $f_D(\mu_d, w)$

$$f_D(\mu_d, w) := \gamma_M^{-1} M(\mu_d) w f_\gamma + \gamma_M \Lambda(\mu_d, w) \quad (\text{A.9})$$

avec $\Lambda(\mu_d, w) = \Lambda_1(\mu_d, w) + \Lambda_2(\mu_d, w) + \Lambda_3(\mu_d, w)$

A.1.2 Calcul de $J\dot{\hat{\Omega}}$

En utilisant (4.4) and (A.7) the term $J\dot{\hat{\Omega}}$ est réécrit en termes de termes connus et inconnus. De (3.34) on a

$$\begin{aligned}
J\dot{\hat{\Omega}} &= J(\dot{\hat{\Omega}} - (\dot{\hat{\Omega}} + \dot{\hat{b}})) = -S(\Omega)J\Omega - J(\varpi + f_{\dot{\hat{\Omega}}}b + \dot{\hat{b}}) + \Delta\tau + d \\
&= -S(\Omega_s - b)J(\Omega_s - b) - J(\varpi + f_{\dot{\hat{\Omega}}}b + \dot{\hat{b}}) + \Delta\tau + d \\
&= -S(\Omega_s)J\Omega_s + (S(b)J - S(Jb))\Omega_s - S(b)Jb - J(\varpi + \dot{\hat{b}}) - Jf_{\dot{\hat{\Omega}}}b + \Delta\tau + d \\
&= -\left(S(\Omega_s)M_1(\Omega_s) + M_1(\varpi + \dot{\hat{b}})\right)\theta_1 + M_2(\Omega_s)\theta_2 - \theta_3 - M_3(f_{\dot{\hat{\Omega}}})\theta_4 + \Delta\tau + d \\
&:= F(\Omega_s, \varpi + \dot{\hat{b}}, f_{\dot{\hat{\Omega}}})\Theta + \Delta\tau + d, \tag{A.10}
\end{aligned}$$

où

$$\Theta := [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4] \in \mathbb{R}^{36} \tag{A.11}$$

avec $\theta_1 := [J_{11} \ J_{22} \ J_{33} \ J_{23} \ J_{13} \ J_{12}]^T$ est composé des éléments de la matrice d'inertie J . Le vecteur $\theta_2 := [\theta_{21} \ \theta_{22} \ \theta_{23} \ \theta_{24} \ \theta_{25} \ \theta_{26} \ \theta_{27} \ \theta_{28} \ \theta_{29}]$ est composé des éléments de la matrice inconnue $(S(b)J - S(Jb))$. Le vecteur θ_3 est défini comme $\theta_3 := S(b)Jb$ et le vecteur θ_4 , est donné par l'équation (A.18), est composé de la combinaison des éléments de J and b . Les mappings $M_1, M_2 : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times n}$, agissant sur un vecteur $u = [u_1, u_2, u_3]^T$ sont définis dans (A.12), et le mapping $M_3 : \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^{3 \times n}$ agissant sur la matrice 3-fois-3, $z = [z_{ij}]_{i,j=1\dots 3}$, est défini dans (A.14). Nous avons utilisé le fait que $Ju = M_1(u)\theta_1$, $(S(b)J - S(Jb))u = M_2(u)\theta_2$, et $Jf_{\dot{\hat{\Omega}}}b = M_3(f_{\dot{\hat{\Omega}}})\theta_4$.

$$M_1(u) = \begin{bmatrix} u_1 & 0 & 0 & 0 & u_3 & u_2 \\ 0 & u_2 & 0 & u_3 & 0 & u_1 \\ 0 & 0 & u_3 & u_2 & u_1 & 0 \end{bmatrix} \tag{A.12}$$

$$M_2(u) = \begin{bmatrix} u_1 & u_2 & u_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_1 & u_2 & u_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u_1 & u_2 & u_3 \end{bmatrix} \tag{A.13}$$

$$M_3(z) = \begin{bmatrix} M_{3_{s_2}}(z) & M_{3_{s_1}}(z) & M_{3_{s_3}}(z) \end{bmatrix} \quad (\text{A.14})$$

où

$$M_{3_{s_1}}(z) = \begin{bmatrix} z_{11} & z_{21} & z_{31} & z_{12} & z_{22} & z_{32} \\ 0 & z_{11} & 0 & 0 & z_{12} & 0 \\ 0 & 0 & z_{11} & 0 & 0 & z_{12} \end{bmatrix} \quad (\text{A.15})$$

$$M_{3_{s_2}}(z) = \begin{bmatrix} z_{13} & z_{23} & z_{33} & 0 & 0 & 0 \\ 0 & z_{13} & 0 & z_{21} & z_{31} & z_{22} \\ 0 & 0 & z_{13} & 0 & z_{21} & 0 \end{bmatrix} \quad (\text{A.16})$$

$$M_{3_{s_3}}(z) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ z_{32} & z_{23} & z_{33} & 0 & 0 & 0 \\ z_{22} & 0 & z_{23} & z_{31} & z_{32} & z_{33} \end{bmatrix} \quad (\text{A.17})$$

$$\theta_4 = \begin{bmatrix} \theta_{4_{s_1}} & \theta_{4_{s_2}} \end{bmatrix}^T \quad (\text{A.18})$$

où

$$\theta_{4_{s_1}} = \begin{bmatrix} J_{11}b_1 & J_{12}b_1 & J_{13}b_1 & J_{11}b_2 & J_{12}b_2 & J_{13}b_2 & J_{11}b_3 & J_{12}b_3 & J_{13}b_3 \end{bmatrix} \quad (\text{A.19})$$

$$\theta_{4_{s_2}} = \begin{bmatrix} J_{22}b_1 & J_{23}b_1 & J_{22}b_2 & J_{23}b_2 & J_{22}b_3 & J_{23}b_3 & J_{33}b_1 & J_{33}b_2 & J_{33}b_3 \end{bmatrix} \quad (\text{A.20})$$

Références

- [1] S. Berkane, “Hybrid attitude control and estimation on $SO(3)$,” Thèse de doctorat, *University of Western Ontario*, 2017.
- [2] Krstic Peter, *Spacecraft Attitude Dynamics*. (Dover Books on Engineering). Dover Publications, 1986.
- [3] J.-Y. Wen and K. Kreutz-Delgado, “The attitude control problem,” *CRC press*, 1994.
- [4] F. Lizarralde and J. T. Wen, “Attitude control without angular velocity measurement: A passivity approach,” *IEEE transactions on Automatic Control*, vol. 41, no. 3, pp. 468–472, 1996.
- [5] W. Luo, Y.-C. Chu, and K.-V. Ling, “Inverse optimal adaptive control for attitude tracking of spacecraft,” *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1639–1654, 2005.
- [6] A. Tayebi and S. McGilvray, “Attitude stabilization of a vtol quadrotor aircraft,” *IEEE Transactions on Control Systems Technology*, vol. 14, pp. 562–571, May 2006.
- [7] A. Tayebi, “Unit quaternion-based output feedback for the attitude tracking problem,” *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1516–1520, 2008.
- [8] A. Abdessameud and A. Tayebi, “Global trajectory tracking control of vtol-uavs without linear velocity measurements,” *Automatica*, vol. 46, no. 6, pp. 1053–1059, 2010.
- [9] A. Roberts and A. Tayebi, “Adaptive position tracking of vtol uavs,” *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 129–142, 2011.

- [10] D. Cabecinhas, R. Cunha, and C. Silvestre, “A globally stabilizing path following controller for rotorcraft with wind disturbance rejection,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 708–714, 2015.
- [11] K. Lu and Y. Xia, “Adaptive attitude tracking control for rigid spacecraft with finite-time convergence,” *Automatica*, vol. 49, no. 12, pp. 3591–3599, 2013.
- [12] Q. Shen, D. Wang, S. Zhu, and E. K. Poh, “Inertia-free fault-tolerant spacecraft attitude tracking using control allocation,” *Automatica*, vol. 62, pp. 114–121, 2015.
- [13] R. C. Avram, X. Zhang, and J. Muse, “Nonlinear adaptive fault-tolerant quadrotor altitude and attitude tracking with multiple actuator faults,” *IEEE Transactions on Control Systems Technology*, vol. 62, no. 2, pp. 701–707, 2018.
- [14] X. Tang, G. Tao, and S. M. Joshi, “Adaptive actuator failure compensation for nonlinear mimo systems with an aircraft control application,” *Automatica*, vol. 43, no. 11, pp. 1869–1883, 2007.
- [15] M. D. Shuster, “A survey of attitude representations,” *Journal of Astronautical Sciences*, vol. 41, no. 4, pp. 439–517, 1993.
- [16] Murray, R. M., Sastry, S. S., and Zexiang, “A Mathematical Introduction to Robotic Manipulation.,” *IEEE Transactions on Automatic control*, vol. 36, no. 10, pp. 1148–1162, 1991.
- [17] A. D. Roberts, “Attitude Estimation and Control of VTOL UAVs,” PhD thesis *University of Western Ontario*, 2011.
- [18] M. Wang, “Attitude Control of a Quadrotor UAV: Experimental Results,” Master thesis *Lakehead University*, 2015.
- [19] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” *Matrix*, vol. 58, pp. 15–16, 2006.
- [20] D. Koditschek, “Application of a new Lyapunov function to global adaptive attitude tracking,” Proceedings of the 27th IEEE Conference on Decision and Control (CDC) , pp. 5233–5238, Dec 1988.
- [21] Nicholson, *Linear Algebra with Applications*. PWS Pub. Co. (Boston), 1995.

- [22] J. Pffimlin, P. Soueres, and T. Hamel, “Position control of a ducted fan vtol uav in crosswind,” *International Journal of Control*, vol. 80, no. 5, pp. 666–683, 2007.
- [23] R. Reynolds, “Quaternion Parameterization and a simple algorithm for global attitude estimation,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 4, pp. 669–672, 1998.
- [24] A. Roberts and A. Tayebi, “Adaptive position tracking of vtol uavs,” Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, pp. 5233–5238, Dec 2009.
- [25] H.K. Khalil, *Nonlinear systems (2nd ed.)*. Prentice-Hall, 1996.
- [26] J.-J. Slotine et W. Li, *Applied nonlinear control*. Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- [27] P.A. Ioannou et J. Sun, *Robust Adaptive Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1996.
- [28] A. Boubakir, “Contribution à la commande sans modèle Des systèmes non linaires avec applications ,”, Thèse de doctorat, *Ecole Nationale Polytechnique d Alger*, 2014.
- [29] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and adaptive control design*. Wiley, 1995.
- [30] Lim, H., Park, J., Lee, D., Kim, H. J., “Build your own quadrotor: Open-source projects on unmanned aerial vehicles,” *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 33–45, 2012.
- [31] J. Singh, “Study and Development of Target Following Capability on Nano-Sized Unmanned Aerial Vehicles,” Master thesis *ETH Zurich*, 2017.
- [32] B. Landry, “Planning and Control for Quadrotor Flight through Cluttered Environments,” Master thesis *Massachusetts Institute of Technology*, 2015.
- [33] M. Greiff, “Modelling and Control of the Crazyflie Quadrotor for Aggressive and Autonomous Flight by Optical Flow Driven State Estimation,” Master thesis *Lund University*, 2017.

- [34] F. POIRIER, “Control of the quadcopter Crazyflie,” Master thesis *The University of Manchester*, 2018.
- [35] Bitcraze. *Bitcraze UWB LPS* (visited on 02-2019). URL:<https://www.bitcraze.io/loco-pos-system/>.
- [36] T. J. Koo and S. Sastry, “Output tracking control design of a helicopter model based on approximate linearization,” Proceedings of the 37th IEEE Conference on Decision and Control, (Tampa, FL, USA), pp. 3635–3640, Dec 1998.
- [37] E. Frazzoli, M. A. Dahleh, and E. Feron, “Trajectory tracking control design for autonomous helicopters using a backstepping algorithm,” American Control Conference, (Chicago, IL, USA), pp. 4102–4107, June 2000.
- [38] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, “Dynamic modeling and configuration stabilization for an x4-flyer,” IFAC World Congress, (Barcelona, Spain), pp. 4102–4107, 2002.
- [39] A. P. Aguiar and J. P. Hespanha, “Position tracking of underactuated vehicles,” American Control Conference, (Denver, CO, USA), pp. 1988–1993, June 2003.
- [40] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, “A control approach for thrust-propelled underactuated vehicles and its application to vtol drones,” *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1837–1853, 2009.
- [41] A. Roberts and A. Tayebi, “A new position regulation strategy for vtol uavs using imu and gps measurements,” *Automatica*, vol. 49, no. 2, pp. 433–440, 2013.
- [42] A. Roza and M. Maggiore, “A class of position controllers for underactuated vtol vehicles,” *IEEE Transactions on Automatic control*, vol. 59, no. 9, pp. 2580–2585, 2014.
- [43] RR. Benrezki, A. Tayebi and M. Tadjine, “Adaptive trajectory tracking control for VTOL-UAVs with unknown inertia, gyro-bias, and actuator LOE ,” *International Journal of Robust and Nonlinear Control*, vol. 28, no. 17, pp. 5247–5261, 2018.
- [44] A. Z. Sedor, “UWB Localization for Autonomous Indoor Position Control of VTOL UAVs,” Master thesis *Lakehead University*, 2018.